Statistical Methods for Samples of Interaction Networks

George Bolt, B.Sc., M.Res

Submitted for the degree of Doctor of Philosophy at Lancaster University.

December 2023

STOR-i
excellence with impact

For Mum & Dad

Abstract

Network data arises through the observation of relational information between a collection of entities. An ubiquitous example of such data are social networks, where friendships amongst a sample of people are observed. However, there has begun to appear other subtly different forms of data which also fit this description. Notably, work in the literature has recently considered when (i) the units of observation within a network are edges or paths, often referred to as interaction networks, with examples such as emails between people or a series of page visits to a website by a user, and (ii) one observes a *sample* of networks, for example, in neuroscience applications, brain scan data of a single patient is often processed into a network representation, with a multi-patient study thus leading to a sample of networks.

However, the intersection of (i) and (ii) has presently not been considered, that is, where a sample of interaction networks are observed. For example, one might observe a *sample* of users navigating the same website. Use of currently proposed methods to analyse such data would either be inappropriate or require one to first aggregate data into another form, incurring a potential loss of information. Motivated by this gap in the literature, this thesis proposes statistical methods suitable for the analysis of samples of interaction networks.

In this regard, two main contributions are made. Firstly, the problem of measuring the distance between two interaction networks is considered. Distances are an incredibly useful and versatile tool, opening to door to a variety of analytical methodologies,

such as dimension reduction and clustering algorithms. Secondly, building upon this work, the problem of summarising a sample of interaction networks is considered. Of particular focus is obtaining analogues of the mean and variance in this non-trivial scenario, that is, where data points are themselves interaction networks. To this end, a novel Bayesian modelling framework is proposed. Given a user-specified distance measure, we construct Gaussian-like distributions over the space of interaction networks, that is, models parameterised via location and scale. This approach raises significant computational challenges; not only are resulting posterior distributions doubly-intractable, but the parameter space includes the space of interaction networks, which is both discrete and multidimensional. As such, specialised Markov chain Monte Carlo (MCMC) algorithms are developed which circumvent these issues, facilitating parameter inference for the proposed models. Crucially, the location and scale parameters provide analogues of the mean and variance, respectively, resulting in the desired summary measures.

Across both pieces of work, simulation studies are undertaken to confirm the efficacy of proposed methods and to explore their properties. Additionally, their practical applications are illustrated through example analyses of two open-source datasets: (i) an in-play football dataset released by StatsBomb, and (ii) a dataset of user interactions with the location-based social network Foursquare.

Acknowledgements

First and foremost, I would like to thank my supervisors, Simón and Chris. Its been a tough journey, particularly given to the convenient occurrence of a global pandemic along the way. Nonetheless, I have enjoyed it thoroughly, and will always look back with fond memories. The guidance you both provided, and the patience you have shown, I will forever be grateful for. I will always be glad I had you both as my supervisors.

I would also like to thank everyone at STOR-i, including fellow students, senior staff and the admin team. Not only did they provide me this opportunity, but the family-like culture and support structures they have cultivated undoubtedly had a positive impact on my personal development, and consequently this work. Of particular mention are Jake, Graham, Luke and Drupad. From the early days of group projects during the MRes, Madibaland being one of note, to our often tangential discussions over lunch. The laughs were always what was needed to make it through.

Of course, my family must also be thanked. Mum and Dad, who have supported me the whole way, my siblings Sam, Katie (and Co) and Will (and Co), and my late Gran, who was my number one fan and I know would have been very proud. Having that steady support, particularly during the challenging periods, was a blessing I will always be grateful for.

Finally, it would be amiss to not acknowledge the contributions made by coffee, without which I am unsure this thesis would have ever been completed.

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.

The word count for this thesis is approximately 70,000 words.

George Bolt

Contents

List of Figures

List of Tables

List of Abbreviations

MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
EM	Expectation maximisation
LCS	Longest common subsequence
LSP	Longest common subpath
EMD	Earth mover's distance
DTW	Dynamic time warping
SIS	Spherical interaction sequence
SIM	Spherical interaction multiset
ER	Erdős-Rényi
SBM	Stochastic blockmodel
LSM	Latent space model
RDPG	Random dot product graph
ERGM	Exponential random graph model
HW	Hollywood
SNF	Spherical network family
CER	Centered Erdős-Rényi
ME	Measurement error
MDS	Multidimensional scaling
PCA	Principal components analysis

t-SNE	t-distributed stochastic neighbourhood embedding
UMAP	Uniform manifold approximation and projection
HDBSCAN	Hierarchical density-based spatial clustering of applications with noise
i.i.d.	Independent and identically distributed

List of Symbols

$\mathcal{G}$	Graph or multigraph
$A^{\mathcal{G}}$	Adjacency matrix of graph $\mathcal{G}$
$\mathcal{V}$	Vertex set
V	Size of vertex set $\mathcal{V}$
$\mathcal{I}$	Interaction (path)
$\mathcal{S}$	Interaction sequence
$\mathcal{E}$	Interaction multiset
$m_{\mathcal{E}}(\cdot)$	Multiplicity function of multiset $\mathcal{E}$
$\mathrm{Supp}(\mathcal{E})$	Support of multiset $\mathcal{E}$
$\mu_{\mathcal{E}}$	Distribution obtained by normalising multiset $\mathcal{E}$
$\mathcal{G}_{\mathcal{S}}$	Aggregate multigraph of interaction sequence $\mathcal{S}$
$\mathcal{G}_{\mathcal{E}}$	Aggregate multigraph of interaction multiset $\mathcal{E}$
$v^{\mathcal{S}}$	Aggregate vector of interaction sequence $\mathcal{S}$
$v^{\mathcal{E}}$	Aggregate vector of interaction multiset $\mathcal{E}$
$\boldsymbol{\mathcal{I}}^*$	Space of all interactions
$\boldsymbol{\mathcal{S}}^*$	Space of all interaction sequences
$\boldsymbol{\mathcal{E}}^*$	Space of all interaction multisets
$\boldsymbol{\mathcal{I}}^*_K$	Space of interactions with length at most K
$\boldsymbol{\mathcal{S}}^*_{K,L}$	Space of interaction sequences with at most L interactions, each of length at most K

$\boldsymbol{\mathcal{E}}^*_{K,L}$	Space of interaction multisets with at most L interactions, each of length at most K
$\mathcal{M}$	Matching (sequence or multiset)
$\mathcal{C}$	Sequence coupling
$\mathbf{P}$	Distribution coupling matrix
$\mathbb{P}(\cdot)$	Probability
$\mathbb{E}[\cdot]$	Expectation
$\mathbb{V}\mathrm{ar}[\cdot]$	Variance

Chapter 1

Introduction

Network data, in the most general sense, arises through observation of relational information amongst a collection of entities. A typical example is social network data, where friendships amongst a sample of people are observed. In this case, people would correspond to the 'entities', whilst friendships would represent observed 'relational information'. However, this is but one example, and improvements in data collection technologies combined with the inherent connectivity of the world in which we live has led to the appearance of various other forms of data fitting this description.

