# Projection Methods for Clustering and Semi-supervised Classification

David Paul Hofmeyr, B.Sc. (Hons), M.Sc., M.Res.

Submitted for the degree of Doctor of Philosophy
at Lancaster University
November, 2015

# Abstract

This thesis focuses on data projection methods for the purposes of clustering and semi-supervised classification, with a primary focus on clustering. A number of contributions are presented which address this problem in a principled manner; using projection pursuit formulations to identify subspaces which contain useful information for the clustering task. Projection methods are extremely useful in high dimensional applications, and situations in which the data contain irrelevant dimensions which can be counter-informative for the clustering task. The final contribution addresses high dimensionality in the context of a data stream. Data streams and high dimensionality have been identified as two of the key challenges in data clustering.

The first piece of work is motivated by identifying the minimum density hyperplane separator in the finite sample setting. This objective is directly related to the problem of discovering clusters defined as connected regions of high data density, which is a widely adopted definition in non-parametric statistics and machine learning. A thorough investigation into the theoretical aspects of this method, as well as the practical task of solving the associated optimisation problem efficiently is presented. The proposed methodology is applied to both clustering and semi-supervised classification problems, and is shown to reliably find low density hyperplane separators in both contexts.

The second and third contributions focus on a different approach to clustering based on graph cuts. The minimum normalised graph cut objective has gained considerable attention as relaxations of the objective have been developed, which make them solvable for reasonably well sized problems. This has been adopted by the highly popular spectral clustering methods. The second piece of work focuses on identifying the optimal subspace in which to perform spectral clustering, by minimising the second eigenvalue of the graph Laplacian for a graph defined over the data within that subspace. A rigorous treatment of this objective is presented, and an algorithm is proposed for its optimisation. An approximation method is proposed which allows this method to be applied to much larger problems than would otherwise be possible. An extension of this work deals with the spectral projection pursuit method for semi-supervised classification.

The third body of work looks at minimising the normalised graph cut using hyperplane separators. This formulation allows for the exact normalised cut to be computed, rather than the spectral relaxation. It also allows for a computationally efficient method for optimisation. The asymptotic properties of the normalised cut based on a hyperplane separator are investigated, and shown to have similarities with the clustering objective based on low density separation. In fact, both the methods in the second and third works are shown to be connected with the first, in that all three have the same solution asymptotically, as their relative scaling parameters are reduced to zero.

The final body of work addresses both problems of high dimensionality and incremental clustering in a data stream context. A principled statistical framework is adopted, in which clustering by low density separation again becomes the focal objective. A divisive hierarchical clustering model is proposed, using a collection of low density hyperplanes. The adopted framework provides well founded methodology for determining the number of clusters automatically, and also identifying changes in the data stream which are relevant to the clustering objective. It is apparent that no existing methods can make both of these claims.

# Acknowledgements

To begin with I would like to thank my supervisors. Nicos, your enthusiasm for the work we have done during my thesis has been a great source of motivation. Idris, you have always provided an encouraging and somehow calming perspective on my research, and research in general. I have learned a lot from both of you, and I am extremely grateful. Thanks too for putting up with me these past few years, I know that my repeatedly ignoring your advice and comments can't always have gone unnoticed.

I would also like to thank both the EPSRC and the Oppenheimer Memorial Trust for providing funding during my Ph.D.

To Jon, Idris and Kevin; the STOR-i doctoral training centre has been the ideal environment in which to do my Ph.D. You've outdone yourselves. Thank you.

To the friends I have made during my time here in Lancaster, in particular my year group in STOR-i, you've always provided sufficient distractions from work to make the Ph.D seem manageable, even when things have been difficult. The whole process could not have been nearly as enjoyable without you guys.

To my family, you have been continuous sources of support and encouragement and I will forever be grateful. The admittedly too infrequent chats, though perhaps seemingly just chats, have sometimes felt like the only thing keeping me going.

Finally, to Aeysha, your love and encouragement has been constant and I don't know if I could have managed without that. There are a lot of things I could say, but I think what captures it well (at least it does for me), is that you stubbornly refuse to ever accept that I could have shortcomings; you always manage to find some other explanation for why I could be struggling. Thank you, for your oblivion, for everything.

# Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

David Paul Hofmeyr

# Contents

**Abstract** iii

**Acknowledgements** v

**Declaration** vii

**Thesis Details** xiii

**List of Figures** xiii

**List of Tables** xvii

**1 Introduction** 1
- 1 Clustering . . . 4
  - 1.1 Centroid-Based Clustering . . . 5
  - 1.2 Connectivity-Based Clustering . . . 6
  - 1.3 Graph Partitioning Based Clustering . . . 8
  - 1.4 Density-Based Clustering . . . 10
  - 1.5 Model-Based Clustering . . . 11
- 2 Semi-supervised Classification . . . 12
  - 2.1 Low Density Separation Methods . . . 13
  - 2.2 Graph Partition Based Methods . . . 14
- 3 Challenges in Data Clustering . . . 15
  - 3.1 High Dimensionality . . . 15
  - 3.2 Data Streams . . . 17
- 4 Focus of Thesis . . . 19
  - 4.1 Contributions . . . 19

**2 Minimum Density Hyperplane: An Unsupervised and Semi-supervised Classifier** 21
- 1 Introduction . . . 22
- 2 Problem Formulation . . . 25
  - 2.1 Clustering . . . 26

2.2 Semi-Supervised Classification . . . . . 30
3 Connection to Maximum Margin Hyperplanes . . . . . 31
3.1 MDP$^2$ for Clustering . . . . . 34
3.2 MDP$^2$ for Semi-Supervised Classification . . . . . 35
4 Estimation of Minimum Hyperplanes . . . . . 36
4.1 Computational Complexity . . . . . 38
5 Experimental Results . . . . . 38
5.1 Clustering . . . . . 39
5.2 Semi-Supervised Classification . . . . . 45
5.3 Summary of Experimental Results . . . . . 48
6 Conclusions . . . . . 49
Appendix. Proof of Theorem 1 . . . . . 50
7 Proof of Theorem 1 . . . . . 50

**3 Projection Pursuit Based on Spectral Connectivity** **54**

**A. Minimum Spectral Connectivity Projection Pursuit for Unsupervised Classification** **55**
1 Introduction . . . . . 55
2 Related Work . . . . . 58
3 Background on Spectral Clustering . . . . . 59
4 Projection Pursuit for Spectral Connectivity . . . . . 61
4.1 Continuity and Differentiability . . . . . 62
4.2 Minimising $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$. . . . . 67
4.3 Computing Similarities . . . . . 69
4.4 Correlated and Orthogonal Projections . . . . . 71
5 Connection with Maximal Margin Hyperplanes . . . . . 74
6 Speeding up Computation using Microclusters . . . . . 79
7 Experimental Results . . . . . 86
7.1 Details of Implementation . . . . . 87
7.2 Clustering Results . . . . . 89
7.3 Summarising Clustering Performance . . . . . 93
7.4 Sensitivity Analysis . . . . . 95
8 Conclusions . . . . . 98
Appendix. Derivatives . . . . . 100

**B. Semi-supervised Spectral Connectivity Projection Pursuit** **103**
1 Introduction . . . . . 103
2 Spectral Clustering . . . . . 105
3 Methodology . . . . . 105
3.1 Computational Complexity . . . . . 110
4 Experiments . . . . . 111
5 Conclusions . . . . . 112

Appendix. Proofs . . . . . 114

**4 Clustering by Minimum Cut Hyperplanes** **118**
1 Introduction . . . . . 118
2 Problem Formulation . . . . . 120
2.1 Background on Normalised Graph Cuts . . . . . 121
2.2 Normalised Cuts Across Hyperplanes . . . . . 121
2.3 Connection with Maximum Margin Hyperplanes . . . . 126
3 Methodology . . . . . 128
3.1 Optimal NCut of the Marginal Data Set $v \cdot \mathcal{X}$ . . . . . 128
3.2 Optimising $\Phi(v|\mathcal{X})$ . . . . . 130
3.3 Beyond Bi-partitioning . . . . . 133
4 Experimental Results . . . . . 134
4.1 Parameter Settings for NCutH and NCutH$_0$ . . . . . 135
4.2 Clustering Performance . . . . . 136
4.3 Run Time Analysis . . . . . 138
5 Conclusion . . . . . 141
Appendix. Proofs . . . . . 142

**5 Divisive Clustering of High Dimensional Data Streams** **150**
1 Introduction . . . . . 150
2 Related Work . . . . . 153
3 Problem Description . . . . . 154
4 Methodology . . . . . 157
4.1 Learning High Variance Projections . . . . . 157
4.2 Splitting Based on a Projected Sample . . . . . 158
4.3 Handling Population Drift . . . . . 164
5 The HSDC Algorithm . . . . . 166
5.1 Computational Complexity . . . . . 167
6 Experimental Results . . . . . 168
6.1 Simulations . . . . . 170
6.2 Publicly Available Data Sets . . . . . 175
6.3 Discussion of Experimental Results . . . . . 178
7 Conclusion . . . . . 179
Appendix. Proofs . . . . . 180

**6 Conclusion** **184**
1 Summary of Contributions . . . . . 184
2 An Experimental Comparison of the Contributions . . . . . 187
2.1 Projected Divisive Clustering . . . . . 187
2.2 Large Margin Clustering . . . . . 188
2.3 A Summary of Clustering Performance . . . . . 189
2.4 Semi-supervised Classification . . . . . 192

3 Possible Extensions and Future Work . . . . . 195

**Bibliography** **197**
References . . . . . 198

# Thesis Details

| | |
|---|---|
| **Thesis Title:** | Projection Methods for Clustering and Semi-supervised Classification |
| **Ph.D. Student:** | David Paul Hofmeyr |
| **Supervisors:** | Dr. Nicos Pavlidis, Lancaster University |
| | Prof. Idris Eckley, Lancaster University |
| **Examiners:** | Dr. Ludger Evers, University of Glasgow |
| | Dr. Chris Sherlock, Lancaster University |

The main body of this thesis consist of the following papers.

- Chapter 2 has been submitted for publication as; Pavlidis, N. G., Hofmeyr, D. P and Tasoulis, S. K. "Minimum Density Hyperplanes," 2016.
- Chapter 3 A. has been submitted for publication as; Hofmeyr, D. P, Pavlidis, N. G. and Eckley, I. "Minimum Spectral Connectivity Projection Pursuit for Unsupervised Classification," 2016.
- Chapter 3 B. has been accepted for publication as; Hofmeyr, D. P. and Pavlidis, N. G. "Semi-supervised Spectral Connectivity Projection Pursuit", *PRASA-RobMech International Conference*, 2015.
- Chapter 4 has been submitted for publication as; Hofmeyr, D. P. "Clustering by Minimum Cut Hyperplanes", 2016.
- Chapter 5 has been accepted for publication as; Hofmeyr, D. P., Pavlidis, N. G. and Eckley, I. "Divisive Clustering of High Dimensional Data Streams," *Statistics and Computing*, issn: 0960-3174, doi: 10.1007/s11222-015-9597-y, Springer, 2015.

The majority of works involved collaboration with others, most notably my supervisors. In the case of Chapter 2, Nicos Pavlidis took an especially significant role.

# List of Figures


2.1 Binary partitions induced by 100 MDHs estimated through SQP and MDP$^2$ . . . 29
2.2 Impact of choice of $\alpha$ on minimum density hyperplane. . . . 42
2.3 Evolution of the minimum density hyperplane through consecutive iterations. . . . 43
2.4 Performance and Regret Distributions for all Methods Considered . . . 46
2.5 Classification error for different number of labelled examples for datasets with two clusters. . . . 47
2.6 Classification error for different numbers of labelled examples over all pairwise combinations of classes. . . . 48
2.7 Two dimensional illustration of Lemma 8 . . . 51

3.1 Effect of $T_\Delta$ on Distances and Similarities. . . . 72
3.2 Two dimensional projections of optical recognition of handwritten digits dataset arising from the minimisation of $\lambda_2(L(\boldsymbol{\theta}))$, for different values of $\beta$. In addition, the initialisation through PCA is also shown. The top row of plots shows the true clusters, while the bottom row shows resulting bi-partitions. . . . 73
3.3 Large Euclidean separation of the yeast cell cycle dataset. The left plots show the result from a 2 dimensional projection pursuit using the proposed method. The middle plots show the 1 dimensional projection pursuit result. The right plots show the result of the maximum margin clustering method of Zhang et al. (2009). . . . 79
3.4 Approximation Error Plots for S1 data set. . . . 85
3.5 Box plots of relative purity with additional red dots to indicate means. Methods are ordered with decreasing mean value. . . . 96
3.6 Box plots of relative $V$-measure with additional red dots to indicate means. Methods are ordered with decreasing mean value. . . . 97

3.7 Sensitivity analysis for varying $\sigma$. Standard Laplacian. The $x$-axis contains the multiplication factor applied to the default scaling parameter used in the experiments. . . . . 97
3.8 Sensitivity analysis for varying $\sigma$. Normalised Laplacian. The $x$-axis contains the multiplication factor applied to the default scaling parameter used in the experiments. . . . . 98
3.9 Sensitivity analysis for varying number of microclusters, $K$. Plots show median and interquartile ranges of performance measures from 30 datasets simulated from 50 dimensional Gaussian mixtures with 5 clusters and 1000 observations. . . . . 99
3.10 Sensitivity analysis for fixed number of microclusters, $K = 200$, and varying number of data. Plots show median and interquartile ranges of performance measures from datasets simulated from 50 dimensional Gaussian mixtures with 5 clusters and between 1000 and 10 000 observations. . . . . 100
3.11 Two projections, one admitting a separation of the classes (Left) and the other not (Right). . . . . 109

4.1 Optimal hyperplanes based on NCut (left) and RatioCut (right) from the same 100 initialisations . . . . . 126
4.2 Regret distributions of Purity (top) and V-Measure (bottom) across all 15 benchmark data sets. . . . . 140
4.3 Run time analysis from Gaussian mixtures. The plot shows the medians and interquartile ranges from 50 replications for each value of $n$. The number of clusters is fixed at 5. . . . . 140

5.1 Different changes in distribution and their impact on the clustering model. Red lines indicate necessity of model revision. Green lines indicate changes which can be addressed by extending the model without revision . . . . . 165
5.2 Performance on Static Data Stream with 20 Classes in 500 Dimensions . . . . . 171
5.3 Clustering Performance. Static Environment with Irrelevant Features. 20 Classes in 100 Relevant and 100 Irrelevant Dimensions with Moderate Variability . . . . . 172
5.4 Clustering Performance. Decreasing Classes. 500 Dimensions . 174
5.5 Clustering Performance. Increasing Classes. 500 Dimensions . 175
5.6 Clustering Performance. Distribution Overhaul. 20 Classes in 500 Dimensions . . . . . 176
5.7 Class Proportions of Forest Cover Type . . . . . 177
5.8 Clustering Performance. Forest Cover Type Data . . . . . 177
5.9 Clustering Performance. Gas Sensor Array Data . . . . . 178

6.1 Box plots of relative purity with additional red dots to indicate means. Methods are ordered with decreasing mean value. . . 192
6.2 Box plots of relative V-measure with additional red dots to indicate means. Methods are ordered with decreasing mean value. . . . . 193
6.3 Box plots of regret based on purity with additional red dots to indicate means. Methods are ordered with increasing mean value. . . . . 194
6.4 Box plots of regret based on V-measure with additional red dots to indicate means. Methods are ordered with increasing mean value. . . . . 195

# List of Tables


2.1 Details of Benchmark Data Sets . . . . . 39
2.2 Performance on the task of binary partitioning. (Ties in best performance were resolved by considering more decimal places) 45

3.1 Purity results for spectral clustering using the standard Laplacian, $L$. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold. . . . . 90
3.2 $V$-measure results for spectral clustering using the standard Laplacian, $L$. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold. . . . . 91
3.3 Purity results for spectral clustering using the normalised Laplacian, $L_{\text{norm}}$. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold. . . . . 92
3.4 $V$-measure results for spectral clustering using the normalised Laplacian, $L_{\text{norm}}$. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold. . . . . 92
3.5 Purity results for large margin clustering. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold. . . . . 93
3.6 $V$-measure results for large margin clustering. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold. . . . . 94
3.7 UCIMLR Classification Data Sets. Average Accuracy (%) over 10 Splits. . . . . 113
3.8 SSC Benchmark Data Sets. Average Accuracy (%) over 12 Splits. 113

4.1 Details of Benchmark Data Sets . . . . . 135
4.2 100 × Purity on Benchmark Data Sets. Highest Performance Highlighted in Bold. . . . . 138
4.3 100 × V-Measure on Benchmark Data Sets. Highest Performance Highlighted in Bold. . . . . 139
4.4 Run Time on Benchmark Data Sets (in Seconds) . . . . . 141

5.1 Clustering Performance. Static environments. . . . . 171
5.2 Clustering Performance. Static environments with irrelevant features. . . . . 173
5.3 Clustering Performance. Drifting environments. . . . . 176

6.1 A Comparison of the Performance of Proposed Methods for Projected Divisive Clustering. The Table Shows 100 × Purity on Benchmark Data Sets. Highest Performance in Each Case is Highlighted in Bold. . . . . 188
6.2 A Comparison of the Performance of Proposed Methods for Large Margin Clustering. The Table Shows 100 × V-Measure on Benchmark Data Sets. Highest Performance in Each Case is Highlighted in Bold. . . . . 189
6.3 A Comparison of the Performance of Proposed Methods for Large Margin Clustering. The Table Shows 100 × Purity on Benchmark Data Sets. Highest Performance in Each Case is Highlighted in Bold. . . . . 190
6.4 A Comparison of the Performance of Proposed Methods for Large Margin Clustering. The Table Shows 100 × V-Measure on Benchmark Data Sets. Highest Performance in Each Case is Highlighted in Bold. . . . . 191
6.5 A Comparison of the Proposed Methods for Semi-supervised Classification applied to UCI Machine Learning Repository Classification Data Sets. Average Accuracy (%) over 10 Splits. . . . . 193
6.6 A Comparison of the Proposed Methods for Semi-supervised Classification applied to Benchmark Data Sets taken from Chapelle et al. (2006b). Average Accuracy (%) over 12 Splits. . . . . 194

# Chapter 1

# Introduction

The problem of identifying groups of related objects is one of the fundamental tasks in knowledge discovery from data. This problem has been extensively studied in the literature on statistics, machine learning, data mining and pattern recognition because of the numerous applications in summarisation, learning and segmentation (Jain and Dubes, 1988; Aggarwal and Reddy, 2013). Such applications include,

- Engineering: In manufacturing, group technology seeks to identify similar items so that manufacturing and design concepts can be borrowed, thus speeding up the manufacturing lifecycle of emerging items (Pham and Afify, 2007). In radar signal analysis, the direction of arrival of pulses is crucial for object locating, however the sheer density of signals creates a computational challenge which is mitigated by identifying groups of pulses (Zhu et al., 2010). Outlier rejection deals with separating a single group of objects from remaining nuisance observations which do not fit within the group's context, this has applications in robotics in the form of consistent hypothesis identification (Olson et al., 2005).

- Computer Science: Web mining deals with organising the billions of web pages on the internet, so that queries can be handled efficiently. Grouping similar web objects, often by textual content, significantly aids this challenging task (Chen and Chau, 2004). Computer vision and image segmentation tasks require identifying different planar ranges in an image, which may be achieved by grouping small sections of an image, or sequence of images, to determine separate planes and using their respective orientation and geometry (Frigui and Krishnapuram, 1999).

- Life Sciences: In the study of spatial population genetics, tracing geo-

graphical ancestries can be aided by finding similar allele groups from spatially recorded genetic information (Rosenberg et al., 2002). Grouping lifestyle factors which have higher combined prevalence in individuals suffering certain diseases than would be predicted given their individual prevalence has use in preventative medicine (Schuit et al., 2002).

- Social Sciences: Person perception, in the field of psychology, looks at the different mental processes used to form impressions of people. Grouping people based on their perceptions of archetypal life figures has been useful in identifying psychopathologies (Rosenberg et al., 1996). The success of an education institution is predicated on both academic performance and enrollment management. Grouping students based on their persistence with academic courses can be useful in both areas, especially for early identification of potential defectors (Luan, 2002).

- Commerce: Market segmentation can be achieved by identifying groups of related consumers or products (Punj and Stewart, 1983), and is a central feature in targeted marketing strategy. Outlier rejection, as in the robotics application, can also be used to identify fraudulent behaviour of consumers or organisations, comparing behavioural patterns with the general behaviour of a group (Phua et al., 2010).

Broadly speaking the task of assigning a set of objects to groups may be termed *classification* (Jain and Dubes, 1988), and exists on three distinct levels in terms of the assumed available information. In the machine learning literature the amount of information available may be described by degrees of *supervision* for the learning task. On one end of the spectrum is the fully supervised classification task, in which the true groupings of all data used in the learning phase of the task are known. An inductive model is then built which can be used to predict the groupings of future data. It is the fully supervised task which is commonly given the name “classification". On the other end of the spectrum lies unsupervised classification. In this context there is no explicit information regarding how data should be grouped. The relationships between data must therefore be learned by other means. In this context any model which assigns groupings to data is used only within the context of the data used to build the model, and is not used to predict the groupings of future unseen data. Between these extremes lies semi-supervised classification. The motivation for semi-supervised classification can be seen as follows. When using a (supervised) classification model to predict the groupings of new data, there is an implicit assumption that the nature of those new data, in terms of their grouping tendencies, is somewhat related to the nature of the data used in building the model. If this assumption is true, then utilis-

ing whatever information about the new data is available, within the context of *all* the data, may be useful in determining the groups to which they belong. Semi-supervised classification is extremely useful in situations where identifying the true group *labels* for data is expensive. In such situations the number of labelled data may be small, and making inference or prediction from a (supervised) classification model can be unreliable, and the prediction task can be substantially aided by utilising every bit of available information.

This thesis focuses on the latter two, with a primary focus on the unsupervised problem. Semi-supervised classification is treated within the same framework, as an extension of the unsupervised case. Utilising information about data which does not explicitly determine their groups is therefore the central feature of the work presented. Throughout the remainder it is assumed that a data set is described by a fixed set of *features*, each taking a (real) numerical value. Data may therefore be thought of as occupying a vector space defined over the real numbers in which the number of dimensions is equal to the number of features describing the data.

A common assumption when data arise from multiple groups, or classes, is that the data tend to contain multiple *clusters*, and that objects within the same cluster are more likely to represent the same group. In the context of semi-supervised classification this is referred to as the *cluster assumption* (Chapelle and Zien, 2005), and in the unsupervised case leads to the field of cluster analysis. Cluster analysis has a very rich history, and remains one of the most active and important areas of research in fields relying on the analysis of data. The term cluster analysis, or simply *clustering*, is often used to refer to the task of unsupervised classification.

Accepting the cluster assumption immediately begs the question of what constitutes a cluster of data, and numerous definitions have been proposed, each leading to hosts of methods for identifying clusters which adhere to these definitions. Arguably the single common concept underpinning all of these, though, is that the spatial relationships between data contain useful information for determining clusters. The spatial relationships in this context are defined by the topological structure bestowed on the vector space containing the data, usually via the Euclidean (or $L_2$) norm, although other metrics may be used.

This thesis is motivated by two of the key challenges associated with data clustering, both from practical and theoretical points of view. The principal motivation comes from the problem of dimension reduction. Dimension reduction forms a crucial part of the analysis of data which are either high dimensional, i.e., data containing a very large number of features, or data which contain features, or combinations of features, that may be irrelevant or even counter-informative for identifying clusters. A number of contri-

butions are presented which address this problem in a principled manner; using projection-pursuit formulations to identify subspaces which contain useful information for the clustering task. A secondary motivation arises from the challenge associated with clustering data incrementally, where data are received sequentially in a data stream. The final contribution of this thesis addresses both challenges, and proposes a method for clustering high dimensional data streams within a principled statistical framework. Further details of these contributions are given at the end of this chapter.

The remainder of this chapter is organised as follows. In Section 1 a number of cluster definitions are explored, as well as some important methods for finding clusters which fit within their respective definitions. In Section 2 a discussion of semi-supervised classification methods will be presented, with particular attention to those adhering to the cluster assumption. Following that, in Section 3 some fundamental challenges associated with clustering methods from practical, theoretical and philosophical perspectives will be presented. The focus of the remaining thesis will then be outlined in Section 4.

Neither of Sections 1 and 2 is intended to be a comprehensive account of the entire literature on these problems, but rather provides a representative cross section of concepts and methods which are either highly influential or of relevance in the remaining thesis. Each subsequent chapter will contain its own review, documenting existing approaches which are of particular importance for the context of that chapter.

# 1 Clustering

This section is dedicated to the discussion of existing methods for clustering. Important concepts and methods in the clustering literature are discussed under the headings relating to different definitions of what constitutes a cluster. Clustering methods can also be split between two distinct approaches to model structure. Hierarchical clustering models are nested sequences of partitions (Jain and Dubes, 1988) and may be further categorised into divisive and agglomerative clustering. In divisive clustering, beginning with the entire data set being a cluster, clusters are recursively partitioned until a stopping criterion is met. Conversely, agglomerative clustering begins with every datum in its own cluster, and repeatedly merges clusters until all data are contained in a single cluster. Partitional clustering models instead directly assign data to their final clusters in a single step. Partitional methods may also be referred to as generating a *flat clustering*. The focus of this section will be more strongly motivated by the concept of what constitutes a cluster, than by the model structure in which the clusters reside. The reason being that

the philosophy behind a particular approach to establishing or identifying related groups of objects is much more closely related to the group definitions than their representation.

## 1.1 Centroid-Based Clustering

Centroid-based clustering defines clusters in relation to single representative points (centroids). Data are thought to collect around these points, forming clusters. Models derived from these clustering methods may be summarised by a set of centroids, and data are classified based on which centroid is nearest to them (Leisch, 2006). This approach generates a partitional clustering model.

Formally, the clustering task may be stated in relation to the following optimisation problem,

$$\min_{C \in \mathcal{F}^k} \sum_{i=1}^{n} \min_{c \in C} f(d(x_i, c)). \tag{1.1}$$

Here the $x_i, i = 1, ..., n$ represent the data points and the set $C$ is the set of centroids over which the optimisation takes place. The set $\mathcal{F}$ represents the collection of feasible centroid values. The function $f : \mathbb{R}^+ \to \mathbb{R}^+$ is non-decreasing, and operates on the distances between the data and their associated centroids, via the distance metric $d(\cdot, \cdot)$. In these approaches the number of centroids, and therefore clusters, $k$, is chosen by the user.

The ***k*-means** clustering method is seen as one of the simplest and most classical approaches to data clustering (Jain, 2010) and remains one of the most widely used in practice, largely due to its simplicity (Aggarwal and Reddy, 2013). In the $k$-means approach the optimal clustering model is defined as the set of $k$ centroids which minimises the sum of the squared Euclidean distances between each datum and its cluster centroid. In terms of the above optimisation, (1.1), one therefore has $f(x) = x^2$ and $d(x, y) = \|x - y\|_2$. The centroids are essentially unconstrained, and so the set $\mathcal{F}$ is given by the whole space from which the data set arose. It is straightforward to show that with this objective, the centroids for the optimal model are defined as the means of the data assigned to each cluster. A simple iterative algorithm was proposed by Lloyd (1957, 1982). The algorithm is initialised with a set of $k$ potential centroids. It then alternates between assigning the data to their nearest centroid, and then updating the centroids by giving them the value equal to the mean of the data assigned to them. This is repeated until the solution converges.

Solving the objective in (1.1), where instead the $L_1$ norm is used and simple distances, rather than squared distances as in $k$-means, determine the function $f$ leads to the ***k*-medians** clustering problem (Bradley et al., 1997).

The structure of the algorithm for solving this problem is essentially the same as for the $k$-means objective. Both $k$-means and $k$-medians have worst case computational complexity which is non-polynomial in general, even for finding local optima, however most implementations have an empirical run time which is of $\mathcal{O}(nkD)$, where $D$ the number of dimensions.

In ***k*-medoids** clustering, the centroids are selected from the data set itself. Therefore the set $\mathcal{F}$ is given by $\{x_1, ..., x_n\}$. This is especially useful when the objects being clustered may not permit a reasonable interpretation of mean or median (Aggarwal and Reddy, 2013). While the number of iterations is often more than is required for solving either $k$-means or $k$-medians (Aggarwal and Reddy, 2013), this approach does have the benefit that distance calculations can be recycled since many of the pairwise inter-distance calculations will have been performed in previous iterations.

The above methods represent three of the fundamental centroid-based clustering methods, largely due to the fact that the centroids admit closed form solutions, making them especially attractive from a computational point of view. The problem formulation, however, allows for any general distance function to be used, and modern optimisation techniques have allowed for these to be implemented practically (Leisch, 2006).

Centroid-based clustering methods benefit from their ease of implementation, and fast computation in most practical examples. They also have close connections with model-based clustering, for example the $k$-means solution can be seen as an approximation of the Gaussian Mixture Model solution for a fixed number of clusters, and isotropic covariance matrices. The fundamental limitations of these approaches are the low flexibility of cluster shape, since cluster boundaries are given by the Voronoi tesselation of the centroids, and the fact that the number of clusters must be prespecified.

## 1.2 Connectivity-Based Clustering

Unlike in centroid-based clustering, the pairwise distances between data points are the driving force in connectivity-based clustering. The term *connectivity* refers to the algorithmic approach of merging data, or clusters of data already determined, until all data are connected as a single cluster. This generates an agglomerative model, and different levels in the hierarchy provide the clustering result at different granularity/scale.

Within **single-linkage** clustering, at each step in the agglomerative procedure precisely two clusters (which may be singletons) are merged. The pair selected for merging is that which has the minimum distance between the clusters, based on the standard metric extension to sets. That is, the smallest pairwise distance *between* them. Formally, if $C_1, ..., C_k$ represent the clusters at iteration $n - k + 1$, then at iteration $n - k + 2$ clusters $C_i$ and $C_j$ are replaced

with $C_i \cup C_j$, where $i, j$ minimise

$$\min_{\{l,m\}\subset\{1,\ldots,k\}} \min_{x\in C_l, y\in C_m} d(x,y). \tag{1.2}$$

This approach is called single linkage as only a single pair of data belonging to the two clusters being merged need to be close together, i.e., a single link of short distance must exist. Sibson (1973) developed a quadratic time, linear storage algorithm for generating this hierarchical model. Both of these complexities have been shown to be optimal for this problem.

In **complete-linkage** clustering on the other hand, the pair of clusters merged is the pair between which the largest distance is minimal. In other words, (1.2) is replaced with,

$$\min_{\{l,m\}\subset\{1,\ldots,k\}} \max_{x\in C_l, y\in C_m} d(x,y). \tag{1.3}$$

A similar algorithm to that of Sibson (1973) for single linkage clustering was proposed by Defays (1977), which again has quadratic time complexity in the number of data. This has been shown to be the optimal complexity for the complete linkage problem as well.

Other analogous models have been proposed, wherein the only difference is the rule for selecting the next pair of clusters to be merged. An example of this is the **average linkage** approach, or **U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic mean (**UPGMA**), proposed by Sokal and Michener (1958). A quadratic time algorithm for the method was later developed by Murtagh and Raftery (1984).

Alternative to the strategy of merging a single pair of clusters repeatedly is an approach in which all pairs of clusters satisfying a connectedness criterion are merged. Weakening the connectedness criterion, for example by increasing the minimum distance required to satisfy connectedness, leads to more and more clusters being merged. If all pairwise distances are different, then it is clear that this approach can be made equivalent to the single-linkage approach above. Within this formulation, at each iteration the clusters can be interpreted as the connected components of a graph defined over the clusters present at the previous iteration. An edge is present in the graph if and only if the two corresponding clusters are merged at the current iteration. Graph based methods will be discussed in greater detail below, where more general graphs, i.e., with edges weights assuming continuous values, will be permitted.

An advantage of connectivity based methods is that they admit clusters of arbitrary shape, and utilise information in the data set at a local level. In practice, though, the single linkage approach has been criticised for its tendency to emphasise elongated clusters caused by the chaining effect inherent in its formulation. The computational complexity of these methods limits them to data sets of only moderate size.

## 1.3 Graph Partitioning Based Clustering

Graph partitioning approaches for clustering draw on the wealth of existing graph theoretic methods to produce clustering models. In order to do so, a graph must be defined which is relevant to the clustering task. Certain data structures, such as networks, immediately lend themselves to graph formulations, however it is possible to define a graph which has useful properties for clustering an arbitrary set of objects provided a quantitative measure of similarity between pairs of objects is available. Consider a complete graph in which each object is designated a vertex, and edge weights take values equal to the similarity between their adjacent vertices. Then subgraphs containing comparatively high edge weights correspond to collections of objects which are mostly similar, and can therefore be interpreted as clusters. Alternatively, assigning edge weights equal to the distance between the adjacent vertices means that subgraphs with relatively low edge weights may correspond to clusters of data which are close together. Some notions of optimality in this context have been introduced, and will be discussed below.

Using graph cuts is an intuitive way of obtaining clusters of similar objects. A graph cut is given by the sum of the edge weights connecting different components of a partition of the graph. If the edges represent the similarities between data, then minimising the cut will form a clustering of a data set in which the data in different clusters have low similarity. Graph cuts can be usefully formulated using an *affinity matrix*, $A \in \mathbb{R}^{n\times n} : A_{i,j} = \text{similarity}(x_i, x_j)$. For a partition of the data set into clusters $C_1, ..., C_k$, the graph cut may then be given by

$$\text{Cut}(C_1, ..., C_k) = \frac{1}{2}\sum_{l=1}^{k}\sum_{x_i\in C_l, x_j\notin C_l} A_{ij}. \tag{1.4}$$

Minimising this graph cut objective has been found to often result in very small clusters (von Luxburg, 2007). This is because the number of edges being "broken" by the cut is equal to $(|C|(N-|C|))$, which is minimised if either $|C| = 1$ or $|C| = N-1$. Forcing clusters to be above a specific size, or normalising the graph cut objective to emphasise more balanced partitions makes the problem NP-hard (Wagner and Wagner, 1993). It can be shown that two popular normalisations of the graph cut objective, namely RatioCut and normalised cut (NCut), can be formulated under the following problem structure,

$$\min_{C_1,...,C_k} \text{trace}(H^\top LH) \tag{1.5a}$$

$$\text{s.t. } H^\top KH = I \tag{1.5b}$$

$$H_{ij} = \begin{cases} 1/\sqrt{\text{size}(C_j)}, & x_i \in C_j \\ 0, & \text{otherwise.} \end{cases} \tag{1.5c}$$

The matrix $L := D - A \in \mathbb{R}^{n \times n}$ is called the *graph Laplacian*, where $D$ is the diagonal matrix with $ii$-th entry $\sum_{j=1}^{n} A_{ij}$, and the matrix $H \in \mathbb{R}^{n \times k}$ encodes the cluster memberships for each of $k$ clusters. For RatioCut, the size of a cluster is measured by its cardinality, and the matrix $K$ is simply the identity. For NCut, size is measured by the *volume* of a cluster, given by $\sum_{x_i \in C} D_{ii}$, and $K = D$. Spectral methods can be used to find approximate solutions to the normalised cut problem (Hagen and Kahng, 1992; Shi and Malik, 2000), in which the constraint on the matrix $H$ given by (1.5c) is relaxed. The solution in this case is given if the columns of $H$ are replaced with the $k$ smallest eigenvectors of $L$, or of $D^{-1/2}LD^{-1/2}$ in the case of NCut. However, in this case the clusters are not fully determined, and a second clustering step is performed on the rows of $H$ to determine the final clustering of the data. This leads to the popular **spectral clustering** algorithms (von Luxburg, 2007). A more detailed account of spectral clustering and normalised cuts is given in Chapters 3 and 4. The second step in which the final clusters are determined can provide an alternative interpretation of spectral clustering, in which the matrix $H$ represents a partial embedding of the data within a kernel space. In particular, if the clustering step is performed using $k$-means, then spectral clustering can be shown to be a special case of so-called **kernel *k*-means** (Dhillon et al., 2004).

An alternative approach to graph cuts uses minimum weight spanning trees (MSTs). When edges correspond to the distance between the adjacent vertices, the MST gives the fully connected graph which contains the shortest total distance between the data. The edges in the MST are likely therefore to contain the important connections between data within clusters. A theorem from Zahn (1971) shows that if a bi-partition of a data set attains the largest possible distance between the two elements of the partition, based on the minimum pairwise distance between data, then the restriction of the MST to each element in the partition remains connected, i.e., is a subtree. That is, if $C$ is the solution to

$$\max_{B \subset \{x_1, \ldots, x_n\}} \min_{x \in B, y \notin B} d(x, y), \tag{1.6}$$

then the subset of edges in the MST which connect elements of $C$ is a connected subgraph. A direct consequence of this result is that by removing the largest weighted edge from the MST, the remaining subgraphs define the two-way clustering of the data set which maximises the distance between them. To generate a full clustering model, what remains is a method for removing the edges from the MST which are likely to exist between clusters, rather than within them. In the **geometric minimum spanning tree** clustering method (Brandes et al., 2003) a performance measure is computed for each clustering obtained by removing from the minimum spanning tree the edges with weight above a threshold. Since the tree has only $n - 1$ edges, where $n$ is the size of the data set, only $n - 1$ such thresholds need to be

considered. The **fuzzy C-means minimum spanning tree** method (Foggia et al., 2007) instead clusters the edges based on their weights using the fuzzy C-means algorithm, and retains only those in the cluster of smaller edge weights.

## 1.4 Density-Based Clustering

In density-based clustering, clusters are defined as regions of high data density which are separated from other high density regions by a region of low data density. The density at a point may be related to the number of data falling within a specified neighbourhood size or by using a smoother kernel-based estimate (Aggarwal and Reddy, 2013).

The **DBSCAN** clustering algorithm (Ester et al., 1996) uses the former of the above definitions. In this case *high density points* are those points within a specific distance of at least a chosen minimum number of other data. Each high density point is then connected to all points lying within the specified distance, and sets of connected points are defined as clusters. Any point which is not within the specified distance of at least one high density point is interpreted as an outlier, not belonging to any cluster.

DBSCAN can be seen as approximating the *level sets* of a kernel density estimate of the data distribution, where the uniform kernel is used. In the non-parametric statistics literature, this method is often applied to a more general density estimate (Azzalini and Torelli, 2007; Stuetzle and Nugent, 2010). In this case clusters are defined as maximally connected components of the level set of a probability density function (Hartigan, 1975). The level set of a function is the subset of the function's domain upon which the functional value lies above a chosen threshold level. Formally, the level set of a function $f : \mathcal{X} \to \mathbb{R}$ at level $\lambda$ is defined as,

$$\{x \in \mathcal{X} | f(x) \geq \lambda\}. \tag{1.7}$$

Computing the level sets of an unknown probability density directly is extremely challenging even in moderate dimensions (Stuetzle and Nugent, 2010). Certain approaches approximate these level sets as the union of spheres around those data at which the estimate of the density is above the threshold level (Cuevas and Fraiman, 1997; Rinaldo and Wasserman, 2010). This method has compelling consistency properties (Rinaldo and Wasserman, 2010), in that these approximations form disjoint neighbourhoods of the true components of the level sets of the underlying density with high probability. However, in the clustering context it is only the groups of data occupying these components of the level sets which are of interest. Other methods therefore attempt to connect data (i.e., assign them to the same cluster) by establishing if there is a path between them lying completely within the level set of the density (Azzalini and Torelli, 2007; Stuetzle and Nugent, 2010).

Different interpretations of where the density is "high", i.e., using different threshold levels for the level set, gives rise to different clusterings, and using this approach for a range of thresholds results in a hierarchical clustering model, known as the *cluster tree*. A more in depth account of density-based clustering, especially from the non-parametric statistical perspective is provided in Chapters 2 and 5.

An alternative approach to density-based clustering is via a grid formulation. In this case it is the cells of a grid defined over the space occupied by the data which undergoes clustering. In this case, the concept of adjacency has a far more interpretable meaning, and so establishing connections between grid cells is less challenging than for data points. Here the grid cells containing sufficiently many points are seen as high density cells, and adjacent high density cells are joined to produce clusters. The final clustering of the data connects data belonging to the same clusters of grid cells. An advantage of this approach is that they are at least theoretically applicable in high dimensional applications, because the lower dimensional grids define clusters on subsets of the dimensions. Such a hierarchical grid structure, where the hierarchy is defined over the dimensions, can be seen in the **STING** clustering method (Wang et al., 1997).

A major advantage of density-based clustering is that the number of clusters can be estimated automatically, and moreover they provide a natural framework for handling outliers. In addition, they are well founded from a statistical perspective in that a feature of the underlying probability distribution is being estimated directly, and are capable of representing the full underlying distribution. From a computational point of view, in the general case density methods can be complex. These methods are also highly limited in their applicability to higher dimensions, since the sparsity of data makes estimating the underlying density unreliable, and in grid-based approaches the size of the grid grows exponentially with the number of dimensions.

## 1.5 Model-Based Clustering

Model-based clustering again assumes an underlying probability distribution has generated the data. Unlike the non-parametric approach in the previous subsection, however, the data are assumed to be a sample from a finite mixture distribution in which each component has a known parametric form. Formally, the data set is assumed to be a sample of realisations of a random variable $X$ with density function $f$, where $f$ may be expressed as,

$$f(x) = \sum_{i=1}^{k} \pi_i f_i(x|\theta_i). \tag{1.8}$$

Here the $\pi_i's$ are the *mixing proportions*, i.e., $\sum_{i=1}^{k} \pi_i = 1, \pi_i > 0$, and each $f_i$ is a density function parameterised by $\theta_i$. If all parameters of the mixture distri-

bution can be estimated, then data are classified according to which component in the mixture has the greatest posterior probability of having generated it (Fraley and Raftery, 2002). In other words each datum is classified by,

$$\text{Class}(x_j) = \text{argmax}_{i \in \{1,\ldots,k\}} \pi_i f_i(x_j|\theta_i). \tag{1.9}$$

In general all components are assumed to belong to the same family of distributions, i.e., $f_i = f_j \; \forall i, j$, with the Gaussian mixture model being the most common.

Both partitional and agglomerative clustering approaches have been applied to this setting. In the partitional case (McLachlan and Basford, 1988; Celeux and Govaert, 1995), parameter estimation can be done using Expectation Maximisation for mixture models. The agglomerative method uses the same algorithmic structure as connectivity based cluster, where in this case the criterion for merging two clusters is based on classification likelihood (Murtagh and Raftery, 1984; Banfield and Raftery, 1993).

Once a family of distributions has been chosen for the individual components in the mixture, methods can be further divided by how much freedom is allowed in the estimation of parameters. If parameter values are unconstrained, then the number of parameters that need to be estimated can be large, leading to computational issues. Moreover, the reliability in estimation can be negatively affected if few data are used for each parameter, and in the extreme case this approach can lead to a lack of identifiability. In the Gaussian mixture case, constraints on the covariance matrices of the different components can be introduced; either forcing all covariances to be equal to the same scaled identity matrix, or allowing for an arbitrary covariance but ensuring it is shared by all components (Friedman and Rubin, 1967). A more general framework was given by Banfield and Raftery (1993) where covariances matrices are described by their eigen-decomposition and certain elements in this decomposition are fixed for all components, while others are allowed to vary.

Probably the most attractive feature of model based clustering is that the problem of determining the number of clusters is stated within the thorough statistical framework of model selection, where measures such as the Bayesian Information Criterion (Schwartz, 1978) can be used. The fundamental limitation is that clusters should be describable by a known parametric distribution, and also that all clusters should fall into the same family of distributions.

## 2 Semi-supervised Classification

In semi-supervised classification the true groupings, or class identities, of some of the data are known (*labeled* data) and the task is to assign the data

whose classes are unknown (*unlabeled* data) to one of the classes defined within the labeled data. The motivation behind many approaches to solving this problem is the assumption that data which lie within the same cluster are likely to belong to the same class. One of the earliest approaches to semi-supervised classification was based on this cluster assumption (Chapelle et al., 2006b). In this section, a selection of cluster motivated semi-supervised classification methods will be discussed.

## 2.1 Low Density Separation Methods

If clusters are seen as regions of high data density which are separated by relatively sparse regions, then an assumption equivalent to the cluster assumption is the so-called *low density separation* assumption: The boundary separating classes should lie in a low density region (Chapelle et al., 2006b). Many modern approaches to the semi-supervised classification problem focus more on the low density separation assumption, and try to identify low density regions which separate the known classes.

A common approach to determining the low density boundaries is through Support Vector Machines (Vapnik and Sterin, 1977). SVMs were originally designed for supervised classification, and are used to find the linear separator (hyperplane) which separates the classes and which attains the largest *margin* on the data, in other words the linear separator which is as far as possible away from its nearest datum. Kernel methods can be used to find non-linear separators by implicitly embedding the data in a higher dimensional space. In semi-supervised classification, the **T**ransductive **SVM** (**TSVM**) is the hyperplane which separates the classes and attains the largest margin on both the labeled *and* unlabeled data (Joachims, 2006). The corresponding optimisation problem is posed in the context of the standard SVM, except that the class labels for the unlabeled data are treated as decision variables. Since the set of labels is discrete, this corresponds to a mixed integer quadratic programme, for which no efficient algorithms exist (Joachims, 2006).

Various authors have attempted to solve the problem exactly (Vapnik and Sterin, 1977; Bennett and Demiriz, 1998), but this approach is limited to cases with at most hundreds of unlabeled data points. The $\mathbf{SVM}^{light}$ approach proposed by Joachims (1999) does not find the global optimum, but can handle problems with up to 100000 data. The method uses a local descent approach which iteratively swaps two labels assigned to unlabeled data. A similar approach was proposed by Demiriz and Bennet (2000), which differs from $\text{SVM}^{light}$ in the number of labels and which selection of labels are swapped, as well as the heuristics used to avoid local optima (Joachims, 2006). Finally De Bie and Cristianini (2004) used a convex relaxation via semi-definite programming. Chapelle et al. (2006a) used a continuation approach to avoid local minima in the TSVM objective. In this approach a sequence of optimisa-

tion problems is solved, where each is initialised with the solution to the previous probelm. In each the objective is convolved with a Gaussian smoothing kernel for a shrinking sequence of bandwidths. As the bandwidth decreases the objective function for the convolved problem becomes closer to the true underlying objective, and this procedure has the potential to find very good solutions.

The **L**ow **D**ensity **S**eparation (**LDS**) approach of Chapelle and Zien (2005) attempts to establish if unlabeled points can be connected to a labeled point by a path of high density. A robust estimate of the lowest density point along the best path between pairs of data is used to establish a *density sensitive distance measure*, which is then used to define a kernel used in the training of a TSVM.

## 2.2 Graph Partition Based Methods

The graph formulation described in relation to clustering provides a convenient framework for incorporating additional information, such as known class labels as in semi-supervised classification. Since the graph may be defined via edges taking values equal to the pairwise similarities between data, edges joining data known to belong to the same class can be assigned maximum similarity, while edges between data known to belong to different classes may be assigned the minimum similarity value. Using a normalised cut technique would therefore tend to separate the classes in the labeled data as well as connect unlabeled data which are similar to the known classes under the resulting clustering. This sort of approach has been implemented (Chen and Feng, 2012) in the related field of *semi-supervised clustering*; a problem in which no class labels are known but pairs of data may be known to belong to the same class or not, thereby introducing constraints in the clustering formulation.

In the semi-supervised classification context, a number of approaches have been proposed. **Label Propagation** uses an iterative procedure to *propagate* labels through a similarity graph (Bengio et al., 2006). A labelling vector is updated by repeatedly multiplying by the normalised similarity matrix until convergence. The early algorithm by Zhu and Ghahramani (2002) has been extended to allow for more general clusters and to improve the stability of the algorithm (Bengio et al., 2006), or by smoothing the actual labelling vector in an approach referred to as *label spreading* (Zhou et al., 2004).

In addition to these methods, some authors have combined the kernel based classification objective with the spectral clustering objective to obtain so-called **Laplacian Regularisation** or **Laplacian SVM** (Sindhwani et al., 2006).

# 3 Challenges in Data Clustering

This section investigates two challenges associated with data clustering which have received considerable attention in the literature. Subection 3.1 looks at the problem of partitioning data with a very large number of features, so-called *high dimensional* data. This is a fundamental challenge faced in a variety of problems in data analysis, and extends beyond the obvious computational difficulty associated with processing data objects of a much larger size. Subsection 3.2 covers the *data stream* paradigm. In a data stream environment, the objects being partitioned arrive in sequence and cannot be stored in memory. The model used for assigning data to groups must therefore be built incrementally.

## 3.1 High Dimensionality

The fundamental challenge, from a theoretical point of view, associated with analysing high dimensional data is that data points become increasingly sparse as dimensionality increases (Steinbach et al., 2004). This concept is most easily intuited using a grid-based representation. For a fixed number of grid cells partitioning each dimension, the number of total grid cells in the full dimensional space grows exponentially with the dimension. Unless the number of available data grows with at least the same rate, then for a very large number of dimensions the ratio of non-empty cells to empty cells approaches zero. The space is, in a sense, “almost everywhere" sparse.

From a practical point of view, one of the major challenges for data partitioning is that certain distance measures lose meaning in very high dimensions (Kriegel et al., 2009). This is related to the fact that pairwise distances between points tend to be more uniform in high dimensions (Beyer et al., 1999; Aggarwal et al., 2001). This is expressed theoretically by Beyer et al. (1999), who show that for certain distributions underlying the data, the difference between the largest and smallest distance in a data set, divided by the smallest distance, tends to zero in probability as dimension approaches infinity. There is *poor discrimination* between the nearest and furthest neighbour (Aggarwal et al., 2001).

The standard approach to handling high-dimensional data is via dimension reduction. Dimension reduction can be performed as a preprocessing task before any attempt to partition a data set is undertaken, or it can be performed in conjunction with the partitioning step. Dimension reduction techniques can also help significantly even in relatively low dimensions, by removing the effect of features which are irrelevant for determining clusters, or identifying pairs of features which are highly correlated with one another.

*Subspace clustering* usually refers to the case where it is assumed that only a subset of the features contain information which is relevant for defining

clusters (Steinbach et al., 2004). A challenge in this context is that different subsets may be relevant for different clusters, and so attempting to cluster directly using only a single subset of features may not lead to meaningful results. Grid-based clustering, rather counterintuitively, offers a useful means for subspace clustering. It is counterintuitive since the number of grid cells to process is so large that it seems grid-based methods would be particularly limited in high dimensional applications. Their use is well described by Agrawal et al. (1998) in relation to the **CLIQUE** algorithm. The observation is that a density-based cluster defined in a subset of dimensions, when *projected* onto each of those dimensions will exhibit a (one-dimensional) high density region. Importantly, the intersection of two or more one-dimensional high density regions does not necessarily correspond to a dense grid cell in those dimensions. Low dimensional dense grid cells, when intersected, therefore represent the potential locations of higher dimensional clusters. Only those intersections need to be considered when searching for clusters, rather than trying to find dense regions over an exponentially large number of grid cells.

Other cluster definitions, such as centroid-based, have also been considered for subspace clustering. For example, the **PROCLUS** algorithm (Aggarwal et al., 1999) used a $k$-median based approach in which each cluster has an associated set of dimensions within which the associated data are most compact, or have least variability. Distance calculations for each cluster are only computed within their relevant subspace, and using the $L_1$ norm.

Subspace clustering is somewhat limited by restricting attention to clusters defined in axis-parallel subspaces. The term *projected clustering* will be used to refer to clustering techniques which attempt to find clusters in arbitrarily oriented subspaces. It is important to note that other authors have used “projected clustering" to refer to the subspace clustering above, and may refer to clustering within arbitrary subspaces as “correlation clustering".

The most common approach to projected clustering uses Principal Component Analysis (PCA), either locally (on subsets of the data set) or globally, to determine subspaces within which data have high and low variability (Kriegel et al., 2009). **ORCLUS** (Aggarwal and Yu, 2000) is an extension of PROCLUS based on low order (low variability) PCA projections. Variations on this approach all use PCA on a local level.

The **P**rincipal **D**irection **D**ivisive **P**artitioning algorithm (**PDDP**) (Boley, 1998) uses PCA iteratively within a divisive hierarchical procedure. First the entire data set is projected onto the first principal component (the univariate subspace in which the variability is maximised). The data are then split in two at their mean within this subspace. This process is then repeated recursively on the resulting subsets, selecting the next subset to be partitioned based on a heuristic measure of cohesion, called *scatter value*. When the number of subsets reaches a chosen number the process terminates. This algorithm is

not motivated by a particular cluster definition, but rather uses the reasoning that subspaces in which the data have high variability are likely to display high *between cluster* variability, in which case partitioning at the projected mean is likely to separate clusters, rather than cut through them.

Two extensions to the PDDP algorithm were considered by Tasoulis et al. (2010). Both are motivated by density-based clustering, and the equivalent low density separation assumption. The **d**ensity **e**nhanced PDDP (**dePDDP**) algorithm projects a data set onto its first principal component, as in PDDP, and then uses a kernel density estimate of the projected data to find a low density separator. It then splits the subset which induces the lowest density separation based on its respective density estimate. The **i**nterval PDDP (**iPDDP**) method is similar, but rather than using a kernel estimate of the density it splits a data set at the largest gap between consecutive projected points, thereby separating by the largest margin hyperplane orthogonal to the first principal component.

The generality offered by projected clustering over subspace clustering is clearly beneficial in many cases. PCA projections have been successfully applied in many areas, however it is trivial to construct examples where PCA is inappropriate. *Projection pursuit* refers to a class of optimisation problems aimed at finding the most “interesting" subspace within a multivariate data set (Jones and Sibson, 1987). The interestingness of a data set within a given subspace is referred to as the *projection index*. The term projection pursuit is attributed to Friedman and Tukey (1974), however an associated practice dates back to Kruskal (1969). By defining a projection index which is relevant to the ultimate task at hand, e.g. clustering, it is possible to overcome some of the shortcomings associated with using *off-the-shelf* dimension reduction techniques like PCA. While these off-the-shelf methods have been extremely useful in the modern era of data analysis, the subsequent task, while eased by the reduced size of the data, often remains a challenging problem. By performing dimension reduction in tandem with the corresponding analysis, the ultimate task can be made much easier. This may be of particular relevance in relation to clustering. In a theoretical study of the concept of *clusterability*, Ackerman and Ben David (2009) observed that “Although most of the common clustering tasks are NP-hard, finding a close-to-optimal clustering for well clusterable data sets is easy (computationally)".

## 3.2 Data Streams

A data stream may be characterised by the sequential arrival of data. Such situations arise when there is a relative overabundance of data in terms of available computing and storage resources. This can occur either where data sets are so large that they cannot be stored in memory, or where they arrive with such high velocity that standard approaches would be unable to keep

pace. Random access to past observations is costly, and so a single linear scan of the data is the only acceptable method for processing the data (Guha et al., 2003; Silva et al., 2013). A defining property is that the generative process which is assumed to underlie the arriving data may be subject to change, a phenomenon known as *population drift* (Babcock et al., 2002). An algorithm for determining clusters within a data stream must therefore be able to build a model incrementally, and the computational requirements for updating the model must be bounded, and so cannot increase as more data are observed. In addition, it should be able to identify when new clusters emerge, or clusters disappear, or when clusters undergo some sort of structural change (Jain, 2010).

The majority of data stream clustering algorithms are variations on standard approaches. They are comprised of an *online* phase which updates a set of data structures as new data arrive, and an *offline* phase which processes the data structures (Aggarwal et al., 2003; Cao et al., 2006). The data structures usually represent summaries of subsets of the data processed so far, and the offline phase is analogous to an existing clustering algorithm, but which operates on these summaries rather than on individual data points. Population drift is usually accommodated in a heuristic manner by discounting the emphasis of older data on the data structures (Aggarwal et al., 2003, 2004; Cao et al., 2006).

The practical challenges associated with clustering data streams are clear, however the data stream paradigm also poses philosophical questions about what defines a cluster. Consider a situation in which the generative process undergoes an abrupt change, such that the data distribution after the change is essentially unrecognisable in the context before the change. Even if the number of clusters under the new distribution is the same, how can one associate data which arrived before the change with those arriving afterwards? If the location of such abrupt changes is known, it could be argued that the previous clusters no longer exist, and any information drawn from the new clusters should be independent of what came before. However, abrupt changes might not affect all clusters, and discarding past information could be detrimental if unnecessary. If the location of changes is unknown, then the problem becomes immeasurably more complex. Attempting to describe clusters as persistent entities is almost paradoxical if clusters are described as groups of data. It seems necessary, therefore, to attempt to estimate features of the underlying generative process and define clusters relative to them. In addition, it is preferable to identify changes in the generative process which affect the cluster definitions, rather than discounting information from previous data which may or may not be related to the data relevant to the current cluster definitions. Ideally, in addition the identification of changes to the process should be at a local level so that information from persistent features of the process is not discarded.

# 4 Focus of Thesis

Within the previous sections, a selection of existing approaches to clustering and semi-supervised classification were discussed, highlighting some advantages and limitations. The remainder of this thesis presents a number of new methods for partitioning data sets. It is widely accepted that no single approach will be usefully applicable to every data set (Jain and Dubes, 1988; Aggarwal and Reddy, 2013), indeed it is unlikely that a single approach can reasonably overcome all limitations associated with existing methods.

These contributions are aimed at general methodology and are intended for wide applicability; they are therefore not proposed with any particular application or application area in mind. As such, they do not attempt to address every potential limitation and challenge associated with clustering and semi-supervised classification, nor solve the respective problem completely within a single applied context. Rather they represent pragmatic and principled approaches to the problem of data partitioning that are motivated by fundamental cluster definitions, under the unifying framework of projected divisive partitioning.

## 4.1 Contributions

The body of this thesis consists of four chapters. In Chapter 2 a new hyperplane-based classification method is proposed for unsupervised and semi-supervised classification problems. The formulation is motivated by the low density separation assumption, and the optimal hyperplane is defined as that which has the minimum integrated density along it. This density is estimated using kernel density estimation. The *minimum density hyperplane* attains the least possible upper bound on the full dimensional density evaluated along it, and so is highly unlikely to cut through high density clusters. Like the maximum margin hyperplane methods, a naive approach to solving this problem is faced with many local minima. To mitigate this problem, a projection pursuit formulation is developed in which the projection index is given by the minimum integrated density of *all* hyperplanes orthogonal to the corresponding subspace, penalised to emphasise a useful partition of the data. The projection pursuit is therefore directly connected with the clustering objective. This is perhaps the first direct attempt to implement the low density separation assumption in a finite sample setting. A theoretical investigation reveals that the minimum density hyperplane converges to the maximum margin hyperplane as the bandwidth in the kernel density estimator is reduced to zero.

Chapter 3 contains two parts. Part A. contains a thorough exposition of projection pursuit based on spectral connectivity for unsupervised partitioning. The spectral connectivity of a data set relates to the optimal solution of the relaxed normalised graph cut problem, and therefore the optimal sub-

space based on spectral connectivity is one in which the data are most clusterable by spectral clustering. It is widely acknowledged that partitions resulting from graph cut based clustering methods tend to satisfy the low density separation assumption. The theoretical discussion in Chapter 3 offers a new perspective on this, as it is established that the optimal subspace for spectral clustering converges to the subspace normal to the largest margin hyperplane through the data as the scaling parameter is reduced to zero, and so the methods in the first two chapters are in fact intrinsically connected. In Part B. this methodology is extended to the semi-supervised setting. It is shown that if the labels are incorpoarted in a particular way, then the maximum margin result extends to this setting, and the optimal subspace for semi-supervised spectral connectivity converges to the subspace normal to the optimal TSVM solution.

Chapter 4 investigates how the exact normalised graph cut can be addressed for hyperplane based partitions. A theoretical investigation into the asymptotic properties of the normalised graph cut measured across a hyperplane is presented. It is shown that the asymptotic value of the normalised cut has desirable properties for clustering in that it both achieves low density integral, and also is likely to separate the modes of the full dimensional density without the need for penalisation. On a fixed sample, the optimal hyperplane is again shown to be asymptotically connected with the maximum margin hyperplane as the scaling parameter is reduced to zero. A highly efficient algorithm is proposed for the problem, provided the Laplace, or double exponential kernel is used to define pairwise similarities.

Finally, in Chapter 5 projected divisive partitioning is looked at within a data stream environment. Projection pursuit requires access to the entire data sample throughout the optimisation procedure, and is unsuitable for data streams in its general form. Incremental algorithms for estimating principal components, however, have been very well studied (Weng et al., 2003). Resting on the success of PDDP variants, a PCA based projected divisive clustering method is proposed. A statistically robust procedure for determining when a cluster should be split is introduced, which makes use of a proposed data stream version of the dip test for unimodality (Hartigan and Hartigan, 1985). When a cluster contains multiple modes in the underlying probability density, it is split by finding a low density hyperplane orthogonal to the PCA projection. A principled adaptive strategy is designed to handle population drift, which continuously monitors the density level on the separating hyperplanes in the model, and revises the model whenever a significant increase in density is detected.

# Chapter 2

# Minimum Density Hyperplane: An Unsupervised and Semi-supervised Classifier


## Abstract

*Associating distinct groups of objects (clusters) with contiguous regions of high probability density (high-density clusters), is central to many statistical and machine learning approaches to the classification of unlabelled data. We propose a novel hyperplane classifier for clustering and semi-supervised classification which is motivated by this objective. The proposed minimum density hyperplane minimises the integral of the empirical probability density function along it, thereby avoiding intersection with high density clusters. We show that the minimum density and the maximum margin hyperplanes are asymptotically equivalent, thus linking this approach to maximum margin clustering and semi-supervised support vector classifiers. We propose a projection pursuit formulation of the associated optimisation problem which allows us to find minimum density hyperplanes efficiently in practice, and evaluate its performance on a range of benchmark datasets. The proposed approach is found to be very competitive with state of the art methods for clustering and semi-supervised classification.*

# 1 Introduction

We study the fundamental learning problem: *Given a random sample from an unknown probability distribution with no, or partial label information, identify a separating hyperplane that avoids splitting any of the distinct groups (clusters) present in the sample.* We adopt the cluster definition given by Hartigan (Hartigan, 1975, chap. 11), in which a *high-density cluster* is defined as a maximally connected component of the level set of the probability density function, $p(\mathbf{x})$, at level $c \geqslant 0$,

$$\operatorname{lev}_c p(\mathbf{x}) = \left\{ \mathbf{x} \in \mathbb{R}^d \,|\, p(\mathbf{x}) > c \right\}.$$

An important advantage of this approach over other methods is that it is well founded from a statistical perspective, in the sense that a well-defined population quantity is being estimated.

However, since $p(\mathbf{x})$ is typically unknown, detecting high-density clusters necessarily involves estimates of this function, and standard approaches to nonparametric density estimation are reliable only in low dimensions. A number of existing *density clustering* algorithms approximate the level sets of the empirical density through a union of spheres around points whose estimated density exceeds a user-defined threshold (Walther, 1997; Cuevas et al., 2000, 2001; Rinaldo and Wasserman, 2010). The choice of this threshold affects both the shape and number of detected clusters, while an appropriate threshold is typically not known in advance. The performance of these methods deteriorates sharply as dimensionality increases, unless the clusters are assumed to be clearly discernible (Rinaldo and Wasserman, 2010). An alternative is to consider the more specific problem of allocating observations to clusters, which shifts the focus to local properties of the density, rather than its global approximation. The central idea underlying such methods is that if a pair of observations belong to the same cluster they must be connected through a path traversing only high-density regions. Graph theory is a natural choice to address this type of problem. Azzalini and Torelli (2007); Stuetzle and Nugent (2010) and Menardi and Azzalini (2014) have recently proposed algorithms based on this approach. Even these approaches however are limited to problems of low dimensionality by the standards of current applications (Menardi and Azzalini, 2014).

An equivalent formulation of the density clustering problem is to assume that clusters are separated through contiguous regions of low probability density; known as the *low-density separation* assumption. In both clustering and semi-supervised classification, identifying the hyperplane with the maximum margin is considered a direct implementation of the low-density separation approach. Motivated by the success of support vector machines (SVMs) in classification, maximum margin clustering (MMC) (Xu et al., 2004), seeks the maximum margin hyperplane to perform a binary partition (bi-

partition) of unlabelled data. MMC can be equivalently viewed as seeking the binary labelling of the data sample that will maximise the margin of an SVM estimated using the assigned labels.

In a plethora of applications data can be collected cheaply and automatically, while labelling observations is a manual task that can be performed for a small proportion of the data only. Semi-supervised classifiers attempt to exploit the abundant unlabelled data to improve the generalisation error over using only the scarce labelled examples. Unlabelled data provide additional information about the marginal density, $p(\mathbf{x})$, but this is beneficial only insofar as it improves the inference of the class conditional density, $p(\mathbf{x}|y)$. Semi-supervised classification relies on the assumption that a relationship between $p(\mathbf{x})$ and $p(\mathbf{x}|y)$ exists. The most frequently assumed relationship is that high-density clusters are associated with a single class (cluster assumption), or equivalently that class boundaries pass through low-density regions (low-density separation assumption). The most widely used semi-supervised classifier based on the low-density separation assumption is the semi supervised support vector machine ($S^3$VM) (Vapnik and Sterin, 1977; Joachims, 1999; Chapelle and Zien, 2005). $S^3$VMs implement the low-density separation assumption by partitioning the data according to the maximum margin hyperplane with respect to both labelled and unlabelled data.

Encouraging theoretical results for semi-supervised classification have been obtained under the cluster assumption. If $p(\mathbf{x})$ is a mixture of class conditional distributions, Castelli and Cover (1995, 1996) have shown that the generalisation error will be reduced exponentially in the number of labelled examples if the mixture is identifiable. More recently, Singh et al. (2009) showed that the mixture components can be identified if $p(\mathbf{x})$ is a mixture of a finite number of smooth density functions, and the separation between mixture components is large. Rigollet (2007) considers the cluster assumption in a nonparametric setting, that is in terms of density level sets, and shows that the generalisation error of a semi-supervised classifier decreases exponentially given a sufficiently large number of unlabelled data. However, the cluster assumption is difficult to verify with a limited number of labelled examples. Furthermore, the algorithms proposed by Rigollet (2007) and Singh et al. (2009) are difficult to implement efficiently even if the cluster assumption holds. This renders them impractical for real-world problems (Ji et al., 2012).

Although intuitive, the claim that maximising the margin over (labelled and) unlabelled data is equivalent to identifying the hyperplane that goes through regions with the lowest possible probability density has received surprisingly little attention. The work of Ben-David et al. (2009) is the only attempt we are aware of to theoretically investigate this claim. Ben-David et al. (2009) quantify the notion of a low-density separator by defining the *density on a hyperplane*, as the integral of the probability density function over

the hyperplane. They study the existence of universally consistent algorithms to compute the hyperplane with minimum density. The maximum hard margin classifier is shown to be consistent only in one dimensional problems. In higher dimensions only a soft-margin algorithm is a consistent estimator of the minimum density hyperplane. Ben-David et al. (2009) do not provide an algorithm to compute low density hyperplanes.

This paper introduces a novel approach to clustering and semi-supervised classification which directly identifies low-density hyperplanes in the finite sample setting. In this approach the density on a hyperplane criterion proposed by Ben-David et al. (2009) is directly minimised with respect to a kernel density estimator that employs isotropic Gaussian kernels. The density on a hyperplane provides a uniform upper bound on the value of the empirical density at points that belong to the hyperplane. This bound is tight and proportional to the density on the hyperplane. Therefore, the smallest upper bound on the value of the empirical density on a hyperplane is achieved by hyperplanes that minimise the density on a hyperplane criterion. An important feature of the proposed approach is that the density on a hyperplane can be evaluated exactly through a one-dimensional kernel density estimator, constructed from the projections of the data sample onto the vector normal to the hyperplane. This renders the computation of minimum density hyperplanes tractable even in high dimensional applications.

We establish a connection between the minimum density hyperplane and the maximum margin hyperplane in the finite sample setting. In particular, as the bandwidth of the kernel density estimator is reduced towards zero, the minimum density hyperplane converges to the maximum margin hyperplane. An intermediate result establishes that there exists a positive bandwidth such that the partition of the data sample induced by the minimum density hyperplane is identical to that of the maximum margin hyperplane. Unlike MMC and $S^3$VMs the estimation of which involves an inherently nonconvex combinatorial optimisation problem, estimating minimum density hyperplanes is a nonconvex but continuous optimisation problem, and so offers considerable computational benefits.

The remaining paper is organized as follows: The formulation of the minimum density hyperplane problem as well as basic properties are presented in Section 2. Section 3 establishes the connection between minimum density hyperplanes and maximum margin hyperplanes. Section 4 discusses the estimation of minimum density hyperplanes and the computational complexity of the resulting algorithm. Experimental results are presented in Section 5, followed by concluding remarks and future research directions in Section 6.

# 2 Problem Formulation

We study the problem of estimating a hyperplane to partition a finite dataset, $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^d$, without splitting any of the high-density clusters present. We assume that $\mathcal{X}$ is an i.i.d. sample of a random variable $\mathbf{X}$ on $\mathbb{R}^d$, with unknown probability density function $p : \mathbb{R}^d \to \mathbb{R}^+$. A hyperplane is defined as $H(\mathbf{v}, b) := \{\mathbf{x} \in \mathbb{R}^d \,|\, \mathbf{v} \cdot \mathbf{x} = b\}$, where without loss of generality we restrict attention to hyperplanes with unit normal vector, i.e., those parameterised by $(\mathbf{v}, b) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}$, where $\mathrm{bd}(\mathbb{B}^d) = \{\mathbf{v} \in \mathbb{R}^d \,|\, \|\mathbf{v}\| = 1\}$. Following Ben-David et al. (2009) we define the *density on the hyperplane* $H(\mathbf{v}, b)$ as the integral of the probability density function along the hyperplane,

$$I(\mathbf{v}, b) := \int_{H(\mathbf{v},b)} p(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{2.1}$$

We approximate $p(\mathbf{x})$ through a kernel density estimator with isotropic Gaussian kernels,

$$\hat{p}(\mathbf{x} | \mathcal{X}, h^2 I) = \frac{1}{n(2\pi h^2)^{d/2}} \sum_{i=1}^{n} \exp\left\{ -\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2h^2} \right\}. \tag{2.2}$$

This class of kernel density estimators has the useful property that the integral in Eq. (2.1) can be evaluated exactly by projecting $\mathcal{X}$ onto $\mathbf{v}$; constructing a one-dimensional density estimator with Gaussian kernels and bandwidth $h$; and evaluating the density at $b$,

$$\begin{aligned} \hat{I}(\mathbf{v}, b | \mathcal{X}, h^2 I) &:= \int_{H(\mathbf{v},b)} \hat{p}\left(\mathbf{x} | \mathcal{X}, h^2 I\right) \mathrm{d}\mathbf{x}, \\ &= \frac{1}{n\sqrt{2\pi h^2}} \sum_{i=1}^{n} \exp\left\{ -\frac{(b - \mathbf{v} \cdot \mathbf{x}_i)^2}{2h^2} \right\} =: \hat{p}_{\mathbf{v}}\left(b \,|\, \{\mathbf{v} \cdot \mathbf{x}_i\}_{i=1}^n, h^2\right). \end{aligned} \tag{2.3}$$

The univariate kernel estimator $\hat{p}_{\mathbf{v}}\left(\cdot \,|\, \{\mathbf{v} \cdot \mathbf{x}_i\}_{i=1}^n, h^2\right)$ approximates the *projected density on* $\mathbf{v}$, that is, the density function of the random variable, $X_{\mathbf{v}} = \mathbf{X} \cdot \mathbf{v}$. Henceforth we use $\hat{I}(\mathbf{v}, b)$ to approximate $I(\mathbf{v}, b)$. To simplify terminology we refer to $\hat{I}(\mathbf{v}, b)$ as the *density on* $H(\mathbf{v}, b)$, or the *density integral on* $H(\mathbf{v}, b)$, rather than the empirical density, or the empirical density integral, respectively. For notational convenience we also write $\hat{p}_{\mathbf{v}}(b)$ for $\hat{p}_{\mathbf{v}}\left(b \,|\, \{\mathbf{v} \cdot \mathbf{x}_i\}_{i=1}^n, h^2\right)$, and $\hat{I}(\mathbf{v}, b)$ for $\hat{I}(\mathbf{v}, b | \mathcal{X}, h^2 I)$, where $\mathcal{X}$ and $h$ are apparent from context.

The following Lemma, adapted from (Tasoulis et al., 2010, Lemma 3), shows that $\hat{I}(\mathbf{v}, b)$ provides an upper bound for the maximum value of the empirical density at any point that belongs to the hyperplane.

**Lemma 1** *Let $\mathcal{X}=\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^d$, and $\hat{p}(\mathbf{x}|\mathcal{X},h^2I)$ a kernel density estimator with isotropic Gaussian kernels. Then, for any $(\mathbf{v},b)\in \mathrm{bd}(\mathbb{B}^d)\times\mathbb{R}$,*

$$\max_{\mathbf{x}\in H(\mathbf{v},b)} \hat{p}(\mathbf{x}|\mathcal{X},h^2I) \leqslant (\sqrt{2\pi}h)^{1-d}\hat{I}(\mathbf{v},b). \tag{2.4}$$

This lemma shows that a hyperplane, $H(\mathbf{v},b)$, cannot intersect level sets of the empirical density with level higher than $(\sqrt{2\pi}h)^{1-d}\hat{I}(\mathbf{v},b)$. The proof of the lemma relies on the fact that projection contracts distances, and follows from simple algebra. In Eq. (2.4) equality holds if and only if there exists $\mathbf{x}\in H(\mathbf{v},b)$ and $\mathbf{c}\in\mathbb{R}^n$ such that all $\mathbf{x}_i\in\mathcal{X}$, can be written as $\mathbf{x}_i=\mathbf{x}+c_i\mathbf{v}$. It is therefore not possible to obtain a uniform upper bound on the value of the empirical density at points that belong to $H(\mathbf{v},b)$ that is lower than $(\sqrt{2\pi}h)^{1-d}\hat{I}(\mathbf{v},b)$ using only one-dimensional projections. Since the upper bound of Lemma 1 is tight and proportional to $\hat{I}(\mathbf{v},b)$, minimising the density on the hyperplane leads to the lowest upper bound on the maximum value of the empirical density along the hyperplane separator.

To obtain hyperplane separators that are meaningful for clustering and semi-supervised classification, it is necessary to constrain the set of feasible solutions, because the density on a hyperplane can be made arbitrarily low by considering a hyperplane that intersects only the tail of the density. In other words, for any $\mathbf{v}$, $\hat{I}(b|\mathbf{v})$ can be made arbitrarily low for sufficiently large $|b|$. In both problems the constraints restrict the feasible set to a subset of the hyperplanes that intersect the interior of the convex hull of $\mathcal{X}$. In detail, let $\mathrm{conv}\,\mathcal{X}$ denote the convex hull of $\mathcal{X}$, and assume $\mathrm{Int}(\mathrm{conv}\,\mathcal{X})\neq\emptyset$, where $\mathrm{Int}(\cdot)$ denotes interior. Define $C$ to be the set of hyperplanes that intersect $\mathrm{Int}(\mathrm{conv}\,\mathcal{X})$,

$$C=\left\{ H(\mathbf{v},b) \,\middle|\, (\mathbf{v},b)\in\mathrm{bd}(\mathbb{B}^d)\times\mathbb{R}, \exists \mathbf{z}\in\mathrm{Int}(\mathrm{conv}\,\mathcal{X}) \text{ s.t. } \mathbf{v}\cdot\mathbf{z}=b \right\}. \tag{2.5}$$

Then denote by $F$ the set of feasible hyperplanes, where $F\subset C$. We define the *minimum density hyperplane* (MDH), $H(\mathbf{v}^\star,b^\star)\in F$ to satisfy,

$$\hat{I}(\mathbf{v}^\star,b^\star)=\min_{(\mathbf{v},b)|H(\mathbf{v},b)\in F} \hat{I}(\mathbf{v},b). \tag{2.6}$$

In the following subsections we discuss the specific formulations for clustering and semi-supervised classification in turn.

