# Subspace Clustering with Active Learning

Hankui Peng
*STOR-i Centre for Doctoral Training*
*Lancaster University*
Lancaster, UK
h.peng3@lancaster.ac.uk

Nicos G. Pavlidis
*Department of Management Science*
*Lancaster University*
Lancaster, UK
n.pavlidis@lancaster.ac.uk

*Abstract*—Subspace clustering is a growing field of unsupervised learning that has gained much popularity in the computer vision community. Applications can be found in areas such as motion segmentation and face clustering. It assumes that data originate from a union of subspaces, and clusters the data depending on the corresponding subspace. In practice, it is reasonable to assume that a limited amount of labels can be obtained, potentially at a cost. Therefore, algorithms that can effectively and efficiently incorporate this information to improve the clustering model are desirable. In this paper, we propose an active learning framework for subspace clustering that sequentially queries informative points and updates the subspace model. The query stage of the proposed framework relies on results from the perturbation theory of principal component analysis, to identify influential and potentially misclassified points. A constrained subspace clustering algorithm is proposed that monotonically decreases the objective function subject to the constraints imposed by the labelled data. We show that our proposed framework is suitable for subspace clustering algorithms including iterative methods and spectral methods. Experiments on synthetic data sets, motion segmentation data sets, and Yale Faces data sets demonstrate the advantage of our proposed active strategy over state-of-the-art.

*Index Terms*—high dimensionality; active learning; subspace clustering; constrained clustering

## I. INTRODUCTION

In recent years crowdsourcing [1] for data annotation has drawn much attention in the computer vision community, due to the need to make use of as much data as possible and the lack of sufficient labelled data. Clustering is commonly used as an initial step to provide a coarse preliminary grouping in the absence of labelled data. For example, there are plant recognition apps that allow one to take a photo of a plant and identify its species. In video surveillance, one may wish to identify the points in a sequence of frames, be it people or cars etc. Usually some form of external information is available in these applications, either through crowdsourcing websites, or through paid manual work to conduct a limited amount of labelling. In either case, obtaining labels involves a cost which is either in time, money or both. Therefore, effective and efficient ways of carrying out data annotation are desirable.

The process of iteratively annotating the potentially misclassified data and subsequently updating the model is generally known as active learning [2]. It is a subfield of machine learning that aims to improve both supervised and unsupervised algorithms. In supervised learning, points that are near the decision boundary are likely to be misclassified. In unsupervised learning, the notion of potentially misclassified points is less clear and is open to interpretation.

In subspace clustering, points are clustered according to their underlying subspaces. There are different ways of measuring how likely a point is misclassified. One approach is to consider points whose projection onto the associated subspace is large as potentially misclassified [3]. Alternatively, points that are almost equidistant to their two nearest subspaces are likely to be misclassified [4]. Such ideas are based on the notion of *reconstruction error* between the original point and its projection to the subspace that defines the cluster. The total reconstruction error is the objective function of the $K$-Subspace Clustering (KSC) algorithm [5]. We therefore argue that effective active learning strategies should explicitly associate the query procedure with the optimisation of this objective. However as we will discuss in greater detail in the next section, the points with the largest reconstruction error are not necessarily the most informative from the perspective of updating the entire subspace clustering model.

Motivated by the connection between the reconstruction error and the KSC objective, we consider a point to be influential if querying its true class can lead to a large decrease in the total reconstruction error. Given a set of cluster assignments, the optimal linear subspace for each cluster can be trivially estimated through Principal Component Analysis (PCA) [6]. In particular, the basis for each subspace (cluster) can be defined through the set of eigenvectors of the covariance matrix of the points that are assigned to this cluster. We make use of ideas from perturbation analysis of PCA [7] to evaluate efficiently how influential each point is, and query the class of the most informative point(s). Once the true classes of the influential points have been identified, our proposed $K$-subspace clustering with constraints (KSCC) algorithm monotonically reduces the reconstruction error while satisfying all the constraints imposed by the labelled data. The active learning process iterates between these two query and update procedures until the query budget is exhausted.

The rest of the paper is organised as follows. We review related work in active learning in Section II, and introduce our proposed active framework in Section III. Experimental results on synthetic and real data are discussed in Section IV. The paper finishes in Section VI with conclusions and directions for future work.

## II. RELATED WORK

There are three main approaches to active learning [2]: uncertainty sampling [8], query by committee [9], and expected model change [10].

*Uncertainty sampling* queries the points the learning algorithm is least confident about. Classic uncertainty sampling methods are generally ignorant to the data distribution, thus prone to select outliers [12]. It is suggested in [13] to measure the informativeness of each point by the probability margin between the label it is assigned to and its second most likely label. Other versions of uncertainty sampling have been proposed to balance the density of a region and the uncertainty in that region [14]. When building supervised models, one may also choose the unlabelled points near the decision boundary.

*Query by committee (QBC)* is a type of active learning strategy designed for classifier ensembles [9]. It constructs a committee of models based on the labelled training data, and chooses to query the unlabelled points upon which the predictions of the classifiers in the ensemble disagree the most. It enables the training of accurate classifiers using a small subset of the data. To use this strategy, one has to provide both the type of classifier and a measure of disagreement among the classifiers. It has been shown in [15] that rapid decrease in the misclassification error is guaranteed if the queries have high expected information gain.

*Expected model change* is an active learning framework that bases its query strategy on the idea that a point is informative if knowing its true class can cause a big change in the current model [10]. This is mostly applied to discriminative probabilistic modelling, in which the gradient of the model is used as an indicator for the informativeness of a point. It is widely applied to image retrieval and text classification [11]. The method we propose also adopts this approach, but we are the first to consider updating an unsupervised learning model describing all the data rather than just the labelled data.

As is implied above, most active learning approaches have been developed for supervised learning. However less attention has been paid to the unsupervised counterpart. To the best of our knowledge, only a few active learning strategies have been proposed for subspace clustering [3], [4]. In [3], two active strategies *MaxResid* and *MinMargin* for KSC are proposed. *MaxResid* queries points that have large reconstruction error to their allocated subspaces. *MinMargin* queries points that are maximally equidistant to their two closest subspaces. These two strategies are effective in identifying the points that are most likely to be misclassified. However, these points are not necessarily the most informative points in terms of updating the full clustering model.

## III. ACTIVE LEARNING FRAMEWORK

In this section, we first formulate the subspace clustering problem and then present the proposed active learning framework. There are two iterative procedures within this framework. The first is to identify the most influential and potentially misclassified points. The second is to update the cluster assignment for all the data given the labelled points.

### A. K-Subspace Clustering

A $q$-dimensional linear subspace $\mathcal{S}_k$, $k \in \{1, \ldots, K\}$, can be defined through an orthonormal matrix $V_k \in \mathbb{R}^{P \times q}$ as

$$\mathcal{S}_k = \left\{ \mathbf{x} \in \mathbb{R}^P : \mathbf{x} = V_k \mathbf{y} \right\}, \qquad (1)$$

where the columns of $V_k$ constitute a basis for $\mathcal{S}_k$.

In subspace clustering, the overall objective is to find the set of optimal cluster assignments for all data points such that the total reconstruction error between each data point to their corresponding subspaces is minimised. The loss function value $L(\mathbf{x}_i, V_{k_i})$ and the objective $f(\mathcal{X}, \mathcal{V})$ can be written as

$$L(\mathbf{x}_i, V_{k_i}) = \|\mathbf{x}_i - V_{k_i} V_{k_i}^\top \mathbf{x}_i\|_2^2, \text{ and} \qquad (2)$$

$$f(\mathcal{X}, \mathcal{V}) = \min_{V_1, \ldots, V_K} \sum_{i=1}^N \min_{k_i \in \{1, \ldots, K\}} L(\mathbf{x}_i, V_{k_i}), \qquad (3)$$

where $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ contains the set of all $N$ points, and $\mathcal{V} = \{V_1, \ldots, V_K\}$ represents the set of all subspace bases. This objective can be minimised through a $K$-means-like iterative algorithm by alternating between *subspace estimation* and *cluster assignment* [5].