This thesis concerns the proposal of statistical methods to analyse a form of network data which has at present not been considered. In particular, it concerns the setting in which a sample of interaction networks are observed (Figure 1.0.1). At a high-level, an interaction network consists of a series of paths over a set of vertices (representing the entities of interest). As a motivating example, consider a user navigating a website. Here vertices correspond to web pages, whilst an interaction might represent a single online session, with a user visiting a series of pages in succession. In this way, a single interaction network would represent the historical website navigation of a single user. For example, a single row of Figure 1.0.1 would represent five

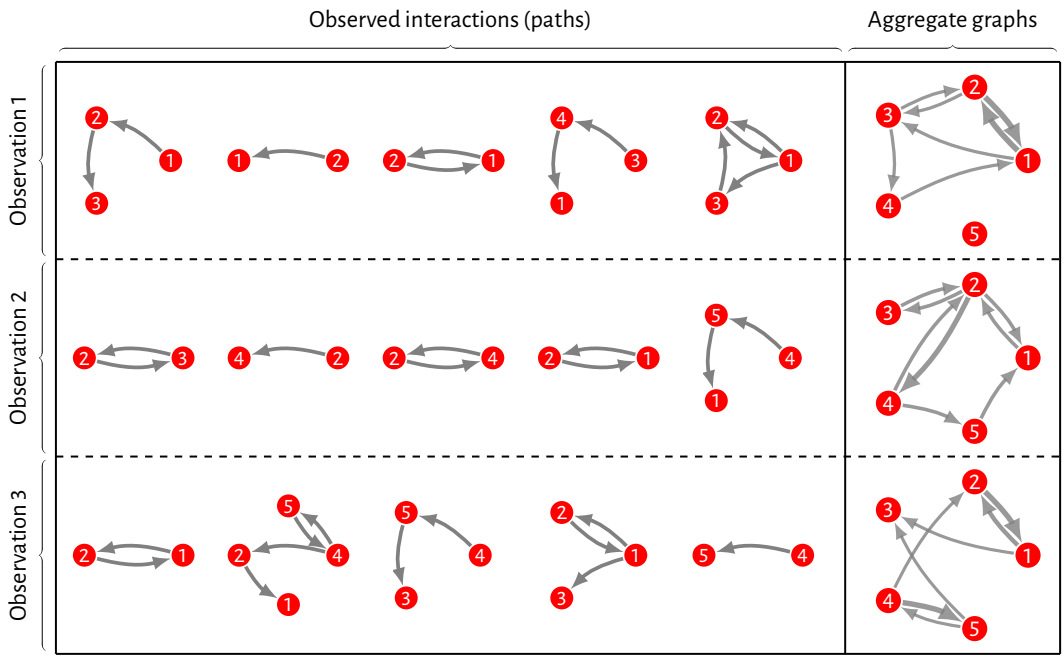


Figure 1.0.1: Visualising a sample of interaction networks. Here each row represents an observation, consisting of paths over a shared set of vertices. On the right is shown each observation's aggregate graph, where the weight of an edge is proportional to the number of traversals between the given vertices.

observed sessions of one user. Moreover, provided there is more that one user of this website, this would result in a *sample* of interaction networks.

When faced with data of this form, that is, a sample wherein each observation or data point is itself an interaction network, there are various questions one may consider answering. For example:

- Can we identify clusters of observations which are similar?
- How can we summarise a sample of observations? For example:
  - What is an average in this context?
  - How can variability of these data be quantified?
- Given two samples, how do we conduct a two-sample hypothesis test? What about a k-sample hypothesis test?

- Given covariate information at the level of observations, for example, a user's age, is it possible to predict this given an observed interaction network?

One might be inclined to ask what is the benefit of being able to answer such questions? Of course, this will depend on the specific application, but to provide some examples, let us return the motivating scenario of analysing website navigation data. Considering the perspective of the website owner, methods which can answer the above questions could, for example, be used to inform site improvements. In particular, clustering could be used to uncover groups of users who interact with the website in a similar manner. Combining this with summarisation, which would assist interpretation of inferred clusters, an overall picture of the different ways in which the site is being used would be obtained, along with the number of users involved. With such information, one could identify areas of the site to improve, or use it to inform the recommendation of relevant content. Moreover, there may be users which one knows little about, but for which there is some historical website navigation data. In this case, predictive methods could be used to infer values of interest, such as their age or area of occupation, which could further inform the personalisation of content.

Though methods have been proposed to answer such questions when faced with a sample of networks, none have considered samples of *interaction* networks. Instead, these methods generally assume observations are represented as graphs, and thus would require first aggregating observations as shown in Figure 1.0.1, bringing with it a potential loss of information.