## 2.1 Clustering

Since high-density clusters are formed around the modes of $p(\mathbf{x})$, the convex hull of these modes would be a natural choice to define the set of feasible hyperplanes. Unfortunately, this convex hull is unknown and difficult to estimate. We instead propose to constrain the distance of hyperplanes to the

origin, $b$. Such a constraint is inevitable as for any $\mathbf{v} \in \mathrm{bd}(\mathbb{B}^d)$, $\hat{p}_{\mathbf{v}}(b)$ can become arbitrarily close to zero for sufficiently large $|b|$. Obviously, such hyperplanes are inappropriate for the purposes of bi-partitioning as they assign all the data to the same partition. Rather than fixing $b$ to a constant, we constrain it in the interval,

$$F(\mathbf{v}) = [\mu_{\mathbf{v}} - \alpha\sigma_{\mathbf{v}}, \mu_v + \alpha\sigma_{\mathbf{v}}], \tag{2.7}$$

where $\mu_{\mathbf{v}}$ and $\sigma_{\mathbf{v}}$ denote the mean and standard deviation, respectively, of the projections $\{\mathbf{v} \cdot \mathbf{x}_i\}_{i=1}^n$. The parameter $\alpha \geqslant 0$, controls the width of the interval, and has a probabilistic interpretation from Chebyshev's inequality. Smaller values of $\alpha$ favour more balanced partitions of the data at the risk of excluding low density hyperplanes that separate clusters more effectively. On the other hand, increasing $\alpha$ increases the risk of separating out only a few outlying observations. We discuss in detail how to set this parameter in the experimental results section. If $\mathrm{Int}(\mathrm{conv}\,\mathcal{X}) \neq \emptyset$, then there exists $\alpha > 0$ such that the set of feasible hyperplanes for clustering, $F_{\mathrm{CL}}$, satisfies,

$$F_{\mathrm{CL}} = \left\{ H(\mathbf{v}, b) \,\middle|\, (\mathbf{v}, b) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}, b \in F(\mathbf{v}) \right\} \subset C, \tag{2.8}$$

where $C$ is the set of hyperplanes that intersect $\mathrm{Int}(\mathrm{conv}\,\mathcal{X})$, as defined in Eq. (2.5).

The minimum density hyperplane for clustering is the solution to the following constrained optimisation problem,

$$\min_{(\mathbf{v}, b) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}} \hat{I}(\mathbf{v}, b), \tag{2.9a}$$

$$\text{subject to: } b - \mu_{\mathbf{v}} + \alpha\sigma_{\mathbf{v}} \geqslant 0, \tag{2.9b}$$

$$\mu_{\mathbf{v}} + \alpha\sigma_{\mathbf{v}} - b \geqslant 0. \tag{2.9c}$$

Since the objective function and the constraints are continuously differentiable, MDHs can be estimated through constrained optimisation methods like sequential quadratic programming (SQP). Unfortunately the problem of local minima due to the nonconvexity of the objective function seriously hinders the effectiveness of this approach.

To mitigate this we propose a parameterised optimisation formulation, which gives rise to a projection pursuit approach. Projection pursuit methods optimise a measure of "interestingness" of a linear projection of a data sample, known as the projection index. For our problem the natural choice of projection index for $\mathbf{v}$ is the minimum value of the projected density within the feasible region, $\min_{b \in F(\mathbf{v})} \hat{p}_{\mathbf{v}}(b)$. This index gives the minimum density integral of feasible hyperplanes with normal vector $\mathbf{v}$. To ensure the differentiability of the projection index we incorporate a penalty term into the

objective function. We define the penalised density integral as,

$$f_{\mathrm{UL}}(\mathbf{v},b)=\hat{p}_{\mathbf{v}}(b)+\frac{L}{\eta^{\epsilon}}\max\{0,\mu_{\mathbf{v}}-\alpha\sigma_{\mathbf{v}}-b,b-\mu_{\mathbf{v}}-\alpha\sigma_{\mathbf{v}}\}^{1+\epsilon}, \tag{2.10}$$

where, $L=\left(e^{1/2}h^2\sqrt{2\pi}\right)^{-1}\geqslant\sup_{b\in\mathbb{R}}|\hat{p}'_{\mathbf{v}}(b)|$, $\epsilon\in(0,1)$ is a constant term that ensures that the penalty function is everywhere continuously differentiable, and $\eta\in(0,1)$. Other penalty functions are possible, but we only consider the above due to its simplicity, and the fact that its parameters offer a direct interpretation: $L$ in terms of the derivative of $\hat{I}(\mathbf{v},b)$; and $\eta$ in terms of the desired accuracy of the minimisers of $f_{\mathrm{UL}}(\mathbf{v},b)$ relative to the minimisers of Eq. (2.9), as discussed in the following proposition.

**Proposition 2** *For* $\mathbf{v}\in\mathrm{bd}(\mathbb{B}^d)$*, define, the set of minimisers,*

$$B(\mathbf{v})=\operatorname*{arg\,min}_{b\in F(\mathbf{v})}\hat{p}_{\mathbf{v}}(b), \tag{2.11}$$

$$B_C(\mathbf{v})=\operatorname*{arg\,min}_{b\in\mathbb{R}}f_{\mathrm{UL}}(\mathbf{v},b) \tag{2.12}$$

*For every* $b^\star\in B(\mathbf{v})$ *there exists* $b^\star_C\in B_C(\mathbf{v})$ *such that* $|b^\star-b^\star_C|\leqslant\eta$*. Moreover, there are no minimisers of* $f_{\mathrm{UL}}(\mathbf{v},b)$ *outside the interval* $[\mu_v-\alpha\sigma_{\mathbf{v}}-\eta,\mu_v+\alpha\sigma_{\mathbf{v}}+\eta]$*,*

$$B_C(\mathbf{v})\cap\mathbb{R}\setminus[\mu_v-\alpha\sigma_{\mathbf{v}}-\eta,\mu_v+\alpha\sigma_{\mathbf{v}}+\eta]=\emptyset.$$

**Proof** Any minimiser in the interior of the feasible region, $b^\star\in B(\mathbf{v})\cap\mathrm{Int}(F(\mathbf{v}))$, also minimises the penalised function, since $f_{\mathrm{UL}}(\mathbf{v},b)=\hat{p}_{\mathbf{v}}(b)\ \forall\ b\in\mathrm{Int}(F(\mathbf{v}))$, hence $b^\star\in B_C(\mathbf{v})$.

Next we consider the case when either or both of the boundary points of $F(\mathbf{v})$, $b^-=\mu_v-\alpha\sigma_{\mathbf{v}}$ and $b^+=\mu_v+\alpha\sigma_{\mathbf{v}}$, are contained in $B(\mathbf{v})$. It suffices to show that, $f_{\mathrm{UL}}(\mathbf{v},b)>\hat{p}_{\mathbf{v}}(b^-)$ for all $b<b^--\eta$, and $f_{\mathrm{UL}}(\mathbf{v},b)>\hat{p}_{\mathbf{v}}(b^+)$ for all $b>b^++\eta$. We discuss only the case $b>b^++\eta$ as the treatment of $b<b^--\eta$ is identical. Assume that $\hat{p}_{\mathbf{v}}(b)<\hat{p}_{\mathbf{v}}(b^+)$ (since in the opposite case the result follows immediately: $f_{\mathrm{UL}}(\mathbf{v},b)>\hat{p}_{\mathbf{v}}(b)>\hat{p}_{\mathbf{v}}(b^+)$). From the mean value theorem there exists $\xi\in(b^+,b)$ such that,

$$\begin{aligned}\hat{p}_{\mathbf{v}}(b^+)&=\hat{p}_{\mathbf{v}}(b)-(b-b^+)\hat{p}'_{\mathbf{v}}(\xi)\\&\leqslant\hat{p}_{\mathbf{v}}(b)+(b-b^+)L\\&<\hat{p}_{\mathbf{v}}(b)+\frac{L(b-b^+)^{1+\epsilon}}{\eta^{\epsilon}}=f_{\mathrm{UL}}(\mathbf{v},b).\end{aligned}$$

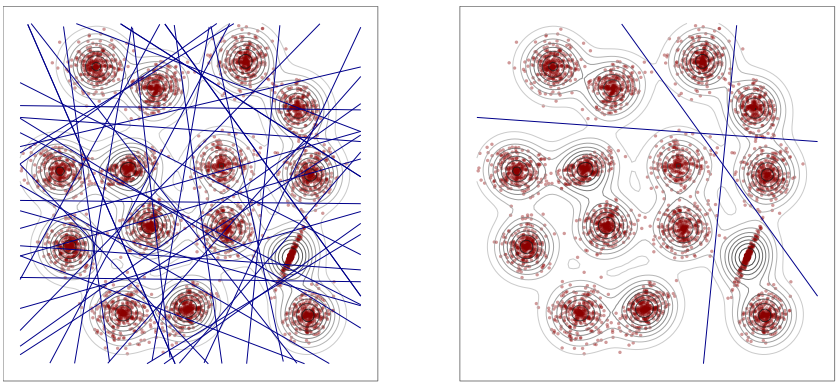
In the above we used the following facts: $\hat{p}'_{\mathbf{v}}(\xi)<0$, $L\geqslant\sup_{\xi\in\mathbb{R}}|\hat{p}'_{\mathbf{v}}(\xi)|$, and $\frac{b-b^+}{\eta}>1$. ■

We define the projection index for the clustering problem as the minimum of the penalised density integral,

$$\phi_{\mathrm{UL}}(\mathbf{v}) = \min_{b \in \mathbb{R}} f_{\mathrm{UL}}(\mathbf{v}, b). \tag{2.13}$$

Since the optimisation problem of Eq. (2.13) is one-dimensional it is simple to compute the set of global minimisers $B_C(\mathbf{v})$. As we discuss in Section 4, this is necessary to compute directional derivatives of the projection index, as well as, to determine whether $\phi_{\mathrm{UL}}$ is differentiable. We call the optimisation of $\phi_{\mathrm{UL}}$, *minimum density projection pursuit* (MDP$^2$). For each $\mathbf{v}$, MDP$^2$ considers only the optimal choice of $b$. This enables it to avoid local minima of the $\hat{p}_{\mathbf{v}}(\cdot)$. Most importantly MDP$^2$ is able to accommodate a discontinuous change in the location of the global minimiser(s), $\operatorname{argmin}_{b \in \mathbb{R}} f_{\mathrm{UL}}(\mathbf{v}, b)$, as $\mathbf{v}$ changes. Neither of the above can be achieved when the optimisation is jointly over $(\mathbf{v}, b)$ as in the original constrained optimisation problem, Eq. (2.9). The projection index $\phi_{\mathrm{UL}}$ is continuous, but it is not guaranteed to be everywhere differentiable when $B_C(\mathbf{v})$ is not a singleton. The resulting optimisation problem is therefore nonsmooth and nonconvex.

To illustrate the effectiveness of MDP$^2$ to estimate MDHs, we compare this approach with a direct optimisation of the constrained problem given in Eq. (2.9) using SQP. To enable visualisation we consider the two-dimensional S1 dataset (Fränti and Virmajoki, 2006), constructed by sampling from a Gaussian mixture distribution with fifteen components, where each component corresponds to a cluster. Figure 2.1 depicts MDHs obtained over 100 random



(a) SQP (b) MDP$^2$

**Fig. 2.1:** Binary partitions induced by 100 MDHs estimated through SQP and MDP$^2$

initialisations of SQP and MDP$^2$. It is evident that SQP frequently yields

hyperplanes that intersect regions with high probability density thus splitting clusters. As SQP always converged in these experiments the poor performance is solely due to convergence to local minima. In contrast, MDP$^2$ converges to three different solutions over the 100 experiments, all of which induce high-quality partitions, and none intersects a high-density cluster.

## 2.2 Semi-Supervised Classification

In semi-supervised classification labels are available for a subset of the data sample. The resulting classifier needs to predict as accurately as possible the labelled examples, while avoiding intersection with high-density regions of the empirical density. The formulation of the minimum density hyperplane problem can readily accommodate partially labelled data by incorporating the linear constraints associated with the labelled data into the clustering formulation. Without loss of generality assume that the first $\ell$ examples are labelled by $\mathbf{y}=(y_1,\ldots,y_\ell)^\top\in\{-1,1\}^\ell$. The MDH for semi-supervised classification is the solution to the problem,

$$\min_{(\mathbf{v},b)\in\mathrm{bd}(\mathbb{B}^d)\times\mathbb{R}} \hat{I}(\mathbf{v},b), \tag{2.14a}$$

$$\text{subject to:}\quad y_i(\mathbf{v}\cdot\mathbf{x}_i-b)\geqslant 0,\ \forall i=1,\ldots,\ell, \tag{2.14b}$$

$$b-\mu_{\mathbf{v}}+\alpha\sigma_{\mathbf{v}}\geqslant 0, \tag{2.14c}$$

$$\mu_{\mathbf{v}}+\alpha\sigma_{\mathbf{v}}-b\geqslant 0, \tag{2.14d}$$

where $\hat{I}(\mathbf{v},b),\mu_{\mathbf{v}}$, and $\sigma_{\mathbf{v}}$ are computed over the entire data set. If the labelled examples are linearly separable by a hyperplane, $H(\mathbf{v},b)$, satisfying $b\in F(\mathbf{v})$, then the constraints in Eq. (2.14) define a nonempty feasible set of hyperplanes,

$$F_{\mathrm{LB}}=\left\{H(\mathbf{v},b)\,|\,(\mathbf{v},b)\in\mathrm{bd}(\mathbb{B}^d)\times\mathbb{R}, b\in F(\mathbf{v}), y_i(\mathbf{v}\cdot\mathbf{x}_i-b)\geqslant 0,\ \forall i\in\{1,\ldots,\ell\}\right\}\subset C. \tag{2.15}$$

Eqs. (2.14c) and (2.14d) act as a *balancing constraint* which discourages MDHs that classify the vast majority of unlabelled data to a single class. Balancing constraints are included in the estimation of S$^3$VMs for the same reason (Joachims, 1999; Chapelle and Zien, 2005).

As in the case of clustering, the direct minimisation of Eq. (2.14) frequently leads to locally optimal solutions. To mitigate this we again propose a projection pursuit formulation. We define the penalised density integral for semi-supervised classification as,

$$f_{\mathrm{SSC}}(\mathbf{v},b)=f_{\mathrm{UL}}(\mathbf{v},b)+\gamma\sum_{i=1}^{l}\max\{0,-y_i(\mathbf{v}\cdot\mathbf{x}_i-b)\}^{1+\epsilon} \tag{2.16}$$

where, $\gamma > 0$ is a user-defined constant, which controls the trade-off between reducing the density on the hyperplane, and misclassifying the labelled examples. The projection index is then defined as the minimum of the penalised density integral,

$$\phi_{\mathrm{SSC}}(\mathbf{v}) = \min_{b \in \mathbb{R}} f_{\mathrm{SSC}}(\mathbf{v}, b). \tag{2.17}$$

Notice also that the projection pursuit formulation benefits from always having a solution, even in the event that the labelled data are not linearly separable.

# 3 Connection to Maximum Margin Hyperplanes

In this section we discuss the connection between MDHs and maximum (hard) margin hyperplane separators. The margin of a hyperplane $H(\mathbf{v}, b)$ with respect to a data set $\mathcal{X}$ is defined as the minimum Euclidean distance between the hyperplane and its nearest datum,

$$\operatorname{margin} H(\mathbf{v}, b) = \min_{\mathbf{x} \in \mathcal{X}} |\mathbf{v} \cdot \mathbf{x} - b|. \tag{2.18}$$

The points whose distance to the hyperplane $H(\mathbf{v}, b)$ is equal to the margin of the hyperplane, that is, $\arg\min_{\mathbf{x} \in \mathcal{X}} |\mathbf{v} \cdot \mathbf{x} - b|$, are called the *support points* of $H(\mathbf{v}, b)$. Let $F$ denote the set of feasible hyperplanes; then the *maximum margin hyperplane* (MMH) $H(\mathbf{v}^m, b^m) \in F$ satisfies,

$$\operatorname{margin} H(\mathbf{v}^m, b^m) = \max_{(\mathbf{v}, b) | H(\mathbf{v}, b) \in F} \operatorname{margin} H(\mathbf{v}, b). \tag{2.19}$$

The main result of this section is Theorem 1, which states that as the bandwidth parameter, $h$, is reduced to zero the MDH converges to the maximum margin hyperplane. An intermediate result, Lemma 4, shows that there exists a positive bandwidth, $h' > 0$ such that, for all $h \in (0, h')$, the partition of the data set induced by the MDH is identical to that of maximum margin hyperplane.

To begin with we discuss some assumptions which allow us to present the associated theoretical results of this section. As before we assume a fixed and finite data set $\mathcal{X} \subset \mathbb{R}^d$, and approximate its (assumed) underlying probability density function via a kernel density estimator using Gaussian kernels with isotropic bandwidth matrix $h^2 I$. We assume that the interior of the convex hull of the data, $\operatorname{Int}(\operatorname{conv} \mathcal{X})$, is non-empty, and define $C$ as the set of hyperplanes that intersect $\operatorname{Int}(\operatorname{conv} \mathcal{X})$, as in Eq. (2.5). The set of feasible hyperplanes, $F$, for either clustering or the semi-supervised classification satisfies $F \subset C$. By construction every $H(\mathbf{v}, b) \in F$ defines a hyperplane which partitions $\mathcal{X}$ into two non-empty subsets. Observe that if for each $\mathbf{v} \in \operatorname{bd}(\mathbb{B}^d)$

the set $\{b \in \mathbb{R} | H(\mathbf{v},b) \in F\}$ is compact, then by the compactness of $\mathrm{bd}(\mathbb{B}^d)$ a maximum margin hyperplane in $F$ exists. For both the clustering and semi-supervised classification problems this compactness holds by construction.

Now, for any $h>0$, let $(\mathbf{v}_h^\star, b_h^\star) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}$ parameterise a hyperplane which achieves the minimal density integral over all hyperplanes in $F$, for bandwidth matrix $h^2 I$. That is,

$$\hat{I}(\mathbf{v}_h^\star, b_h^\star) = \min_{(\mathbf{v},b)|H(\mathbf{v},b)\in F} \hat{I}(\mathbf{v},b|\mathcal{X},h^2 I), \tag{2.20}$$

where we briefly return to the explicit notation $\hat{I}(\mathbf{v},b|\mathcal{X},h^2 I)$ to emphasise the dependence on the bandwidth parameter, $h$. Following the approach of Tong and Koller (2000) we first show that as the bandwidth, $h$, is reduced towards zero, the density on a hyperplane is dominated by its nearest point. This is achieved by establishing that for all sufficiently small values of $h$, a hyperplane with non-zero margin has lower density integral than any other hyperplane with smaller margin.

**Lemma 3** *Take $H(\mathbf{v},b) \in F$ with non-zero margin and $0 < \delta < \mathrm{margin}\, H(\mathbf{v},b) := M_{\mathbf{v},b}$. Then $\exists h' > 0$ such that $h \in (0,h')$ and $M_{\mathbf{w},c} := \mathrm{margin}\, H(\mathbf{w},c) \leqslant M_{\mathbf{v},b} - \delta$ implies $\hat{I}(\mathbf{v},b) < \hat{I}(\mathbf{w},c)$.*

**Proof** Using Eq. (2.3) it is easy to see that,

$$\hat{I}(\mathbf{v},b) \leqslant \frac{1}{h\sqrt{2\pi}} \exp\left\{-\frac{M_{\mathbf{v},b}^2}{2h^2}\right\},$$

$$\inf\left\{\hat{I}(\mathbf{w},c) \,|\, M_{\mathbf{w},c} \leqslant M_{\mathbf{v},b} - \delta\right\} \geqslant \frac{1}{nh\sqrt{2\pi}} \exp\left\{-\frac{(M_{\mathbf{v},b}-\delta)^2}{2h^2}\right\}.$$

Therefore,

$$0 \leqslant \lim_{h\to 0^+} \frac{\hat{I}(\mathbf{v},b)}{\inf\left\{\hat{I}(\mathbf{w},c) \,|\, M_{\mathbf{w},c} \leqslant M_{\mathbf{v},b} - \delta\right\}} \leqslant \lim_{h\to 0^+} \frac{n \exp\left\{-\frac{M_{\mathbf{v},b}^2}{2h^2}\right\}}{\exp\left\{-\frac{(M_{\mathbf{v},b}-\delta)^2}{2h^2}\right\}} = 0.$$

Therefore, $\exists h' > 0$ such that $h \in (0,h') \Rightarrow \dfrac{\hat{I}(\mathbf{v},b)}{\inf\left\{\hat{I}(\mathbf{w},c) \,\middle|\, M_{\mathbf{w},c} \leqslant M_{\mathbf{v},b} - \delta\right\}} < 1.$

An immediate corollary of Lemma 3 is that as $h$ tends to zero the margin of the minimum density hyperplane tends to the maximum margin. However, this does not necessarily ensure the stronger result that the sequence of minimum density hyperplanes converges to the maximum margin hyperplane. To establish this we require two technical results, which describe some algebraic properties of the maximum margin hyperplane, and are provided

as part of the proof of Theorem 1 which is given in the appendix to this chapter.

The next lemma uses the previous result to show that there exists a positive bandwidth, $h' > 0$, such that an MDH estimated using $h \in (0, h')$ induces the same partition of $\mathcal{X}$ as the MMH. The result assumes that the maximum margin hyperplane is unique. Notice that if $\mathcal{X}$ is a sample of realisations of a continuous random variable then this uniqueness holds with probability 1.

**Lemma 4** *Suppose there is a unique hyperplane in $F$ with maximum margin, which can be parameterised by $(\mathbf{v}^m, b^m) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}$. Then $\exists h' > 0$ s.t. $h \in (0, h') \Rightarrow H(\mathbf{v}^\star_h, b^\star_h)$ induces the same partition of $\mathcal{X}$ as $H(\mathbf{v}^m, b^m)$.*

**Proof** Let $M = \mathrm{margin} H(\mathbf{v}^m, b^m)$. Since $\mathcal{X}$ is finite $\exists \delta > 0$ s.t. any hyperplane $H(\mathbf{w}, c) \in F$ inducing a partition of $\mathcal{X}$ different than that induced by $H(\mathbf{v}^m, b^m)$, satisfies $\mathrm{margin} H(\mathbf{w}, c) \leqslant M - \delta$. By Lemma 3, $\exists h' > 0$ s.t.,

$$h \in (0, h') \Rightarrow H(\mathbf{v}^\star_h, b^\star_h) \notin \{H(\mathbf{w}, c) \,|\, \mathrm{margin} H(\mathbf{w}, c) \leqslant M - \delta\},$$

which completes the proof.

The next theorem is the main result of this section, and states that the MDH converges to the MMH as the bandwidth parameter is reduced to zero. Notice that by the non-unique representation of hyperplanes, the maximum margin hyperplane has two parameterisations in $C$, namely $(\mathbf{v}^m, b^m)$ and $(-\mathbf{v}^m, -b^m)$. Convergence to the maximum margin hyperplane is therefore equivalent to showing that,

$$\min\{\|(\mathbf{v}^\star_h, b^\star_h) - (\mathbf{v}^m, b^m)\|, \|(\mathbf{v}^\star_h, b^\star_h) + (\mathbf{v}^m, b^m)\|\} \to 0 \text{ as } h \to 0^+.$$

**Theorem 1** *Suppose there is a unique hyperplane in $F$ with maximum margin, which can be parameterised by $(\mathbf{v}^m, b^m) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}$. Then,*

$$\lim_{h \to 0^+} \min\{\|(\mathbf{v}^\star_h, b^\star_h) - (\mathbf{v}^m, b^m)\|, \|(\mathbf{v}^\star_h, b^\star_h) + (\mathbf{v}^m, b^m)\|\} = 0.$$

The set $F$ used in Theorem 1 is generic so it can capture the constraints associated with both clustering and semi-supervised classification, Eq. (2.9), and Eq. (2.14) respectively. In the case of semi-supervised classification we must also assume that the labelled data are linearly separable. Theorem 1 is not directly applicable to the MDP$^2$ formulations as in this case the function being minimised is not the density on a hyperplane. The next two subsections establish this result for the MDP$^2$ formulation of the unsupervised and semi-supervised problem.

## 3.1 MDP$^2$ for Clustering

We have shown that for the constrained optimisation formulation the minimum density hyperplane converges to the maximum margin hyperplane within the feasible set, $F_{\mathrm{CL}} \subset C$. In addition, Proposition 2 places a bound on the difference between the optima for the two problems, in terms of the parameter $\eta$. Combining these we can show that the optimal solution to the penalised MDP$^2$ formulation converges to the maximum margin hyperplane in $F_{\mathrm{CL}}$, provided the parameters within the penalty term suitably depend on the bandwidth parameter, $h$. While the general case can be shown, for ease of exposition we make the simplifying assumption that the maximum margin hyperplane is strictly feasible, i.e., if $(\mathbf{v}^m, b^m)$ parameterises the maximum margin hyperplane then $b^m \in (\mu_{\mathbf{v}^m} - \alpha\sigma_{\mathbf{v}^m}, \mu_{\mathbf{v}^m} + \alpha\sigma_{\mathbf{v}^m})$.

For $h, \eta, L > 0$ define $(\mathbf{v}^\star_{h,\eta,L}, b^\star_{h,\eta,L})$ to be any global minimiser of $f_{\mathrm{CL}}$, i.e.,

$$f_{\mathrm{CL}}(\mathbf{v}^\star_{h,\eta,L}, b^\star_{h,\eta,L}) = \min_{(\mathbf{v},b)\in \mathrm{bd}(\mathbb{B}^d)\times\mathbb{R}} f_{\mathrm{CL}}(\mathbf{v}, b).$$

**Lemma 5** *Suppose there is a unique hyperplane in $F_{\mathrm{CL}}$ with maximum margin, which can be parameterised by $(\mathbf{v}^m, b^m) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}$. Suppose further that $b^m \in (\mu_{\mathbf{v}^m} - \alpha\sigma_{\mathbf{v}^m}, \mu_{\mathbf{v}^m} + \alpha\sigma_{\mathbf{v}^m})$. For $h > 0$, let $L(h) = (e^{1/2}h^2\sqrt{2\pi})^{-1}$, and $0 < \eta(h) \leqslant h$. Then,*

$$\lim_{h\to 0^+} \min\{\|(\mathbf{v}^\star_{h,\eta(h),L(h)}, b^\star_{h,\eta(h),L(h)}) - (\mathbf{v}^m, b^m)\|, \\ \|(\mathbf{v}^\star_{h,\eta(h),L(h)}, b^\star_{h,\eta(h),L(h)}) + (\mathbf{v}^m, b^m)\|\} = 0.$$

**Proof** Let $M = \mathrm{margin} H(\mathbf{v}^m, b^m)$ and as in the proof of Lemma 4, let $\delta > 0$ be such that any hyperplane inducing a different partition from $H(\mathbf{v}^m, b^m)$ has margin at most $M - \delta$. Since $H(\mathbf{v}_m, b_m)$ is strictly feasible it must be the unique maximum margin hyperplane in $F^\delta_{\mathrm{CL}} := \{(\mathbf{v}, b) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R} | b \in \mathbb{B}_{\delta/2}(F(\mathbf{v}))\}$, since the margins of the locally maximum margin hyperplanes for each partition of $\mathcal{X}$ can increase by at most $\delta/2$. We have used the notation $\mathbb{B}_{\delta/2}(F(\mathbf{v}))$ to denote the neighbourhood of $F(\mathbf{v})$ given by $\{r \in \mathbb{R} | d(r, F(\mathbf{v})) < \delta/2\}$. Observe now that for $0 < h < \delta/2$ we have $H(\mathbf{v}^\star_{h,\eta(h),L(h)}, b^\star_{h,\eta(h),L(h)}) \in F^\delta_{\mathrm{CL}}$, by Proposition 2. In addition, by Theorem 1, we know that the minimisers of $\hat{I}(\mathbf{v}, b)$ over $F^\delta_{\mathrm{CL}}$, say $H(\mathbf{v}^\delta_h, b^\delta_h)$, satisfy

$$\lim_{h\to 0^+} \min\left\{\|(\mathbf{v}^\delta_h, b^\delta_h) - (\mathbf{v}^m, b^m)\|, \|(\mathbf{v}^\delta_h, b^\delta_h) + (\mathbf{v}^m, b^m)\|\right\} = 0.$$

Now, since $H(\mathbf{v}^m, b^m)$ is strictly feasible $\exists \epsilon' > 0$ s.t. $(\mathbf{v}, b) \in \mathbb{B}_{\epsilon'}(\{(\mathbf{v}^m, b^m), -(\mathbf{v}^m, b^m)\}) \Rightarrow H(\mathbf{v}, b) \in F_{\mathrm{CL}}$. Then for any $0 < \epsilon < \epsilon'$ there exists $h' > 0$ s.t. for $0 < h < h'$ both $(\mathbf{v}^\delta_h, b^\delta_h) \in \mathbb{B}_\epsilon(\{(\mathbf{v}^m, b^m), -(\mathbf{v}^m, b^m)\}) \Rightarrow H(\mathbf{v}^\delta_h, b^\delta_h) \in F_{\mathrm{CL}}$ *and* $H(\mathbf{v}^\star_{h,\eta(h),L(h)}, b^\star_{h,\eta(h),L(h)}) \in F^\delta_{\mathrm{CL}}$. Now for $H(\mathbf{v}, b) \in F^\delta_{\mathrm{CL}} \setminus F_{\mathrm{CL}}$ we know that $\hat{I}(\mathbf{v}, b)$

$< f_{\mathrm{CL}}(\mathbf{v},b)$, whereas for $H(\mathbf{v},b) \in F_{\mathrm{CL}}, \hat{I}(\mathbf{v},b) = f_{\mathrm{CL}}(\mathbf{v},b)$ and therefore the minimiser of $f_{\mathrm{CL}}(\mathbf{v},b)$ must lie in the neighbourhood $\mathbb{B}_\epsilon(\{(\mathbf{v}^m,b^m),-(\mathbf{v}^m,b^m)\})$, and the result follows.

## 3.2 MDP$^2$ for Semi-Supervised Classification

Denote the set of hyperplanes which correctly classify the labelled data by $F_{\mathrm{LB}}$. Under the assumption that $\exists H(\mathbf{v},b) \in F_{\mathrm{LB}} \cap F_{\mathrm{CL}}$ with non-zero margin, we can show that, provided the parameter $\gamma$ does not shrink too quickly with $h$, the hyperplane that minimises $f_{\mathrm{SSC}}$ converges to the maximum margin hyperplane contained in $F_{\mathrm{LB}} \cap F_{\mathrm{CL}}$, where as before we assume that such a maximum margin hyperplane is strictly feasible. To establish this result it is sufficient to show that there exists $h' > 0$ such that for all $h \in (0,h')$, the optimal hyperplane $H(\mathbf{v}^\star_{h,\eta,L,\gamma}, b^\star_{h,\eta,L,\gamma})$ correctly classifies all the labelled examples. If this holds, then $f_{\mathrm{SSC}}(\mathbf{v}^\star_{h,\eta,L,\gamma}, b^\star_{h,\eta,L,\gamma}) = f_{\mathrm{UL}}(\mathbf{v}^\star_{h,\eta,L,\gamma}, b^\star_{h,\eta,L,\gamma})$ for all sufficiently small $h$, and hence Lemma 5 can be applied to establish the result. The proof relies on the fact that the penalty terms associated with the known labels in Eq. (2.16) are polynomials in $b$. Provided that $\gamma$ is bounded below by a polynomial in $h$, the value of the penalty terms for hyperplanes that do not correctly classify the labelled data dominate the value of the density integral as $h$ approaches zero. Therefore the optimal hyperplane must correctly classify the labelled data for small values of $h$.

**Lemma 6** *Define* $F_{\mathrm{LB}} = \{H(\mathbf{v},b) \,|\, y_i(\mathbf{v}\cdot x_i - b) > 0, \forall i = 1,\ldots,\ell\}$ *and* $F_{\mathrm{CL}} = \{H(\mathbf{v},b) \,|\, \mu_{\mathbf{v}} - \alpha\sigma_{\mathbf{v}} \leqslant b \leqslant \mu_{\mathbf{v}} + \alpha\sigma_{\mathbf{v}}\}$ *and assume that* $F_{\mathrm{SSC}} = F_{\mathrm{LB}} \cap F_{\mathrm{CL}} \neq \emptyset$ *and that* $\exists H(\mathbf{v},b) \in F_{\mathrm{SSC}}$ *with non-zero margin. For* $h > 0$, *let* $L(h) = (e^{1/2}h^2\sqrt{2\pi})^{-1}$, $0 < \eta(h) \leqslant h$ *and* $\gamma(h) \geqslant h^r$ *for some* $r > 0$. *Then there exists* $h' > 0$ *s.t.* $h \in (0,h') \Rightarrow H\left(\mathbf{v}^\star_{h,\eta(h),L(h),\gamma(h)}, b^\star_{h,\eta(h),L(h),\gamma(h)}\right) \in F_{\mathrm{LB}}$.

**Proof** Consider $H(\mathbf{v},b) \notin F_{\mathrm{LB}}$. Then,

$$f_{\mathrm{SSC}}(\mathbf{v},b) \geqslant \frac{1}{n\sqrt{2\pi}h}\exp(-\nu_\star^2/2h^2) + \gamma(h)\nu_\star^{1+\epsilon} > \gamma(h)\nu_\star^{1+\epsilon},$$

where $\nu_\star > 0$ minimises $\frac{1}{n\sqrt{2\pi}h}\exp(-\nu^2/2h^2) + \gamma(h)\nu^{1+\epsilon}$. Therefore, $\nu_\star$ is the unique positive number satisfying,

$$\begin{aligned}
&\frac{1}{n\sqrt{2\pi}h}\exp\left(-\frac{\nu_\star^2}{2h^2}\right)\left(-\frac{\nu_\star}{h^2}\right) + (1+\epsilon)\gamma(h)\nu_\star^\epsilon = 0 \\
&\qquad\qquad \Rightarrow \nu_\star^{1-\epsilon} = (1+\epsilon)\gamma(h)n\sqrt{2\pi}h^3\exp\left(\frac{\nu_\star^2}{2h^2}\right) \\
&\qquad\qquad \Rightarrow \nu_\star \geqslant \left((1+\epsilon)\gamma(h)n\sqrt{2\pi}h^3\right)^{1/1-\epsilon}.
\end{aligned}$$

We therefore have,

$$\begin{aligned} f_{\text{SSC}}(\mathbf{v},b) &> \gamma(h)\left((1+\epsilon)\gamma(h)n\sqrt{2\pi}h^3\right)^{\frac{1+\epsilon}{1-\epsilon}} \\ &= K\gamma(h)^{\frac{2}{1-\epsilon}}h^{\frac{3(1+\epsilon)}{1-\epsilon}} \\ &\geqslant Kh^{\frac{2r+3(1+\epsilon)}{1-\epsilon}}, \end{aligned}$$

where $K$ is a constant which can be chosen independent of $(\mathbf{v},b)$. Finally, for any $H(\mathbf{v}',b') \in F_{\text{SSC}}$ with non-zero margin, $\exists h' > 0$ s.t.

$$h \in (0,h') \Rightarrow f_{\text{SSC}}(\mathbf{v}',b') = \hat{I}(\mathbf{v}',b') < Kh^{\frac{2r+3(1+\epsilon)}{1-\epsilon}} < f_{\text{SSC}}(\mathbf{v},b).$$

Since $K$ is independent of $(\mathbf{v},b)$, the result follows. The final set of inequalities holds since the hyperplane $H(\mathbf{v}',b')$ is assumed to have non-zero margin, say $M_{\mathbf{v}',b'} > 0$, and hence $\hat{I}(\mathbf{v}',b') \leqslant \frac{1}{h\sqrt{2\pi}} \exp\{-M_{\mathbf{v}',b'}/2h^2\}$, which tends to zero faster than any polynomial in $h$, as $h \to 0^+$.

# 4 Estimation of Minimum Hyperplanes

In this section we discuss the computation of minimum density hyperplanes. We first investigate the continuity and differentiability properties required to optimise the projection indices $\phi_{\text{UL}}(\mathbf{v})$ and $\phi_{\text{SSC}}(\mathbf{v})$.

Since the domain of both projection indices, $\phi_{\text{UL}}(\mathbf{v})$ and $\phi_{\text{SSC}}(\mathbf{v})$, is the boundary of the unit-sphere in $\mathbb{R}^d$ it is more convenient to express $\mathbf{v}$ in terms of spherical coordinates,

$$v_i(\theta) = \begin{cases} \cos(\theta_i)\prod_{j=1}^{i-1}\sin(\theta_j), & i = 1,\ldots,d-1 \\ \prod_{j=1}^{d-1}\sin(\theta_j), & i = d, \end{cases} \tag{2.21}$$

where $\theta \in \Theta = [0,\pi]^{d-2} \times [0,2\pi]$ is called the *projection angle*. Using spherical coordinates renders the domain, $\Theta$, convex and compact, and reduces dimensionality by one.

As the following discussion applies to both $\phi_{\text{UL}}(\mathbf{v})$ and $\phi_{\text{SSC}}(\mathbf{v})$ we denote a generic projection index $\phi : \Theta \to \mathbb{R}$, and the associated set of minimisers, as,

$$\phi(\theta) = \min_{b \in A} f(\mathbf{v}(\theta),b), \tag{2.22}$$

$$B^\star(\theta) = \left\{ b \in A \,\middle|\, f(\mathbf{v}(\theta),b) = \phi(\theta) \right\}, \tag{2.23}$$

where $f(\mathbf{v}(\theta),b)$ is continuously differentiable, $A \subset \mathbb{R}$ is compact and convex, and the correspondence $B^\star(\theta)$ gives the set of global minimisers of $f(\mathbf{v}(\theta),b)$ for each $\theta$. The definition of $A$ is not critical our formulation. Setting,

$$A \supset \left[ \min_{\mathbf{v} \in \text{bd}(\mathbb{B}^d)} \{\mu_{\mathbf{v}}\} - \alpha\sigma_p - \eta, \max_{\mathbf{v} \in \text{bd}(\mathbb{B}^d)} \{\mu_{\mathbf{v}}\} + \alpha\sigma_p + \eta \right], \tag{2.24}$$

where $\sigma_p^2$ is the variance of the projections along the first principal component, ensures that the set of hyperplanes that satisfy the constraint of Eq. (2.7) will be a subset of $A$ for all $\mathbf{v}$.

Berge's maximum theorem (Berge, 1963; Polak, 1987), establishes the continuity of $\phi(\theta)$ and the upper-semicontinuity (u.s.c.) of the correspondence $B^\star(\theta)$. Theorem 3.1 in (Polak, 1987) enables us to establish that $\phi(\theta)$ is locally Lipschitz continuous. Using Theorem 4.13 of Bonnans and Shapiro (2000) we can further show that $\phi(\theta)$ is directionally differentiable everywhere. The directional derivative at $\theta$ in the direction $\nu$ is given by,

$$d\phi(\theta;\nu) = \min_{b \in B^\star(\theta)} D_\theta f(\mathbf{v}(\theta), b) \cdot \nu, \tag{2.25}$$

where $D_\theta$ denotes the derivative with respect to $\theta$. It is clear from Eq. (2.25) that $\phi(\theta)$ is differentiable if $D_\theta f(\mathbf{v}(\theta), b)$ is the same for all $b \in B^\star(\theta)$. If $B(\theta)$ is a singleton then this condition is trivially satisfied and $\phi(\theta)$ is continuously differentiable at $\theta$.

It is possible to construct examples in which $B(\theta)$ is not a singleton. However, with the exception of contrived examples, our experience with real and simulated datasets indicates that when $h$ is set through standard bandwidth selection rules $B(\theta)$ is almost always a singleton over the optimisation path.

**Proposition 7** *Suppose $B(\theta)$ is a singleton for almost all $\theta \in \Theta$. Then $\phi(\theta)$ is continuously differentiable almost everywhere.*

**Proof** The result follows immediately from the fact that if $B(\theta) = \{b\}$ is a singleton, then the derivative $D\phi(\theta) = D_\theta f(\mathbf{v}(\theta), b)$, which is continuous.

Wolfe (1972) has provided early examples of how standard gradient-based methods can fail to converge to a local optimum when used to minimise nonsmooth functions. In the last decade a new class of nonsmooth optimisation algorithms has been developed based on gradient sampling (Burke et al., 2006). Gradient sampling methods use generalised gradient descent to find local minima. At each iteration points are randomly sampled in a radius $\varepsilon$ of the current candidate solution, and the gradient at each point is computed. The convex hull of these gradients serves as an approximation of the $\varepsilon$-Clarke generalised gradient (Burke et al., 2002). The minimum element in the convex hull of these gradients is a descent direction. The gradient sampling algorithm progressively reduces the sampling radius so that the convex hull approximates the Clarke generalised gradient. When the origin is contained in the Clarke generalised gradient there is no direction of descent, and hence the current candidate solution is a local minimum. Gradient sampling achieves almost sure global convergence for functions that are locally Lipschitz continuous and almost everywhere continuously differentiable. It is also well documented that it is an effective optimisation method for functions that are only locally Lipschitz continuous.

### 4.1 Computational Complexity

In this subsection we analyse the computational complexity of MDP$^2$. At each iteration the algorithm projects the data sample onto $\mathbf{v}(\theta)$ which involves $\mathcal{O}(nd)$ operations. To compute the projection index, $\phi(\theta)$, we need to minimise the penalised density integral, $f(\mathbf{v}(\theta),b)$. This can be achieved by first evaluating $f(\mathbf{v}(\theta),b)$ on a grid of $m$ points, to bracket the location of the minimiser, and then applying bisection to compute the minimiser(s) within the desired accuracy. The main computational cost of this procedure is due to the first step which involves $m$ evaluations of a kernel density estimator with $n$ kernels. Using the improved fast Gauss transform (Morariu et al., 2008) this can be performed in $\mathcal{O}(m+n)$ operations, instead of $\mathcal{O}(mn)$. Bisection requires $\mathcal{O}(-\log_2 \varepsilon)$ iterations to locate the minimiser with accuracy $\varepsilon$.

If the minimiser of the penalised density integral $b^\star = \arg\min_{b\in A} f(\mathbf{v}(\theta),b)$, is unique the projection index is continuously differentiable at $\theta$. To obtain the derivative of the projection index it is convenient to define the projection function, $P(\mathbf{v}) = (\mathbf{x}_1\cdot\mathbf{v},\ldots,\mathbf{x}_n\cdot\mathbf{v})^\top$. An application of the chain rule yields,

$$d_\theta\phi = D_\theta f(\mathbf{v}(\theta),b^\star) = D_P f(\mathbf{v}(\theta),b^\star) D_\mathbf{v} P D_\theta \mathbf{v} \tag{2.26}$$

where the derivative of the projections of the data sample with respect to $\mathbf{v}$ is equal to the data matrix, $D_\mathbf{v} P = (\mathbf{x}_1,\ldots,\mathbf{x}_n)^\top$; and $D_\theta\mathbf{v}$ is the derivative of $\mathbf{v}$ with respect to the projection angle, which yields a $d\times(d-1)$ matrix. The computation of the derivative therefore requires $\mathcal{O}(d(n+d))$ operations.

The original GS algorithm requires $\mathcal{O}(d)$ gradient evaluations at each iteration which is costly. Curtis and Que (2013) have developed an adaptive gradient sampling algorithm that requires $\mathcal{O}(1)$ gradient evaluations in each iteration. More recently, Lewis and Overton (2013) have strongly advocated that for the minimisation of nonsmooth, nonconvex, locally Lipschitz functions, a simple BFGS method using inexact line searches is much more efficient in practice than gradient sampling, although no convergence guarantees have been established for this method. BFGS requires a single gradient evaluation at each iteration and a matrix vector operation to update the Hessian matrix approximation. In our experiments we use this algorithm.

## 5 Experimental Results

In this section we assess the empirical performance of the minimum density hyperplane approach for clustering and semi-supervised classification. We compare performance with existing state-of-the-art methods for both problems on the following 14 benchmark datasets: Banknote authentication (banknote), Breast Cancer Winsconsin original (br. cancer), Forest type mapping (forest), Ionosphere, Optical recognition of handwritten digits (optidigits),

**Table 2.1:** Details of Benchmark Data Sets

| | $n$ | $d$ | $c$ |
|---|---|---|---|
| banknote[a] | 1372 | 4 | 2 |
| br. cancer[a] | 699 | 9 | 2 |
| forest[a] | 523 | 27 | 4 |
| ionosphere[a] | 351 | 33 | 2 |
| optidigits[a] | 5618 | 64 | 10 |
| pendigits[a] | 10992 | 16 | 10 |
| seeds[a] | 210 | 7 | 3 |
| smartphone [a] | 10929 | 561 | 12 |
| image seg.[a] | 2309 | 18 | 7 |
| satellite[a] | 6435 | 36 | 6 |
| synth[a] | 600 | 60 | 6 |
| voting[a] | 435 | 16 | 2 |
| wine[a] | 178 | 13 | 3 |
| yeast[b] | 698 | 72 | 5 |

a. UCI machine learning repository `https://archive.ics.uci.edu/ml/datasets.html`

b. Stanford Yeast Cell Cycle Analysis Project `http://genome-www.stanford.edu/cellcycle/`

Pen-based recognition of hand-written digits (pendigits), Seeds, Smartphone-Based Recognition of Human Activities and Postural Transitions (smartphone), Statlog Image Segmentation (image seg.), Statlog Landsat Satellite (satellite), Synthetic control chart time series (synth control), Congressional voting records (voting), Wine, and Yeast cell cycle analysis (yeast). Details of these data sets, in terms of their size $n$, dimensionality $d$ and number of clusters $c$, can be seen in Table 2.1.

## 5.1 Clustering

Since MDHs yield a bi-partition of a dataset rather than a complete clustering, we propose two measures to assess the quality of a binary partition of a dataset containing an arbitrary number of clusters. Both take values in $[0,1]$ with larger values indicating a better partition. These measures are motivated by the fact that a good binary partition should (a) avoid dividing clusters between elements of the partition, and (b) be able to discriminate at least one cluster from the rest of the data. To capture this we modify the cluster labels of the data by assigning each cluster to the element of the binary partition which contains the majority of its members. In the case of a tie the cluster is assigned to the smaller of the two partitions. We thus merge the true clusters into two aggregate clusters, $C_1$ and $C_2$.

The first measure we use is the binary V-measure which is simply the V-measure (Rosenberg and Hirschberg, 2007) computed on $C_1, C_2$ with respect to the binary partition, which we denote $\Pi_1, \Pi_2$. The V-measure is the harmonic mean of homogeneity and completeness. For a dataset containing clusters $C_1, \ldots, C_c$, partitioned as $\Pi_1, \ldots, \Pi_k$, homogeneity is defined as the conditional entropy of the cluster distribution within each partition, $\Pi_i$. Completeness is symmetric to homogeneity and measures the conditional entropy of each partition within each cluster, $C_j$. An important characteristic of the V-measure for evaluating binary partitions is that if the distribution of clusters within each partition is equal to the overall cluster distribution in the data set then the V-measure is equal to zero (Rosenberg and Hirschberg, 2007). This means that if an algorithm fails to distinguish the majority of any of the clusters from the remainder of the data, the binary V-measure returns zero performance. Other evaluation metrics for clustering, such as purity and the Rand index, can assign a high value to such partitions.

To define the second performance measure we first determine the number of correctly and incorrectly classified samples. The error of a binary partition, $\mathrm{E}(\Pi_1, \Pi_2)$, given in Eq. (2.27), is defined as the number of elements of each aggregate cluster which are not in the same partition as the majority of their original clusters. In contrast, the success of a partition, $\mathrm{S}(\Pi_1, \Pi_2)$, Eq. (2.28), measures the number of samples which are in the same partition as the majority of their original clusters. The Success Ratio, $\mathrm{SR}(\Pi_1, \Pi_2)$, Eq. (2.29), captures the extent to which the majority of at least one cluster is well-distinguished from the rest of the data.

$$\mathrm{E}(\Pi_1, \Pi_2) = \min\{|\Pi_1 \cap C_1| + |\Pi_2 \cap C_2|, |\Pi_1 \cap C_2| + |\Pi_2 \cap C_1|\}, \tag{2.27}$$

$$\mathrm{S}(\Pi_1, \Pi_2) = \min\{\max\{|\Pi_1 \cap C_1|, |\Pi_1 \cap C_2|\}, \max\{|\Pi_2 \cap C_1|, |\Pi_2 \cap C_2|\}\}, \tag{2.28}$$

$$\mathrm{SR}(\Pi_1, \Pi_2) = \frac{\mathrm{S}(\Pi_1, \Pi_2)}{\mathrm{S}(\Pi_1, \Pi_2) + \mathrm{E}(\Pi_1, \Pi_2)}. \tag{2.29}$$

Similarly to the binary V-measure defined above, Success Ratio takes the value zero if an algorithm fails to distinguish the majority of any cluster from the remainder of the data.

### Parameter Settings for MDP$^2$

The two most important settings for the performance of the proposed approach are the initial projection direction, and the choice of $\alpha$, which controls the width of the interval $F(\mathbf{v})$ within which the optimal hyperplane falls. Despite the ability of the MDP$^2$ formulation to mitigate the effect of local minima in the projected density $\hat{p}_{\mathbf{v}}(b)$, the problem remains non-convex and local minima in the projection index can still lead to suboptimal performance.

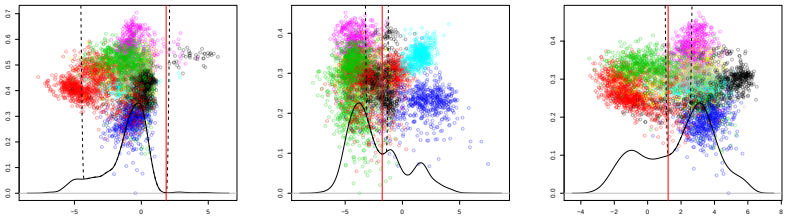
We have found that this effect is amplified in general when either or both the number of dimensions, and the number of high density clusters in the dataset is large. To better handle the effect of local optima, we use multiple initialisations and select the MDH that maximises the *relative depth* criterion, defined in Eq. (2.30). The relative depth of an MDH, $H(\mathbf{v},b)$, is defined as the smaller of the relative differences in the density on the MDH and its two adjacent modes in the projected density, $\hat{p}_{\mathbf{v}}(\cdot)$,

$$\text{RelativeDepth}(\mathbf{v},b) = \min\left\{\frac{\hat{p}_{\mathbf{v}}(m_l)-\hat{p}_{\mathbf{v}}(b)}{\hat{p}_{\mathbf{v}}(b)}, \frac{\hat{p}_{\mathbf{v}}(m_r)-\hat{p}_{\mathbf{v}}(b)}{\hat{p}_{\mathbf{v}}(b)}\right\}, \tag{2.30}$$

where $m_l$ and $m_r$ are the two adjacent modes in $\hat{p}_{\mathbf{v}}(\cdot)$. If an MDH does not separate the modes of the projected density, $\hat{p}_{\mathbf{v}}(\cdot)$, then its relative depth is set to zero, signalling a failure of MDP$^2$ to identify a meaningful bi-partition. The relative depth is appealing because it captures the fact that a high quality separating hyperplane should have a low density integral, and separate well the modes of the projected density $\hat{p}_{\mathbf{v}}(\cdot)$. Note also that the relative depth is equivalent to the inverse of a measure used to define cluster overlap in the context of Gaussian mixtures (Aitnouri et al., 2000). In all the reported experiments we initialise MDP$^2$ to the first and second principal component and select the MDH with the largest relative depth. For the data sets listed above it was never the case that both initialisations led to MDHs with zero relative depth.

The choice of $\alpha$ determines the trade-off between a balanced bi-partition and the ability to discover lower density hyperplanes. The difficulties associated with choosing this parameter are illustrated in Figure 2.2. In each sub-figure the horizontal axis is the candidate projection vector, $\mathbf{v}$, while the right vertical axis is the direction of maximum variability orthogonal to $\mathbf{v}$. Points correspond to projections of the data sample onto this two-dimensional space, while colour indicates cluster membership. The solid line depicts the projected density on $\mathbf{v}$, $\hat{p}_{\mathbf{v}}(\cdot)$, while the dotted line depicts the penalised function, $f_{\text{UL}}(\mathbf{v},\cdot)$. The scale of both functions is depicted in the left vertical axis. The solid vertical line indicates the MDH along $\mathbf{v}$. Setting $\alpha$ to a large value can cause MDP$^2$ to focus on hyperplanes that have low density because they partition only a small subset of the dataset as shown in Figure 2.2(a). In contrast smaller values of $\alpha$ may cause the algorithm to disregard valid lower density hyperplane separators (see Figure 2.2(b)), or for the separating hyperplane to not be a local minimiser of the projected density (see Figure 2.2(c)).

Rather than selecting a single value for $\alpha$ we recommend solving MDP$^2$ repeatedly for an increasing sequence of values in the range $\{\alpha_{\min},\alpha_{\max}\}$, where each implementation beyond the first is initialised using the solution to the previous. Setting $\alpha_{\min}$ close to zero forces MDP$^2$ to seek low density hyperplanes that induce a balanced data partition. This tends to find projec-



(a) MDH separating few observations

(b) Lower density hyperplane beyond feasible region

(c) MDH not a minimiser of the projected density

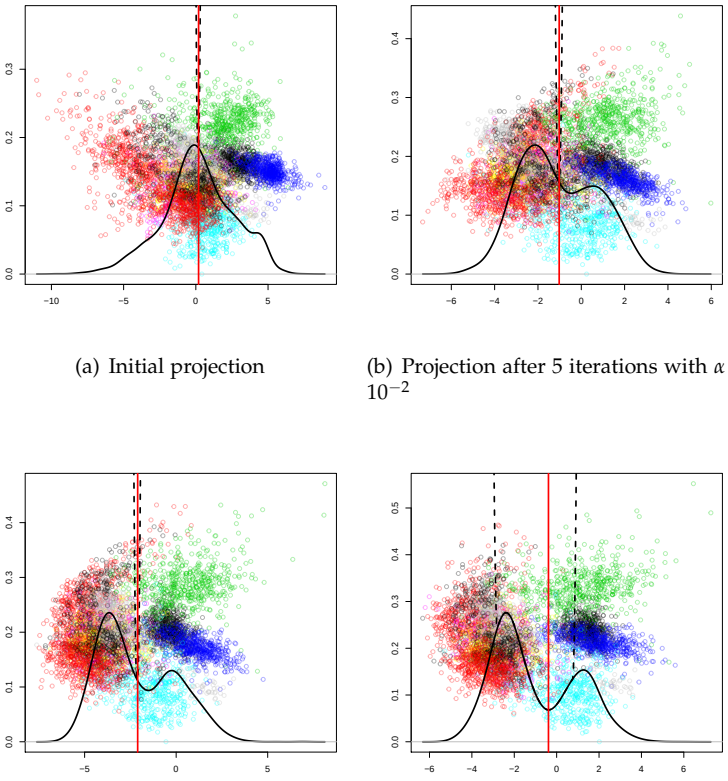
**Fig. 2.2:** Impact of choice of $\alpha$ on minimum density hyperplane.

tions which display strong multimodal structure, yet prevents convergence to hyperplanes that have low density because they partition a few observations, as in the case shown in Figure 2.2(a). Increasing $\alpha$ progressively fine-tunes the location of the MDH. To avoid sensitivity to the value of $\alpha_{\max}$ (set to 0.9) the output of the algorithm is the last hyperplane that corresponds to a minimiser of $\hat{p}_{\mathbf{v}}(\cdot)$. Figure 2.3 illustrates this approach using the optical recognition of handwritten digits dataset from the UCI machine learning repository (Lichman, 2013). Figure 2.3(a) depicts the projected density on the initial projection direction, which in this case is the second principal component. As shown, the density is unimodal and the clusters are not well separated along this vector. Although not shown, if a large value of $\alpha$ is used from the outset, MDP$^2$ will identify a vector for which $\hat{p}_{\mathbf{v}}(\cdot)$ is unimodal and skewed. Figure 2.3(b) shows that after five iterations with $\alpha = 10^{-2}$ MDP$^2$ has identified a projection vector such that $\hat{p}_{\mathbf{v}}(\cdot)$ is bimodal. In subsequent iterations the two modes become more clearly separated, Figure 2.3(c), while increasing $\alpha$ enables MDP$^2$ to locate an MDH that corresponds to a minimiser of $\hat{p}_{\mathbf{v}}(\cdot)$, as illustrated in Figure 2.3(d).

### Performance Evaluation

We compare the performance of MDP$^2$ for clustering with the following methods:

1. $k$-means++ (Arthur and Vassilvitskii, 2007), a version of $k$-means that is guaranteed to be $\mathcal{O}(\log k)$–competitive to the optimal $k$-means clustering.
2. The adaptive linear discriminant analysis guided $k$-means (LDA-$k$m) (Ding and Li, 2007). LDA-$k$m attempts to discover the most discriminative linear subspace for clustering by iteratively using $k$-means, to assign labels



(a) Initial projection

(b) Projection after 5 iterations with $\alpha = 10^{-2}$

(c) Projection after 25 iterations with $\alpha = 10^{-2}$

(d) Final projection with $\alpha = \alpha_{\max}$

**Fig. 2.3:** Evolution of the minimum density hyperplane through consecutive iterations.

to observations, and LDA to identify the most discriminative subspace.

3. The principal direction divisive partitioning (PDDP) (Boley, 1998), and the density-enhanced PDDP (dePDDP) (Tasoulis et al., 2010). Both methods project the data onto the first principal component. PDDP splits at the mean of the projections, while dePDDP splits at the lowest local minimum of the one-dimensional density estimator.
4. The iterative support vector regression algorithm for MMC (Zhang et al., 2009) using the inner product and Gaussian kernel, iSVR-L and iSVR-G respectively. Both are initialised with the output of 2-means++.
5. Normalised cut spectral clustering (SCn) (Ng et al., 2002) using the Gaussian affinity function, and the automatic bandwidth selection method of Zelnik-Manor and Perona (2004). This choice of kernel and band-

width produced substantially better performance than alternative choices considered. For datasets that are too large for the eigen decomposition of the Gram matrix to be feasible we employed the Nyström method (Fowlkes et al., 2004).

We also considered the density-based clustering algorithm PdfCluster (Menardi and Azzalini, 2014), but this algorithm could not be executed on the larger datasets and so its performance is not reported herein. With the exception of SCn and iSVR-G, the methods considered bi-partition the data through a hyperplane in the original feature space. For the 2-means and LDA-2m algorithm the hyperplane separator bisects the line segment joining the two centroids. iSVR-L directly seeks the maximum margin hyperplane in the original space, while iSVR-G seeks the maximum margin hyperplane in the feature space defined by the Gaussian kernel. PDDP and dePDDP use a hyperplane whose normal vector is the first principal component. PDDP uses a fixed split point while dePDDP uses the hyperplane with minimum density along the fixed projection direction.

Table 2.2 reports the performance of the considered methods with respect to the success ratio (SR) and the binary V-measure (V-m) on the fourteen datasets. In addition Figures 2.4(a) and 2.4(b) provide summaries of the overall performance on all datasets using boxplots of the raw performance measures as well as the associated *regret*. The regret of an algorithm on a given dataset is defined as the difference between the best performance attained on this dataset and the performance of this algorithm. By comparing against the best performing clustering algorithm regret accommodates for differences in difficulty between clustering problems, while also making use of the magnitude of performance differences between algorithms. The distribution of performance with respect to both SR and V-m is negatively skewed for most methods, and as a result the median is higher than the mean (indicated with a red dot).

It is clear from Table 2.2 that no single method is consistently superior to all others, although MDP$^2$ achieves the highest or tied highest performance on seven datasets (more than any other method). More importantly MDP$^2$ is among the best performing methods in almost all cases. This fact is better captured by the regret distributions in Figure 2.4(b). Here we see that the average, median, and maximum regret of MDP$^2$ is substantially lower than any of the competing methods. In addition MDP$^2$ achieves the highest mean and median performance with respect to both SR and V-m, while also having much lower variability in performance when compared with most other methods.

Pairwise comparisons between MDP$^2$ and other methods reveal some less obvious facts. SCn achieves higher performance than MDP$^2$ in more examples (six) than any other competing method, however it is much less consis-

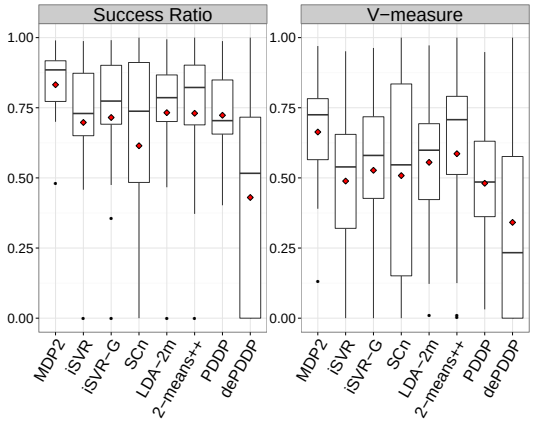
| | MDP$^2$ | | iSVR-L | | iSVR-G | | SCn | | LDA-2m | | 2-means++ | | PDDP | | dePDDP | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | SR | V-m | SR | V-m | SR | V-m | SR | V-m | SR | V-m | SR | V-m | SR | V-m | SR | V-m |
| banknote | **0.79** | **0.55** | 0.00 | 0.00 | 0.35 | 0.00 | 0.46 | 0.10 | 0.00 | 0.01 | 0.37 | 0.01 | 0.40 | 0.03 | 0.00 | 0.03 |
| br. cancer | **0.91** | **0.79** | 0.73 | 0.56 | 0.73 | 0.56 | 0.00 | 0.13 | 0.87 | 0.71 | 0.87 | 0.72 | 0.91 | 0.78 | 0.90 | 0.77 |
| forest | 0.78 | 0.67 | 0.90 | 0.72 | **0.91** | **0.74** | 0.56 | 0.41 | 0.76 | 0.63 | 0.72 | 0.58 | 0.64 | 0.36 | 0.00 | 0.00 |
| image seg. | 0.89 | 0.72 | 0.82 | 0.59 | 0.88 | 0.71 | 0.92 | 0.87 | 0.78 | 0.58 | 0.78 | 0.71 | 0.87 | 0.67 | **1.00** | **1.00** |
| ionosphere | 0.48 | 0.13 | 0.47 | 0.13 | 0.47 | 0.13 | **0.55** | **0.22** | 0.47 | 0.12 | 0.47 | 0.12 | 0.47 | 0.12 | 0.42 | 0.09 |
| optidigits | **0.93** | **0.85** | 0.63 | 0.29 | 0.82 | 0.60 | 0.00 | 0.00 | 0.81 | 0.62 | 0.92 | 0.82 | 0.68 | 0.30 | 0.00 | 0.00 |
| pendigits | 0.74 | 0.39 | 0.79 | 0.55 | **0.88** | **0.68** | 0.80 | 0.68 | 0.79 | 0.55 | 0.78 | 0.57 | 0.79 | 0.54 | 0.61 | 0.42 |
| satellite | 0.89 | 0.75 | 0.73 | 0.40 | 0.73 | 0.40 | **0.92** | **0.86** | 0.73 | 0.40 | 0.87 | 0.81 | 0.71 | 0.37 | 0.00 | 0.00 |
| seeds | 0.88 | 0.73 | 0.71 | 0.53 | 0.71 | 0.53 | 0.89 | 0.76 | **0.96** | **0.90** | 0.86 | 0.70 | 0.75 | 0.59 | 0.73 | 0.60 |
| smartphone | **0.99** | **0.97** | 0.99 | 0.95 | 0.99 | 0.96 | 0.99 | 0.94 | 0.99 | 0.97 | 0.99 | 0.94 | 0.99 | 0.95 | 0 .00 | 0.00 |
| synth | 0.98 | 0.94 | 0.94 | 0.83 | 0.94 | 0.83 | **1.00** | **1.00** | 0.88 | 0.76 | **1.00** | **1.00** | 0.69 | 0.51 | **1.00** | **1.00** |
| voting | **0.70** | **0.43** | 0.46 | 0.09 | 0.00 | 0.00 | 0.00 | 0.05 | 0.69 | 0.41 | 0.00 | 0.00 | 0.70 | 0.40 | 0.68 | 0.38 |
| wine | **0.77** | **0.61** | 0.70 | 0.52 | 0.69 | 0.50 | 0.67 | 0.48 | 0.66 | 0.48 | 0.68 | 0.49 | 0.65 | 0.46 | 0.68 | 0.49 |
| yeast | **0.92** | **0.76** | 0.89 | 0.68 | 0.91 | 0.72 | 0.84 | 0.61 | 0.86 | 0.63 | 0.91 | 0.73 | 0.87 | 0.65 | 0.00 | 0.00 |
| Average Improvement | | | 0.13 | 0.18 | 0.12 | 0.14 | 0.22 | 0.16 | 0.10 | 0.11 | 0.10 | 0.08 | 0.11 | 0.18 | 0.40 | 0.32 |

**Table 2.2:** Performance on the task of binary partitioning. (Ties in best performance were resolved by considering more decimal places)

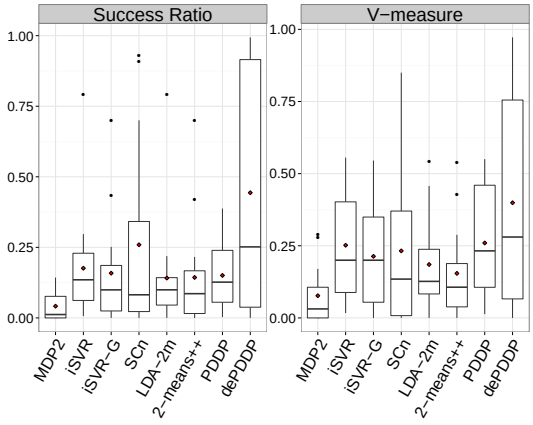
tent in its performance, obtaining very poor performance on five of the data sets. The iSVR maximum margin clustering approach is arguably the closest competitor to MDP$^2$. iSVR-L and iSVR-G achieve the second and third highest average performance with respect to V-m and SR respectively. The PDDP algorithm is the second best performing method on average with respect to SR, but performs poorly with respect to V-m. The density enhanced variant, dePDDP, performs on average much worse than MDP$^2$. This approach is similarly motivated by obtaining hyperplanes with low density integral, and its low average performance indicates the usefulness of searching for high quality projections as opposed to always using the first principal component. Finally, neither of the *k*-means variants appears to be competitive with MDP$^2$ in general.

## 5.2 Semi-Supervised Classification

In this section we evaluate MDHs for semi-supervised classification. We compare MDHs against three state-of-the-art semi-supervised classification methods: Laplacian Regularised Support Vector Machines (LapSVM) (Belkin et al., 2006), Simple Semi-Supervised Learning (SSSL) (Ji et al., 2012), and Correlated Nyström Views (XNV) (McWilliams et al., 2013). For all methods the inner product kernel was used to render the resulting classifiers linear, and




(a) Raw Performance Measure



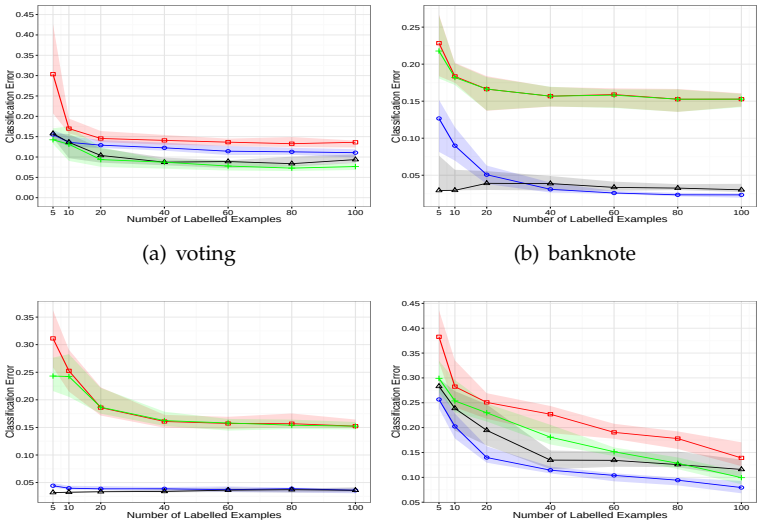

(b) Regret

**Fig. 2.4:** Performance and Regret Distributions for all Methods Considered

thereby comparable to the minimum density hyperplane approach. As the MDH is asymptotically equivalent to a linear $S^3$VM we also considered the continuous formulation for the estimation of a $S^3$VM proposed by Chapelle and Zien (2005). These results are omitted as this method was not competitive on any of the considered datasets.

### Parameter Settings for MDP$^2$

The existence of a few labelled examples enables an informed initialisation of MDP$^2$. We consider the first and second principal components as well as the weight vector of a linear SVM trained on the labelled examples only, and




(a) voting (b) banknote

(c) breast cancer (d) ionosphere

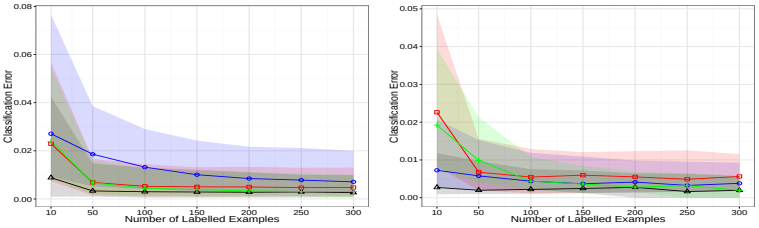
MDP$^2$ median (—△—), LapSVM median (—○—), SSSL median (—□—), XNV median (—+—), with corresponding interquartile ranges given by shaded regions.

**Fig. 2.5:** Classification error for different number of labelled examples for datasets with two clusters.

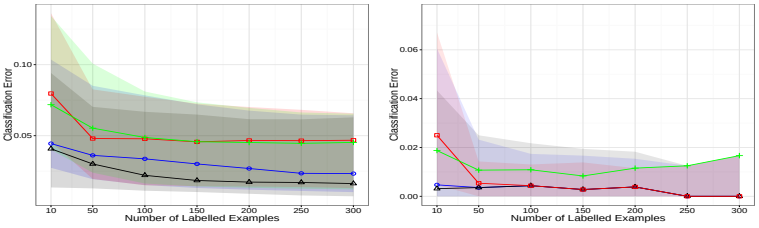
initialise MDP$^2$ with the vector that minimises the value of the projection index, $\phi_{\text{SSC}}$. The penalty parameter $\gamma$ is first set to 0.1 and with this setting $\alpha$ is progressively increased in the same way as for clustering. After this, $\alpha$ is kept at $\alpha_{\max}$ and $\gamma$ is increased to 1 and then 10. Thus the emphasis is initially on finding a low density hyperplane with respect to the marginal density $\hat{p}(\mathbf{x})$. As the algorithm progresses the emphasis on correctly classifying the labelled examples increases, so as to obtain a hyperplane with low training error within the region of low density already determined.

### Performance Evaluation

To assess the effect on performance of the number of labelled examples, $\ell$, we consider a range of values. We compare the methods using the subset of datasets used in the previous section in which the size of the smallest class exceeds 100. In total eight datasets are used. For each value of $\ell$, 30 random partitions into labelled and unlabelled data are considered. As classes are balanced in the datasets considered, performance is measured only in terms of classification error on the unlabelled data. For datasets with more than two classes all pairwise combinations of classes are considered and aggregate performance is reported.




(a) pendigits (b) optidigits




(c) satellite (d) image segmentation

MDP$^2$ median (—△—), LapSVM median (—○—), SSSL median (—□—), XNV median (—+—), with corresponding interquartile ranges given by shaded regions.

**Fig. 2.6:** Classification error for different numbers of labelled examples over all pairwise combinations of classes.

Figure 2.5 provides plots of the median and interquartile range of the classification error for values of $\ell$ between 5 and 100 for the four datasets with two classes. Overall MDP$^2$ appears to be most competitive when the number of labelled examples is small. In addition, MDP$^2$ is comparable with the best performing method in almost every case. The only exception is the ionosphere dataset where LapSVM outperforms MDP$^2$ for all values of $\ell$. Figure 2.6 provides plots of the median and interquartile range of the aggregate classification error on datasets containing more than two classes. As these datasets are larger we consider up to 300 labelled examples. Note that the interquartile range for XNV is not depicted for the satellite dataset. The variability of performance of XNV on this dataset was so high that including the interquartile range would obscure all other information in the figure. MDP$^2$ exhibits the best performance overall, and obtains the lowest median classification error, or tied lowest, for all datasets and values of $\ell$.

## 5.3 Summary of Experimental Results

We evaluated the performance of the MDP$^2$ formulation for finding minimum density hyperplanes for both clustering and semi-supervised classification, on a large collection of benchmark datasets, and in comparison with

state-of-the-art methods for both problems.

For clustering, we found that no single method was consistently superior to all others. This is a result of the vastly differing nature of the datasets in terms of size, dimensionality, number and shape of clusters, etc. MDP$^2$ achieved the best performance on more datasets than any of the competing methods, and importantly was competitive with the best performing method in almost every dataset considered. All other methods performed poorly in at least as many examples. Boxplots of both the raw performance and performance regret, which measures the difference between each method and the best performing method on each dataset, allowed us to summarise the comparative performance of the different methods across datasets. The mean and median raw performance of MDP$^2$ is substantially higher than the next best performing method, and the regret is also substantially lower.

In the case of semi-supervised classification it was apparent that MDP$^2$ is extremely competitive when the number of labelled examples is (very) small, but that in some cases its performance does not improve as much as that of the other methods considered, when the labelled examples become more abundant. Our experiments suggest that overall MDP$^2$ is very competitive with the state-of-the-art for semi-supervised classification problems.

# 6 Conclusions

We proposed a new hyperplane classifier for clustering and semi-supervised classification. The proposed approach is motivated by determining low density linear separators of the high-density clusters within a dataset. This is achieved by minimising the integral of the empirical density along the hyperplane, which is computed through kernel density estimation. To the best of our knowledge this is the first direct implementation of the low density separation assumption that underlies high-density clustering and numerous influential semi-supervised classification methods. We show that the minimum density hyperplane classifier is asymptotically connected with maximum margin support vector classifiers, thereby establishing an important link between the proposed approach, maximum margin clustering, and semi-supervised support vector machines.

The proposed formulation allows us to evaluate the integral of the density on a hyperplane by projecting the data onto the vector normal to the hyperplane, and estimating a univariate kernel density estimator. This enables us to apply our method effectively and efficiently on datasets of much higher dimensionality than is generally possible for density based clustering methods. To mitigate the problem of convergence to locally optimal solutions we proposed a projection pursuit formulation.