Given a set of cluster assignments $\Omega = \{k_1, \ldots, k_N\}$, we need to obtain the set of subspace bases such that the total reconstruction error in Eq. (3) is minimised. The basis matrix $V_k$ for each subspace $k$ can be obtained through the eigen-decomposition of its covariance matrix as

$$(X_k - \boldsymbol{\mu}_k)^T (X_k - \boldsymbol{\mu}_k) = V_k^* \Lambda_k^* (V_k^*)^T. \qquad (4)$$

We denote $X_k \in \mathbb{R}^{n_k \times P}$ as the data matrix that contains the $n_k$ points assigned to cluster $k$, and $\boldsymbol{\mu}_k$ as the column-wise mean vector of $X_k$. $V_k^*$ is a $P \times P$ matrix whose columns correspond to the eigenvectors of the covariance matrix of $X_k$, and $\Lambda_k^*$ is a diagonal matrix containing the $P$ eigenvalues. We denote $V_k$ as the subset of eigenvectors in $V_k^*$ that corresponds to the $q$ largest eigenvalues.

Given the subspace bases $\mathcal{V} = \{V_1, \ldots, V_K\}$, the cluster assignment $k_i$ for each point $\mathbf{x}_i \in \mathcal{X}$ can be obtained as

$$k_i = \arg \min_{k \in \{1, \ldots, K\}} \left\| \mathbf{x}_i - V_k V_k^T \mathbf{x}_i \right\|_2^2. \qquad (5)$$

The algorithm terminates when the loss function value in Eq. (3) stops decreasing, which means either a local or global optimum is reached.

### B. Query Procedure

The first element in quantifying the influence of an unlabelled point is the reduction in the reconstruction error that would be achieved if this point is removed from its currently assigned cluster. It is important to note that removing a point from a cluster implies that the basis for the associated linear subspace can change, because $V_k$ is a function of $X_k$ (see Eq. (4)). Explicitly, we define $U_1(\mathbf{x}_s, V_{k_s})$ as the decrease in the reconstruction error after removing the queried point $\mathbf{x}_s$ from cluster $k_s$ as follows

$$U_1(\mathbf{x}_s, V_{k_s}) = \sum_{\mathbf{x} \in \mathcal{X}_{k_s}} L(\mathbf{x}, V_{k_s}) - \sum_{\mathbf{x} \in \mathcal{X}_{k_s} \setminus \{\mathbf{x}_s\}} L(\mathbf{x}, \tilde{V}_{k_s}), \quad (6)$$

where $\mathcal{X}_{k_s}$ denotes the set of points in cluster $k_s$, and $V_{k_s}$ denotes the basis for cluster $k_s$. We use $\tilde{V}_{k_s}$ to denote the potentially perturbed basis after point $\mathbf{x}_s$ is removed.

The second element in quantifying the influence of an unlabelled point is to consider the increase in the reconstruction error of the cluster that $\mathbf{x}_s$ will be assigned to (after being removed from its current cluster $k_s$). As before, adding a point to a cluster implies that the associated basis for this cluster can change. Given that points are allocated to their closest subspace, it is thus sensible to assume the cluster that $\mathbf{x}_s$ has the second smallest reconstruction error to is where $\mathbf{x}_s$ would be assigned next. This can be expressed as

$$k_s^* = \arg \min_{k\in\{1,\ldots,K\}\setminus\{k_s\}} L(\mathbf{x}_s, V_k). \tag{7}$$

Then we can define $U_2(\mathbf{x}_s, V_{k_s^*})$ as the increase in the reconstruction error after adding $\mathbf{x}_s$ to cluster $k_s^*$, which can be expressed as

$$U_2(\mathbf{x}_s, V_{k_s^*}) = \sum_{\mathbf{x}\in\mathcal{X}_{k_s^*}\cup\{\mathbf{x}_s\}} L(\mathbf{x}, \tilde{V}_{k_s^*}) - \sum_{\mathbf{x}\in\mathcal{X}_{k_s^*}} L(\mathbf{x}, V_{k_s^*}). \tag{8}$$

Here $\tilde{V}_{k_s^*}$ is the $P \times q$ basis matrix whose columns are the eigenvectors of the covariance matrix of the points in the set $\{\mathcal{X}_{k_s^*} \cup \{\mathbf{x}_s\}\}$.

Combining the above two measures of influence together, we determine the most informative and likely to be misclassified point $\mathbf{x}_s^*$ as

$$\mathbf{x}_s^* = \arg \max_{\mathbf{x}_s\in\mathcal{X}_U} \left\{ U_1(\mathbf{x}_s, V_{k_s}) - U_2(\mathbf{x}_s, V_{k_s^*}) \right\}, \tag{9}$$

where $\mathcal{X}_U$ denotes the set of unlabelled points, and $\mathcal{X}_L$ the set of labelled points. Eq. (9) gives the point that brings the largest decrease in the reconstruction error once removed from its allocated cluster $k_s$, and the smallest increase in reconstruction error upon being reallocated to its most probable cluster $k_s^*$.

Although these two measures of influence can be quantified and calculated exactly, the number of required SVD computations is $\mathcal{O}(N^2)$ throughout all iterations. Not to mention that the computational complexity of SVD is $\min\{N^2 P, P^2 N\}$ [20]. Every time a point is removed from or added to a cluster, the subspace bases change and need to be recalculated through PCA. Hence the need to seek for an alternative approach, which could be pursued through the perturbation analysis of PCA [7].

We approximate the perturbed covariance matrix, the perturbed eigenvectors and eigenvalues through power series expansions [16]. As such, we can obtain expressions for the updated reconstruction error without having to recompute all the updated eigenvalues and eigenvectors after data deletion or addition. The algorithmic form of the query strategy is provided in Algorithm 1 before we detail the specifics of how the two influence measures are calculated.

**The influence of data deletion.** Let $S$ denote the sample covariance matrix for the points that belong to the same cluster, $\lambda_1 \ldots, \lambda_P$ denote its eigenvalues in descending order, and $\mathbf{v}_1, \ldots, \mathbf{v}_P$ denote its eigenvectors. Let $I$ denote a set

---

**Algorithm 1:** Query Strategy

**Input :** Data matrix $X \in \mathbb{R}^{N\times P}$
Number of clusters $K$
Initial cluster assignment $\{k_1, \ldots, k_N\}$

**repeat**
  **for** $\boldsymbol{x}_s \in \mathcal{X}_U$ **do**
    Compute the influence $U_1(\mathbf{x}_s, V_{k_s})$ of removing
    $\mathbf{x}_s$ from its allocated cluster $k_s$
    Calculate
    $k_s^* = \arg\min_{k\in\{1,\ldots,K\}\setminus\{k_s\}} L(\mathbf{x}_s, V_k)$
    Calculate $U_2(\mathbf{x}_s, V_{k_s^*})$ using Eq. (8)
  **end**
  Optimise Eq. (9) to query $\mathbf{x}_s^*$ and its true class
  $l_s \in \{1, \ldots, K\}$
**until** *Budget $T$ or desired performance is reached*

---

of $l$ points to be removed from the cluster. As a result, the covariance matrix, its eigenvectors and eigenvalues will be perturbed by a certain amount. Under small perturbations $(0 < \epsilon < 1)$, the perturbed covariance matrix $S(\epsilon)$, the $k$th perturbed eigenvalue $\lambda_k(\epsilon)$ and the $k$th perturbed eigenvector $\mathbf{v}_k(\epsilon)$ can be written as the following convergent power series

$$
\begin{aligned}
S(\epsilon) &= S + S^{(1)}\epsilon + S^{(2)}\epsilon^2 + \cdots + S^{(m)}\epsilon^m + \cdots, \\
\lambda_k(\epsilon) &= \lambda_k + \alpha_1\epsilon + \alpha_2\epsilon^2 + \cdots + \alpha_m\epsilon^m + \cdots, \\
\mathbf{v}_k(\epsilon) &= \mathbf{v}_k + \boldsymbol{\psi}_1\epsilon + \boldsymbol{\psi}_2\epsilon^2 + \cdots + \boldsymbol{\psi}_m\epsilon^m + \cdots.
\end{aligned}
\tag{10}
$$

For sufficiently small $\epsilon$, the order of the eigenvalues is maintained, so are the signs of the eigenvectors [17].