With this, this thesis considers the proposal of novel methodologies capable of answering such questions whilst respecting the data structure. Towards this end, two main contributions are made:

1. In Chapter 3, the problem of measuring the dissimilarity of two interaction networks is considered. Here various measures are surveyed, drawing on inspiration from areas such as optimal transport and time series analysis. For each

distance, their practical use is guided by the statement and proof of theoretical properties along with discussions of computational schemes and their associated costs. Simulation studies are also undertaken to highlight the relative strengths and weaknesses of the distances introduced, and what features they can and cannot capture. Finally, through example data analyses it is illustrated how distances can be used to both cluster interaction networks and to predict network-level covariate information;

2. In Chapter 4, the problem of summarising a sample of interaction networks is considered. Here a novel Bayesian modelling framework is proposed, building upon the work of Chapter 3. Namely, given a practitioner-specified distance measure, we construct families of models via location and scale parameters, akin to a Gaussian distribution over the space of networks. The location and scale parameters can thus operate as analogues of the mean and variance respectively, providing statistically reasoned answers to questions alluded to above. To facilitate parameter inference, a specialised Markov chain Monte Carlo (MCMC) algorithm is proposed, capable of not only circumventing issues pertaining to double-intractability posterior distributions, but also navigating a non-trivial discrete multi-dimensional parameter space.

The remainder of this thesis is structured as follows. In Chapter 2, the relevant literature on statistical network analysis will be reviewed, then, in Chapters 3 and 4, the two main contributions of this thesis are outlined, as discussed above. Finally, conclusions are drawn in Chapter 5, including discussion around limitations of the present work and potential future directions.

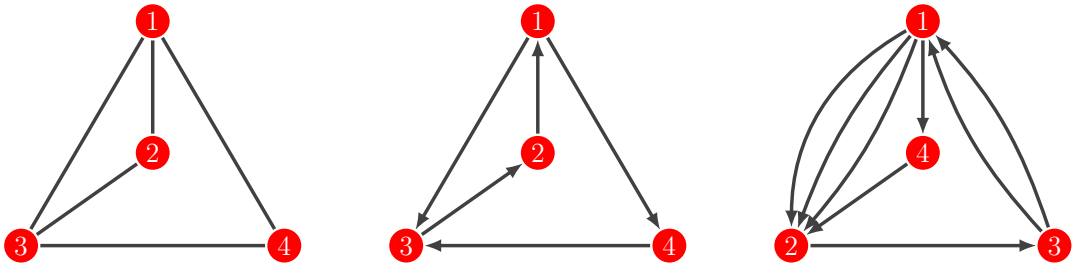
Chapter 2

Statistical Network Analysis

Interest in the analysis of network data has grown rapidly in recent decades, resulting in the development of a burgeoning sub-field thereof often referred to as statistical network analysis (Salter-Townshend et al., 2012; Kolaczyk and Csárdi, 2014), which, as the name suggests, seeks to apply statistical methods to the analysis of networks. In this section, various advances in this area relevant to the contributions of this thesis will be outlined.

2.1 Graph representation

The ubiquitous approach to analysing network data is to interpret it mathematically as a *graph*. A graph consists of two sets $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ where $\mathcal{V}$ is a set of *vertices* and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of *edges* whereby and edge $(u, v) \in \mathcal{E}$ for $u \in \mathcal{V}$ and $v \in \mathcal{V}$ encodes the presence of a relation from u to v. Considering the general definition of network data provided in Chapter 1, the vertices represent 'entities' whilst edges represent 'relational information'. For example, in a social network $\mathcal{V}$ would represent a group of people and $(u, v) \in \mathcal{E}$ would mean person u was friends with person v. Letting $V := |\mathcal{V}|$ denote the size of the vertex set, generally one assumes $\mathcal{V} = \{1, \ldots, V\}$; if not one can always construct a mapping from whatever entities one is considering to



(a) Undirected graph (b) Directed graph (c) Directed multigraph

Figure 2.1.1: Visualising different graph types, each with $\mathcal{V} = \{1, \ldots, 4\}$ and four edges, where we have (a) an undirected graph, (b) a directed graph and (c) a directed multigraph.

such values.

A graph can be undirected or directed (Figures 2.1.1a and 2.1.1b), where if undirected then $(u, v) \in \mathcal{E} \iff (v, u) \in \mathcal{E}$, so that edges are always reciprocated, whilst in a directed graph this need not be the case, with edges instead having a direction. Returning to the social network example, an undirected graph would imply friendships are always mutual, whereas a directed graph could represent a scenario whereby one person considers themselves friends with another whilst the other person does not. Graphs may or may not have self edges, that is, $(v, v) \in \mathcal{E}$ for some $v \in \mathcal{V}$, and an undirected graph with no self edges is typically referred to as a *simple graph*.

Graphs can also be *weighted*, whereby each edge $e \in \mathcal{E}$ is associated a weight w_e, typically assumed to be a positive real value. In the special case where weights are positive integers $w_e \in \mathbb{N}$ the resultant $\mathcal{G}$ is also referred to as a *multigraph* (Figure 2.1.1c). In this case, one assumes instead that $\mathcal{E}$ is a *multiset* of edges, whereby an edge can appear more than once, with w_e representing the multiplicity of the edge e in the multigraph $\mathcal{G}$.

An alternative representation for graphs frequently used is the *adjacency matrix*. For a (non-weighted) graph $\mathcal{G}$ the adjacency matrix $A^{\mathcal{G}} \in \{0, 1\}^{V \times V}$ is defined as

follows

$$A_{ij}^{\mathcal{G}} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$$

that is, a non-zero entry in the (i,j)th entry indicates the presence of the respective edge in the graph. Observe if $\mathcal{G}$ is undirected then $A^{\mathcal{G}}$ is symmetric, and moreover if $\mathcal{G}$ is a simple graph then the diagonal entries will all be zero. When $\mathcal{G}$ is a weighted graph $A^{\mathcal{G}}$ is defined by letting the non-zero entries be equal to the edge weights, that is

$$A_{ij}^{\mathcal{G}} = \begin{cases} w_e & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$$

so that in the special case where $\mathcal{G}$ is a multigraph one will have $A^{\mathcal{G}} \in \mathbb{Z}_{\geq 0}^{V \times V}$ with $A_{ij}^{\mathcal{G}}$ denoting the multiplicity of edge (i,j).

The final concept to define is that of a *subgraph*. We say $\mathcal{H} = (\mathcal{V}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$ is a subgraph of $\mathcal{G}$ if $\mathcal{V}_{\mathcal{H}} \subseteq \mathcal{V}$ and $\mathcal{E}_{\mathcal{H}} \subseteq \mathcal{E}$, with the following notable examples: (i) *triangles*, where three nodes are connected by three edges, for example in Figure 2.1.1a we see a triangle between vertices 1,2 and 3, and (ii) K*-stars*, where one central node is connected to $K-1$ others with a single edge, for example in Figure 2.1.1a we see a 3-star between vertices 1,3, and 4.