We evaluated the minimum density hyperplane approach on a large col-

lection of benchmark datasets. The experimental results obtained indicate that the method is competitive with state-of-the-art methods for clustering and semi-supervised classification. Importantly the performance of the proposed approach displays low variability across a variety of datasets, and is robust to differences in data size, dimensionality, and number of clusters. In the context of semi-supervised classification, the proposed approach shows especially good performance when the number of labelled data is small.

## Acknowledgements

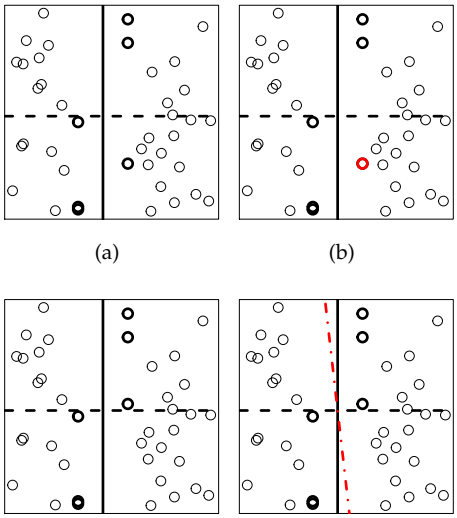
We wish to thank the reviewers for their valuable comments and suggestions which greatly improved this paper. Nicos Pavlidis would like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the programme Inference for Change-Point and Related Processes where work on this paper was undertaken. David Hofmeyr gratefully acknowledges the support of the EPSRC funded EP/H023151/1 STOR-i centre for doctoral training, as well as the Oppenheimer Memorial Trust. We thank Prof. David Leslie, and Dr. Teemu Roos for valuable comments and suggestions on this work.

## 7 Proof of Theorem 1

Before proving Theorem 1 we require the following two technical lemmata which establish some algebraic properties of the maximum margin hyperplane. The following lemma shows that any hyperplane orthogonal to the maximum margin hyperplane results in a different partition of the support points of the maximum margin hyperplane. The proof relies on the fact that if this statement does not hold then a hyperplane with larger margin exists which is a contradiction. Figure 2.7 provides an illustration of why this result holds. (a) Any hyperplane orthogonal to MMH generates a different partition of the support points of MMH, e.g., the point highlighted in red in (b) is grouped with the lower three by the dotted line but with the upper two by the solid line, the MMH. If an orthogonal hyperplane *can* generate the same partition (c), then a larger margin hyperplane than the proposed MMH exists (d).

**Lemma 8** *Suppose there is a unique hyperplane in $F$ with maximum margin, which can be parameterised by $(\mathbf{v}^m, b^m) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}$. Let $M = \mathrm{margin}\, H(\mathbf{v}^m, b^m)$, $C^+ = \{\mathbf{x} \in \mathcal{X} \,|\, \mathbf{v}^m \cdot \mathbf{x} - b^m = M\}$ and $C^- = \{\mathbf{x} \in \mathcal{X} \,|\, b^m - \mathbf{v}^m \cdot \mathbf{x} = M\}$. Then, $\forall \mathbf{w} \in \mathrm{Null}(\mathbf{v}^m)$, $c \in \mathbb{R}$ either $\min\{\mathbf{w} \cdot \mathbf{x} - c \,|\, \mathbf{x} \in C^+\} \leqslant 0$, or $\max\{\mathbf{w} \cdot \mathbf{x} - c \,|\, \mathbf{x} \in C^-\} \geqslant 0$.*

**Proof** Suppose the result does not hold, then $\exists (\mathbf{w}, c)$ with $\|\mathbf{w}\| = 1, \mathbf{w} \cdot \mathbf{v}^m = 0$ and $\min\{\mathbf{w} \cdot \mathbf{x} - c | \mathbf{x} \in C^+\} > 0$ and $\max\{\mathbf{w} \cdot \mathbf{x} - c | \mathbf{x} \in C^-\} < 0$. Let $m = \min\{|\mathbf{w} \cdot$



(a) (b)

(c) (d)

Proposed MMH ——, Orthogonal hyperplane - - -, Hyperplane with larger margin - · - · -, Regular points ◯, Support points ◯,
Differently assigned support point ◯

**Fig. 2.7:** Two dimensional illustration of Lemma 8

$\mathbf{x}-c\,|\,\mathbf{x}\in C^+\cup C^-\}$. Define $\lambda=\frac{m}{2M}<1$. Define $\mathbf{u}=\frac{1}{\sqrt{\lambda^2+(1-\lambda)^2}}(\lambda\mathbf{w}+(1-\lambda)\mathbf{v}^m)$ and $d=\frac{\lambda c+(1-\lambda)b^m}{\sqrt{\lambda^2+(1-\lambda)^2}}$. By construction $\|\mathbf{u}\|=1$. For any $\mathbf{x}_+\in C^+$ we have,

$$\begin{aligned}\mathbf{u}\cdot\mathbf{x}_+-d&=\frac{\lambda(\mathbf{w}\cdot\mathbf{x}_+-c)+(1-\lambda)(\mathbf{v}^m\cdot\mathbf{x}_+-b^m)}{\sqrt{\lambda^2+(1-\lambda)^2}}\\&\geqslant\frac{\lambda m+(1-\lambda)M}{\sqrt{\lambda^2+(1-\lambda)^2}}\\&=\frac{m^2+2M^2-Mm}{\sqrt{m^2+(2M-m)^2}}\\&>M.\end{aligned}$$

Similarly one can show that $d-\mathbf{u}\cdot\mathbf{x}_-<M$ for any $\mathbf{x}_-\in C^-$, meaning that $(\mathbf{u},d)$ achieves a larger margin on $C^+$ and $C^-$ than $(\mathbf{v}^m,b^m)$, a contradiction.

The next lemma uses the above result to provide an upper bound on the distance between pairs of support points projected onto any vector, in terms of the angle between that vector and $\mathbf{v}^m$.

**Lemma 9** *Suppose there is a unique hyperplane in $F$ with maximum margin, which can be parameterised by $(\mathbf{v}^m, b^m) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}$. Define $M = \mathrm{margin}\, H(\mathbf{v}^m, b^m)$, $C^+ = \{\mathbf{x} \in \mathcal{X} | \mathbf{v}^m \cdot \mathbf{x} - b^m = M\}$, and $C^- = \{\mathbf{x} \in \mathcal{X} | b^m - \mathbf{v}^m \cdot \mathbf{x} = M\}$. There is no vector $\mathbf{w} \in \mathbb{R}^d$ for which $\mathbf{w} \cdot \mathbf{x}_+ - \mathbf{w} \cdot \mathbf{x}_- > 2M \mathbf{v}^m \cdot \mathbf{w}$ for all pairs $\mathbf{x}_+ \in C^+, \mathbf{x}_- \in C^-$.*

**Proof** Suppose such a vector exists. Define $\mathbf{w}' = \mathbf{w} - (\mathbf{v}^m \cdot \mathbf{w})\mathbf{v}^m$. By construction $\mathbf{w}' \in \mathrm{Null}(\mathbf{v}^m)$. For any pair $\mathbf{x}_+ \in C^+, \mathbf{x}_- \in C^-$ we have

$$\begin{aligned} \mathbf{w}' \cdot x_+ - \mathbf{w}' \cdot x_- &= \mathbf{w} \cdot x_+ - (\mathbf{v}^m \cdot \mathbf{w}) \mathbf{v}^m \cdot x_+ - \mathbf{w} \cdot x_- + (\mathbf{v}^m \cdot \mathbf{w}) \mathbf{v}^m \cdot x_- \\ &> \mathbf{v}^m \cdot \mathbf{w} (2M - \mathbf{v}^m \cdot x_+ + b^m - b^m + \mathbf{v}^m \cdot x_-) \\ &= 0. \end{aligned}$$

Define $c := \frac{1}{2}(\min\{\mathbf{w}' \cdot \mathbf{x}_+ \,|\, \mathbf{x}_+ \in C^+\} + \max\{\mathbf{w}' \cdot \mathbf{x}_- \,|\, \mathbf{x}_- \in C^-\})$. Then $\min\{\mathbf{w}' \cdot \mathbf{x}_+ - c | \mathbf{x}_+ \in C^+\} > 0$ and $\max\{\mathbf{w}' \cdot \mathbf{x}_- - c | \mathbf{x}_- \in C^-\} < 0$, a contradiction.

We are now in a position to provide the main proof of this appendix. The theorem states that if the maximum margin hyperplane is unique, and can be parameterised by $(\mathbf{v}^m, b^m) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}$, then

$$\lim_{h \to 0^+} \min\{\|(\mathbf{v}_h^\star, b_h^\star) - (\mathbf{v}^m, b^m)\|, \|(\mathbf{v}_h^\star, b_h^\star) + (\mathbf{v}^m, b^m)\|\} = 0,$$

where $\{H(v_h^\star, b_h^\star)\}_h$ is any collection of minimum density hyperplanes indexed by their bandwidth $h > 0$.

**Proof of Theorem 1** Define $M = \mathrm{margin}\, H(\mathbf{v}^m, b^m)$, $C^+ = \{\mathbf{x} \in \mathcal{X} \,|\, \mathbf{v}^m \cdot \mathbf{x} - b^m = M\}$ and $C^- = \{\mathbf{x} \in \mathcal{X} \,|\, b^m - \mathbf{v}^m \cdot \mathbf{x} = M\}$. Let $B = \max\{\|\mathbf{x}\| \,\big|\, \mathbf{x} \in \mathcal{X}\}$. Take any $\epsilon > 0$ and set $0 < \delta$ to satisfy $\frac{2\delta}{M}(1 + B^2) + 2B\delta^{3/2}\sqrt{\frac{2}{M}} + \delta^2 = \epsilon^2$. Now, suppose $(\mathbf{w}, c) \in \mathrm{bd}(\mathbb{B}^d) \times \mathbb{R}$ satisfies,

$$\mathbf{w} \cdot \mathbf{x}_+ - c > M - \delta, \ \forall \mathbf{x}_+ \in C^+ \text{ and } c - \mathbf{w} \cdot \mathbf{x}_- > M - \delta, \ \forall \mathbf{x}_- \in C^-.$$

By Lemma 9 we know that $\exists \mathbf{x}_+ \in C^+, \mathbf{x}_- \in C^-$ s.t. $\mathbf{w} \cdot \mathbf{x}_+ - \mathbf{w} \cdot \mathbf{x}_- \leqslant 2M \mathbf{v}^m \cdot \mathbf{w}$. Thus

$$\begin{aligned} \mathbf{v}^m \cdot \mathbf{w} &\geqslant \frac{\mathbf{w} \cdot \mathbf{x}_+ - \mathbf{w} \cdot \mathbf{x}_-}{2M} \\ &= \frac{\mathbf{w} \cdot \mathbf{x}_+ - c + c - \mathbf{w} \cdot \mathbf{x}_-}{2M} \\ &> \frac{2M - 2\delta}{2M} = 1 - \frac{\delta}{M}. \end{aligned}$$

Thus $\|\mathbf{v}^m-\mathbf{w}\|^2<\frac{2\delta}{M}$. Now, for each $\mathbf{x}_+\in C^+, \mathbf{v}^m\cdot\mathbf{x}_+-b=M$ and for each $\mathbf{x}_-\in C^-, b-\mathbf{v}^m\cdot\mathbf{x}_-=M$. Thus for any such $\mathbf{x}_+,\mathbf{x}_-$ we have,

$$\begin{gathered}
M-\delta+\mathbf{w}\cdot\mathbf{x}_-<c<\mathbf{w}\cdot\mathbf{x}_+-M+\delta,\\
b^m-\mathbf{v}^m\cdot\mathbf{x}_--\delta+\mathbf{w}\cdot\mathbf{x}_-<c<\mathbf{w}\cdot\mathbf{x}_+-\mathbf{v}^m\cdot\mathbf{x}_++b^m+\delta,\\
b^m-\delta-(\mathbf{v}^m-\mathbf{w})\cdot\mathbf{x}_-<c<b^m+\delta+(\mathbf{w}-\mathbf{v}^m)\cdot\mathbf{x}_+,\\
b^m-\delta-B\|\mathbf{v}^m-\mathbf{w}\|<c<b^m+\delta+B\|\mathbf{w}-\mathbf{v}^m\|,\\
|c-b^m|<|\delta+B\|\mathbf{w}-\mathbf{v}^m\||.
\end{gathered}$$

We can now bound the distance between $(\mathbf{w},c)$ and $(\mathbf{v}^m,b^m)$,

$$\begin{aligned}
\|(\mathbf{v}^m,b^m)-(\mathbf{w},c)\|^2 &= \|\mathbf{v}^m-\mathbf{w}\|^2+|b^m-c|^2\\
&< \|\mathbf{v}^m-\mathbf{w}\|^2(1+B^2)+2B\delta\|\mathbf{v}^m-\mathbf{w}\|+\delta^2\\
&< \frac{2\delta}{M}(1+B^2)+2B\delta\sqrt{\frac{2\delta}{M}}+\delta^2\\
&= \epsilon^2.
\end{aligned}$$

We have shown that for any hyperplane $H(\mathbf{w},c)$ that achieves a margin larger than $M-\delta$ on the support points of the maximum margin hyperplane, $\mathbf{x}\in C^+\cup C^-$, the distance between $(\mathbf{w},c)$ and $(\mathbf{v}^m,b^m)$ is less than $\epsilon$. Equivalently, any hyperplane $H(\mathbf{w},c)$ such that $\|(\mathbf{w},c)-(\mathbf{v}^m,b^m)\|>\epsilon$ has a margin less than $M-\delta$, as $\min\{|\mathbf{w}\cdot\mathbf{x}-c|\,|\,\mathbf{x}\in C^+\cup C_-\}<M-\delta$. By symmetry, the same holds for any $(\mathbf{w},c)$ within distance $\epsilon$ of $(-\mathbf{v}^m,-b^m)$.

By Lemma 4 $\exists h_1>0$ such that for all $h\in(0,h_1)$, the minimum density hyperplane for $h$, $H(\mathbf{v}_h^\star,b_h^\star)$, induces the same partition of $\mathcal{X}$ as the maximum margin hyperplane, $H(\mathbf{v}^m,b^m)$. By Lemma 3 $\exists h_2>0$ such that for all $h\in(0,h_2)$, $\operatorname{margin}H(\mathbf{v}_h^\star,b_h^\star)>M-\delta$. Therefore for $h\in(0,\min\{h_1,h_2\})$, we have $\min\{\|(\mathbf{v}_h^\star,b_h^\star)-(\mathbf{v}^m,b^m)\|,\|(\mathbf{v}_h^\star,b_h^\star)+(\mathbf{v}^m,b^m)\|\}<\epsilon$. Since $\epsilon>0$ was arbitrarily chosen, this gives the result. ■

# Chapter 3

# Projection Pursuit Based on Spectral Connectivity

This chapter contains two parts. The first part provides a rigorous investigation into projection pursuit based on spectral connectivity for the purpose of unsupervised data partitioning. The spectral connectivity of a data set refers to the optimal value of a relaxation of the normalised graph cut, in terms of a graph defined over the data set in which edges correspond to the similarities between data. A projection along which a data set has low spectral connectivity in general corresponds to a projection along which the data display multiple, well separated clusters. In this context a cluster is a subgraph containing relatively high edge weights, and hence mostly similar data points.

The second part extends the methodology developed in the first part to the problem of semi-supervised classification. The graph formulation provides a useful framework for incorporating additional information, such as the known class labels of some of the data used in semi-supervised classification. If the cluster assumption holds, then determining subspaces which provide a good separation of clusters of data, and which also allows a separation of the known classes is likely to provide high quality classifications of the data whose classes are unknown.

# A. Minimum Spectral Connectivity Projection Pursuit for Unsupervised Classification


## Abstract

*We study the problem of determining the optimal low dimensional projection for maximising the separability of a binary partition of an unlabelled dataset, as measured by spectral graph theory. This is achieved by finding projections which minimise the second eigenvalue of the Laplacian matrices of the projected data, which corresponds to a non-convex, non-smooth optimisation problem. We show that the optimal univariate projection based on spectral connectivity converges to the vector normal to the maximum margin hyperplane through the data, as the scaling parameter is reduced to zero. This establishes a connection between connectivity as measured by spectral graph theory and maximal Euclidean separation. It also allows us to apply our methodology to the problem of finding large margin linear separators. The computational cost associated with each eigen-problem is quadratic in the number of data. To mitigate this problem, we propose an approximation method using microclusters with provable approximation error bounds. We evaluate the performance of the proposed method on a large collection of benchmark datasets and find that it compares favourably with existing methods for projection pursuit and dimension reduction for unsupervised data partitioning.*


## 1 Introduction

The classification of unlabelled data is fundamental to many statistical and machine learning applications. Such applications arise in the context of clustering and semi-supervised classification. Underpinning these tasks is the

assumption of a clusterable structure within the data, and importantly that this structure is relevant to the classification task. The assumption of a clusterable structure, however, begs the question of how a cluster should be defined. Centroid based methods, such as the ubiquitous $k$-means algorithm, define clusters in reference to single points, or centers (Leisch, 2006). In the non-parametric statistical approach to clustering, clusters are associated with the modes of a probability density function from which the data are assumed to arise (Hartigan, 1975, Chapter 11). We consider the definition as given in the context of graph partitioning, and the relaxation given by spectral clustering. Spectral clustering has gained considerable interest in recent years due to its strong performance in diverse application areas. In this context clusters are defined as strongly connected components of a graph defined over the data, wherein vertices correspond to data points and edge weights represent pairwise similarities (von Luxburg, 2007).

The minimum cut graph problem seeks to partition a graph such that the sum of the edges connecting different components of the partition is minimised. To avoid partitions containing small sets of vertices, a normalisation is introduced which helps to emphasise more balanced partitions. The normalisation, however, makes the problem NP-hard (Wagner and Wagner, 1993), and so a continuous relaxation is solved instead. The relaxed problem, known as spectral clustering, is solved by the eigenvectors of the *graph Laplacian* matrices. We give a brief introduction to spectral clusering in Section 3.

Crucial to all cluster definitions is the relevance of spatial similarity of points. In multivariate data analysis, however, the presence of irrelevant or noisy features can significantly obscure the spatial structure in a data set. Moreover, in very high dimensional applications the curse of dimensionality can make spatial similarities unreliable for distinguishing clusters (Steinbach et al., 2004; Beyer et al., 1999). Dimension reduction techniques seek to mitigate the effect of irrelevant features and of the curse of dimensionality by finding low dimensional representations of a set of data which retain as much information as possible. Most commonly these low dimensional representations are defined by the projection of the data into a linear subspace. Information retention is crucial for the success of any subsequent tasks. For unsupervised classification this information must, therefore, be relevant in the context of cluster structure. Classical dimension reduction techniques such as principal component analysis (PCA) cannot guarantee the structural relevance of the low dimensional subspace. Moreover a single subspace may not suffice to distinguish all clusters, which may have their structures defined within differing subspaces. Recently a number of dimension reduction methods with an explicit objective which is relevant to cluster structure have been proposed (Krause and Liebscher, 2005; Niu et al., 2011; Pavlidis et al., 2015). We discuss these briefly in Section 2.

We consider the problem of learning the optimal subspace for the purpose

of data bi-partitioning, where optimality is measured by the connectivity of the projected data, as defined in spectral graph theory. We formulate the problem in the context of *projection pursuit*; a class of optimisation problems which aim to find *interesting* subspaces within potentially high dimensional data sets, where interestingness is captured by a predefined objective, called the *projection index*. With very few exceptions, the optimisation of the projection index does not admit a closed form solution, and is instead numerically optimised. The projection indices considered in the proposed method are the second smallest eigenvalues of the graph Laplacians, which measure the quality of a binary partition arising from the normalised minimum cut graph problem. These eigenvalues are non-smooth and non-convex, and so specialised techniques are required to optimise them. We establish conditions under which they are Lipschitz and almost everywhere continuously differentiable, and discuss how to find local optima with guaranteed convergence properties.

In this paper we establish an asymptotic connection between optimal univariate subspaces for bi-partitioning based on spectral graph theory, and maximum margin hyperplanes. Formally, we show that as the scaling parameter defining pairwise similarities is reduced to zero, the optimal univariate subspace for bi-partitioning converges to the subspace normal to the largest margin hyperplane through the data. This establishes a theoretical connection between connectivity as measured by spectral graph theory and maximal Euclidean separation. It also provides an alternative methodology for learning maximum margin clustering models, which have attracted considerable interest in recent years (Xu et al., 2004; Zhang et al., 2009). We introduce a way of modifying the similarity function which avoids focusing on outliers, and allows us to further control the balance of the induced partition. The importance of controlling this balance has been observed in the context of large margin clustering (Xu et al., 2004; Zhang et al., 2009) and low density separators (Pavlidis et al., 2015).

The computation cost associated with the eigen-problem underlying our projection index is quadratic in the number of data. To mitigate this computational burden we propose a data preprocessing step using micro-clusters which significantly speeds up the optimisation. We establish theoretical error bounds for this approximation method, and provide a sensitivity study which shows no degradation in clustering performance, even for a coarse approximation.

The remainder of the paper is organised as follows. In Section 2 we briefly discuss related work on dimension reduction for unsupervised data partitioning. A brief outline of spectral clustering is provided in Section 3. Section 4 presents the methodology for finding optimal projections to perform binary partitions. Section 5 describes the theoretical connection between optimal subspaces for spectral bi-partitioning and maximum margin hyperplanes. In

Section 6 we discuss an approximation method in which the computational speed associated with finding the optimal subspace can be significantly improved, with provable approximation error bounds. Experimental results and sensitivity analyses are presented in Section 7, while Section 8 is devoted to concluding remarks.

## 2 Related Work

The literature on clustering high dimensional data is vast, and we will focus only on methods with an explicit dimension reduction formulation, as in projection pursuit. Implicit dimension reduction methods based on learning sparse covariance matrices (which impose an implicit low dimensional projection of the data/clusters), such as quadratic discriminant analysis, can be limited by the assumption that clusters are determined by their covariance matrices. Projection pursuit approaches can be made more versatile by defining objectives which admit more general cluster definitions.

Principal component analysis and independent component analysis have been used in the context of clustering, however their objectives do not correspond exactly with those of the clustering task and the justification of their use is based more on common-sense reasoning. Nonetheless, these methods have shown good empirical performance on a number of applications (Boley, 1998; Tasoulis et al., 2010; Kriegel et al., 2009). Some recent approaches to projection pursuit for clustering rely on the non-parametric statistical notion clusters, i.e., that clusters are regions of high density in a probability distribution from which the data are assumed to have arisen. Krause and Liebscher (2005) proposed using as projection index the *dip statistic* (Hartigan and Hartigan, 1985) of the projected data. The dip is a measure of departure from unimodality, and so maximising the dip tends to projections which have mutlimodal marginal density, and therefore separate high density clusters. The authors establish that the dip is differentiable for any projection vector onto which the projected data are unique, and use a simple gradient ascent method to find local optima.

The minimum density hyperplane approach (Pavlidis et al., 2015) is posed as a projection pursuit for the univariate subspace normal to the hyperplane with minimal integrated density along it, thereby establishing regions of low density which separate the modes of the underlying probability density. The projection index in this case is the minimum of the kernel density estimate of the projected data, penalised to avoid hyperplanes which do not usefully split the data. The authors show an asymptotic connection between the hyperplane with minimal integrated density and the maximum margin hyperplane. The result we show in Section 5 therefore establishes that the optimal subspace for bi-partitioning based on spectral connectivity is asymptotically

connected with the minimum integrated density hyperplane.

A number of direct approaches to maximum margin clustering have also been proposed (Xu et al., 2004; Zhang et al., 2009). These can be viewed as a projection pursuit for the subspace normal to the maximum margin hyperplane intersecting the data. The iterative support vector regression approach (Zhang et al., 2009) uses support vector methods and so for the linear kernel explicitly learns the corresponding projection vector, $v$.

Most similar to our work is that of Niu et al. (2011), who also proposed a method for dimension reduction based on spectral clustering. The authors show an interesting connection between optimal subspaces for spectral clustering and *sufficient dimension reduction*. For the case of a binary partition, their objective is equivalent to one of the objectives we consider, i.e., that of minimising the second smallest eigenvalue of the normalised Laplacian (cf. Sections 3 and 4). However, our methodology differs substantially from theirs. Niu et al. (2011) define their objective by

$$\max_{U,W} \quad \operatorname{trace}(U^\top D^{-1/2} A D^{-1/2} U) \tag{3.1a}$$

$$s.t. \quad U^\top U = I \tag{3.1b}$$

$$A_{ij} = s(\|W^\top x_i - W^\top x_j\|) \tag{3.1c}$$

$$W^\top W = I. \tag{3.1d}$$

The matrix $A$ is the affinity matrix containing pairwise similarities of points projected into the subspace defined by $W$, and $D$ is the diagonal degree matrix of $A$, with $i$-th diagonal element equal to the $i$-th row sum of $A$. Further details of these objects can be found in Section 3. The approach used by the authors to maximise this objective alternates between using spectral clustering to determine the columns of $U$, and then using a gradient ascent method to maximise $\operatorname{trace}(U^\top D^{-1/2} A D^{-1/2} U)$ over $W$, where the dependence of this objective on the projection matrix $W$ is through Eq. (3.1c). Within this gradient ascent step the matrices $U$ and $D$ are kept fixed. This process is iterated until convergence. However, the authors do not address the fact that the matrix $D$ is determined by $A$, and therefore depends on the projection matrix $W$. An ascent direction for the objective assuming a fixed $D$ is therefore not necessarily an ascent direction for the overall objective. Despite this fact the method has shown good empirical performance on a number of problems (Niu et al., 2011). In Section 4 we derive expressions for the gradient of the overall objective, which allows us to optimise it directly.

# 3 Background on Spectral Clustering

In this section we provide a brief introduction to spectral clustering, with particular attention to bi-partitioning, which underlies the focus of this work. Bi-

partitioning using spectral clustering has been considered previously by Shi and Malik (2000), where a full clustering can be obtained by recursively inducing bi-partitions of (subsets of) the data. With a data sample, $X = \{x_1,...,x_N\}$, spectral clustering associates a graph $G=(V,E)$, in which vertices correspond to observations, and the *undirected* edges assume weights equal to the pairwise *similarity* between observations. Pairwise similarities can be determined in a number of ways, including nearest neighbours and similarity metrics. In general, similarities are determined by the spatial relationships between points, and pairs which are closer are assigned higher similarity than those which are more distant.

The information in $G$ can be represented by the *adjacency*, or affinity matrix, $A \in \mathbb{R}^{N\times N}$, with $A_{ij} = E_{ij}$. The *degree* of each vertex $v_i$ is defined as, $d_i = \sum_{j=1}^{N} A_{ij}$. The *degree matrix*, $D$, is then defined as the diagonal matrix with $i$-th diagonal element equal to $d_i$. For a subset $C \subset X$, the size of $C$ can be defined either by the cardinality of $C$, $|C|$, or by the *volume* of $C$, $\text{vol}(C) = \sum_{i:x_i \in C} d_i$.

**Definition** The *normalised min-cut graph problem* for a binary partition is defined as the optimisation problem

$$\min_{C\subset X} \sum_{i,j:x_i\in C, x_j \in X\setminus C} A_{ij}\left(\frac{1}{\text{size}(C)} + \frac{1}{\text{size}(X\setminus C)}\right). \tag{3.2}$$

It has been shown (Hagen and Kahng, 1992; Shi and Malik, 2000) that the two normalised min-cut graph problems (corresponding to the two definitions of size) can be formulated in terms of the *graph Laplacian* matrices,

$$(standard)\ L = D - A, \tag{3.3}$$

$$(normalised)\ L_{\text{norm}} = D^{-1/2} L D^{-1/2}, \tag{3.4}$$

as follows. For $C \subset X$ define $u^C \in \mathbb{R}^N$ to be the vector with $i$-th entry,

$$u_i^C = \begin{cases} \sqrt{\text{size}(X\setminus C)/\text{size}(C)}, & \text{if } x_i \in C \\ -\sqrt{\text{size}(C)/\text{size}(X\setminus C)}, & \text{if } x_i \in X\setminus C. \end{cases} \tag{3.5}$$

For $\text{size}(C) = |C|$, the optimisation problem in (3.2) can be written as,

$$\min_{C\subset X} (u^C)^\top L u^C \text{ s.t. } u^C \perp \mathbf{1},\ \|u^C\| = \sqrt{N}. \tag{3.6}$$

Similarly, if $\text{size}(C) = \text{vol}(C)$ the problem in (3.2) is equivalent to,

$$\min_{C\subset X} (u^C)^\top L u^C \text{ s.t. } D u^C \perp \mathbf{1},\ (u^C)^\top D u^C = \text{vol}(V). \tag{3.7}$$

Both problems in (3.6) and (3.7) are NP-hard (Wagner and Wagner, 1993), and so continuous relaxations of these, in which the discreteness condition

on $u^C$ given in Eq. (3.5) is removed, are solved instead (Hagen and Kahng, 1992; Shi and Malik, 2000). The solutions to the relaxed problems are given by the second eigenvector of $L$ and the second eigenvector of the generalised eigen equation $Lu = \lambda Du$ respectively, the latter thus equivalently solved by $D^{-1/2}u$, where $u$ is the second eigenvector of $L_{\text{norm}}$. In particular, we have

$$\lambda_2(L) \leq \frac{1}{N}(u^S)^\top L u^S \tag{3.8}$$

$$\lambda_2(L_{\text{norm}}) \leq \frac{1}{\text{vol}(V)}(u^N)^\top L u^N, \tag{3.9}$$

where $\lambda_2(L)$ and $\lambda_2(L_{\text{norm}})$ are the second eigenvalues of $L$ and $L_{\text{norm}}$ and $u^S$ and $u^N$ are the solutions to (3.6) and (3.7) respectively.

The following properties of the matrices $L$ and $L_{\text{norm}}$ will be useful in establishing our proposed methodology and the associated theoretical results. These properties can be found in (von Luxburg, 2007, Propositions 2 and 3).

1. For any $v \in \mathbb{R}^N$ we have

$$v^\top L v = \frac{1}{2}\sum_{i,j} A_{ij}(v_i - v_j)^2 \tag{3.10}$$

$$v^\top L_{\text{norm}} v = \frac{1}{2}\sum_{i,j} A_{ij}\left(\frac{v_i}{\sqrt{d_i}} - \frac{v_j}{\sqrt{d_j}}\right)^2. \tag{3.11}$$

2. $L$ and $L_{\text{norm}}$ are symmetric and positive semi-definite.

3. The smallest eigenvalue of $L$ is 0 with corresponding eigenvector $\mathbf{1}$, the constant 1 vector

4. The smallest eigenvalue of $L_{\text{norm}}$ is 0 with corresponding eigenvector $D^{1/2}\mathbf{1}$.

The extension of clustering via the normalised min-cut to a $K$-partition of the data is similar, and can be solved approximately by the first $K$ eigenvectors of either $L$ or $L_{\text{norm}}$ (von Luxburg, 2007).

# 4 Projection Pursuit for Spectral Connectivity

In this section we study the problem of minimising the second eigenvalue of the graph Laplacian matrices of the projected data. If the projected data are split in two through spectral clustering, then the projection that minimises the second eigenvalue of the corresponding graph Laplacian minimises the connectivity of the two components, as measured by spectral graph theory.

Note that while we discuss explicitly the minimisation of the second eigenvalue, the methodology we present in fact applies to an arbitrary eigenvalue of the graph Laplacians. As a result, the method discussed herein trivially extends to the problem of determining a $K$-partition by minimising the sum of the $K$ smallest eigenvalues of the Laplacians.

To begin with, let $X=\{x_1,...,x_N\}$ be a $d$-dimensional data set and let $V\in\mathbb{R}^{d\times l}$ be a *projection matrix*, where $l\in\mathbb{N}$ is the dimension of the projection, and the columns of $V$, $\{V_1,...,V_l\}$, have unit norm. With this formulation it is convenient to consider a parameterisation of $V$ through polar coorindates as follows. Let $\Theta=[0,\pi)^{(d-1)\times l}$ and for $\boldsymbol{\theta}\in\Theta$, the projection matrix $V(\boldsymbol{\theta})\in\mathbb{R}^{d\times l}$ is given by,

$$V(\boldsymbol{\theta})_{ij}=\begin{cases}\cos(\boldsymbol{\theta}_{ij})\prod_{k=1}^{i-1}\sin(\boldsymbol{\theta}_{kj}), & i=1,...,d-1\\ \prod_{k=1}^{d-1}\sin(\boldsymbol{\theta}_{kj}), & i=d.\end{cases} \tag{3.12}$$

From this we define the $l$ dimensional *projected data set* by $P(\boldsymbol{\theta})=\{p(\boldsymbol{\theta})_1,...,p(\boldsymbol{\theta})_N\}=\{V(\boldsymbol{\theta})^\top x_1,...,V(\boldsymbol{\theta})^\top x_N\}$, and we let $L(\boldsymbol{\theta})$ (resp. $L_{\text{norm}}(\boldsymbol{\theta})$) be the Laplacian (resp. normalised Laplacian) of the graph of $P(\boldsymbol{\theta})$. Edge weights are determined by a positive function $s:(\mathbb{R}^l)^N\times\{1\dots N\}^2\to\mathbb{R}^+$, in that the affinity matrix is given by $A(\boldsymbol{\theta})_{ij}:=s(P(\boldsymbol{\theta}),i,j)$. In the simplest case we may imagine $s$ being fully determined by the Euclidean distance between two elements of the projected data, i.e., $s(P(\boldsymbol{\theta}),i,j)=k(\|p(\boldsymbol{\theta})_i-p(\boldsymbol{\theta})_j\|)$, for some function $k:\mathbb{R}\to\mathbb{R}^+$. However we prefer to allow for a more general definition. We discuss this further in Section 4.3.

Henceforth we will use $\lambda_i(\cdot)$ to be the $i$-th (smallest) eigenvalue of its (in all cases herein) real symmetric matrix argument, and we assume that all eigenvectors are unit-norm. The objectives $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ are, in general, non-convex and non-smooth in $\boldsymbol{\theta}$, and so specialised techniques are required to optimise them. In the following subsections we investigate their differentiability properties, and discuss how alternating between a naive gradient descent method and a descent step based on a directional derivative can be used to find locally optimal solutions.

## 4.1 Continuity and Differentiability

In this subsection we explore the continuity and differentiability properties of the second eigenvalue of the graph Laplacians, viewed as a function of the *projection angle*, $\boldsymbol{\theta}$. We will view the data set $X$ as a $d\times N$ matrix with $i$-th column equal to $x_i$, and similarly the projected data set as an $l\times N$ matrix, $P(\boldsymbol{\theta})=V(\boldsymbol{\theta})^\top X$, with $i$-th column $p(\boldsymbol{\theta})_i$.

**Lemma 10** *Let $X=\{x_1,...,x_N\}\subset\mathbb{R}^d$ and let $s(P,i,j)$ be Lipschitz continuous in $P\in\mathbb{R}^{l\times N}$ for fixed $i,j\in\{1\dots N\}$. Then $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ are Lipschitz continuous in $\boldsymbol{\theta}$.*

**Proof** We show the case of $L(\boldsymbol{\theta})$, where that of $L_{\text{norm}}(\boldsymbol{\theta})$ is similar. The result follows from the fact that $L(\boldsymbol{\theta})$ is element-wise Lipschitz as a composition of Lipschitz functions ($V(\boldsymbol{\theta})$ is Lipschitz in $\boldsymbol{\theta}$ as a collection of products of Lipschitz functions) and the fact that

$$|\lambda_i(L(\boldsymbol{\theta})) - \lambda_i(L(\boldsymbol{\theta}'))| \leq \|L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}')\| \leq N\sqrt{\max_{ij}|L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}')|_{ij}},$$

where the first inequality is due to Weyl (1912), and the second comes from Schur's inequality (Schur, 1911).

Rademacher's theorem therefore establishes that both objectives are almost everywhere differentiable (Polak, 1987). This almost everywhere differentiability can also be seen by considering that simple eigenvalues of real symmetric matrices are differentiable, e.g. Magnus (1985), and establishing that under certain conditions on the function $s$ the eigenvalues of $L(\boldsymbol{\theta})$ and $L_{\text{norm}}(\boldsymbol{\theta})$ are simple for almost all $\boldsymbol{\theta}$.

Tao and Vu (2014) have shown that the real symmetric matrices with non-simple spectrum lie in a subspace of co-dimension 2. If we denote the space of real valued $N \times N$ symmetric matrices by $\mathcal{S}_N$, and denote this subspace by $S$, then $\mathcal{S}_N \setminus S$ is open and dense in $\mathcal{S}_N$. Sufficient conditions on the function $s$ for the almost everywhere simplicity of $\lambda_2(L(\boldsymbol{\theta}))$ (resp. $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$) are therefore that it is continuous in $P$ for each $i, j$ and for all $B \in \mathcal{S}_N$ and $U$ open in $\Theta$, $\exists \boldsymbol{\theta} \in U$ s.t. $\text{trace}(L(\boldsymbol{\theta})B) \neq 0$ (resp. $\text{trace}(L_{\text{norm}}(\boldsymbol{\theta})B) \neq 0$). Continuity of $s$ ensures continuity of the functions $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$, and therefore the openness of the preimage of $\mathcal{S}_N \setminus S$. The latter condition ensures that for each open $U \subset \Theta$, the span of the image of $U$ under $\lambda_2(\cdot)$ is $\mathcal{S}_N$. Therefore, in every open $U \subset \Theta, \exists \boldsymbol{\theta} \in U$ s.t. $L(\boldsymbol{\theta}) \notin S$. Therefore the pre-image of $\mathcal{S}_N \setminus S$ is dense in $\Theta$.

Generalised gradient based optimisation methods are the natural framework for finding the optimal subspace for spectral bi-partitioning. Eigenvalue optimisation is, in general, a challenging problem due to the fact that eigenvalues are not differentiable where they coincide. The majority of approaches in the literature focus on the problems of minimising the largest eigenvalue or the sum of a predetermined number of largest eigenvalues (Overton and Womersley, 1993). Both of these problems tend to lead to a coalescence of eigenvalues, making the issue of non-differentiability especially problematic. Conversely the minimisation of the smallest eigenvalue tends to lead to a separation of eigenvalues, and so non-differentiability is less of a concern (Lewis and Overton, 1996).

If the similarity function $s$ is strictly positive, then both $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ are bounded away from zero, and hence minimising these has the same benefits as does minimising the smallest eigenvalue in general,

in that the corresponding optimisation tends to separate them from other eigenvalues. Despite this practical advantage, the simplicity of $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ is not guaranteed over the entire optimisation. We discuss a way of handling points of non-differentiability in Section 4.2. This approach uses the directional derivative formulation given by Overton and Womersley (1993), and allows us to find descent directions which also tend to lead to a decoupling of eigenvalues.

Global convergence of gradient based optimisation algorithms relies on the continuity of the derivatives (where they exist). To establish this continuity, we first derive expressions for the derivatives of $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ as functions of $\boldsymbol{\theta}$. Theorem 1 of Magnus (1985) provides a useful formulation of eigenvalue derivatives. If $\lambda$ is a simple eigenvalue of a real symmetric matrix $M$, then $\lambda$ is infinitely differentiable on a neighbourhood of $M$, and the differential at $M$ is given by (Magnus, 1985),

$$d\lambda = u^\top d(M) u, \tag{3.13}$$

where $u$ is the corresponding eigenvector. Let us assume that $s(P,i,j)$ is differentiable in $P \in \mathbb{R}^{l\times N}$ for fixed $i,j \in \{1\ldots N\}$. For brevity we temporarily drop the notational dependence on $\boldsymbol{\theta}$ and denote the second eigenvalue of the Laplacian by $\lambda$, and the corresponding eigenvector by $u$. The derivative $D_{\boldsymbol{\theta}}\lambda$ is given by the $(d-1)\times l$ matrix with $i$-th column $D_{\boldsymbol{\theta}_i}\lambda$, where we consider the chain rule decomposition $D_{\boldsymbol{\theta}_i}\lambda = D_P \lambda D_V P D_{\boldsymbol{\theta}_i} V$. Here $D_\cdot$ is the differential operator. Since only the $i$-th column of $V$ depends on $\boldsymbol{\theta}_i$, and only the $i$-th row of $P$ depends on $V_i$, this product can be simplified as $D_{\boldsymbol{\theta}_i}\lambda = D_{P_i}\lambda D_{V_i} P_i D_{\boldsymbol{\theta}_i} V_i$, where $P_i$ is used to denote the $i$-th row of $P$, while $V_i$ and $\boldsymbol{\theta}_i$ are, as usual, the $i$-th columns of $V$ and $\boldsymbol{\theta}$ respectively. We provide expressions for each of these terms below.

We first consider the standard Laplacian $L$. By Eq. (3.13) we have $d\lambda = u^\top d(L)u = u^\top d(D)u - u^\top d(A)u$. Now,

$$\frac{\partial D_{ii}}{\partial P_{mn}} = \sum_{j=1}^{N} \frac{\partial A_{ij}}{\partial P_{mn}} = \sum_{j=1}^{N} \frac{\partial s(P,i,j)}{\partial P_{mn}}, \text{ and } \frac{\partial A_{ij}}{\partial P_{mn}} = \frac{\partial s(P,i,j)}{\partial P_{mn}}, \tag{3.14}$$

and so,

$$\frac{\partial \lambda}{\partial P_{mn}} = u^\top \frac{\partial L}{\partial P_{mn}} u = \frac{1}{2}\sum_{i,j}(u_i - u_j)^2 \frac{\partial s(P,i,j)}{\partial P_{mn}}. \tag{3.15}$$

For the normalised Laplacian, $L_{\text{norm}}$, consider first

$$\begin{aligned} d(L_{\text{norm}}) &= d(D^{-1/2} L D^{-1/2}) \\ &= d(D^{-1/2}) L D^{-1/2} + D^{-1/2} d(D) D^{-1/2} - D^{-1/2} d(A) D^{-1/2} \\ &\quad + D^{-1/2} L d(D^{-1/2}). \end{aligned}$$

We will again use $\lambda$ and $u$ to denote the second eigenvalue and corresponding eigenvector. Using the fact that $LD^{-1/2}u=\lambda D^{1/2}u$,

$$\begin{aligned} d\lambda &= u^\top d(D^{-1/2})LD^{-1/2}u+u^\top D^{-1/2}d(D)D^{-1/2}u-u^\top D^{-1/2}d(A)D^{-1/2}u \\ &\quad +u^\top D^{-1/2}Ld(D^{-1/2})u \\ &= \lambda u^\top d(D^{-1/2})D^{1/2}u+u^\top D^{-1/2}d(D)D^{-1/2}u-u^\top D^{-1/2}d(A)D^{-1/2}u \\ &\quad +\lambda u^\top D^{1/2}d(D^{-1/2})u \\ &= \lambda u^\top d(I)u+(1-\lambda)u^\top D^{-1/2}d(D)D^{-1/2}u-u^\top D^{-1/2}d(A)D^{-1/2}u, \end{aligned}$$

since

$$\begin{aligned} d(D^{-1/2})D^{1/2}+D^{1/2}d(D^{-1/2}) &= d(D^{-1/2})DD^{-1/2}+D^{-1/2}Dd(D^{-1/2}) \\ &= d(D^{-1/2})DD^{-1/2}+D^{-1/2}d(D)D^{-1/2}+D^{-1/2}Dd(D^{-1/2}) \\ &\quad -D^{-1/2}d(D)D^{-1/2} \\ &= d(I)-D^{-1/2}d(D)D^{-1/2}. \end{aligned}$$

We therefore have,

$$\begin{aligned} d\lambda &= (1-\lambda)u^\top D^{-1/2}d(D)D^{-1/2}u-u^\top D^{-1/2}d(A)D^{-1/2}u \\ &= u^\top D^{-1/2}d(L)D^{-1/2}u-\lambda u^\top D^{-1/2}d(D)D^{-1/2}u. \end{aligned}$$

Therefore,

$$\frac{\partial\lambda}{\partial P_{mn}}=\frac{1}{2}\sum_{i,j}\left(\frac{u_i}{\sqrt{d_i}}-\frac{u_j}{\sqrt{d_j}}\right)^2\frac{\partial s(P,i,j)}{\partial P_{mn}}-\lambda\sum_{i,j}\frac{u_i^2}{d_i}\frac{\partial s(P,i,j)}{\partial P_{mn}}. \tag{3.16}$$

The component $D_{V_i}P_i$ is simply the $N\times d$ transposed data matrix, and the $d\times(d-1)$ matrix, $D_{\boldsymbol{\theta}_i}V_i$, is given by

$$\begin{bmatrix} -\sin(\boldsymbol{\theta}_{1i}) & 0 & \dots & 0 \\ \cos(\boldsymbol{\theta}_{1i})\cos(\boldsymbol{\theta}_{2i}) & -\sin(\boldsymbol{\theta}_{1i})\sin(\boldsymbol{\theta}_{2i}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\boldsymbol{\theta}_{1i})\prod_{k=2}^{d-1}\sin(\boldsymbol{\theta}_{ki}) & \cos(\boldsymbol{\theta}_{2i})\prod_{k\neq 2}\sin(\boldsymbol{\theta}_{ki}) & \dots & \cos(\boldsymbol{\theta}_{d-1,i})\prod_{k=1}^{d-2}\sin(\boldsymbol{\theta}_{ki}) \end{bmatrix}. \tag{3.17}$$

Having derived expressions for the derivatives of $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$, we can address their continuity properties. The components $D_V PD_{\boldsymbol{\theta}}V$ clearly form a continuous product in $\boldsymbol{\theta}$. The continuity of the elements $\partial\lambda/\partial P_{mn}$ can be reduced to addressing the continuity of the eigenvalue itself, of its associated eigenvector and a continuity assumption on the derivative of the function $s$. It is well known that the eigenvalues of a matrix are continuous (Zedek, 1965), while the continuity of the elements of the eigenvector

come from the fact that we have assumed $\lambda$ to be simple (Magnus, 1985). We provide full expressions for the derivatives of $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$, for the similarity function used in our experiments, in the Appendix.

The eigenvalues of a real, symmetric matrix can be expressed as the difference between two convex matrix functions (Fan, 1949). If the similarity function, $s$, is Lipschitz continuous and differentiable we therefore have that $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ are directionally differentiable everywhere. Overton and Womersley (1993) describe a way of expressing the directional derivative of the sum of the $k$ largest eigenvalues of a matrix whose elements are continuous functions of a parameter, at a point of non-simplicity of the $k$-th largest eigenvalue. We will discuss the case of $\lambda_2(L(\boldsymbol{\theta}))$, where $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ is analogous. If we denote the sum of the $k$ largest eigenvalues of $L(\boldsymbol{\theta})$ by $F^k(\boldsymbol{\theta})$ then,

$$\lambda_2(L(\boldsymbol{\theta})) = F^{N-1}(\boldsymbol{\theta}) - F^{N-2}(\boldsymbol{\theta}). \tag{3.18}$$

Now suppose that $\boldsymbol{\theta}$ is such that

$$\begin{aligned}
&\lambda_N(L(\boldsymbol{\theta})) \geq \ldots \geq \lambda_{N-r+1}(L(\boldsymbol{\theta})) > \\
&\lambda_{N-r}(L(\boldsymbol{\theta})) = \ldots = \lambda_{N-k+1}(L(\boldsymbol{\theta})) = \ldots = \lambda_{N-r-t+1}(L(\boldsymbol{\theta})) > \\
&\lambda_{N-r-t}(L(\boldsymbol{\theta})) \geq \ldots \geq \lambda_1(L(\boldsymbol{\theta})).
\end{aligned}$$

That is, the $k$-th largest eigenvalue has multiplicity $t$ and $k-r$ are included in the sum defining $F^k(\boldsymbol{\theta})$. Then the directional derivative of $F^k(\boldsymbol{\theta})$ in direction $\theta$ is given by (Overton and Womersley, 1993)

$$F^{k\prime}(\boldsymbol{\theta};\theta) = \sum_{i=1}^{d-1}\sum_{j=1}^{l}\theta_{ij}\text{trace}(R^\top L_{ij} R) + \max_{U\in\Phi_{t,k-r}}\sum_{i=1}^{d-1}\sum_{j=1}^{l}\theta_{ij}\text{trace}(Q^\top L_{ij} Q U), \tag{3.19}$$

where $L_{ij} = \partial L(\boldsymbol{\theta})/\partial\boldsymbol{\theta}_{ij}$, the matrix $R \in \mathbb{R}^{N\times r}$ has $j$-th column equal to the eigenvector of the $j$-th largest eigenvalue of $L(\boldsymbol{\theta})$ and the matrix $Q \in \mathbb{R}^{N\times t}$ has $j$-th column equal to the eigenvector of the $(r+j)$-th largest eigenvalue of $L(\boldsymbol{\theta})$. In addition the set $\Phi_{a,b}$ is defined as,

$$\Phi_{a,b} := \{U \in \mathcal{S}_a | U \text{ and } I-U \text{ are positive semi-definite and } \text{trace}(U) = b\}. \tag{3.20}$$

Overton and Womersley (1993) have shown that $F^{k\prime}(\boldsymbol{\theta};\theta)$ is the sum of the eigenvalues of $\sum_{i=1}^{d-1}\sum_{j=1}^{l}\theta_{ij}R^\top L_{ij}R$ plus the sum of the $k-r$ largest eigenvalues of $\sum_{i=1}^{d-1}\sum_{j=1}^{l}\theta_{ij}Q^\top L_{ij}Q$. Therefore, the directional derivative of $\lambda_2(L(\boldsymbol{\theta}))$ in the direction $\theta$ is given by the smallest eigenvalue of $\sum_{i=1}^{d-1}\sum_{j=1}^{l}\theta_{ij}Q^\top L_{ij}Q$, where the matrix $Q$ is constructed by any complete set of eigenvectors corresponding to the eigenvalue $\lambda = \lambda_2(L(\boldsymbol{\theta}))$.

## 4.2 Minimising $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$.

Applying standard gradient descent methods to functions which are almost everywhere differentiable can result in convergence to sub-optimal points (Wolfe, 1972). This occurs when the method for determining the gradient is applied at a point of non-differentiability and results in a direction which is not a descent. In addition, gradients close to points of non-differentiability may be poorly conditioned from a computational perspective leading to poor performance of the optimisation.

The second eigenvalues of the graph Laplacian matrices, while not differentiable everywhere, benefit from the fact that their minimisation tends to lead to a separation from other eigenvalues. Thus a naive gradient descent algorithm tends to perform well. Notice also that if $u \in \mathbb{R}^N$ with $\|u\| = 1$ and $u \perp \mathbf{1}$ is such that $u^\top L(\boldsymbol{\theta}) u = \lambda_2(L(\boldsymbol{\theta}))$ for some $\boldsymbol{\theta} \in \Theta$, then for any $\boldsymbol{\theta}' \in \Theta$ with $u^\top L(\boldsymbol{\theta}') u < u^\top L(\boldsymbol{\theta}) u$ we have $\lambda_2(L(\boldsymbol{\theta}')) < \lambda_2(L(\boldsymbol{\theta}))$, since $u^\top L(\boldsymbol{\theta}') u$ is an upper bound for $\lambda_2(L(\boldsymbol{\theta}'))$. Thus even if $\lambda_2(L(\boldsymbol{\theta}))$ is a repeated eigenvalue, a descent direction for $u^\top L(\boldsymbol{\theta}) u$ is a descent direction for $\lambda_2(L(\boldsymbol{\theta}))$, where $u$ is any corresponding eigenvector. However, this property does not necessarily hold for $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ since the first eigenvector of $L_{\text{norm}}(\boldsymbol{\theta})$ depends on $\boldsymbol{\theta}$, and thus its second eigenvector $u$ will not necessarily be orthogonal to the first eigenvector of $L_{\text{norm}}(\boldsymbol{\theta}')$.

We assume that the similarity function, $s$, is Lipschitz continuous and continuously differentiable in $P$ for each $i, j$, and hence the Laplacian matrices $L(\boldsymbol{\theta})$ and $L_{\text{norm}}(\boldsymbol{\theta})$ are element-wise Lipschitz continuous and continuously differentiable in $\boldsymbol{\theta}$. These conditions are sufficient for the everywhere directional differentiability of $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$. Our approach for finding locally minimal solutions for $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ can be seen as a simplification of the general method of Overton and Womersley (1993). Our method alternates between a naive application of a standard gradient based optimisation algorithm, in which the simplicity of the second eigenvalue is assumed to hold everywhere along the optimisation path, and a descent step which (in general) decouples the second eigenvalue. We again discuss only $\lambda_2(L(\boldsymbol{\theta}))$ explicitly, where $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ is analogous. A description of the method is found in Algorithm 1. Notice that upon convergence of a gradient descent algorithm which assumes the simplicity of $\lambda_2(L(\boldsymbol{\theta}))$, if $\lambda_2(L(\boldsymbol{\theta}))$ is simple then the solution is a local minimum, and so the algorithm terminates. If $\lambda_2(L(\boldsymbol{\theta}))$ is not simple, then the solution may or may not be a local minimum. As we discuss in Section 4.1, if $\boldsymbol{\theta}$ is such that $\lambda_2(L(\boldsymbol{\theta}))$ is not simple, then the directional derivative of $\lambda_2(L(\boldsymbol{\theta}))$ in direction $\theta$ is given by the smallest eigenvalue of $\sum_{i=1}^{d-1} \sum_{j=1}^{l} \theta_{ij} Q^\top L_{ij} Q$, where $Q$ is the matrix with columns corresponding to a complete set of

eigenvectors for $\lambda=\lambda_2(L(\boldsymbol{\theta}))$, and $L_{ij}=\partial L(\boldsymbol{\theta})/\partial\boldsymbol{\theta}_{ij}$. If $Q^\top L_{ij}Q=\mathbf{0}$ for all $i=1,...,d-1; j=1,...,l$, then $\boldsymbol{\theta}$ is a local minimum and the method terminates, otherwise $\exists\theta\in\Theta$ s.t. $\lambda_1\left(\sum_{i=1}^{d-1}\sum_{j=1}^{l}\theta_{ij}Q^\top L_{ij}Q\right)<0$, and thus $\theta$ is a descent direction for $\lambda_2(L(\boldsymbol{\theta}))$. It is possible to find a locally steepest descent direction by minimising $\lambda_1\left(\frac{1}{\|\theta\|}\sum_{i=1}^{d-1}\sum_{j=1}^{l}\theta_{ij}Q^\top L_{ij}Q\right)$ over $\theta$, however the added computational cost associated with this subproblem outweighs the benefit over a simply chosen unit coordinate vector. Notice that the directional derivative of $\lambda_{k+2}(L(\boldsymbol{\theta}))$ in direction $\theta$ is given by the $(k+1)$-th eigenvalue of $\sum_{i=1}^{d-1}\sum_{j=1}^{l}\theta_{ij}Q^\top L_{ij}Q$, for $k=0,1,...,t-1$, where $t$ is the multiplicty of the eigenvalue $\lambda=\lambda_2(L(\boldsymbol{\theta}))$. Therefore if there exists $i\in\{1,...,d-1\}, j\in\{1,...,l\}$ s.t. $\lambda_t(Q^\top L_{ij}Q)>0$ and is simple then $-e_{ij}$, where $-e_{ij}$ is the $(d-1)\times l$ matrix with zeros except in the $i,j$-th entry where it takes the value 1, is a descent direction and $\exists\gamma>0$ s.t. $\lambda_2(L(\boldsymbol{\theta}-\gamma' e_{ij}))<\lambda_3(L(\boldsymbol{\theta}-\gamma' e_{ij}))$ for all $0<\gamma'<\gamma$. On the other hand if $\lambda_1(Q^\top L_{ij}Q)<0$ and is simple, then $e_{ij}$ is such a descent direction. If no such pair $i,j$ exists, then we select $i,j$ which maximises $\max\{\lambda_t(Q^\top L_{ij}Q), -\lambda_1(Q^\top L_{ij}Q)\}$ and set $\theta=-e_{ij}$ if the maximum was determined by the largest eigenvalue and equal to $e_{ij}$ otherwise.

### Computational Complexity

Here we give a very brief discussion of the computational complexity of the proposed method. At each iteration in the gradient descent, computing the projected dataset, $P(\boldsymbol{\theta})$, requires $\mathcal{O}(Nld)$ operations. Computing all pairwise similarities from elements of the $l$-dimensional $P(\boldsymbol{\theta})$ has computational complexity $\mathcal{O}(lN^2)$, and determining both Laplacian matrices, and their associated eigenvalue/vector pairs adds a further computational cost $\mathcal{O}(N^2)$. Each evaluation of the objectives $\lambda_2(L(\boldsymbol{\theta}))$ or $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ therefore requires $\mathcal{O}(lN(N+d))$ operations.

In order to compute the gradients of these objectives, the partial derivatives with respect to each element of the projected data matrix need to be calculated. As we discuss in the appendix to this chapter, the majority of the terms in the sums in Eq.'s (3.15) and (3.16) are zero, and in fact each partial derivative can be computed in $\mathcal{O}(N)$ time, and so all such partial derivatives can be computed in $\mathcal{O}(lN^2)$ time. The matrix derivatives $D_{\boldsymbol{\theta}_i}V_i, i=1,...,l$, in (3.17) can each be computed with $\mathcal{O}(d(d-1))$ operations. Finally, determining the gradients with respect to each column of $\boldsymbol{\theta}$ involves computing the matrix product $D_{\boldsymbol{\theta}_i}\lambda=D_{P_i}\lambda D_{V_i}P_i D_{\boldsymbol{\theta}_i}V_i$, where $D_{P_i}\lambda\in\mathbb{R}^{1\times N}$, $D_{V_i}P_i\in\mathbb{R}^{N\times d}$ and $D_{\boldsymbol{\theta}_i}V_i\in\mathbb{R}^{d\times(d-1)}$. This has complexity $\mathcal{O}(Nd(d-1))$. The complete gradient calculation therefore requires $\mathcal{O}(lN(N+d(d-1)))$ operations.

We have found that the directional derivative step is seldom, if ever required, and moreover that this does not constitute the bottleneck in the running time of the method in practice. The complexity of this step can be com-

**Algorithm 1:** Minimising $\lambda_2(L(\boldsymbol{\theta}))$

1. Initialise $\boldsymbol{\theta}$.
2. Apply gradient based optimisation to $\lambda_2(L(\boldsymbol{\theta}))$ assuming differentiability
3. **if** $\lambda_2(L(\boldsymbol{\theta}))$ is simple **then**
   **return** $\boldsymbol{\theta}$
4. Find $Q \in \mathbb{R}^{N\times t}$, a complete set of $t$ eigenvectors for eigenvalue $\lambda=\lambda_2(L(\boldsymbol{\theta}))$.
   Find $L_{ij}=\partial L(\boldsymbol{\theta})/\partial\boldsymbol{\theta}_{ij}$ for $i=1,\ldots,d-1$, and $j=1,...,l$
5. **if** $Q^\top L_{ij} Q=\mathbf{0}\ \forall i=1,\ldots,d-1; j=1,...,l$ **then**
   **return** $\boldsymbol{\theta}$
6. **if** $\exists i\in\{1,\ldots,d-1\}; j\in\{1,...,l\}$ s.t. $\lambda_t(Q^\top L_{ij}Q)>0$ and is simple **then**
   $\boldsymbol{\theta}\leftarrow \text{argmin}_{\boldsymbol{\theta}'}\lambda_2(L(\boldsymbol{\theta}'))$ s.t. $\boldsymbol{\theta}'=\boldsymbol{\theta}-\gamma e_{ij}$, $\gamma>0$, $\lambda_2(L(\boldsymbol{\theta}'))$ is simple
   **go to** 2.
7. **if** $\exists i\in\{1,\ldots,d-1\}; j\in\{1,...,l\}$ s.t. $\lambda_1(Q^\top L_{ij}Q)<0$ and is simple **then**
   $\boldsymbol{\theta}\leftarrow \text{argmin}_{\boldsymbol{\theta}'}\lambda_2(L(\boldsymbol{\theta}'))$ s.t. $\boldsymbol{\theta}'=\boldsymbol{\theta}+\gamma e_{ij}$, $\gamma>0$, $\lambda_2(L(\boldsymbol{\theta}'))$ is simple
   **go to** 2.
8. $(I,J)\leftarrow \text{argmax}_{(i,j)}\max\{\lambda_t(Q^\top L_{ij}Q), -\lambda_1(Q^\top L_{ij}Q)\}$
9. **if** $\lambda_t(Q^\top L_{IJ}Q)>-\lambda_1(Q^\top L_{IJ}Q)$ **then**
   $\boldsymbol{\theta}\leftarrow \text{argmin}_{\boldsymbol{\theta}'}\lambda_2(L(\boldsymbol{\theta}'))$ s.t. $\boldsymbol{\theta}'=\boldsymbol{\theta}-\gamma e_{IJ}, \gamma>0$
   **go to** 4.
10. $\boldsymbol{\theta}\leftarrow \text{argmin}_{\boldsymbol{\theta}'}\lambda_2(L(\boldsymbol{\theta}'))$ s.t. $\boldsymbol{\theta}'=\boldsymbol{\theta}+\gamma e_{IJ}, \gamma>0$
    **go to** 4.
11. **end**

puted along similar lines, and be found to be $\mathcal{O}(t^2 lN(N+d(d-1)))$, where $t$ is the multiplicity of the eigenvalue $\lambda=\lambda_2(L(\boldsymbol{\theta}))$.

The total complexity of the projection pursuit optimisation depends on the number of iterations in the gradient descent method, where in general this number is bounded for a given accuracy level. For our experiments we use the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm as this has been found to perform well even on non-smooth functions (Lewis and Overton, 2013).

## 4.3 Computing Similarities

It is common to define pairwise similarities of points via a decreasing function of the distance between them. That is, for a decreasing function $k:\mathbb{R}^+\to\mathbb{R}^+$, the similarity function $s$ may be written,

$$s(P,i,j)=k\left(\frac{d(p_i,p_j)}{\sigma}\right), \tag{3.21}$$

where $d(\cdot,\cdot)$ is a metric and $\sigma>0$ is a *scaling parameter*. We have found that the projection pursuit method which we propose can be susceptible to outliers when the standard Euclidean distance metric is used, especially in the case of minimising $\lambda_2(L(\boldsymbol{\theta}))$. In this subsection we discuss how to embed a balancing constraint into the distance function. By including this balancing mechanism the projection pursuit is steered away from projections which result in only few data being separated from the remainder of the data set.

While the normalisation of the graph cut objective, given in (3.2), is extremely effective in emphasising balanced partitions in the general spectral clustering problem (von Luxburg, 2007), we have found that in the projection pursuit formulation a further emphasis on balance is sometimes required. This is especially the case in high dimensional applications. Consider the extreme case where $d>N$. Then the projection equation, $V^\top X=P$, is an underdetermined system of linear equations. Therefore for any desired projected data set $P$ there exist $\boldsymbol{\theta}\in\Theta, c\in\mathbb{R}\setminus\{0\}$ s.t. $V(\boldsymbol{\theta})^\top X=cP$. In other words the projected data can be made to have any distribution, up to a scaling constant. In particular we can generally find projections which induce a sufficient separation of a small group of points from the remainder of the data that the normalisation in (3.2) is inadequate to obtain a balanced partition. We have observed that in practice even for problems of moderate dimension this situation can occur. The importance of including a balancing constraint in the context of projection pursuit for clustering has been observed previously by Zhang et al. (2009) and Pavlidis et al. (2015).

Emphasising balanced partitions is achieved through the use of a compact constraint set $\Delta$, which may be defined using the distribution of the projected data set $P$. By defining the metric $d(\cdot,\cdot)$ in such a way that distances between points extending beyond $\Delta$ are reduced, we increase the similarity of points outside $\Delta$ with others. If $P$ is $l$ dimensional then we define $\Delta$ as the rectangle $\Delta=\prod_{i=1}^{l}\Delta_i$, where each $\Delta_i$ is an interval in $\mathbb{R}$ which is defined using the distribution of the $i$-th component of $P$. A convenient way of increasing similarties with points lying outside $\Delta$ is with a transformation $T_\Delta:\mathbb{R}^l\to\mathbb{R}^l$, defined as follows,

$$T_\Delta(y)=\left(t_{\Delta_1}(y_1),\ldots,t_{\Delta_l}(y_l)\right), \tag{3.22}$$

$$t_{\Delta_i}(z):=\begin{cases} -\delta\left(\min\Delta_i-z+(\delta(1-\delta))^{\frac{1}{\delta}}\right)^{1-\delta}+\delta(\delta(1-\delta))^{\frac{1-\delta}{\delta}}, & z<\min\Delta_i \\ z-\min\Delta_i, & z\in\Delta_i \\ \delta\left(z-\max\Delta_i+(\delta(1-\delta))^{\frac{1}{\delta}}\right)^{1-\delta}-\delta(\delta(1-\delta))^{\frac{1-\delta}{\delta}}, & \\ +\mathrm{Diam}(\Delta_i), & z>\max\Delta_i, \end{cases} \tag{3.23}$$

where $\delta\in(0,.5]$ is the distance reducing parameter. Each $t_{\Delta_i}$ is linear on $\Delta_i$ but has a smaller gradient outside $\Delta_i$. As a result we have $\|T_\Delta(x)-T_\Delta(y)\|\leq\|x-$

$y\|$ for any $x,y\in\mathbb{R}^l$, with strict inequality whenever either $x$ or $y$ lies outside $\Delta$. We define $T_\Delta$ in this way so that it is continuously differentiable even at the boundaries of $\Delta$, and so does not affect the differentiability properties of the similarity function, $s$. Figure 3.1 illustrates how the function $T_\Delta$ influences distances and similarities in the univariate case.

In the context of projection pursuit it is convenient to define a full dimensional convex constraint set $\mathbf{\Delta}\subset\mathbb{R}^d$ and define the univariate constraint intervals, which we now index by the corresponding projection angles, via the projection of $\mathbf{\Delta}$ onto each $V(\boldsymbol{\theta})_i$. That is,

$$\Delta_{\boldsymbol{\theta}i} := \left[\min\{V(\boldsymbol{\theta})_i^\top x | x\in\mathbf{\Delta}\}, \max\{V(\boldsymbol{\theta})_i^\top x | x\in\mathbf{\Delta}\}\right]. \tag{3.24}$$

In our implementation, we define $\mathbf{\Delta}$ to be a scaled covariance ellipsoid centered at the mean of the data. The projections of $\mathbf{\Delta}$ are thus given by intervals of the form,

$$\Delta_{\boldsymbol{\theta}i} = [\mu_{\boldsymbol{\theta}i} - \beta\sigma_{\boldsymbol{\theta}i}, \mu_{\boldsymbol{\theta}i} + \beta\sigma_{\boldsymbol{\theta}i}], \tag{3.25}$$

where $\mu_{\boldsymbol{\theta}i}$ and $\sigma_{\boldsymbol{\theta}i}$ are the mean and standard deviation of the $i$-th component of the projected data set $P(\boldsymbol{\theta})$ and the parameter $\beta\geq 0$ determines the width of the projected constraint interval $\Delta_{\boldsymbol{\theta}i}$.

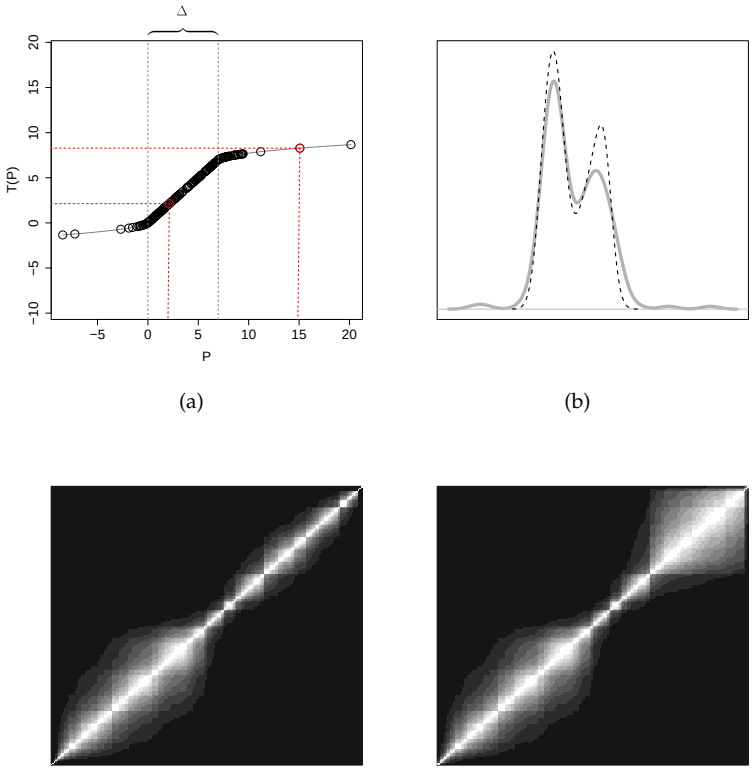
Figure 3.2 shows two dimensional projections of the 64 dimensional optical recognition of handwritten digits dataset[1]. The leftmost plot shows the PCA projection which is used as initialisation for the projection pursuit. The remaining plots show the projections arising from the minimisation of $\lambda_2(L(\boldsymbol{\theta}))$ for a variety of values of $\beta$. For $\beta=\infty$, i.e., an unconstrained projection, the projection pursuit focuses only on a few data and leaves the remainder of the dataset almost unaffected by the projection. Setting $\beta=2.5$ causes the projection pursuit to focus on a larger proportion of the tail of the data distribution. Setting $\beta=1.5$ however allows the projection pursuit to identify the cluster structure in the data and find a projection which provides a good separation of three of the clusters in the data, i.e., those shown as black, orange and turquoise in the top right plot.

## 4.4 Correlated and Orthogonal Projections

The formulation of the optimisation problem associated with projection pursuit based on spectral connectivity places no constraints on the projection matrix, $V$, except that its columns should have unit norm. A common consideration in dimension reduction methods is that the columns in the projection matrix should be orthogonal, i.e., $V_i^\top V_j = 0$ for all $i\neq j$. In the context of projection pursuit it is common to generate the columns of $V$ individually, so that orthogonality of the columns can easily be enforced. Huber

[1] `https://archive.ics.uci.edu/ml/datasets.html`

**Fig. 3.1:** Effect of $T_\Delta$ on Distances and Similarities.



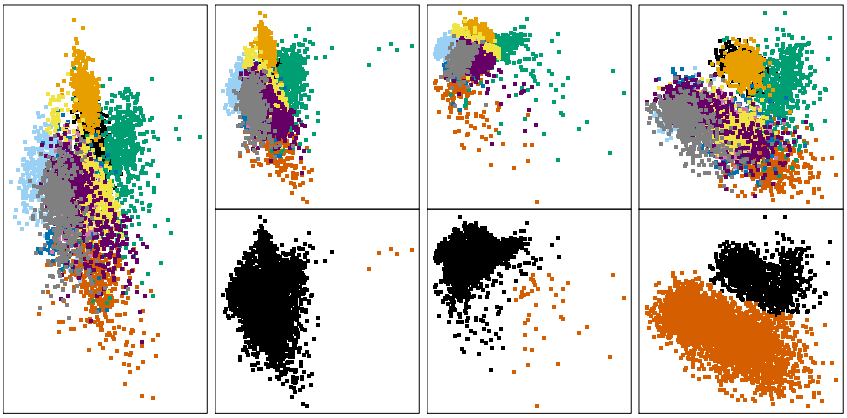

(a) (b)

(c) (d)

(a) The univariate data set $P$ is plotted against the transformed data $T_\Delta(P)$. The point at $\approx$15 lies outside $\Delta$ and its distance to other points, e.g. the point at $\approx$2, is smaller within $T_\Delta(P)$ (vertical axis) than in $P$ (horizontal axis). (b) The kernel density estimate of the transformed data $T_\Delta(P)$ (- - -) has a stronger bimodal structure, i.e., two well defined clusters, than that of $P$ (——), which has multiple small modes caused by outliers. The connection between spectral connectivity and density based clustering has been investigated theoretically by Narayanan et al. (2006) showing that the normalised graph cut is asymptotically related to the density on the separating surface. (c) The affinity matrix of the data set $P$ has a weaker cluster structure than that of $T_\Delta(P)$, shown in (d).

(1985) proposes first learning $V_1$ using the original data, and then for each subsequent column the data are first projected into the null space of all the columns learnt so far. An alternative approach (Niu et al., 2011), is to instead project the gradient of the objective into this null space during a gradient

**Fig. 3.2:** Two dimensional projections of optical recognition of handwritten digits dataset arising from the minimisation of $\lambda_2(L(\boldsymbol{\theta}))$, for different values of $\beta$. In addition, the initialisation through PCA is also shown. The top row of plots shows the true clusters, while the bottom row shows resulting bi-partitions.



(a) PCA (b) $\beta=\infty$ (c) $\beta=2.5$ (d) $\beta=1.5$

based optimisation. Notice, however, that orthogonality in the columns of $V$ is not essential for the underlying problem. Another common approach (Huber, 1985) is to transform the data after each column is determined in such a way that the columns learned so far are no longer "interesting", i.e., have low projection index. This does not enforce orthogonality, but rather steers the projection pursuit away from the columns already learned by making them unattractive to the optimisation method.

We propose a different approach which allows us to learn all of the columns of $V$ simultaneoulsy. This is achieved by introducing an additional term to the objective function which controls the level of orthogonality, or alternatively correlation, within the projection matrix. In particular, we consider the objective,

$$\min_{\boldsymbol{\theta}\in\Theta} \lambda_2(L(\boldsymbol{\theta})) + \omega \sum_{i\neq j} (V(\boldsymbol{\theta})_i^\top V(\boldsymbol{\theta})_j)^2, \tag{3.26}$$

or replacing $\lambda_2(L(\boldsymbol{\theta}))$ with $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ in the normalised case. This approach serves a dual purpose. In the first case, setting $\omega>0$ leads to approximately orthogonal projections, without the need to optimise separately over different projection vectors as is standard. Alternatively, setting $\omega<0$ leads to approximately perfect correlation, i.e., $V(\boldsymbol{\theta})_i \approx \pm V(\boldsymbol{\theta})_j$ for all $i,j$. In the latter case the resulting projection is therefore equivalent to a univariate projection. This is similar to simultaneously considering multiple initialisations, and allowing the optimisation procedure to select from them automatically. This is

important as the objectives $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ are non-convex, and as a result applying gradient based optimisation can only guarantee convergence to a local optimum.

Notice also that the formulation in Eq. (3.26) offers computational benefits over the alternative of optimising separately over each projection vector, since the eigenvalues/vectors computed in each function and gradient evaluation need only be computed once for each iteration over the multiple projection dimensions.

# 5 Connection with Maximal Margin Hyperplanes

In this section we establish a connection between the optimal univariate projection for spectral bi-partitioning using the standard Laplacian and large margin separators. In particular, under suitable conditions, as the scaling parameter tends to zero the optimal projection for spectral bi-partitioning converges to the vector admitting the largest margin hyperplane through the data. This establishes a theoretical connection between spectral connectedness and separability of the resulting clusters in terms of Euclidean distance. Large margin separators are ubiquitous in the machine learning literature, and were first introduced in the context of supervised classification via support vector machines (SVM, Vapnik and Kotz (1982)). In more recent years they have shown to be very useful for unsupervised partitioning in the context of maximum margin clustering as well (Xu et al., 2004; Zhang et al., 2009).

Our result pertains to univariate projections, and therefore the $d \times 1$ projection matrix is equivalently viewed as a *projection vector* in $\mathbb{R}^d$. We therefore use the notation $v(\boldsymbol{\theta})$, instead of $V(\boldsymbol{\theta})$ as before.