The main interest lies in finding the coefficients in the power series approximations. First the perturbed sample covariance matrix $S_{(I)}^-$ can be deduced from basic definitions of the covariance matrix [18], [19],

$$
\begin{aligned}
S_{(I)}^- = S &+ \frac{l}{n-l}\left[(S - S_I) - (\bar{\mathbf{x}}_I - \bar{\mathbf{x}})(\bar{\mathbf{x}}_I - \bar{\mathbf{x}})^T\right] \\
&- \frac{l^2}{(n-l)^2}(\bar{\mathbf{x}}_I - \bar{\mathbf{x}})(\bar{\mathbf{x}}_I - \bar{\mathbf{x}})^T.
\end{aligned}
\tag{11}
$$

In the above expression, $n$ denotes the original number of points in the cluster. We use $\bar{\mathbf{x}} \in \mathbb{R}^P$ to denote the feature-wise mean vector of the data before the removal of $l$ points, and $\bar{\mathbf{x}}_I$ the feature-wise mean vector of the $l$ points to be removed. Lastly, we use $S$, $S_I$, and $S_{(I)}^-$ to denote the original covariance matrix, the covariance matrix of the deleted data, and the covariance matrix of the perturbed data, respectively. We can associate $\frac{l}{n-l}$ and $\left((S - S_I) - (\bar{\mathbf{x}}_I - \bar{\mathbf{x}})(\bar{\mathbf{x}}_I - \bar{\mathbf{x}})^T\right)$ with $\epsilon$ and $S^{(1)}$ as in Eq. (10). Similarly, the correspondence can be made for the second order coefficients. We use a first order approximation for our purpose from now on, as it has been shown to be sufficiently accurate [18].

As for the coefficients in the approximations for the eigenvalues and eigenvectors, Lemma 2 in [18] provides us with the following results

$$\alpha_1 = \mathbf{v}_k^T S^{(1)} \mathbf{v}_k, \;\; \alpha_m = \mathbf{v}_k^T S^{(1)} \boldsymbol{\psi}_{m-1}; \tag{12}$$

and

$$\boldsymbol{\psi}_1 = -(S - \lambda_k \mathbf{I})^+ S^{(1)} \mathbf{v}_k,$$

$$\boldsymbol{\psi}_m = -(S - \lambda_k \mathbf{I})^+ \left( S^{(1)} \boldsymbol{\psi}_{m-1} - \sum_{i=1}^{m-1} \alpha_i \boldsymbol{\psi}_{m-i} \right). \quad (13)$$

In the above expression, we have the Moore-Penrose inverse $(S - \lambda_k I)^+ = \sum_{j \neq k} \frac{\mathbf{v}_j \mathbf{v}_j^T}{(\lambda_j - \lambda_k)}$ [20]. Based on the above, we can deduce expressions for the perturbed eigenvalues, and the influence of data deletion as expressed in Eq. (6).

We start with writing the first order approximation of the $k$th ($k \in \{1, \dots, P\}$) perturbed eigenvalue as follows

$$\begin{aligned}
\lambda_k(\epsilon) &= \lambda_k + \epsilon \lambda_k^{(1)} + \mathcal{O}(\epsilon^2) \\
&\approx \lambda_k + \epsilon \mathbf{v}_k^T S^{(1)} \mathbf{v}_k \\
&= \lambda_k + \frac{l}{n-l} \mathbf{v}_k^T \left[ S - \frac{1}{l} \sum_{i \in I} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \right] \mathbf{v}_k \\
&= \lambda_k + \frac{l}{n-l} \left[ \lambda_k - \frac{1}{l} \sum_{i \in I} \alpha_{ki}^2 \right] \\
&= \frac{n}{n-l} \lambda_k - \frac{1}{n-l} \sum_{i \in I} \alpha_{ki}^2,
\end{aligned}$$
$$(14)$$

where $\alpha_{ki} = \mathbf{v}_k^T (\mathbf{x}_i - \bar{\mathbf{x}})$. Then using the expression in Eq. (6), we can write the influence of removing a set $I$ of data $X_I \in \mathbb{R}^{l \times P}$ from cluster $k$ as

$$\begin{aligned}
U_1(X_I, k) &= \sum_{\mathbf{x} \in \mathcal{X}_k} L(\mathbf{x}, V_k) - \sum_{\mathbf{x} \in \mathcal{X}_k \backslash \{\mathbf{x}_i : i \in I\}} L(\mathbf{x}, \tilde{V}_k) \\
&= \sum_{k=q+1}^{P} \lambda_k - \sum_{k=q+1}^{P} \lambda_k(\epsilon) \\
&= \sum_{k=q+1}^{P} \left( \frac{1}{n-l} \sum_{i \in I} \alpha_{ki}^2 - \frac{l}{n-l} \lambda_k \right).
\end{aligned}$$
$$(15)$$

One can obtain the influence for the deletion of one point by plugging in $l = 1$. The deduction follows due to the equivalence between the reconstruction error and the sum of the unused eigenvalues in representing the subspace [6].

**The influence of data addition.** In the previous section, we have shown the influence of data deletion through perturbation analysis of the eigenvalues and eigenvectors. The aim is to find influential points whose true classes might differ from their currently allocated labels. Now we assess the impact on the reconstruction error for the cluster to which the removed points are added. Following the same line of analysis as before, we now let $X$ denote the data matrix that the set of $l$ points are to be added to, and $n$ the number of points in $X$. We denote the data after the addition of $l$ points as $X_{I_+}$, and the corresponding sample covariance matrix $S_{(I)}^+$ which combines the original data $X$ and the data to be added $X_I$.

**Proposition 1.** *The form of $S_{(I)}^+$ can be expressed as follows,*

$$\begin{aligned}
S_{(I)}^+ = S &+ \frac{l}{n+l} \left[ (S_I - S) - (\bar{\boldsymbol{x}}_I + \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_I + \bar{\boldsymbol{x}})^T \right] \\
&+ \frac{l^2}{(n+l)^2} (\bar{\boldsymbol{x}}_I + \bar{\boldsymbol{x}}) (\bar{\boldsymbol{x}}_I + \bar{\boldsymbol{x}})^T.
\end{aligned}$$
$$(16)$$

The proof of Proposition 1 can be found in the Appendix. It is easy to see that this can be matched exactly with the first two orders of the power series expansion. Interestingly, the form of the perturbed covariance matrix in the case of single data addition cannot be obtained trivially by setting $l = 1$ in Eq. (16). We show the perturbed form of the covariance matrix for the case of single data addition in Proposition 2, with the proof included in the Appendix.

**Proposition 2.** *The perturbed covariance matrix in the case when $l = 1$ can be expressed as*

$$\begin{aligned}
S_{(i)}^+ = S &+ \frac{1}{n+1} \left[ (\bar{\boldsymbol{x}} - \boldsymbol{x}_i) (\bar{\boldsymbol{x}} - \boldsymbol{x}_i)^T - S \right] \\
&- \frac{1}{(n+1)^2} (\bar{\boldsymbol{x}} - \boldsymbol{x}_i) (\bar{\boldsymbol{x}} - \boldsymbol{x}_i)^T,
\end{aligned}$$
$$(17)$$

*in which $\frac{1}{n+1}$ and $\left[ (\bar{\boldsymbol{x}} - \boldsymbol{x}_i) (\bar{\boldsymbol{x}} - \boldsymbol{x}_i)^T - S \right]$ correspond to $\epsilon$ and $S^{(1)}$ respectively.*

Using the above expression for the perturbed covariance matrix and the results in Eq. (12), we express the first order approximation of the $k$th perturbed eigenvalue for $l = 1$ as

$$\begin{aligned}
\lambda_k(\epsilon) &= \lambda_k + \epsilon \lambda_k^{(1)} + \mathcal{O}(\epsilon^2) \\
&\approx \lambda_k + \epsilon \mathbf{v}_k^T S^{(1)} \mathbf{v}_k \\
&= \lambda_k + \frac{1}{n+1} \mathbf{v}_k^T \left( (\bar{\mathbf{x}} - \mathbf{x}_i) (\bar{\mathbf{x}} - \mathbf{x}_i)^T - S \right) \mathbf{v}_k \\
&= \frac{1}{n+1} \alpha_{ki}^2 + \frac{n}{n+1} \lambda_k,
\end{aligned}$$
$$(18)$$

where $\alpha_{ki} = \mathbf{v}_k^T (\mathbf{x}_i - \bar{\mathbf{x}})$ as before. Hence, the change in the reconstruction error for cluster $k_s^*$ after the addition of $\mathbf{x}_s$ can be expressed as

$$\begin{aligned}
U_2(\mathbf{x}_s, V_{k_s^*}) &= \sum_{\mathbf{x} \in \mathcal{X}_{k_s^*} \cup \{\mathbf{x}_s\}} L(\mathbf{x}, \tilde{V}_{k_s^*}) - \sum_{\mathbf{x} \in \mathcal{X}_{k_s^*}} L(\mathbf{x}, V_{k_s^*}) \\
&= \sum_{k=q+1}^{P} (\lambda_k(\epsilon) - \lambda_k) \\
&= \sum_{k=q+1}^{P} \frac{\alpha_{ki}^2 - \lambda_k}{n+1}.
\end{aligned}$$
$$(19)$$

Using the perturbation analysis results, the influence of data addition and deletion can be calculated directly after $K$ SVD computations per iteration. This means we only need $(T \cdot K)$ SVD computations for all $T$ iterations as compared to $\mathcal{O}(T \cdot N^2)$.