2.2 Analysing a single network

In the standard scenario, a single network is observed, that is, a single graph $\mathcal{G}$, and interest is in assessing its structural properties. Popular methods proposed in this regard include the use of descriptive summaries, as well as model-based approaches, which assume the observed graph was sampled at random from some unknown distribution. In the following subsections, notable examples of both appearing in the literature will be outlined.

2.2.1 Network statistics

Given a network, it is natural to ask what are its distinctive features? How well-connected is it? Are there particular connective patterns? Are vertices of the network grouped into communities? These and other questions have motivated the development of measures to quantify the presence of different qualitative features within a network. In this section, some popular measures will be outlined.

The *degree* of a node $v \in \mathcal{V}$, which we denote $d(v)$, is defined to be the number of incident edges, for example, considering the graph of Figure 2.1.1a, $d(1) = 3$, $d(2) = 2$, $d(3) = 3$ and $d(4) = 2$. Note in a directed graph one will typically distinguish the direction of incident edges, leading to both in and out degrees. Noticing the degree is local to a node, towards an overall picture of the whole network, one can consider what is referred to as the *degree sequence*, which is simply the vector of degrees $(d(v))_{v\in\mathcal{V}}$, which can also be aggregated to form the *degree distribution*, given by

$$D_{\mathcal{G}}(k) := \frac{|\{v \in \mathcal{V} \,:\, d(v) = k\}|}{V}$$

representing the probability a randomly chosen node has degree k. For example, in a social network one might expect many people to have a few friends along with perhaps a hand-full of very well-connected people with many friends. This property would thus manifest itself via a heavy-tailed degree distribution.

Though the degree of a node gives a sense of how well connected it is within the network, it is somewhat local, considering only its immediate neighbours. As such, other measures have been proposed which look more globally at the role of the vertex within the network, often referred to as *centrality* measures. For example, the *closeness centrality* of node $v \in \mathcal{V}$ is given by

$$\mathrm{CL}_{\mathcal{G}}(v) = \frac{V-1}{\sum_{u\in\mathcal{V}} d(u,v)}$$

where $d(u, v)$ denotes the length of the shortest path in $\mathcal{G}$ between vertices u and v, also known as the *geodesic distance*. Observe when a node is very well connected it will have a low geodesic distance to many vertices, leading to a higher closeness centrality. Moreover, the multiplication by $V - 1$ ensures that $\mathrm{CL}_{\mathcal{G}}(v) \in [0, 1]$, making it comparable across networks of different size. As with vertex degrees, an overall picture of the network can be obtained by considering the sequence of centralities $(\mathrm{CL}_{\mathcal{G}}(v))_{v \in \mathcal{V}}$ or their empirical distribution.

A property also of interest is the propensity for two vertices sharing a common neighbour of being connected, often referred to a *transitivity*. Again, one can imagine this is common in social networks, where the likelihood of two people being friends might be inflated given they have a friend in common. It can be defined via a ratio of subgraph counts as follows

$$\mathcal{T}(\mathcal{G}) = \frac{\tau_{\triangle}(\mathcal{G})}{\tau_3(\mathcal{G})}$$

where $\tau_{\triangle}(\mathcal{G})$ counts the total number of triangles and $\tau_3(\mathcal{G})$ counts the total number of connected triples, that is, 3-stars, representing an empirical estimate of the probability two nodes are connected given they share a common neighbour.

Finally, often networks exhibit a structure whereby one can partition vertices into groups such that there are more edges within groups than between them. In the literature, such groups of vertices are often referred to as "communities", with this being referred to the presence of "community structure" in the network. A popular way to quantify this is via the *modularity* (Newman and Girvan, 2004). Supposing that $\boldsymbol{z} = (z_1, \ldots, z_V)$ is vector which divides the V nodes of $\mathcal{G}$ into K communities, whereby $z_v = k$ if vertex v is in the kth community, the modularity essentially measures how well separated the communities defined by $\boldsymbol{z}$ are. First, let E_{kl} denote the proportion of edges in $\mathcal{G}$ between communities k and l, whilst also letting $a_k = \sum_l E_{kl}$ denote the fractions of edges involving a vertex in the kth community. If $\boldsymbol{z}$ partitions the graph into communities well we expect the sum of within-community propor-

tions $\sum_{k=1}^{K} E_{kk}$ to be large. However, this would have the highest value for the $\boldsymbol{z}$ placing all vertices in a single community. To combat this, Newman and Girvan (2004) propose to make a slight adjustment. They note if edges occur randomly with no regard for communities to which vertices belong one would expect $E_{kl} = a_k a_l$, which leads to the following definition of the modularity

$$\mathcal{Q}(\mathcal{G}, \boldsymbol{z}) := \sum_{k=1}^{K} (E_{kk} - a_k^2),$$

where the adjustment a_k^2 has been included, representing a baseline proportion of edges one expects within the kth community, given its prominence within the network.

These various summaries provide a means to quantitatively assess some structural aspect of a given network. Not only is this useful for contrasting and comparing observed networks, but also provides a means to inform the development of network models, whereby one can seek to propose models which generate networks exhibiting often-observed features. In the next subsection, some well-known statistical network models that have done exactly this will be outlined.

2.2.2 Statistical network models

The statistical modelling approach to analysing a network $\mathcal{G}$ is to assume it was drawn at random from some probability distribution $p(\mathcal{G}|\theta)$ where θ are some unknown model parameters. The hope is by conducting statistical inference of θ given $\mathcal{G}$ one gains insight regarding some structural aspect of the network. In this section, four influential models that have been proposed in the literature will be outlined.

Note here the vertex set $\mathcal{V}$ is assumed to be fixed, so that only the edges $\mathcal{E}$ are random. For simplicity, it will also be assumed $\mathcal{G}$ is a simple graph, that is, an undirected unweighted graph with no self-edges. In this way, each model will correspond

to a different distribution $p(\mathcal{G}|\theta)$ over the discrete space of all simple graphs over the vertex set $\mathcal{V}$.