The result holds for all similarities for which the function $k$, in Eq. (3.21), satisfies the tail condition $\lim_{x\to\infty} k(x+\epsilon)/k(x) = 0$ for all $\epsilon > 0$. This conditions is satisfied by functions with exponentially decaying tails, including the popular Gaussian and Laplace kernels. It is, however, not satisfied by those with polynomially decaying tails.

The constraint set $\boldsymbol{\Delta}$ again plays an important role as in many cases the largest margin hyperplane through a set of data separates only a few points from the rest, making it meaningless for the purpose of clustering. We therefore prefer to restrict the hyperplane to intersect the set $\boldsymbol{\Delta}$. What we in fact show in this section is that there exists a set $\boldsymbol{\Delta}' \subset \boldsymbol{\Delta}$ satisfying $\boldsymbol{\Delta}' \cap X = \boldsymbol{\Delta} \cap X$, such that, as the scaling parameter tends to zero, the optimal projection for $\lambda_2(L(\boldsymbol{\theta}))$ converges to the projection admitting the largest margin hyperplane that intersects $\boldsymbol{\Delta}'$. The distinction between the largest margin hyperplane intersecting $\boldsymbol{\Delta}'$ and that intersecting $\boldsymbol{\Delta}$ is scarcely of practical relevance, but plays an important role in the theory we present in this section. It accounts

for situations when the largest margin hyperplane intersecting $\mathbf{\Delta}$ lies close to its boundary and the distance between the hyperplane and the nearest point outside $\mathbf{\Delta}$ is larger than to the nearest point inside $\mathbf{\Delta}$. Aside from this very specific case, the two in fact coincide.

A hyperplane is a translated subspace of co-dimension 1, and can be parameterised by a non-zero vector $v \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ and scalar $b$ as the set $H(v,b) = \{x \in \mathbb{R}^d \big| v^\top x = b\}$. Clearly, for any $c \in \mathbb{R} \setminus \{0\}$, one has $H(v,b) = H(cv,cb)$, and so we can assume that $v$ has unit norm, thus the same parameterisation by $\boldsymbol{\theta}$ can be used. For a finite set of points $X \subset \mathbb{R}^d$, the *margin* of hyperplane $H(v(\boldsymbol{\theta}),b)$ w.r.t. $X$ is the minimal Euclidean distance between $H(v(\boldsymbol{\theta}),b)$ and $X$. That is,

$$\text{margin}(v(\boldsymbol{\theta}),b) = \min_{x \in X} |v(\boldsymbol{\theta})^\top x - b|. \tag{3.27}$$

Connections between maximal margin hyperplanes and Bayes optimal hyperplanes as well as minimum density hyperplanes have been established (Tong and Koller, 2000; Pavlidis et al., 2015).

In this section we use the notation $v^\top X = \{v^\top x_1, ..., v^\top x_N\}$, and for a set $P \subset \mathbb{R}$ and $y \in \mathbb{R}$ we write, for example, $P_{>y}$ for $P \cap (y, \infty)$. For scaling parameter $\sigma > 0$ and distance reduction factor $\delta > 0$ we define $\boldsymbol{\theta}_{\sigma,\delta} := \text{argmin}_{\boldsymbol{\theta} \in \Theta} \lambda_2(L(\boldsymbol{\theta},\sigma,\delta))$, where $L(\boldsymbol{\theta},\sigma,\delta)$ is as $L(\boldsymbol{\theta})$ from before, but with an explicit dependence on the scaling parameter and distance reducing parameter used in the similarity function. That is, $\boldsymbol{\theta}_{\sigma,\delta}$ defines the projection generating the minimal spectral connectivity of $X$ for a given pair $\sigma,\delta$.

Before proving the main result of this section, we require the following supporting results. Lemma 11 provides a lower bound on the second eigenvalue of the graph Laplacian of a one dimensional data set in terms of the largest Euclidean separation of adjacent points, with respect to a constraint set $\Delta$. This lemma also shows how we construct the set $\mathbf{\Delta}'$. Lemma 12 uses this result to show that a projection angle $\boldsymbol{\theta} \in \Theta$ leads to lower spectral connectivity than all projections admitting smaller maximal margin hyperplanes intersecting $\mathbf{\Delta}'$ for all pairs $\sigma,\delta$ sufficiently close to zero.

**Lemma 11** *Let $k : \mathbb{R}_+ \to \mathbb{R}_+$ be a non-increasing, positive function and let $\sigma > 0, \delta \in (0, 0.5]$. Let $P = \{p_1, ..., p_N\}$ be a univariate data set and let $\Delta = [a,b]$ for $a < b \in \mathbb{R}$. Suppose that $|P \cap \Delta| \geq 2$ and $a \geq \min\{P\}, b \leq \max\{P\}$. Define $\Delta' = [a', b']$, where $a' = (a + \min\{P \cap \Delta\})/2$, $b' = (b + \max\{P \cap \Delta\})/2$. Let $M = \max_{x \in \Delta'} \{\min_{i=1...N} |x - p_i|\}$. Define $L(P)$ to be the Laplacian of the graph with vertices $P$ and similarities according to $s(P,i,j) = k(|T_\Delta(p_i) - T_\Delta(p_j)|/\sigma)$. Then $\lambda_2(L(P)) \geq \frac{1}{|P|^3} k((2M + \delta C)/\sigma)$, where $C = \max\{D, D^{1-\delta}\}$, $D = \max\{a - \min\{P\}, \max\{P\} - b\}$.*

**Proof** We can assume that $P$ is sorted in increasing order, i.e. $p_i \leq p_{i+1}$, since this does not affect the eigenvalues of $L(P)$. We first show that $s(P,i,i+1) \geq$

$k((2M+\delta C)/\sigma)$ for all $i=1,...,N-1$. To this end observe that for $x \geq 0$ we have $\delta\left(x+\left(\delta(1-\delta)^{\frac{1}{\delta}}\right)\right)^{1-\delta} - \delta(\delta(1-\delta))^{\frac{1-\delta}{\delta}} \leq \delta \max\{x, x^{1-\delta}\}$.

- If $p_i, p_{i+1} \leq a$ then $s(P,i,i+1) = k((T_\Delta(p_{i+1}) - T_\Delta(p_i))/\sigma) \geq k((T_\Delta(a) - T_\Delta(p_i))/\sigma) \geq k((2M+\delta C)/\sigma)$ by the definition of $C$ and using the above inequality, since $k$ is non-increasing. The case $p_i, p_{i+1} \geq b$ is similar.
- If $p_i, p_{i+1} \in \Delta$ then $p_i, p_{i+1} \in \Delta' \Rightarrow |p_i - p_{i+1}| \leq 2M \Rightarrow s(P,i,i+1) \geq k(2M/\sigma) \geq k((2M+\delta C)/\sigma)$ since $M$ is the largest margin in $\Delta'$.
- If none the above hold, then we lose no generality in assuming $p_i < a$, $a < p_{i+1} < b$ since the case $a < p_i < b$, $p_{i+1} > b$ is analogous. We must have $p_{i+1} = \min\{P \cap \Delta\}$ and so $a' = (a + p_{i+1})/2$. If $p_{i+1} - a > 2M$ then $\min_{j=1...N} |a' - p_j| > M$, a contradiction since $a' \in \Delta'$ and $M$ is the largest margin in $\Delta'$. Therefore $p_{i+1} - a \leq 2M$. In all $T_\Delta(p_{i+1}) - T_\Delta(p_i) = (p_{i+1} - a) + \delta(a - p_i + (\delta(1-\delta))^{\frac{1}{\delta}})^{1-\delta} - \delta(\delta(1-\delta))^{\frac{1-\delta}{\delta}} \leq 2M + \delta C \Rightarrow s(P,i,i+1) \geq k((2M+\delta C)/\sigma)$.

Now, let $u$ be the second eigenvector of $L(P)$. Then $\|u\| = 1$ and $u \perp \mathbf{1}$ and therefore $\exists i,j$ s.t. $u_i - u_j \geq \frac{1}{\sqrt{|P|}}$. We thus know that there exists $m$ s.t. $|u_m - u_{m+1}| \geq \frac{1}{|P|^{3/2}}$. By (von Luxburg, 2007, Proposition 1), we know that $u^\top L(P) u = \frac{1}{2}\sum_{i,j} s(P,i,j)(u_i - u_j)^2 \geq s(P,m,m+1)(u_m - u_{m+1})^2 \geq \frac{1}{|P|^3} k((2M+\delta C)/\sigma)$ since all consecutive pairs $p_m$, $p_{m+1}$ have similarity at least $k((2M+\delta C)/\sigma)$, by above. Therefore $\lambda_2(L(P)) \geq \frac{1}{|P|^3} k((2M+\delta C)/\sigma)$ as required.

In the above Lemma we have assumed that $\Delta$ is contained within the convex hull of the points $P$, however the results of this section can easily be modified to allow for cases where this does not hold. In particular, if an unconstrained large margin hyperplane is sought, then setting $\mathbf{\Delta}$ to be arbitrarily large allows for this. We have merely stated the results in the most convenient context for our practical implementation.

The set $\Delta'$ in the above is defined in terms of the one dimensional constraint set $[a,b]$. We define the full dimensional set $\mathbf{\Delta}'$ along the same lines by,

$$\mathbf{\Delta}' := \{x \in \mathbb{R}^d | v(\boldsymbol{\theta})^\top x \in \Delta'_{\boldsymbol{\theta}} \ \forall \boldsymbol{\theta} \in \Theta\},$$
$$\Delta'_{\boldsymbol{\theta}} = \left[\frac{\min \Delta_{\boldsymbol{\theta}} + \min\{v(\boldsymbol{\theta})^\top X \cap \Delta_{\boldsymbol{\theta}}\}}{2}, \frac{\max \Delta_{\boldsymbol{\theta}} + \max\{v(\boldsymbol{\theta})^\top X \cap \Delta_{\boldsymbol{\theta}}\}}{2}\right]. \quad (3.28)$$

Here we assume that $\mathbf{\Delta}$ is contained within the convex hull of the $d$-dimensional data set $X$. Notice that since $\mathbf{\Delta}$ is convex, we have $v(\boldsymbol{\theta})^\top \mathbf{\Delta}' = \Delta'_{\boldsymbol{\theta}}$. In what follows we show that as $\sigma$ and $\delta$ are reduced to zero the optimal projection for spectral partitioning converges to the projection admitting the largest margin

hyperplane intersecting $\boldsymbol{\Delta}'$. If it is the case that the largest margin hyperplane intersecting $\boldsymbol{\Delta}$ also intersects $\boldsymbol{\Delta}'$, as is often the case, although this fact will not be known, then it is actually not necessary that $\delta$ tend towards zero. In such cases it only needs to satisfy $\delta \leq 2M/C$ for the corresponding values of $M$ and $C$ over all possible projections. In particular, choosing $\max\{\text{Diam}(X), \text{Diam}(X)^{1-\delta}\}$ instead of $C$ is appropriate for all projections.

**Lemma 12** *Let $\boldsymbol{\theta} \in \Theta$ and let $k : \mathbb{R}_+ \to \mathbb{R}_+$ be non-increasing, positive, and satisfy*

$$\lim_{x\to\infty} k(x(1+\epsilon))/k(x) = 0$$

*for all $\epsilon > 0$. Then for any $0 < m < \max_{b\in\Delta'_{\boldsymbol{\theta}}} \mathit{margin}(v(\boldsymbol{\theta}), b)$ there exists $\sigma' > 0$ and $\delta' > 0$ s.t. $0 < \sigma < \sigma'$, $0 < \delta < \delta'$ and $\max_{c\in\Delta'_{\boldsymbol{\theta}'}} \mathit{margin}(v(\boldsymbol{\theta}'), c) < \max_{b\in\Delta'_{\boldsymbol{\theta}}} \mathit{margin}(v(\boldsymbol{\theta}), b) - m \Rightarrow \lambda_2(L(\boldsymbol{\theta}, \sigma, \delta)) < \lambda_2(L(\boldsymbol{\theta}', \sigma, \delta))$.*

**Proof** Let $B = \text{argmax}_{b\in\Delta'_{\boldsymbol{\theta}}} \text{margin}(v(\boldsymbol{\theta}), b)$ and $M = \text{margin}(v(\boldsymbol{\theta}), B)$. We assume that $M \neq 0$, since otherwise there is nothing to show. Now, since spectral clustering solves a relaxation of the minimum normalised cut problem we have,

$$\begin{aligned}
\lambda_2(L(\boldsymbol{\theta},\sigma,\delta)) &\leq \frac{1}{|X|} \min_{C\subset X} \sum_{\substack{i,j: x_i\in C \\ x_j\notin C}} s(P(\boldsymbol{\theta}), i, j) \left( \frac{1}{|C|} + \frac{1}{|X\setminus C|} \right) \\
&\leq \frac{1}{|X|} \sum_{\substack{i,j: v(\boldsymbol{\theta})^\top x_i < B \\ v(\boldsymbol{\theta})^\top x_j > B}} s(P(\boldsymbol{\theta}), i, j) \left( \frac{1}{|(v(\boldsymbol{\theta})^\top X)_{<B}|} + \frac{1}{|(v(\boldsymbol{\theta})^\top X)_{>B}|} \right) \\
&= \frac{1}{|X|} \sum_{\substack{i,j: v(\boldsymbol{\theta})^\top x_i < B \\ v(\boldsymbol{\theta})^\top x_j > B}} k\left( \frac{T_{\Delta_{\boldsymbol{\theta}}}(v(\boldsymbol{\theta})^\top x_j) - T_{\Delta_{\boldsymbol{\theta}}}(v(\boldsymbol{\theta})^\top x_i)}{\sigma} \right) \\
&\qquad\qquad \times \left( \frac{|X|}{|(v(\boldsymbol{\theta})^\top X)_{<B}||(v(\boldsymbol{\theta})^\top X)_{>B}|} \right) \\
&\leq \left|(v(\boldsymbol{\theta})^\top X)_{<B}\right| \left|(v(\boldsymbol{\theta})^\top X)_{>B}\right| k\left( \frac{2M}{\sigma} \right) \left( \frac{1}{|(v(\boldsymbol{\theta})^\top X)_{<B}||(v(\boldsymbol{\theta})^\top X)_{>B}|} \right) \\
&= k(2M/\sigma).
\end{aligned}$$

The final inequality holds since for any $i, j$ s.t. $v(\boldsymbol{\theta})^\top x_i < B$ and $v(\boldsymbol{\theta})^\top x_j > B$ we must have $T_{\Delta_{\boldsymbol{\theta}}}(v(\boldsymbol{\theta})^\top x_j) - T_{\Delta_{\boldsymbol{\theta}}}(v(\boldsymbol{\theta})^\top x_i) \geq 2M$. Now, for any $\boldsymbol{\theta}' \in \Theta$, let $M_{\boldsymbol{\theta}'} = \max_{c\in\Delta'_{\boldsymbol{\theta}'}} \text{margin}(v(\boldsymbol{\theta}'), c)$. By Lemma 11 we know that $\lambda_2(L(\boldsymbol{\theta}', \sigma, \delta)) \geq$

$\frac{1}{|X|^3}k((2M_{\boldsymbol{\theta}'}+\delta C/\sigma)$, where $C=\max\{\text{Diam}(X), \text{Diam}(X)^{1-\delta}\}$. Therefore,

$$\lim_{\substack{\sigma\to 0^+,\\ \delta\to 0^+}} \frac{\lambda_2(L(\boldsymbol{\theta},\sigma,\delta))}{\inf_{\boldsymbol{\theta}'\in\Theta}\{\lambda_2(L(\boldsymbol{\theta}',\sigma,\delta))\big|M_{\boldsymbol{\theta}'}<M-m\}} \leq \lim_{\substack{\sigma\to 0^+,\\ \delta\to 0^+}} \frac{|X|^3 k(2M/\sigma)}{k((2(M-m)+\delta C)/\sigma)} = 0.$$

This gives the result.

Lemma 12 shows almost immediately that the margin admitted by the optimal projection for spectral bi-partitioning converges to the largest margin through $\boldsymbol{\Delta}'$ as $\sigma$ and $\delta$ go to zero. The main result of this section, Theorem 13, shows the stronger result that the optimal projection itself converges to the projection admitting the largest margin.

**Theorem 13** *Let $X=\{x_1,...,x_N\}$ and suppose that there is a unique hyperplane, which can be parameterised by $(v(\boldsymbol{\theta}^\star),b^\star)$, intersecting $\boldsymbol{\Delta}'$ and attaining maximal margin on $X$. Let $k:\mathbb{R}_+\to\mathbb{R}_+$ be decreasing, positive and satisfy $\lim_{x\to\infty}k((1+\epsilon)x)/k(x)=0$ for all $\epsilon>0$. Then,*

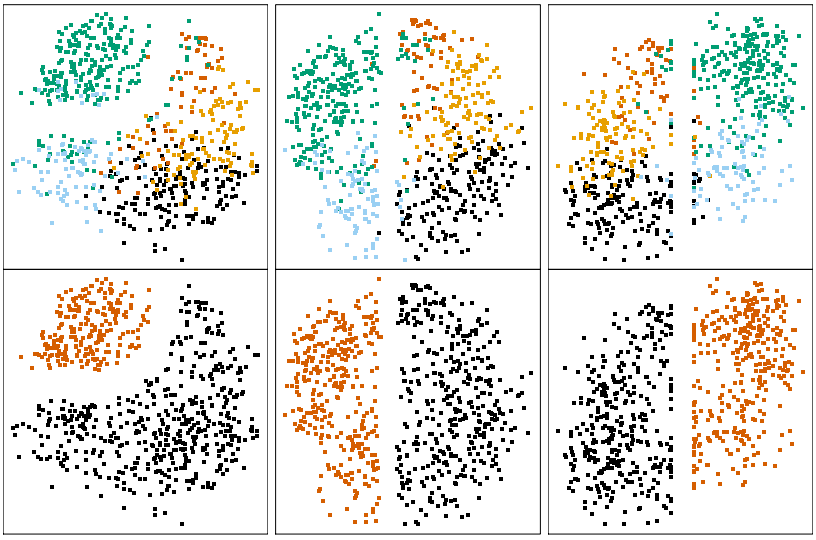
$$\lim_{\sigma\to 0^+,\delta\to 0^+} v(\boldsymbol{\theta}_{\sigma,\delta}) = v(\boldsymbol{\theta}^\star).$$

**Proof** Take any $\epsilon>0$. Pavlidis et al. (2015) have shown that $\exists m_\epsilon>0$ s.t. for $w\in\mathbb{R}^d, c\in\mathbb{R}$, we have $\|(w,c)/\|w\|-(v(\boldsymbol{\theta}^\star),b^\star)\|>\epsilon \Rightarrow \text{margin}(w/\|w\|,c/\|w\|) < \text{margin}(v(\boldsymbol{\theta}^\star),b^\star)-m_\epsilon$. By Lemma 12 we know $\exists\sigma'>0$, $\delta'>0$ s.t. if $0<\sigma<\sigma'$ and $0<\delta<\delta'$ then $\exists c\in\Delta_{\boldsymbol{\theta}}$ s.t. $\text{margin}(v(\boldsymbol{\theta}_{\sigma,\delta}),c) \geq \text{margin}(v(\boldsymbol{\theta}^\star),b^\star)-m_\epsilon$, since $\boldsymbol{\theta}_{\sigma,\delta}$ is optimal for the pair $\sigma,\delta$. Thus, by above, $\|(v(\boldsymbol{\theta}_{\sigma,\delta}),c)-(v(\boldsymbol{\theta}^\star),b^\star)\|\leq\epsilon$. But $\|(v(\boldsymbol{\theta}_{\sigma,\delta}),c)-(v(\boldsymbol{\theta}^\star),b^\star)\|\geq\|v(\boldsymbol{\theta}_{\sigma,\delta})-v(\boldsymbol{\theta}^\star)\|$ for any $c\in\mathbb{R}$. Since $\epsilon>0$ was arbitrary, we therefore have $v(\boldsymbol{\theta}_{\sigma,\delta})\to v(\boldsymbol{\theta}^\star)$ as $\sigma,\delta\to 0^+$.

While the results of this section apply only for univariate projections, we have observed empirically that if a shrinking sequence of scaling parameters is employed for a multivariate projection, then the projected data tend to display large Euclidean separation. This is illustrated in Figure 3.3 which shows two dimensional plots of the 72 dimensional yeast cell cycle analysis dataset[2]. As in Figure 3.2 the top plots show the true clusters in the data and the bottom plots show the clustering assignments. The left plots show the result of a two dimensional projection pursuit using the proposed method. In the middle plots the first projection is learnt using one dimensional projection pursuit, and the second is set to be the direction of maximum variance within the null space of the first projection. The right plots use as the first projection the result of the iterative support vector regression method for maximum margin clustering (Zhang et al., 2009), and again the second projection is the direction of maximum variance within its null space.

[2] `http://genome-www.stanford.edu/cellcycle/`

**Fig. 3.3:** Large Euclidean separation of the yeast cell cycle dataset. The left plots show the result from a 2 dimensional projection pursuit using the proposed method. The middle plots show the 1 dimensional projection pursuit result. The right plots show the result of the maximum margin clustering method of Zhang et al. (2009).



A similar intuition which underlies the theoretical results of this section can be used to reason why this will occur in multivariate projections, in that as the scaling parameter reduces to zero the value of $\lambda_2(L(\boldsymbol{\theta}))$ is controlled by the smallest distance between observations in different elements of the induced partition. It is however elusive how to formulate this rigorously in the presence of the constraint set $\boldsymbol{\Delta}$ in more than one dimension.

# 6 Speeding up Computation using Microclusters

In this section we discuss how a preprocessing of the data using *microclusters* can be used to significantly speed up the optimisation process. We derive theoretical bounds on the error induced by this approximation. Our approach uses a result from matrix perturbation theory for diagonally dominant matrices, and therefore only applies to the standard Laplacian, $L(\boldsymbol{\theta})$. However, we have seen empirically that a close approximation of the optimisation surface is obtained for both $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$.

The concept of a microcluster was introduced by Zhang and Ramakrishnan (1996) in the context of clustering very large data sets. Microclusters are small clusters of data which can in turn be clustered to generate a clustering of the entire data set. A microcluster like approach in the context of spectral clustering has been considered by Yan et al. (2009), where the authors ob-

tain bounds on the mis-clustering rate induced by the approximation. Rather than using microclusters as an intermediate step towards determining a final clustering model, we use them to form an approximation of the optimisation surface for projection pursuit which is less computationally expensive to explore. The error bound depends on the ratio of cluster radii to scaling parameter. As such, this method does not provide a good approximation when $\sigma$ is close to zero. Our bounds rely on the following result from perturbation theory.

**Theorem 14** *Ye (2009)*
*Let $A=[a_{ij}]$ and $\tilde{A}=[\tilde{a}_{ij}]$ be two symmetric positive semidefinite diagonally dominant matrices, and let $\lambda_1\leq\lambda_2\leq...\leq\lambda_n$ and $\tilde{\lambda}_1\leq\tilde{\lambda}_2\leq...\leq\tilde{\lambda}_n$ be their respective eigenvalues. If, for some $0\leq\epsilon<1$, $|a_{ij}-\tilde{a}_{ij}|\leq\epsilon|a_{ij}|\ \forall i\neq j$, and $|v_i-\tilde{v}_i|\leq\epsilon v_i\ \forall i$, where $v_i=a_{ii}-\sum_{j\neq i}|a_{ij}|$, and similarly for $\tilde{v}_i$, then*

$$|\lambda_i-\tilde{\lambda}_i|\leq\epsilon\lambda_i\ \forall i.$$

An inspection of the proof of Theorem 14 reveals that $\epsilon<1$ is necessary only to ensure that the signs of $a_{ij}$ are the same as those of $\tilde{a}_{ij}$. In the case of Laplacian matrices this equivalence of signs holds by design, and so in this context the requirement that $\epsilon<1$ can be relaxed.

In the microcluster approach, the data set $X=\{x_1,...,x_N\}$ is replaced with $K$ points $c_1,...,c_K$ which represent the centers of a $K$-clustering of $X$. By projecting these microcluster centers during subspace optimisation, rather than the data themselves, the computational cost associated with each eigen problem is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(K^2)$. If we define the radius, $\rho$, of a cluster $C$ to be the greatest distance between one of its members and its center, that is,

$$\rho(C)=\max_{x\in C}\left\|x-\frac{1}{|C|}\sum_{x\in C}x\right\|, \tag{3.29}$$

then we expect the approximation error to be small whenever the microcluster radii are small. The bounds on the approximation error which we present in this section are worst case and rely on standard eigenvalue bounds, and so can be pessimistic. To obtain a reasonable bound on the approximation surface, as many as $K\approx 0.6N$ might be needed, leading to only a threefold speed up. We have observed empirically, however, that even for $K=0.1N$ (and sometimes lower) one still obtains a close approximation of the optimisation surface. This makes the projection pursuit of the order of 100 times faster.

**Lemma 15** *Let $\mathcal{C}=C_1,...,C_K$ be a $K$-clustering of $X$ with centers $c_1,...,c_K$, radii $\rho_1,...,\rho_K$ and counts $n_1,...,n_K$. For $\boldsymbol{\theta}\in\Theta$ define $N(\boldsymbol{\theta}),B(\boldsymbol{\theta})\in\mathbb{R}^{K\times K}$ where $N(\boldsymbol{\theta})$ is*

*the diagonal matrix with*

$$N(\boldsymbol{\theta})_{i,i} = \sum_{j=1}^{K} n_j s(P^c(\boldsymbol{\theta}), i, j)$$

*and*

$$B(\boldsymbol{\theta})_{i,j} = \sqrt{n_i n_j}\, s(P^c(\boldsymbol{\theta}), i, j),$$

*where* $P^c(\boldsymbol{\theta}) = \{V(\boldsymbol{\theta})^\top c_1, ..., V(\boldsymbol{\theta})^\top c_K\}$ *are the projected microcluster centers and the similarities are given by* $s(P^c(\boldsymbol{\theta}), i, j) = k(\|T_{\Delta_{\boldsymbol{\theta}}}(V(\boldsymbol{\theta})^\top c_i) - T_{\Delta_{\boldsymbol{\theta}}}(V(\boldsymbol{\theta})^\top c_j)\|/\sigma)$*, and* $k(x)$ *is positive and non-increasing for* $x \geq 0$*. Then,*

$$\frac{|\lambda_2(L(\boldsymbol{\theta})) - \lambda_2(N(\boldsymbol{\theta}) - B(\boldsymbol{\theta}))|}{\lambda_2(L(\boldsymbol{\theta}))} \leq \max_{i \neq j} \max \left\{ 1 - \frac{k(D_{ij}/\sigma)}{k((D_{ij} - \rho_i - \rho_j)^+/\sigma)}, \frac{k(D_{ij}/\sigma)}{k((D_{ij} + \rho_i + \rho_j)/\sigma)} - 1 \right\},$$

*where* $D_{ij} = \|T_{\Delta_{\boldsymbol{\theta}}}(V(\boldsymbol{\theta})^\top c_i) - T_{\Delta_{\boldsymbol{\theta}}}(V(\boldsymbol{\theta})^\top c_j)\|$ *and* $(x)^+ = \max\{0, x\}$.

**Proof** For brevity we temporarily drop the notational dependence on $\boldsymbol{\theta}$. Let $P^{c\prime} = \{V^\top c_1, V^\top c_1, ..., V^\top c_K, V^\top c_K\}$, where each $V^\top c_i$ is repeated $n_i$ times. Let $L^{c\prime}$ be the Laplacian of the graph with vertices $P^{c\prime}$ and edges given by $s(P^{c\prime}, i, j)$. We begin by showing that $\lambda_2(L^{c\prime}) = \lambda_2(N - B)$. Take $v \in \mathbb{R}^K$, then,

$$\begin{aligned} v^\top (N - B) v &= \sum_{i,j} s(P^c, i, j)(v_i^2 n_j - v_i v_j \sqrt{n_i n_j}) \\ &= \frac{1}{2} \sum_{i,j} s(P^c, i, j)(v_i^2 n_j + v_j^2 n_i - 2 v_i v_j \sqrt{n_i n_j}) \\ &\geq 0, \end{aligned}$$

and so $N - B$ is positive semi-definite. In addition, it is straightforward to verify that $(N - B)(\sqrt{n_1}\ \ldots\ \sqrt{n_K}) = \mathbf{0}$, and hence 0 is the smallest eigenvalue of $N - B$ with eigenvector $(\sqrt{n_1}\ \ldots\ \sqrt{n_K})$. Now, let $u$ be the second eigenvector of $L^{c\prime}$. Then $u_j = u_k$ for pairs of indices $j,k$ aligned with the same $V^\top c_i$ in $P^{c\prime}$. Define $u^c \in \mathbb{R}^K$ s.t. $u_i^c = \sqrt{n_i} u_j$ where index $j$ is aligned with $V^\top c_i$ in $P_j^{c\prime}$. Then $(u^c)^\top (\sqrt{n_1}\ \ldots\ \sqrt{n_K}) = \sum_{i=1}^K u_i^c \sqrt{n_i} = \sum_{i=1}^K n_i u_{j_i}$ where index $j_i$ is aligned with $V^\top c_i$ in $P_{j_i}^{c\prime}$ for each $i$. Therefore $n_i u_{j_i} = \sum_{j : P^{c\prime} = V^\top c_i} u_j$ and hence $(u^c)^\top (\sqrt{n_1}\ \ldots\ \sqrt{n_K}) = \sum_{i=1}^K \sum_{j : P_j^{c\prime} = V^\top c_i} u_j = \sum_{i=1}^N u_i = 0$ since $\mathbf{1}$ is the smallest eigenvector of $L^{c\prime}$ and so $u \perp \mathbf{1}$. Similarly $\|u^c\|^2 = \sum_{i=1}^K n_i u_{j_i}^2 = \sum_{i=1}^N u_i^2 = 1$. Thus $u^c \perp (\sqrt{n_1}\ \ldots\ \sqrt{n_K})$ and $\|u^c\| = 1$ and so is a candidate for the second eigenvector of $N - B$. In addition it is straightforward to show that $(u^c)^\top (N - B) u^c = u \cdot L^{c\prime} u$. Now, suppose by way of contradiction that $\exists w \perp (\sqrt{n_1}\ \ldots\ \sqrt{n_K})$

with $\|w\|=1$ s.t. $w^\top(N-B)w<(u^c)^\top(N-B)u^c$. Let $w'=(w_1/\sqrt{n_1}\ w_1/\sqrt{n_1}\ \ldots\ w_K/\sqrt{n_K})$ where each $w_i/\sqrt{n_i}$ is repeated $n_i$ times. Then $\|w'\|=1$, $(w')^\top\mathbf{1}=w^\top(\sqrt{n_1}\ \ldots\ \sqrt{n_K})=0$ and $w^\top L^{c\prime}w<u^\top L^{c\prime}u$, a contradiction since $u$ is the second eigenvector of $L^{c\prime}$.

Now, let $i,j,m,n$ be such that $x_m\in C_i$ and $x_n\in C_j$. We temporarily drop the notational dependence on $\Delta$. Then,

$$\begin{aligned}\|T(V^\top x_m)-T(V^\top x_n)\|&=\|T(V^\top x_m)-T(V^\top c_i)+T(V^\top c_i)-T(V^\top c_j)\\&\quad+T(V^\top c_j)-T(V^\top x_n)\|\\&\le\|T(V^\top x_m)-T(V^\top c_i)\|+\|T(V^\top c_i)-T(V^\top c_j)\|\\&\quad+\|T(V^\top c_j)-T(V^\top x_n)\|\\&\le\rho_i+\rho_j+D_{ij},\end{aligned}$$

since $T$ contracts distances and $\rho_i$ and $\rho_j$ are the radii of $C_i$ and $C_j$. Since $k$ is non-increasing we therefore have,

$$\frac{k(D_{ij}/\sigma)}{k((D_{ij}-\rho_i-\rho_j)^+/\sigma)}\le\frac{k(D_{ij}/\sigma)}{k(\|T(V^\top x_m)-T(V^\top x_n)\|/\sigma)}\le\frac{k(D_{ij}/\sigma)}{k((D_{ij}+\rho_i+\rho_j)/\sigma)}$$

$$\Rightarrow 1-\frac{k(D_{ij}/\sigma)}{k(\|T(V^\top x_m)-T(V^\top x_n)\|/\sigma)}\le 1-\frac{k(D_{ij}/\sigma)}{k((D_{ij}-\rho_i-\rho_j)^+/\sigma)}$$

and

$$\frac{k(D_{ij}/\sigma)}{k(\|T(V^\top x_m)-T(V^\top x_n)\|/\sigma)}-1\le\frac{k(D_{ij}/\sigma)}{k((D_{ij}+\rho_i+\rho_j)/\sigma)}-1.$$

Therefore

$$\left|\frac{k(D_{ij}/\sigma)}{k(\|T(V^\top x_m)-T(V^\top x_n)\|/\sigma)}-1\right|\le\max\left\{1-\frac{k(D_{ij}/\sigma)}{k((D_{ij}-\rho_i-\rho_j)^+/\sigma)},\ \frac{k(D_{ij}/\sigma)}{k((D_{ij}+\rho_i+\rho_j)/\sigma)}-1\right\}.$$

Now, we lose no generality by assume that $X$ is ordered such that for each $i$ the elements of cluster $C_i$ are aligned with $V^\top c_i$ in $P^{c\prime}$, since this does not affect the eigenvalues of the Laplacian of $V^\top X$, $L$. By the design of the Laplacian matrix the "$v_i$" of Theorem 14 are exactly zero. For off diagonal terms $m,n$ with corresponding $i,j$ as above, consider

$$\begin{aligned}\frac{|L_{mn}-L^{c\prime}_{mn}|}{|L_{mn}|}&=\frac{|k(D_{ij}/\sigma)-k(\|T(V^\top x_m)-T(V^\top x_n)\|/\sigma)|}{k(\|T(V^\top x_m)-T(V^\top x_n)\|/\sigma)}\\&=\left|\frac{k(D_{ij}/\sigma)}{k(\|T(V^\top x_m)-T(V^\top x_n)\|/\sigma)}-1\right|.\end{aligned}$$

Theorem 14 thus gives the result.

The above bound depends on $\boldsymbol{\theta}$ via the quantity $D_{ij}$ and for some functions $k$ it is difficult to remove this dependence. We consider the class of functions, parameterised by $\alpha \geq 0$, and given by

$$k(x) = \left(\frac{|x|}{\alpha} + 1\right)^{\alpha} \exp(-|x|), \tag{3.30}$$

where we adopt the convention $(\frac{a}{0})^0 = 1$ for any $a \in \mathbb{R}$. For $\alpha = 0$ this is equivalent to the Laplace kernel, but for $\alpha > 0$ has the useful property of being differentiable at 0. We have found the choice of $k$ to matter little in the results of the proposed approach. The above class of functions is chosen as it allows us to obtain a uniform bound on the error induced by the above approximation. Note the parameter $\alpha$ is not intended as a tuning parameter, but rather we set $\alpha$ close to zero to obtain a function similar to the Laplace kernel, but which is differentiable at zero.

**Corollary 16** *Let the conditions of Lemma 15 hold, and let $k(x)$ be defined as in Eq. (3.30). Then,*

$$\frac{|\lambda_2(L(\boldsymbol{\theta})) - \lambda_2(N(\boldsymbol{\theta}) - B(\boldsymbol{\theta}))|}{\lambda_2(L(\boldsymbol{\theta}))} \leq \max_{i \neq j} \left(\frac{Diam(X) + \sigma\alpha}{Diam(X) + \rho_i + \rho_j + \sigma\alpha}\right)^{\alpha} \exp\left(\frac{\rho_i + \rho_j}{\sigma}\right) - 1.$$

**Proof** Firstly, consider

$$\frac{k(D_{ij}/\sigma)}{k((D_{ij} + \rho_i + \rho_j)/\sigma)} - 1 = \left(\frac{D_{ij} + \sigma\alpha}{D_{ij} + \rho_i + \rho_j + \sigma\alpha}\right)^{\alpha} \exp\left(\frac{\rho_i + \rho_j}{\sigma}\right) - 1.$$

Now, the function $\left(\frac{x+\sigma\alpha}{x+y+\sigma\alpha}\right)^{\alpha} \exp(y/\sigma)$ is non-decreasing in $x$ for $x, y, \alpha, \sigma \geq 0$, therefore by above

$$\frac{k(D_{ij}/\sigma)}{k((D_{ij} + \rho_i + \rho_j)/\sigma)} - 1 \leq \left(\frac{\mathrm{Diam}(X) + \sigma\alpha}{\mathrm{Diam}(X) + \rho_i + \rho_j + \sigma\alpha}\right)^{\alpha} \exp\left(\frac{\rho_i + \rho_j}{\sigma}\right) - 1.$$

Secondly, consider the case $D_{ij} \geq \rho_i + \rho_j$, then

$$\begin{aligned} 1 - \frac{k(D_{ij}/\sigma)}{k((D_{ij} - \rho_i - \rho_j)^+/\sigma)} &= 1 - \left(\frac{D_{ij} + \sigma\alpha}{D_{ij} - \rho_i - \rho_j + \sigma\alpha}\right)^{\alpha} \exp\left(-\frac{\rho_i + \rho_j}{\sigma}\right) \\ &\leq 1 - \left(\frac{\mathrm{Diam}(X) + \rho_i + \rho_j + \sigma\alpha}{\mathrm{Diam}(X) + \sigma\alpha}\right)^{\alpha} \exp\left(-\frac{\rho_i + \rho_j}{\sigma}\right), \end{aligned}$$

since $\left(\frac{x+\sigma\alpha}{x-y+\sigma\alpha}\right)^{\alpha} \exp(y/\sigma)$ is non-increasing in $x$ for $x, y, \alpha, \sigma \geq 0$. On the other

hand, if $D_{ij} < \rho_i + \rho_j$ then,

$$\begin{aligned}1 - \frac{k(D_{ij}/\sigma)}{k((D_{ij} - \rho_i - \rho_j)^+/\sigma)} &= 1 - k\left(\frac{D_{ij}}{\sigma}\right) \leq 1 - k\left(\frac{\rho_i + \rho_j}{\sigma}\right) \\ &= 1 - \left(\frac{\rho_i + \rho_j + \sigma\alpha}{\sigma\alpha}\right)^{\alpha} \exp\left(-\frac{\rho_i + \rho_j}{\sigma}\right) \\ &\leq 1 - \left(\frac{\mathrm{Diam}(X) + \rho_i + \rho_j + \sigma\alpha}{\mathrm{Diam}(X) + \sigma\alpha}\right)^{\alpha} \exp\left(-\frac{\rho_i + \rho_j}{\sigma}\right),\end{aligned}$$

where the first inequality comes from the fact that $D_{ij} < \rho_i + \rho_j$ and $k$ is decreasing. Now, using the identity $1 - \frac{1}{x} \leq x - 1$ for $x \neq 0$, we have

$$\begin{aligned}1 - \left(\frac{\mathrm{Diam}(X) + \rho_i + \rho_j + \sigma\alpha}{\mathrm{Diam}(X) + \sigma\alpha}\right)^{\alpha} \exp\left(-\frac{\rho_i + \rho_j}{\sigma}\right) \leq \\ \left(\frac{\mathrm{Diam}(X) + \sigma\alpha}{\mathrm{Diam}(X) + \rho_i + \rho_j + \sigma\alpha}\right)^{\alpha} \exp\left(\frac{\rho_i + \rho_j}{\sigma}\right) - 1,\end{aligned}$$

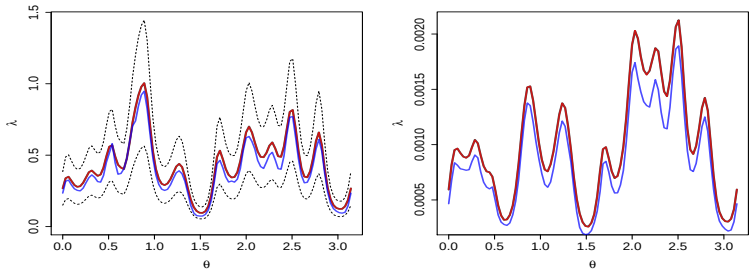
and so Lemma 15 gives the result.

Tighter bounds can be derived if pairwise distances between elements from pairs of clusters are compared directly to the distances between the cluster centers, and for higher dimensional cases the additional tightness can be significant. We prefer to state the result as above due to the sole reliance on the internal cluster radii relative to scaling parameter.

While bounds of the above type are not verifiable for $L_{\mathrm{norm}}$ due to the fact that it is not diagonally dominant, a similar degree of agreement between the true and approximate eigenvalues has been observed in all cases considered. In this case the $K \times K$ matrix is given by the normalised Laplacian of the graph of $P^C(\boldsymbol{\theta})$ with similarities given by $n_i n_j s(P^C(\boldsymbol{\theta}), i, j)$. This matrix has the same structure as the original normalised Laplacian, the only difference being the introduction of the factors $n_i, n_j$.

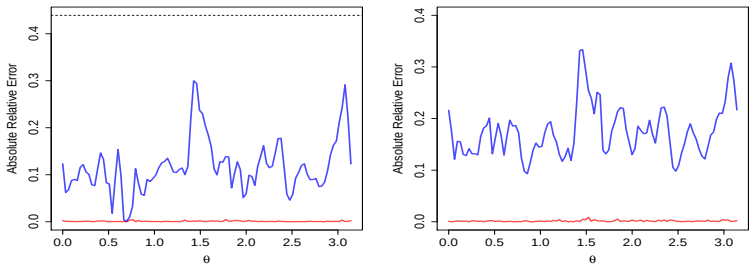
Figure 3.4 shows (a) $\lambda_2(L(\boldsymbol{\theta}))$ and (b) $\lambda_2(L_{\mathrm{norm}}(\boldsymbol{\theta}))$ plotted against the single projection angle $\boldsymbol{\theta}$ for the 2 dimensional S1 data set (Fränti and Virmajoki, 2006). The parameter $\sigma$ was chosen using the same method as for our experiments. A complete linkage clustering was performed for 3000 microclusters (=60% of total number of data), as well as for 200 microclusters for comparison. The true values of $\lambda_2(L(\boldsymbol{\theta}))$ and those based on approximations using 3000 microclusters are almost indistinguishable. The approximations based on 200 microclusters also show a good approximation of the optimisation surface, and lie well within the bounds pertaining to the 3000 microcluster case. The same sort of agreement can be seen for $\lambda_2(L_{\mathrm{norm}}(\boldsymbol{\theta}))$. Importantly,

**Fig. 3.4:** Approximation Error Plots for S1 data set.




(a) $\lambda_2(L(\boldsymbol{\theta}))$ with Theoretical Error Bounds

(b) $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$




(c) $\frac{|\lambda_2(L(\boldsymbol{\theta}))-\hat{\lambda}|}{\lambda_2(L(\boldsymbol{\theta}))}$ with Theoretical Bound

(d) $\frac{|\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))-\hat{\lambda}|}{\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))}$

True eigenvalue (——), bounds based on 3000 microclusters (- - -), approximation using 3000 microclusters (——), approximation using 200 microclusters (——)

while the approximations based on 200 microclusters slightly underestimate the true eigenvalues, the location of the local minima, and indeed the shape of the optimisation surface, are very similar to the truth, and so optimising over this approximate surface leads to near optimal projections. We also show the absolute relative error, (c) and (d), as described in Lemma 16. The pessimism of the bound is clearly evident in the bottom left plot where the values of $\frac{|\lambda_2(L(\boldsymbol{\theta}))-\lambda_2(N(\boldsymbol{\theta})-B(\boldsymbol{\theta}))|}{\lambda_2(L(\boldsymbol{\theta}))}$ appear very close to zero on the scale of the theoretical bound.

# 7 Experimental Results

In this section we evaluate the proposed method on a large collection of benchmark datasets. We compare our approach with existing dimension reduction methods for clustering, where the final clustering result is determined using spectral clustering. In addition we consider solving our problem iteratively for a shrinking sequence of scaling parameters to find large margin separators, relying on the theoretical results presented in Section 5. We compare these results with the iterative support vector regression approach of Zhang et al. (2009)[3], a state-of-the-art maximum margin clustering algorithm.

We compare the different methods based on two popular evaluation metrics for clustering, namely purity (Zhao and Karypis, 2004) and $V$-measure (Rosenberg and Hirschberg, 2007). Both metrics compare the label assignments made by a clustering algorithm with the true class labels of the data. They take values in $[0,1]$ with larger values indicating a better agreement between the two label sets, and hence a superior clustering result. Purity is the weighted average of the largest proportion of each cluster which can be represented by a single class. $V$-measure is defined as the harmonic mean of measures of completeness and homogeneity. Homogeneity is similar to purity, in that it measures the extent to which each cluster may be represented by a single class, but is given by the weighted average of the entropy of the class distribution within each cluster. Completeness is symmetric to homogeneity, and measures the entropy of the cluster distribution within each class. $V$-measure therefore also captures the extent to which classes are split between clusters.

We will use the following notation throughout this section:

- SCP$^2$ and SC$_n$P$^2$ refer to the proposed projection pursuits for minimising $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$ respectively.
- LMSC refers to the proposed approach of finding large margin separators, based on repeatedly minimising $\lambda_2(L(\boldsymbol{\theta}))$ for a shrinking sequence of scaling parameters.
- Subscripts "$o$" and "$c$" indicate whether we use an orthogonal projection ($\omega > 0$) or a correlated one ($\omega < 0$), respectively.
- SC and SC$_n$ refer to spectral clustering based on the eigen-decompositions of $L$ and $L_{\text{norm}}$ respectively.
- Subscripts "$PCA$" and "$ICA$" indicate principal and independent component analysis projections respectively. For example, $SC_{nPCA}$ refers to

[3]We are grateful to Dr. Kai Zhang for supplying us with code to implement this method.

spectral clustering using the normalised Laplacian applied to the data projected into a principal component oriented subspace.

- DRSC abbreviates dimensionality reduction for spectral clustering, proposed by Niu et al. (2011). This existing approach applies only to the normalised Laplacian.

- iSVR$_L$ and iSVR$_G$ denote the iterative support vector regression approach for maximum margin clustering (Zhang et al., 2009), using the linear and Gaussian kernels respectively.

## 7.1 Details of Implementation

To extend our approach to datasets containing multiple clusters, we simply recursively partition (subsets of) the dataset until the desired number of clusters is obtained. We prefer this approach to the alternative of directly seeking a projection which yields a full $K$ way partition of the dataset, i.e., by minimising the sum of the first $K$ eigenvalues of the Laplacians, as it is not always clear that all the clusters present in the data can be exposed using a single projection of fixed dimension. At each iteration in this recursive bi-partitioning we split the largest remaining cluster.

The scaling parameter and initialisation are set for each bi-partition, given values determined by the subset of the data being partitioned. For the fixed scaling parameter approaches, SCP$^2$ and SC$_n$P$^2$, we set $\sigma = \sqrt{l\lambda_d}N^{-1/5}$, where $l$ is the dimension of the projection, $N$ is the size of the (subset of the) data and $\lambda_d$ is the largest eigenvalue of the covariance matrix. The value $\sqrt{\lambda_d}$ captures the scale of the data, while $\sqrt{l}$ accounts for the fact that distances scale roughly with the square root of the dimension. The denominator term, $N^{-1/5}$, is borrowed from kernel density methods and we have found it to work reasonably well for our applications as well. For the large margin approach, LMSC, $\sigma$ is initialised at $\sqrt{l\lambda_d}N^{-1/5}$ and decreased by a factor of two with each minimisation of $\lambda_2(L(\boldsymbol{\theta}))$, until convergence of the projection matrix. The initialisation of $V(\boldsymbol{\theta})$ is via the first $l$ principal components. For the orthogonal projections we use a two dimensional projection, as this is the lowest dimensional space which can expose non-linear separation between clusters. For the correlated projections we provide a three dimensional initialisation, and it was found that in most cases a high quality univariate projection could be determined from this.

For the LMSC approach, because the values within the Laplacian matrix approach zero, the optimisation becomes less robust, and we found that the correlated approach did not always lead to large margin separation. We believe this is as a result of the term controlling the correlation becoming too dominant relative to the decreasing eigenvalue unless very careful tuning

of $\omega$ is performed. We therefore consider a univariate projection instead of the multivariate correlated approach in this case.

Recall that the parameter $\beta$ controls the size of the constraint set $\Delta$. It is clear that smaller values of $\beta$ will tend to lead to more balanced partitions, but a precise interpretation of the resulting cluster sizes is unavailable. At best bounds on the cluster sizes can be computed using Chebyshev's inequality. Rather than relying on these bounds, which may be loose and difficult to interpret in multivariate projections, we recommend applying the proposed method for a range of values of $\beta$ and selecting the solution corresponding to the largest value of $\beta$ which induces a specified balance in the partition. We define this balance to be satisfied if the smallest cluster size is at least half the average, i.e., $N/2K$. In this way the effect of the constraint is limited while still producing the desired result. We initialise with a large value of $\beta$ and decrease by 0.5 until the balance is met. If this balance is not met for $\beta=0.5$, then the corresponding "unbalanced" result is returned anyway.

The parameter $\delta$ is set to $\min\{0.01,\sigma^2\}$ and $\alpha$ to 0.1. We have found these two parameters not to significantly influence the performance of the method. It is important to note, however, that that parameter $\alpha$ controls the shape of the similarity function, and as a result there is an interplay between this value and the value of $\sigma$. For substantially larger values of $\alpha$ we expect a smaller value of $\sigma$ to be more appropriate.

For competing approaches based on spectral clustering we do the following. Whenever the number of data exceeds 1000 we use the approximation method of Yan et al. (2009). Following Niu et al. (2011), we set the reduced dimension to $K-1$, where $K$ is the number of clusters. We compute clustering results for all values of $\sigma$ in $\{0.1,0.2,0.5,1,2,5,10,20,50,\ 100,200\}$ as well as for the local scaling approach of Zelnik-Manor and Perona (2004), and report the highest performance in each case. For DRSC we also considered the parameter setting used for our method, and to implement the local scalings of Zelnik-Manor and Perona (2004) these were recomputed with each iteration in the corresponding projected subspace. We also provided DRSC with a warm start via PCA as this improved performance over a random initialisation, and offers a more fair comparison. Because of this extensive search over scaling parameters we expect competing methods to achieve very high performance whenever the corresponding dimension reduction is capable of exposing the clusters in the data well.

For the iSVR maximum margin clustering method, we set the balancing parameter equal to 0.3 as suggested by Zhang et al. (2009) when the cluster sizes are not balanced. We argue that the balance of the clusters will not be known in practice, and the unbalanced setting led to superior performance compared with the balanced setting in the examples considered. The iSVR approach also generates only a bi-partition, and to generate multiple clusters we apply the same recursive approach as in our method.

## 7.2 Clustering Results

The following benchmark datasets were used for comparison:

- Optical recognition of handwritten digits (Opt. Digits).[4] 5620 8×8 compressed images of handwritten digits in $\{0,...,9\}$, resulting in 64 dimensions with 10 classes.
- Pen based recognition of handwritten digits (Pen Digits).[2] 10992 observations, each corresponding to a stylus trajectory ($x$,$y$ coordinates) from a handwritten digit in $\{0,...,9\}$, i.e., 10 classes. The trajectories are sampled at 8 time points, resulting in 16 dimensions.
- Satellite.[2] 6435 multispectral values from 3×3 pixel squares from satellite images, which results in 36 dimensions. There are 6 classes corresponding to different land types.
- Breast cancer Wisconsin (Br. Cancer).[2] 699 observations with 9 attributes relating to tumour masses. There are 2 classes corresponding to benign and malignant masses.
- Congressional votes (Voters).[2] 435 sets of 16 binary decisions on US congressional ballots. The 2 classes correspond to political party membership.
- Dermatology.[2] 366 observations corresponding to dermatology patients, each containing 34 dimensions derived from clinical and histopathological features. There are 6 classes corresponding to different skin diseases.
- Yeast cell cycle analysis (Yeast).[5] 698 yeast genes across 72 conditions (dimensions). There are 5 classes corresponding to different genotypes.
- Synthetic control chart (Chart).[2] 600 simulated time series of length 60 displaying one of 6 fundamental characteristics, leading to 6 classes.
- Multiple feature digits (M.F. Digits).[2] 2000 handwritted digits in $\{0,...,9\}$ taken from Dutch utility maps. Following Niu et al. (2011) we use only the 216 profile correlation features.
- Statlog image segmentation (Image Seg.).[2] 2310 observations containing 19 features derived from 3×3 pixel squares from 7 outdoor images. Each image constitutes a class.

[4] `https://archive.ics.uci.edu/ml/datasets.html`
[5] `http://genome-www.stanford.edu/cellcycle/`

**Table 3.1:** Purity results for spectral clustering using the standard Laplacian, $L$. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold.

| | $SCP_o^2$ | $SCP_c^2$ | $SC_{PCA}$ | $SC_{ICA}$ | SC |
|---|---|---|---|---|---|
| Opt. Digits | $0.81_{0.05}$ | $\mathbf{0.83}_{0.06}$ | $0.33_{0.05}$ | $0.20_{0.02}$ | $0.11_{0.00}$ |
| Pen Digits | $\mathbf{0.78}_{0.01}$ | $0.76_{0.01}$ | $0.64_{0.02}$ | $0.53_{0.03}$ | $0.62_{0.03}$ |
| Satellite | $\mathbf{0.75}_{0.00}$ | $0.73_{0.03}$ | $0.63_{0.01}$ | $0.72_{0.01}$ | $0.62_{0.02}$ |
| Br. Cancer | $\mathbf{0.97}_{0.00}$ | $\mathbf{0.97}_{0.00}$ | $\mathbf{0.97}_{0.00}$ | $\mathbf{0.97}_{0.00}$ | $0.96_{0.00}$ |
| Voters | $0.84_{0.00}$ | $0.83_{0.00}$ | $\mathbf{0.86}_{0.00}$ | $\mathbf{0.86}_{0.00}$ | $0.78_{0.00}$ |
| Dermatology | $\mathbf{0.94}_{0.00}$ | $0.90_{0.00}$ | $0.91_{0.00}$ | $0.89_{0.00}$ | $0.56_{0.00}$ |
| Yeast | $\mathbf{0.75}_{0.00}$ | $0.74_{0.00}$ | $0.67_{0.00}$ | $0.62_{0.00}$ | $0.60_{0.00}$ |
| Chart | $\mathbf{0.84}_{0.00}$ | $0.83_{0.00}$ | $0.72_{0.06}$ | $0.65_{0.07}$ | $0.56_{0.05}$ |
| M.F. Digits | $\mathbf{0.83}_{0.02}$ | $0.79_{0.02}$ | $0.53_{0.03}$ | $0.34_{0.03}$ | $0.31_{0.04}$ |
| Image Seg. | $0.62_{0.03}$ | $\mathbf{0.68}_{0.03}$ | $0.53_{0.02}$ | $0.46_{0.02}$ | $0.56_{0.03}$ |

Before applying the clustering algorithms, data were rescaled so that every feature had unit variance. This is a standard approach to handle situations where different features are captured on different scales and an appropriate rescaling is not obviously apparent from the context. For consistency we used this same preprocessing approach for all datasets.

### Spectral Clustering Using the Standard Laplacian

Tables 3.1 and 3.2 report the purity and $V$-measure respectively for the proposed method and spectral clustering using the standard Laplacian applied to the original data, as well as their projection into PCA and ICA oriented subspaces. The tables report the average and standard deviation (as subscript) from 30 repetitions. The highest average performance for each dataset is highlighted in bold. Both the orthogonal and correlated projection approaches achieve substantially higher performance than other methods in the majority of cases. There are few cases where they are not competitive with the best performing of the competing approaches, while there are mulitple examples where the proposed methods strongly outperform all others. The two versions of the proposed method are closely comparable with one another on average, with the correlated approach offering a slightly better worst case comparison. This, however, does not appear highly significant beyond sampling variation both within each dataset and with respect to the collection of datasets used for comparison. What is evident is that the added flexibility offered by multivariate projections does not result in a substantial improvement over univariate projections, which induce linear cluster boundaries.

**Table 3.2:** $V$-measure results for spectral clustering using the standard Laplacian, $L$. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold.

| | $SCP^2_o$ | $SCP^2_c$ | $SC_{PCA}$ | $SC_{ICA}$ | SC |
|---|---|---|---|---|---|
| Opt. Digits | $0.77_{0.03}$ | $\mathbf{0.78}_{0.04}$ | $0.40_{0.05}$ | $0.21_{0.03}$ | $0.01_{0.00}$ |
| Pen Digits | $\mathbf{0.75}_{0.01}$ | $0.74_{0.02}$ | $0.66_{0.01}$ | $0.54_{0.03}$ | $0.64_{0.02}$ |
| Satellite | $\mathbf{0.60}_{0.00}$ | $0.59_{0.03}$ | $0.50_{0.00}$ | $\mathbf{0.60}_{0.01}$ | $0.50_{0.01}$ |
| Br. Cancer | $0.79_{0.00}$ | $\mathbf{0.80}_{0.00}$ | $0.78_{0.00}$ | $0.78_{0.00}$ | $0.77_{0.00}$ |
| Voters | $\mathbf{0.42}_{0.00}$ | $0.38_{0.00}$ | $0.41_{0.00}$ | $0.41_{0.00}$ | $0.30_{0.00}$ |
| Dermatology | $\mathbf{0.89}_{0.00}$ | $0.83_{0.00}$ | $0.86_{0.00}$ | $0.82_{0.00}$ | $0.58_{0.00}$ |
| Yeast | $\mathbf{0.54}_{0.00}$ | $\mathbf{0.54}_{0.00}$ | $0.51_{0.00}$ | $0.40_{0.00}$ | $0.41_{0.00}$ |
| Chart | $0.77_{0.00}$ | $0.77_{0.00}$ | $\mathbf{0.81}_{0.02}$ | $0.73_{0.04}$ | $0.70_{0.02}$ |
| M.F. Digits | $\mathbf{0.76}_{0.01}$ | $0.73_{0.02}$ | $0.56_{0.02}$ | $0.36_{0.05}$ | $0.38_{0.05}$ |
| Image Seg. | $0.62_{0.01}$ | $\mathbf{0.65}_{0.02}$ | $0.53_{0.01}$ | $0.46_{0.01}$ | $0.58_{0.02}$ |

### Spectral Clustering Using the Normalised Laplacian

Tables 3.3 and 3.4 report the purity and $V$-measure respectively for the proposed approach based on minimising $\lambda_2(L_{\text{norm}}(\boldsymbol{\theta}))$, the dimensionality reduction for spectral clustering algorithm (Niu et al., 2011) and spectral clustering based on the normalised Laplacian applied to the original data and their PCA and ICA projections. Again the tables show the average and standard deviation from 30 runs of each method, with the highest average performance on each dataset highlighted in bold.

The proposed approach using both correlated and orthogonal projections is again competitive with all other methods in almost all cases considered. In addition both versions of the proposed approach substantially outperform the other methods in multiple examples. Unlike in the case of the standard Laplacian, here there is evidence that the orthogonal projection achieves better clustering results in general, outperforming the correlated approach in the majority of examples.

### Large Margin Clustering

It is important to note that the method described in Section 6 does not provide a close approximation as $\sigma \to 0^+$. For the datasets containing more than 1000 data we use the microcluster approach for all values of $\sigma$ and therefore only guarantee a large separation between the microclusters. It is arguable that this is a preferable objective as the maximum margin is not robust in the presence of noise, and it is not clear that it converges in the general setting (Ben-David et al., 2009). Microclusters have the potential to absorb some

**Table 3.3:** Purity results for spectral clustering using the normalised Laplacian, $L_{\text{norm}}$. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold.

| | $SC_nP_o^2$ | $SC_nP_c^2$ | DRSC | $SC_{nPCA}$ | $SC_{nICA}$ | $SC_n$ |
|---|---|---|---|---|---|---|
| Opt. Digits | $\mathbf{0.81}_{0.05}$ | $0.74_{0.06}$ | $0.80_{0.03}$ | $0.66_{0.03}$ | $0.64_{0.01}$ | $0.66_{0.02}$ |
| Pen Digits | $\mathbf{0.78}_{0.00}$ | $0.74_{0.02}$ | $0.69_{0.04}$ | $0.76_{0.03}$ | $0.74_{0.01}$ | $0.77_{0.04}$ |
| Satellite | $0.75_{0.00}$ | $0.74_{0.03}$ | $0.73_{0.01}$ | $\mathbf{0.76}_{0.01}$ | $0.73_{0.02}$ | $0.74_{0.01}$ |
| Br. Cancer | $\mathbf{0.97}_{0.00}$ | $\mathbf{0.97}_{0.00}$ | $0.96_{0.00}$ | $\mathbf{0.97}_{0.00}$ | $\mathbf{0.97}_{0.00}$ | $\mathbf{0.97}_{0.00}$ |
| Voters | $0.85_{0.00}$ | $0.84_{0.00}$ | $\mathbf{0.86}_{0.00}$ | $\mathbf{0.86}_{0.00}$ | $\mathbf{0.86}_{0.00}$ | $0.85_{0.00}$ |
| Dermatology | $0.86_{0.00}$ | $0.91_{0.00}$ | $0.87_{0.00}$ | $0.92_{0.02}$ | $0.91_{0.00}$ | $\mathbf{0.95}_{0.00}$ |
| Yeast | $\mathbf{0.76}_{0.00}$ | $0.70_{0.00}$ | $0.62_{0.00}$ | $0.71_{0.00}$ | $0.69_{0.01}$ | $0.60_{0.00}$ |
| Chart | $\mathbf{0.87}_{0.00}$ | $0.85_{0.00}$ | $0.75_{0.00}$ | $0.71_{0.06}$ | $0.80_{0.07}$ | $0.75_{0.00}$ |
| M.F. Digits | $\mathbf{0.84}_{0.01}$ | $0.79_{0.01}$ | $0.77_{0.03}$ | $0.77_{0.03}$ | $0.79_{0.01}$ | $0.77_{0.02}$ |
| Image Seg. | $0.66_{0.01}$ | $\mathbf{0.70}_{0.01}$ | $0.65_{0.04}$ | $0.61_{0.03}$ | $0.55_{0.02}$ | $0.62_{0.02}$ |

**Table 3.4:** $V$-measure results for spectral clustering using the normalised Laplacian, $L_{\text{norm}}$. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold.

| | $SC_nP_o^2$ | $SC_nP_c^2$ | DRSC | $SC_{nPCA}$ | $SC_{nICA}$ | $SC_n$ |
|---|---|---|---|---|---|---|
| Opt. Digits | $\mathbf{0.77}_{0.03}$ | $0.72_{0.04}$ | $0.75_{0.02}$ | $0.62_{0.02}$ | $0.60_{0.01}$ | $0.62_{0.01}$ |
| Pen Digits | $\mathbf{0.76}_{0.00}$ | $0.74_{0.01}$ | $0.66_{0.02}$ | $0.72_{0.01}$ | $0.68_{0.01}$ | $0.73_{0.02}$ |
| Satellite | $0.61_{0.01}$ | $0.60_{0.03}$ | $0.57_{0.01}$ | $\mathbf{0.62}_{0.01}$ | $0.60_{0.05}$ | $0.60_{0.01}$ |
| Br. Cancer | $0.79_{0.00}$ | $\mathbf{0.81}_{0.00}$ | $0.76_{0.00}$ | $\mathbf{0.81}_{0.00}$ | $\mathbf{0.81}_{0.00}$ | $0.79_{0.00}$ |
| Voters | $\mathbf{0.42}_{0.00}$ | $0.41_{0.00}$ | $\mathbf{0.42}_{0.00}$ | $0.41_{0.00}$ | $0.41_{0.00}$ | $\mathbf{0.42}_{0.00}$ |
| Dermatology | $0.85_{0.00}$ | $0.86_{0.00}$ | $0.80_{0.00}$ | $0.86_{0.02}$ | $0.83_{0.00}$ | $\mathbf{0.91}_{0.00}$ |
| Yeast | $\mathbf{0.57}_{0.00}$ | $0.45_{0.00}$ | $0.41_{0.00}$ | $0.54_{0.00}$ | $0.53_{0.00}$ | $0.45_{0.00}$ |
| Chart | $\mathbf{0.81}_{0.00}$ | $0.80_{0.00}$ | $0.75_{0.00}$ | $\mathbf{0.81}_{0.02}$ | $0.80_{0.05}$ | $0.73_{0.00}$ |
| M.F. Digits | $\mathbf{0.76}_{0.01}$ | $0.74_{0.02}$ | $0.69_{0.02}$ | $0.71_{0.02}$ | $0.75_{0.01}$ | $0.70_{0.02}$ |
| Image Seg. | $0.65_{0.01}$ | $\mathbf{0.68}_{0.01}$ | $0.62_{0.04}$ | $0.60_{0.02}$ | $0.46_{0.02}$ | $0.60_{0.02}$ |

**Table 3.5:** Purity results for large margin clustering. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold.

| | $\text{LMSC}_o$ | LMSC | $\text{iSVR}_L$ | $\text{iSVR}_G$ |
|---|---|---|---|---|
| Opt. Digits | $0.74_{0.05}$ | $0.69_{0.05}$ | $\mathbf{0.76}_{0.00}$ | $0.61_{0.01}$ |
| Pen Digits | $\mathbf{0.80}_{0.02}$ | $0.71_{0.04}$ | $0.78_{0.00}$ | $0.78_{0.00}$ |
| Satellite | $\mathbf{0.75}_{0.01}$ | $0.72_{0.03}$ | $0.68_{0.00}$ | $0.68_{0.00}$ |
| Br. Cancer | $0.96_{0.00}$ | $\mathbf{0.97}_{0.00}$ | $0.90_{0.00}$ | $0.95_{0.00}$ |
| Voters | $\mathbf{0.84}_{0.00}$ | $\mathbf{0.84}_{0.00}$ | $0.81_{0.00}$ | $\mathbf{0.84}_{0.00}$ |
| Dermatology | $\mathbf{0.86}_{0.00}$ | $\mathbf{0.86}_{0.00}$ | $0.80_{0.00}$ | $0.80_{0.00}$ |
| Yeast | $\mathbf{0.75}_{0.00}$ | $\mathbf{0.75}_{0.00}$ | $\mathbf{0.75}_{0.00}$ | $0.70_{0.00}$ |
| Chart | $\mathbf{0.89}_{0.00}$ | $0.83_{0.00}$ | $0.72_{0.00}$ | $0.72_{0.00}$ |
| M.F. Digits | $\mathbf{0.82}_{0.05}$ | $0.74_{0.04}$ | $0.67_{0.00}$ | $0.60_{0.01}$ |
| Image Seg. | $0.60_{0.04}$ | $0.65_{0.03}$ | $0.64_{0.00}$ | $\mathbf{0.66}_{0.01}$ |

of the noise in the data, and in the event that they are of roughly equal density, maximising the margin over the microcluster centers has a similar effect to that of minimising the empirical density in a neighbourhood of the corresponding hyperplane separator. This is reminiscent of the soft-margin approach which does enjoy strong convergence properties (Ben-David et al., 2009). In addition, since the optimisation is reinitialised for each value of $\sigma$, we are able to recompute the microclusters by performing the coarse clustering on the projected data with each iteration. This tends to lead to the margins in the microclusters being more closely related to the margins in the full dataset along the optimal projections.

Tables 3.5 and 3.6 report the average and standard deviation of the proposed LMSC as well as the iterative support vector regression approach (Zhang et al., 2009) using both a linear and Gaussian kernel for comparison. Both versions of LMSC, using an orthogonal two dimensional and a one dimensional projection, outperform both versions of the iterative support vector regression in the majority of cases, with substantially higher performance in multiple examples. There is strong evidence that the two dimensional $\text{LMSC}_o$ obtains better quality clustering results than the one dimensional alternative, showing sunstantially higher performance in the vast majority of cases considered.

## 7.3 Summarising Clustering Performance

Thus far we have compared different approaches for standard and normalised spectral clustering and for large margin clustering separately. These separate

**Table 3.6:** $V$-measure results for large margin clustering. Average performance from 30 runs on each dataset, with standard deviation as subscript. The highest average performance in each case is highlighted in bold.

| | $\text{LMSC}_o$ | LMSC | $\text{iSVR}_L$ | $\text{iSVR}_G$ |
|---|---|---|---|---|
| Opt. Digits | $0.70_{0.03}$ | $0.62_{0.04}$ | $\mathbf{0.72}_{0.00}$ | $0.57_{0.00}$ |
| Pen Digits | $\mathbf{0.76}_{0.02}$ | $0.66_{0.03}$ | $0.72_{0.01}$ | $0.73_{0.00}$ |
| Satellite | $\mathbf{0.61}_{0.01}$ | $0.57_{0.03}$ | $0.55_{0.00}$ | $0.55_{0.00}$ |
| Br. Cancer | $0.76_{0.00}$ | $\mathbf{0.78}_{0.00}$ | $0.55_{0.00}$ | $0.72_{0.00}$ |
| Voters | $\mathbf{0.43}_{0.00}$ | $0.38_{0.00}$ | $0.34_{0.00}$ | $0.42_{0.00}$ |
| Dermatology | $\mathbf{0.86}_{0.00}$ | $0.85_{0.00}$ | $0.77_{0.00}$ | $0.74_{0.01}$ |
| Yeast | $0.56_{0.00}$ | $\mathbf{0.58}_{0.00}$ | $0.55_{0.01}$ | $0.53_{0.00}$ |
| Chart | $\mathbf{0.85}_{0.00}$ | $0.78_{0.00}$ | $0.66_{0.00}$ | $0.72_{0.00}$ |
| M.F. Digits | $\mathbf{0.75}_{0.03}$ | $0.69_{0.03}$ | $0.60_{0.00}$ | $0.62_{0.01}$ |
| Image Seg. | $0.60_{0.03}$ | $\mathbf{0.63}_{0.03}$ | $\mathbf{0.63}_{0.00}$ | $0.60_{0.01}$ |

comparisons are important to understand the benefits of the proposed methods, however when considering the clustering problem abstractly it is necessary to compare all methods jointly. It is already clear that no method is uniformly superior to all others, since even within the separate comparisons no method outperformed the rest in every example. We find it important to reiterate the fact that for competing methods based on spectral clustering an extensive search over scaling parameters was performed and the best performance reported, whereas for our method only a simple data driven heuristic was used in every example. Such a search is not possible in practice since the true lables will not be known, and hence the results reported for these methods likely overestimate their true expected performance in practice. What was evident, however, is that the local scaling approach of Zelnik-Manor and Perona (2004) is very effective and yielded the highest performance in roughly half the cases considered.

It is clearly apparent from the performance of the various methods that the clustering problem differs vastly in difficulty across the different datasets considered. To combine the results from the different datasets we standardise them as follows. For each dataset $D$ we compute for each method the relative deviation from the average performance of all methods when applied to $D$. That is, for each method $M_i$ we compute the relative purity,

$$\text{Rel.Purity}(M_i, D) = \frac{\text{Purity}(M_i, D) - \frac{1}{\#\text{Methods}} \sum_{j=1}^{\#\text{Methods}} \text{Purity}(M_j, D)}{\frac{1}{\#\text{Methods}} \sum_{j=1}^{\#\text{Methods}} \text{Purity}(M_j, D)}, \quad (3.31)$$

and similarly for $V$-measure. We can then compare the distributions of the

relative performance measures from all datasets and for all methods. It is clear from Table 3.1 that the competing methods SC, $\text{SC}_{PCA}$ and $\text{SC}_{ICA}$ are not competitive with other methods in general, due to their vastly inferior performance on multiple datasets. Moreover, their performance is sufficiently low to obscure the comparisons between others. These three methods are therefore omitted from this comparison. Figures 3.5 and 3.6 show boxplots of the relative performance measures for all other methods. The additional red dots indicate the mean relative performance measures for each method, and methods are ordered in decreasing order of their means. In the case of purity, all of the proposed methods outperform every method used for comparison, and except for the univariate large margin method, LMSC, the difference between the proposed methods and the methods used for comparison is substantial. In the case of $V$-measure, the same is true except that in this case LMSC is outperformed on average by spectral clustering using the normalised Laplacian applied to the PCA projected data. Notice that the most relevant comparison for LMSC is with $\text{iSVR}_L$ because of their similar objectives. In terms of both purity and $V$-measure, LMSC significantly outperformed the existing large margin clustering method.

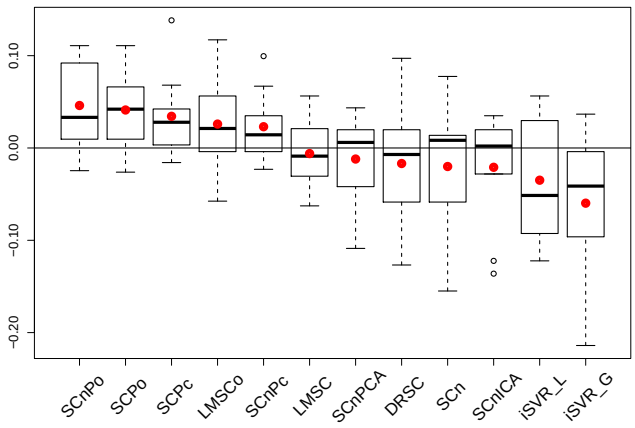
Among the methods used for comparison, it is evident that spectral clustering is capable of outperforming existing large margin clustering methods, provided an appropriate scaling parameter can be determined. Of those spectral clustering variants, PCA projections showed the best overall performance. While the DRSC method (Niu et al., 2011) in some cases showed a substantial improvement over the simpler dimension reduction of PCA, it did not yield consistently higher performance on the datasets considered.

Overall it is apparent that the proposed approach for projection pursuit based on spectral connectivity is highly competitive with existing dimension reduction methods. Moreover, a simple data driven heuristic allowed us to select the important scaling parameter automotically without tuning it for each dataset, as is recommended for the DRSC method (Niu et al., 2011). Among the variants of the proposed approaches, it is evident that while the flexibility of the multivariate projections offered higher performance on average than the corresponding univariate projections, it is only in the case of the large margin separation methods that this improvement is significant beyond the variation from the collection of datasets used for comparison.

## 7.4 Sensitivity Analysis

In this subsection we investigate the sensitivity of the proposed approach to the setting of the important scaling parameter, $\sigma$. In addition we consider the effect on performance of the number of microclusters used in approximating the optimisation surface. For the former we consider the breast cancer, voters, dermatology, yeast and chart datasets as these exhibited very low vari-

**Fig. 3.5:** Box plots of relative purity with additional red dots to indicate means. Methods are ordered with decreasing mean value.




ability in performance and offer more interpretable comparisons. Figures 3.7 and 3.8 show plots of the purity and $V$-measure values for $\sigma$ taking values in $\{0.1\sigma_0, 0.2\sigma_0, 0.5\sigma_0, \sigma_0, 2\sigma_0, 5\sigma_0, 10\sigma_0\}$, where $\sigma_0 = \sqrt{l\lambda_d}N^{-1/5}$ is the value used in the experiments above. There is some variability in the performance for different values, with no clear pattern inditcating that a higher or lower value than the one used is better in general. Importantly there are very few occurences of substantially poorer performance than that obtained with our simply chosen heuristic, and also it is clear that in the majority of cases performance could be improved from what is reported above if an appropriate tuning of $\sigma$ is possible.