## C. Update Procedure

After the class memberships of some points are queried, we will know the pairwise must-link and cannot-link relationships among them. However, we do not know to which cluster labels we should assign each of these points to. The next step is to update the subspace model under the grouping constraints. That is, the queried points that belong to the same class must be assigned to the same cluster label. Additionally, the queried points that do not belong to the same class should be assigned different cluster labels.

We can naturally extend KSC into an iterative constrained clustering algorithm with three stages. The first two stages involve the estimation of subspace bases and the cluster assignment of each point to the closest subspace. In the third stage, we satisfy the grouping constraints as mentioned above. This gives us a new constrained clustering objective, which we can divide into two parts.

For the set of unlabelled data $\mathcal{X}_U$, the subspace clustering objective is to minimise

$$L(\mathcal{X}_U, \mathcal{V}) = \sum_{\mathbf{x}_u \in \mathcal{X}_U} \left\{ \min_{m \in \{1,...,K\}} \left\| \mathbf{x}_u - V_m V_m^T \mathbf{x}_u \right\|_2^2 \right\}, \quad (20)$$

where $V_m$ is a $P \times q$ basis matrix that is determined by the points that are currently allocated to subspace $m$. Note that the basis matrix of the $m$th cluster $V_m$ is determined by points that are both labelled and unlabelled.

For the set of labelled data $\mathcal{X}_L$, we need to both minimise the reconstruction error without violating any of the grouping constraints. Among $K$ groups of queried points, there are $K!$ ways of matching each group to a unique cluster label. This is a combinatorial optimisation problem, and we can denote as $\mathcal{P}(K)$ the set of all possible permutations. Let $P_n$ be one realisation of $\mathcal{P}(K)$ that contains $K$ unique assignment labels to be matched with the queried points, and $P_{nl}$ be the assigned cluster label in the $n$th permutation that corresponds to true class $l$. Then we can write the subspace clustering objective for the labelled data $\mathcal{X}_L$ as

$$L(\mathcal{X}_L, \mathcal{V}) = \min_{\substack{P_n \in \mathcal{P}(K), \\ n \in \{1,...,K!\}}} \left\{ \sum_{l=1}^{K} \left\| X_l - V_{P_{nl}} V_{P_{nl}}^T X_l \right\|_2^2 \right\}, \quad (21)$$

where $X_l$ is a $n_l \times P$ matrix that contains the $n_l$ queried points from class $l$.

When $K$ is small, it is easy to simply evaluate all $K!$ permutations and choose the one with the smallest overall cost. However as the number of clusters grows, it is computationally prohibitive to evaluate all combinatorial possibilities. It is also known as the *minimum weight perfect matching problem*, which can be solved in polynomial time through the Hungarian algorithm [21]. We first construct a $K$ by $K$ cost matrix $P$ in which the $(n, l)$-th entry $P_{nl}$ denotes the total reconstruction error of allocating data from class $n$ to cluster label $l$. In this algorithm, the computational cost is upper bounded by $\mathcal{O}(K^3)$. We adopt the Hungarian algorithm as an alternative approach to exhaustive search to our problem in stage 3 when $K!$ is larger than $K^3$.

To combine both the unlabelled and labelled objectives together, we can write a constrained objective as

$$g(\mathcal{X}, \mathcal{V}) = \sum_{\mathbf{x}_u \in \mathcal{X}_U} \left\{ \min_{m \in \{1,...,K\}} \left\| \mathbf{x}_u - V_m V_m^T \mathbf{x}_u \right\|_2^2 \right\}$$
$$+ \min_{\substack{P_n \in \mathcal{P}(K), \\ n \in \{1,...,K!\}}} \left\{ \sum_{l=1}^{K} \left\| X_l - V_{P_{nl}} V_{P_{nl}}^T X_l \right\|_2^2 \right\}. \quad (22)$$

The procedural form of *KSC with Constraints (KSCC)* is detailed in Algorithm 2. This three-stage procedure ensures that the constrained subspace clustering objective decreases monotonically while satisfying all grouping constraints. A detailed proof for this can be found in Theorem 1.

**Theorem 1.** *The $K$-subspace clustering with constraints (KSCC) algorithm decreases the objective in Eq.* (22) *monotonically throughout iterations.*

*Proof.* Our proof borrows ideas from the proof for the monotonicity of the $K$-means algorithm. In the first iteration, we have as input a set of cluster assignment $\Omega^{(1)} = \left\{ k_1^{(1)}, \ldots, k_N^{(1)} \right\}$, the set of unlabelled data $\mathcal{X}_U$ and labelled data $\mathcal{X}_L$. In the first iteration, we can calculate a set of bases matrices $\mathcal{V}^{(1)}$ for all subspaces. Let $g(\mathcal{X}, \mathcal{V}^{(1)})$ be the combined reconstruction error at iteration 1, then at iteration $t$ ($t = 1, \ldots, T$) we have

$$g(\mathcal{X}, \mathcal{V}^{(t)}) = \sum_{\mathbf{x}_u \in \mathcal{X}_U} \left\| \mathbf{x}_u - V_{k_u^{(t)}}^{(t)} [V_{k_u^{(t)}}^{(t)}]^T \mathbf{x}_u \right\|_2^2$$
$$+ \sum_{l=1}^{K} \left\| X_l - V_{P_{nl}^{(t)}}^{(t)} [V_{P_{nl}^{(t)}}^{(t)}]^T X_l \right\|_2^2$$
$$\geq \sum_{\mathbf{x}_i \in \mathcal{X}} \left\| \mathbf{x}_i - V_{k_i^{(t)}}^{(t+1)} [V_{k_i^{(t)}}^{(t+1)}]^T \mathbf{x}_i \right\|_2^2$$
$$= \sum_{\mathbf{x}_u \in \mathcal{X}_U} \left\| \mathbf{x}_u - V_{k_u^{(t)}}^{(t+1)} [V_{k_u^{(t)}}^{(t+1)}]^T \mathbf{x}_u \right\|_2^2$$
$$+ \sum_{l=1}^{K} \left\| X_l - V_{P_{nl}^{(t)}}^{(t+1)} [V_{P_{nl}^{(t)}}^{(t+1)}]^T X_l \right\|_2^2 \quad (23)$$
$$\geq \sum_{\mathbf{x}_u \in \mathcal{X}_U} \left\| \mathbf{x}_u - V_{k_u^{(t+1)}}^{(t+1)} [V_{k_u^{(t+1)}}^{(t+1)}]^T \mathbf{x}_u \right\|_2^2$$
$$+ \sum_{l=1}^{K} \left\| X_l - V_{P_{nl}^{(t)}}^{(t+1)} [V_{P_{nl}^{(t)}}^{(t+1)}]^T X_l \right\|_2^2$$
$$= \sum_{\mathbf{x}_u \in \mathcal{X}_U} \left\| \mathbf{x}_u - V_{k_u^{(t+1)}}^{(t+1)} [V_{k_u^{(t+1)}}^{(t+1)}]^T \mathbf{x}_u \right\|_2^2$$
$$+ \sum_{l=1}^{K} \left\| X_l - V_{P_{nl}^{(t+1)}}^{(t+1)} [V_{P_{nl}^{(t+1)}}^{(t+1)}]^T X_l \right\|_2^2$$
$$= g(\mathcal{X}, \mathcal{V}^{(t+1)}).$$
$$\square$$

The first line of the proof says that, at iteration $t$ we have a set of cluster assignments for the unlabelled data $\Omega_U^{(t)}$ and

for the labelled data $\Omega_L^{(t)}$ that satisfies all constraints imposed upon knowing the true classes of the points in $\mathcal{X}_L$. When we proceed into the next step of updating the set of bases $\mathcal{V}^{(t+1)}$ at iteration $t+1$, the new set of bases minimise the reconstruction error within each cluster of points given the assignment $\Omega^{(t)}$ (as stated in the second and third lines of the proof). Next in the assignment update stage for the unlabelled data, we obtain the fourth line of the proof. It says that the assignment $k_u^{(t+1)}$ in the $(t+1)$th iteration would only be different from $k_u^{(t)}$ if it gives a smaller reconstruction error for $\mathbf{x}_u$. Finally in the last step of the KSCC algorithm, we update the matching between the cluster assignment and true classes, and it only gets updated if some other matching has a smaller overall reconstruction error for the labelled data $\mathcal{X}_L$. This is reflected in the last two lines of the proof.