Arguably the simplest model is that proposed by Erdős et al. (1960). Known appropriately as the Erdős-Rényi (ER) model, this assumes each edge $(v, u) \in \mathcal{V} \times \mathcal{V}$ is included with some fixed probability, independent of all others, which can be written in terms of sampling the entries of the adjacency matrix independently as follows

$$A^{\mathcal{G}}_{ij} \,|\, p \sim \text{Bernoulli}(p)$$

where $p \in (0, 1)$ is the single model parameter denoting the probability of an edge appearing between any two vertices.

Being such a simple model, the ER model is unable to capture many of the network features mentioned in the previous section, such as transitivity or community structure. This has motivated the proposal of models which impose some further structure on how edges are sampled.

One such model is the stochastic block model (SBM), which seeks to engender community structure in the sampling of edges. We here present the formulation of Nowicki and Snijders (2001), though there have been various extensions proposed (see Lee and Wilkinson 2019 for an extensive review). This assumes vertices are partitioned into K communities, encoded via the $V \times K$ matrix $Z = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_V)^{\mathrm{T}}$ where $\boldsymbol{z}_i = (0, \ldots, 0, 1, 0, \ldots, 0)^{\mathrm{T}}$ with $z_{ik} = 1$ indicating the ith vertex belongs to the kth community. Given a $K \times K$ matrix B, with $B_{kl} \in (0, 1)$ representing the probability of an edge between the kth and lth communities, a graph $\mathcal{G}$ is then sampled via it adjacency matrix as follows

$$A^{\mathcal{G}}_{ij} \,|\, B, Z \sim \text{Bernoulli}(\boldsymbol{z}_i^{\mathrm{T}} B \boldsymbol{z}_j),$$

where matrix multiplication is used to index the entry of B corresponding to the com-

munities of the ith and jth vertices. The model parameters in this case are thus the matrix B along with the community memberships Z, where the latter are typically treated a unknown latent variables.

Nowicki and Snijders (2001) approached parameter inference from a Bayesian perspective, using Beta priors for the entries of B and a combination of Dirichlet and Multinomial distributions for setting a prior on the community memberships Z, before using MCMC to obtain samples from the posterior. Others have since considered alternative approaches to inference, such as variational methods (see Lee and Wilkinson 2019).

Another influential model is the latent space model (LSM) proposed by Hoff et al. (2002).[1] Here, vertices are assumed to be embedded in Euclidean space with those closer together more likely to be connected. Observe that, thanks to the geometry of Euclidean space, such a model naturally captures transitivity: if one point is close to two others, they are also likely to be close. Given chosen dimension d, let the $V \times d$ matrix $Z = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_V)^{\mathrm{T}} \in \mathbb{R}^{V \times d}$ denote the latent positions of all vertices, whereby the ith row $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{id})^{\mathrm{T}} \in \mathbb{R}^d$ denotes the latent position of the ith vertex. For the LSM model, a graph $\mathcal{G}$ is then sampled via its adjacency matrix as follows

$$A^{\mathcal{G}}_{ij} \,|\, \alpha, Z \sim \text{Bernoulli}(p_{ij})$$
$$\text{logit}(p_{ij}) = \alpha - |\boldsymbol{z}_i - \boldsymbol{z}_j|$$

where $\alpha \in \mathbb{R}$ is an intercept parameter controlling density of the network, $|\boldsymbol{z}_i - \boldsymbol{z}_j|$ denotes the Euclidean distance between the ith and jth latent positions, and $\text{logit}(p) := \log(p/(1-p))$ denotes the logit link function.

Inference in this case regards estimation of α and the latent positions Z. Hoff et al. (2002) took a Bayesian approach, assuming Gaussian priors for α and the la-

[1] Hoff et al. (2002) actually considered two models, the latent *distance* model and latent *projection* model, both of which also incorporate covariate information at the level of edges. For simplicity, only the former will be presented here and the covariate terms left out.

tent positions before sampling from the posterior via MCMC. This model has also been extended by Handcock et al. (2007), who assume a Gaussian mixture prior for the latent coordinates to permit the recovery of community structure, again taking a Bayesian inference approach via MCMC. Further work has also been done on alternative inferential approaches, such as Salter-Townshend and Murphy (2013), who consider variational Bayesian inference, Raftery et al. (2012), who proposed a faster Frequentist-based approach via a likelihood approximation, or more recently Sharrock et al. (2023), who proposed particle-based variational inference methods for use in general latent variable models, showing how these can be invoked to fit the LSM.

Another model which has seen recent attention in the literature is the random dot product graph (RDPG) model. Originally appearing in Young and Scheinerman (2007), it has since been extensively reviewed by Athreya et al. (2017). Much like the LSM model, the RDPG model assumes vertices have some latent position in Euclidean space, constructing edge probabilities by relating the latent positions of vertices. However, in contrast with the LSM, the RDPG uses the dot product as a means to compare two latent positions. Supposing that $Z \in \mathbb{R}^{V \times d}$ denotes a matrix of latent positions for all vertices, as seen in the LSM model above, the RDPG model samples a graph $\mathcal{G}$ via is adjacency matrix as follows

$$A^{\mathcal{G}}_{ij} \mid Z \sim \text{Bernoulli}(\boldsymbol{z}_i^{\mathrm{T}} \boldsymbol{z}_j),$$

where since one must have $\boldsymbol{z}_i^{\mathrm{T}} \boldsymbol{z}_j \in [0, 1]$ it is typically assumed the latent positions lie in a suitably constrained subset of $\mathbb{R}^d$. Recalling the property of the dot product

$$\boldsymbol{z}_i^{\mathrm{T}} \boldsymbol{z}_i = |\boldsymbol{z}_i||\boldsymbol{z}_j| \cos(\theta)$$

where θ is the angle between $\boldsymbol{z}_i$ and $\boldsymbol{z}_i$ (as vectors), observe the probability of two vertices having an edge is larger when their latent positions are closer in *angle*. More-

over, the magnitude of a vertex $|\boldsymbol{z}_i|$ increases the probability of edges with all other vertices, and in this way it reflects the general connectivity of the ith vertex.