To investigate the effect of microclusters on clustering accuracy we simulated datasets from Gaussian mixtures containing 5 components (clusters) in 50 dimensions. This allows us to generate datasets of any desired size. For these experiments 30 sets of parameters for the Gaussian mixtures were generated randomly. In the first case a single dataset of size 1000 was simulated from each set of parameters, and clustering solutions obtained for a number of microclusters, $K$, ranging from 100 to 1000, the final value therefore applying no approximation. Figure 3.9 shows the median and interquartile range of both performance measures for 10 values of $K$. It is evident that aside from $K = 100$, performance is similar for all other values, and so using a small value, say $K = 200$, should be sufficient to obtain a good approximation of the underlying optimisation surface.

In the second, we fix the number of microclusters, $K = 200$, and for each set of parameters simulate datasets with between 1000 and 10 000 observations. In the most extreme case, therefore, the number of microclusters is only 2% of the total number of data. Figure 3.10 shows the corresponding perfor-

**Fig. 3.6:** Box plots of relative $V$-measure with additional red dots to indicate means. Methods are ordered with decreasing mean value.

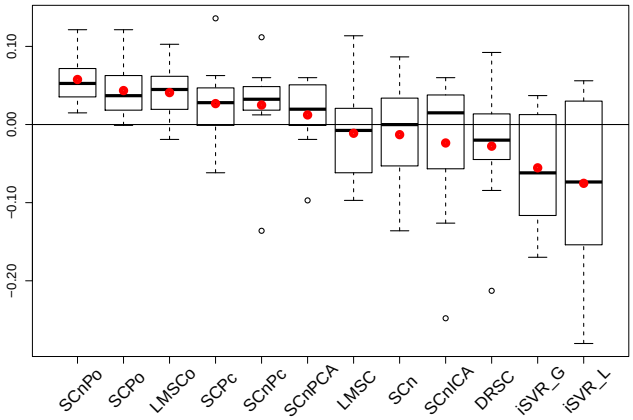
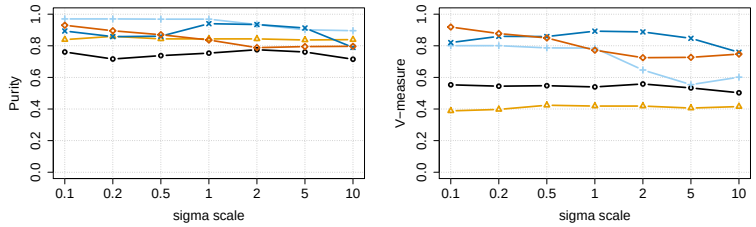
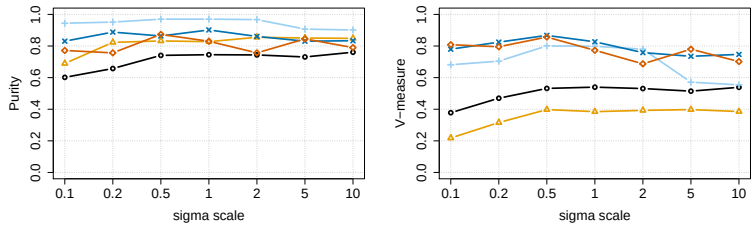



**Fig. 3.7:** Sensitivity analysis for varying $\sigma$. Standard Laplacian. The $x$-axis contains the multiplication factor applied to the default scaling parameter used in the experiments.




(a) $\mathrm{SCP}_o^2$

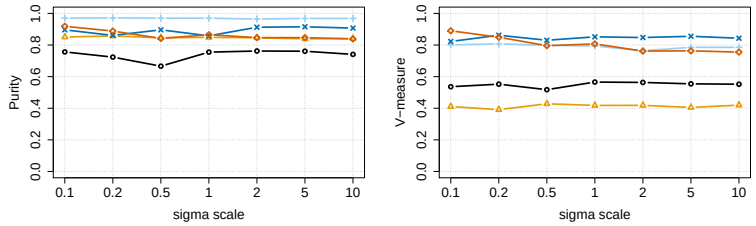



(b) $\mathrm{SCP}_c^2$

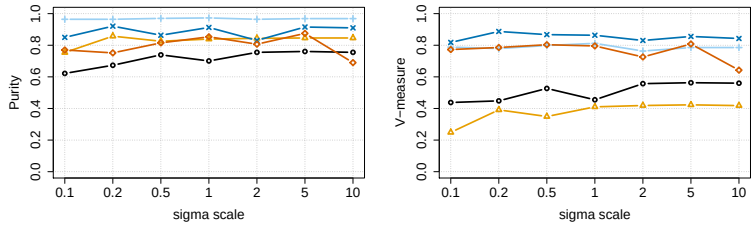
Br. Cancer (−+−), Voters (−△−), Dermatology (−×−), Yeast (−○−), Chart (−◇−)

mance plots, again containing the medians and interquartile ranges. Even for datasets of size 10 000, the coarse approximation of the dataset through 200 microclusters is sufficient to obtain a high quality projection using the proposed approach.

**Fig. 3.8:** Sensitivity analysis for varying $\sigma$. Normalised Laplacian. The $x$-axis contains the multiplication factor applied to the default scaling parameter used in the experiments.




(a) $\mathrm{SC}_n\mathrm{P}_o^2$




(b) $\mathrm{SC}_n\mathrm{P}_c^2$

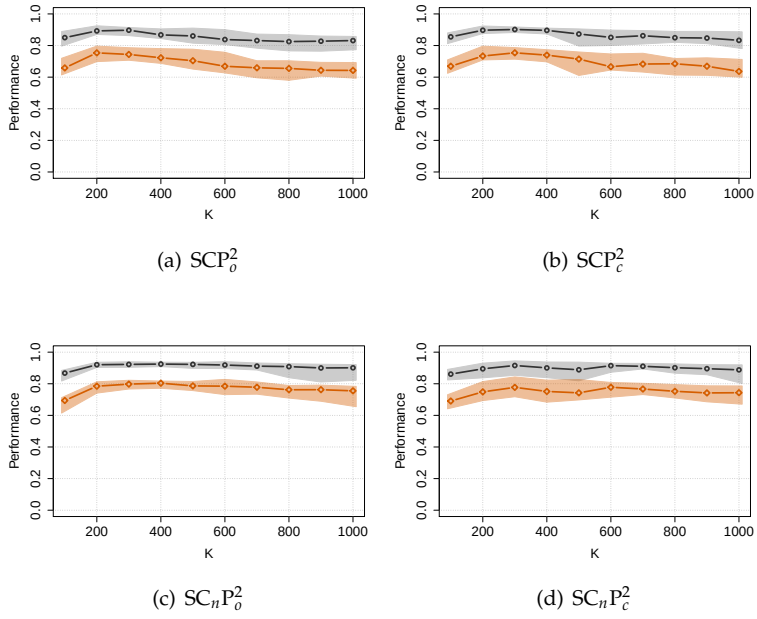
Br. Cancer (–+–), Voters (–△–), Dermatology (–×–), Yeast (–○–), Chart (–◇–)

# 8 Conclusions

We proposed a projection pursuit method for finding the optimal subspace in which to perform a binary partition of unlabelled data. The proposed method optimises the separability of the projected data, as measured by spectral graph theory, by minimising the second smallest eigenvalue of the graph Laplacians. The Lipschitz continuity and differentiability properties of this projection index with respect to the projection matrix were established, which enabled us to apply a generalised gradient descent method to find locally optimal solutions. Compared with existing dimension reduction for spectral clustering, we derive expressions for the gradient of the overall objective and so find solutions within a single generalised gradient descent scheme, with guaranteed convergence to a local optimum. Our experiments suggest that the proposed method substantially outperforms spectral clustering applied to the original data as well as existing dimensionality reduction methods for spectral clustering.

A connection to maximal margin hyperplanes was established, showing that in the univariate case, as the scaling parameter of the similarity function is reduced towards zero, the binary partition of the projected data maximises the linear separability between the two clusters. Implementing our method

**Fig. 3.9:** Sensitivity analysis for varying number of microclusters, $K$. Plots show median and interquartile ranges of performance measures from 30 datasets simulated from 50 dimensional Gaussian mixtures with 5 clusters and 1000 observations.




(a) $\text{SCP}^2_o$ (b) $\text{SCP}^2_c$

(c) $\text{SC}_n\text{P}^2_o$ (d) $\text{SC}_n\text{P}^2_c$

Purity (–○–), $V$-measure (–◇–)

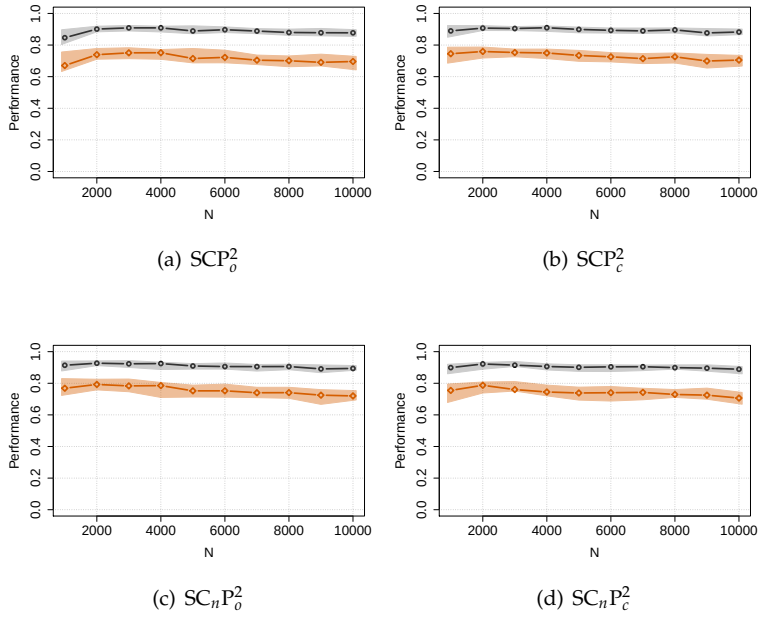
for a shrinking sequence of scaling parameters thus allows us to find large margin separators practically. We found that this approach outperforms state of the art methods for maximum margin clustering on a large collection of datasets.

The computational cost of the proposed projection pursuit method per iteration is $\mathcal{O}(N(N+d(d-1)))$, where $N$ is the number of observations, and $d$ is the dimensionality, which can become prohibitive for large datasets. To ameliorate this an approximation method using microclusters, with provable error bounds is proposed. Our sensitivity analysis, based on clustering performance, indicates that even for relatively few microclusters, the approximation of the optimisation surface is adequate for finding high quality subspaces for clustering.

# Acknowledgements

David Hofmeyr gratefully acknowledges the support of the EPSRC funded EP/H023151/1 STOR-i centre for doctoral training, as well as the Oppenheimer Memorial Trust. We would also like to thank the anonymous review-

**Fig. 3.10:** Sensitivity analysis for fixed number of microclusters, $K=200$, and varying number of data. Plots show median and interquartile ranges of performance measures from datasets simulated from 50 dimensional Gaussian mixtures with 5 clusters and between 1000 and 10 000 observations.




(a) $\mathrm{SCP}_o^2$ (b) $\mathrm{SCP}_c^2$

(c) $\mathrm{SC}_n\mathrm{P}_o^2$ (d) $\mathrm{SC}_n\mathrm{P}_c^2$

Purity (–○–), $V$-measure (–◇–)

ers for the useful comments which enabled us to improve the contents of this work substantially.

# Appendix. Derivatives

In the general case we may consider a set of $K$ microclusters with centers $c_1,...,c_K$ and counts $n_1,...,n_K$. The derivations we provide in this appendix are valid for $n_i=1\ \forall i\in\{1,\dots,K\}$, and so apply to the exact formulation of the problem as well. Let $\boldsymbol{\theta}\in\Theta$ and let $P$ be the repeated projected cluster centers, $P=\{p_1,,p_1,...p_K,p_K\}=\{V(\boldsymbol{\theta})^\top c_1,,V(\boldsymbol{\theta})^\top c_1,\dots,\ V(\boldsymbol{\theta})^\top c_K\}$, where each $V(\boldsymbol{\theta})^\top c_i$ is repeated $n_i$ times. In Section 4 we expressed $D_{\boldsymbol{\theta}}\lambda$ via the chain rule decomposition $D_P\lambda D_v P D_{\boldsymbol{\theta}} v$. The compression of $P$ to the size $K$ non-repeated projected set, $P^C=\{p_1,...,p_K\}$, requires a slight restructuring, as described in Section 6.

We begin with the standard Laplacian, and define $N(\boldsymbol{\theta})$ and $B(\boldsymbol{\theta})$ as in Lemma 15. That is, $N(\boldsymbol{\theta})$ is the diagonal matrix with $i$-th diagonal element equal to $\sum_{j=1}^K n_j s(P^C,i,j)$ and $B(\boldsymbol{\theta})_{i,j}=\sqrt{n_i n_j}s(P^C,i,j)$. The derivative of the

second eigenvalue of the Laplacian of $P$ relies on the corresponding eigenvector, $u$. However, this vector is not explicitly available as we only solve the $K \times K$ eigen-problem of $N(\boldsymbol{\theta}) - B(\boldsymbol{\theta})$. Let $u^C$ be the second eigenvector of $N(\boldsymbol{\theta}) - B(\boldsymbol{\theta})$. As in the proof of Lemma 15 if $i,j$ are such that the $i$-th element of $P$ corresponds to the $j$-th microcluster, then $u_j^C = \sqrt{n_j} u_i$. The derivative of $\lambda_2(N(\boldsymbol{\theta}) - B(\boldsymbol{\theta}))$ with respect to the $i$-th column of $\boldsymbol{\theta}$, and thus equivalently of the second eigenvalue of the Laplacian of $P$, is therefore given by

$$\frac{1}{2}\left(\sum_{j,k}\left(\frac{u_j^C}{\sqrt{n_j}} - \frac{u_k^C}{\sqrt{n_k}}\right)^2 n_j n_k \frac{\partial s(P^C,j,k)}{\partial P_{i1}} \; \ldots \; \sum_{j,k}\left(\frac{u_j^C}{\sqrt{n_j}} - \frac{u_k^C}{\sqrt{n_k}}\right)^2 n_j n_k \frac{\partial s(P^C,j,k)}{\partial P_{iK}}\right) \begin{pmatrix} c_1 & \ldots & c_K \end{pmatrix}^\top D_{\boldsymbol{\theta}_i} V_i, \tag{3.32}$$

where $(c_1 \; \ldots \; c_K)$ is the matrix with $i$-th column $c_i$, $P$ is treated as a $l \times N$ matrix with $i$-th column $p_i$, and $D_{\boldsymbol{\theta}_i} V_i$ is given in Eq. (3.17). Now, the use of the constraint set $\Delta_{\boldsymbol{\theta}}$ and the associated transformation makes a further decomposition convenient. Let $T = \{t_1,\ldots,t_k\} = \{T_{\Delta_{\boldsymbol{\theta}}}(p_1),\ldots,T_{\Delta_{\boldsymbol{\theta}}}(p_K)\}$. We provide expressions for the specific constraint sets used, i.e., $\Delta_{\boldsymbol{\theta}} = \prod_{i=1}^{l}[\mu_{\boldsymbol{\theta}_i} - \beta\sigma_{\boldsymbol{\theta}_i}, \mu_{\boldsymbol{\theta}_i} + \beta\sigma_{\boldsymbol{\theta}_i}]$, where $\mu_{\boldsymbol{\theta}_i} = \frac{1}{N}\sum_{j=1}^{K} n_j P_{ij}$ and $\sigma_{\boldsymbol{\theta}_i}$ is approximated by $\sqrt{\frac{1}{N}\sum_{j=1}^{K} n_j (P_{ij} - \mu_{\boldsymbol{\theta}_i})^2}$. For ease of exposition we assume that each $\mu_{\boldsymbol{\theta}_i}$ is equal to zero, noting that no generality is lost through this simplification since the value of the eigenvalue of the Laplacian is location independent. The data can therefore be centered prior to projection pursuit and the following formulation employed. We can then express the first component of (3.32) as $D_{T_i}\lambda D_{P_i^C} T_i$, where

$$D_{T_i}\lambda = \frac{1}{2}\left(\sum_{j,k}\left(\frac{u_j^C}{\sqrt{n_j}} - \frac{u_k^C}{\sqrt{n_k}}\right)^2 n_j n_k \frac{\partial k\left(\frac{\|t_j - t_k\|}{\sigma}\right)}{\partial T_{i1}} \ldots \right.$$
$$\left. \ldots \sum_{j,k}\left(\frac{u_j^C}{\sqrt{n_j}} - \frac{u_k^C}{\sqrt{n_k}}\right)^2 n_j n_k \frac{\partial k\left(\frac{\|t_j - t_k\|}{\sigma}\right)}{\partial T_{iK}}\right) \tag{3.33}$$

and $D_{P_i^C} T_i$ is the $K \times K$ matrix with

$$(D_{P_i^C} T_i)_{j \neq k} = \begin{cases} \frac{\delta(1-\delta)\beta n_k P_{ik} / N\sigma_{\boldsymbol{\theta}_i}}{(-\beta\sigma_{\boldsymbol{\theta}_i} - P_{ij} + (\delta(1-\delta))^{1/\delta})^{\delta}}, & P_{ij} < -\beta\sigma_{\boldsymbol{\theta}_i} \\ \frac{\beta n_k P_{ik}}{N\sigma_{\boldsymbol{\theta}_i}}, & -\beta\sigma_{\boldsymbol{\theta}_i} \leq P_{ij} \leq \beta\sigma_{\boldsymbol{\theta}_i} \\ \frac{2\beta n_k P_{ik}}{N\sigma_{\boldsymbol{\theta}_i}} - \frac{\delta(1-\delta)\beta n_k P_{ik} / N\sigma_{\boldsymbol{\theta}_i}}{(P_{ij} - \beta\sigma_{\boldsymbol{\theta}_i} + (\delta(1-\delta))^{1/\delta})^{\delta}}, & P_{ij} > \beta\sigma_{\boldsymbol{\theta}_i} \end{cases} \tag{3.34}$$

$$(D_{P_i^C} T_i)_{jj} = \begin{cases} \frac{\delta(1-\delta)(1+\beta n_j P_{ij} / N\sigma_{\boldsymbol{\theta}_i})}{(-\beta\sigma_{\boldsymbol{\theta}_i} - P_{ij} + (\delta(1-\delta))^{1/\delta})^{\delta}}, & P_{ij} < -\beta\sigma_{\boldsymbol{\theta}_i} \\ 1 + \frac{\beta n_j P_{ij}}{N\sigma_{\boldsymbol{\theta}_i}}, & -\beta\sigma_{\boldsymbol{\theta}_i} \leq P_{ij} \leq \beta\sigma_{\boldsymbol{\theta}_i} \\ \frac{2\beta n_j P_{ij}}{N\sigma_{\boldsymbol{\theta}_i}} + \frac{\delta(1-\delta)(1-\beta n_j P_{ij} / N\sigma_{\boldsymbol{\theta}_i})}{(P_{ij} - \beta\sigma_{\boldsymbol{\theta}_i} + (\delta(1-\delta))^{1/\delta})^{\delta}}, & P_{ij} > \beta\sigma_{\boldsymbol{\theta}_i}. \end{cases} \tag{3.35}$$

In the above we have used the lower case $t_j$ to denote the $j$-th element of the transformed projected dataset, where the upper case $T_{ij}$ denotes the $ij$-th element of the $l \times N$ matrix with $j$-th column equal to $t_j$. The benefit of this further decomposition lies in the fact that the majority of terms in the sums in (3.33) are zero. In fact,

$$\frac{1}{2}\sum_{j,k}\left(\frac{u_j^C}{\sqrt{n_j}} - \frac{u_k^C}{\sqrt{n_k}}\right) n_j n_k \frac{\partial k\left(\frac{\|t_j - t_k\|}{\sigma}\right)}{\partial T_{im}} = \sum_{j \neq m}\left(\frac{u_j^C}{\sqrt{n_j}} - \frac{u_m^C}{\sqrt{n_m}}\right) n_j n_m \frac{\partial k\left(\frac{\|t_j - t_m\|}{\sigma}\right)}{\partial T_{im}}, \tag{3.36}$$

where for the function given in Eq. (3.30) we have,

$$\frac{\partial k\left(\frac{\|t_j - t_m\|}{\sigma}\right)}{\partial T_{im}} = \frac{T_{ij} - T_{im}}{\sigma^2 \alpha}\left(\frac{\|t_j - t_m\|}{\sigma\alpha} + 1\right)^{\alpha - 1} \exp\left(\frac{\|t_j - t_m\|}{\sigma}\right). \tag{3.37}$$

For the normalised Laplacian, the reduced $K \times K$ eigenproblem has precisely the same form as the original $N \times N$ problem, with the only difference being the introduction of the factors $n_j n_k$. In particular, the second eigenvalue of the normalised Laplacian of $P$ is equal to the second eigenvalue of the Laplacian of the graph of $P^C$ with similarities given by $n_j n_k s(P^C, j, k)$. With the derivation in Section 4 we can see that the corresponding derivative is as for the standard Laplacian, except that the coefficients $(u_j^C/\sqrt{n_j} - u_k^C/\sqrt{n_k})^2 n_j n_k$ in Eq. (3.36) are replaced with $(u_j^C/\sqrt{d_j} - u_k^C/\sqrt{d_k})^2 - \lambda((u_j^C)^2/d_j + (u_k^C)^2/d_k)$, where $\lambda$ is the second eigenvalue of the normalised Laplacian of $P^C$, $u^C$ is the corresponding eigenvector and $d_j$ is the degree of the $j$-th element of $P^C$.

# B. Semi-supervised Spectral Connectivity Projection Pursuit


## Abstract

*We propose a projection pursuit method based on semi-supervised spectral connectivity. The projection index is given by the second eigenvalue of the graph Laplacian of the projected data. An incomplete label set is used to modify pairwise similarities between data in such a way that penalises projections which do not admit a separation of the classes (within the training data). We show that the global optimum of the proposed problem converges to the Transductive Support Vector Machine solution, as the scaling parameter is reduced to zero. We evaluate the performance of the proposed method on benchmark data sets.*


## 1 Introduction

Projection pursuit is a data driven optimisation problem, defined as follows. For a data set $X=\{x_1,...,x_N\}$ in $\mathbb{R}^d$, optimise over the set of unit-norm vectors, $\{v\in\mathbb{R}^d \,|\, \|v\|=1\}$, a predefined measure of quality of the projected data set, $v\cdot X=\{v\cdot x_1,...,v\cdot x_N\}$. These unit-norm vectors are referred to as *projection vectors*, or simply *projections*, and the measured quality of the projected data set is referred to as the *projection index*.

Semi-supervised classification refers to the construction of a classifier, i.e., a map from the data space to a set of class labels, using a set of “training" data whose true class labels are known as well as a set of “test" data whose labels are to be inferred from the classifier. In supervised classification, on the other hand, only the training data and associated labels are used in the construction of the classifier. In using a classifier for class prediction there is an implicit assumption that the distribution of the test data resemble somewhat that of

the training data, and therefore utilising spatial distribution information of the test data might be useful in better predicting their class memberships.

There are numerous approaches to the problem of semi-supervised classification, see Chapelle et al. (2006b) for a recent review of standard methods. Underlying many of these methods is the so-called *cluster assumption*; that different classes manifest single clusters, and so can be separated by data sparse regions. Of these methods, arguably the most popular are those based on Transductive Support Vector Machines (TSVMs). The original TSVM problem (Vapnik and Sterin, 1977) is formulated as follows. Given labelled data $\mathcal{X}^L = \{x_1,...,x_l\}$ with labels $Y^L \in \{-1,+1\}^l$ and unlabelled data $\mathcal{X}^U = \{x_{l+1},...,x_{l+u}\}$, find $Y^U \in \{-1,+1\}^u$ s.t. a Support Vector Machine (SVM) classifier trained on $\mathcal{X}^L \cup \mathcal{X}^U, Y^L \cup Y^U$ achieves the largest margin. This is a combinatorial problem and difficult to solve for any reasonably sized data set. Approximations based on local search heuristics (Joachims, 1999) or continuous relaxations are solved instead (Chapelle et al., 2006a), but these are highly susceptible to local optima.

Accepting the cluster assumption leads us to consider semi-supervised methodology that is consistent with popular notions of clusterability. Spectral clustering has become increasingly popular due to its strong performance in a variety of application areas (von Luxburg, 2007). In spectral clustering, clusters are defined as strongly connected components of a graph defined over the data in which edge weights assume values equal to the similarity between the adjacent vertices. A continuous relaxation of the minimum ratio cut problem is solved, using the eigenvectors of the graph *Laplacian* matrix. Recently a projection pursuit method for learning the projection along which a set of data are minimally connected under this cluster definition was proposed (Hofmeyr et al., 2015). The authors show that the projection along which the data are minimally connected converges to the vector normal to the largest margin hyperplane through the data, as the scaling parameter is reduced to zero. In this paper we extend this work to include partial supervision via an incomplete set of labels, as in semi-supervised classification. We show that if the labels are incorporated in a specific way, then the convergence result of Hofmeyr et al. (2015) extends to the semi-supervised setting, i.e., the optimal projection for semi-supervised spectral connectivity converges to the vector normal to the optimal TSVM hyperplane. This establishes an asymptotic connection between our proposed method and popular semi-supervised classification methods.

The Remainder of this paper is organised as follows. In Section 2 we give a brief introduction to spectral clustering. In Section 3 we introduce the proposed methodology. We provide experimental results in Section 4 and give some concluding remarks in Section 5.

## 2 Spectral Clustering

In this section we give a very brief introduction to spectral clustering, with particular attention to binary partitioning. For a thorough introduction, the reader is directed to von Luxburg (2007). Let $X=\{x_1,...,x_N\}$ be a given data set. The ratio cut problem is defined as follows,

$$\min_{C\subset X} \sum_{\substack{i,j:x_i\in C\\ x_j\notin C}} \text{similarity}(x_i,x_j)\left(\frac{1}{|C|}+\frac{1}{|X\setminus C|}\right), \tag{3.38}$$

where $|\cdot|$ is the cardinality operator. The similarity between two points is generally determined by a non-negative, decreasing function, $k:\mathbb{R}\to\mathbb{R}^+$, of the distance between them. The above problem can be formulated in terms of the *Laplacian matrix*, $L:=D-A$, where $A$ is the *affinity matrix*, with $A_{ij}=$ similarity$(x_i,x_j)$, and the diagonal matrix $D$ is called the *degree matrix*, with $D_{ii}=\sum_{j=1}^N A_{ij}$. For $C\subset X$ define the vector $f^C\in\mathbb{R}^N$ such that,

$$f_i^C=\begin{cases}\sqrt{|X\setminus C|/|C|}, & x_i\in C\\ -\sqrt{|C|/|X\setminus C|}, & x\notin C.\end{cases} \tag{3.39}$$

Then (3.38) can be written as,

$$\min_{C\subset X} f^C\cdot Lf^C \text{ s.t. } f^C\perp\mathbf{1}, \|f^C\|=\sqrt{N}. \tag{3.40}$$

The above problem is NP-hard (Wagner and Wagner, 1993), and so instead a continuous relaxation, in which the discreteness condition on the vector $f^C$ (3.39) is relaxed, is solved instead. The solution to the relaxed problem is given by the second eigenvector of $L$. The second eigenvalue therefore provides a lower bound for the normalised aggregated similarities from pairs of data belonging to different elements of the optimal partition, arising from the solution to (3.38). While the optimal solution to the relaxed problem can induce partitions which are arbitrarily far from the optimal solution to (3.40) (Guattery and Miller, 1998), in many practical applications the solutions tend to be similar. Furthermore, in the univariate case the resulting partition tends to be very similar to the true optimum. Obtaining the projection which minimises the second eigenvalue of the Laplacian therefore tends to result in projections along which the optimal partition arising from (3.38) is loosely connected, i.e., the elements of the partition are well separated.

## 3 Methodology

In this section we provide details for how to find locally optimal projections for bi-partitioning based on semi-supervised spectral clustering. We formu-

late the problem as a projection pursuit, where the projection index is given by the second eigenvalue of the Laplacian of the projected data. We assume we have a set of $N$ data, $l$ of which have labels in $\{-1,+1\}$ which define their class membership, and $u=N-l$ are unlabelled.

Following the method of Hofmeyr et al. (2015) we formulate the projection vectors in terms of their polar coordinates. Let $\Theta=[0,\pi)^{d-2}\times[0,2\pi)$ and for $\boldsymbol{\theta}\in\Theta$, define the projection vector $v(\boldsymbol{\theta})$ by,

$$v(\boldsymbol{\theta})_i=\begin{cases}\cos(\boldsymbol{\theta}_i)\prod_{j=1}^{i-1}\sin(\boldsymbol{\theta}_j), & i=1,\ldots,d-1\\ \prod_{j=1}^{d-1}\sin(\boldsymbol{\theta}_j), & i=d.\end{cases} \tag{3.41}$$

We will use the following notation. For $\boldsymbol{\theta}\in\Theta$, we write $L(\boldsymbol{\theta})$ for the Laplacian of the projected data set $P(\boldsymbol{\theta}):=v(\boldsymbol{\theta})\cdot X$, and $\lambda_2(\boldsymbol{\theta})$ for the second eigenvalue of $L(\boldsymbol{\theta})$. If the similarities between pairs of projected data are Lipschitz and continuously differentiable functions of $\boldsymbol{\theta}$, then $\lambda_2(\boldsymbol{\theta})$ is Lipschitz and continuously differentiable almost everywhere (Hofmeyr et al., 2015). This allows us to find local optima via generalised gradient descent. The derivative of $\lambda_2(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ can be decomposed using the chain rule into the product $D_{P(\boldsymbol{\theta})}\lambda_2(\boldsymbol{\theta})D_{v(\boldsymbol{\theta})}P(\boldsymbol{\theta})D_{\boldsymbol{\theta}}v(\boldsymbol{\theta})$, where $D_{\cdot}\cdot$ is the differential operator. Derivations of the following can be found in Hofmeyr et al. (2015). If $\lambda_2(\boldsymbol{\theta})$ is a simple eigenvalue, then

$$\frac{\partial\lambda_2(\boldsymbol{\theta})}{\partial P(\boldsymbol{\theta})_k}=\frac{1}{2}\sum_{ij}(u_i-u_j)^2\frac{\partial A(\boldsymbol{\theta})_{ij}}{\partial P(\boldsymbol{\theta})_k}, \tag{3.42}$$

where $A(\boldsymbol{\theta})$ is the affinity matrix of the projected data set.

The matrix $D_{v(\boldsymbol{\theta})}P(\boldsymbol{\theta})\in\mathbb{R}^{N\times d}$ has $i$-th row $x_i^\top$, and the matrix $D_{\boldsymbol{\theta}}v(\boldsymbol{\theta})\in\mathbb{R}^{d\times(d-1)}$ has $i,j$-th element,

$$\frac{\partial v(\boldsymbol{\theta})_i}{\partial\boldsymbol{\theta}_j}=\begin{cases}0, & i<j\\ -\sin(\boldsymbol{\theta}_j)\prod_{k=1}^{j-1}\sin(\boldsymbol{\theta}_k), & i=j<d\\ \cos(\boldsymbol{\theta}_j)\cos(\boldsymbol{\theta}_i)\prod_{k<i,k\neq j}\sin(\boldsymbol{\theta}_k), & j<i<d\\ \cos(\boldsymbol{\theta}_j)\prod_{k\neq j}\sin(\boldsymbol{\theta}_k), & i=d.\end{cases} \tag{3.43}$$

What remains is to address the similarity function. Within spectral clustering, pairwise similarities between data are defined by a decreasing function of the distance between them. That is, similarity$(x_i,x_j)=k(d(x_i,x_j))$, where $k:\mathbb{R}^+\to\mathbb{R}^+$ is decreasing and $d(\cdot,\cdot)$ is a metric. The function $k$ often takes the form of a kernel, and we use the Gaussian kernel, given by

$$k(x)=\exp\left(-\frac{x^2}{2}\right). \tag{3.44}$$

Controlling the balance of the partition, i.e., the relative sizes of the resulting clusters, is an important feature in semi-supervised classification (Chapelle et al., 2006b). We control this balance in the same way as in Hofmeyr et al. (2015) within the metric $d(\cdot,\cdot)$. In particular, for a univariate data set $P$,

$$d(P_i,P_j):=\frac{|T(P_i)-T(P_j)|}{\sigma}, \tag{3.45}$$

where $\sigma>0$ is the *scaling parameter*, and the function $T$ is used to decrease the distance between points lying outside a chosen interval $[m,M]$ and other points, to induce more balanced splits.

$$T(x):=\begin{cases} -\delta\left(m-x+(\delta(1-\delta))^{\frac{1}{\delta}}\right)^{1-\delta} \\ \qquad\qquad +\,\delta(\delta(1-\delta))^{\frac{1-\delta}{\delta}}, & x<m \\ \qquad\qquad x-m, & x\in[m,M] \\ \delta\left(x-M+(\delta(1-\delta))^{\frac{1}{\delta}}\right)^{1-\delta} \\ \qquad\qquad -\,\delta(\delta(1-\delta))^{\frac{1-\delta}{\delta}}, & x>M \end{cases} \tag{3.46}$$

We set $m=\mu_P-\beta\sigma_P$ and $M=\mu_P+\beta\sigma_P$, where $\mu_P$ and $\sigma_P$ are the mean and standard deviation of $P$ respectively and $\beta$ is used to control the width of the interval $[m,M]$. See Hofmeyr et al. (2015) for details on the effect of the function $T$. The parameter $\delta$ takes values in $(0,0.5]$, with smaller values increasing the similarity of points outside $[m,M]$ with other points to a greater degree. The value of this parameter does not play a huge role in performance (Hofmeyr et al., 2015).

So far we have not discussed how we incorporate label information into this framework. The ratio cut is inherently a connectivity based partitioning method, and though the spectral clustering solution is a relaxation, its behaviour mimics this connectivity property. We wish to use these labels to modify the pairwise similarities in such a way that projections which do not admit a separation of the (known) classes are penalised. By this we mean projections for which $\exists i,j$ s.t. $y_i=+1,y_j=-1$ but $P(\boldsymbol{\theta})_i<P(\boldsymbol{\theta})_j$, i.e., the positive labelled projections do not all lie above all negative labelled projections. This can be achieved by ensuring that along any such projection there is a chain of high pairwise similarities connecting the entire projected data set, as follows. For brevity we temporarily drop the notational dependence on $\boldsymbol{\theta}$. Define,

$$A_{ij}=\begin{cases} k(d(P_i,P_j)), & x_i,x_j\in\mathcal{X}^U \\ k(d(P_i,P_j))+\left((P_i-P_j)^+\right)^{1+\epsilon}, & y_i=-1,y_j\neq-1 \\ k(d(P_i,P_j))+\left((P_j-P_i)^+\right)^{1+\epsilon}, & y_i=+1,y_j\neq+1 \\ H, & y_i=y_j, \end{cases} \tag{3.47}$$

where $(x)^+ = \max\{0,x\}$ and $H \geq 1$ is a chosen constant which affects the influence of the known labels. We use $y_i \neq +1$ (resp. $y_i \neq -1$) to mean that $y_i = -1$ or $x_i$ is unlabelled (resp. $y_i = +1$ or $x_i$ is unlabelled). The exponent $1+\epsilon$, where $\epsilon$ is some small positive number, ensures continuous differentiability of the associated additions while having a practical influence much like the hinge loss function. We'll refer to these additions as penalties. With the above formulation we can derive expressions for $\partial A_{ij}/\partial P_k$ for all $i,j,k$. If $x_i, x_j \in \mathcal{X}^U$ then,

$$\begin{aligned}\frac{\partial A_{ij}}{\partial P_k} &= \frac{\partial A_{ij}}{\partial T(P_j)}\frac{\partial T(P_j)}{\partial P_k} = \frac{\partial k(d(P_i,P_j))}{\partial T(P_j)}\frac{\partial T(P_j)}{\partial P_k} \\ &= \frac{T(P_i)-T(P_j)}{\sigma^2}\exp\left(-\frac{d(P_i,P_j)^2}{2}\right)\frac{\partial T(P_j)}{\partial P_k}.\end{aligned}$$

If $y_i = +1, y_j \neq +1$, then if $P_i > P_j$ we have the same formulation as above. Otherwise,

$$\begin{aligned}\frac{\partial A_{ij}}{\partial P_k} &= \frac{\partial k(d(P_i,P_j))}{\partial T(P_j)}\frac{\partial T(P_j)}{\partial P_k} + \frac{\partial (P_j-P_i)^{1+\epsilon}}{\partial P_k} \\ \frac{\partial (P_j-P_i)^{1+\epsilon}}{\partial P_k} &= \begin{cases}(1+\epsilon)(P_j-P_i)^{\epsilon}, & k=j \\ -(1+\epsilon)(P_j-P_i)^{\epsilon}, & k=i \\ 0, & \text{otherwise.}\end{cases}\end{aligned}$$

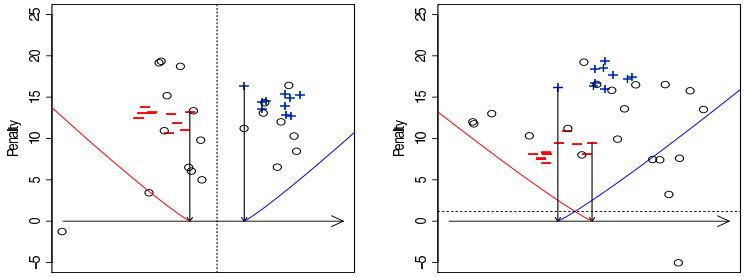
The formulation for $y_i = -1, y_j \neq -1$ is analogous. Finally, for $j \neq k$

$$\frac{\partial T(P_j)}{\partial P_k} = \begin{cases} -\frac{\delta(1-\delta)(1-\frac{\beta(P_k-\mu_P)(N-1)}{N\sigma_P})}{N(\mu_P-\beta\sigma_P-P_j+(\delta(1-\delta))^{1/\delta})^{\delta}}, & P_j < m \\ \frac{1}{N}\left(\frac{\beta(P_k-\mu_P)(N-1)}{N\sigma_P}-1\right), & m \leq P_j \leq M \\ \frac{\delta(1-\delta)(1-\frac{\beta(P_k-\mu_P)(N-1)}{N\sigma_P})}{N(P_j-\mu_P-\beta\sigma_P+(\delta(1-\delta))^{1/\delta})^{\delta}} & \\ \quad +\frac{2\beta(P_k-\mu_P)(N-1)}{N^2\sigma_P}, & P_j > M, \end{cases}$$

and if $j = k$,

$$\frac{\partial T(P_j)}{\partial P_j} = \begin{cases} -\frac{\delta(1-\delta)(1-N-\frac{\beta(P_j-\mu_P)(N-1)}{N\sigma_P})}{N(\mu_P-\beta\sigma_P-P_j+(\delta(1-\delta))^{1/\delta})^{\delta}}, & P_j < m \\ 1-\frac{1}{N}\left(\frac{\beta(P_j-\mu_P)(N-1)}{N\sigma_P}-1\right), & M \leq P_j \leq M \\ \frac{\delta(1-\delta)(N-1-\frac{\beta(P_j-\mu_P)(N-1)}{N\sigma_P})}{N(P_j-\mu_P-\beta\sigma_P+\delta(1-\delta))^{\delta}} & \\ \quad +\frac{2\beta(P_j-\mu_P)(N-n_j)}{N^2\sigma_P}, & P_j > M. \end{cases}$$

**Fig. 3.11:** Two projections, one admitting a separation of the classes (Left) and the other not (Right).




+'s and -'s indicate labelled data, while unlabelled data are indicated by ○'s. The horizontal arrow represents the projection direction. Vertical arrows indicate the maximum projected datum from class $-1$, say $p^-$, and the minimum projected datum from class $+$, $p^+$. The red and blue lines indicate the penalties induced by these two projections. Each unlabelled datum is connected to either $p^-$ or $p^+$ by the maximum of these two lines. In the right panel, the minimum of the maximum of these two lines is indicated by the horizontal dashed line, say with value $\alpha > 0$. The points $p^-$ and $p^+$ are also connected by at least this value, and are connected to their respective classes with similarity $H$. There is therefore a chain connecting all data with minimum similarity $\min\{\alpha, H\}$. In the left panel, partitioning the data above/below the vertical dashed line leads to a ratio cut with no penalties included.

We can thus evaluate the derivative of $\lambda_2(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ provided it is simple. We use the non-smooth optimisation method described in Hofmeyr et al. (2015) to find locally optimal solutions, which alternates between a naive application of gradient descent, in which the simplicity of $\lambda_2(\boldsymbol{\theta})$ is assumed to hold everywhere, and a descent step based on the directional derivative of $\lambda_2(\boldsymbol{\theta})$ when it is not simple. We found that the directional step was not required in any of our experiments, and so omit its formulation. Interested readers are referred to the paper (Hofmeyr et al., 2015).

While it is perhaps counterintuitive to increase the similarity between data known to belong to different classes, as in (3.47), this formulation ensures that projections which do not admit a separation of the known classes are penalised, while those which do admit such a separation allow for partitions which do not include any of the penalised similarities in the ratio cut computation. Figure 3.11 illustrates this fact. The following lemma shows that projections admitting a separation of the classes have a lower spectral connectivity than those which do not, for small values of the scaling parameter $\sigma$.

**Lemma 17** *Let $k$ be non-increasing, Lipschitz and satisfy $k(x) \in o(x^{-(1+\epsilon)})$ as $x \to \infty$. Let $\boldsymbol{\theta}_1$ be such that $\min\{P(\boldsymbol{\theta}_1)_i | y_i = +1\} > \max\{P(\boldsymbol{\theta}_1)_j | y_j = -1\}$. Then $\exists \sigma' > 0$ s.t. for any $0 < \sigma < \sigma'$ and $\boldsymbol{\theta}_2$ s.t. $\min\{P(\boldsymbol{\theta}_2)_i | y_i = +1\} \leq \max\{P(\boldsymbol{\theta}_2)_j | y_j = -1\}$ we have $\lambda_2(\boldsymbol{\theta}_1) < \lambda_2(\boldsymbol{\theta}_2)$.*

We see then that the formulation given in (3.47) does indeed induce a penalty, and the penalty forces the optimal solution to admit a separation of the classes if $\sigma$ is small enough, assuming that the classes can be separated. We can extend the above result to show that the optimal projection converges to the vector normal to the TSVM solution, as $\sigma \to 0^+$. We discuss the result in the context of a constrained solution, i.e., one which induces a balanced partition by intersecting a scaled covariance ellipsoid. The result holds for all values of $\beta$, and so setting $\beta$ arbitrarily large proves the result relative to the original TSVM problem.

**Lemma 18** *Suppose $\exists v \in \mathbb{R}^d, b \in \mathbb{R}$ s.t. $y_i(v \cdot x_i - b) > 0$ for all $i \in \{1, \ldots, l\}$. Let $k : \mathbb{R}^+ \to \mathbb{R}^+$ satisfy the following:*

1. *$k$ is non-increasing*
2. *$k$ is Lipschitz*
3. *$\lim_{x \to \infty} k(x+\epsilon)/k(x) = 0$ for all $\epsilon > 0$*
4. *$k(x) \in o(x^{-(1+\epsilon)})$ as $x \to \infty$.*

*For $\sigma, \delta > 0$ define $\boldsymbol{\theta}_{\sigma,\delta} = argmin_{\boldsymbol{\theta} \in \Theta} \lambda_2(\boldsymbol{\theta}, \sigma, \delta)$, where $\lambda_2(\boldsymbol{\theta}, \sigma, \delta)$ is the same as $\lambda_2(\boldsymbol{\theta})$ from before but with an explicit dependence on $\sigma$ and $\delta$. Let $(v^\star, b^\star)$ define the largest margin hyperplane which correctly classifies all labelled data and satisfies $\bar{m} < b^\star < \bar{M}$, where $\bar{m}$ lies halfway between $\mu_{v^\star \cdot X} - \beta\sigma_{v^\star \cdot X}$ and the smallest element of $P(\boldsymbol{\theta}^\star)$ above $\mu_{v^\star \cdot X} - \beta\sigma_{v^\star \cdot X}$ and similarly $\bar{M}$ lies halfway between $\mu_{v^\star \cdot X} + \beta\sigma_{v^\star \cdot X}$ and the largest element of $P(\boldsymbol{\theta}^\star)$ below $\mu_{v^\star \cdot X} + \beta\sigma_{v^\star \cdot X}$. Then,*

$$\lim_{\sigma, \delta \to 0^+} v(\boldsymbol{\theta}_{\sigma,\delta}) = v^\star$$

The distinction between $\bar{m}$ and $\mu_{v^\star \cdot X} - \beta\sigma_{v^\star \cdot X}$, and analogously for $\bar{M}$, is not of much practical concern, but is important for proving the associated theory. The reader may refer to Hofmeyr et al. (2015) for additional discussion.

## 3.1 Computational Complexity

It is clear to see that the proposed method has the same computational complexity as the corresponding unsupervised projection pursuit given by Hofmeyr et al. (2015), since the difference between the semi-supervised approach described herein and the unsupervised problem lies only in the modification

of pairwise similarities, which does not affect the computational complexity. Hofmeyr et al. (2015) have shown the the computational cost for each evaluation of the projection index is $\mathcal{O}(N(N+d))$, and each gradient computation has complexity $\mathcal{O}(N(N+d(d-1)))$. The total complexity of the method is therefore $\mathcal{O}(N(N+d(d-1))t)$, where $t$ is the number of steps in the gradient descent.

# 4 Experiments

In this section we evaluate the performance of the proposed method, which we will refer to as Semi-Supvervised Spectral Connectivity Projection Pursuit ($S^3CP^2$), on classification data sets taken from the UCI Machine Learning Repository (UCIMLR).[6] and benchmark data sets for semi-supervised classification.[7] For each UCIMLR data set we generated 30 sets of labels, 10 for each of the sizes 2%, 10% and 25% the total number of data. The label sets were generated uniformly at random, however sets which did not contain at least one label from each class were rejected and replaced with another. For the data sets taken from Chapelle et al. (2006b) we used the same 24 label sets, 12 for each of 10 and 100 labels. The UCIMLR data sets considered were: Mammographic Mass (Mam.): Distinguish benign from malignant masses found during mammography screening. Voters (Vote.): Determine political party affiliation from votes made by US congress members. Breast Cancer (Canc.): Distinguish benign from malignant tumour masses, given physical characteristics. Ionosphere (Iono.): Distinguish radio signals which show evidence of structure in the ionosphere from those which don't. Parkinsons (Park.): A range of biomedical voice measurements used to determine the presence of Parkinson's disease. The data sets taken from Chapelle et al. (2006b) were: g241c: A mixture of 2 Gaussian distributions for which the cluster assumption holds. g241d: A 4 component Gaussian mixture in which cluster structure is misleading for class membership. Digit1: Artificially generated images of the digit "1" varied by translation, ratation, line thickness and length, obscured to increase difficulty. Class boundary at the 0 rotation angle. The cluster assumption holds. USPS: U.S. Postal Service handwritten digits. Digits "2" and "5" form class $+1$ with the remaining digits forming class $-1$. The images are obscured using the same transformation as for Digit1. BCI: A brain-computer interface experiment in which a subject imagined movements with their right, class $+1$, and left hand, class $-1$. The features are EEG readings taken during the experiment.

[6] M. Lichman. UCI Machine Learning Respository `http://archive.ics.uci.edu/ml`. Irvine, CA. University of California, School of Information and Computer Science. 2013

[7] A selection of data sets used in Chapelle et al. (2006b) `http://olivier.chapelle.cc/ssl-book/benchmarks.html`.

We compare performance with a standard SVM[8] trained only on the labelled data, and Semi-Supervised SVM ($S^3$VM)[9]. We use the linear kernel for SVM and $S^3$VM since this provides the most meaningful comparison with our proposed method. Non-linear separators are possible for our method by an explicit embedding of the data within the kernel space, as in Chapelle et al. (2006a). For SVM we use the default parameters given in the package. For $S^3$VM we initialised using SVM and built classifiers for each of 5 values of $C^\star$, which determines the penalty for the unlabelled data violating the large margin, in each experiment using the UCIMLR data sets. We then report the highest average performance of those 5 for each case. For the data sets arising from Chapelle et al. (2006b) we use the results presented there. For $S^3CP^2$ we initialise the projection pursuit using the SVM solution. For the UCIMLR data sets we set $\beta = 1.5$, $\sigma = 0.3nn_{0.99}/\sqrt{d}$ where $nn_{0.99}$ is the 99th centile of the nearest neighbour distances in the data and $d$ is the dimension, and $H = u/l$, the ratio of the number of unlabelled and labelled data. The data sets taken from Chapelle et al. (2006b) are more challenging, and we selected $\sigma$ and $\beta$ from the sets $\{0.1nn_{0.99}/\sqrt{d}, 0.5nn_{0.99}/\sqrt{d}\}$ and $\{0.5, 1.5\}$ respectively using cross validation.

Tables 3.7 and 3.8 report the average classification accuracy over the different label sets. The highest average performance is highlighted in bold. The proposed method achieves the highest average performance in roughly half of the cases considered, and is competitive with the highest performing method in almost all. Importantly the method achieved higher performance than linear $S^3$VM in the majority of applications.

Parameter tuning is a very difficult task in semi-supervised classification (Chapelle et al., 2006b), and though our method contains numerous parameters, the majority do not play a significant role in its performance. The parameter $\sigma$, and to a lesser extent $\beta$, plays the most crucial role in the performance of the method, and determining an appropriate value is necessary for the successful application of the method. We used a simple reference rule which has worked well on many of the examples considered, but believe considerable improvements can be made if a more principled tuning method is employed.

# 5 Conclusions

We propose a new method for semi-supervised classification, which is based on learning the optimal univariate subspace to perform a binary partition of the projected data set using semi-supervised spectral clustering. The labels

[8] We use the `R` package `e1071`, which implements the `libSVM` library

[9] We use the SVM-light implementation of T. Joachims available at `http://svmlight.joachims.org/`

**Table 3.7:** UCIMLR Classification Data Sets. Average Accuracy (%) over 10 Splits.

| | Mam. | Vote. | Canc. | Iono. | Park. |
|---|---|---|---|---|---|
| | 2% Labelled Examples | | | | |
| $S^3CP^2$ | 79.62 | 84.53 | **96.18** | 66.44 | **74.71** |
| SVM | **80.58** | 84.48 | 91.59 | 70.82 | 74.14 |
| $S^3VM$ | 79.91 | **86.90** | 95.84 | **72.83** | 65.50 |
| | 10% Labelled Examples | | | | |
| $S^3CP^2$ | **81.64** | **90.74** | 96.04 | **85.71** | 79.94 |
| SVM | 80.77 | 89.00 | 95.26 | 80.29 | **80.23** |
| $S^3VM$ | 81.61 | 89.74 | **96.66** | 85.62 | 71.09 |
| | 25% Labelled Examples | | | | |
| $S^3CP^2$ | 82.54 | **90.34** | **96.49** | **87.18** | 80.68 |
| SVM | 82.22 | 89.33 | 96.26 | 84.98 | **82.60** |
| $S^3VM$ | **83.04** | 89.20 | 96.45 | 86.92 | 74.93 |

**Table 3.8:** SSC Benchmark Data Sets. Average Accuracy (%) over 12 Splits.

| | g241c | g241d | Digit1 | USPS | BCI |
|---|---|---|---|---|---|
| | 10 Labelled Examples | | | | |
| $S^3CP^2$ | **82.02** | 50.62 | **89.51** | 76.17 | 50.90 |
| SVM | 55.28 | **56.16** | 72.90 | **79.12** | **52.61** |
| $S^3VM$ | 79.05 | 53.65 | 79.41 | 69.34 | 49.96 |
| | 100 Labelled Examples | | | | |
| $S^3CP^2$ | **86.15** | 72.93 | **92.77** | 86.62 | 70.86 |
| SVM | 74.67 | 71.98 | 90.10 | **86.89** | **71.56** |
| $S^3VM$ | 81.82 | **76.24** | 81.95 | 78.88 | 57.33 |

of the training data are incorporated into the model in such a way that the globally optimal solution must admit a separation of the classes within the training data, if such a solution exists, and for all scaling parameters close to zero. We also show that asymptotically this optimal solution converges to the subspace normal to the optimal TSVM hyperplane, as the scaling parameter is reduced to zero, thereby providing a theoretical connection between our proposed method and popular semi-supervised classification methodology. Experimental results indicate the proposed method is competitive with state-of-the-art TSVM implementation in terms of classification accuracy.

# Appendix. Proofs

The following lemma is useful for proving Lemmas 17 and 18.

**Lemma 19** *Let $k$ be non-increasing and Lipschitz with constant $K$, and let $k(0)=1$. For $\boldsymbol{\theta}\in\Theta$ let*

$$\Delta_{\boldsymbol{\theta}}=[\bar{m}_{\boldsymbol{\theta}},\bar{M}_{\boldsymbol{\theta}}]\cap[\max\{P(\boldsymbol{\theta})_i|y_i=-1\},\min\{P(\boldsymbol{\theta})_j|y_j=+1\}],$$

*where*

$$\bar{m}_{\boldsymbol{\theta}}=\frac{\mu_{P(\boldsymbol{\theta})}-\beta\sigma_{P(\boldsymbol{\theta})}+\min\{P(\boldsymbol{\theta})\cap[\mu_{P(\boldsymbol{\theta})}-\beta\sigma_{P(\boldsymbol{\theta})},\infty)\}}{2}$$
$$\bar{M}_{\boldsymbol{\theta}}=\frac{\mu_{P(\boldsymbol{\theta})}+\beta\sigma_{P(\boldsymbol{\theta})}+\max\{P(\boldsymbol{\theta})\cap(-\infty,\mu_{P(\boldsymbol{\theta})}+\beta\sigma_{P(\boldsymbol{\theta})}]\}}{2}.$$

*Let $\sigma'=\frac{K}{(1+\epsilon)^{1/1+\epsilon}}$. If $\Delta_{\boldsymbol{\theta}}\neq\emptyset$ then set $G_{\boldsymbol{\theta}}=\max_{b\in\Delta_{\boldsymbol{\theta}}}\min_{i\in\{1,\ldots N\}}|P(\boldsymbol{\theta})_i-b|$. Then for $\sigma\leq\sigma'$ we have,*

$$\lambda_2(\boldsymbol{\theta})\geq\min\left\{\frac{1}{9|X|^3}k\left(\frac{2G_{\boldsymbol{\theta}}+\delta D}{\sigma}\right),\frac{1}{9|X|}\left(\frac{\sigma}{K}\right)^{1+\epsilon}\right\},$$

*where $D=\max\{Diam(X),Diam(X)^{1-\delta}\}$. If $\Delta_{\boldsymbol{\theta}}=\emptyset$ then we simply have*

$$\lambda_2(\boldsymbol{\theta})\geq\frac{1}{9|X|}\left(\frac{\sigma}{K}\right)^{1+\epsilon},$$

*for all $\sigma\leq\sigma'$.*

**Proof** Since $k$ has Lipschitz constant $K$ we have

$$k(x/\sigma)+x^{1+\epsilon}\geq(k(0)-\frac{K}{\sigma}x)^+ +x^{1+\epsilon}=(1-\frac{K}{\sigma}x)^+ +x^{1+\epsilon}.$$

Since $\sigma\leq\sigma'=\frac{K}{(1+\epsilon)^{1/1+\epsilon}}$ we can show that $(1-\frac{K}{\sigma}x)^+ +x^{1+\epsilon}\geq\left(\frac{\sigma}{K}\right)^{1+\epsilon}$ for all $x\geq 0$.

First consider the case $\Delta_{\boldsymbol{\theta}}=\emptyset$. For each $j$, we must have either $P(\boldsymbol{\theta})_j \leq \max\{P(\boldsymbol{\theta})_i|y_i=-1\}$ or $P(\boldsymbol{\theta})_j \geq \min\{P(\boldsymbol{\theta})_i|y_i=+1\}$. Let $I$ and $J$ be the indices corresponding to $\max\{P(\boldsymbol{\theta})_i|y_i=-1\}$ and $\min\{P(\boldsymbol{\theta})_i|y_i=+1\}$ respectively. Therefore, for each $j$ s.t. $x_j \in \mathcal{X}^U$ either

$$A_{jI} \geq k(|P(\boldsymbol{\theta})_j - P(\boldsymbol{\theta})_I)|/\sigma) + |P(\boldsymbol{\theta})_j - P(\boldsymbol{\theta})_I|^{1+\epsilon} \geq \left(\frac{\sigma}{K}\right)^{1+\epsilon}$$

or

$$A_{jJ} \geq k(|P(\boldsymbol{\theta})_j - P(\boldsymbol{\theta})_J|/\sigma) + |P(\boldsymbol{\theta})_j - P(\boldsymbol{\theta})_J|^{1+\epsilon} \geq \left(\frac{\sigma}{K}\right)^{1+\epsilon}.$$

In addition we have

$$A_{IJ} \geq k(|P(\boldsymbol{\theta})_I - P(\boldsymbol{\theta})_J|/\sigma) + |P(\boldsymbol{\theta})_I - P(\boldsymbol{\theta})_J|^{1+\epsilon} \geq \left(\frac{\sigma}{K}\right)^{1+\epsilon},$$

and for each $j$ s.t. $y_j=+1$ and $i$ s.t. $y_i=-1$ we similarly have $A_{jI}, A_{iJ} \geq \left(\frac{\sigma}{K}\right)^{1+\epsilon}$. Now, let $u$ be the second eigenvector of $L(\boldsymbol{\theta})$, then $\|u\|=1, u \perp \mathbf{1}$ and so $\exists i,j$ s.t. $u_i - u_j \geq \frac{1}{\sqrt{|X|}}$. If $|u_I - u_J| \leq \frac{1}{3\sqrt{|X|}}$ then either $|u_i - u_I| \geq \frac{1}{3\sqrt{|X|}}$ and $|u_i - u_J| \geq \frac{1}{3\sqrt{|X|}}$ or $|u_j - u_I| \geq \frac{1}{3\sqrt{|X|}}$ and $|u_j - u_J| \geq \frac{1}{3\sqrt{|X|}}$. Then,

$$\begin{aligned}
\lambda_2(L(\boldsymbol{\theta})) = u \cdot L(\boldsymbol{\theta}) u &= \frac{1}{2}\sum_{k,l} A_{kl}(u_k - u_l)^2 \\
&\geq A_{iI}(u_i - u_I)^2 + A_{iJ}(u_i - u_J)^2 + A_{jI}(u_j - u_I)^2 + A_{jJ}(u_j - u_J)^2 \\
&\geq \frac{1}{9|X|}\left(\frac{\sigma}{K}\right)^{1+\epsilon}
\end{aligned}$$

in all possible cases, since $\left(\frac{\sigma}{K}\right)^{1+\epsilon} \leq 1 \leq H$. On the other hand we have $|u_I - u_J| \geq \frac{1}{3\sqrt{|X|}}$ and so $\lambda_2(L(\boldsymbol{\theta})) \geq A_{IJ}(u_I - u_J)^2 \geq \left(\frac{\sigma}{K}\right)^{1+\epsilon}/9|X|$ as required.

Now consider the case $\Delta_{\boldsymbol{\theta}} \neq \emptyset$. Define $u, I, J$ as above. If

$$\max_{i,j: P(\boldsymbol{\theta})_i, P(\boldsymbol{\theta})_j \in [P(\boldsymbol{\theta})_I, P(\boldsymbol{\theta})_J]} (u_i - u_j) \leq \frac{1}{3\sqrt{|X|}},$$

then since $\exists i,j$ with $u_i - u_j \geq \frac{1}{\sqrt{|X|}}$ we must have either $P(\boldsymbol{\theta})_i \notin [P(\boldsymbol{\theta})_I, P(\boldsymbol{\theta})_J]$ and $|u_i - u_I| \geq \frac{1}{3\sqrt{|X|}}$ and $|u_i - u_J| \geq \frac{1}{3\sqrt{|X|}}$ or $P(\boldsymbol{\theta})_j \notin [P(\boldsymbol{\theta})_I, P(\boldsymbol{\theta})_J]$ and $|u_j - u_I| \geq \frac{1}{3\sqrt{|X|}}$ and $|u_j - u_J| \geq \frac{1}{3\sqrt{|X|}}$. Suppose w/o loss of generality that $P(\boldsymbol{\theta})_i$ satisfies these three conditions. If $x_i \in \mathcal{X}^U$ then since $P(\boldsymbol{\theta})_i \notin [P(\boldsymbol{\theta})_I, P(\boldsymbol{\theta})_J]$ we have either $A_{iI} \geq \left(\frac{\sigma}{K}\right)^{1+\epsilon}$ or $A_{iJ} \geq \left(\frac{\sigma}{K}\right)^{1+\epsilon}$ and the result follows similarly to above. If $x_i \in \mathcal{X}^L$ then either $A_{iI}=1$ or $A_{iJ}=1 \geq \left(\frac{\sigma}{K}\right)^{1+\epsilon}$, and again the result follows as above. If instead we have

$$\max_{i,j: P(\boldsymbol{\theta})_i, P(\boldsymbol{\theta})_j \in [P(\boldsymbol{\theta})_I, P(\boldsymbol{\theta})_J]} (u_i - u_j) > \frac{1}{3\sqrt{|X|}},$$

then the result follows analogously to (Hofmeyr et al., 2015, Lemma 2), with the addition of the factor $\frac{1}{3}$ on the distance between elements of $u$ in consideration. ■

**Proof of Lemma 17** Let $G_{\boldsymbol{\theta}_1} > 0$ be defined as in Lemma 19, and let $b$ be the corresponding point at which the distance is maximised. Let $L$ and $R$ be the number of projected data lying to the left and right of $b$ respectively. Then, since spectral clustering solves the relaxation of the ratio cut, we have

$$\begin{aligned}\lambda_2(\boldsymbol{\theta}_1) &\leq \frac{1}{|X|}\min_{C\subset X}\sum_{\substack{i,j:x_i\in C\\ x_j\notin C}} A(\boldsymbol{\theta}_1)_{ij}\left(\frac{1}{|C|}+\frac{1}{|X\setminus C|}\right)\\ &\leq \frac{1}{|X|}\sum_{\substack{i,j:P(\boldsymbol{\theta}_1)_i<b\\ P(\boldsymbol{\theta}_1)_j>b}} k(d(P(\boldsymbol{\theta}_1)_i,P(\boldsymbol{\theta}_1)_j))\left(\frac{1}{L}+\frac{1}{R}\right)\\ &\leq k(2G_{\boldsymbol{\theta}_1}/\sigma).\end{aligned}$$

For any $\boldsymbol{\theta}_2$ s.t. $\Delta_{\boldsymbol{\theta}_2}$ defined as in Lemma 19 is empty, we have

$$\frac{\lambda_2(\boldsymbol{\theta}_1)}{\lambda_2(\boldsymbol{\theta}_2)} \leq 9|X|K^{1+\epsilon}k(2G_{\boldsymbol{\theta}_1}/\sigma)\sigma^{-(1+\epsilon)}$$

This right hand side is independent of $\boldsymbol{\theta}_2$, and converges to 0 as $\sigma\to 0^+$ since $k(x)\in o(x^{-(1+\epsilon)})$ as $x\to\infty$. Therefore the result holds. ■

**Proof of Lemma 18** By Lemma 17 we know that $\exists\sigma'>0$ s.t. $0<\sigma<\sigma'\Rightarrow \Delta_{\boldsymbol{\theta}_{\sigma,\delta}}\neq\emptyset$, where $\Delta_{\boldsymbol{\theta}}$ is as in Lemma 19. Take $\gamma>0$. It has been shown Pavlidis et al. (2015) that $\exists m_\gamma>0$ s.t. for $w\in\mathbb{R}^d$ and $c\in\mathbb{R}$ one has $\|(w,c)/\|w\| - (v^\star,b^\star)\|>\gamma\Rightarrow \text{margin}(w/\|w\|,c/\|w\|)<\text{margin}(v^\star,b^\star)-m_\gamma$. Let $\boldsymbol{\theta}^\star$ be such that $v(\boldsymbol{\theta}^\star)=v^\star$ so that $\text{margin}(v^\star,b^\star)=G_{\boldsymbol{\theta}^\star}$, where $G_{\boldsymbol{\theta}}$ is as in Lemma 19. As in the proof of Lemma 17 we have

$$\lambda_2(\boldsymbol{\theta}^\star)\leq k(2G_{\boldsymbol{\theta}^\star}/\sigma).$$

In addition, for small enough $\sigma>0$ we have

$$\frac{1}{9|X|^3}k\left(\frac{2G+\delta D}{\sigma}\right) < \frac{1}{9|X|}\left(\frac{\sigma}{K}\right)^{1+\epsilon}$$

holding uniformly in $\delta>0$ for any $G>0$, where $K,D$ are as in Lemma 19. Therefore, for small $\sigma,\delta$ we have

$$\lambda_2(\boldsymbol{\theta}_{\sigma,\delta}) \geq \frac{1}{9|X|^3}k\left(\frac{2G_{\boldsymbol{\theta}_{\sigma,\delta}}+\delta D}{\sigma}\right),$$

and hence

$$\frac{1}{9|X|^3}k\left(\frac{2G_{\boldsymbol{\theta}_{\sigma,\delta}}+\delta D}{\sigma}\right)\leq k\left(\frac{2G_{\boldsymbol{\theta}^\star}}{\sigma}\right)$$

since $\lambda_2(\boldsymbol{\theta}_{\sigma,\delta})\leq\lambda_2(\boldsymbol{\theta}^\star)$. Now, take $\delta'$ s.t. $\delta' D<\frac{m_\gamma}{2}$. Since $\lim_{x\to\infty}k(x+\epsilon)/k(x)=0$ for all $\epsilon>0$, $\exists\sigma'>0$ s.t. $9|X|^3K(2G_{\boldsymbol{\theta}^\star}/\sigma)<k((2G_{\boldsymbol{\theta}^\star}-\frac{m_\gamma}{2})/\sigma)$ for all $\sigma<\sigma'$. For $\sigma<\sigma',\delta<\delta'$ we have,

$$\begin{aligned}
k\left(\frac{2G_{\boldsymbol{\theta}_{\sigma,\delta}}+\frac{m_\gamma}{2}}{\sigma}\right)\leq k\left(\frac{2G_{\boldsymbol{\theta}_{\sigma,\delta}}+\delta D}{\sigma}\right)\leq 9|X|^3k\left(\frac{2G_{\boldsymbol{\theta}^\star}}{\sigma}\right)\leq k\left(\frac{2G_{\boldsymbol{\theta}^\star}-\frac{m_\gamma}{2}}{\sigma}\right)\\
\Rightarrow 2G_{\boldsymbol{\theta}_{\sigma,\delta}}+\frac{m_\gamma}{2}\geq 2G_{\boldsymbol{\theta}^\star}-\frac{m_\gamma}{2}\\
\Rightarrow \max_{b\in\Delta_{\boldsymbol{\theta}_{\sigma,\delta}}}\text{margin}(v(\boldsymbol{\theta}_{\sigma,\delta}),b)\geq\text{margin}(v^\star,b^\star)-\frac{m_\gamma}{2}\\
\Rightarrow\|(v(\boldsymbol{\theta}_{\sigma,\delta}),b)-(v^\star,b^\star)\|\leq\gamma\\
\Rightarrow\|v(\boldsymbol{\theta}_{\sigma,\delta})-v^\star\|\leq\gamma
\end{aligned}$$

Since $\gamma>0$ was arbitrary and the above holds for all $\sigma<\sigma',\delta<\delta'$ we must have $\lim_{\sigma,\delta\to0^+}v(\boldsymbol{\theta}_{\sigma,\delta})=v^\star$ as required. ■

# Chapter 4

# Clustering by Minimum Cut Hyperplanes


## Abstract

*Minimum normalised graph cuts are highly effective ways of partitioning unlabeled data, having been made popular by the success of spectral clustering. This work presents a novel method for learning hyperplane separators which minimise this graph cut objective. The optimisation problem associated with the proposed method can be formulated as a sequence of univariate subproblems, in which the optimal hyperplane orthogonal to a given vector is determined. These subproblems can be solved in log-linear time, by exploiting the trivial factorisation of the exponential function. Experimentation suggests that the empirical runtime of the overall algorithm is also log-linear in the number of data. This compares favourably with existing methods based on normalised graph cuts.*

*Asymptotic properties of the minimum cut hyperplane, both for a finite sample, and for an increasing sample assumed to arise from an underlying probability distribution are discussed. In the finite sample case the minimum cut hyperplane converges to the maximum margin hyperplane as the scaling parameter is reduced to zero.*

*Applying the proposed methodology, both for fixed scaling, and the large margin asymptotes, is shown to produce high quality clustering models in comparison with state-of-the-art clustering algorithms in experiments using a large collection of benchmark datasets.*


## 1 Introduction

Clustering is fundamental to many statistical and machine learning applications, and deals with partitioning sets of data into groups which are believed

to be related, without any explicit prior information regarding the group associations of any of the data. Such applications arise in diverse fields from computer vision (Tatiraju and Mehta, 2008), to bio-informatics (Sturn et al., 2002) to marketing (Punj and Stewart, 1983).

Common to many clustering methods is the notion of similarity, and the clustering problem can be abstractly stated as follows: determine a partition of a data set such that data within groups are more similar than data between groups. The popular $K$-means algorithm, as well as other *centroid* based methods, define groups of similar data by how they cluster around a small number of cluster centroids (Leisch, 2006). Density based clustering methods form groups of data which fall in connected regions of high data density. In the statistics literature these *high density clusters* are interpreted via the level sets of an assumed underlying probability density (Hartigan, 1975). A third approach, and that which forms the motivation for this work, is clustering by graph cuts. This approach provides a highly principled framework within which to address the notion of similarity, as the objective which drives the partitioning deals explicitly with the pairwise similarities between data. The minimum graph cut problem directly minimises the sum of the similarities between data belonging to different groups in a partition. Normalisations of the graph cut objective are used to emphasise partitions which have high within cluster similarity, and to induce more balanced partitions. The normalised graph cut problem, however, is NP-hard (Wagner and Wagner, 1993). The relaxations given by spectral clustering (Hagen and Kahng, 1992; Shi and Malik, 2000) mitigate this problem, however the resulting complexity remains $\mathcal{O}(n^2)$, where $n$ is the number of data. Further approximations have been developed to allow these methods to be applied to larger problems. The Nyström method generates a low rank approximation of the matrix of pairwise similarities before applying spectral clustering to this approximate *affinity matrix* (Fowlkes et al., 2004). An alternative approach reduces the size of the affinity matrix by first performing a coarse clustering of the data using a comparatively simple method (e.g. $K$-means), and then computing the pairwise similarities between the resulting clusters (Yan et al., 2009).

This paper introduces a new divisive hierarchical clustering method, in which each partition in the hierarchy is induced by a hyperplane separator. Each separating hyperplane is the solution to an optimisation problem which is motivated by the graph partitioning objective, and minimises the normalised graph cut measured across the hyperplane. Restricting attention to hyperplane based partitions, while slightly limiting in the fact that the resulting clusters must be linearly separated, offers considerable computational benefits. Moreover, the empirical performance of the method is highly competitive with state-of-the-art clustering algorithms in terms of clustering accuracy. The formulation presented can be viewed as a sequence of univariate subproblems, each of which can be solved in $\mathcal{O}(n \log n)$ time, where $n$ is

the number of data. An empirical study indicates that the overall algorithm also runs in $\mathcal{O}(n \log n)$ time, allowing for the application of the proposed approach to large problems without the need for approximations. Moreover, the empirical runtime of the proposed method compares favourably with the approximate spectral clustering methods.

Asymptotic properties of the normalised cut across a hyperplane are explored, showing desirable qualities in the context of the non-parametric statistical formulation of the clustering problem (Hartigan, 1975). In particular the optimal hyperplane will both have low density integral, therefore passing through regions of low density, and also will tend to separate the modes of the underlying probability density, which correspond to high density clusters. Additionally the optimal hyperplane for the proposed objective, evaluated on a fixed data set, is shown to be connected with maximum margin hyperplane separators, and the proposed methodology can be modified to find large margin hyperplanes practically. This is an important result as maximum margin clustering has gained considerable attention in recent years for its good performance in many application areas including image analysis, and text mining (Xu et al., 2004; Zhang et al., 2009). Experiments with the proposed approach for finding large margin hyperplanes indicates that it is highly competitive with existing methods for the problem. In addition, the approach significantly reduces the computation time required to find large margin hyperplanes.

The remainder of this paper is organised as follows. Section 2 introduces graph partitioning and presents the proposed problem formulation. The asymptotic properties of the proposed method are investigated and discussed. In Section 3 the methodology for solving the proposed problem is presented in detail. Section 4 presents the results from extensive experiments on benchmark data sets for clustering. Finally, concluding remarks are given in Section 5.

## 2 Problem Formulation

This section presents an introduction to graph partitioning by (normalised) graph cuts, and provides the hyperplane based formulation which forms the focus of this paper. A theoretical discussion of two popular normalisation techniques, RatioCut (Hagen and Kahng, 1992) and normalised cut, or NCut (Shi and Malik, 2000), is presented. The conclusion is, at least in relation to the proposed hyperplane formulation, that NCut has preferable characteristics in the context of data clustering. Following this analysis, the NCut approach is adopted as the focus of the subsequent methodology presented in the next section. In addition, a connection between the optimal hyperplane based on the NCut objective and the maximum margin hyperplane is

established, showing that as the scaling parameter in the Ncut formulation is reduced to zero, the optimal NCut hyperplane converges to the maximum margin hyperplane through the data.

## 2.1 Background on Normalised Graph Cuts

The graph cut problem for data partitioning is given as follows. For a set of data $\mathcal{X}=\{x_1,...,x_n\}$ in $\mathbb{R}^d$, define the graph $G(\mathcal{X})=(\mathcal{X},E)$ with vertices equal to the elements of $\mathcal{X}$ and undirected edges assuming weights equal to the similarity between pairs of data in $\mathcal{X}$. In practice the similarity between points relates to their relative spatial relationship, in that pairs of data which are further apart tend to be given lower similarity value than those which are closer. The information in $G(\mathcal{X})$ can be represented by the affinity matrix $A(\mathcal{X})\in\mathbb{R}^{n\times n}$ which has $i,j$-th element equal to the similarity between data $x_i$ and $x_j$. From the affinity matrix one can define the diagonal *degree matrix* $D(\mathcal{X})\in\mathbb{R}^{n\times n}$ with $i$-th diagonal element equal to $\sum_{j=1}^{n} A(\mathcal{X})_{i,j}$.

Now, for a subset $C\subset\mathcal{X}$ the graph cut between $C$ and $\mathcal{X}\setminus C$ is given by the sum of the similarities between data in $C$ and those in $\mathcal{X}\setminus C$. Formally,

$$\text{Cut}(C,\mathcal{X}\setminus C):=\sum_{\substack{i,j:x_i\in C\\ x_j\notin C}} A(\mathcal{X})_{i,j}. \tag{4.1}$$

Minimising (4.1) over all subsets $C\subset\mathcal{X}$ can be performed in polynomial time (Stoer and Wagner, 1997), however this approach tends to result in either $C$ or $\mathcal{X}\setminus C$ containing very few points (von Luxburg, 2007). Normalisations are used to induce more balanced partitions. Two such normalisations are common, known as RatioCut and NCut, defined as follows,

$$\text{RatioCut}(C,\mathcal{X}\setminus C):=\text{Cut}(C,\mathcal{X}\setminus C)\left(\frac{1}{|C|}+\frac{1}{|\mathcal{X}\setminus C|}\right) \tag{4.2}$$

$$\text{NCut}(C,\mathcal{X}\setminus C):=\text{Cut}(C,\mathcal{X}\setminus C)\left(\frac{1}{\sum\limits_{i:x_i\in C} D(\mathcal{X})_i}+\frac{1}{\sum\limits_{i:x_i\notin C} D(\mathcal{X})_i}\right). \tag{4.3}$$

The minimisation of both RatioCut and NCut over all subsets $C\subset\mathcal{X}$ is NP-hard (Wagner and Wagner, 1993). Considering only hyperplane based partitions, however, results in a far simpler problem, as will be discussed in the remainder.