---

**Algorithm 2:** KSC with Constraints (KSCC)

**Input** : Labelled and unlabelled data $\mathcal{X}_L$, $\mathcal{X}_U$
Initial cluster assignment $\{k_1, \ldots, k_N\}$
subspace dimension $q$

**repeat**

    *% Stage 1: fitting subspaces*
    **for** $X_k \in \mathcal{X}$ *($k = 1, \ldots, K$)* **do**
        Calculate the eigen-decomposition on the covariance matrix of $X_k$:
        $\text{cov}(X_k) = V_k \Lambda_k V_k^T$
    **end**

    *% Stage 2: updating assignment*
    **for** $\boldsymbol{x}_i \in \mathcal{X}$ *($i = 1, \ldots, N$)* **do**
        Determine the cluster assignment for $\mathbf{x}_i$:
        $k_i = \arg\min_{k_i \in \{1, \ldots, K\}} \left\| \mathbf{x}_i - V_k V_k^T \mathbf{x}_i \right\|_2^2$
    **end**

    *% Stage 3: satisfying constraints*
    Find the best vector $P_n^* \in \mathcal{P}(K)$ to match with the true classes by solving Eq. (21):

$$P_n^* = \arg \min_{\substack{P_n \in \mathcal{P}(K), \\ n \in \{1, \ldots, K!\}}} \left\{ \sum_{l=1}^{K} \left\| X_l - V_{P_{nl}} V_{P_{nl}}^T X_l \right\|_2^2 \right\}$$

    using the Hungarian algorithm

**until** *Iteration number $T$ is reached or the total reconstruction error stops decreasing*

---

## IV. EXPERIMENTAL RESULTS

In this section, we conduct a series of experiments with both synthetic and real data to evaluate the performance of our proposed active learning strategies against three other competing strategies. The cluster performance is measured by the well-known normalised mutual information (NMI) [22].

In order to inspect the influence of data addition and deletion separately, we use three versions of our proposed active learning strategy: *SCAL-A* and *SCAL-D* only take into account the influence of data addition or deletion respectively, and *SCAL* is the combined strategy. We compare the performance of our proposed active learning strategies with three alternative schemes: *MaxResid*, *MinMargin* [3], and *Random* strategy. *MaxResid* selects data that have the largest reconstruction error to their corresponding subspaces. *MinMargin* selects the data that are most equidistant to their two closest subspaces. Lastly as a benchmark, we compare to random sampling and satisfy the constraints using the proposed KSCC algorithm.

### A. Synthetic data

For all synthetic experiments, we initialise the cluster assignments with the best initialisation (the one with the lowest reconstruction error) out of 50 runs of the KSC algorithm. We set the total number of iterations $T$ of the active learning procedure to be $N$, and one point is queried at a time.

**Experiment 1: Varying noise level ($\sigma$).** We investigate the effectiveness of our proposed active learning strategy under varying levels of additive noise. The effectiveness of various strategies are compared under various levels of additive noise. The data corrupted by noise can be expressed as $Y = X + E$, where $X$ is the noise-free data and $E$ the noise component. The noise-free data matrix $X$ are generated from the standard Normal distribution with $N(0, 1)$ along their bases directions. Each entry in the noise data matrix $E$ is generated from standard Normal distribution $N(0, \sigma^2)$, with zero mean and variance $\sigma^2$. We add additive noise levels of $\sigma = 0.2$, 0.4, and 0.6 respectively. Across all noise levels, there are 5 clusters in each data set and each cluster contains 200 points from the same subspace of dimension 10 out of the full dimension 20.

The performance of various strategies under all settings are shown in Table I. As a general trend, all strategies require high proportion of points to be queried to reach perfection as the noise level goes up. It seems most of the advantage of *SCAL* comes from the influence of data addition *SCAL-A*, and it is difficult to say there is a difference between *SCAL* and *SCAL-A*. It is worth noting that *MinMargin* has similar performance to *SCAL* and *SCAL-A* when $\sigma = 0.2$.

**Experiment 2: Varying angles between subspaces ($\theta$).** In order to fix a vector to rotate the subspaces, we apply various active learning strategies on three 3-$D$ examples with subspace dimension $q = 2$. 600 points are generated in total from 3 classes, and every cluster contains 200 points each. Noise with $\sigma = 0.1$ is added to the data, and the between-subspace angle is specified to be 30, 50, and 70 degrees respectively.

The performance results under all scenarios are shown in Table I. Our proposed active strategy *SCAL* and *SCAL-A* outperform all other strategies significantly. For these two strategies, the proportion of data needed in order to achieve perfection decreases as the between-subspace angle increases. Other strategies have to query almost all points in order to achieve perfect performance apart from *MinMargin*, which is our close competitor in the varying noise setting.

Again *SCAL-D* strategy as part of our proposed *SCAL* strategy barely distinguishes itself from *Random* strategy. This is within our expectations for two reasons. First, misclassified points are most likely to belong to the nearest cluster that they have the second least reconstruction error to. Secondly, those

| Parameters | SCAL | SCAL-A | SCAL-D | MaxResid | MinMargin | Random |
|---|---|---|---|---|---|---|
| $\sigma = 0.2$ | **0.30%** | 0.40% | 45.20% | 19.20% | 0.70% | 23.00% |
| $\sigma = 0.4$ | **43.10%** | 46.10% | 99.00% | 98.00% | 83.10% | 99.50% |
| $\sigma = 0.6$ | 85.60% | **85.40%** | 99.90% | 99.10% | 89.50% | 99.50% |
| $\theta = 30$ | **41.67%** | 44.17% | 98.67% | 99.83% | 96.00% | 99.00% |
| $\theta = 50$ | 37.17% | **36.83%** | 99.00% | 98.17% | 69.50% | 99.50% |
| $\theta = 70$ | 32.17% | **31.83%** | 98.83% | 98.50% | 77.67% | 99.83% |