Observe the RDPG, in using latent positions in this way, will share many properties of the LSM. In particular, transitivity will naturally be captured, whilst community structure can be captured by the clustering of vertices in the latent space. Moreover, the SBM can also be seen as a special case of the RDPG model: provided one can write $B = \tilde{B}\tilde{B}^{\mathrm{T}}$ for some $\tilde{B} \in [0,1]^{K\times d}$ then, writing $\tilde{B} = (\tilde{\boldsymbol{b}}_1, \ldots, \tilde{\boldsymbol{b}}_K)^{\mathrm{T}}$, if the ith vertex is in the kth community we let $\boldsymbol{z}_i = \tilde{\boldsymbol{b}}_k$ (the kth row of $\tilde{B}$).

Inference for the RDPG model amounts to estimating the unknown latent positions Z. As outlined in Athreya et al. (2017), this is typically approached from a Frequentist perspective and achieved via a decomposition of the adjacency matrix (or a transformation thereof). In this way, this is generally a much faster model to fit compared with the others presented in this section. Athreya et al. (2017) also establish consistency results for these estimation procedures, including a result which says, when the true model is an SBM, as the number of vertices $V \to \infty$ the estimated latent positions will be distributed according to a multivariate Gaussian mixture.

A final model of note is what is typically referred to as the exponential (family) random graph model (ERGM). Notice the three previous models outlined above were all conditionally independent, that is, given model parameters (including latent variables), the edges are sampled independently. The ERGM deviates from this slightly, instead modelling the edges of the graph jointly. In particular, this model assumes the probability of observing the graph $\mathcal{G}$ has the following form

$$p(\mathcal{G}|\theta) = \exp\{\theta^{\mathrm{T}} S(\mathcal{G}) - Z(\theta)\}$$

where $S(\mathcal{G}) = (S_1(\mathcal{G}), \ldots, S_K(\mathcal{G}))$ is a vector of practitioner-specified summary statistics, $\theta = (\theta_1, \ldots, \theta_K)$ is a vector of associated parameters, and $Z(\theta)$ is a normalising constant. The $S_i(\mathcal{G})$ could include for example (i) count of edges, (ii) counts of tri-

angles, or even (iii) the degree for a single vertex (with a separate statistic for each vertex). With this, the interpretation of model parameters θ are dependent on the choice of summary statistics. For example, if $S_i(\mathcal{G})$ was the number of triangles, then a high θ_i would imply a higher probability of sampling graphs with many triangles.

As discussed in the review of Salter-Townshend et al. (2012), this is sometimes referred to as the p^* model, building upon the previously proposed p_1 model (Holland and Leinhardt, 1981), which one arrives at by letting the vector of summary statistic include (i) the total number of edges in $\mathcal{G}$, (ii) the in and out degrees for each vertex, and (iii) the total number of reciprocal relations in $\mathcal{G}$ (e.g. the number of mutual friendships).

Inference for the ERGM amounts to estimating the vector of parameters θ. Unfortunately, this is complicated by evaluation of the normalising constant $Z(\theta)$ requiring a summation of $p(\mathcal{G}|\theta)$ over all graphs, which quickly becomes intractable for graphs of even modest size. Nonetheless, there have been both Frequentist and Bayesian approaches proposed which manage to circumvent these issues, including maximum pseudo-likelihood and MCMC-based approaches, respectively (Salter-Townshend et al., 2012, Sec. 4).

2.2.3 Edge-exchangeable models

All the models in the previous section are what is know as *vertex-exchangeable*, whereby vertices are seen as the units of observation. This is arguably sensible in many cases. For example, consider constructing a social network by assessing who is friends with (or follows) who on a social media platform. To gather data would be to sequentially include individuals, that is, vertices. In this context, vertex exchangeability essentially says the order in which individuals are included does not matter; the probability of observing the resulting network will be the same.

More formally, given some permutation σ of the integers 1 to V, if we let $\mathcal{G}^\sigma =$

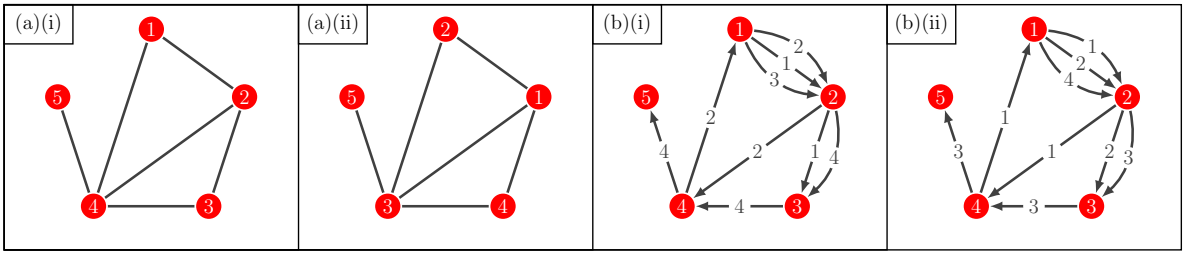


Figure 2.2.1: Comparing vertex and edge exchangeability. In (a) both (i) and (ii) would have equal probability under a vertex-exchangeable model, similarly in (b), where edges are labelled by the order in which they appear, both (i) and (ii) would have equal probability under an edge-exchangeable model. In both (a) and (b) the permutation $\sigma = (12)(34)$ is used, that is, labels 1 and 2 are swapped, as are labels 3 and 4.

$(\mathcal{V}, \mathcal{E}^\sigma)$ where $\mathcal{E}^\sigma = \{(\sigma(u), \sigma(v)) : (u, v) \in \mathcal{E}\}$, denoting the graph obtained by permuting the vertex labels of edges (Figure 2.2.1), then a vertex-exchangeable model assumes $p(\mathcal{G}|\theta) = p(\mathcal{G}^\sigma|\theta^\sigma)$, where θ^σ represents the permutation of model parameters required to ensure congruence with the new vertex labels. For example, with the SBM one would permute the rows of the block membership matrix $Z^\sigma = (\boldsymbol{z}_{\sigma(1)}, \ldots, \boldsymbol{z}_{\sigma(V)})^{\mathrm{T}}$, implying vertices continue to be in the same community though their own label has changed.