## 2.2 Normalised Cuts Across Hyperplanes

A hyperplane in $\mathbb{R}^d$ is a translated subspace of co-dimension 1, and can be parameterised by a unit vector $v\in\mathbb{R}^d, \|v\|=1$, and scalar $b\in\mathbb{R}$ as the set,

$$H(v,b):=\{x\in\mathbb{R}^d | v\cdot x=b\}. \tag{4.4}$$

Notice that the unit-vector $v$ defines the subspace normal to the hyperplane. Henceforth denote the set of unit norm vectors in $\mathbb{R}^d$ by $\mathbb{B}^d$.

Hyperplanes induce a binary partition of $\mathbb{R}^d$, and therefore of any data set residing in $\mathbb{R}^d$, based on the negative/non-negative elements of the subspace, i.e., separating those $x \in \mathbb{R}^d$ s.t. $v \cdot x < b$ from those satisfying $v \cdot x \geq b$. The following notation will be used to define these *half spaces*,

$$H(v,b)^- := \{x \in \mathbb{R}^d | v \cdot x < b\} \tag{4.5}$$

$$H(v,b)^+ := \{x \in \mathbb{R}^d | v \cdot x \geq b\}. \tag{4.6}$$

Hyperplane based clustering algorithms have been successfully applied in a number of application areas, including text mining, microarray analysis, and image segmentation (Boley, 1998; Tasoulis et al., 2010; Zhang et al., 2009).

Connections between clustering by normalised graph cuts and the so-called *low density separation assumption* have been established (Narayanan et al., 2006), in that the value of the normalised cut is asymptotically related to the integrated density along the surface inducing the cut. The low density separation assumption states that clusters are separated by contiguous regions of relatively low density, and is equivalent to the definition of clusters as high density regions which underlies density clustering. Within the non-parametric statistical approach to clustering, high density clusters may be associated with the level sets of an (assumed) underlying probability density (Hartigan, 1975). A separating hyperplane for clustering should therefore, as well as possible, separate regions of high density (it should separate clusters) while avoiding intersection with any such high density regions (individual clusters should remain intact after the partition). In particular, the surface integral of the underlying probability density along the hyperplane should be low, while there should exist regions of high density lying both sides of the hyperplane. A further theoretical investigation into the asymptotic properties of normalised cuts, and their relationship to high density clustering, is presented below. Particular attention is paid to hyperplane separators, and in relation to the criteria set out above.

For a random variable $X$ in $\mathbb{R}^d$ with probability density $p : \mathbb{R}^d \to \mathbb{R}^+$, the surface integral of the density along a hyperplane, $H(v,b)$, is simply the value of the marginal density of the univariate random variable $v \cdot X$ evaluated at $b$. Henceforth, for $v \in \mathbb{B}^d$, the marginal density of $v \cdot X$ will be written as $p^v$, i.e.,

$$p^v(b) = \int_{x \in H(v,b)} p(x) dx. \tag{4.7}$$

While exceptions exist, in general we find that if a hyperplane $H(v,b)$ separates modes in the marginal density $p^v$ then there tend to exist modes in the

full dimensional joint density, $p$, on both sides of $H(v,b)$. A natural assumption when clustering data assumed to arise from an underlying probability distribution is that the distribution represents a mixture of comparatively simple components. If these components are, for example, elliptical, then a hyperplane separating modes in the marginal density will always intersect the convex hull of the modes of the individual components, as shown in the following simple proposition.

**Proposition 20** *Let $p(x)=\sum_{i=1}^{k}\pi_i p_i(x)$, where $\sum_{i=1}^{k}\pi_i=1,\pi_i>0,i=1,...,k$ and each $p_i$ is elliptical. Then a hyperplane $H(v,b)$ which intersects the convex hull of the modes of $p^v$ intersects the convex hull of the modes of the $p_i$.*

The quality of a hyperplane for clustering a data set $\mathcal{X}=\{x_1,...,x_n\}$ can therefore, in general, be defined in terms of the properties of the empirical distribution of the *marginal data set* $v\cdot\mathcal{X}:=\{v\cdot x_1,...,v\cdot x_n\}$, i.e., the data set projected into the subspace normal to the hyperplane. Considering the properties of the data only within this subspace has benefits in terms of computational tractability, as well as theoretically in the context of high dimensional applications. Many authors have addressed the problems associated with clustering high dimensional data sets (Agrawal et al., 1998; Kriegel et al., 2009; Steinbach et al., 2004), with the observation that as dimensionality grows the relative distance between points tends to be more uniform, making distance/similarity based clustering unreliable. Moreover the potential irrelevance of certain dimensions to the cluster structure of the data, or of the probability distribution from which they arose, can significantly affect the quality of the clustering model and any inference made from it. By reducing the dimension of the data by considering univariate projections, these problems are substantially mitigated.

The normalised graph cut of a data set $\mathcal{X}$ based on a hyperplane, $H(v,b)$, is therefore defined in terms of the marginal data set, $v\cdot\mathcal{X}$. First, let $K:\mathbb{R}^+\to\mathbb{R}^+$ be a decreasing function. For $\sigma>0$, $v\in\mathbb{B}^d$ define

$$K_\sigma:\mathbb{R}\times\mathbb{R}\to\mathbb{R}^+:(x,y)\mapsto K\left(\frac{|x-y|}{\sigma}\right) \tag{4.8}$$

$$K_\sigma^v:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}^+:(x,y)\mapsto K\left(\frac{|v\cdot x-v\cdot y|}{\sigma}\right). \tag{4.9}$$

$K_\sigma^v$ is used to define the similarity between elements of the marginal data along $v$, while when in the context of a univariate data set similarities are

defined using $K_\sigma$. The parameter $\sigma > 0$ is the scaling parameter. Then define,

$$\text{RatioCut}(v,b|\mathcal{X}) := \sum_{\substack{x\in\mathcal{X}\cap H(v,b)^- \\ y\in\mathcal{X}\cap H(v,b)^+}} K_\sigma^v(x,y)\left(\frac{1}{|\mathcal{X}\cap H(v,b)^-|}+\frac{1}{|\mathcal{X}\cap H(v,b)^+|}\right) \tag{4.10}$$

$$\text{NCut}(v,b|\mathcal{X}) := \sum_{\substack{x\in\mathcal{X}\cap H(v,b)^- \\ y\in\mathcal{X}\cap H(v,b)^+}} K_\sigma^v(x,y)\left(\frac{1}{\sum\limits_{\substack{x\in H(v,b)^- \\ y\in\mathcal{X}}} K_\sigma^v(x,y)}+\frac{1}{\sum\limits_{\substack{x\in H(v,b)^+ \\ y\in\mathcal{X}}} K_\sigma^v(x,y)}\right), \tag{4.11}$$

adopting the convention that an empty sum is equal to 0 and that $\frac{0}{0} = 0$.

In what follows the asymptotic properties of RatioCut and NCut are briefly discussed. While the results of this section apply for fairly general *similarity kernel*, $K(\cdot)$, it is assumed henceforth and in the proofs of those results, that $K(x) = \exp(-x)$. This particular similarity function is chosen as it allows for fast (local) optimisation of the normalised cut hyperplane. This is discussed further in the subsequent section, where corresponding derivations are presented.

**Lemma 21** *Let $\mathcal{X} = \{X_1, ..., X_n\}$ be an i.i.d. random sample in $\mathbb{R}^d$ where each $X_i$ is a continuous random variable, with density function $p$ and support $S$. Let $H(v, b)$ be a hyperplane intersecting $Int(S)$ such that $p^v$ is differentiable at $b$. Let $\sigma_n$ be a null sequence satisfying $\lim_{n\to\infty} n\sigma_n^{4+\epsilon} = \infty$ for any fixed $\epsilon > 0$. Then,*

$$\frac{1}{n\sigma_n^2}\mathit{RatioCut}(v,b|\mathcal{X}) \xrightarrow{a.s.} \frac{p^v(b)^2}{\mathbb{P}(X\in H(v,b)^-)\mathbb{P}(X\in H(v,b)^+)}$$

*as $n\to\infty$.*

The above lemma shows that the asymptotic properties of the RatioCut have some similarities with features relevant in the context of clustering. In particular, the minimum RatioCut hyperplane is likely to avoid intersecting regions of high density, captured by the square of the integrated density along it. Moreover, the balance of the resulting partition is captured in the division by the probabilities of lying either side of the hyperplane. However, even for relatively short tailed distributions such as a mixture of Gaussians, the quantity $\frac{p^v(b)^2}{\mathbb{P}(X\in H(v,b)^-)\mathbb{P}(X\in H(v,b)^+)} \to 0$ as $b\to\infty$, and hence the minimum RatioCut solution will not converge as the number of data increases, and instead the optimal partitioning hyperplane will tend to diverge into the tail. This makes the RatioCut an unappealing objective for clustering by hyperplane separators.

The NCut solution is stable for a much richer class of distributions. In particular, it is shown that if the marginal distributions of the collection of random variables $\{v \cdot X | v \in \mathbb{B}^d\}$ all have unbounded hazard function, then there exists an optimal hyperplane based on the asymptotic value of the NCut.

**Lemma 22** *Let $\mathcal{X} = \{X_1, ..., X_n\}$ be an i.i.d. random sample in $\mathbb{R}^d$ where each $X_i$ is a continuous random variable, with density function $p$ and support $S$. Let $H(v, b)$ be a hyperplane intersecting $Int(S)$ such that $p^v$ is differentiable at $b$. Let $\sigma_n$ be a null sequence satisfying $\lim_{n\to\infty} n\sigma_n^{4+\epsilon} = \infty$ for any fixed $\epsilon > 0$. Then,*

$$\frac{2}{\sigma_n} NCut(v,b|\mathcal{X}) \xrightarrow{a.s.} p^v(b)^2 \left( \frac{1}{\int_{-\infty}^{b} p^v(x)^2 dx} + \frac{1}{\int_{b}^{\infty} p^v(x)^2 dx} \right)$$

*as $n \to \infty$.*

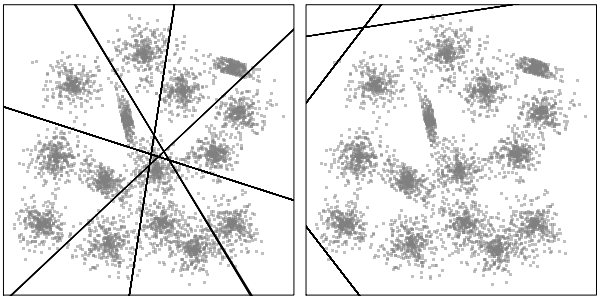
The crucial difference between the above and the corresponding RatioCut limit lies in squaring the marginal density within the denominator integrals. Both RatioCut and Ncut will lead to hyperplanes with low integrated density, however normalising by the integrated square density will favour solutions in which the marginal density concentrates around at least one location either side of the hyperplane, these regions of concentrated density corresponding to modes in the density function. This is illustrated in Figure 4.1, which shows the results from 100 random initialisations based on NCut (left) and RatioCut (right) on the S2 dataset (Fränti and Virmajoki, 2006). The data represent a sample from a Gaussian mixture with 15 components. Minimising NCut resulted in 4 solutions, all of which provide a visually pleasing partition of the data, avoiding intersection with any of the high density regions, but also providing a meaningful partition of the data. The RatioCut solutions, however, all result in only a very small number of data being partitioned from the rest.

The following establishes sufficient conditions for the existence of an optimal hyperplane based on the asymptotic value of the NCut. Let $Y$ be a continuous random variable with support $S \subset \mathbb{R}$. Let $f$ and $F$ be the density and distribution functions of $Y$ respectively. Then the *hazard function* of $Y$ is defined as,

$$H : Int(S) \to \mathbb{R}^+ : \; y \mapsto \frac{f(y)}{1 - F(y)}$$

**Proposition 23** *Let $X$ be a continuous random variable in $\mathbb{R}^d$ with continuous density $p$ and support $S$. Assume that $p$ has finitely many modes and that for all*

**Fig. 4.1:** Optimal hyperplanes based on NCut (left) and RatioCut (right) from the same 100 initialisations



*$v \in \mathbb{B}^d$ the random variable $v \cdot X$ has unbounded hazard function. Then there exists a hyperplane, $H(v, b)$, which minimises the asymptotic NCut criterion defined as*

$$p^v(b)^2 \left( \frac{1}{\int_{-\infty}^{b} p^v(y)^2 dy} + \frac{1}{\int_{b}^{\infty} p^v(y)^2 dy} \right).$$

The conditions of the above result can be relaxed somewhat, to include for example lower semi-continuous densities. For ease of exposition, however, the above formulation is preferred. Notice that the class of distributions with unbounded hazard functions for all of its marginals includes Gaussian mixtures, but not those with polynomially decaying tails.

There is much debate over which normalisation of the graph cut is preferable (von Luxburg et al., 2008). The analysis above suggests, at least for the hyperplane formulations in Eq.'s (4.10) and (4.11), that NCut is preferable in the context of clustering. The remainder of this paper therefore focuses on NCut. The minimum NCut hyperplane is defined as the solution to the optimisation problem,

$$\min_{(v,b)\in\mathcal{F}} \text{NCut}(v, b|\mathcal{X}) \tag{4.12}$$
$$\mathcal{F} := \{(v, b) \in \mathbb{B}^d \times \mathbb{R} \big| H(v, b) \cap Int(\mathbf{conv}(\mathcal{X})) \neq \emptyset\},$$

where $\mathbf{conv}(\mathcal{X})$ denotes the convex hull of the data set. Only hyperplanes which intersect the interior of the convex hull of the data set $\mathcal{X}$ are considered, as it is necessary that the hyperplane partitions $\mathcal{X}$.

## 2.3 Connection with Maximum Margin Hyperplanes

In this subsection the finite sample properties of minimum NCut hyperplanes are discussed in relation to large margin separation. For a data set $\mathcal{X} =$

$\{x_1, ..., x_n\}$ in $\mathbb{R}^d$, the margin of a hyperplane on $\mathcal{X}$ is the Euclidean distance between the hyperplane and $\mathcal{X}$, i.e.,

$$\text{margin}H(v, b) := d(H(v, b), \mathcal{X}) = \min_{i \in \{1,...,n\}} |v \cdot x_i - b|. \tag{4.13}$$

Maximum margin hyperplane classifiers have become extremely popular over the past few decades, owing to the success of Support Vector Machines (SVM, (Vapnik and Kotz, 1982)). SVM classifiers were introduced for supervised classification, and extended to semi-supervised learning via Transductive Support Vector Machines (TSVM, (Vapnik, 1998)). More recently the support vector approach has been successfully applied to the fully unsupervised problem of maximum margin clustering (Xu et al., 2004; Zhang et al., 2009). The maximum margin clustering problem can be equvalently stated as identifying the binary labelling of a data that will maximimise the margin of an SVM estimated using the assigned labels. Early maximum margin clustering algorithms used semi-definite programming formulations, which limits their application to data sets of only a few hundred points (Xu et al., 2004). The iterative Support Vector Regression approach (Zhang et al., 2009) repeatedly solves a support vector regression problem, updating the labels assigned to the data after each iteration. This method has shown strong performance empirically and has allowed the application of large margin clustering to much larger problems than was feasible previously.

The following lemma establishes the convergence of the minimum NCut hyperplane to the maximum margin hyperplane, as the scaling parameter is reduced to zero. Notice that for $v \in \mathbb{B}^d$ and for $b_1, b_2 \in \mathbb{R}$ s.t. $\mathcal{X} \cap H(v, b_1)^+ = \mathcal{X} \cap H(v, b_2)^+$ one has $\text{NCut}(v, b_1|\mathcal{X}) = \text{NCut}(v, b_2|\mathcal{X})$. As a result there is not a unique minimum cut hyperplane. The convergence to the maximum margin hyperplane is therefore discussed only in relation to the normal vector. Notice that if the normal vector to the maximum margin hyperplane is known, then obtaining the maximum margin hyperplane is trivial. These two results are thus practically equivalent.

**Lemma 24** *Let $\mathcal{X} = \{x_1, ..., x_n\}$ be a data set in $\mathbb{R}^d$. Suppose there is a unique hyperplane intersecting $\mathbf{conv}(\mathcal{X})$ with maximum margin, parameterised by $(v_m, b_m) \in \mathbb{B}^d \times \mathbb{R}$. For $\sigma > 0$ define $(v_\sigma, b_\sigma)$ to be any minimiser of the optimisation problem $\min_{(v,b)\in\mathcal{F}} NCut(v, b|\mathcal{X}, \sigma)$, where there is now an explicit dependence on the scaling parameter, $\sigma$. Then,*

$$\lim_{\sigma \to 0^+} d(v_\sigma, \{v_m, -v_m\}) = 0.$$

The above result relies on the fact that the NCut objective can be viewed as a sort of smoothing of the maximum margin clustering problem, in that for large values of $\sigma$ the global structure of the data set defines the strength of

the clustering solution, but as $\sigma$ decreases towards zero points nearer the hyperplane become more influential, and asymptotically it is dominated by the nearest points. Solving the minimum cut hyperplane problem repeatedly for a shrinking sequence of scaling parameters is thus strongly reminiscent of the homotopy continuation approach to non-convex optimisation (Allgower and Georg, 2012), in which an objective function is convolved with a sequence of smoothing functions the last of which is the identity function. Initialising each smoothed problem using the solution to the previous provides a sequence of solutions which converges to a good solution to the original optimisation problem.

The above result provides a new approach for finding large margin hyperplanes for clustering. One of the challenges associated with maximum margin clustering is the fact that the objective is plagued with local minima (Zhang et al., 2009). The above approach, due to its similarity with homotopy continuation, has the potential to mitigate this problem, and therefore generate higher quality solutions than those based directly on support vector methods.

# 3 Methodology

In the previous section, properties of hyperplane based graph cuts were introduced. It was established that the optimal hyperplane based on NCut is a preferable objective to that for RatioCut. This section is dedicated to optimising $\mathrm{NCut}(v, b|\mathcal{X})$ over $(v, b) \in \mathcal{F}$, where $\mathcal{X}$ is a data set in $\mathbb{R}^d$. The optimisation procedure proposed treats optimising over $b$ as a subproblem to the master problem of optimising over $v$.

The global objective is thus formulated as,

$$\min_{v \in \mathbb{B}^d} \Phi(v|\mathcal{X}), \tag{4.14}$$

$$\Phi(v|\mathcal{X}) := \min_{b \in Int(\mathbf{conv}(v \cdot \mathcal{X}))} \mathrm{NCut}(v, b|\mathcal{X}). \tag{4.15}$$

Subsection 3.1 describes the optimisation procedure of the subproblem in Eq. (4.15). A log-linear time method is presented, wherein the log factor arises solely from the requirement that the marginal data set, $v \cdot \mathcal{X}$, must be sorted. Subsection 3.2 is dedicated to the optimisation of $\Phi(v|\mathcal{X})$. These suffice to generate a bipartition of $\mathcal{X}$, which is the main focus of this work. Extensions to multi-cluster partitions are covered briefly in subsection 3.3.

## 3.1 Optimal NCut of the Marginal Data Set $v \cdot \mathcal{X}$

In this subsection it is assumed that the vector $v \in \mathbb{B}^d$ is fixed, and the optimal value of $b$ is to be determined based on the marginal data set, $v \cdot \mathcal{X}$. For

brevity of notation the $i$-th element of the sorted marginal data set is denoted $x_i^v$. That is, $x_1^v \leq x_2^v \leq ... \leq x_n^v$. Ties may be broken in an arbitrary but well defined manner, such as via the order of indices in the original data set.

Observe that for $b_1, b_2 \in (x_i^v, x_{i+1}^v]$ we have $\text{NCut}(v, b_1|\mathcal{X}) = \text{NCut}(v, b_2|\mathcal{X})$ and hence one needs only consider at most $n-1$ possible values of $b$, corresponding to the midpoints between consecutive distinct elements of the marginal data set. To that end, define

$$b_i := \frac{x_i^v + x_{i+1}^v}{2}, \ i \in \{1, ..., n-1\}. \tag{4.16}$$

Define $CS(\cdot)$ to be the cumulative sum function. That is, if $w$ has length $n$,

$$CS(w) = \left( w_1, \sum_{i=1}^{2} w_i, \ldots, \sum_{i=1}^{n} w_i \right). \tag{4.17}$$

Let $EG$ be the vector of exponentials of the scaled differences between $x_1^v$ and each element in the sorted marginal data set, i.e.,

$$EG := \left( \exp\left(\frac{x_1^v - x_1^v}{\sigma}\right), \exp\left(\frac{x_1^v - x_2^v}{\sigma}\right), ..., \exp\left(\frac{x_1^v - x_n^v}{\sigma}\right) \right). \tag{4.18}$$

Then, for $i \leq j$,

$$K_\sigma(x_i^v, x_j^v) = \exp\left(\frac{x_i^v - x_j^v}{\sigma}\right) = \frac{\exp\left(\frac{x_1^v - x_j^v}{\sigma}\right)}{\exp\left(\frac{x_1^v - x_i^v}{\sigma}\right)} = \frac{EG_j}{EG_i}$$

Therefore, the graph cut at $b_k$ is given by,

$$\begin{aligned}\sum_{i=1}^{k}\sum_{j=k+1}^{n} K_\sigma(x_i^v, x_j^v) &= \sum_{i=1}^{k}\sum_{j=k+1}^{n} \frac{EG_j}{EG_i} = \sum_{i=1}^{k} \frac{1}{EG_i} \sum_{j=k+1}^{n} EG_j \\ &= CS\left(\frac{1}{EG}\right)_k (CS(EG)_n - CS(EG)_k).\end{aligned}$$

Similarly, the sum of the first $k$ degrees in the graph of $v \cdot \mathcal{X}$ is given by,

$$\begin{aligned}\sum_{i=1}^{k}\sum_{j=1}^{n} K_\sigma(x_i^v, x_j^v) &= \sum_{i=1}^{k}\sum_{j=i+1}^{n} \frac{EG_j}{EG_i} + \sum_{i=1}^{k}\sum_{j=1}^{i} \frac{EG_i}{EG_j} \\ &= \sum_{i=1}^{k} \frac{1}{EG_i}(CS(EG)_n - CS(EG)_i) + \sum_{i=1}^{k} EG_i CS\left(\frac{1}{EG}\right)_i \\ &= CS\left(\frac{1}{EG} \cdot (CS(EG)_n - CS(EG))\right)_k + CS\left(EG \cdot CS\left(\frac{1}{EG}\right)\right)_k.\end{aligned}$$

The NCut for any $b_k$ can therefore be extracted from a collection of cumulative sum vectors. Computing these vectors has computational cost $\mathcal{O}(n)$, with the extraction of each $\text{NCut}(v, b_k|\mathcal{X})$ costing $\mathcal{O}(1)$. The solution to the problem $\min_b \text{NCut}(v, b|\mathcal{X})$ can therefore be computed in $\mathcal{O}(n \log n)$ time, with the log $n$ factor only arising in the sorting of the marginal data set, $v \cdot \mathcal{X}$.

## 3.2 Optimising $\Phi(v|\mathcal{X})$

In this subsection the optimisation of the master problem, $\min_{v \in \mathbb{B}^d} \Phi(v|\mathcal{X})$ is discussed. Notice that $\Phi(v|\mathcal{X})$ may not be continuous in $v$, since if $v \cdot \mathcal{X}$ contains repeated points these cannot be assigned to different elements of the partition, while in any neighbourhood of $v$ there exist projections along which they can be. In practice the following optimisation problem is considered instead,

$$\min_{v \| \|v\| \geq 1} \min_{k \in \{1,\dots,n-1\}} \phi_k\left(\frac{v}{\|v\|}|\mathcal{X}\right), \tag{4.19}$$

$$\phi_k(v|\mathcal{X}) := \sum_{i=1}^{k}\sum_{j=k+1}^{n} K_\sigma(x_i^v, x_j^v)\left(\frac{1}{\sum_{i=1}^{k}\sum_{j=1}^{n} K_\sigma(x_i^v, x_j^v)} + \frac{1}{\sum_{i=k+1}^{n}\sum_{j=1}^{n} K_\sigma(x_i^v, x_j^v)}\right), \tag{4.20}$$

where the computation of $\min_k \phi_k(v/\|v\||\mathcal{X})$ is described in the previous subsection. It is straightforward to see that the above problem is practically equivalent to the minimisation of $\Phi(v|\mathcal{X})$ in general. Notice that if $v$ is such that $v \cdot \mathcal{X}$ contains no repeated points, then

$$\min_{k \in \{1,\dots,n-1\}} \phi_k\left(\frac{v}{\|v\|}|\mathcal{X}\right) = \Phi\left(\frac{v}{\|v\|}|\mathcal{X}\right).$$

If the full dimensional data set, $\mathcal{X}$, contains no repeated points and for all $v \neq \mathbf{0}$ one has $Int(\mathbf{conv}(v \cdot \mathcal{X})) \neq \emptyset$, then this occurs on an open and dense set in $\mathbb{R}^d$. For those $v$ s.t. $v \cdot \mathcal{X}$ *does* contain repeated points, evaluating $\min_k \phi_k(v/\|v\||\mathcal{X})$ is the same as for $\Phi(v/\|v\||\mathcal{X})$, except that some partitions which separate equal elements of $v \cdot \mathcal{X}$ are permitted. Therefore, $\min_k \phi_k(v/\|v\||\mathcal{X}) \leq \Phi(v/\|v\||\mathcal{X})$ and hence for all $v \neq 0$ one has

$$\text{liminf}_{w \to v} \Phi(w/\|w\||\mathcal{X}) = \min_k \phi_k(v/\|v\||\mathcal{X}).$$

The objective in (4.19) benefits from being continuous, and is in fact Lipschitz continuous over the set $\{v \in \mathbb{R}^d | \|v\| \geq 1\}$, as discussed below. The objective is therefore differentiable almost everywhere, and so gradient based

methods can be used to find local optima. In general the objective is not differentiable at points where $\min_k \phi_k(v/\|v\||\mathcal{X})$ is satisfied by multiple indices $k$, and where the order statistics corresponding to the minimum coincide. Considerable work has been done on optimising such *non-smooth* objectives, however it has been observed that the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm tends to perform well on many non-smooth functions (Lewis and Overton, 2013). Indeed, applying this method to optimising $\min_k \phi_k(v/\|v\||\mathcal{X})$ did not fail to find a local optimum in any examples considered for this work.

The Lipschitz continuity of the objective $\min_k \phi_k(v/\|v\||\mathcal{X})$ is now briefly discussed. Consider $v \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ and $u \in \mathbb{R}^d$, $u \neq -v$. The order statistics of the marginal data set, $v \cdot \mathcal{X}$, are Lipschitz continuous as functions of $v$ and so $\exists L > 0$ s.t. for any $i, j \in \{1, ..., n\}$,

$$\left|(x_i^v - x_j^v) - (x_i^{v+u} - x_j^{v+u})\right| \leq L\|u\|.$$

It is therefore straightforward to show that,

$$\left|\frac{x_i^v - x_j^v}{\|v\|} - \frac{x_i^{v+u} - x_j^{v+u}}{\|v+u\|}\right| \leq (L + \text{Diam}(\mathcal{X}))\frac{\|u\|}{\|v\|}$$

$$\Rightarrow \left|\exp\left(-\frac{|x_i^v - x_j^v|}{\|v\|\sigma}\right) - \exp\left(-\frac{|x_i^{v+u} - x_j^{v+u}|}{\|v+u\|\sigma}\right)\right| \leq (L + \text{Diam}(\mathcal{X}))\frac{\|u\|}{\|v\|\sigma},$$

since $\exp(-x)$ has Lipschitz constant 1. From this it can be shown that for $\|u\| \leq 1$, $\exists H > 0$ s.t. each $\phi_k$ satisfies

$$\left|\phi_k\left(\frac{v}{\|v\|}|\mathcal{X}\right) - \phi_k\left(\frac{v+u}{\|v+u\|}|\mathcal{X}\right)\right| \leq H\frac{\|u\|}{\|v\|},$$

and hence the objective $\min_k \phi_k(v/\|v\||\mathcal{X})$ is Lipschitz continuous in $v$ over the set $\{v \in \mathbb{R}^d | \|v\| \geq 1\}$.

In what follows, derivations of the gradient of $\min_k \phi_k(v/\|v\||\mathcal{X})$ are presented for those $v$ where the minimum is unique, and where order statistics corresponding to the minimum do not coincide. The chain rule decomposition below is used.

$$D_v\phi_k\left(\frac{v}{\|v\|}|\mathcal{X}\right) = D_{\frac{v}{\|v\|}\cdot\mathcal{X}}\text{NCut}(\{x_1^{\frac{v}{\|v\|}},...,x_k^{\frac{v}{\|v\|}}\},\{x_{k+1}^{\frac{v}{\|v\|}},...,x_n^{\frac{v}{\|v\|}}\})D_v\frac{v}{\|v\|}\cdot\mathcal{X}, \tag{4.21}$$

where $D_{\cdot}\cdot$ is the differential operator. First observe that,

$$D_v\frac{v}{\|v\|}\cdot\mathcal{X} = \frac{1}{\|v\|}\mathcal{X} - \frac{1}{\|v\|^3}\mathcal{X}(vv^\top),$$

where here $\mathcal{X}$ is treated as a matrix in $\mathbb{R}^{n\times d}$ in which the $i$-th row corresponds to $x_i$. Therefore,

$$(D_v \frac{v}{\|v\|} \cdot \mathcal{X}) \cdot v = \mathbf{0}$$
$$\Rightarrow \left\| v - \delta D_v \phi_k \left( \frac{v}{\|v\|} | \mathcal{X} \right) \right\| \geq \|v\|$$

for any $\delta > 0$. This means that iterations in a simple gradient descent method have increasing norm. We have also observed that this behaviour transfers to the BFGS algorithm, even though the approximate Hessian is included in the search direction. Thus the constraint $\|v\| \geq 1$ in the optimisation (4.19) is inactive provided the initial solution has norm at least 1. The optimisation can therefore be treated as unconstrained in practice.

What remains is to address the first component in the chain rule decomposition (4.21). For ease of notation, let $x_i^{\hat{v}} = x_i^{v/\|v\|}$. Define,

$$E := \left( \exp\left( \frac{x_1^{\hat{v}} - x_k^{\hat{v}}}{\sigma} \right), \exp\left( \frac{x_2^{\hat{v}} - x_k^{\hat{v}}}{\sigma} \right), \ldots, \exp\left( \frac{x_n^{\hat{v}} - x_k^{\hat{v}}}{\sigma} \right) \right).$$

Then for $l \leq k$, and using a similar derivation to those in Subsection 3.1, one has the following,

$$\frac{\partial}{\partial x_l^{\hat{v}}} \sum_{i=1}^{k} \sum_{j=k+1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}}) = \frac{1}{\sigma} E_l \left( CS\left(\frac{1}{E}\right)_n - CS\left(\frac{1}{E}\right)_k \right).$$

In addition,

$$\frac{\partial}{\partial x_l^{\hat{v}}} \sum_{i=1}^{k} \sum_{j=1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}}) = -\frac{2}{\sigma} \frac{CS(E)_{l-1}}{E_l} + \frac{2}{\sigma} E_l \left( CS\left(\frac{1}{E}\right)_k - CS\left(\frac{1}{E}\right)_l \right)$$
$$+ \frac{1}{\sigma} E_l \left( CS\left(\frac{1}{E}\right)_n - CS\left(\frac{1}{E}\right)_k \right).$$

If $l = 1$ then the first term above is 0. In a similar way one finds,

$$\frac{\partial}{\partial x_l^{\hat{v}}} \sum_{i=k+1}^{n} \sum_{j=1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}}) = \frac{1}{\sigma} E_l \left( CS\left(\frac{1}{E}\right)_n - CS\left(\frac{1}{E}\right)_k \right),$$

and for $l > k$, the following,

$$\frac{\partial}{\partial x_l^{\hat{v}}} \sum_{i=1}^{k} \sum_{j=k+1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}}) = -\frac{1}{\sigma} \frac{CS(E)_k}{E_l}$$

$$\frac{\partial}{\partial x_l^{\hat{v}}} \sum_{i=1}^{k} \sum_{j=1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}}) = -\frac{1}{\sigma} \frac{CS(E)_k}{E_l}$$

$$\frac{\partial}{\partial x_l^{\hat{v}}} \sum_{i=k+1}^{n} \sum_{j=1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}}) = \frac{2}{\sigma} E_l \left( CS\left(\frac{1}{E}\right)_n - CS\left(\frac{1}{E}\right)_l \right) - \frac{2}{\sigma} \frac{CS(E)_{l-1} - CS(E)_k}{E_l} - \frac{1}{\sigma} \frac{CS(E)_k}{E_l}.$$

The partial derivative of $\mathrm{NCut}(\{x_1^{\hat{v}}, \ldots, x_k^{\hat{v}}\}, \{x_{k+1}^{\hat{v}}, \ldots, x_n^{\hat{v}}\})$ with respect to the $l$-th element of $\frac{v}{\|v\|} \cdot \mathcal{X}$ is given by

$$\frac{\partial}{\partial x_l^{\hat{v}}} \sum_{i=1}^{k} \sum_{j=k+1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}}) \left( \frac{1}{\sum_{i=1}^{k} \sum_{j=1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}})} + \frac{1}{\sum_{i=k+1}^{n} \sum_{j=1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}})} \right)$$

$$- \sum_{i=1}^{k} \sum_{j=k+1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}}) \left( \frac{\frac{\partial}{\partial x_l^{\hat{v}}} \sum_{i=1}^{k} \sum_{j=1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}})}{\left( \sum_{i=1}^{k} \sum_{j=1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}}) \right)^2} + \frac{\frac{\partial}{\partial x_l^{\hat{v}}} \sum_{i=k+1}^{n} \sum_{j=1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}})}{\left( \sum_{i=k+1}^{n} \sum_{j=1}^{n} K_\sigma(x_i^{\hat{v}}, x_j^{\hat{v}}) \right)^2} \right).$$

Therefore the collection of all partial derivatives can be computed in $\mathcal{O}(n)$ time, since each can be extracted in constant time from a collection of cumulative sum vectors.

## 3.3 Beyond Bi-partitioning

So far methodology for inducing a bipartition of a data set $\mathcal{X}$ using the minimum cut hyperplane has been discussed. This divisive procedure can be applied recursively to (subsets of) the data set to generate a binary partitioning tree. An indexing policy is required which assigns an ordering to the leaves of the tree, dictating which should be further partitioned at the next iteration. This process is repeated until a predetermined number of leaves results, and the data assigned to each leaf are interpreted as clusters. Pseudocode for a generic divisive clustering algorithm is given in Algorithm 1.

Two methods are considered. The first uses the procedure in Subsections 3.1 and 3.2 to induce bipartitions, and selects the leaf which has the

**Algorithm 2:** Divisive Clustering

Input: Data set $\mathcal{X}$, #clusters $K$, indexing function $I$, partitioning function $\Pi$

$\mathcal{C} \leftarrow \{\mathcal{X}\}$
**while** $|\mathcal{C}| < K$ **do**
  $C' \leftarrow \text{argmax}_{C \in \mathcal{C}}\ I(C)$
  $[C_1, C_2] \leftarrow \Pi(C')$
  $\mathcal{C} \leftarrow (\mathcal{C} \setminus \{C'\}) \cup \{C_1, C_2\}$
**end while**
**return** $\mathcal{C}$

smallest NCut value based on the optimal hyperplane. This divisive clustering algorithm will be referred to as the Normalised Cut Hyperplane, NCutH, approach. The second is based on the result in Lemma 24, which establishes the convergence of the minimum cut hyperplane to the maximum margin hyperplane, and repeatedly implements the procedure in Subsections 3.1 and 3.2 for a shrinking sequence of scaling parameters to find large margin separators. The NCut value for small $\sigma$ approaches 0 for all data sets, and so comparing the NCut values might not be reliable. Instead the simple indexing which selects the leaf containing the largest number of data to be partitioned at each iteration is used. This approach based on the asymptotic convergence of the minimum cut hyperplane will be referred to as $\text{NCutH}_0$.

The largest margin through a data set often only separates a few outlying points from the bulk of the data. For the $\text{NCutH}_0$ approach, each hyperplane is forced to induce a reasonably balanced partition of the data assigned to its respective leaf. This constraint is trivially incorporated by modifying the optimisation in (4.19) as,

$$\min_{v \| \|v\| \geq 1} \min_{k \in \{c, \ldots, n-c\}} \phi_k \left( \frac{v}{\|v\|} | \mathcal{X} \right), \tag{4.22}$$

where $c$ is a chosen parameter. The importance of including a balancing constraint for maximum margin clustering was also observed by Xu et al. (2004) and Zhang et al. (2009).

# 4 Experimental Results

In this section results from experiments using the proposed methods, NCutH and $\text{NCutH}_0$, are presented. Performance is compared with existing state-of-the-art methods for clustering on the following 15 benchmark data sets:

**Table 4.1:** Details of Benchmark Data Sets

| | $n$ | $d$ | $c$ |
|---|---|---|---|
| br. cancer[a] | 699 | 9 | 2 |
| ionosphere[a] | 351 | 33 | 2 |
| opt. digits[a] | 5618 | 64 | 10 |
| pen digits[a] | 10992 | 16 | 10 |
| voters[a] | 435 | 16 | 2 |
| image seg.[a] | 2309 | 19 | 7 |
| satellite[a] | 6435 | 36 | 6 |
| chart[a] | 600 | 60 | 6 |
| yeast[b] | 698 | 72 | 5 |
| smartphone[a] | 10929 | 561 | 12 |
| soybean[a] | 682 | 35 | 19 |
| dermatology[a] | 366 | 34 | 6 |
| glass[a] | 214 | 9 | 6 |
| isolet[a] | 6238 | 617 | 26 |
| parkinsons[a] | 195 | 22 | 2 |

[a] `https://archive.ics.uci.edu/ml/datasets.html`

[b] `http://genome-www.stanford.edu/cellcycle/`

Breast cancer Wisconsin original (br. cancer), Ionosphere, Optical recognition of handwritten digits (opt. digits), Pen based recognition of handwritten digits (pen digits), Congressional voting records (voters), Smart-phone based recognition of human activities and postural transitions (smartphone), Statlog image segmentation (image seg.), Statlog landsat satellite (satellite), Synthetic control charts (chart), Yeast cell cycle analysis (yeast), Soybean disease (soybean), Dermatology, Glass, Isolet and Parkinsons. Details of these data sets can be seen in Table 4.1, where $n$ is the number of data, $d$ is the number of dimensions and $c$ the number of classes.

## 4.1 Parameter Settings for NCutH and NCutH$_0$

The setting driving the performance of NCutH is the scaling parameter $\sigma$, while in the case of NCutH$_0$ it is the balancing constraint $c$ which has the greatest effect on performance. Within the divisive hierarchical clustering approach, these parameters are set at each iteration in the divisive procedure, and the values are determined relative to the subset of the data being partitioned.

The scaling parameter $\sigma$ relates to the overall scale of the data, and was set proportional to $\sqrt{\lambda_1}$, where $\lambda_1$ is the largest eigenvalue of the covariance matrix. This setting is equal to the standard deviation of the data pro-

jected along the first principal component. The analysis in Section 2 suggests setting $\sigma \propto n^{-1/(4+\epsilon)}$ for some $\epsilon > 0$, and $\epsilon$ is set to 1 for the experiments herein. To determine an appropriate constant of proportionality, a simple simulation study using Gaussian mixtures was performed. Simulations are extremely useful for determining appropriate values of tuning parameters as the true class labels are known, and so the clustering performance can be determined reliably. In this study Gaussian mixtures containing between 2 and 10 components, and in 5, 10 and 50 dimensions were simulated. The data arising from each component were assigned the same class label and these were compared with the cluster assignments made by NCutH for values of $\sigma$ in $\{\sigma_0, 5\sigma_0, 10\sigma_0, 50\sigma_0, 100\sigma_0, 200\sigma_0\}$ where $\sigma_0 = \sqrt{\lambda_1} n^{-1/5}$. Aside from $\sigma = \sigma_0$, the mean performance for all other values of $\sigma$ was approximately the same accross the range of dimensions and number of clusters. However for $\sigma = 50\sigma_0$ and $\sigma = 100\sigma_0$ the variability of performance was slightly lower. For the experiments presented below the value $\sigma = 100\sigma_0$ was used, but it is apparent from this study that performance is robust over a range of values. One observation made possible by this study is that the cluster sizes tend to be more balanced for larger values of $\sigma$, although this is not guaranteed.

The value of the balancing constraint parameter $c$ was set to $n/5$, which is equivalent to the setting used by Zhang et al. (2009) for maximum margin clustering when the cluster sizes are not balanced. Since this parameter offers a direct interpretation in terms of the minimum cluster size, prior information about the sizes of clusters can be incorprated into the method. To find large margin hyperplanes using the NCutH$_0$ approach, the scaling parameter was initialised at $100\sigma_0$, as above, and subsequently decreased by a factor of 5 at each iteration. This was repeated until convergence of the optimal hyperplane.

In addition, since the minimisation of NCut$(v, b)$ is non-convex, the initialisation of $v$ also plays a significant role. For initialisation of the vector $v$, the first principal component of the data was used. It is intuitively the case that directions with high data variability are likely to admit high quality hyperplanes for clustering, and so are likely to present promising initialisations for the proposed optimisation method.

## 4.2 Clustering Performance

The following benchmark algorithms were chosen for comparison with NCutH and NCutH$_0$:

1. $K$-means using the default implementation in `R`. Results from the best solution based on the $K$-means objective from 10 initialisations are reported.

2. Bisecting $K$-means (Bis.K-m) (Steinbach et al., 2000). The bisecting ver-

sion of $K$-means offers a useful comparison with the proposed method due to the similar model structure.

3. Normalised Spectral Clustering (SCn). Due to the high computation time required, when the number of data exceeds 1000 the approximation method given by Yan et al. (2009) was used.[1] A range of scaling parameters was used, and the highest performance is reported. Spectral clustering is a highly influential approach to clustering, and is based on the same normalised graph cut objective as the proposed method.

4. Iterative Support Vector Regression (iSVR). A state-of-the-art method for maximum margin clustering (Zhang et al., 2009).[2] To generate multiple clusters the same hierarchical method as for $\text{NCutH}_0$ is used, splitting the largest remaining cluster at each iteration.

5. Density enhanced principal direction divisive partitioning (dePDDP) (Tasoulis et al., 2010). An improvement of one of the earliest divisive hierarchical clustering algorithms using hypeprlanes, Principal Direction Divisive Partitioning. To make the comparison fair the correct number of clusters is supplied to the algorithm.

The algorithms are compared based on cluster *Purity* (Zhao and Karypis, 2004) and *V-Measure* (Rosenberg and Hirschberg, 2007). Both measures compare the true classes in the data with the cluster assignments made by an algorithm. They take values in $[0, 1]$ with higher values indicating superior performance. Purity is defined as the weighted average of the largest proportion of each cluster which is represented by a single class. V-Measure is defined as the harmonic mean of *homogeneity* and *completeness*. Homogeneity measures the conditional entropy of the class distribution within each cluster. Completeness is symmetric to homogeneity, and measures the conditional entropy of the cluster distribution within each class. V-Measure therefore captures both the extent to which each cluster can be associated with a single class, but also penalises cluster assignments which divide single classes between multiple clusters.

Tables 4.2 and 4.3 report the Purity and V-Measure respectively. The highest performance in each case is highlighted in bold, while - indicates that a clustering result could not be obtained in reasonable time. It is clear that no method is uniformly superior to all others, which is due to the vastly different natures of the data sets in terms of size, dimension, number and shape of clusters. It is also clear that the problems of clustering the various data sets differ in difficulty. Both NCutH and $\text{NCutH}_0$ obtained the highest performance at least as often as any of the competing methods, and in the case of

[1] The implementation available at `http://www.math.umassd.edu/~dyan/fasp.html` is employed

[2] Thanks to Dr. Kai Zhang for providing code for this method.

**Table 4.2:** 100 × Purity on Benchmark Data Sets. Highest Performance Highlighted in Bold.

| | NCutH | $\text{NCutH}_0$ | iSVR | dePDDP | K-means | Bis.K-m | SCn |
|---|---|---|---|---|---|---|---|
| br. cancer | 96.85 | 96.85 | 90.41 | 96.71 | 95.42 | 95.42 | **97.28** |
| ionosphere | 71.23 | 71.23 | 70.37 | 68.95 | 70.66 | 70.66 | **72.93** |
| opt. digits | 78.71 | **79.69** | 75.99 | 26.43 | 63.28 | 64.54 | 48.34 |
| pen digits | 77.98 | 76.83 | **78.25** | 56.73 | 72.23 | 72.09 | 41.71 |
| voters | 84.60 | 84.37 | 81.15 | **85.06** | 84.60 | 84.60 | 84.27 |
| image seg. | 62.19 | 63.79 | **64.62** | 29.32 | 59.95 | 58.55 | 55.91 |
| satellite | 74.00 | **75.76** | 68.33 | 75.01 | 74.11 | 73.92 | 68.75 |
| chart | 66.67 | **83.83** | 72.00 | 68.00 | 76.33 | 66.67 | 78.17 |
| yeast | 75.07 | 74.93 | 74.93 | **77.51** | 70.92 | 75.50 | 71.92 |
| smartphone | 68.81 | **72.47** | - | 39.06 | 63.81 | 64.51 | 50.14 |
| soybean | **80.79** | 67.45 | 71.85 | 60.56 | 67.74 | 78.01 | 72.87 |
| dermatology | **96.17** | 85.52 | 80.33 | 85.52 | 86.07 | 86.07 | 94.54 |
| glass | 54.21 | **58.88** | 51.40 | 48.13 | 54.21 | 53.27 | 51.40 |
| isolet | 51.67 | 59.17 | - | 21.88 | **60.05** | 52.07 | 33.71 |
| parkinsons | **75.38** | **75.38** | **75.38** | **75.38** | **75.38** | **75.38** | **75.38** |

V-Measure considerably more often. Much more importantly, however, there are very few examples in which NCutH and $\text{NCutH}_0$ are not among the best performing methods, whereas all competing algorithms fare poorly in multiple examples. This fact is better captured by *regret*. The regret of an algorithm on a specific data set is the difference between that algorithm's performance and the best performance from among all algorithms when applied to that data set. Figure 4.2 shows boxplots of the regret of each algorithm over all 15 benchmark data sets, where the additional red dots indicate the mean. Both the mean and median regret for Purity and V-Measure are substantially lower in the case of NCutH and $\text{NCutH}_0$ than for any other method. Because of their similar objectives, the comparisons between NCutH and SCn and between $\text{NCutH}_0$ and iSVR are arguably the most important. It is clear that the overall performance of the proposed methods is a substantial improvement over the existing methods.

## 4.3 Run Time Analysis

The analysis presented in Section 3 showed that the optimal NCut for a given $v$ can be computed in $\mathcal{O}(n \log n)$ time. The number of iterations in the BFGS algorithm may depend on $n$ in an unknown way, and so the overall complex-

**Table 4.3:** 100 × V-Measure on Benchmark Data Sets. Highest Performance Highlighted in Bold.

| | NCutH | $\text{NCutH}_0$ | iSVR | dePDDP | K-means | Bis.K-m | SCn |
|---|---|---|---|---|---|---|---|
| br. cancer | 78.80 | 78.68 | 55.34 | 77.99 | 71.59 | 71.59 | **82.45** |
| ionosphere | 13.49 | 13.49 | 12.64 | 9.82 | 12.50 | 12.50 | **20.67** |
| opt. digits | 71.62 | **74.65** | 71.49 | 28.52 | 61.30 | 61.68 | 50.01 |
| pen digits | 70.95 | **73.07** | 72.53 | 59.15 | 68.93 | 65.71 | 48.63 |
| voters | **42.82** | 42.39 | 33.59 | 39.25 | 42.30 | 42.30 | 40.48 |
| image seg. | 59.38 | **63.81** | 62.61 | 35.15 | 59.87 | 58.63 | 47.90 |
| satellite | 59.63 | **62.68** | 54.61 | 61.05 | 61.30 | 60.15 | 53.30 |
| chart | 79.65 | 77.05 | 66.41 | 77.75 | 77.43 | 76.07 | **82.36** |
| yeast | **59.92** | 59.01 | 53.91 | 55.80 | 52.55 | 57.96 | 51.30 |
| smartphone | **61.22** | 60.04 | - | 51.03 | 56.95 | 56.25 | 50.30 |
| soybean | **80.29** | 71.87 | 75.38 | 67.68 | 74.13 | 80.19 | 73.46 |
| dermatology | **93.56** | 83.82 | 76.84 | 87.39 | 86.33 | 86.00 | 90.29 |
| glass | **31.58** | 30.85 | 27.55 | 30.17 | 31.46 | 31.18 | 28.79 |
| isolet | 65.34 | 70.36 | - | 42.34 | **71.67** | 63.74 | 51.94 |
| parkinsons | **21.96** | **21.96** | 6.38 | 1.78 | 12.42 | 12.42 | 1.20 |

ity of the method is investigated empirically. Figure 4.3 shows the run time of NCutH for $n$ ranging from 100 to 10000. The run time of the overall algorithm increases with rate approximately $\mathcal{O}(n \log n)$, as might be expected from the analysis.

In addition the computation time of NCutH and $\text{NCutH}_0$ is compared with existing methods on the benchmark data sets considered before. Table 4.4 shows the results of this investigation. All methods were implemented in `R`. The default implementation of $K$-means was used. The iSVR method relies on the `e1071` package which implements the `libSVM` library. It is again important to highlight the comparisons between NCutH and SCn and between $\text{NCutH}_0$ and iSVR. In both cases the proposed methods are orders of magnitude faster than the existing methods. An important fact to note too is that the run time of both $K$-means and Bis.$K$-m is overestimated as the reported numbers are for 10 initialisations, since the performance based on a single intialisation can be highly variable. A single run of $K$-means is therefore of the order of 10 times faster than NCutH.

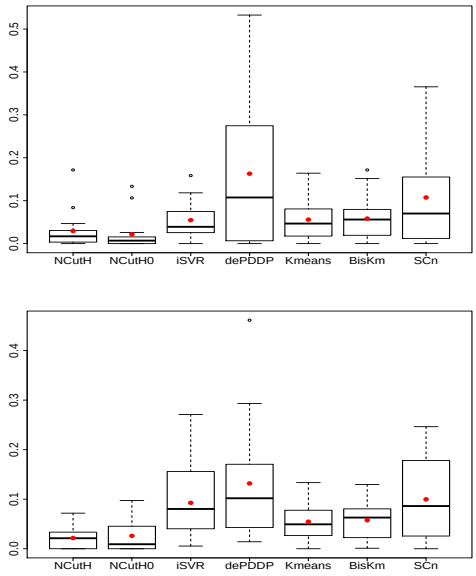



**Fig. 4.2:** Regret distributions of Purity (top) and V-Measure (bottom) across all 15 benchmark data sets.

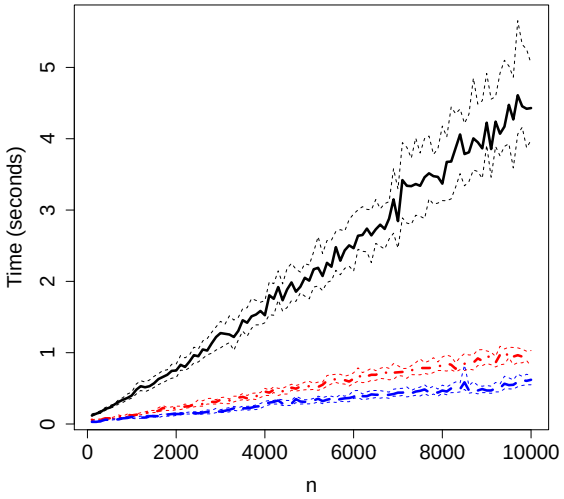



**Fig. 4.3:** Run time analysis from Gaussian mixtures. The plot shows the medians and interquartile ranges from 50 replications for each value of $n$. The number of clusters is fixed at 5.

5 Dimensions, 10 Dimensions, 50 Dimensions

**Table 4.4:** Run Time on Benchmark Data Sets (in Seconds)

| | NCutH | NCut$H_0$ | iSVR | dePDDP | K-means | Bis.K-m | SCn |
|---|---|---|---|---|---|---|---|
| br. cancer | 0.01 | 0.23 | 1.81 | 0.01 | 0.05 | 0.05 | 0.67 |
| ionosphere | 0.01 | 0.16 | 2.47 | 0.02 | 0.03 | 0.05 | 0.20 |
| opt. digits | 4.40 | 12.01 | 496.14 | 0.19 | 1.97 | 4.04 | 28.37 |
| pen digits | 1.55 | 6.37 | 347.77 | 0.12 | 0.79 | 2.35 | 24.29 |
| voters | 0.02 | 0.16 | 1.09 | 0.01 | 0.03 | 0.03 | 0.27 |
| image seg. | 0.39 | 1.22 | 18.19 | 0.06 | 0.17 | 0.52 | 6.15 |
| satellite | 0.81 | 3.26 | 203.30 | 0.11 | 0.95 | 2.54 | 21.67 |
| chart | 0.20 | 0.78 | 14.32 | 0.12 | 0.11 | 0.40 | 1.61 |
| yeast | 0.31 | 1.06 | 21.82 | 0.06 | 0.19 | 0.55 | 2.34 |
| smartphone | 81.04 | 217.38 | - | 14.71 | 82.29 | 103.36 | 1915.10 |
| soybean | 0.47 | 1.48 | 12.21 | 0.15 | 0.08 | 0.29 | 0.19 |
| dermatology | 0.13 | 0.56 | 1.17 | 0.06 | 0.03 | 0.14 | 0.31 |
| glass | 0.06 | 0.31 | 0.32 | 0.02 | 0.00 | 0.03 | 0.08 |
| isolet | 80.86 | 220.19 | - | 16.43 | 54.06 | 77.95 | 1511.53 |
| parkinsons | 0.01 | 0.10 | 0.39 | 0.01 | 0.01 | 0.01 | 0.05 |

# 5 Conclusion

This paper presents a novel hyperplane based method for clustering. The optimal hyperplane is that which minimises the normalised graph cut measured across it. The asymptotic properties for an incrceasing sample assumed to arise from an underlying probability distribution are established, showing that the minimum NCut hyperplane tends to have low density along it, as well as separate modes of the underlying probability density. Both of these are key features in the context of data clustering. In the finite sample case, the minimum cut hyperplane is asymptotically connected with the maximum margin hyperplane through the data, as the scaling parameter decreases to zero. Applying the proposed method for a shrinking sequence of scaling parameters also therefore provides a new method for finding large margin separators. Exploiting the trivial factorisation of the exponential function allows for the derivation of a linear time method for extracting the optimal normalised cut of a sorted univariate marginal data set. The overall algorithm has empirical time complexity which is log-linear in the number of data, allowing for the application to larger data sets than is generally true of graph based partitioning methods.

In experiments on a large collection of benchmark data sets the proposed method showed superior performance to a collection of influential clustering

methods from the literature in the majority of examples.

## Appendix. Proofs

**Proof of Proposition 20** The result follows almost immediately by observing that if $p_i$ is elliptical then the marginal density $p_i^v$ is unimodal and the mode of $p_i^v$ is the projection of the mode of $p_i$ onto $v$. Therefore, if a hyperplane $H(v,b)$ is such that all of the modes of the components $p_i$ lie in $H(v,b)^+$, then the marginal density $p^v$ is non-increasing on $[b,\infty)$, and hence $H(v,b)$ cannot intersect the modes of the marginal density $p^v$. The case where all modes lie in $H(v,b)^-$ is the analogous. This proves the result. ■

The proofs of Lemmas 21 and 22 borrow some ideas from Narayanan et al. (2006). In particular, the use of the generalisation of McDiarmid's inequality given by Kutin (2002) is useful.

**Definition** (Kutin, 2002, Definition 1.6) Let $\Omega_1,...,\Omega_m$ be probability spaces. Let $\Omega = \prod_{k=1}^m \Omega_k$, and let $X$ be a random variable on $\Omega$. We say $X$ is *strongly difference-bounded* by $(b,c,\delta)$ if the following holds: there is a "bad" subset $B \subset \Omega$, where $\delta = \mathbb{P}(\omega \in B)$. If $\omega, \omega' \in \Omega$ differ only in the $k$-th coordinate, and $\omega \notin B$, then

$$|X(\omega) - X(\omega')| \leq c.$$

Furthermore, for any $\omega$ and $\omega'$ differing only in the $k$-th coordinate,

$$|X(\omega) - X(\omega')| \leq b.$$

**Theorem 25** *(Kutin, 2002, Theorem 3.6) Let $\Omega_1,...,\Omega_m$ be probability spaces. Let $\Omega = \prod_{k=1}^m \Omega_k$, and let $X$ be a random variable on $\Omega$ which is strongly difference-bounded by $(b,c,\delta)$. Assume $b \geq c > 0$. Let $\mu = \mathbb{E}[X]$. Then, for any $\tau > 0$,*

$$\mathbb{P}(|X-\mu| \geq \tau) \leq 2\left(\exp\left(-\frac{\tau^2}{8mc^2}\right) + \frac{mb\delta}{c}\right).$$

**Proof of Lemma 21** Let $q = \mathbb{P}(X \in H(v,b)^-)$, then $q \in (0,1)$ since $H(v,b)$ intersects the interior of $S$. For $n \in \mathbb{N}$ define,

$$B_n := \Big\{ \mathcal{X} = \{x_1,...,x_n\} \in (\mathbb{R}^d)^n \Big| \\ |\mathcal{X} \cap H(v,b)^-| \leq \frac{n}{2}q \text{ or } |\mathcal{X} \cap H(v,b)^+| \leq \frac{n}{2}(1-q) \Big\}.$$

$B_n$ is defined to represent the set of potential random samples of size $n$ arising from independent realisations of $X$ in which either the number of data falling

in $H(v,b)^+$ or the number falling in $H(v,b)^-$ is less than or equal to half the expected number. Using Hoeffding's inequality one can show that,

$$\mathbb{P}(\mathcal{X} \in B_n) \leq \exp\left(-2\frac{(nq - \frac{n}{2}q)^2}{n}\right) + \exp\left(-2\frac{(n(1-q) - \frac{n}{2}(1-q))^2}{n}\right).$$

Now, for $\mathcal{X} = \{x_1, ..., x_n\} \notin B_n$, define $\mathcal{X}'$ to be $\mathcal{X}$ except one of the elements is replaced with another independent realisation of $X$. Define $C = \mathcal{X} \cap H(v,b)^-$, $C' = \mathcal{X}' \cap H(v,b)^-$. Then,

$$\left|\frac{\sum_{x\in C}\sum_{y\in\mathcal{X}\setminus C} K_\sigma^v(x,y)}{|C|(n-|C|)} - \frac{\sum_{x\in C'}\sum_{y\in\mathcal{X}'\setminus C'} K_\sigma^v(x,y)}{|C'|(n-|C'|)}\right| \leq \max\left\{\frac{1}{|C|-1}, \frac{1}{n-|C|-1}\right\}.$$

Since $\mathcal{X} \notin B_n$ we know that $|C| > \frac{n}{2}q$ and $n - |C| > \frac{n}{2}(1-q)$, and hence this right hand side can be replaced with $\frac{2}{n\min\{q,1-q\}}$.

Now, if $\mathcal{X}$ and $\mathcal{X}'$ are again defined as above except $\mathcal{X}$ is allowed to lie in $B_n$, then we instead have

$$\left|\frac{\sum_{x\in C}\sum_{y\in\mathcal{X}\setminus C} K_\sigma^v(x,y)}{|C|(n-|C|)} - \frac{\sum_{x\in C'}\sum_{y\in\mathcal{X}'\setminus C'} K_\sigma^v(x,y)}{|C'|(n-|C'|)}\right| \leq 1.$$

The random variable $\xi := \frac{1}{n\sigma^2}\text{RatioCut}(v,b|\mathcal{X})$ is therefore strongly difference bounded by,

$$\left(\frac{1}{\sigma^2}, \frac{2}{n\min\{q,1-q\}\sigma^2}, \mathbb{P}(\mathcal{X} \in B_n)\right).$$

By the generalisation of McDiarmid's inequality we thus have,

$$\mathbb{P}\left(|\xi - \mathbb{E}[\xi]| > \gamma\right) \leq 2\exp\left(-\frac{\gamma^2 n\min\{q,1-q\}^2\sigma^4}{32}\right) + n^2\min\{q,1-q\}\mathbb{P}(\mathcal{X} \in B_n).$$

Now, let $Y, Z$ be random variables with the same distribution as $v \cdot X$. Then we can write,

$$\mathbb{E}[\xi] = \mathbb{E}\left[\frac{1}{\sigma^2}K_\sigma(Z,Y)\middle|Z < b, Y \geq b\right].$$

Therefore,

$$\begin{aligned}
\mathbb{E}[\xi] &= \int_-_{-\infty}^{b}\int_b^\infty \frac{1}{\sigma^2}K_\sigma(z,y)\frac{p^v(z)}{\mathbb{P}(v\cdot X < b)}\frac{p^v(y)}{\mathbb{P}(v\cdot X \geq b)}dydz \\
&= \int_{-\infty}^b\int_b^\infty \frac{1}{\sigma^2}\exp\left(\frac{z-y}{\sigma}\right)\frac{p^v(z)}{\mathbb{P}(v\cdot X < b)}\frac{p^v(y)}{\mathbb{P}(v\cdot X \geq b)}dydz \\
&= \int_{-\infty}^b \frac{1}{\sigma}\exp\left(\frac{z-b}{\sigma}\right)\frac{p^v(z)}{\mathbb{P}(v\cdot X < b)}\int_b^\infty \frac{1}{\sigma}\exp\left(\frac{b-y}{\sigma}\right)\frac{p^v(y)}{\mathbb{P}(v\cdot X \geq b)}dydz.
\end{aligned}$$

Now, the quantity $\int_{-\infty}^{b} \frac{1}{\sigma} \exp\left(\frac{z-b}{\sigma}\right) \frac{p^v(z)}{\mathbb{P}(v\cdot X<b)} dz$ is the expected value of a kernel estimate of the density of the random variable $v\cdot X$ truncated above at $b$ evaluated at the point $b$, and using a Laplace kernel with the reflection method at the boundary. Therefore,

$$\int_{-\infty}^{b} \frac{1}{\sigma} \exp\left(\frac{z-b}{\sigma}\right) \frac{p^v(z)}{\mathbb{P}(v\cdot X<b)} dz = \frac{p^v(b)}{\mathbb{P}(v\cdot X<b)} + \mathcal{O}(\sigma).$$

Similarly,

$$\int_{b}^{\infty} \frac{1}{\sigma} \exp\left(\frac{b-y}{\sigma}\right) \frac{p^v(y)}{\mathbb{P}(v\cdot X\geq b)} dy = \frac{p^v(b)}{\mathbb{P}(v\cdot X\geq b)} + \mathcal{O}(\sigma).$$

Therefore,

$$\mathbb{E}[\xi] = \frac{p^v(b)^2}{\mathbb{P}(v\cdot X<b)\mathbb{P}(v\cdot X\geq b)} + \mathcal{O}(\sigma).$$

In all, there exist $D_1, D_2 > 0$ s.t. for large $n$ and with probability at least $1 - D_1 \exp\left(-\gamma^2\sigma^4 n\right)$ we have,

$$|\xi - \mathbb{E}[\xi]| \leq \gamma \text{ and } \left| \frac{p^v(b)}{\mathbb{P}(v\cdot X<b)\mathbb{P}(v\cdot X\geq b)} - \mathbb{E}[\xi] \right| \leq D_2\sigma.$$

Setting $\gamma \in \mathcal{O}(\sigma^{\epsilon/2})$ and allowing $\sigma \to 0$ appropriately as $n \to \infty$ therefore gives the result. ■

**Proof of Lemma 22** Let $\mathcal{X}$ be a sample of realisations of $X$ of size $n$. Define the following,

$$\begin{aligned}
\eta_- &= \frac{1}{2\sigma} \frac{\sum_{x\in\mathcal{X}\cap H(v,b)^-} \sum_{y\in\mathcal{X}} K^v_\sigma(x,y)}{|\mathcal{X}\cap H(v,b)^-|(n-1)} \\
&= \frac{1}{2\sigma} \frac{\sum_{x\in\mathcal{X}\cap H(v,b)^-} \sum_{y\in\mathcal{X}, y\neq x} K^v_\sigma(x,y)}{|\mathcal{X}\cap H(v,b)^-|(n-1)} + \frac{1}{2\sigma(n-1)} \\
\eta_+ &= \frac{1}{2\sigma} \frac{\sum_{x\in\mathcal{X}\cap H(v,b)^+} \sum_{y\in\mathcal{X}} K^v_\sigma(x,y)}{|\mathcal{X}\cap H(v,b)^+|(n-1)} \\
&= \frac{1}{2\sigma} \frac{\sum_{x\in\mathcal{X}\cap H(v,b)^+} \sum_{y\in\mathcal{X}, y\neq x} K^v_\sigma(x,y)}{|\mathcal{X}\cap H(v,b)^+|(n-1)} + \frac{1}{2\sigma(n-1)},
\end{aligned}$$

where we adopt the convention $\frac{0}{0} = 0$. Similar to the previous proof it can be shown that $\eta_-$ and $\eta_+$ are strongly difference bounded by $(\frac{1}{2\sigma}, \frac{1}{n\min\{q,1-q\}\sigma}, \mathbb{P}(\mathcal{X}\in B_n))$, with $q$ and $B_n$ defined as before. Thus for $\gamma > 0$ both $\mathbb{P}(|\eta_- - \mathbb{E}[\eta_-]| > \gamma)$ and $\mathbb{P}(|\eta_+ - \mathbb{E}[\eta_+]| > \gamma)$ are bounded above by

$$2\exp\left(-\frac{\gamma^2 n \min\{q,1-q\}^2\sigma^2}{8}\right) + n^2 \min\{q,1-q\}\mathbb{P}(\mathcal{X}\in B_n).$$

Now, again if we consider random variables $Y, Z$ with the same distribution as $v \cdot X$, then we can write

$$\mathbb{E}[\eta_-] = \mathbb{E}\left[\frac{1}{2\sigma} K_\sigma(Y,Z) | Y < b, Z \neq Y\right] + \frac{1}{2\sigma(n-1)}$$
$$\mathbb{E}[\eta_+] = \mathbb{E}\left[\frac{1}{2\sigma} K_\sigma(Y,Z) | Y \geq b, Z \neq Y\right] + \frac{1}{2\sigma(n-1)}.$$

Thus,

$$\begin{aligned}\mathbb{E}[\eta_-] &= \int_{-\infty}^{b}\int_{-\infty}^{\infty} \frac{1}{2\sigma} K_\sigma(y,z) \frac{p^v(y)}{\mathbb{P}(v \cdot X < b)} p^v(z) dz dy \\ &\quad + \frac{1}{2\sigma(n-1)} \\ &= \frac{1}{q}\int_{-\infty}^{b} p^v(y) \int_{-\infty}^{\infty} \frac{1}{2\sigma} \exp\left(-\frac{|y-z|}{\sigma}\right) p^v(z) dz dy \\ &\quad + \frac{1}{2\sigma(n-1)},\end{aligned}$$

where $\int_{-\infty}^{\infty} \frac{1}{2\sigma} \exp\left(-\frac{|y-z|}{\sigma}\right) p^v(z) dz$ is the expected value of a kernel estimate of the density of $p^v$ at $y$ using the Laplace kernel. Therefore,

$$\mathbb{E}[\eta_-] = \frac{1}{q}\int_{-\infty}^{b} p^v(y)^2 dy + \mathcal{O}(\sigma + (\sigma n)^{-1}).$$

Similarly,

$$\mathbb{E}[\eta_+] = \frac{1}{1-q}\int_{b}^{\infty} p^v(y)^2 dy + \mathcal{O}(\sigma + (\sigma n)^{-1}).$$

With $\xi$ defined as before, we therefore have

$$\begin{aligned}\mathbb{E}[\xi]\left(\frac{1-q}{\mathbb{E}[\eta_-]} + \frac{q}{\mathbb{E}[\eta_+]}\right) &= p^v(b)^2 \left(\frac{1}{\int_{-\infty}^{b} p_v(x)^2 dx} + \frac{1}{\int_{b}^{\infty} p_v(x)^2 dx}\right) \\ &\quad + \mathcal{O}(\sigma + (\sigma n)^{-1}).\end{aligned}$$

Now consider the following,

$$\begin{aligned}
&\mathbb{P}\left(\left|\frac{\xi(1-q)}{\eta_-}-\frac{2}{\sigma}\frac{\sum_{x\in\mathcal{X}\cap H(v,b)^-}\sum_{y\in\mathcal{X}\cap H(v,b)^+}K_\sigma^v(x,y)}{\sum_{x\in\mathcal{X}\cap H(v,b)^-}\sum_{y\in\mathcal{X}}K_\sigma^v(x,y)}\right|>\gamma\right)\\
&=\mathbb{P}\left(\frac{2}{\sigma}\frac{\sum_{x\in\mathcal{X}\cap H(v,b)^-}\sum_{y\in\mathcal{X}\cap H(v,b)^+}K_\sigma^v(x,y)}{\sum_{x\in\mathcal{X}\cap H(v,b)^-}\sum_{y\in\mathcal{X}}K_\sigma^v(x,y)}\left|\frac{(n-1)(1-q)}{|\mathcal{X}\cap H(v,b)^+|}-1\right|>\gamma\right)\\
&\leq\mathbb{P}\left(\frac{2}{\sigma}\left|\frac{(n-1)(1-q)}{|\mathcal{X}\cap H(v,b)^+|}-1\right|>\gamma\middle||\mathcal{X}\cap H(v,b)^-|\geq 1\right)\\
&\leq\exp\left(-2n(1-q)^2\left(\frac{\sigma\gamma/2}{1+\sigma\gamma/2}\right)^2\right)+\exp\left(-2\frac{(1-q)^2}{n}\left(\frac{n\sigma\gamma/2-1}{1-\sigma\gamma/2}\right)^2\right)
\end{aligned}$$

using, as before, tail bounds for the binomial distribution. A similar inequality holds for $\frac{\xi q}{\eta_+}$.

Putting this all together, and setting $\gamma=\sigma^{\epsilon/2}$ and allowing $\sigma\to 0$ as $n\to\infty$ s.t. $n\sigma^{4+\epsilon}\to\infty$, we have the following. There exist constants $D_1,D_2,D_3,D_4,D_5>0$ s.t. for large $n$ the following hold,

$$\begin{aligned}
\mathbb{P}(|\xi-\mathbb{E}[\xi]|<\gamma)&\geq 1-D_1\exp\left(-\gamma^2\sigma^4 n\right)\\
\mathbb{P}(|\eta_--\mathbb{E}[\eta_-]|<\gamma)&\geq 1-D_2\exp\left(-\gamma^2\sigma^2 n\right)\\
\mathbb{P}(|\eta_+-\mathbb{E}[\eta_+]|<\gamma)&\geq 1-D_3\exp\left(-\gamma^2\sigma^2 n\right)\\
\mathbb{P}\left(\left|\frac{\xi(1-q)}{\eta_-}-\frac{2}{\sigma}\frac{\sum_{x\in\mathcal{X}\cap H(v,b)^-}\sum_{y\in\mathcal{X}\cap H(v,b)^+}K_\sigma^v(x,y)}{\sum_{x\in\mathcal{X}\cap H(v,b)^-}\sum_{y\in\mathcal{X}}K_\sigma^v(x,y)}\right|<\gamma\right)&\geq 1-D_4\exp\left(-n\sigma^2\gamma^2\right)\\
\mathbb{P}\left(\left|\frac{\xi q}{\eta_+}-\frac{2}{\sigma}\frac{\sum_{x\in\mathcal{X}\cap H(v,b)^-}\sum_{y\in\mathcal{X}\cap H(v,b)^+}K_\sigma^v(x,y)}{\sum_{x\in\mathcal{X}\cap H(v,b)^+}\sum_{y\in\mathcal{X}}K_\sigma^v(x,y)}\right|<\gamma\right)&\geq 1-D_5\exp\left(-n\sigma^2\gamma^2\right).
\end{aligned}$$

Therefore, there exist constants $E_1,E_2>0$ s.t. with probability at least $1-$

$E_1\exp(-\gamma^2\sigma^4 n)$ we have

$$\left|\frac{2}{\sigma}\sum_{\substack{x\in\mathcal{X}\cap H(v,b)^-\\ y\in\mathcal{X}\cap H(v,b)^+}} K_\sigma^v(x,y)\left(\frac{1}{\sum_{\substack{x\in\mathcal{X}\cap H(v,b)^-\\ y\in\mathcal{X}}} K_\sigma^v(x,y)}\right.\right.$$
$$\left.\left.+\frac{1}{\sum_{\substack{x\in\mathcal{X}\cap H(v,b)^+\\ y\in\mathcal{X}}} K_\sigma^v(x,y)}\right)-\mathbb{E}[\xi]\left(\frac{1-q}{\mathbb{E}[\eta_-]}+\frac{q}{\mathbb{E}[\eta_+]}\right)\right|<\gamma E_2.$$

Combining this with the fact that

$$\mathbb{E}[\xi]\left(\frac{1-q}{\mathbb{E}[\eta_-]}+\frac{q}{\mathbb{E}[\eta_+]}\right)=p^v(b)^2\left(\frac{1}{\int_{-\infty}^b p_v(x)^2dx}+\frac{1}{\int_b^\infty p_v(x)^2dx}\right)$$
$$+\mathcal{O}(\sigma+(\sigma n)^{-1}),$$

gives the result. ■

The following technical result is useful for proving Proposition 23

**Proposition 26** *Let $Y$ be a univariate random variable with distribution $F$, density $f$ and support $S$. Assume that $f$ is continuous and has finitely many modes. Then $\exists z<\sup S$ s.t.,*

$$\frac{f(x)}{\int_x^\infty f(y)^2dy}\geq\frac{1}{1-F(x)},$$

*for all $x\in[z,\sup S)$.*

**Proof** Since $f$ is continuous and has finitely many modes $\exists z\in Int(S)$ s.t. $f$ is non-increasing on $[z,\sup S)$. For $x\in[z,\sup S)$, and integrating by parts, consider

$$\begin{aligned}\int_x^\infty f(y)^2dy&=\int_x^\infty f(y)dF(y)\\&=[F(y)f(y)]_x^\infty-\int_x^\infty F(y)df(y)\\&\leq[F(y)f(y)]_x^\infty-\int_x^\infty df(y)\\&=-[f(y)(1-F(y))]_x^\infty\\&=f(x)(1-F(x)).\end{aligned}$$

A simple rearranging provides the result. ■

**Proof of Proposition 23** Take $v \in \mathbb{B}^d$ and let $S^v$ be the support of $v \cdot X$. Then $p^v$ is continuous with finitely many modes, and so by Proposition 26 $\exists z^v < \sup S$ s.t.

$$\frac{p^v(x)}{\int_x^{\sup S^v} p^v(y)^2 dy} \geq \frac{1}{1 - P^v(x)},$$

for all $x \in [z^v, \sup S)$. For such $x$ consider,

$$\begin{aligned} p^v(x)^2 \left( \frac{1}{\int_{\inf S^v}^x p^v(y)^2 dy} + \frac{1}{\int_x^{\sup S^v} p^v(y)^2 dy} \right) &\geq \frac{p^v(x)^2}{\int_x^{\sup S^v} p^v(y)^2 dy} \\ &\geq \frac{p^v(x)^2}{p^v(x)(1 - P^v(x))} \\ &= \frac{p^v(x)}{(1 - P^v(x))}. \end{aligned}$$

This is simply the hazard function for the random variable $v \cdot X$, which is unbounded by assumption. Therefore $\exists b^v < \sup S$ s.t. $p^v(b)^2 (1/\int_{\inf S^v}^b p^v(y)^2 dy + 1/\int_b^{\sup S^v} p^v(y)^2 dy)$ is non-decreasing in $b$ over $[b^v, \sup S^v)$, and is strictly increasing on part of this domain. By symmetry the same holds at the lower end of the support of $v \cdot X$. Therefore for each $v$ there is a $b$ which minimises $p^v(b)^2 (1/\int_{\inf S^v}^b p^v(y)^2 dy + 1/\int_b^{\sup S^v} p^v(y)^2 dy)$, since it is continuous and increasing outside a compact set. Notice also that since $p$ is continuous, the value $\min_b p^v(b)^2 (1/\int_{\inf S^v}^b p^v(y)^2 dy + 1/\int_b^{\sup S^v} p^v(y)^2 dy)$ is continuous in $v$, and since $\mathbb{B}^d$ is compact there exists a minimum in $(v, b) \in \mathbb{B}^d \times \mathbb{R}$. ■

**Proof of Lemma 24** Take $v \in \mathbb{B}^d$. Assume without loss of generality that the marginal data set $v \cdot \mathcal{X}$ is sorted in non-decreasing order, i.e., $v \cdot x_1 \leq v \cdot x_2 \leq ... \leq v \cdot x_n$. For $w \in \{1, ..., n-1\}$ observe,

$$\begin{aligned} \exp\left( \frac{v \cdot x_w - v \cdot x_{w+1}}{\sigma} \right) &\leq \sum_{i=1}^{w} \sum_{j=w+1}^{n} \exp\left( \frac{v \cdot x_i - v \cdot x_j}{\sigma} \right) \\ &\leq w(n-w) \exp\left( \frac{v \cdot x_w - v \cdot x_{w+1}}{\sigma} \right). \end{aligned}$$

Similarly,

$$\begin{aligned} 1 &\leq \sum_{i=1}^{w} \sum_{j=1}^{n} \exp\left( -\frac{|v \cdot x_i - v \cdot x_j|}{\sigma} \right) \leq nw \\ 1 &\leq \sum_{i=w+1}^{n} \sum_{j=1}^{n} \exp\left( -\frac{|v \cdot x_i - v \cdot x_j|}{\sigma} \right) \leq n(n-w). \end{aligned}$$

Therefore, for $b \in (v \cdot x_w, v \cdot x_{w+1}]$ we have

$$\frac{1}{w(n-w)} \exp\left(\frac{v \cdot x_w - v \cdot x_{w+1}}{\sigma}\right) \leq \text{NCut}(v, b | \mathcal{X}) \leq 2w(n-w) \exp\left(\frac{v \cdot x_w - v \cdot x_{w+1}}{\sigma}\right).$$

Take $\epsilon > 0$. It can be shown (Pavlidis et al., 2015) that $\exists m_\epsilon > 0$ s.t. for all $(w,c) \in \mathbb{B}^d \times \mathbb{R}$, $\min\{\|(w,c) - (v_m, b_m)\|, \|(w,c) - (-v_m, -b_m)\|\} > \epsilon \Rightarrow \text{margin}H(w,c) < \text{margin}H(v_m, b_m) - m_\epsilon$.