TABLE I: The percentage of points queried before perfect cluster performance is reached on synthetic data sets.
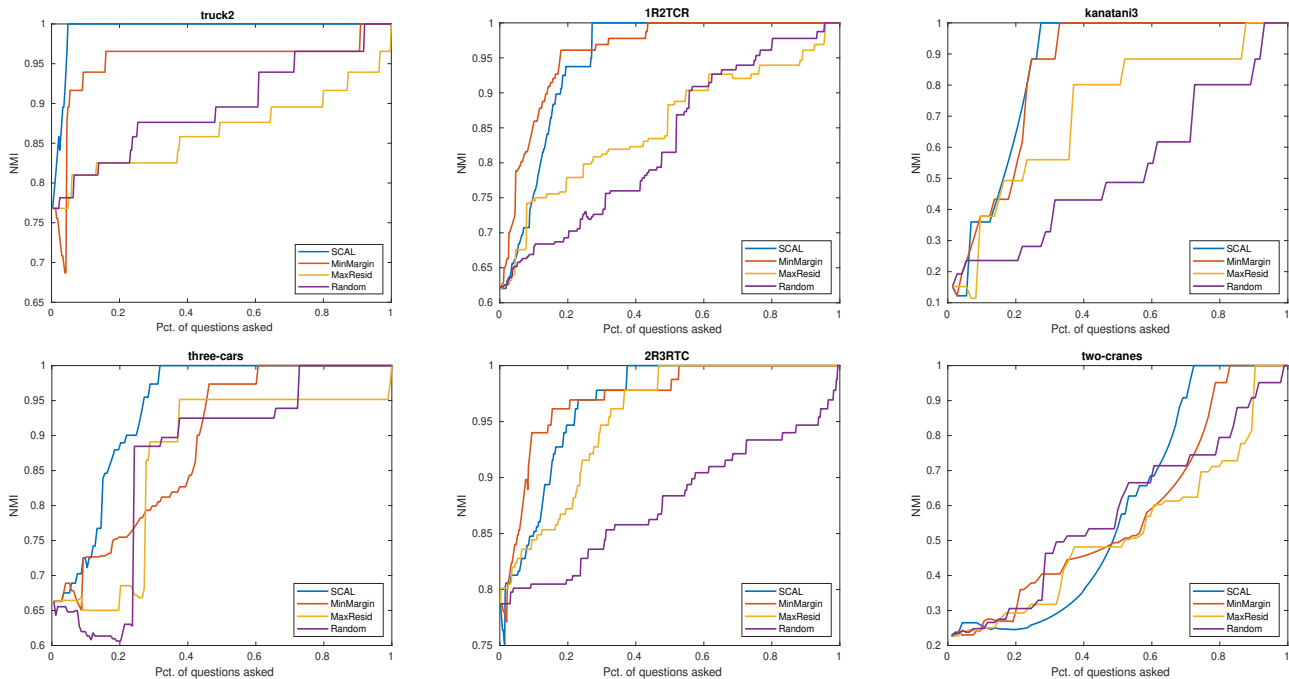


Fig. 1: Performance results measured by NMI on six motion segmentation data sets with KSC initialisation.

points whose deletion has a large influence on their allocated subspaces are likely to be correctly classified in the first place due to the noise in the data.

### B. Real Data

In this section, we conduct experiments on real-world data comparing *SCAL* to various competing strategies. We show the advantage of our proposed active learning strategy when the data exhibit subspace structure. Specifically, we experiment with data sets in motion segmentation and face clustering that have been used previously to demonstrate the effectiveness of subspace clustering [23].

For KSC-based experiments, we experiment with two initialisation schemes. First we initialise with the output given by KSC, which is the set of labels that gives the smallest reconstruction error out of 50 runs. Secondly, we initialise with the output from Sparse Subspace Clustering (SSC) [23] under the default model parameters. Due to the excellent performance of SSC, the aim is to investigate whether the correct initialisation of subspace bases would help accelerate the performance improvement.

All performance results are presented in two measures: the first row of each data set presents the percentage of data that

need to be queried before perfect clustering is reached; the second row presents the area under the curve as a percentage of the total area.

**Motion segmentation.** In this set of experiments, we evaluate the performance of all strategies on six motion segmentation data sets [24]. Motion segmentation refers to the problem of separating the points in a sequence of frames that compose one video after being combined consecutively. Each point can be represented by a $2F$-dimensional vector, in which $F$ is the number of frames in the video [23]. Following the parameter setting from [23], we set the subspace dimensionality $q = 3$ and one point is queried at each iteration.

The performance results are summarised in Table II. It is worth noting that SSC achieves perfect performance on 'truck2' and 'kanatani3', thus there is no need for active learning. The performance improvement over iterations is shown in Fig. 1. We see that the performance of *MinMargin* is very similar to that of *SCAL* most of the time. This is also reflected in the second row of the performance of each data set in Table II. However *SCAL* always achieves perfect cluster performance first, which is what we expect to see due to its ability to query potentially misclassified points that are
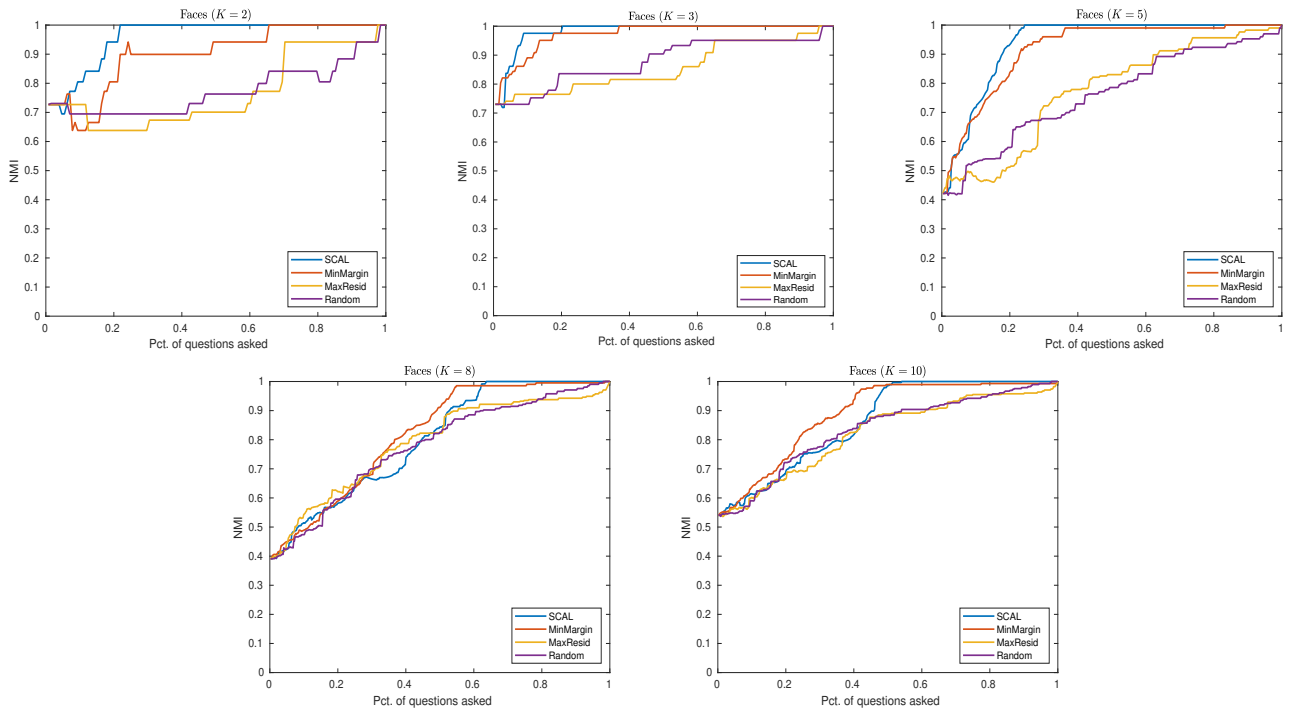
Fig. 2: Performance results measured by NMI on Yale Faces datasets with KSC initialisation.

also informative. The performance of *MaxResid* also improves rapidly in most scenarios, but it struggles to find the points that lead to perfect performance.

| | KSC update (KSC initialisation) | | | |
|---|---|---|---|---|
| | *SCAL* | *MinMargin* | *MaxResid* | *Random* |
| truck2 | **4.53%** | 90.63% | 99.70% | 91.84% |
| | **99.36%** | 95.46% | 86.60% | 89.75% |
| 1R2TCR | **26.98%** | 43.35% | 95.68% | 88.13% |
| | 94.89% | **95.96%** | 85.44% | 83.12% |
| kanatani3 | **26.03%** | 31.51% | 86.30% | 91.78% |
| | **86.05%** | 84.86% | 72.10% | 53.10% |
| three-cars | **31.21%** | 60.12% | 99.42% | 72.25% |
| | **94.18%** | 89.07% | 86.73% | 87.16% |
| 2R3RTC | **37.28%** | 52.51% | 46.49% | 99.20% |
| | 96.68% | **97.23%** | 95.37% | 87.86% |
| two-cranes | **71.28%** | 81.92% | 89.36% | 97.87% |
| | **59.52%** | 58.09% | 53.09% | 58.49% |
| | KSC update (SSC initialisation) | | | |
| 1R2TCR | **26.98%** | 43.35% | 95.68% | 98.02% |
| | 94.89% | **95.96%** | 85.44% | 83.09% |
| three-cars | **1.73%** | 83.82% | 99.42% | 84.39% |
| | **99.94%** | 97.61% | 95.52% | 97.49% |
| 2R3RTC | **9.82%** | 40.88% | 59.92% | 78.96% |
| | **99.76%** | 99.27% | 98.21% | 97.73% |
| two-cranes | **73.40%** | 75.53% | 98.94% | 98.94% |
| | 64.38% | **71.23%** | 45.72% | 65.80% |

TABLE II: Performance on motion segmentation data sets.