However, it is not necessarily always the case that vertices can naturally be seen as the units of observation. For example, rather than observing friendships explicitly one might instead observe physical interactions over time, whereby an edge occurs if two individuals (vertices) interacted in some way. In this way, edges rather than vertices represent the units of observation. This has led to the recent proposal of so-called *edge-exchangeable* models (Cai et al., 2016; Crane and Dempsey, 2018).

The set-up for these models is as follows. For some (possibly infinite) set of vertices $\mathcal{V}$, it is supposed one observes

$$\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$$

where $\mathcal{I}_i = (x_{i1}, \ldots, x_{in_i})$ with $x_{ij} \in \mathcal{V}$ represent paths over vertices, referring to

$\mathcal{S}$ as an interaction sequence, and the $\mathcal{I}_i$ as interactions.[2] For example, Figure 2.2.1 visualises two interaction sequences, wherein edges are numbered according to the interaction in which they appeared.

A statistical model in this context thus amounts to specification of probability distributions $p(\mathcal{S}|\theta)$ over interaction sequences, with θ being model parameters. Such a model is said to be edge-exchangeable if $p(\mathcal{S}|\theta) = p(\mathcal{S}^\sigma|\theta^\sigma)$ where $\mathcal{S}^\sigma = (\mathcal{I}_{\sigma(1)}, \ldots, \mathcal{I}_{\sigma(N)})$ for some permutation σ of the integers 1 to N, whilst θ^σ is again a permutation (if required) of model parameters ensuring congruence with the new interaction labels. For example, in Figure 2.2.1 the two visualised interaction sequences are equal up to a permutation of edge labels via $\sigma = (12)(34)$, and thus would have equal probability under an edge-exchangeable model.

An example of an edge-exchangeable model is the Hollywood (HW) model proposed by Crane and Dempsey (2018), which is based around a central "rich-get-richer" idea. The illustrative example provided therein is that of movie casts (hence the name), whereby one views each interaction $\mathcal{I}_i$ as the cast of a film, with the rich-get-richer concept here translating to an assumption that the actors most likely to appear in a film now are those that often appeared in the past. With $\{\nu_k\}_{k\geq 1}$ a probability distribution over the natural numbers and two model parameters α and θ, the HW model samples $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ by sampling each $\mathcal{I}_i$ in turn via the following procedure

1. Sample the length $n_i \sim \{v_k\}_{k\geq 1}$ of interaction $\mathcal{I}_i$

[2] This notation via a sequence of paths is derived from that used by Crane and Dempsey (2018). The use of paths and not just edges (paths of length two) is a slight generalisation allowing the inclusion of multiple edges in a single step.

2. Sample $\mathcal{I}_i = (x_{i1}, \ldots, x_{in_i})$ by drawing x_{ij} for $j = 1, \ldots, n_i$ from

$$p(x_{ij} = v \,|\, \mathbf{x}_{<ij}) \propto \begin{cases} D_{ij}(v) - \alpha & v = 1, \ldots, V_{ij} \\ \theta + \alpha V_{ij} & v = V_{ij} + 1, \end{cases}$$

where

$$\mathbf{x}_{<ij} = (x_{11}, \ldots, x_{1n_1}, x_{21}, \ldots, x_{(i-1)n_{i-1}}, x_{i1}, \ldots, x_{i(j-1)})$$

denotes all sampled vertices up to but *not* including the jth entry in the ith interaction, $D_{ij}(v)$ denotes the number of times vertex v has appeared up to but not including the jth entry of the ith interaction, and V_{ij} denotes the number of unique vertices seen up to but not including the jth entry of the ith interaction. Here α controls how the probability of sampling a vertex depends on the number of times it has previously appeared, whilst θ controls the probability that a new vertex is sampled. Crane and Dempsey (2018) consider two schemes for the HW model, assuming (i) the vertex set $\mathcal{V}$ is infinite via $0 < \alpha < 1$ and $\theta > -\alpha$, and (ii) the vertex set is finite with $V < \infty$ vertices, via $\alpha < 0$ and $\theta = -\alpha V$. In either case, given this sampling scheme one can derive $p(\mathcal{S}|\theta)$ to show it is invariant to the ordering of the interactions, that is, the HW model is edge-exchangeable. Inference for the HW model is achieved straight forwardly, with Crane and Dempsey (2018) considering a maximum likelihood approach.

Though edge-exchangeable models were introduced above via considerations of observational units, it is worth noting they also have theoretical motivations. In particular, due to the so-called Aldous-Hoover theorem, any vertex-exchangeable model is known to produce graphs which are almost surely dense or empty as the number of vertices $V \to \infty$ (Orbanz and Roy, 2014, Sec. 7), whilst it is known that many observed networks exhibit sparsity. Here density and sparsity regard how many edges are in a network as the number of vertices grow, whereby sparsity occurs when the

number of edges grows sub-quadratically in the number of vertices. In contrast, many of the currently proposed edge-exchangeable models are capable of capturing sparsity (Cai et al., 2016; Crane and Dempsey, 2018; Williamson, 2016). Moreover, this has motivated models employing other forms of exchangeability that also engender sparsity, such as that of Caron and Fox (2017).

Observe also the similarity of an interaction sequence as defined above with the notion of an interaction network that was introduced in Chapter 1 (Figure 1.0.1). As will be detailed in Section 2.4, in this thesis an interaction sequence will indeed be viewed as a representation of an interaction network; though a dual representation thereof will also be considered that disregards the order of interactions.

2.3 Samples of networks

In the previous section, the problem of analysing a single network (or sequence of interactions) was considered. However, there has been a recent focus in the literature on the scenario where one instead observes a *sample* of networks

$$\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(n)}$$

where each $\mathcal{G}^{(i)} = (\mathcal{V}, \mathcal{E}^{(i)})$, that is, a sample of graphs sharing a common vertex set but with possibly different edges. Such data appears frequently in neuroscience, for example, where network representations are used to represent brain scan data (Behrens and Sporns, 2012; Chung et al., 2021). There, vertices correspond to brain regions whilst edges represent some inferred cognitive dependence, and since typically scans are collected for multiple patients in a single study, a sample of graphs is observed.