Now, for $\sigma > 0$ let $b$ be the midpoint of the largest gap between consecutive elements of $v_\sigma \cdot \mathcal{X}$. Suppose $\text{margin}H(v_\sigma, b) < \text{margin}H(v_m, b_m) - m_\epsilon$. Let $i_\sigma = |(v_\sigma \cdot \mathcal{X}) \cap (-\infty, b)|$, $i_m = |(v_m \cdot \mathcal{X}) \cap (-\infty, b_m)|$. Then,

$$\begin{aligned} \frac{1}{n^2} \exp\left(\frac{(v_\sigma \cdot \mathcal{X})_{(i_\sigma)} - (v_\sigma \cdot \mathcal{X})_{(i_\sigma+1)}}{\sigma}\right) &\leq \text{NCut}(v_\sigma, b_\sigma) \\ &\leq \text{NCut}(v_m, b_m) \\ &\leq 2n^2 \exp\left(\frac{(v_m \cdot \mathcal{X})_{(i_m)} - (v_m \cdot \mathcal{X})_{(i_m+1)}}{\sigma}\right), \end{aligned}$$

Therefore,

$$\begin{aligned} \sigma &\geq \frac{(v_m \cdot \mathcal{X})_{(i_m+1)} - (v_m \cdot \mathcal{X})_{(i_m)} - ((v_\sigma \cdot \mathcal{X})_{(i_\sigma+1)} - (v_\sigma \cdot \mathcal{X})_{(i_\sigma)})}{\log(2n^4)} \\ &> \frac{2m_\epsilon}{\log(2n^4)}. \end{aligned}$$

We therefore have,

$$\begin{aligned} \sigma \leq \frac{2m_\epsilon}{\log(2n^4)} &\Rightarrow \max_{b \in \mathbb{R}} \text{margin}H(v_\sigma, b) \geq \text{margin}H(v_m, b_m) - m_\epsilon \\ &\Rightarrow \exists b \in \mathbb{R} \text{ s.t. } \min\{\|(v_\sigma, b) - (v_m, b_m)\|, \|(v_\sigma, b) - (-v_m, -b_m)\|\} \leq \epsilon \\ &\Rightarrow \min\{\|v_\sigma - v_m\|, \|v_\sigma - (-v_m)\|\} \leq \epsilon. \end{aligned}$$

Since $\epsilon > 0$ was arbitrary, we have $\forall\, \epsilon > 0, \exists \sigma' > 0$ s.t. $\sigma \leq \sigma' \Rightarrow d(v_\sigma, \{v_m, -v_m\}) \leq \epsilon$. This proves the result. ■

# Chapter 5

# Divisive Clustering of High Dimensional Data Streams


## Abstract

*Clustering streaming data is gaining importance as automatic data acquisition technologies are deployed in diverse applications. We propose a fully incremental projected divisive clustering method for high-dimensional data streams that is motivated by high density clustering. The method is capable of identifying clusters in arbitrary subspaces, estimating the number of clusters, and detecting changes in the data distribution which necessitate a revision of the model.*

*The empirical evaluation of the proposed method on numerous real and simulated datasets shows that it is scalable in dimension and number of clusters, is robust to noisy and irrelevant features, and is capable of handling a variety of types of non-stationarity.*


## 1 Introduction

High dimensional data stream clustering is increasingly relevant as automatic data generation and acquisition technologies are adopted in diverse applications. Data streams are encountered in a variety of settings. These include: computer network traffic monitoring, Web page requests, customer click streams, sensor networks, as well as transactions data from stock and foreign exchange markets, to name a few. The volume of data involved in these applications is far too large to fit in main memory. Hence random access to past observations is costly. Linear scans are the only acceptable access method in terms of computational efficiency (Guha et al., 2003; Silva et al., 2013). A defining property of data streams is that the population distribution

is subject to changes over time. This phenomenon is known as *population drift* (Babcock et al., 2002). These characteristics pose significant challenges for clustering. Streaming clustering algorithms must be incremental, with time and storage requirements independent of the size of the stream. In addition they must be capable of adapting to population drift by revising the cluster structure, distinguishing emerging clusters from noise, and discarding expired clusters (Jain, 2010).

The majority of existing streaming clustering algorithms consist of two components. The first is an online component that incrementally updates data structures that summarise the data sample. The second component operates offline, and performs the actual clustering, operating on the data summaries, rather than the original data samples (Aggarwal et al., 2003; Cao et al., 2006). However, clustering algorithms for static datasets invariably involve parameters which are application dependent, e.g. the number of clusters in $k$-means. Using such methods in the offline component of a streaming algorithm implicitly assumes that appropriate values for these parameters remain constant over the duration of the data stream. This is hard to justify in the presence of population drift. Existing algorithms attempt to handle population drift through simple heuristics, like sliding windows (Aggarwal et al., 2003; Kranen et al., 2009) and forgetting factors (Aggarwal et al., 2004; Cao et al., 2006), which are user-determined and static over the length of the stream. The important aspect of change detection is largely ignored (Silva et al., 2013). Finally, the majority of traditional clustering algorithms rely on the Euclidean distance between data samples, which becomes less meaningful as dimensionality increases (Kriegel et al., 2009). Few streaming algorithms are capable of handling very high dimensional data, and in general these are limited to detecting clusters only in axis parallel subspaces (Aggarwal et al., 2004; Ntoutsi et al., 2012; Hassani et al., 2102, 2014).

The framework we propose draws on the standard nonparametric statistical definition of clusters as regions of high density in the underlying probability distribution (Hartigan, 1975; Cuevas and Fraiman, 1997; Rigollet and Vert, 2009). In this approach, a *high-density cluster* is defined as a connected component of the level set of the (unknown) density function. When the density is unimodal the level set is connected, otherwise it can be connected or not. If it is disconnected, high-density clusters correspond to regions around modes of the density (Menardi and Azzalini, 2014). Identifying connected regions of high density of an unknown density is a challenging task even in moderate dimensions. Existing methods rely on an approximation of the density (Azzalini and Torelli, 2007; Cuevas et al., 2001), or attempt to infer local properties of the density (Menardi and Azzalini, 2014; Stuetzle and Nugent, 2010). Due to the curse of dimensionality such approaches are only effective on problems in up to tens of dimensions (Menardi and Azzalini, 2014). In higher dimensions, graph theoretic formulations have been used

to approximate these high density regions (Cuevas et al., 2001; Rinaldo and Wasserman, 2010).

The influential DBSCAN algorithm (Ester et al., 1996), combines a kernel density estimate using uniform kernel with a graph theoretic approach to determine clusters. Data points whose density exceeds a chosen threshold, $\lambda > 0$, are considered high-density points. A graph is constructed by connecting each high density point with each other point within a radius equal to the bandwidth of the kernel density estimator. Connected components of the graph define clusters, while singletons are interpreted as noise. This approach is efficient from a computational perspective, but the connected components of the resulting graph are not guaranteed to coincide with the components of the corresponding level set of the estimated density.

A basic weakness of algorithms that attempt to identify high-density clusters for a single, user-defined density level, is that both the number and the shape of the clusters depends on the choice of this parameter. In addition using a single density threshold can fail to detect clusters of varied densities (Ankerst et al., 1999; Stuetzle, 2003). To overcome this limitation one can compute the clustering structure that arises by considering all possible values of the density level. The collection of clusters which arises is known as the *cluster tree* (Hartigan, 1975). Recent algorithms that attempt to estimate the cluster tree include OPTICS (Ankerst et al., 1999), and Gslclust (Stuetzle and Nugent, 2010). A general approach to detect clusters at a local level from a cluster tree is discussed in (Campello et al., 2013).

In this article we propose a framework for streaming data clustering that relies on a different approach to high-density clustering. Instead of attempting to estimate high-density clusters directly, it partitions the data sample hierarchically using linear separators (hyperplanes) that pass through regions of low density. It thereby avoids splitting high-density clusters. This was first proposed for static data clustering by Tasoulis et al. (2010). An attractive feature of this approach from a computational perspective is that it requires only one-dimensional projections to identify low density separators. Expanding on this we propose a framework for streaming data clustering, which we refer to as High-dimensional Streaming Divisive Clustering (HSDC), that is able to: (i) identify clusters of arbitrary orientation; (ii) estimate the number of clusters automatically using a statistically motivated divisive procedure; (iii) update the clustering result incrementally without any offline component; (iv) utilise information about structural variation of the model to influence forgetting, instead of relying on static parameters; and (v) identify changes in the population distribution that require the revision of the clustering model. This is achieved through synthesising and extending results from incremental dimensionality reduction, kernel density estimation, and change detection.

The remaining paper is organised as follows. In Section 2 we discuss some

existing data stream clustering algorithms. Section 3 gives a more formal description of the problem we consider and discusses challenges associated with a data stream implementation. Section 4 describe our methodology for introducing components to the model, as well as how to accommodate population drift. In Section 5 we give a brief summary of the algorithmic structure of the method, as well as investigate the computational complexity of the model updates. In Section 6 we document the results of an extensive simulation study and performance on publicly available data sets. Finally in Section 7 we give some concluding remarks.

## 2 Related Work

Many existing data stream clustering algorithms extend classical clustering algorithms, such as $k$-means, $k$-medians, fuzzy $c$-means, and DBSCAN, to the data stream framework (Guha et al., 2003; Zhang and Ramakrishnan, 1996; Aggarwal et al., 2003, 2004; Cao et al., 2006; Kranen et al., 2009).

One of the most influential data stream clustering algorithms is CluStream (Aggarwal et al., 2003). CluStream uses *microclusters* to summarise the data received by the algorithm incrementally. These microclusters are then clustered offline using a weighted $k$-means algorithm. Microclusters store first and second order summary statistics of spatial and temporal information, and possess useful additive, subtractive and multiplicative properties, making them well suited to windows and forgetting factors. To handle non-stationarity, CluStream stores snapshots of the microclusters which enable it to approximate the clustering result over a window, which is specified by the user.

HPStream (Aggarwal et al., 2004) is a modification of CluStream to handle high dimensional data. Distance calculations are performed within axis parallel subspaces so as to minimise the radii of the microclusters. Assigning potentially differing subspaces to the clusters negates the additive and subtractive properties of the microclusters, and so HPStream treats the microclusters as actual clusters rather than data summaries. Snapshots also become meaningless, and so temporal variation is handled by fixed forgetting factors.

DenStream (Cao et al., 2006) is a density-based algorithm that uses microclusters. To handle noise it distinguishes between *outlier* and *potential* microclusters, the latter defined by a threshold on the weighted number of points falling within a sphere of fixed radius. Weights are exponentially decreasing functions of time, enabling the algorithm to adapt to population drift. The offline component of DenStream is a variant of DBSCAN, thus, enabling the estimation of the number of clusters, and the detection of clusters of arbitrary shape. Recently proposed extensions of DenStream include

HDDStream (Ntoutsi et al., 2012), PreDeConStream (Hassani et al., 2102), and DMMStream (Amini et al., 2014a). HDDStream and PreDeConStream handle high dimensional data by scaling up the contribution of *preferred* dimensions within distance calculations. DMMStream enables the detection of density connected clusters on differing scales, through the use of mini microclusters.

Grid based density clustering algorithms have also been proposed. The DStream (Chen et al., 2007) algorithm is similar to DenStream, except that dense grid cells (cells containing relatively high approximate integrated density) are used instead of microclusters. The number of grid cells however depends exponentially on the dimension of the data. DDStream (Jia et al., 2008) is an extension of DStream which allows the absorption of data at the boundaries of clusters into adjacent dense cells, thereby reducing the number of *active* grid cells.

A potential feature of data streams is that the rate at which new data is observed can vary over time. *Anytime* algorithms are able to produce a clustering result after any amount of processing time, but are also capable of refining this result when more time is available (Kranen, 2011). Anytime stream clustering was first discussed in relation to the ClusTree algorithm (Kranen et al., 2009). ClusTree stores a hierarchy of microclusters, with each internal node representing an aggregation of its children. Arriving data are inserted at the root and traverse the hierarchy via the nearest microcluster at each level. This insertion is halted if there is insufficient time. Halted data can "hitchhike" further down the hierarchy with similar arriving data at a later stage. In this way CluStree is capable of handling not only extremely high velocity data streams, but also streams in which the velocity varies. SubClusTree (Hassani et al., 2014) extends ClusTree to high dimensional applications by establishing multiple hierarchies, each existing within a different subspace.

A comprehensive review and categorisation of existing data stream clustering algorithms is provided in two recent surveys (Aggarwal, 2013; Silva et al., 2013). A review focused on density based methods for streaming data is provided in (Amini et al., 2014b).

## 3 Problem Description

Our aim is to generate a hierarchical partition of the Euclidean space $\mathbb{R}^d$ such that the modes of a probability density, $f$, over $\mathbb{R}^d$, are uniquely contained within different cells of the partition. The density, $f$, is not known, and instead we receive a sequence of realisations of the random variable $X$ with density $f$. The learning process is constrained by standard memory and computation limits associated with data stream learning. We do not assume

that $f$ is constant in time, and so modify the model as changes in the empirical distribution of realisations suggests is necessary.

This problem can be formulated in the context of high density clustering, wherein the modes of the density $f$ can be associated with its *level sets*.

**Level Set** For level $\lambda \geq 0$ and density function $f$, the level set of $f$ above $\lambda$, or $\lambda$ *level set of* $f$, is defined as $\overline{\{x \in \text{Support}(f) | f(x) \geq \lambda\}}$.

As $\lambda$ increases, the $\lambda$ level set centers around the modes of $f$ above $\lambda$, and therefore the number of modes, or clusters, can be associated with the number of maximal connected subsets of the level sets. We refer to these maximal connected subsets as the *components* of the level set. Identifying the components of level sets of a high dimensional probability density function is extremely costly in terms of computational effort, and often these are approximated using graph theoretic formulations (Cuevas et al., 2001; Rinaldo and Wasserman, 2010). In addition, the density function $f$ is unknown and must be approximated. Standard methods, such as Kernel Density Estimation (KDE), become less effective at accurately representing the underlying probability density as dimensionality increases (Scott, 2009). Building an approximation of $f$ incrementally in a data stream setting introduces yet further challenges, since only summaries of the data can be stored and hence further approximations are necessary.

The dePDDP algorithm (Tasoulis et al., 2010) attempts to separate the modes of a distribution via a hierarchy of low density separating hyperplanes. The algorithm recursively projects (subsets of) the data into a one dimensional subspace and splits the projected data above and below the lowest antimode of their estimated density. The KDE of the projected data provides an upper bound on the value of the full dimensional KDE, as shown in the following lemma, which is adapted from Tasoulis et al. (2010).

**Lemma 27** *Let $\mathcal{X} = \{x_1, ..., x_N\}$ be a d-dimensional data set, and let $v \in \mathbb{R}^d$ have unit length and let $b \in \mathbb{R}$. Let $\hat{f}$ denote the d-dimensional kernel density estimate of the distribution of $\mathcal{X}$ with bandwidth matrix $hI$ using the multivariate Gaussian kernel. Let $\hat{f}_1$ be the univariate kernel density estimate of the distribution of $v \cdot \mathcal{X}$ with bandwidth $h$ using the univariate Gaussian kernel. Then for any $x \in \mathbb{R}^d$ s.t. $v \cdot x = b$,*

$$\hat{f}(x) \leq h^{d-1} \hat{f}_1(b).$$

Splitting at the lowest antimode of the projected density estimate, therefore, avoids intersecting high level sets of the full dimensional estimated density. Separating clusters by regions of low density has also been shown to yield more stable clusters (von Luxborg, 2010), which fits well with the possibility of smooth time variations in $f$. While this approach avoids the explicit

estimation of the full dimensional density, the model structure is limited to cases where the modes of the density can be separated by a hierarchy of hyperplane separators. In particular it is limited to cases in which the convex hulls of the modes are non-overlapping. Despite this limitation, the approach has shown good empirical performance in a number of high dimensional applications (Boley, 1998; Tasoulis et al., 2010). The dePDDP algorithm projects data onto their first principal component, using the justification that directions with high variability are likely to display high *between cluster* variability. This will tend to lead to good separation of the clusters, and hence low antimodes in the projected density estimate.

A hyperplane can be parameterised by a vector $v \in \mathbb{R}^d$ and scalar $b \in \mathbb{R}$ as the set $\{x | v \cdot x = b\}$. No generality is lost by assuming that $v$ is unit length. We adopt the approach of learning $v$, which we will refer to as the *projection vector*, from the full dimensional realisations of $X$, and then using the projections of these realisations onto $v$ to inform our choice of $b$, which we refer to as the *split point*. The problem of incremental principal component analysis is well studied (Artac et al., 2002; Li et al., 2003; Weng et al., 2003), and computationally efficient algorithms exist. In order to determine the split point, projections of the data onto the projection vector are used to approximate the density of the random variable $v \cdot X$. Low empirical density regions in this density approximation suggest the location of low density hyperplanes. Combining these provides a readily available framework for an online version of dePDDP, such as that adopted by SPDC (Tasoulis et al., 2012). Such a straightforward implementation, however, has important limitations.

Incremental updates to the projection vector, which we henceforth index by $t$ to indicate the $t$-th step estimate, mean that the sample of projections at time $t$ is given by $\{v_1 \cdot x_1, v_2 \cdot x_2, ..., v_t \cdot x_t\}$, which is not a sample from the random variable $v_t \cdot X$. With successive updates to $v_t$ the accuracy of the projected points at estimating the empirical distribution of $v_t \cdot X$ diminishes, which affects the accuracy of the spit point. Futhermore, if the splitting rule at a node in the hierarchy is updated with each observation, then the sets of data being passed to its children will vary. This variability propagates down the hierarchy and renders projection vector updates and splitting decisions at lower levels increasingly inaccurate and unstable. Moreover, if the underlying distribution changes in time, these projections and splitting rules are rendered even more inaccurate, unless these changes are suitably accommodated.

In what follows we detail our approach to overcoming these limitations. We describe the incremental updates to a node of the hierarchy. The time indices, $t$, relate to the $t$-th update to the node, and not the $t$-th update to the entire hierarchy. Similarly, the $t$-th datum refers to the $t$-th datum received by the node, and not the $t$-th datum in the entire stream.

# 4 Methodology

In this section we discuss in detail the three components of the proposed High-dimensional Streaming Divisive Clustering (HDSC) framework. HSDC constructs incrementally a hierarchical clustering model which consists of a collection of separating hyperplanes. The hyperplanes pass through regions of low density, and are orthogonal to directions of high variance. These hyperplane separators are maintained within the model until there is sufficient evidence indicating that they no longer pass through regions of low density. Whenever this occurs the hyperplane in question, and therefore the part of the hierarchy rooted at it, is removed from the model and the corresponding node is re-initialised.

Section 4.1 details how we find high variance projection directions incrementally using the CCIPCA algorithm (Weng et al., 2003). Section 4.2 describes how low density hyperplanes are identified using information from the projected data only. In Section 4.3 we discuss how population drift can be handled within this framework.

## 4.1 Learning High Variance Projections

The $k$-th principal component of a data set, $\mathcal{X}$, is given by the $k$-th largest eigenvector of its covariance matrix. Many incremental methods for principal component estimation require that the full covariance matrix be approximated, leading to high computation and storage costs for large problems. The CCIPCA algorithm (Weng et al., 2003) instead focuses directly on the eigen-problem $\mathrm{Cov}(\mathcal{X})u = \lambda u$. The algorithm is based on the recursion,

$$v_t = \frac{t-1}{t} v_{t-1} + \frac{1}{t} \frac{x_t \cdot v_{t-1}}{\|v_{t-1}\|} x_t, \tag{5.1}$$

where $\{x_t\}_{t=1}^{\infty}$ is a sequence with zero mean. Weng et al. (2003) have shown that the recursion of Eq. (5.1) converges almost surely to $\pm\lambda u$ for the maximum value of $\lambda$, i.e., $u$ is the first principal component. Lower order eigenvectors are found by first projecting data into the null space of all approximate higher order eigenvectors.

Early updates to the projection vector are highly variable, making it more challenging to approximate the marginal distribution along it. Passing promising projection vectors down the hierarchy to act as initial projection vectors in the child nodes can help to ameliorate this problem. When a node is split, a hyperplane orthogonal to its projection vector is introduced to the model. The truncation induced by a separating hyperplane tends to reduce the variability in the normal direction (i.e., along the projection direction) more than in directions orthogonal to it. The second most highly variable direction is therefore a good candidate for a high variance projection in the child nodes.

We investigate a variation on the basic HSDC model, which we call *projection inheritance*, in which each node (besides the root node) learns both its own projection and a high variance projection to be passed to its children. So that this inheritance is not lost to natural variations in the data, it is given additional weight in subsequent updates equal to the number of updates already undergone. Orthogonality to the parent's projection vector is also enforced in subsequent updates after being passed down. This increased stability in the projection vector can be crucial to estimating the distribution along it. If the updates are highly variable, then the projections made onto it will be less reliable at representing the target distribution.

Below we describe formally the updates to the projection and inheritance vector with the arrival of the $t$-th datum $x_t$. Let $u_t$ be the (unnormalised) projection vector at time $t$, and $z_t$ the inheritance vector. Also, if $u_t$ was initialised by inheritance, let $N$ be the number of updates undergone prior to inheritance and let $p$ be the projection vector of the parent node (otherwise assume $p = \mathbf{0}$ and adopt the convention that $0/0 = 0$).

$$
\begin{aligned}
\bar{x}_t &= \frac{t-1}{t}\bar{x}_{t-1} + \frac{1}{t}x_t \\
x_C &:= x_t - \bar{x}_t \\
x_0 &:= x_C - \frac{x_C \cdot p}{\|p\|^2}p \\
u_t &= \frac{t+N-1}{t+N}u_{t-1} + \frac{1}{t+N}\frac{x_0 \cdot u_{t-1}}{\|u_{t-1}\|}x_0 \\
x_C &= x_C - \frac{x_C \cdot u_t}{\|u_t\|^2}u_t \\
z_t &= \begin{cases} x_C, & t = 1 \\ \frac{t-1}{t}z_{t-1} + \frac{1}{t}\frac{x_C \cdot z_{t-1}}{\|z_{t-1}\|}x_C, & \text{otherwise.} \end{cases}
\end{aligned}
$$

In subsequent sections we will assume that the projection vector is normalised, and denote it by $v_t$.

## 4.2 Splitting Based on a Projected Sample

High density clustering associates clusters with modes of the underlying probability density. Introducing a new split to the hierarchical model is therefore only done when the corresponding truncation of the density, induced by the hierarchical partition of $\mathbb{R}^d$, contains multiple modes. Assessing the modality of a high dimensional probability density is difficult, however by considering one dimensional projections of the underlying random variable, $X$, we only need to assess the modality of a univariate sample. Cases exist in which the full dimensional density is unimodal, but in which the marginal distribution of a univariate projection is multimodal, and vice versa, and in

these cases the accuracy of our model will be compromised as components may be split between different elements of the partition.

### Assessing Modality

In this subsection we assume that we observe a univariate sample corresponding to the projections of the underlying random variable. The *excess mass* test (Müller and Sawitzki, 1991) is used to asses the modality of a sample from an unknown distribution function $F$ over $\mathbb{R}$, with density function $f$. The excess mass of $f$ at level $\lambda$ is defined as,

$$E(\lambda) = \int_{\mathbb{R}} (f(x) - \lambda)^+ \mathrm{d}x.$$

The excess mass therefore measures the integrated density above level $\lambda$. The excess mass can also be formulated in terms of the distribution function $F$,

$$E(\lambda) = \sup_{I_1,\ldots,I_{c(\lambda)}} \sum_{i=1}^{c(\lambda)} \left(F(I_i) - \lambda \|I_i\|\right), \tag{5.2}$$

where $c(\lambda)$ is the number of connected components of the $\lambda$ level set of $f$ and $\|I\|$ is the diameter of the set $I$. The supremum is taken over all collections of size $c(\lambda)$ of disjoint intervals. The latter formulation allows for the empirical excess mass based on a sample, $\mathcal{X}$, from $F$ to be calculated by replacing $F$ in (5.2) with the empirical distribution $F_{\mathcal{X}}$, defined as

$$F_{\mathcal{X}}(z) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbb{I}[z \geq x].$$

In practice, the number of connected components $c(\lambda)$ will not be known, and so the empirical excess mass is compared for different values. We use the notation $\hat{E}_c(\lambda)$ to mean the empirical excess mass for $c$ intervals. The excess mass statistic for comparing $c_1$ with $c_2 > c_1$ components is defined as

$$\Delta(c_1, c_2) = \sup_{\lambda} \{\hat{E}_{c_2}(\lambda) - \hat{E}_{c_1}(\lambda)\}.$$

The larger $\Delta(c_1, c_2)$, the more evidence in favour of $c_2$ over $c_1$. In our context, we are interested in whether or not a density has more than one mode, and therefore are interested in the case $c_1 = 1, c_2 = 2$. This case can equivalently be assessed via the *dip* (Hartigan and Hartigan, 1985). The dip of a distribution function $F$ over $\mathbb{R}$ measures the departure from unimodality of $F$ and is given by the supremal distance between $F$ and the distribution function with unimodal density for which this supremal distance is minimal. Formally,

$$\mathrm{Dip}(F) = \min_{U \in \mathcal{U}} \|F - U\|_\infty,$$

where $\mathcal{U}$ is the class of distribution functions with unimodal density. It has been shown that the dip is equal to half the *excess mass* statistic $\Delta(1,2)$, and can therefore be used equivalently to assess multimodality when the dip of $F_{\mathcal{X}}$ is considered. To assess the significance of the dip or excess mass, a null unimodal distribution is specified and the quantiles under this null distribution estimated using Monte Carlo simulation. The benefits of using the dip relate to the computationally efficient algorithm given by Hartigan (1985), which is linear in the sample size. The corresponding unimodal distribution can be extracted from the algorithm, and hence the value of $\lambda$ corresponding to the excess mass can be obtained. We make use of this value of $\lambda$ in the approximation of the underlying density (Section 4.2).

While the calculation of the dip is linear in the size of the sample, it cannot be calculated incrementally. In the context of a data stream, this violates the fixed storage and computation limits. To remedy this, we propose an approximation method which requires bounded memory and computation time. This is achieved by approximating the sample using a fixed number of compact intervals which are dynamically adjusted to always contain the entire sample. Storing only the endpoints of the intervals and the number of data falling in them allows us to construct an approximation of the sample which leads to a lower bound on the dip of the empirical distribution of the sample. Thus, this approximation method leads to a uniformly more conservative test of unimodality, which fits well with our objective to avoid prematurely splitting clusters.

### Compactly Approximating the Sample

Our approximation method relies on the notion of a uniform set, which is defined as follows.

**Uniform Set** Let $\mathcal{X}$ be a finite sample in $\mathbb{R}$ and $I=[a,b], a\leq b\in\mathbb{R}$. Then the *uniform set* of $\mathcal{X}$ and $I$ is defined as

$$Unif(\mathcal{X},I)=\bigcup_{i=1}^{n}\left\{m+\frac{i-1}{n-1}(M-m)\right\},$$

where $n=|\mathcal{X}\cap I|$, $m=\min\{\mathcal{X}\cap I\}$, and $M=\max\{\mathcal{X}\cap I\}$.

We lose no generality by assuming that the endpoints of the interval $I$, $a$ and $b$, are elements of $\mathcal{X}$. For the purpose of approximating the empirical distribution, the uniform set replaces the empirical distribution on $I$ with the distribution function of the random variable $Y_I:=\frac{\|I\|}{|\mathcal{X}\cap I|-1}U+\min\{I\}$, where $U\sim\mathcal{U}[0,|\mathcal{X}\cap I|-1]$ is the discrete uniform random variable on the support $\{0,1,...,|\mathcal{X}\cap I|-1\}$. Notice that we have again adopted the convention $0/0=$

0. For a collection of disjoint intervals $I_1 < I_2 < ... < I_k$, which jointly contain the entire sample, the approximate distribution is given by,

$$\tilde{F}(x) = \frac{1}{|\mathcal{X}|} \sum_{i=1}^{k} |\mathcal{X} \cap I_i| F_{Y_{I_i}},$$

where $F_{Y_{I_i}}$ is the distribution function of $Y_{I_i}$, defined as above.

With the arrival of a new datum, $x$, $\tilde{F}$ must be updated such that the number of intervals used does not exceed the predefined limit. If $x$ lies within one of the existing intervals, then no adjustment to the above formulation is necessary. Otherwise, an interval $I = [x, x]$ is added, and then two adjacent intervals are replaced with the convex hull of their union. The intervals *merged* in this way are the adjacent pair which minimise the supremal distance between $\tilde{F}$ before and after the merger. If intervals $i, i+1$ are merged, then this distance is given by,

$$\frac{1}{|\mathcal{X}|} \max \left\{ \left| |\mathcal{X} \cap I_i| - \left\lceil \frac{\|I_i\|}{\|I_{i:i+1}\|} \right\rceil |\mathcal{X} \cap I_{i:i+1}| \right|, \right.$$
$$\left. \left| |\mathcal{X} \cap I_i| + 1 - \left\lceil \frac{\min I_{i+1} - \min I_i}{\|I_{i:i+1}\|} \right\rceil |\mathcal{X} \cap I_{i:i+1}| \right| \right\},$$

where $I_{i:i+1} = I_i \cup I_{i+1}$. With the above formulation of $\tilde{F}$ we arrive at the following result, the derivation of which can be found in the appendix. We also discuss therein how a slight modification to the dip algorithm allows one to calculate the dip of the sample approximation in $\mathcal{O}(k)$ time, where $k$ is the number of intervals.

**Lemma 28** *Let $\mathcal{X}$ be a univariate sample of distinct points. For any collection of disjoint, compact intervals $I_1 < I_2 < ... < I_k$ satisfying*

$$\mathcal{X} \subset \bigcup_{j=1}^{k} I_j,$$

*we have $Dip(\tilde{F}) \leq Dip(F_{\mathcal{X}})$.*

This result ensures that the approximation method used cannot lead to additional false discovery of multimodality when compared with the true sample of observations. While the result is stated for an unweighted sample, it also holds for weighted samples for which the data within each interval are given the same weight. This is important as in the next subsection we describe how reweighting the data can be useful in better approximating the distribution of a sample projected onto a vector which is continually being updated.

### Accommodating a Shifting Projection

Our aim is to approximate the distribution of the projected random variable, $v \cdot X$, where $\|v\| = 1$. However, we only observe realisations of a sequence of random variables $v_t \cdot X_t$, where under the assumption that $X_1, X_2, \ldots$ are i.i.d., we know that $v_t$ converges almost surely to a vector $v$. Even under this assumption, the realisations $v_t \cdot x_t$ still represent a sample from a nonstationary distribution due to the shifting projection $v_t$. With consecutive updates to $v_t$, the accuracy of the observed projections as a representation of the target distribution diminishes. The influence of these observations on the approximate distribution should therefore diminish with subsequent updates. *Forgetting factors* impose a decaying weight mechanism to control the influence of past observations on the current estimate, however they are difficult to tune in practice. If $w_{t,i}$ is the weight associated with observation $i$ at time $t$, then $w_{t,i} = (1-\lambda_t) w_{t-1,i}$, where $\lambda_t \in [0,1]$ is the forgetting factor at time $t$. Weights associated with new observations are initialised at 1, and so we have,

$$\begin{aligned} W_t := \sum_{i=1}^{t} w_{t,i} &= 1 + (1-\lambda_t) \sum_{i=1}^{t} w_{t-1,i} \\ &= 1 + (1-\lambda_t) W_{t-1}. \end{aligned}$$

Our approximate distribution $\tilde{F}$ is a mixture of discrete uniform distributions, in which the weight associated with each component is equal to the number of atoms in its support. Using forgetting factors we can adjust these weights to obtain a more accurate approximation to the distribution on the current projection. The update to $\tilde{F}$, which we now index by $t$ to represent the approximation after $t$ observations, is therefore given by,

$$\tilde{F}_t = \frac{1}{W_t} F_{Y_{[x_t, x_t]}} + (1-\lambda_t) \frac{W_{t-1}}{W_t} \tilde{F}_{t-1},$$

where $x_t$ is the observation at time $t$. If the number of intervals then exceeds the upper limit, the merging of two adjacent intervals is performed as described in Section 4.2. As $\lambda_t$ approaches zero, past and present observations become equally weighted, while higher values of $\lambda_t$ increase the influence of recent observations on $\tilde{F}$. In problems with an explicit loss function, forgetting factors can be tuned using stochastic gradient descent (Haykin, 1999; Anagnostopoulos et al., 2012; Pavlidis et al., 2011), but this is not true in our context. We propose an adaptive scheme which is complementary to the incremental estimation of the projection in that it uses information about the angles between consecutive updates to $v_t$ to quantify the variability of the projected distribution over time. In detail, with the arrival of the $(t+1)$-th datum, first $v_t$ is updated as described in Section 4.1, and then $\lambda_t$ is set to,

$$\lambda_{t+1} = \min\{\Lambda, \gamma \lambda_t + (1-\gamma) \arccos(v_{t+1} \cdot v_t)\}, \tag{5.3}$$

where $\Lambda \in (0,1]$ is a chosen maximum forgetting factor. We use an exponentially weighted moving average (EWMA) with parameter $\gamma \in (0,1)$ to smooth the impact on $\lambda_t$ of isolated large fluctuations in $\arccos(v_{t+1} \cdot v_t)$ arising from natural variation in $v_t$. The following proposition states that the adaptive scheme of (5.3) converges almost surely to 0, whenever $v_t$ converges almost surely. The distribution approximation will therefore stabilise as the projection converges, which is almost sure under the CCIPCA algorithm as long as the underlying distribution does not change.

**Proposition 29** *If $\{v_t\}_{t=1}^{\infty}$ converges almost surely and $\|v_t\| = 1 \; \forall t$, then $\lambda_t \xrightarrow{a.s.} 0$, where $\lambda_t$ is as in (5.3).*

Reweighting the projected data in this way means that the approximate dip or excess mass must be compared with the quantiles for a sample of size equal to the *effective sample size* of the reweighted data, which we calculate as the sum of the weights in the approximation $\tilde{F}$.

When the null hypothesis of unimodality is rejected based on the dip of the approximate sample distribution, the associated node is split. The projection direction and split point are then kept fixed. In the following we describe how to approximate local minima in the density along the projection, thereby allowing one to determine such a split point based on this density.

### Approximating Anti-modes

The distribution function $\tilde{F}$ is discontinuous, and therefore using this distribution directly to approximate an antimode in the underlying density is challenging. A standard approach to generating smooth density approximations is to consider a convolution with a smooth distribution function, having density $K$. The density $K$ is referred to as a *kernel*. The convolution of a discrete distribution associated with a random variable $Y$, having mass function $p(y)$, with a smooth distribution, gives rise to the canonical kernel density estimate,

$$\hat{f}(x) = \frac{1}{h} \sum_{y \in \text{Support}(Y)} p(y) K\left(\frac{x-y}{h}\right).$$

The parameter $h$ is called the *bandwidth*, and controls the smoothness of the resulting density estimate. A common choice of kernel is the standard Gaussian distribution given by,

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

In this case the bandwidth directly relates to the standard deviation of the smoothing density. The support of the random variable underlying our approximation $\tilde{F}$ still contains $|\mathcal{X}|$ atoms, despite the compression of its representation. The evaluation of the associated kernel density estimate at a

fixed number of points is $\mathcal{O}(|\mathcal{X}|)$. Knowledge that $\tilde{F}$ represents a mixture of discrete uniform distribution functions leads us to instead consider the convolution of the corresponding continuous uniform distribution functions with the kernel $K$. For those intervals which contain only a single point, there is no associated continuous distribution, and so the standard kernel convolution above is used. The convolution of the uniform density on $[a,b], a<b\in\mathbb{R}$ with the Gaussian distribution with variance $h^2$ is given by

$$f(x)=\frac{\Phi((x-a)/h)-\Phi((x-b)/h)}{b-a},$$

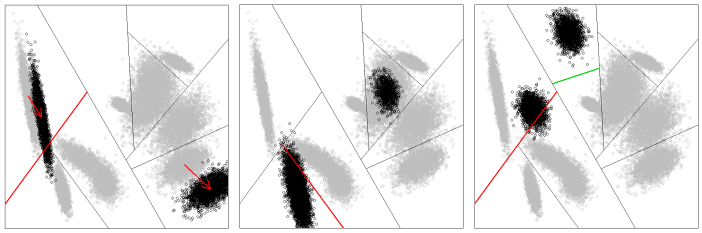
where $\Phi$ is the distribution function of the standard Gaussian random variable. With this formulation the associated kernel density estimate has $k$ components, rather than $|\mathcal{X}|$.

How to choose $h$ remains a very active area of research, and no universal guidelines exist for every context. We use the equivalence of the dip and excess mass tests, and the implications of the rejection of their null hypotheses, to inform our choice of $h$. Rejection of the null hypothesis of the excess mass test is equivalent to favouring two $\lambda$ level set components over one. The corresponding value of $\lambda$ can be extracted from the dip algorithm. We choose $h$ such that the associated density estimate has $\lambda$ level set consisting of two components. The minimum such value of $h$ is chosen, as this leads to more sharply defined modes and anti-modes.

## 4.3 Handling Population Drift

A generic approach to detect time-variations in $f$ that invalidate our clustering model is to sequentially consider the hypothesis that each hyperplane passes through a region of low relative $f$ density (*low-density separation hypothesis*). If this hypothesis is false then a revision of the corresponding part of the hierarchical model is necessary. Notice that the low-density separation hypothesis is not a hypothesis of overall stationarity of $f$. Indeed, as Figure 5.1 shows, considerable variation in $f$ is possible without invalidating the model. The figure also shows that testing this hypothesis for all separating hyperplanes enables us to identify and revise only the relevant part of the clustering hierarchy when a change is detected, rather than resetting the entire model. Moreover, the low-density separation hypothesis is independent of the type (abrupt, or gradual) and the speed of drift. Lastly, testing this hypothesis corresponds to a one-dimensional change detection problem, since each low-density hyperplane is identified through an estimate of a one-dimensional marginal density.

With each hyperplane added to the model we initialise a Bernoulli CUSUM change detection regime, as described in Reynolds and Stoumbos (1999), to



(a) Cluster Location (b) Cluster Shape (c) New Clusters

**Fig. 5.1:** Different changes in distribution and their impact on the clustering model. Red lines indicate necessity of model revision. Green lines indicate changes which can be addressed by extending the model without revision

detect significant increase in the frequency of data arising in a small neighbourhood of the hyperplane. This frequency is determined relative to the frequency in a larger neighbourhood extending to the adjacent modes of the projected density. In this way the local density behaviour is better represented. If such a significant increase is observed, the corresponding node and the subhierarchy it anchors are removed from the model, and the node is re-initialised. We determine the larger region, which we denote by $\mathcal{R}$, by the location of the adjacent modes in the density estimate described in Section 4.2. The smaller neighbourhood, $\mathcal{N}$, of the hyperplane is taken to be some proportion, $\beta \in (0,1)$ of the region $\mathcal{R}$.

To implement a Bernoulli CUSUM, a pre- and post-change frequency must be specified. We obtain an initial estimate of the pre-change frequency, $p_0$, using the estimated density described in Section 4.2, and then update this estimate with new observations to obtain a more accurate estimate. With this updating regime, the threshold parameter must be recalculated with each such update. We set the post-change frequency, $p_1$, equal to the average density over $\mathcal{R}$, since this is the supremal possible frequency while the hyperplane is at an anti-mode of the projected density with adjacent modes beyond the boundaries of $\mathcal{R}$.

The Bernoulli CUSUM statistic, $S_0$, is initialised at 0. Note that the time index here, unlike previously, relates to the $t$-th datum since the hyperplane was introduced to the model. With the arrival of datum $x_{t+1}$ the CUSUM statistic is updated as follows. If $x_{t+1} \notin \mathcal{R}$, then the datum is not relevant,

and $S_{t+1} = S_t$. If $x_{t+1} \in \mathcal{R}$, then,

$$\begin{aligned} B &= \begin{cases} 1, & \text{if } x_{t+1} \in \mathcal{N} \\ 0, & \text{otherwise} \end{cases}, \\ S_{t+1} &= \max\{0, S_t\} + B \\ &\quad + \log\left(\frac{1-p_1}{1-p_0}\right) \log\left(\frac{p_1(1-p_0)}{p_0(1-p_1)}\right)^{-1}. \end{aligned}$$

A change is detected when $S_t$ exceeds a threshold $\alpha$. We set $\alpha$ conservatively, corresponding to the maximum for chosen run lengths under $p_0$ and $p_1$, which we calculate according to the method in Reynolds and Stoumbos (1999).

# 5 The HSDC Algorithm

Having detailed the constituent elements of the method in Section 4, in this section we give a brief summary of the overall algorithm. The clustering model constructed by HSDC comprises a hierarchy of separating hyperplanes, each defined by a projection vector, $v \in \mathbb{R}^d$, of norm 1, and split point $b \in \mathbb{R}$. Associated with each hyperplane is a CUSUM statistic, $S$, in place to detect changes in the underlying distribution which lead to instability of the clustering result. The leaf nodes of the hierarchy (i.e. those for which the truncated density has not been deemed multimodal) each have associated with them an updating projection, as well as an approximate univariate distribution associated with the projection onto it. Leaf nodes might also contain inheritance vectors, $z$, to be passed to their children as projections in the event that the node is split.

Algorithm 3 describes an update to the hierarchical model with the arrival of a datum $x$. We associate with each node an identifying tag, $ID$, which informs the algorithm where to direct data down the hierarchy. Each internal node has 2 children, *LChild* and *RChild*, associated with the halfspaces induced by the node's hyperplane.

The datum traverses the hierarchical structure until it reaches the appropriate leaf node. At each internal node along its path, the corresponding CUSUM statistic is updated as in Section 4.3. If a change is detected, the associated node is reinitialised, and it becomes a leaf. Otherwise, the datum is projected onto the node's projection vector, and is passed to the appropriate child node. Once the datum arrives at a leaf, the leaf's projection vector is updated as described in Section 4.1. The datum is projected onto this updated vector, and this projected datum is used to update the node's sample approximation, as described in Sections 4.2 and 4.2. The dip statistic of the sample

approximation is then calculated, and if the hypothesis of unimodality is rejected, the node is split. The split point is given by the lowest anti-mode between the components of the associated level set of the estimated density, as in Section 4.2.

**Algorithm 3:** HSDC Update

Input: New datum $x$;
[Set index to root node]
$ID = \text{root}$;
[Find relevant leaf node]
**while** $node_{ID}$ *is internal* **do**
  $S_{ID} = \text{updateCUSUM}(S_{ID}|x)$ (Section 4.3);
  **if** $S_{ID} > \alpha_{ID}$ **then**
    removeSubhierarchy$(ID)$, re-initialise node $ID$;
  **else**
    $p := v_{ID}^{\top}(x - \bar{x}_{ID})$;
    $ID = \textbf{ifelse}(p < b_{ID}, LChild_{ID}, RChild_{ID})$;
  **end**
**end**
[Update leaf node and split if necessary]
$(v_{ID}, z_{ID}) = \text{updateProjection}(v_{ID}, z_{ID}|x)$ (Section 4.1);
$p = v_{ID}^{\top}(x - \bar{x}_{ID})$;
$\mathcal{X}_{ID} = \text{updateSample}(\mathcal{X}_{ID}|p)$ (Sections 4.2-4.2);
**if** *Reject* $H_0 := \tilde{F}_{ID}$ *is unimodal* **then**
  $b_{ID} :=$ Minimum Antimode Between $\lambda$ Level Set of $\hat{f}_{ID}$ (Section 4.2);
  initialise nodes $LChild_{ID}$ and $RChild_{ID}$.
**end**

## 5.1 Computational Complexity

We consider the worst case cost of updating the HSDC hierarchy. Suppose the model contains $C$ clusters. The maximum depth of the hierarchy is therefore $C$ (in general the depth of the hierarchy is much lower, with minimum value $\log_2(C)$). For each internal node along a path to a leaf node, a datum is projected onto the corresponding projection vector, with a computational cost $\mathcal{O}(d)$. This projected datum is used within an update to the corresponding CUSUM statistic with cost $\mathcal{O}(1)$. The maximal cost of finding the appropriate leaf node is thus $\mathcal{O}(C(d+1))$. Updates to a leaf node include updating its projection (and inheritance) vector, $\mathcal{O}(d)$, updating the sample approximation, $\mathcal{O}(k)$, and calculating the dip statistic, $\mathcal{O}(k)$. For an update which does not result in a new split being introduced, the computational cost is therefore

$\mathcal{O}((C+1)(d+1)+k)$. In high dimensional applications, we have $Cd \gg k$, and so the behaviour is $\mathcal{O}((C+1)(d+1))$.

When a node is split, we determine the minimum bandwidth, $h$, giving the correct level set of the density estimate. This is done by a bisection method, which has $\log_2((h_{max}-h_{min})/\epsilon)$ iterations, where $h_{max}$ and $h_{min}$ are upper and lower bounds on $h$ respectively, and $\epsilon$ is the tolerance level. Within each iteration, the kernel density estimate is calculated, at a cost of $\mathcal{O}(k^2)$. The split point is calculated within this procedure.

For existing data stream clustering algorithms based on microclusters, the primary cost associated with the online step lies in determining the nearest microcluster. This has computational cost $\mathcal{O}(md)$, where $m$ is the number of microclusters. The time complexity of the offline step depends on the clustering algorithm being employed. Algorithms based on $k$-means are technically NP hard, however practical implementations run in $\mathcal{O}(mkd)$ time. Density connectivity methods based on DBSCAN have time complexity $\mathcal{O}(dm\log(m))$.

Density based methods based on grids have time complexity of the online step $\mathcal{O}(g)$, where $g$ is the number of active grid cells. Without pruning methods, this is exponential in $d$ (Aggarwal, 2013). For the offline component, the approach of DStream (Chen et al., 2007) only requires processing those grid cells which changed since the last offline step. If $t$ is the number of time steps since the last offline step, the computational cost is $\mathcal{O}(t)$. This means that the total cost of all offline steps up to time point $T$ has worst case cost $\mathcal{O}(T)$.

In general the number of microcluster and grids cells is substantially larger than the actual number of clusters, i.e., $C \ll m, g$. (The HPStream algorithm is an exception.) For standard updates therefore HSDC compares favourably with existing data stream clustering algorithms in terms of update time, especially on high dimensional examples. Updates to HSDC which result in the introduction of a new split have an additional cost of $\mathcal{O}(k^2)$. However, in practice these updates requiring a new split being introduced are infrequent, and the overall complexity is dominated by standard update steps. Most importantly however, HSDC has no offline clustering component.

# 6 Experimental Results

We compare the performance of the following methods.

1. CluStream (Aggarwal et al., 2003): We use the implementation in the `R` package `streamMOA`, and the parameters suggested in (Aggarwal et al., 2003).

2. HPStream (Aggarwal et al., 2004): HPStream, like CluStream, requires an offline initialisation step, for which we give it 2000 data and provide

it with the correct number of clusters for the initial stream segment. We set the average dimensionality of the clusters to 80% the total dimensionality of the data, and the forgetting factor was set to 0.002. These parameters improved performance over the suggestions made in (Aggarwal et al., 2004).

3. SPDC (Tasoulis et al., 2012): We set the number of kernels for the $M$-kernel density estimator to 50.

4. Our method, HSDC and HSDC(I) (with projection inheritance): The null distribution for the dip test was the Gaussian, and we use the 95th centile from the Monte Carlo simulations as a threshold. We use 100 intervals for the sample approximation. The smoothing parameter, $\gamma$, for the EWMA associated with the forgetting factor was set to 0.9. Expected run lengths for CUSUM statistics were set to $10^6$ and 250 for $p_0$ and $p_1$ respectively.

We also considered the density based algorithms DenStream (Cao et al., 2006) and DMMStream (Amini et al., 2014a), however neither produced meaningful clusterings in high dimensional applications and therefore results are omitted.

To avoid ambiguity we will refer to true clusters in the data as *classes* and the assignments made by an algorithm as clusters. An ideal clustering model should (i) correctly cluster data from the same class; and (ii) assign data from each class to a single cluster. It is therefore important to consider the class distribution within each discovered cluster as well as the cluster distribution withing each class. We compare algorithms using Purity (Zhao and Karypis, 2001) and V-Measure (Rosenberg and Hirschberg, 2007). Purity takes values in $(0,1]$, with higher values indicating a clustering in which each cluster contains observations almost exclusively from a single class. A disadvantage of this measure is that it does not penalise the splitting of data from one class between multiple clusters. V-Measure is defined as the harmonic mean of *homogeneity* and *completeness*. Homogeneity measures the conditional entropy of the class distribution within each cluster. Completeness is symmetric, and measures the conditional entropy of the cluster distribution within each class. V-Measure takes values in $[0,1]$, with high values indicating a clustering in which each cluster contains almost uniquely data from one class, and each class is contained almost entirely within a single cluster.

To compare the performance of algorithms we evaluate them on stream segments of length 100 taken every 200 time steps. Performance plots show performance evolution through time, indicating convergence rates and the ability of algorithms to react to and recover from non-stationarity, while tables document the overall performance of the algorithms on each stream environment.

## 6.1 Simulations

For all simulations data were generated from a mixture of $C$ multivariate Gaussian distributions. Covariance matrices were randomly generated according to,

$$\Sigma_i = 4\left(\frac{i}{C}\right)^2 S^\top S,\ S_{j,k} \sim N(0,1).$$

The coefficients $4(i/C)^2$ lead to classes with highly variable scales. The mixture proportions were determined by $p_i \propto u_i, u_i \sim U[1,2]$. The component means were uniformly sampled from a $d$-dimensional hypercube, $\mu_i \sim U[0, Cd^{1/4}]^d$. Reported results for simulated experiments are averages over 50 experiments.

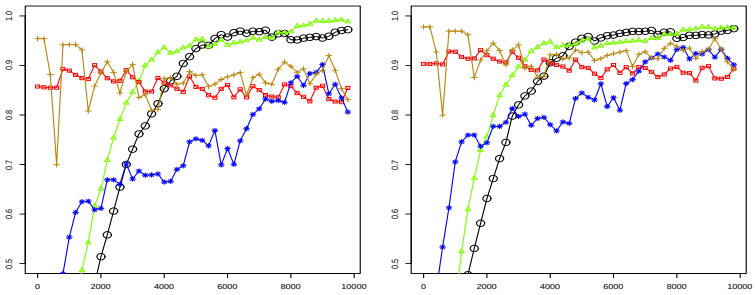
### Static Environments

We first consider static environments of dimensionality 50, 100 and 500 with 10, 20 and 30 classes. Streams were of length $500C$ to allow algorithms without an offline component to build their models. Figure 5.2 shows the case with 20 classes in 500 dimensions. The offline initialisation of CluStream and HPStream ensures good performance from the early stages of the stream. However, our algorithm is quickly able to surpass them. SPDC is less conservative in introducing splits than HSDC, however its instability causes the improvements to tail off rapidly. HSDC(I) generates robust splits more rapidly than HSDC because of projection inheritance, and so it is able to achieve high performance after fewer time steps. Table 5.1 contains a summary of the algorithms' performance on the final stream segment of length 100. The performance in the final segment of a static stream is most indicative of model performance since the algorithms have been given opportunity to converge and maintain their models. Our algorithms achieve substantially higher performance in the high dimensional examples, while being competitive in every case considered.

### Static Environments with Irrelevant Features

In high dimensional applications, often certain features are irrelevant to the class identity of the data. Being able to handle data with irrelevant or noisy features is therefore critical. We consider cases with 20 classes described by 100 relevant features. We explore the robustness of the algorithms to the number of irrelevant features as well as the degree of variability therein. The 100 relevant features were generated as described above. The data were then augmented with scaled standard (zero mean and identity covariance) Gaussian measurements for a variety of dimensions ($d$) and scaling factors ($S$). Figure 5.3 shows the case with 100 irrelevant dimensions with scaling factor

**Fig. 5.2:** Performance on Static Data Stream with 20 Classes in 500 Dimensions



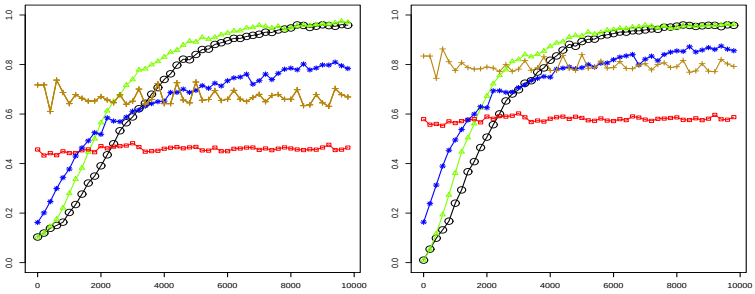
(a) C = 20, d = 500: Purity (b) C = 20, d = 500: V-Measure

HSDC (θ-θ), HSDC(I) (△-△), CluStream (+-+), HPStream (□-□), SPDC(∗-∗)

**Table 5.1:** Clustering Performance. Static environments. Average performance on the final stream segment. Standard deviation in parentheses. Highest performance in bold. Significantly lower performance indicated by *, based on a one sided $t$-test at the 5% level

| | d = 50 | | d = 100 | | d = 500 | |
|---|---|---|---|---|---|---|
| C = 10 | Purity | V-Measure | Purity | V-Measure | Purity | V-Measure |
| HPStream | **0.89** (0.04) | 0.85 (0.03)* | 0.86 (0.05) | 0.84 (0.04)* | 0.81 (0.05)* | 0.82 (0.04)* |
| SPDC | 0.88 (0.14) | **0.87** (0.10) | **0.89** (0.16) | **0.88** (0.15) | 0.79 (0.25)* | 0.82 (0.23)* |
| CluStream | 0.44 (0.06)* | 0.49 (0.06)* | 0.37 (0.05)* | 0.41 (0.06)* | 0.30 (0.04)* | 0.30 (0.05)* |
| HSDC | 0.84 (0.16)* | 0.83 (0.15) | 0.88 (0.19) | 0.86 (0.20) | 0.90 (0.14) | 0.91 (0.12) |
| HSDC(I) | 0.82 (0.16)* | 0.81 (0.15)* | 0.88 (0.17) | 0.86 (0.15) | **0.94** (0.12) | **0.92** (0.08) |
| C = 20 | | | | | | |
| HPStream | 0.87 (0.05)* | 0.91 (0.03)* | 0.86 (0.04)* | 0.89 (0.03)* | 0.85 (0.03)* | 0.89 (0.03)* |
| SPDC | 0.92 (0.15)* | 0.93 (0.11)* | 0.85 (0.21)* | 0.90 (0.15)* | 0.81 (0.16)* | 0.90 (0.11)* |
| CluStream | 0.94 (0.04)* | **0.97** (0.02) | 0.94 (0.04) | **0.97** (0.02) | 0.83 (0.13)* | 0.89 (0.09)* |
| HSDC | **0.98** (0.03) | **0.97** (0.02) | **0.96** (0.13) | 0.95 (0.10) | 0.97 (0.08) | 0.97 (0.05) |
| HSDC(I) | **0.98** (0.02) | 0.96 (0.02)* | **0.96** (0.09) | 0.95 (0.06)* | **0.99** (0.03) | **0.98** (0.02) |
| C = 30 | | | | | | |
| HPStream | 0.90 (0.06)* | 0.92 (0.04)* | 0.85 (0.09)* | 0.90 (0.06)* | 0.84 (0.06)* | 0.90 (0.04)* |
| SPDC | 0.81 (0.20)* | 0.89 (0.13)* | 0.73 (0.22)* | 0.84 (0.16)* | 0.77 (0.20)* | 0.86 (0.18)* |
| CluStream | 0.94 (0.03)* | 0.97 (0.01) * | 0.95 (0.02) | **0.98** (0.01) | 0.94 (0.04)* | 0.97 (0.04) |
| HSDC | **0.98** (0.06) | **0.98** (0.03) | **0.97** (0.12) | 0.97 (0.07) | 0.96 (0.14) | 0.97 (0.11) |
| HSDC(I) | 0.95 (0.13) | 0.96 (0.08) | **0.97** (0.09) | 0.97 (0.05) | **0.98** (0.08) | **0.98** (0.05) |

**Fig. 5.3:** Clustering Performance. Static Environment with Irrelevant Features. 20 Classes in 100 Relevant and 100 Irrelevant Dimensions with Moderate Variability



(a) S = 20, d = 100: Purity (b) S = 20, d = 100: V-Measure

HSDC (θ-θ), HSDC(I) (△-△), CluStream (+-+), HPStream (⊟-⊟), SPDC(∗-∗)

20. The performance of CluStream and HPStream is stable, but substantially diminished by the presence of the irrelevant features. HSDC and HSDC(I) both quickly surpass them and maintain stable performance after convergence. SPDC also outperforms CluStream and HPStream, but cannot achieve the same high levels of performance as our method. Table 5.2 contains a summary of the algorithms' performance on the final stream segment. HSDC and HSDC(I) are robust to the number of irrelevant features and the degree of noise therein, except in the most extreme case (S = 30, d = 200), where the high level of noise coupled with the large number of irrelevant features leads to slower splitting and a much higher incidence of false detection of change. CluStream achieves the highest performance in this most extreme case. HPStream seems unable to distinguish classes when the number of irrelevant features dominates the number of relevant ones (d=200).

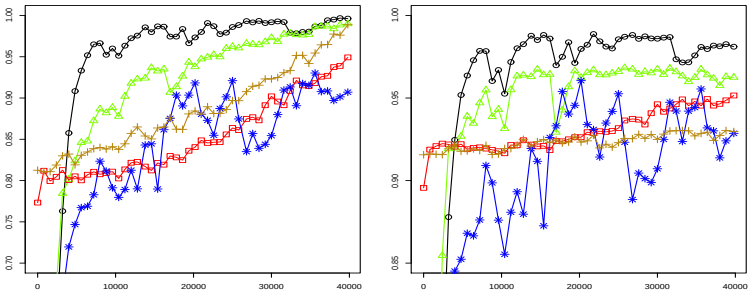
### Non-Stationary Environments

For these experiments we simulated environments in which the distribution undergoes abrupt changes at discrete points in time. Plots for the 500 dimensional cases are found in Figures 5.4-5.6, and a full summary of the results from non-stationary examples is found in Table 5.3. For these experiments we consider the performance of algorithms throughout the data streams. The table therefore reports the average and standard deviation of the average performance of each algorithm on stream segments of length 100 taken every 200 time steps.

**Table 5.2:** Clustering Performance. Static environments with irrelevant features. Average performance on the final stream segment. Standard deviation in parentheses. Highest performance in bold. Significantly lower performance indicated by *, based on a one sided *t*-test at the 5% level

| | d = 50 | | d = 100 | | d = 200 | |
|---|---|---|---|---|---|---|
| S = 10 | Purity | V-Measure | Purity | V-Measure | Purity | V-Measure |
| HPStream | 0.42 (0.12)* | 0.56 (0.10)* | 0.46 (0.08)* | 0.59 (0.08)* | 0.10 (0.02)* | 0.01 (0.04)* |
| SPDC | 0.85 (0.24)* | 0.90 (0.17)* | 0.80 (0.22)* | 0.86 (0.16)* | 0.83 (0.20)* | 0.89 (0.15)* |
| CluStream | 0.94 (0.03)* | **0.97** (0.02) | **0.94** (0.04) | **0.97** (0.02) | 0.91 (0.08)* | 0.95 (0.05) |
| HSDC | **0.99** (0.02) | **0.97** (0.02) | 0.92 (0.21) | 0.92 (0.19) | 0.96 (0.11) | **0.96** (0.07) |
| HSDC(I) | 0.97 (0.07)* | 0.96 (0.03)* | 0.92 (0.20) | 0.91 (0.20) | **0.97** (0.07) | **0.96** (0.05) |
| S = 20 | | | | | | |
| HPStream | 0.44 (0.10)* | 0.58 (0.09)* | 0.46 (0.08)* | 0.59 (0.08)* | 0.11 (0.02)* | 0.01 (0.04)* |
| SPDC | 0.83 (0.22)* | 0.89 (0.17)* | 0.78 (0.22)* | 0.86 (0.17)* | 0.71 (0.24)* | 0.80 (0.19)* |
| CluStream | 0.92 (0.05)* | **0.96** (0.03) | 0.67 (0.07)* | 0.79 (0.05)* | 0.65 (0.08)* | 0.78 (0.06)* |
| HSDC | **0.97** (0.09) | **0.96** (0.06) | 0.96 (0.09) | **0.96** (0.05) | 0.87 (0.21)* | 0.89 (0.19)* |
| HSDC(I) | 0.96 (0.12) | 0.95 (0.08) | **0.97** (0.06) | **0.96** (0.04) | **0.94** (0.10) | **0.95** (0.06) |
| S = 30 | | | | | | |
| HPStream | 0.42 (0.12)* | 0.56 (0.09)* | 0.40 (0.10)* | 0.51 (0.11)* | 0.11 (0.02)* | 0.02 (0.05)* |
| SPDC | 0.73 (0.23)* | 0.81 (0.19)* | 0.64 (0.21)* | 0.76 (0.18)* | 0.49 (0.20)* | 0.62 (0.22)* |
| CluStream | 0.72 (0.07)* | 0.83 (0.05)* | 0.72 (0.07)* | 0.75 (0.08)* | **0.71** (0.07) | **0.79** (0.05) |
| HSDC | **0.88** (0.20) | **0.90** (0.19) | 0.70 (0.28)* | 0.77 (0.26)* | 0.14 (0.10)* | 0.07 (0.18)* |
| HSDC(I) | **0.88** (0.20) | 0.89 (0.18) | **0.85** (0.20) | **0.88** (0.17) | 0.56 (0.27)* | 0.65 (0.28)* |

**Fig. 5.4:** Clustering Performance. Decreasing Classes. 500 Dimensions




(a) Decreasing Classes: Purity (b) Decreasing Classes: V-Measure

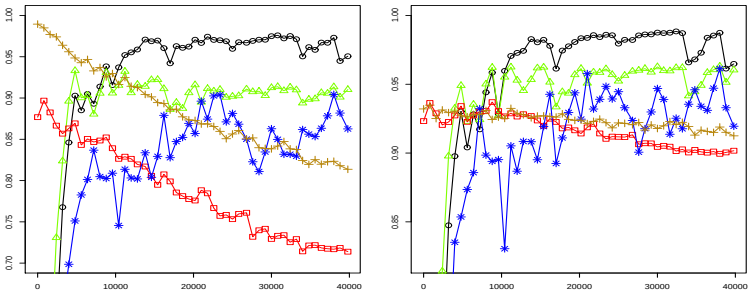
HSDC (θ-θ), HSDC(I) (△-△), CluStream (+-+), HPStream (□-□), SPDC(∗-∗)

**Variable Number of Classes** For results which lend themselves better to interpretation, we simulate two separate cases; streams with an increasing number of classes and streams with a decreasing number. For the former we split a randomly selected class every 1000 time steps, beginning with 20 classes and ending with 60. The reverse procedure was adopted for the case of decreasing number of classes. CluStream requires a fixed number of classes throughout. This was set to 40, the average number of classes over the stream. HPStream was initialised with the correct number for the initial stage of the stream.

For a decreasing number of classes (Figure 5.4) the performance of all algorithms improves as the stream progresses, since the environment becomes easier to model. Following initial convergence, the performance of our method surpasses the others. In the case of increasing number of classes (Figure 5.5) the performance of CluStream and HPStream deteriorates as the stream progresses. This highlights the limitation of having to specify a fixed number of clusters for the entire data stream. In contrast the performance of our method is stable after initial convergence. The sustained high performance indicates that HSDC is able to identify when clusters are being split.

**Distribution Overhaul** In this set of experiments the distribution undergoes complete change at regular intervals. We consider the case with 20 classes, whose parameters are reinitialised every 15000 time steps. The performance plots show a sudden deterioration in the performance of our method following each change. This is expected as the separating hyperplanes are likely redundant or intersect the new classes, and thus the model must be rebuilt from scratch. In all cases, however, HSDC is able to rebuild good qual-

**Fig. 5.5:** Clustering Performance. Increasing Classes. 500 Dimensions



(a) Increasing Classes: Purity (b) Increasing Classes: V-Measure

HSDC (θ-θ), HSDC(I) (△-△), CluStream (+-+), HPStream (□-□), SPDC(∗-∗)

ity clustering models, with no apparent deterioration after multiple changes. The instability of a direct implementation of projected divisive clustering is indicated by the performance of SPDC. CluStream is least affected by the changes, but cannot achieve the high performance of HSDC. Because of the multiple rebuilding stages, CluStream is able to achieve higher average performance than our method when taken over the entire stream.

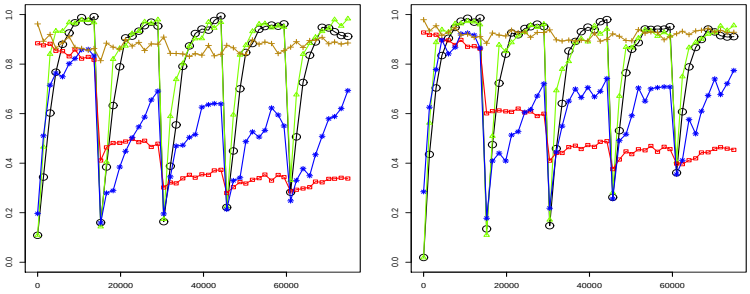
## 6.2 Publicly Available Data Sets

In this section we compare the performance of the algorithms on two publicly available data sets where the true class label is known. The first, Forest Cover Type, is lower dimensional and so (at best) we hope to achieve results comparable with state of the art standard data stream clustering algorithms such as CluStream. The second, Gas Sensor Array, is higher dimensional and we expect HSDC to perform better.

### Forest Cover Type

The Forest Cover Type data set, taken from the UCI Machine Learning Repository (Bache and Lichman, 2016), contains 581012 observations characteristed by 54 features, where each observation corresponds to one of seven forest cover types. As in the analyses in (Aggarwal et al., 2004; Tasoulis et al., 2012) we use only the ten continuous features. A plot of the class proportions (Figure 5.7) suggests considerable variability in the data distribution through the stream. Figure 5.8 shows the performance of the various algorithms through the stream (the series of performance values were smoothed for better interpretability). CluStream achieves the highest performance through the ma-

**Fig. 5.6:** Clustering Performance. Distribution Overhaul. 20 Classes in 500 Dimensions



(a) Distribution Overhaul: Purity (b) Distribution Overhaul: V-Measure

HSDC (θ-θ), HSDC(I) (△-△), CluStream (┼-┼), HPStream (⊟-⊟), SPDC(∗-∗)

**Table 5.3:** Clustering Performance. Drifting environments. Average performance on segments taken every 200 time steps. Standard deviation in parentheses. Highest performance in bold. Significantly lower performance indicated by *, based on a one sided *t*-test at the 5% level

| | Increasing Classes | | Decreasing Classes | | Overhaul | |
|---|---|---|---|---|---|---|
| d = 100 | Purity | V-Measure | Purity | V-Measure | Purity | V-Measure |
| HPStream | 0.85 (0.03)* | **0.93** (0.01) | **0.93** (0.02) | **0.94** (0.01) | 0.91 (0.02)* | 0.93 (0.00)* |
| SPDC | 0.83 (0.08)* | 0.89 (0.06)* | 0.83 (0.07)* | 0.89 (0.04)* | 0.51 (0.06)* | 0.56 (0.07)* |
| CluStream | 0.90 (0.01) | 0.92 (0.00)* | 0.90 (0.01)* | 0.92 (0.01)* | **0.94** (0.00) | **0.97** (0.00) |
| HSDC | **0.91** (0.04) | **0.93** (0.03) | 0.90 (0.04)* | 0.92 (0.03)* | 0.82 (0.04)* | 0.84 (0.04)* |
| HSDC(I) | **0.91** (0.04) | **0.93** (0.02) | 0.89 (0.04)* | 0.92 (0.03)* | 0.82 (0.04)* | 0.83 (0.03)* |
| d = 500 | | | | | | |
| HPStream | 0.79 (0.03)* | 0.92 (0.02)* | 0.85 (0.03)* | 0.93 (0.01)* | 0.46 (0.07)* | 0.57 (0.07)* |
| SPDC | 0.81 (0.08)* | 0.89 (0.05)* | 0.83 (0.07)* | 0.89 (0.05)* | 0.53 (0.06)* | 0.63 (0.05)* |
| CluStream | 0.88 (0.01)* | 0.91 (0.01)* | 0.88 (0.01)* | 0.91 (0.01)* | **0.86** (0.03) | **0.91** (0.03) |
| HSDC | **0.91** (0.03) | **0.93** (0.03) | **0.93** (0.03) | **0.94** (0.03) | 0.78 (0.04)* | 0.80 (0.03)* |
| HSDC(I) | 0.87 (0.03)* | 0.92 (0.02) | 0.89 (0.03)* | 0.93 (0.02)* | 0.82 (0.03)* | 0.83 (0.03)* |

**Fig. 5.7:** Class Proportions of Forest Cover Type

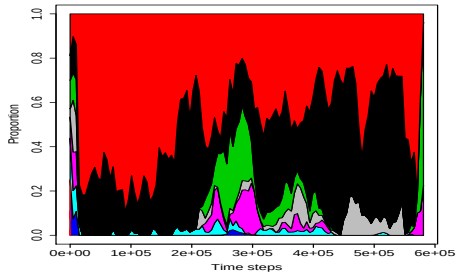
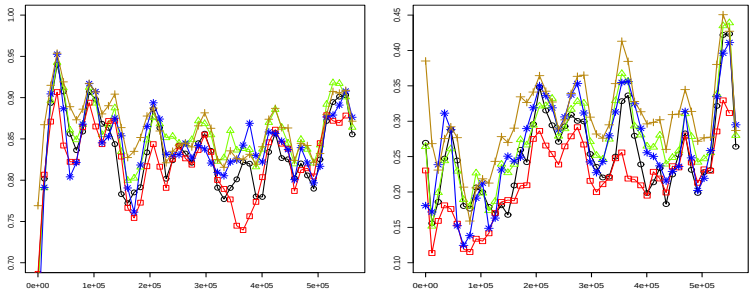



**Fig. 5.8:** Clustering Performance. Forest Cover Type Data



(a) Forest Cover Type: Purity (b) Forest Cover Type: V-Measure

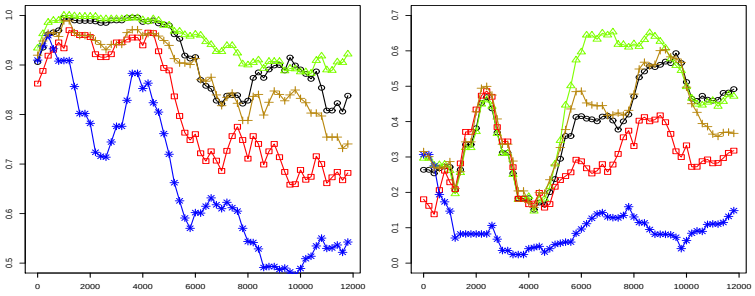
HSDC (θ-θ), HSDC(I) (△-△), CluStream (+-+), HPStream (⊟-⊟), SPDC(∗-∗)

jority of the stream. The average purity of all algorithms is similar; in decreasing order: CluStream = 0.86, HSDC(I) = 0.85, SPDC = 0.84, HSDC = 0.83, HPStream = 0.82. The average V-Measure of CluStream is substantially higher than the other algorithms at 0.31. The other algorithms achieved average V-Measure of: HSDC(I) = 0.27, SPDC = 0.27, HSDC = 0.25, HPStream = 0.22.