**Face clustering.** The original Extended Yale Face Database B [25] consists of 64 images of 38 distinct faces under various lighting conditions. Each original image is of size $192 \times 168$, and have been downsampled to have size $48 \times 42$ [23]. It has previously been shown under the Lambertian assumption that images of a subject lie close to a linear

subspace of dimension 9 [26]. Since the data have intrinsically low dimension, we preprocess the data by projecting onto its first $5K$ principal components as has been done in [8]. Following the experimental settings in [23], we experiment with $K = 2$, 3, 5, 8, and 10. The corresponding data sets are obtained from the SSC package in MATLAB [23].

As before, we apply all active learning strategies with both KSC and SSC initialisations. It is worth noting that SSC achieves perfect performance on the preprocessed data when $K = 2$, thus there is no need for active learning. From the results shown in Table III, we see that the percentage of data needed goes up as $K$ increases. Although the proportion of area under the curve is very similar between *MinMargin* and *SCAL*, *MinMargin* requires a much higher percentage of queries than *SCAL* before perfect clustering is reached.

The performance improvement over time with KSC initialisation is shown in Fig. 2. The initial performance decreases slowly as $K$ increases, and the performance of various active strategies gets closer. With that said, the performance results of *SCAL* and *MinMargin* still stand out from the rest.

## V. EXTENSION TO SPECTRAL CLUSTERING

Finally, we make an initial attempt to extend our active learning framework to the spectral clustering setting. A large number of subspace clustering algorithms are spectral-based methods. These methods construct a pairwise affinity matrix through various optimisation schemes and solve the cluster assignment problem through spectral clustering [23], [27]. We incorporate the queried information in the similarity matrix and compare with other strategies in the spectral setting.

| KSC update (KSC initialisation) | | | | | |
| --- | --- | --- | --- | --- | --- |
| Strategies | $K=2$ | $K=3$ | $K=5$ | $K=8$ | $K=10$ |
| *SCAL* | **21.09%** | **19.79%** | **24.06%** | **63.48%** | **53.91%** |
| | **96.10%** | **98.28%** | **93.80%** | 80.02% | 86.45% |
| *MinMargin* | 64.84% | 36.46% | 83.13% | 98.44% | 99.84% |
| | 90.30% | 97.31% | 91.83% | **81.92%** | **88.95%** |
| *MaxResid* | 96.88% | 95.31% | 99.69% | 99.81% | 99.84% |
| | 77.19% | 85.85% | 77.43% | 79.01% | 82.22% |
| *Random* | 97.66% | 96.35% | 99.69% | 97.85% | 97.97% |
| | 77.59% | 88.08% | 76.27% | 77.77% | 83.46% |

| KSC update (SSC initialisation) | | | | |
| --- | --- | --- | --- | --- |
| Strategies | $K=3$ | $K=5$ | $K=8$ | $K=10$ |
| *SCAL* | **10.94%** | **25.31%** | **28.52%** | **58.91%** |
| | **99.06%** | **99.09%** | **98.24%** | 95.15% |
| *MinMargin* | 16.15% | 38.13% | 92.58% | 62.18% |
| | 98.59% | 98.14% | 98.00% | **97.15%** |
| *MaxResid* | 93.23% | 99.69% | 99.81% | 99.38% |
| | 91.50% | 82.66% | 89.75% | 93.34% |
| *Random* | 91.15% | 76.25% | 99.02% | 91.88% |
| | 92.19% | 94.40% | 91.59% | 94.75% |

| Spectral update (SSC initialisation) | | | | |
| --- | --- | --- | --- | --- |
| Strategies | $K=3$ | $K=5$ | $K=8$ | $K=10$ |
| *SCAL* | **10.94%** | **26.56%** | **40.62%** | 26.56% |
| | **98.91%** | **96.03%** | **91.08%** | 98.24% |
| *SUPERPAC* | 14.06% | 31.25% | **40.62%** | 20.31% |
| | 98.39% | 95.96% | 90.03% | 98.67% |
| *MaxResid* | 90.62% | 98.44% | 98.44% | 57.81% |
| | 93.23% | 92.84% | 86.81% | **98.76%** |
| *Random* | 95.31% | 98.44% | 92.19% | 96.88% |
| | 90.90% | 92.65% | 87.50% | 96.06% |

TABLE III: Performance on Yale Faces data sets.

Using our proposed strategy, the points are queried in the same manner as before. Upon receiving the class information of some points, the constraints are satisfied by editing the affinity matrix. Following the update procedure in [4], we set to ones for those labelled data that belong to the same class and zeros for those that lie in different classes. Spectral clustering is then applied to the edited affinity matrix to obtain labels for all points. As a final step, we apply KSCC to ensure that all grouping constraints are satisfied for the labelled data.

The performance results are shown in the last section of Table III. Note that the authors that propose *MinMargin* have renamed it to *SUPERPAC* for the spectral setting. *SCAL* outperforms other competing strategies in all scenarios apart from when $K=10$. With that said, *SCAL* still enjoys the same level of rate of improvement as *SUPERPAC* and *MaxResid* when $K=10$.

## VI. CONCLUSIONS & FUTURE WORK

We proposed a novel active learning framework for subspace clustering. Ideas from matrix perturbation theory are borrowed to enable efficient estimation of the influence of data deletion and addition as measured by the change in the reconstruction error. New results on the perturbation analysis of data addition are provided as a by-product of our proposed active learning framework. In addition, we propose a constrained subspace clustering algorithm KSCC that monotonically decreases the constrained objective over iterations.

For future research, we plan to consider extensions of our proposed framework in the spectral setting. Our initial experimental results seem promising on the Faces data using the straightforward spectral update on the affinity matrix.

## REFERENCES

[1] H. Su, J. Deng, and L. Fei-Fei, "Crowdsourcing annotations for visual object detection," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
[2] B. Settles, "Curious machines: Active learning with structured instances," Ph.D. dissertation, University of Wisconsin–Madison, 2008.
[3] J. Lipor and L. Balzano, "Margin-based active subspace clustering," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2015.
[4] J. Lipor and L. Balzano, "Leveraging union of subspace structure to improve constrained clustering," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.
[5] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *Journal of Global Optimization*, vol. 16, no. 1, pp. 23–32, 2000.
[6] I. Jolliffe, "Principal component analysis," in *International encyclopedia of statistical science*. Springer, 2011, pp. 1094–1096.
[7] F. Critchley, "Influence in principal components analysis," *Biometrika*, vol. 72, no. 3, pp. 627–636, 1985.
[8] M.-F. Balcan, A. Broder, and T. Zhang, "Margin based active learning," in *International Conference on Computational Learning Theory*, 2007.
[9] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992.
[10] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Advances in neural information processing systems*, 2008.
[11] N. Roy, and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction," in *ICML*, 2001.
[12] P. Donmez, J. G. Carbonell, and P. N. Bennett, "Dual strategy active learning," in *European Conference on Machine Learning*, 2007.
[13] P. Melville and R. J. Mooney, "Diverse ensembles for active learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004.
[14] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proceedings of the twenty-first international conference on Machine learning*, 2004.
[15] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine learning*, vol. 28, no. 2-3, pp. 133–168, 1997.
[16] L. Shi, "Local influence in principal components analysis," *Biometrika*, vol. 84, no. 1, pp. 175–186, 1997.
[17] A. Enguix-González, J. Muñoz-Pichardo, J. Moreno-Rebollo, and R. Pino-Mejías, "Influence analysis in principal component analysis through power-series expansions," *Communications in StatisticsTheory and Methods*, vol. 34, no. 9-10, pp. 2025–2046, 2005.
[18] S.-G. Wang and E. P. Liski, "Effects of observations on the eigensystem of a sample covariance matrix," *Journal of statistical planning and inference*, vol. 36, no. 2-3, pp. 215–226, 1993.
[19] J. Bénasséni, "A correction of approximations used in sensitivity study of principal component analysis," *Computational Statistics*, 2018.
[20] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012.
[21] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
[22] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.
[23] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.
[24] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[25] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 5, pp. 684–698, 2005.

[26] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 2, pp. 218–233, 2003.

[27] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 663–670.

[28] X. Wang and I. Davidson, "Active spectral clustering," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 561–568.

[29] C. Xiong, D. M. Johnson, and J. J. Corso, "Active clustering with model-based uncertainty reduction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 5–17, 2017.

## APPENDIX

### Proof of Proposition 1

In this section, we provide the proof for Proposition 1 in Section III-B of the paper regarding the perturbed form of the covariance matrix after data addition.

**Proposition 1.** The form of $S_{(I)}^+$ can be expressed as,

$$S_{(I)}^+ = S + \frac{l}{n+l}\left[(S_I - S) - (\bar{\mathbf{x}}_I + \bar{\mathbf{x}})(\bar{\mathbf{x}}_I + \bar{\mathbf{x}})^T\right] + \frac{l^2}{(n+l)^2}(\bar{\mathbf{x}}_I + \bar{\mathbf{x}})(\bar{\mathbf{x}}_I + \bar{\mathbf{x}})^T. \tag{24}$$

*Proof.* The sample covariance matrices $S$, $S_I$, and $S_{(I)}^+$ are given as

$$S = \frac{1}{n}X^T\left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)X,$$

$$S_I = \frac{1}{l}X_I^T\left(\mathbf{I}_l - \frac{1}{l}\mathbf{1}_l\mathbf{1}_l^T\right)X_I, \tag{25}$$

$$S_{(I)}^+ = \frac{1}{n+l}X_{I_+}^T\left(\mathbf{I}_{n+l} - \frac{1}{n+l}\mathbf{1}_{n+l}\mathbf{1}_{n+l}^T\right)X_{I_+},$$

in which $\mathbf{I}_n$ is an identity matrix of size $n \times n$ and $\mathbf{1}_n$ is a vector of all ones that has length $n$. Starting from the expression for $S_{(I)}^+$ above, we can write:

$$(n+l)S_{(I)}^+ = X_{I_+}^T X_{I_+} - \frac{1}{n+l}X_{I_+}^T \mathbf{1}_{n+l}\mathbf{1}_{n+l}^T X_{I_+}$$

$$= X^T X + X_I^T X_I - \frac{1}{n+l}\bar{\mathbf{x}}_{I_+}\bar{\mathbf{x}}_{I_+}^T$$

$$= X^T X + X_I^T X_I - \frac{1}{n+l}(n\bar{\mathbf{x}} + l\bar{\mathbf{x}}_I)(n\bar{\mathbf{x}} + l\bar{\mathbf{x}}_I)^T$$

$$= nS + lS_I + \frac{1}{n}X^T\mathbf{1}_n\mathbf{1}_n^T X + \frac{1}{l}X_I^T\mathbf{1}_l\mathbf{1}_l^T X_I$$

$$\quad - \frac{1}{n+l}(n\bar{\mathbf{x}} + l\bar{\mathbf{x}}_I)(n\bar{\mathbf{x}} + l\bar{\mathbf{x}}_I)^T$$

$$= nS + lS_I + n\bar{\mathbf{x}}\bar{\mathbf{x}}^T + l\bar{\mathbf{x}}_I\bar{\mathbf{x}}_I^T$$

$$\quad - \frac{nl}{n+l}\left(\frac{n}{l}\bar{\mathbf{x}}\bar{\mathbf{x}}^T + \bar{\mathbf{x}}\bar{\mathbf{x}}_I^T + \bar{\mathbf{x}}_I\bar{\mathbf{x}}^T + \frac{l}{n}\bar{\mathbf{x}}_I\bar{\mathbf{x}}_I^T\right)$$

$$= nS + lS_I - \frac{nl}{n+l}\left(\bar{\mathbf{x}}\bar{\mathbf{x}}^T + \bar{\mathbf{x}}\bar{\mathbf{x}}_I^T + \bar{\mathbf{x}}_I\bar{\mathbf{x}}^T + \bar{\mathbf{x}}_I\bar{\mathbf{x}}_I^T\right)$$

$$= nS + lS_I - \frac{nl}{n+l}(\bar{\mathbf{x}}_I + \bar{\mathbf{x}})(\bar{\mathbf{x}}_I + \bar{\mathbf{x}})^T$$

$$= (n+l)S + l(S_I - S) - l(\bar{\mathbf{x}}_I + \bar{\mathbf{x}})(\bar{\mathbf{x}}_I + \bar{\mathbf{x}})^T$$

$$\quad + \frac{l^2}{n+l}(\bar{\mathbf{x}}_I + \bar{\mathbf{x}})(\bar{\mathbf{x}}_I + \bar{\mathbf{x}})^T,$$

from which the result follows. $\qquad\square$

### Proof of Proposition 2

In this section, we provide the proof for Proposition 1 in Section III-B of the paper regarding the perturbed form of the covariance matrix after the addition of one single point.

**Proposition 2.** Following the same line of analysis, we can show that the perturbed covariance matrix in the case when $l = 1$ can be expressed as,

$$S_{(i)}^+ = S + \frac{1}{n+1}\left[(\bar{\mathbf{x}} - \mathbf{x}_i)(\bar{\mathbf{x}} - \mathbf{x}_i)^T - S\right] - \frac{1}{(n+1)^2}(\bar{\mathbf{x}} - \mathbf{x}_i)(\bar{\mathbf{x}} - \mathbf{x}_i)^T, \tag{26}$$

in which we can think of $\frac{1}{n+1}$ and $\left[(\bar{\mathbf{x}} - \mathbf{x}_i)(\bar{\mathbf{x}} - \mathbf{x}_i)^T - S\right]$ as $\epsilon$ and $S^{(1)}$ respectively.

*Proof.*

$$(n+1)S_{(i)}^+ = X_{i_+}^T X_{i_+} - \frac{1}{n+1}X_{i_+}^T \mathbf{1}_{n+1}\mathbf{1}_{n+1}^T X_{i_+}$$

$$= X^T X + \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{n+1}(n\bar{\mathbf{x}} + \mathbf{x}_i)(n\bar{\mathbf{x}} + \mathbf{x}_i)^T$$

$$= nS + \frac{1}{n}X^T\mathbf{1}_n\mathbf{1}_n^T X + \mathbf{x}_i\mathbf{x}_i^T$$

$$\quad - \frac{1}{n+l}(n\bar{\mathbf{x}} + l\bar{\mathbf{x}}_I)(n\bar{\mathbf{x}} + l\bar{\mathbf{x}}_I)^T$$

$$= nS + \frac{1}{n}X^T\mathbf{1}_n\mathbf{1}_n^T X$$

$$\quad - \frac{n}{n+1}\left(\bar{\mathbf{x}}\bar{\mathbf{x}}^T + 2\bar{\mathbf{x}}\mathbf{x}_i^T - n\mathbf{x}_i\mathbf{x}_i^T\right)$$

$$= nS + n\bar{\mathbf{x}}\bar{\mathbf{x}} - \frac{n}{n+1}\left(\bar{\mathbf{x}}\bar{\mathbf{x}}^T + 2\bar{\mathbf{x}}\mathbf{x}_i^T - n\mathbf{x}_i\mathbf{x}_i^T\right)$$

$$= nS - \frac{n}{n+1}\left(2\bar{\mathbf{x}}\mathbf{x}_i^T - \bar{\mathbf{x}}\bar{\mathbf{x}}^T - \mathbf{x}_i\mathbf{x}_i^T\right)$$

$$= nS + \frac{n}{n+1}(\bar{\mathbf{x}} - \mathbf{x}_i)(\bar{\mathbf{x}} - \mathbf{x}_i)^T$$

$$= nS + (\bar{\mathbf{x}} - \mathbf{x}_i)(\bar{\mathbf{x}} - \mathbf{x}_i)^T$$

$$\quad - \frac{1}{n+1}(\bar{\mathbf{x}} - \mathbf{x}_i)(\bar{\mathbf{x}} - \mathbf{x}_i)^T.$$

$\qquad\square$