### Gas Sensor Array

The Gas Sensor Array Drift data set, available from the UCI Machine Learning repository (Bache and Lichman, 2016), contains 13910 measurements from each of 16 chemical sensors (amounting to 128 features in total per datum) used in simulations of drift compensation for the purpose of dis-

**Fig. 5.9:** Clustering Performance. Gas Sensor Array Data



(a) Gas Sensor Array: Purity (b) Gas Sensor Array: V-Measure

HSDC (θ-θ), HSDC(I) (△-△), CluStream (┼-┼), HPStream (⊟-⊟), SPDC(∗-∗)

criminating between six different gases (Ammonia, Acetaldehyde, Acetone, Ethylene, Ethanol, and Toluene) at various levels of concentration. The experiment was designed for the task of achieving al degradation in discriminatory performance over time as possible to evaluate strategies able to handle non-stationarity or drift (Vergara et al., 2012). The average number of dimensions for HPStream was set to 80 after experiments indicated this resulted in better performance. Figure 5.9 shows the performance of the algorithms through the stream (again the series are smoothed for better interpretability). Our method is the only one to obtain good discriminatory performance in the latter part of the stream, indicated by the purity performance. The instability of SPDC is again highlighted by its severe performance degradation with drift. The average purity (and V-Measure), in decreasing order: HSDC(I) = 0.94 (0.44), HSDC = 0.90 (0.39), CluStream = 0.87 (0.39), HPStream = 0.80 (0.30), SPDC = 0.67 (0.12).

## 6.3 Discussion of Experimental Results

We compared our method with three existing data stream clustering algorithms from the literature, CluStream, HPStream, and SPDC. We investigated the scalability of the method in terms of dimensionality and number of clusters. HSDC and HSDC(I) outperformed the compared methods in most cases, and especially in the highest dimensional examples. Next we investigated robustness to irrelevant/noisy features. The performance of our method was affected to a lesser degree than the other methods except in the most extreme case. When the number of irrelevant dimensions dominated the number of relevant ones, and in addition the degree of variability in the irrelevant

dimensions was large, HSDC failed to detect clusters reliably. The stability added by projection inheritance improved matters, but the performance was still strongly affected. Following that we introduced non-stationarity, considering three types: Clusters dividing, clusters merging, and distribution overhaul. Our method obtained the highest overall performance when the number of clusters varied (clusters dividing/merging). For distribution overhaul, because our method is required to rebuild its model from scratch, the average performance was strongly affected. Despite this fact, HSDC and HSDC(I) were always able to rebuild high quality clustering models, with no apparent degradation with repeated overhauls.

Finally we considered two real world applications: Clustering Forest Cover Types, and Gas Sensor Array Data. The former is lower dimensional (10 dimensions used), and our method obtained similar performance to the other methods considered. The latter is higher dimensional, and our method strongly outperformed the compared algorithms.

It is important to notice that in almost all cases where our approach does not yield the best performance, it is still close to the best performing method, while there are numerous examples in which our method far outperforms all others.

# 7 Conclusion

We introduced a framework for projected divisive clustering that is consistent with high density clustering, and which accommodates central challenges associated with data streams, including high dimensional data and non-stationarity. The derived method is able to identify clusters in arbitrary subspaces, estimate the number of clusters automatically and identify changes in the data distribution which affect the validity of the model. The algorithm is also fully incremental, requiring no offline component. To our knowledge, no other algorithms achieve all of these simultaneously.

The method constructs a hierarchy of low-density hyperplane separators, thereby enabling it to handle high dimensional data. This is achieved by establishing directions of high variability and approximating the projected distribution along them. We propose a simple approach to speed up the incremental approximation of such directions, hence reducing the time required to construct the cluster hierarchy. We introduce new components to the model only when the marginal distribution along the projection is found to be multimodal. For this purpose, a fixed memory approximation to the dip test is developed, and shown to lower bound the true value.

The framework incorporates a novel formulation to detect arbitrary changes in the population distribution that affect the validity of the current clustering model. The approach relies on detecting increases in the density local to a

separating hyperplane, which signal that the hyperplane intersects regions of high density. This enables us to not only provide indication of the timing of such changes but also to isolate the parts of the hierarchical model that require revision. To our knowledge this is the first general purpose clustering algorithm able to detect changes in the underlying distribution.

Experimental results show that algorithms derived from the proposed framework obtain the best overall performance on a range of problem types, when compared with state of the art algorithms for data stream clustering. Among the competing methods, CluStream achieved an on-average better performance when the population distribution underwent the most extreme type of abrupt change. The reason is that reconstructing the entire clustering hierarchy invariably requires more time than the adaptation of centroid-based methods. This result however is conditional on the a priori specification of the correct number of clusters for such algorithms. When this is not the case our methods fare better.

# Appendix. Proofs

Before we can prove Lemma 28, we require the following preliminaries.

The algorithm for computing the dip of a distribution function $F$ constructs a unimodal distribution function $G$ with the following properties: (i) The modal interval of $G$, $[m,M]$, is equal to the modal interval of the closest unimodal distribution function to $F$, which we denote by $F^U$, based on the supremum norm; (ii) $\|F-G\|_\infty = 2\|F-F^U\|_\infty$; (iii) $G$ is the greatest convex minorant of $F$ on $(-\infty,m]$; (iv) $G$ is the least concave majorant of $F$ on $[M,\infty)$. By construction, the function $G$ is linear between its *nodes*. A node $n \leq m$ of $G$ satisfies $G(n) = \liminf_{x\to n} F(x)$, while a node $n \geq M$ of $G$ satisfies $G(n) = \limsup_{x\to n} F(x)$. If $F$ is the distribution function of a discrete random variable, then $G$ is continuous.

The function $F^U$ can be constructed by finding appropriate values $b < m$, $B > M$ s.t. $F^U$ is equal to $G + \mathrm{Dip}(F)$ on $[b,m]$, equal to $G - \mathrm{Dip}(F)$ on $[M,B]$, linearly interpolating between $G(m)$ and $G(M)$ and given any appropriate tails, which we choose to be linearly decreasing/increasing to 0 and 1 respectively.

Before proving Lemma 28, we require the following preliminary result, which relies on the notion of a *step linear* function.

**Step Linear** A function $f$ is *step linear* on a non-empty, compact interval $I = [a,b]$, if

$$f(x) = \alpha + \beta \left\lfloor (x-a)\frac{n}{b-a} \right\rfloor, \ \forall x \in I,$$

for some $\alpha, \beta \in \mathbb{R}$ and $n \in \mathbb{N}$.

A step linear function is piecewise constant, and has $n$ equally sized jumps of size $\beta$ spaced equally on $I$ with the final jump ocurring at $b$. The approximate empirical distribution function $\tilde{F}$ (Section 4.2) is therefore step linear over the approximating intervals.

**Proposition 30** *Let $f$ be step linear on the closed interval $I=[a,b]$, and satisfy $\lim_{x\to a^-} f(x)=\alpha-\beta$, where $\alpha,\beta$ as in the above definition for $f$. Let $g$ be linear on $I$ and continuous on a neighbourhood of $I$. Then*

$$\sup_{x\in I}|f(x)-g(x)|\leq\max\{limsup_{x\to a}|f(x)-g(x)|,limsup_{x\to b}|f(x)-g(x)|\}.$$

***Proof*** *Let $f_m$ and $f^M$ be linear on a neighbourhood of $I$ s.t. they form the closest lower and upper bounding functions of $f$ on $I$ respectively. Since $f$ is step linear, we have,*

$$\lim_{x\to a^-} f(x)=f_m(a),\quad \lim_{x\to b^-} f(x)=f_m(b),$$
$$f(a)=f^M(a),\qquad f(b)=f^M(b).$$

*We therefore have, by above and the fact that $g,f_m$, and $f^M$ are linear on $I$,*

$$\begin{aligned}\sup_{x\in I}|f(x)-g(x)| &\leq \max\left\{\sup_{x\in I}|f^M(x)-g(x)|,\sup_{x\in I}|f_m(x)-g(x)|\right\}\\ &=\max\Big\{|f^M(b)-g(b)|,|f^M(a)-g(a)|,\\ &\qquad |f_m(a)-g(a)|,|f_m(b)-g(b)|\Big\}\\ &=\max\{limsup_{x\to a}|f(x)-g(x)|,limsup_{x\to b}|f(x)-g(x)|\}.\end{aligned}$$

■

We are now in a position to prove Lemma 28, which states that the dip of a compactly approximated sample, as described in Section 4.2, provides a lower bound on the dip of the true sample.

**Proof of Lemma 28.**

**Proof** Let $I=[a,b]$ be any compact interval and $F_I$ the empirical distribution function of $(\mathcal{X}\cap I^c)\cup\text{Unif}(\mathcal{X},I)$. Assume $|\mathcal{X}\cap I|>1$, since otherwise $F_I=F_{\mathcal{X}}$ and we are done. We can assume that the endpoints of $I$ are elements of $\mathcal{X}$ since this defines the same uniform set. $F_{\mathcal{X}}$ and $F_I$ are therefore equal on $\text{Int}(I)^c$. In fact, since $\mathcal{X}$ consists of unique points, $\exists\epsilon>0$ s.t. $F_I(x)=F_{\mathcal{X}}(x)\ \forall x\notin(a+\epsilon,b-\epsilon)$. Define $F_I'$ to be equal to $F_{\mathcal{X}}^U$ for $x\notin\text{Int}(I)$ and linearly interpolating between $F_X^U(a)$ and $F_X^U(b)$. By construction $F_I'$ is a continuous unimodal distribution function.

We now show $\|F_I - F_I'\|_\infty \leq \|F_\mathcal{X} - F_\mathcal{X}^U\|_\infty$. To see this, suppose that it is not true, i.e., $\exists x$ s.t. $|F_I(x) - F_I'(x)| > \sup_y |F_\mathcal{X}(y) - F_\mathcal{X}^U(y)|$. Clearly $x \in \text{Int}(I)$ due to the equalities discussed above and the construction of $F_I'$. Because of the continuity of $F_\mathcal{X}^U$ and $F_I'$ and the equality of $F_\mathcal{X}$ and $F_I$ on $(a, a+\epsilon) \cup (b-\epsilon, b)$, we have

$$\text{limsup}_{y \to a} |F_I(y) - F_I'(y)| = \text{limsup}_{y \to a} |F_\mathcal{X} - F_\mathcal{X}^U(x)|$$

and

$$\text{limsup}_{y \to b} |F_I(y) - F_I'(y)| = \text{limsup}_{y \to b} |F_\mathcal{X} - F_\mathcal{X}^U(x)|.$$

But by Proposition 30 one of these left hand sides is at least as large as $|F_I(x) - F_I'(x)|$, leading to a contradiction.

We have shown that the addition of a single interval cannot increase the dip. We can apply the same logic to the now modified sample $(\mathcal{X} \cap I^c) \cup \text{Unif}(\mathcal{X}, I)$, iterating the addition of disjoint intervals to obtain a non-increasing sequence of dips. ■

In the above proof, we do not show that $F_I'$ is the closest unimodal distribution function to $F_I$, however its existence necessitates the closest one being at least as close. Now, the sample approximations we employ still contain a full $t$ atoms after $t$ observations, however, they can be stored in $\mathcal{O}(k)$ for $k$ intervals. We can easily show that the dip of such a sample approximation can be computed in $\mathcal{O}(k)$ time.

**Proposition 31** *The dip of a sample consisting of k uniform sets with disjoint ranges can be computed in $\mathcal{O}(k)$ time.*

**Proof** We begin by showing that there exists a unimodal distribution function which is linear on the ranges of the uniform sets and which achieves the minimal distance to the empirical distribution function of the sample.

Let $F$ be a continuous unimodal distribution function s.t. $\|F - \tilde{F}\|_\infty = \text{Dip}(\tilde{F})$. Define $F'$ similarly to in the above proof to be the continuous distribution function which is equal to $F$ outside and at the boundaries of the intervals defining the uniform sets and linearly interpolating on them. Using the same logic, we know that $\sup_x |F'(x) - \tilde{F}(x)| \leq \sup_x |F(x) - \tilde{F}(x)|$, hence $\|F' - \tilde{F}\|_\infty = \text{Dip}(\tilde{F})$.

Proposition 30 ensures that points in the interior of the intervals will not be chosen by the dip algorithm as end points of the modal interval of $G$, nor points at which the difference between the functions is supremal. The possible choices for these locations is therefore $\mathcal{O}(k)$, and the algorithm need not evaluate the functions except at the endpoints of the intervals. ■

Finally, we provide a proof of Proposition 29.

**Proof of Proposition 29.**

**Proof** For $s>1$ we have

$$\|v_s - v_{s-1}\| = \|v_s\|\|v_s - v_{s-1}\| \geq |v_s \cdot (v_s - v_{s-1})| = |v_s v_{s-1} - 1|,$$

since $\|v_t\| = 1 \ \forall t$. Therefore, since $\{v_t\}_{t=1}^{\infty}$ is almost surely convergent, and therefore almost surely Cauchy, we have $v_s \cdot v_{s-1} \xrightarrow{a.s.} 1 \Rightarrow \arccos(v_s \cdot v_{s-1}) \xrightarrow{a.s.} 0$. Now, we can easily show that,

$$\lambda_t \leq \gamma^{t-1}\lambda_1 + (1-\gamma)\sum_{i=1}^{t-2}\gamma^i \arccos(v_{t-i} \cdot v_{t-i-1}).$$

Take $\epsilon > 0$ and $t$ large enough that $\gamma^{t-1}\lambda_1 < \gamma\epsilon$, and $t > k+2$, where $k = \lfloor \log(\epsilon(1-\gamma)/2\pi)/\log(\gamma) - 1 \rfloor$. Consider,

$$\sum_{i=1}^{t-2}\gamma^i \arccos(v_{t-i} \cdot v_{t-i-1}) \leq \sum_{i=1}^{k}\arccos(v_{t-i} \cdot v_{t-i-1}) + \frac{\pi\gamma^{k+1}}{1-\gamma},$$

and $\frac{\pi\gamma^{k+1}}{1-\gamma} \leq \frac{\epsilon}{2}$. In all,

$$\lambda_t > \epsilon \Rightarrow \sum_{i=0}^{k}\arccos(v_{t-i} \cdot v_{t-i-1}) > \epsilon/2.$$

Notice that $k$ does not depend on $t$. With probability 1, for any given $\epsilon > 0$ there is a $\mathcal{T}$ s.t. $T > \mathcal{T}$ implies $\sum_{i=0}^{k}\arccos(v_{T-i} \cdot v_{T-i-1}) \leq \epsilon/2$, implying $\lambda_T \leq \epsilon$ for all $T > \mathcal{T}$, and the result follows. ■

# Acknowledgements

David Hofmeyr gratefully acknowledges funding from both the Engineering and Physical Sciences Research Council (EPSRC) and the Oppenheimer Memorial Trust.

# Chapter 6

# Conclusion

## 1 Summary of Contributions

This thesis consists of four new approaches to the important problem of identifying groups, or clusters of related data, which has applications in all areas of scientific research from robotics to microarray analysis to marketing strategy and many more. The contributions are motivated by two fundamental challenges in the context of data clustering, dimension reduction and to a lesser extent data streams. The problem of dimension reduction is treated in a highly principled manner, by identifying subspaces in which the data represent a clusterable structure in relation to fundamental definitions of *cluster*. The combination of high dimensionality and sequential arrival of data, as in a data stream, is handled by integrating the tasks of dimension reduction and incremental learning which allows for some of the main difficulties in data streams to be tackled effectively.

In Chapter 2 a new hyperplane based classifier is proposed for unsupervised and semi-supervised classification. The optimal hyperplane is defined as that which has the minimum integrated density along it, while inducing a meaningful partition of a data set. This approach is motivated by the widely regarded low density separation assumption; (high density) clusters are regions of relatively high density separated by low density regions. The majority of existing approaches which are based on this assumption attempt to find the hyperplane which has the largest margin on the data set, which is at best asymptotically connected with the actual lowest density separator. The proposed approach directly estimates the density using kernel density estimation, and so has a much more pleasing interpretation in the finite sample setting. It is apparent that no other existing methods have attempted to solve the low density separation problem directly in the finite sample setting.

The optimal hyperplane for the proposed problem is shown to be connected with the more common approach, in that it converges to the largest margin hyperplane as the bandwidth used in the kernel estimator is reduced to zero.

A projection pursuit formulation of the optimisation objective is proposed, which is able to significantly mitigate the problem of convergence to a poor quality local minimum. The resulting objective function is non-smooth, however utilising modern methods for non-smooth optimisation means that the approach can be implemented practically and efficiently. This approach is shown to reliably identify low density hyperplanes in many practical applications.

Chapter 3 is based on a different definition of clusters, which is that a cluster is a relatively highly connected subgraph of a graph defined over the data set, in which edges represent the similarities between data. This definition has been widely adopted in recent years due to the success of spectral clustering. Spectral clustering solves a relaxation of the associated *normalised graph cut* objective via an eigen decomposition of the so-called graph Laplacian matrix. Determining the optimal subspace for this graph partitioning objective requires minimising the eigenvalues of the graph Laplacian of the data within that subspace. Eigenvalue optimisation problems are notoriously difficult due to their non-smooth behaviour. A globally convergent algorithm is proposed which uses directional derivatives where necessary to escape troublesome points. The eigenvalue problems are computationally demanding, requiring $\mathcal{O}(n^2)$ operations each. To mitigate this, an approximation is developed which has provable approximation error and does not result in an appreciable degradation in the quality of the optimal clustering solution.

The spectral clustering solution has been shown to be connected with a simple clustering solution from an embedding of the data within a high dimensional feature space. While highly effective in many applications, this can sometimes make the interpretation of the clusters within the input space difficult to intuit, when compared to something like high density clusters. It has been established, however, that these are in fact related and that the graph partitioning objective associated with spectral clustering also leads to low density separation. A new perspective on this is presented herein, as it is shown in Chapter 3 that the optimal subspace for spectral clustering converges to the subspace normal to the maximum margin hyperplane as the scaling is reduced to zero.

Chapter 4 is very closely related with the contents of Chapter 3 in terms of underlying motivation. The clustering objective is again defined in terms of the normlised graph cut. Neither work can be seen as a strict improvement over the other, however. Chapter 4 provides a more computationally efficient

method which exploits the trivial factorisation of its similarity function to find the optimal partition within a subspace in log-linear time (as opposed to the quadratic time eigen problem). It also computes the normalised cut value exactly, and does not rely on any approximations. It is however limited in that the fast computation is only possible for a specific similarity function, and the partition of the data is constrained to the set of hyperplane separations. One major benefit of the formulation in Chapter 4 is that the theoretical analysis is comparatively less challenging than for the spectral relaxation. It is therefore possible to establish asymptotic results for an increasing sample in terms of an assumed underlying probability distribution. The value of the normalised cut across a hyperplane is shown to converge almost surely, and the asymptotic value has desirable characteristics as an objective for clustering. The optimal hyperplane is likely to both pass through low density regions, and also separate the modes (high density clusters) of the underlying density. The optimal hyperplane is also again shown to converge to the largest margin hyperplane through the data, as the scaling is reduced to zero.

The final contribution is given in Chapter 5, and considers the problem of clustering data which are both high dimensional and arrive in a data stream. It is very challenging to obtain an optimal dimension reduction in an incremental setting, like a data stream. Indeed no existing methods can be found for this problem. The method in Chapter 5 therefore uses the extremely popular principal component analysis to perform dimension reduction. It then incrementally learns the data distribution within the principal subspace. Information from the rate at which the principal subspace is updated is used to inform the rate of forgetting in the estimation of the distribution. Asymptotically the subspace no longer updates, and so the distribution estimation no longer discounts past information. An incremental version of the dip test for unimodality is developed, which means that a statistically robust divisive procedure can be implemented in which a cluster is split by a low density hyperplane only when there is strong evidence that the cluster represents multiple modes in the underlying density.

The model structure provides a convenient framework to address the very challenging problem of change detection in a high dimensional data stream framework. Because it is the low density separating hyperplanes which form the model, it is not necessary to identify every change which affects the clusters. Only when a cluster is intersected by a separating hyperplane is there any need to revise the clustering model. New clusters which spontaneously arise can be separated at a later time by an additional hyperplane, unless of course they unluckily arise where they are intersected. All changes which are relevant to the clustering model can be addressed by the comparatively simple task of detecting an increase in density along the separating hyperplanes, which is approximated by detecting an increase in the frequency of

data arriving in a small neighbourhood around each such hyperplane.

# 2 An Experimental Comparison of the Contributions

This thesis presented a number of new methods for projected divisive clustering, where in each case a fundamental clustering objective was optimised via projection pursuit. In addition, two of the methods were extended to the related problem of semi-supervised classification. All of these methods were found to be very competitive with the state-of-the-art in terms of clustering/classification accuracy. With these comparisons in place, it is interesting to compare these new methods with one another, to better understand their respective merits and possible shortcomings.

## 2.1 Projected Divisive Clustering

This subsection presents a comparison of the projection methods developed in Chapters 2, 3 and 4. The data stream method discussed in Chapter 5 tackles a different problem, and so is not considered in this comparison. In all cases a complete clustering result is obtained by recursively obtaining binary partitions of (subsets of) the datasets, where the binary partitions optimise the following clustering objectives:

1. Low Density Separation: The $\text{MDP}^2$ method minimises the density on the hyperplane separation, as described in Chapter 2.

2. Minimum Spectral Connectivity: The $\text{SCP}_c^2$ and $\text{SCP}_o^2$ methods minimise the second eigenvalue of the standard Laplacian of data projected into one and two dimensional subspaces respectively. Similarly the $\text{SC}_n\text{P}_c^2$ and $\text{SC}_n\text{P}_o^2$ minimise the second eigenvalue of the normalised Laplacian. These methods are discussed in Chapter 3.

3. Minimum Normalised Cut: The NCutH method, Chapter 4, minimises the exact normalised cut measured across a separating hyperplane.

Tables 6.1 and 6.2 show the Purity and V-Measure of the methods applied to a collection of datasets taken primarily from the UCI machine learning repository (Bache and Lichman, 2016), and discussed in greater detail in previous chapters. Note that the methods based on minimising spectral connectivity rely on a non-deterministic approximation, and the results presented in the table are averages from 30 repeated experiments on each dataset.

The $\text{SC}_n\text{P}_o^2$ obtains the highest, or tied highest performance almost twice as often as the next most frequent (NCutH) in terms of purity, while in the

**Table 6.1:** A Comparison of the Performance of Proposed Methods for Projected Divisive Clustering. The Table Shows 100 × Purity on Benchmark Data Sets. Highest Performance in Each Case is Highlighted in Bold.

| | $\text{MDP}^2$ | $\text{SCP}_o^2$ | $\text{SCP}_c^2$ | $\text{SC}_n\text{P}_o^2$ | $\text{SC}_n\text{P}_c^2$ | NCutH |
|---|---|---|---|---|---|---|
| br. cancer | 95.85 | 96.85 | 97.00 | 97.00 | **97.28** | 96.85 |
| ionosphere | **71.23** | **71.23** | 70.09 | **71.23** | 70.09 | **71.23** |
| opt. digits | 80.11 | 81.49 | **82.97** | 81.19 | 73.74 | 78.71 |
| pen digits | 75.17 | 77.70 | 76.46 | **78.17** | 73.97 | 77.98 |
| voters | 84.60 | 84.37 | 82.76 | **84.83** | 83.91 | 84.60 |
| image seg. | 65.40 | 61.60 | 67.81 | 66.21 | **70.12** | 62.19 |
| satellite | **76.39** | 74.95 | 73.38 | 75.41 | 73.85 | 74.00 |
| chart | 79.67 | 83.67 | 83.00 | **86.50** | 85.33 | 66.67 |
| yeast | 73.35 | 75.36 | 74.50 | **75.50** | 70.06 | 75.07 |
| soybean | 68.52 | 67.06 | 65.30 | 66.33 | 69.69 | **80.79** |
| dermatology | 89.07 | 94.00 | 90.16 | 85.79 | 91.26 | **96.17** |
| glass | 51.87 | 49.07 | 56.07 | 58.41 | **60.75** | 54.21 |
| parkinsons | **75.38** | **75.38** | **75.38** | **75.38** | **75.38** | **75.38** |
| m.f. digits | 80.10 | 83.44 | 78.72 | **84.17** | 78.50 | 72.25 |

case of V-Measure these two methods are tied for most frequent highest performance. The comparisons between the other methods are less clear when considering these tables, and a summary of the performance is given in Section 2.3.

## 2.2 Large Margin Clustering

One of the appealing properties of the methods presented in this thesis is that their associated objectives are connected, in that asymptotically, as their respective smoothing parameters are reduced to zero, the optimal solutions all converge to the maximum margin clustering solution. Solving the associated problems for a shrinking sequence of smoothing parameters therefore provides a practical way of locating large margin separators for clustering. The non-convexity of the objectives in the proposed methods, however, means that the globally maximum margin solution is not necessarily obtained. In this subsection, the proposed approaches for large margin clustering are compared with one another.

The procedure of repeatedly solving the projection pursuit problems for a shrinking smoothing parameter was implemented explicitly in the cases of

**Table 6.2:** A Comparison of the Performance of Proposed Methods for Large Margin Clustering. The Table Shows 100 × V-Measure on Benchmark Data Sets. Highest Performance in Each Case is Highlighted in Bold.

| | $\text{MDP}^2$ | $\text{SCP}_o^2$ | $\text{SCP}_c^2$ | $\text{SC}_n\text{P}_o^2$ | $\text{SC}_n\text{P}_c^2$ | NCutH |
|---|---|---|---|---|---|---|
| br. cancer | 73.55 | 78.55 | 79.83 | 79.39 | **81.21** | 78.70 |
| ionosphere | 13.49 | **13.71** | 13.20 | **13.71** | 13.20 | 13.49 |
| opt. digits | 73.51 | 76.89 | **77.55** | 77.35 | 72.22 | 71.62 |
| pen digits | 71.53 | 74.83 | 73.89 | **75.89** | 73.58 | 70.95 |
| voters | **42.82** | 41.87 | 38.47 | 41.80 | 41.02 | **42.82** |
| image seg. | 62.69 | 61.59 | 65.28 | 64.67 | **67.64** | 59.38 |
| satellite | **62.88** | 60.90 | 59.11 | 61.49 | 60.08 | 59.63 |
| chart | 77.15 | 77.21 | 77.28 | **80.70** | 79.56 | 79.65 |
| yeast | 58.25 | 53.97 | 53.97 | 56.57 | 45.50 | **59.92** |
| soybean | 71.24 | 69.77 | 68.91 | 69.60 | 69.98 | **80.29** |
| dermatology | 82.09 | 89.25 | 82.66 | 85.18 | 86.33 | **93.56** |
| glass | 31.88 | 27.66 | **34.38** | 31.98 | 30.89 | 31.58 |
| parkinsons | **27.62** | 20.54 | 16.44 | 19.05 | 17.64 | 21.96 |
| m.f. digits | 75.53 | 75.60 | 72.60 | **75.90** | 73.98 | 67.22 |

both spectral connectivity via the LMSC method, and minimum normalised cut hyperplanes as in $\text{NCutH}_0$. Here the same procedure is applied to the minimum density hyperplane projection pursuit method, which will be labelled $\text{MDP}_0^2$ in the following.

Tables 6.3 and 6.4 show the Purity and V-measure for these methods applied to the same collection of datasets as used above. Here the $\text{MDP}_0^2$ approach achieves the highest performance most often, while the remaining methods offer little for comparison based only on the frequency of highest performance. An exception to this is the large margin spectral connectivity method with univariate projection, LMSC, which only achieved the highest V-Measure performance in a single case.

## 2.3 A Summary of Clustering Performance

Here the results for both the base projection methods and the large margin limit results are combined to provide an overall comparison of all methods developed in this thesis for divisive clustering in the offline setting. For each method and each dataset the relative performance, as described in Chapter 3, and the regret, described in Chapter 2, are computed. Figures 6.1 and

**Table 6.3:** A Comparison of the Performance of Proposed Methods for Large Margin Clustering. The Table Shows 100 × Purity on Benchmark Data Sets. Highest Performance in Each Case is Highlighted in Bold.

| | $\text{MDP}_0^2$ | $\text{LMSC}_o$ | LMSC | $\text{NCutH}_0$ |
|---|---|---|---|---|
| br. cancer | 96.57 | 96.28 | 96.71 | **96.85** |
| ionosphere | 70.94 | 70.66 | **71.23** | **71.23** |
| opt. digits | **84.91** | 73.89 | 68.79 | 79.69 |
| pen digits | 77.21 | **79.86** | 71.18 | 77.98 |
| voters | 83.68 | 83.91 | 83.91 | **84.37** |
| image seg. | 64.23 | 59.51 | **64.80** | 63.79 |
| satellite | **76.01** | 74.94 | 72.49 | 75.76 |
| chart | 81.33 | **88.83** | 82.83 | 83.83 |
| yeast | 73.07 | 74.79 | **75.36** | 74.93 |
| soybean | **67.64** | 67.20 | 67.06 | 67.45 |
| dermatology | **94.54** | 86.07 | 85.52 | 85.52 |
| glass | **59.81** | 54.67 | 57.94 | 58.88 |
| parkinsons | **75.38** | **75.38** | **75.38** | **75.38** |
| m.f. digits | 80.25 | **81.80** | 73.94 | 71.55 |

6.2 show box plots of the relative purity and V-measure for each method across all datasets considered. The methods are ordered with decreasing mean value, where these mean values are indicated with red dots. It is apparent that methods based on minimising spectral connectivity obtained high purity values in general, but the performance of the associated large margin method is relatively poor. Possibly the most obvious observation is that the performance of the one dimensional large margin method based on spectral connectivity shows the worst performance overall. On the other hand the large margin limiting solution for the minimum density hyperplane approach seems to substantially improve purity performance over the base method, and is one of the highest performing methods in terms of purity. In the case of the minimum normalised cut hyperplane, there is very little difference between the base method and the large margin limit. Comparisons based on V-measure lead to quite different conclusions. Here the minimum density approach shows the best overall performance, and in general the minimum spectral connectivity approaches fare worse than others, with the exception being $\text{SC}_n\text{P}_o^2$. The normalised cut hyperplane method has middling performance in terms of both purity and V-measure, being slightly below average in the former and slightly above in the latter.

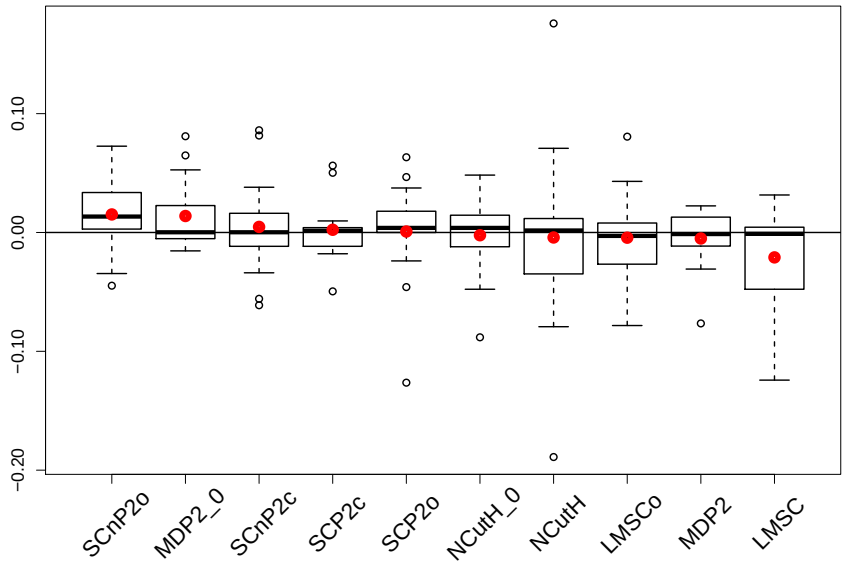
**Table 6.4:** A Comparison of the Performance of Proposed Methods for Large Margin Clustering. The Table Shows 100 × V-Measure on Benchmark Data Sets. Highest Performance in Each Case is Highlighted in Bold.

| | $MDP_0^2$ | $LMSC_o$ | LMSC | $NCutH_0$ |
|---|---|---|---|---|
| br. cancer | 77.46 | 75.86 | 77.99 | **78.68** |
| ionosphere | 12.99 | 12.30 | **13.49** | **13.49** |
| opt. digits | **76.89** | 69.88 | 62.33 | 74.65 |
| pen digits | 72.40 | **76.39** | 66.25 | 73.07 |
| voters | 41.13 | **43.38** | 37.78 | 42.82 |
| image seg. | 60.60 | 59.88 | 63.32 | **63.81** |
| satellite | 61.92 | 60.71 | 56.82 | **62.68** |
| chart | 81.32 | **85.34** | 78.23 | 77.05 |
| yeast | 58.78 | 56.26 | 57.89 | **59.01** |
| soybean | **71.96** | 70.68 | 71.36 | 71.87 |
| dermatology | **91.31** | 85.96 | 85.41 | 83.82 |
| glass | **36.14** | 31.29 | 28.72 | 30.85 |
| parkinsons | **22.84** | 21.96 | 21.96 | 21.96 |
| m.f. digits | **75.77** | 75.39 | 68.84 | 67.39 |

Figures 6.3 and 6.4 contain analogous plots based on regret. Here the methods are ordered with increasing mean value so that overall performance is again decreasing when moving left to right. Again the large margin method arising from the minimum density hyperplane approach and the multivariate minimum spectral connectivity method using the normalised Laplacian, $SC_nP_o^2$, achieve very strong performance when compared with others, while the LMSC method fares relatively poorly. The minimum normalised cut hyperplane approach shows substantially different performance between purity and V-measure, while other methods seem to more or less retain their rank between the two performance measures.

In conclusion it is apparent that the large margin method based on minimum density hyperplanes, and the multivariate normalised spectral connectivity approaches show especially strong performance for clustering. On the other hand the large margin method using spectral connectivity shows comparatively poor performance overall.

**Fig. 6.1:** Box plots of relative purity with additional red dots to indicate means. Methods are ordered with decreasing mean value.




## 2.4 Semi-supervised Classification

The minimum density hyerplane and minimum spectral connectivity approaches were extended to the problem of semi-supervised classification in Chapters 2 and 3 respectively. This section presents a very brief comparison of the performance of these two methods on the collection of datasets used in Chapter 3. Table 6.5 shows the average classification accuracy of these methods on a selection of datasets from the UCI machine learning repository (Bache and Lichman, 2016). The table illustrates that when the number of labelled data is very small, in this case 2% of the total number of data, the minimum density hyperplane approach outperforms the spectral connectivity approach. However, for a larger number of labelled data (10% and 25% of the total number of data) the reverse comparison is made. Table 6.6 again shows the average classification of these methods, this time applied to a selection of benchmark semi-supervised classification datasets taken from Chapelle et al. (2006b). In this case there is a very clear indication that the performance of the minimum spectral connectivity approach is superior to the minimum density hyperplane approach, as it obtains higher average performance in the vast majority of cases and in multiple examples the improvement is substantial.

**Fig. 6.2:** Box plots of relative V-measure with additional red dots to indicate means. Methods are ordered with decreasing mean value.

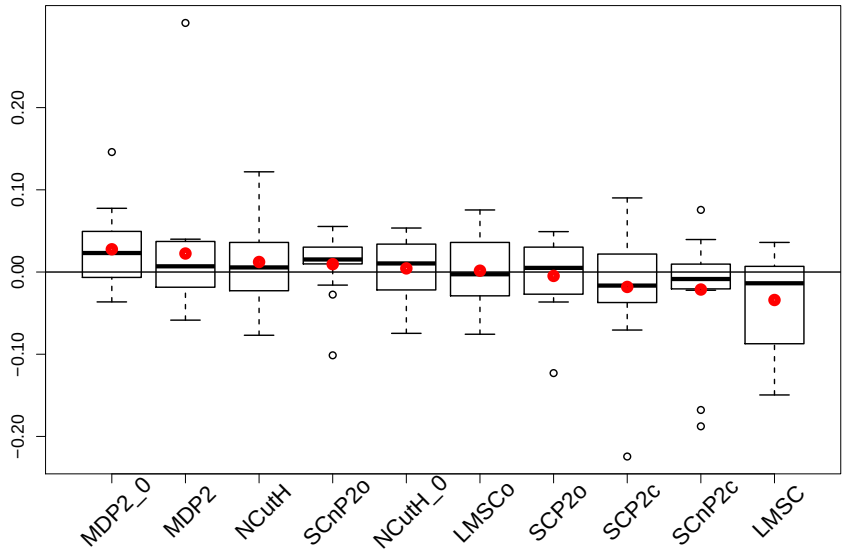



**Table 6.5:** A Comparison of the Proposed Methods for Semi-supervised Classification applied to UCI Machine Learning Repository Classification Data Sets. Average Accuracy (%) over 10 Splits.

| | Mam. | Vote. | Canc. | Iono. | Park. |
|---|---|---|---|---|---|
| | 2% Labelled Examples | | | | |
| $S^3CP^2$ | **79.62** | 84.53 | 96.18 | 66.44 | 74.71 |
| $MDP^2$ | 76.10 | **85.26** | **96.45** | **71.95** | **78.38** |
| | 10% Labelled Examples | | | | |
| $S^3CP^2$ | **81.64** | **90.74** | 96.04 | **85.71** | **79.94** |
| $MDP^2$ | 73.60 | 89.26 | **96.08** | 80.83 | 79.60 |
| | 25% Labelled Examples | | | | |
| $S^3CP^2$ | **82.54** | **90.34** | **96.49** | **87.18** | **80.68** |
| $MDP^2$ | 74.14 | 89.85 | 93.38 | 85.44 | 79.38 |

**Fig. 6.3:** Box plots of regret based on purity with additional red dots to indicate means. Methods are ordered with increasing mean value.

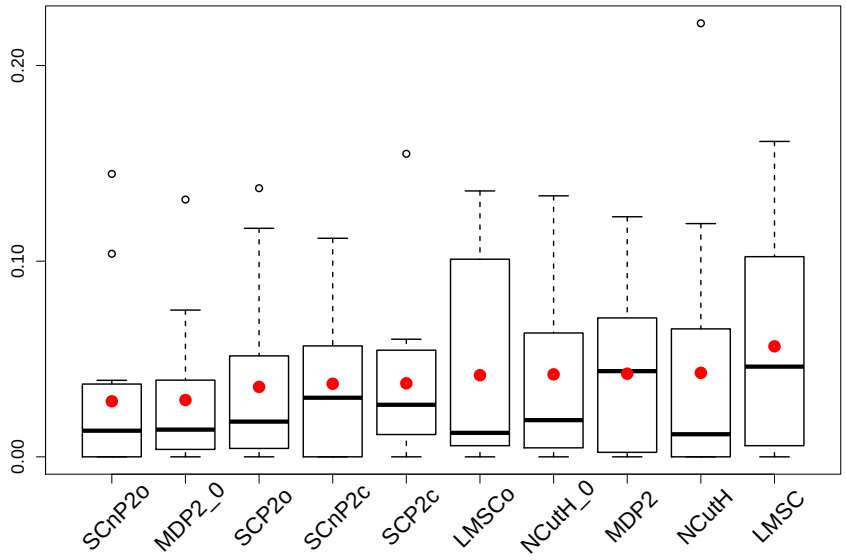
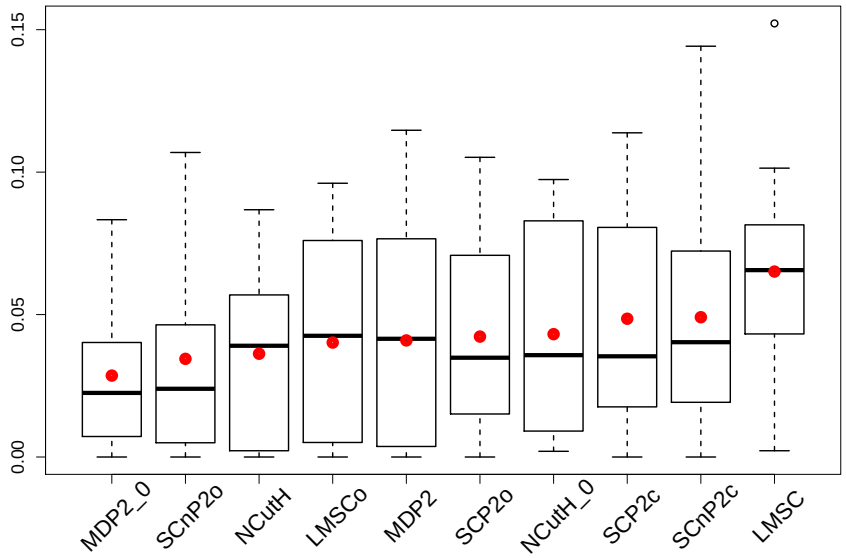



**Table 6.6:** A Comparison of the Proposed Methods for Semi-supervised Classification applied to Benchmark Data Sets taken from Chapelle et al. (2006b). Average Accuracy (%) over 12 Splits.

| | g241c | g241d | Digit1 | USPS | BCI |
|---|---|---|---|---|---|
| | 10 Labelled Examples | | | | |
| $S^3CP^2$ | 82.02 | **50.62** | **89.51** | **76.17** | **50.90** |
| $MDP^2$ | **85.17** | 49.95 | 87.96 | 72.65 | 50.88 |
| | 100 Labelled Examples | | | | |
| $S^3CP^2$ | **86.15** | **72.93** | **92.77** | **86.62** | **70.86** |
| $MDP^2$ | 85.09 | 53.74 | **92.77** | 84.42 | 66.50 |

**Fig. 6.4:** Box plots of regret based on V-measure with additional red dots to indicate means. Methods are ordered with increasing mean value.




# 3 Possible Extensions and Future Work

Clustering is one of the oldest problems in data analysis, and it is testament to the importance of the problem that it remains one of the most active areas of research. The variety and subjectivity of cluster definitions also indicate that it is likely to remain an important and unsolved problem in the future. In this section some ideas for future work on this problem that is related to the work presented in this thesis will be discussed briefly.

The problem of bi-partitioning formed the main focus of many of the contributions in this thesis. It is clear that a bi-partitioning method can be applied recursively to obtain multiple clusters, and this was performed explicitly in Chapters 4 and 5. In Chapter 5 determining the number of clusters automatically was also addressed by only splitting a cluster when there is strong statistical evidence that it in fact represents multiple clusters. It is arguably the case that this problem is easier to address when using non-optimal subspaces. Tasoulis et al. (2010) used a low density separation method where hyperplanes orthogonal to the principal component were used, and a cluster was only split if the estimated density contained multiple modes. While this does not offer statistical confidence in the splitting decision, it is a useful

heuristic.

The principal challenge when looking at optimal subspaces becomes apparent in the extreme case where the number of dimensions is greater than the number of data. If there are $n$ data in $d > n$ dimensions and the data are stored in a matrix $\mathcal{X} \in \mathbb{R}^{n \times d}$, then the projection equation given by $\mathcal{X}v = p$ has infinitely many solutions $v$ for any vector $p \in \mathbb{R}^n$, provided there are at least $n$ linearly independent dimensions in the data set. What this means is that regardless of the actual cluster structure in the data, there is a subspace in which they can be made to look like *any* equally weighted discrete distribution with $n$ atoms, up to a scaling constant. For example, for any bi-partition of the data set, each subset can be projected to a single point, making the data set in some sense maximally clusterable for *any* bi-partition.

So called $L_1$ regularisation has been used for problems like regression in high dimensional applications (Tibshirani, 1996). A regularisation of projection pursuit seems a natural way to overcome the above problem, but this adds to the difficulty in establishing when there is statistical confidence that a cluster should be split.

It would be very interesting to both establish robust rules for determining the presence of multiple clusters for the proposed methods, and also to investigate how regularisations affect the number of degrees of freedom in the resulting solution so that these can be usefully implemented in higher dimensional examples.

In recent years the high dimensionality problem has entered the realms of "Big Data". When the number of dimensions is so large that only a few data points can be stored in a computer's memory it is not possible to process the data in their original form. Random projections have become the go-to approach for dealing with these types of data. The popular result of Johnson and Lindenstrauss (1984) places a probabilistic bound on the error induced by random projections on the structure of the data. This result has been extended such that the error of a clustering model acting on the randomly projected data, rather than the original data set, can be evaluated (Boutsidis et al., 2010; Tasoulis et al., 2013).

An interesting observation made by a number of authors is that data arising from multiple clusters, when projected into a random subspace, tend to appear as a mixture of Gaussians (Dasgupta, 2000; Fern and Brodley, 2003; Tasoulis et al., 2013). Theoretical investigations into the properties of random projections have shown that under certain conditions, asymptotically as the number of dimensions tends to infinity, almost all projections of a data set are Gaussian (Noar and Romik, 2003; Dasgupta et al., 2006). What may be happening in the case of multiple clusters is that the convergence to Gaussianity within the clusters occurs at a faster rate than globally over the whole data set, due to their being somewhat closer to Gaussian to begin with. For

finitely many dimensions, but large enough to observe the convergence to Gaussianity within clusters, one therefore observes a mixture of Gaussians.

The size of the random subspace dictated by Johnson and Lindenstrauss (1984) is often still large enough that further dimension reduction is necessary to render meaningful results. Using the observation of Dasgupta et al. (2006) may offer an assumption that the data are close to a mixture of Gaussians. A projection pursuit based on optimally separating a Gaussian mixture should therefore offer a promising method for learning the optimal subspace in which to perform clustering.

As a final idea, the data stream setting is revisited. In the last section a remark was made about the infeasibility of finding optimal projections in the data stream setting. This does not, however, preclude an attempt to find locally optimal hyperplane separators. Returning to the assumption that high density clusters can be separated by low density regions, it may be possible to find locally minimum density hyperplanes in a fully incremental way.

Optimisation with sequential access to data is very often addressed using stochastic gradient descent (Bottou, 2010). When a noisy estimate of the gradient of the objective function can be observed with each arriving data point, then moving along the negative of these gradients for an appropriately sized sequence of steps leads to convergence to a local minimum in the objective. The bias in the gradient of a kernel based estimate of a probability density is independent of the number of data used in the estimate. The size of the bias is, in fact, controlled by the size of the bandwidth. Repeatedly updating a hyperplane by moving along the negative of the gradient of a kernel located on the most recently observed datum will decrease the density along the hyperplane. Some important caveats exist, however. Firstly, the density tends to zero in the tails of the distribution, and so if the hyperplane escapes the convex hull of the modes of the density then it will not converge. To avoid this problem, a simple bounding method can be used which prevents the hyperplane from deviating too far from the mean of the data. Alternatively, it may be possible to estimate the modes of the density in a similar manner to decreasing the density on a hyperplane, and thereby force the hyperplane to intersect their convex hull. Secondly, though the bias can be controlled by allowing the bandwidth to tend to zero, this has the effect of inflating the variance of the gradient estimates. This is a similar problem to that observed in the algorithm of Kiefer and Wolfowitz (1952). It may be possible to define a sequence of step sizes such that both the bias and the variance in the updates tend to zero asymptotically, in which case convergence can be achieved. Alternatively, a small but non-zero bandwidth can be used once sufficiently many data have been observed, which means that the hyperplane will not necessarily converge to a local minimum but the variance in the update equation is guaranteed to be well controlled.

# Bibliography


Ackerman, M. and Ben David, S. Clusterability: A Theoretical Study. In *Proceedings of AISTATS*, JMLR: W&CP, 5:1-8, 2009.

Aggarwal, C. C., Procopiuc, C. M., Wolf, J. L., Yu, P. S. and Park, J. S. Fast algorithms for projected clustering. In *Proceedings of the ACM International Conference on Management of Data*, 1999.

Aggarwal, C. C. and Yu, P. S. Finding generalized projected clusters in high dimensional space. In *Proceedings of the ACM International Conference on Management of Data*, 2000.

Aggarwal, C. C., Hinneburg, A., and Keim, D. On the surprising behavior of distance metrics in high dimensional space. *In Proceedings of the 8th International Conference on Database Theory*, 2001.

Aggarwal, C. C.: A survey of stream clustering algorithms. *Data Clustering: Algorithms and Applications*, Aggarwal C. C., Reddy C. (eds.), 457-482, 2013.

Aggarwal, C. C., and Reddy, C. K. *Data Clustering: Algorithms and Applications.* CRC Press, 2013.

Aggarwal, C. C., Han, J., Wang, J., Yu, P.: A framework for clustering evolving data streams. *Proceedings of the 29th international conference on Very large data bases*. 29, 81-92, 2003.

Aggarwal, C. C., Han, J., Wang, J., Yu, P. S.: A framework for projected clustering of high dimensional data streams. *Proceedings of the Thirtieth international conference on Very large data bases*. 852-863, 2004.

Agrawal, R., Gehrke, J., Gunopulos, D. and Raghavan, P. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM International Conference on Management of Data*, 1998.

E. Aitnouri, S. Wang, and D. Ziou. On comparison of clustering techniques for histogram pdf estimation. *Pattern Recognition and Image Analysis*, 10(2): 206–217, 2000.

Allgower, E. L. and Georg, K. *Numerical continuation methods: an introduction*, Volume 13. Springer Science & Business Media, 2012.

Amini, A., Saboohi, H., Wah, T.Y., Herawan, T.: Dmm-stream: A density mini-micro clustering algorithm for evolving datastreams. In *Proceedings of the First International Conference on Advanced Data and Information Engineering*, 675-682, 2014.

Amini, A., Wah, T.Y., Saboohi, H.: On density based data streams clustering algorithms: A survey. *Journal of Computer Science and Technology*, 29(1) 116-141, 2014.

Anagnostopoulos, C., Tasoulis, D. K., Adams, N. M., Pavlidis, N. G., Hand, D. J.: Online linear and quadratic discriminant analysis with adaptive forgetting for streaming classification. *Statistical Analysis and Data Mining*. 5(2), 139-166, 2012.

Ankerst, M., Breunig, M., Kriegel, H.P., Sander, J.: OPTICS: ordering points to identify the clustering structure. *Proceedings of the ACM Sigmod Conference*. 49-60, 1999.

Artac, M., Jogan, M., Leonardis, A.: Incremental PCA for on-line visual learning and recognition. *Proceedings of the 16th International Conference on Pattern Recognition*. 3, 781-784, 2002.

D. Arthur and S. Vassilvitskii. *k*-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, 2007.

Azzalini, A. and Torelli, N. Clustering via nonparametric density estimation. *Statistics and Computing*, 17(1):71–80, 2007.

Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 1-16, 2002.

Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/m.

Banfield, J. D. and Raftery, A. E. Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics*, 49, 803-821, 1993

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labelled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006

S. Ben-David, T. Lu, D. Pál, and M. Sotáková. Learning low-density separators. In D. van Dyk and M. Welling, editors, *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, JMLR Workshop and Conference Proceedings, pages 25–32, 2009.

Bennett, K. and Demiriz, A. Semi-supervised support vector machines. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 368–374. MIT Press, 1998.

Bengio, Y., Delalleau, O. and Le Roux, N. Label Propagation and Quadratic Criterion. In Chapelle, O., Schölkopf, B. and Zien, A. (eds) *Semi-Supervised Learning*, MIT Press, 2006

Berge, C. *Topological Spaces*. Macmillan, New York, 1963.

Beyer, K., Goldstein, J., Ramakrishnan, R. and Shaft, U. When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory*, 1999

Boley, D.: Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*. 2(4), 325-344, 1998

Bolton, R. J. and Krzanowski, W. J. *Projection pursuit clustering for exploratory data analysis.* Journal of Computational and Graphical Statistics, 12(1):121-142, 2003

Bonnans, J. F. and Shapiro, A. *Perturbation Analysis of Optimization Problems*. Springer Series in Operations Research. Springer, 2000.

Bottou, L. Large-scale machine learning with stochastic gradient descent. In *International Conference on Computational Statistics*, 177–187, 2010.

Boutsidis C., Zouzias A., Drineas P. Random projections for k-means clustering. *Advances in Neural Information Processing Systems*, 298-306, 2010.

Bradley, P. S. Mangasarian, O. L. and Street, W. N. Clustering via Concave Minimization. *Advances in Neural Information Processing Systems 9*, 368-374, MIT Press, 1997

Brandes, U., Gaertler, M., Wagner, D. Experiments on graph clustering algorithms. In *Proceedings of the 11th European Symposium on Algorithms*, LNCS 2832, Springer-Verlag, pp. 568–579, 2003

Broyden, C. G. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6 (1), 76-90, 1970

Bubeck, S., and von Luxburg, U. Nearest neighbor clustering: A baseline method for consistent clustering with arbitrary objective functions. *Journal of Machine Learning Research*, 10: 657–698, 2009

Burke, J. V., Lewis, A. S. and Overton, M. L. Approximating subdifferentials by random sampling of gradients. *Mathematics of Operations Research*, 27(3): 567–584, 2002.

Burke, J. V., Lewis, A. S. and Overton, M. L.. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2006.

Campello, R. J. G. B., Moulavi, D., Zimek, A., Sander, J.: A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. Data Mining and Knowledge Discovery 27(3): 344-371 (2013)

Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. *Proceedings of the 2006 SIAM International Conference on Data Mining*. 328-339, 2006

Carmichael, J. W., George, G. A., and Julius, R. S. Finding natural clusters. *Systematic Zoology*, 17:144–150, 1968.

V. Castelli and T. M. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16(1):105–111, 1995.

V. Castelli and T. M. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42(6):2102–2117, 1996.

Celeux, G. and Govaert, G. Gaussian Parsimonious Clustering Models. *Pattern Recognition*, 28, 781-793, 1995

Chapelle, O. and Zien, A. Semi-supervised classification by low density separation. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 57–64. Society for Artificial Intelligence and Statistics, 2005.

Chapelle, O., Chi, M. and Zien, A. A continuation method for semi-supervised SVMs. In *Proceedings of the 23rd International Conference on Machine Learning*, volume 148, pages 185–192. ACM Press, 2006a.

Chapelle, O., Schölkopf, B. and Zien, A. *Semi-Supervised Learning*. Adaptive computation and machine learning. MIT Press, 2006b.

Chapelle, O., Sindhwani, V. and Keerthi, S. S.. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9:203–233, 2008.

Chen, H., and Chau, M. Web mining: Machine learning for Web applications. *Annual review of information science and technology*, 38, 289-330, 2004

Chen, W. and Feng, G. Spectral clustering: A semi-supervised approach. *Neurocomputing*, 77(1), 229-242, 2012

Chen Y., Tu L.: Density-based clustering for real-time stream data. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 133-142 (2007)

Clarke, F. H. *Optimization and Nonsmooth Analysis*. Wiley-Interscience, New York, 1983.

Clarke, F. H., Yu. S. Ledyaev, Stern, R. J. and Wolenski, P. R. *Nonsmooth Analysis and Control Theory*. Springer-Verlag New York Inc., Secaucus, NJ, USA, 1998.

Collobert, R., Sinz. F., Weston, J. and Bottou, L. Large scale transductive SVMs. *Journal of Machine Learning Research*, 7:1687–1712, 2006.

Cuevas, A. and Fraiman, R. A plug-in approach to support estimation. *The Annals of Statistics*, 25(6):2300–2312, 1997.

Cuevas, A., Febrero, M. and Fraiman, R. Estimating the number of clusters. *Canadian Journal of Statistics*, 28(2):367–382, 2000.

Cuevas, A., Febrero, M. and Fraiman, R. Cluster analysis: a further approach based on density estimation. *Computational Statistics and Data Analysis*, 36 (4):441–459, 2001

F. E. Curtis and X. Que. An adaptive gradient sampling algorithm for nonsmooth optimization. *Optimization Methods and Software*, 28(6):1302–1324, 2013

Dasgupta, S. Experiments with random projection. In *Proceedings Uncertainty in Artificial Intelligence*. Morgan Kaufmann,143-151, 2000

Dasgupta, S., Hsu, D. J., and Verma, N. A Concentration Theorem for Projections. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 2006.

De Bie, T. and Cristianini, N. Convex methods for transduction. *Advances in Neural Information Processing Systems*, 16, 73–80, 2004

De Bie, T. and Cristianini, N. Semi-supervised learning using semi-definite programming. In O. Chapelle, B. Schoëlkopf, and A. Zien, editors, *Semi-supervised Learning*, pages 119–135. MIT Press, 2006.

Defays, D. An efficient algorithm for a complete link method. *The Computer Journal (British Computer Society)*, 20 (4): 364–366. doi:10.1093/comjnl/20.4.364, 1977

Demiriz, A. and Bennett, K. P. Optimization approaches to semi-supervised learning. *Applications and Algorithms of Complementarity*, 121–141. Kluwer, 2000

Dhillon, I., Guan, Y. and Kulis, B. Kernel k-Means, Spectral Clustering and Normalized Cuts. *Proceedings of the 10th ACM Knowledge Discovery and Data Mining Conference*, 551-556, 2004

C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 521–528, 2007

Eslava, G. and Marriott, F. H. C. *Some criteria for projection pursuit.* Statistics and Computing, 4(1):13-20, 1994

Ester, M., Kriegel, H-P., Sander, J., Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press. pp. 226–231. ISBN 1-57735-004-9. CiteSeerX: 10.1.1.71.1980, 1996

Fan, K. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences of the United States of America*, 35(11):652, 1949

Fern, X. Z. and Brodley, C. E. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Machine Learning, Proceedings of the International Conference on*, 2003.

Foggia, P., Percannella, G., Sansone, C., and Vento, M. Assessing the performance of a graph-based clustering algorithm, in *F. Escolano, M. Vento (Eds.), Lecture Notes in Computer Science*, vol. 4538, Springer-Verlag, Berlin. pp. 215– 227, 2007

Fletcher, R. A new approach to variable metric algorithms. *The computer journal*, 13 (3), 317-322, 1970

C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004

Fraley, C. and Raftery, A. E. Model-Based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association*, 97, 458, 611-631, 2002

Fränti, P. and Virmajoki, O. Iterative shrinking method for clustering problems. *Pattern Recognition*, 39(5):761–775, 2006.

Frigui, H., and Krishnapuram, R. A robust competitive clustering algorithm with applications in computer vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5), 450-465, 1999

Friedman, H. P. and Rubin, J. On Some Invariant Criteria for Grouping Data. *Journal of the American Statistical Association*, 62, 1159-1178, 1967

Friedman,J. H. and Tukey, J. W. A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Transactions on Computers* C–23 (9): 881–890, 1974

Goldfarb, D. A family of variable-metric methods derived by variational means. textitMathematics of computation, 24 (109), 23-26, 1970

Guattery, S. and Miller, G. On the Quality of Spectral Separators. *SIAM Journal of Matrix Analysis and Applications*, 19(3):701-719, 1998.

Guha, S., Meyerson, A., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering data streams: Theory and Practice. *IEEE Transactions on Knowledge and Data Engineering*. 15(3), 515-528,2003

Hagen, L. and Kahng, A. B. New spectral methods for ratio cut partitioning and clustering. *Computer-aided design of integrated circuits and systems, IEEE transactions on*, 11(9):1074–1085, 1992

Hartigan, J. A. *Clustering Algorithms*. Wiley Series in Probability and Mathematical Statistics. Wiley, New York, 1975.

Hartigan. J. A. and Hartigan, P. M. The dip test of unimodality. *The Annals of Statistics*, 13(1):70–84, 1985.

Hartigan, P. M.: Algorithm as 217: Computation of the dip statistic to test for unimodality. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*. 34(3), 320-325, 1985.

Hassani M., Spaus P., Gaber M. M., Seidl T.: Density-based projected clustering of data streams. Proceedings of the 6th International Conference on Scalable Uncertainty Management, 311-324 (2012)

Hassani, M., Kranen, P., Saini, R., Seidl, T.: Subspace anytime stream clustering. Proceedings of the 26th International Conference on Scientific and Statistical Database Management, 37 (2014)

Haykin, S. *Neural Networks: A comprehensive foundation*. Prentice-Hall International, 1999.

Hofmeyr, D., Pavlidis, N. and Eckley, I. Minimum Spectral Connectivity Projection Pursuit for Unsupervised Classification. *arXiv preprint, arXiv:1509.01546*, 2015

P. J. Huber. Projection pursuit. *The annals of Statistics*, pages 435–475, 1985

Hyvärinen, A. and Oja, E. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4):411–430, 2000.

Jain, A. K. and Dubes, R. C. *Algorithms for Clustering Data.* Prentice-Hall advanced reference series. Prentice-Hall, Inc., Upper Saddle River, NJ, 1988

Jain, A. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters,* 31(8):651– 666, 2010.

M. Ji, T. Yang, B. Lin, R. Jin, and J. Han. A simple algorithm for semi-supervised learning with improved generalization error bound. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1223–1230, 2012

Jia, C., Tan, C., Yong, A.: A Grid and Density-based Clustering Algorithm for Processing Data Stream. International Conference on Genetic and Evolutionary Computing (2008)

Joachims, T. Transductive inference for text classification using support vector machines. In *Proceedings of International Conference on Machine Learning (ICML)*, volume 99, pages 200–209, Bled, Slowenien, 1999.

Joachims, T. Transductive Support Vector Machines. In Chapelle, O., Schölkopf, B. and Zien, A. (eds) *Semi-Supervised Learning*, MIT Press, 2006

Johnson, W. B. and Lindenstrauss, J. *Extensions of Lipschitz mappings into a Hilbert space.* Contemporary mathematics, 26(189-206):1, 1984

Jones, M. C. and Sibson, R. What is projection pursuit? *Journal of the Royal Statistical Society, Series A*, 150, 1–36, 1987

Kiefer, J. and Wolfowitz, J. Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics*, 23 (3): 462, 1952.

Kiwiel, K. C. *Methods of Descent for Nondifferentiable Optimization*. Number 1133 in Lecture Notes in Mathematics. Springer–Verlag, Berlin, 1985.

Kranen, P., Assent, I., Baldauf, C., Seidl, T.: Self-Adaptive Anytime Stream Clustering. IEEE International Conference on Data Mining. 249-258 (2009). doi: 10.1109/ICDM.2009.47

Kranen, P.: Anytime algorithms for stream data mining. Diese Dissertation, RWTH Aachen University (2011)

Krause, A. and Liebscher, V. Multimodal projection pursuit using the dip statistic. Technical Report 13, Universität Greifswald, 2005.

Kriegel, H. P., Kröger, P., Zimek, A.: Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*. 3(1), 1-58, 2009

Kruskal, J. B. Toward a practical method which helps uncover the structure of a set of observations by finding the line transformation which optimizes a new "index of condensation". *Statistical computation*, 427–440, 1969

Kutin, S. Extensions to Mcdiarmids inequality when differences are bounded with high probability. Department Computer Science, University of Chicago, Chicago, IL. Technical report TR-2002-04, 2002

Leisch, F. A toolbox for k-centroids cluster analysis. *Computational Statististics and Data Analysis* 51, 526–544, 2006

Lewis, A. S. and Overton, M. L. *Eigenvalue optimization.* Acta numerica, 5:149-190, 1996

A. Lewis and M. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141:135–163, 2013

Li, Y. F., Tsang, I. W., Kwok, J. T. and Zhou, Z. H. Tighter and convex maximum margin clustering. In D. van Dyk and M. Welling, editors, *Proceedings of 12th International Conference on Artificial Intelligence and Statistics*, pages 344–351, 2009.

Li, Y., Xu, L-Q., Morphett, J., Jacobs, R.: An integrated algorithm of incremental and robust pca. *Proceedings of the International Conference on Image Processing*. 1, 245-248, 2009.

Lichman, M. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Lloyd, S. P. Least square quantization in PCM. Bell Telephone Laboratories Paper (1957) Published in journal much later: Lloyd., S. P. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2): 129–137. doi:10.1109/TIT.1982.1056489, 1982

Luan, J. Data Mining and Knowledge Management in Higher Education-Potential Applications. In *Proceedings of AIR Forum*, Toronto, Canada, 2002

Magnus, J. R. *On differentiating eigenvalues and eigenvectors.* Econometric Theory, 1(02):179-191, 1985

Mangasarian, O. L., Setiono, R., and Wolberg, W. H. *Pattern recognition via linear programming: Theory and application to medical diagnosis.* Large-scale numerical optimization, pages 22-31, 1990

McLachlan, G. J. and Basford, K. E. *Mixture Models: Inference and Applications to Clustering*, Marcel Dekker, 1988

B. McWilliams, D. Balduzzi, and J. M. Buhmann. Correlated random features for fast semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 440–448, 2013

Menardi, G. and Azzalini, A. An advancement in clustering via nonparametric density estimation. *Statistics and Computing*, 24(5):753–767, 2014.

Minnotte, M. C. and Scott, D. W. The mode tree: A tool for visualization of nonparametric density features. *Journal of Computational and Graphical Statistics*, 2(1):51-68, 1993

V. I. Morariu, B. V. Srinivasan, V. C. Raykar, R. Duraiswami, and L. S. Davis. Automatic online tuning for fast gaussian summation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1113–1120, 2008

Müller, D. W., Sawitzki, G.: Excess mass estimates and tests for multimodality. *Journal of the American Statistical Association*. 86(415), 738-746, 1991.

Murtagh, F. Complexities of Hierarchic Clustering Algorithms: the state of the art. *Computational Statistics Quarterly* 1: 101–113, 1984.

Murtagh, F. and Raftery, A. E. Fitting Straight Lines to Point Patterns. *Pattern Recognition*, 17, 479-483, 1984.

Narayanan, H., M. Belkin, and P. On the relation between low density separation, spectral clustering and graph cuts. In *Advances in Neural Information Processing Systems*, pp. 1025-1032, 2006

A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, volume 14, pages 849–856, 2002

Niu, D., Dy, J. G., and Jordan, M. I. Dimensionality reduction for spectral clustering. *In International Conference on Artificial Intelligence and Statistics*, pages 552-560, 2011

Naor, A. and Romik, D. Projecting the surface measure of the sphere of $\ell_p^n$. *Annals of the Institute Henri Poincarè, Probability and Statistics*, 39(2), 241–261, 2003.

Ntoutsi, I., Zimek, A., Palpanas, T., Kröger, P., Kriegel, H.P.: Density-based projected clustering over high dimensional data streams. Proceedings SiAM International Conference on Data Mining, 987–998 (2012)

Olson, E., Walter, M., Teller, S. J., and Leonard, J. J. Single-Cluster Spectral Graph Partitioning for Robotics Applications. In *Robotics: Science and Systems*. 265-272, 2005

Overton, M. L. and Womersley, R. S. *Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices.* Mathematical Programming, 62(1-3):321-357, 1993

Pavlidis, N. G.,Tasoulis, D. K., Adams, N. M., Hand, D. J.: $\lambda$-Perceptron: An adaptive classifier for data-streams. *Pattern Recognition*. 44(1), 78-96, 2011.

Pavlidis, N., Hofmeyr, D., and Tasoulis, S. *Minimum density hyperplane: An unsupervised and semi-supervised classifier.* arXiv preprint arXiv:1507.04201, 2015

Pham, D. T., and A. A. Afify. Clustering techniques and their applications in engineering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 221.11: 1445-1459, 2007

Phua, C., Lee, V., Smith, K., and Gayler, R. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*, 2010

Polak, E. On the mathematical foundations of nondifferentiable optimization in engineering design. *SIAM Review*, 29(1):21–89, March 1987.

Punj, G., and Stewart, D. W. Cluster analysis in marketing research: review and suggestions for application. *Journal of marketing research*, 134-148, 1983

Reynolds Jr, M. R., Stoumbos, Z. G. A CUSUM chart for monitoring a proportion when inspecting continuously. *Journal of Quality Technology*. 31(1), 1999

P. Rigollet. Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, 8:1369–1392, 2007

Rigollet, P. and Vert, R. Optimal rates for plug-in estimators of density level sets. *Bernoulli*, 15:1154–1178, 2009

Rinaldo, A. and Wasserman, L. Generalized density clustering. *The Annals of Statistics*, 38(5):2678–2722, 2010.

Rosenberg, S., Van Mechelen, I., and De Boeck, P. A hierarchical classes model: Theory and method with applications in psychology and psychopathology. *Clustering and classification*, 123-155, 1996

Rosenberg, N. A., Pritchard, J. K., Weber, J. L., Cann, H. M., Kidd, K. K., Zhivotovsky, L. A., and Feldman, M. W. Genetic structure of human populations. *Science*, 298(5602), 2381-2385, 2002

Rosenberg, A. and Hirschberg, J. V-measure: A conditional entropy-based external cluster evaluation measure. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, volume 7, pages 410–420, 2007.

Samaria, F. S. and Harter, A. C. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138-142. IEEE, 1994

Schuit, A. J., van Loon, A. J. M., Tijhuis, M., and Ockè, M. C. Clustering of lifestyle risk factors in a general adult population. *Preventive medicine*, 35(3), 219-224, 2002

Schur, J. Bemerkungen zur theorie der beschränkten bilinearformen mit unendlich vielen veränderlichen. *Journal für die reine und Angewandte Mathematik*, 140:1-28, 1911

Schwartz, G. E. Estimating the dimension of a model. *Annals of Statistics* 6 (2): 461–464, doi:10.1214/aos/1176344136, MR 468014, 1978

Scott, D. W.: Multivariate density estimation: theory, practice, and visualization. 383. John Wiley & Sons (2009)

Shanno, D. F. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24 (111), 647-656, 1970

Shi, J. and Malik, J. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 22(8):888–905, 2000

Sibson, R. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal (British Computer Society)* 16 (1): 30–34. doi:10.1093/comjnl/16.1.30, 1973

Silva, J. A., Faria, E. R., Barros, R. C., Hruschka, E. R., de Carvalho, A. C. P. L. F., Gama, J.: Data Stream Clustering: A Survey. *ACM Computing Surveys*. 46(1), 13:1-13:31,2013

Silverman, B. W. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 43(1): 97–99, 1981.

Silverman, B. W. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

Sindhwani, V., Belkin, M. and Niyogi, P. The Geometric Basis of Semi-Supervised Learning. In Chapelle, O., Schölkopf, B. and Zien, A. (eds) *Semi-Supervised Learning*, MIT Press, 2006

A. Singh, R. D. Nowak, and X. Zhu. Unlabeled data: Now it helps, now it doesn't. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1513–1520, 2009

Skajaa, A. Limited memory BFGS for nonsmooth optimization. *Master's thesis, Courant Institute of Mathematical Science*, New York University, 2010.

Sokal, R and Michener, C. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin* 38: 1409–1438, 1958

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. *Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization.* Molecular biology of the cell, 9(12):3273-3297, 1998

Steinbach, M., Karypis, G. and Kumar., V. A comparison of document clustering techniques. *Workshop on Text Mining, KDD*, 2000.

Steinbach, M., Ertöz, L. and Kumar, V. The challenges of clustering high dimensional data. In *New Directions in Statistical Physics*, 273–309. Springer, 2004

M. Stoer and F. Wagner, "A simple min-cut algorithm," *Journal of the ACM (JACM)*, vol. 44, no. 4, pp. 585–591, 1997

Stuetzle, W.: Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. Journal of Classification, 20(5): 25-47 (2003)

Stuetzle, W. and Nugent, R. A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, 19(2):397–418, 2010.

Sturn, A., J. Quackenbush, and Z. Trajanoski. Genesis: cluster analysis of microarray data. *Bioinformatics*, 18 (1), 207-208, 2002.

Tao, T. and Vu, V. *Random matrices have simple spectrum.* arXiv preprint arXiv:1412.1438, 2014

Tasoulis, S. K., Tasoulis, D. K., and Plagianakos, V. P. Enhancing principal direction divisive clustering. *Pattern Recognition*, 43(10):3391–3411, 2010.

Tasoulis, S. K., Tasoulis, D. K., Plagianakos, V. P.: Clustering of high dimensional data streams. *Artificial Intelligence: Theories and Applications*. 223-230, 2012.

Tasoulis, S. K., Tasoulis, D. K., and Plagianakos V. P. Random direction divisive clustering. *Pattern Recognition Letters*, 34(2) 131–139, 2013.

Tatiraju, S. and A. Mehta. Image segmentation using k-means clustering, em and normalized cuts. University Of California Irvine, 2008

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, Vol. 58, No. 1, pages 267-288, 1996.

Tong, S. and Koller, D. Restricted Bayes optimal classifiers. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, pages 658–664, Austin, Texas, August 2000.

Vapnik, V. and Sterin, A. On structural risk minimization or overall risk in a problem of pattern recognition. *Automation and Remote Control*, 10(3): 1495–1503, 1977.

Vapnik, V. N. and Kotz, S. *Estimation of dependences based on empirical data*, volume 40. Springer-verlag New York, 1982

Vapnik, V. *Statistical learning theory*, Volume 1. Wiley New York, 1998

Venkatasubramanian, S. and Wang, Q. *The Johnson-Lindenstrauss transform: An empirical study.* In ALENEX, pages 164-173. SIAM, 2011

Vergara, A., Vembu, S., Ayhan, T., Ryan, M. A., Homer, M. L., Huerta, R.: Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*. 166, 320-329, 2012.

von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007

von Luxburg, U., Belkin, M., and Bousquet, O. Consistency of spectral clustering. *Annals of Statistics*. 36(2), 555–586, 2008.

von Luxburg, U.: Clustering Stability. Now Publishers Inc (2010)

Wagner, D. and Wagner, F. *Between min cut and graph bisection*. Springer, 1993

Walther, G. Granulometric smoothing. *The Annals of Statistics*, 25(6):2273–2299, 1997.

Wang W., Yang J., Muntz R. STING: A Statistical Information Grid Approach to Spatial Data Mining. *Proceedings of the 23rd International Conference on Very Large Data Bases*, Morgan Kaufmann, pp. 186-195., 1997

Weng, J., Zhang, Y., Hwang, W-S.: Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 25(8), 1034-1040, 2003

Weyl, H. *Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung).* Mathematische Annalen, 71(4):441-479, 1912

P. Wolfe. On the convergence of gradient methods under constraint. *IBM Journal of Research and Development*, pages 407–411, 1972

Xu, L., Neufeld, J., Larson, B. and Schuurmans, D. Maximum margin clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 1537–1544, 2004.

Yan, D., Huang, L., and Jordan, M. I. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 907-916. ACM, 2009

Ye, Q. Relative perturbation bounds for eigenvalues of symmetric positive definite diagonally dominant matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(1):11-17, 2009

Zahn, C. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20:68–86, 1971.

Zedek, M. Continuity and location of zeros of linear combinations of polynomials. *Proceedings of the American Mathematical Society*, 16(1):78–84, 1965.

Zelnik-Manor, L. and Perona, P. *Self-tuning spectral clustering.* In Advances in neural information processing systems, pages 1601-1608, 2004

Zhang, T., Ramakrishnan, R., and Livny, M. *Birch: an efficient data clustering method for very large databases.* In ACM SIGMOD Record, volume 25, pages 103-114. ACM, 1996

Zhang, K., Tsang, I. W. and Kwok, J. T. Maximum margin clustering made practical. *IEEE Transactions on Neural Networks*, 20(4):583–596, 2009.

Zhao, B., Wang, F., and Zhang, C. Efficient multiclass maximum margin clustering. In *Proceedings of the 25th international conference on Machine learning*, pages 1248–1255. ACM, 2008.

Zhao, Y. and Karypis, G. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.

Zhao, Y. and Karypis, G. Criterion functions for document clustering: Experiments and analysis. Technical Report TR 01– 40, Department of Computer Science, University of Minnesota, Minneapolis, MN, 2001.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16, 321–328. MIT Press, 2004.

Zhu, X. P., Jin, M., Qian, W. Q., Liu, S., and Wei, Y. M. The application of unsupervised clustering in radar signal preselection based on DOA parameters. In *Pervasive Computing Signal Processing and Applications (PCSPA)*, 2010 First International Conference on. 956-959. IEEE, 2010

Zhu, X. and Ghahramani, Z. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, Pittsburgh, 2002