Notice when faced with a sample of networks the inferential questions that arise differ somewhat from the case where a single network is observed. As seen in Sec-

tion 2.2, with a single network the general focus is on assessing the structure within. In contrast, when faced with a sample of networks, one might instead ask the following

- What is the 'average network'?
- How variable are networks about this average?
- Can we test for differences in two samples of networks?

Notice these are questions one might ask in a standard statistical analysis, only they are being asked of network-valued data. Moreover, they are consistent with those this thesis considers, as discussed in Chapter 1.

Generally speaking, the methods proposed for analysing a single network are not fit to provide such insights. This has motivated a string of recent work on approaches which are; some notable examples of which will now be outlined.

2.3.1 Distances between networks

When faced with a sample of data, of any type, having a notion of distance between data points immediately opens you up to a variety of statistical tools, such as dimension reduction methods like multidimensional scaling, which aid visualisation, or predictive methods like k-nearest neighbours regression. For this reason, the problem of measuring the distance between two graphs has been considered (Donnat and Holmes, 2018; Wills and Meyer, 2020).

Donnat and Holmes (2018) recently reviewed various graph distances and their suitability for the comparison of networks. The simplest distance one can consider is the Hamming distance, define as follows

$$d_{\mathrm{H}}(\mathcal{G}, \mathcal{G}') := \sum_{i<j} |A^{\mathcal{G}}_{ij} - A^{\mathcal{G}'}_{ij}|,$$

whereby one simply counts the number of edges *not* shared between the two graphs. Donnat and Holmes (2018) note, however, that the Hamming distance is inherently local: for each vertex it cares only about its immediate neighbours. It cares not about the wider role of the vertex in the network, or the overall structure of the graph in general.

Given this observation, they proposed various other distances which can, in different ways, capture more global differences between graphs. As an example, they suggest comparing the centrality sequences of each graph via the following

$$d_{\text{centrality}}(\mathcal{G}, \mathcal{G}') = \sqrt{\sum_{v=1}^{V} (c_v - c'_v)^2}$$

where c_v and c'_v represent some centrality measure of the vth vertex in $\mathcal{G}$ and $\mathcal{G}'$, respectively. For example, one could use the closeness centrality $c_v = \text{CL}(v)$, as defined in Section 2.2.1.

2.3.2 Extending single-network models

Notice an analysis based upon the distances of the previous section would generally be model-free, analogous somewhat to the use of network statistics in the single-network case (Section 2.2.1). Towards considering a model-based approach, a natural route is to extend models proposed for a single network. In this section, we will discuss the various work that has been done in this direction regarding the models outlined in Section 2.2.2.

Firstly, the SBM has been extened, with Sweet et al. (2014) assuming a hierarchical model where each observation is drawn from an SBM with its own parameterisation, whilst Stanley et al. (2016) and Reyes and Rodriguez (2016) consider mixtures of SBMs. The LSM has also been extended by Sweet et al. (2013), who assumed a hierarchical model in which each observation is drawn from an LSM with its own param-

eter, with these parameters being linked via a prior, Gollini and Murphy (2016), who assume observations share the same latent coordinates, and Durante et al. (2017), who take a non-parametric approach, using a mixture of LSMs combined with shrinkage priors which induce removal of redundant components and unnecessary dimensions in latent coordinates. Towards extending the RDPG model, Levin et al. (2017) assume observations are drawn i.i.d. from the same RDPG model, whilst Nielsen and Witten (2018), Wang et al. (2019) and Arroyo et al. (2021) consider relaxing this i.i.d. assumption, constructing their models to permit variation in the RDPG parameters across observations, better capturing heterogeneity. Finally, the ERGM framework has similarly been extended, where Lehmann and White (2021) consider a hierarchical model (in similar spirit to Sweet et al., 2013), whilst Yin et al. (2022) consider a finite mixture of ERGMs.

2.3.3 Measurement-error models

In an alternative direction, various works have been proposed which view the sample of observed networks as 'noisy' or perturbed realisations of a single unobserved ground-truth network. The idea is that perhaps differences observed between networks of the sample are due to some natural observational error incurred during data collection. In this way, we refer collectively to such works as measurement-error (ME) models.

As an example, we present a model proposed by Le et al. (2018). Suppose that $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents some ground-truth network. Letting $A^{(1)}, \ldots, A^{(n)}$ denote the adjacency matrices of the observed graphs (dropping the superscript notation here for brevity), this model assumes each $A^{(i)}$ was sampled by randomly perturbing the edges of $\mathcal{G}$, or equivalently, randomly altering the entries of $A^{\mathcal{G}}$, the ground-truth adjacency matrix. This is parameterised by two matrices $P \in [0,1]^{V \times V}$ and $Q \in [0,1]^{V \times V}$ representing false positive and negative probabilities for each edge in the

ground-truth network, that is

$$P_{ij} = \mathbb{P}(A^{(k)}_{ij} = 1 | A^{\mathcal{G}}_{ij} = 0) \quad \text{and} \quad Q_{ij} = \mathbb{P}(A^{(k)}_{ij} = 0 | A^{\mathcal{G}}_{ij} = 1)$$

for each $k = 1, \ldots, n$. With this, they assume, given the ground-truth network $\mathcal{G}$ and parameters P and Q, each edge of each observation $A^{(k)}$ was sampled independently via these probabilities.

Inference in this case becomes estimation of both the ground-truth network $\mathcal{G}$ and the error probability matrices P and Q. Notice the parameter space in general is thus very large, with P and Q for example having V^2 entries each which must be estimated. To reduce the parameter space somewhat, Le et al. (2018) propose to further assume the ground-truth $\mathcal{G}$ was drawn from an SBM, whilst the P and Q have a block structure mirroring that of the ground-truth, that is, all edges allocated to the same block in the model for $\mathcal{G}$ will have the same false positive and negative probabilities. With this, they considered a Frequentist approach to parameter inference, proposing a scheme based around the expectation maximisation (EM) algorithm to estimate the ground-truth network $\mathcal{G}$, including parameters of the SBM it was assumed to be drawn from, and the error matrices P and Q.

As mentioned, others have considered modelling samples of networks in a similar way. Notably, Newman (2018) and Peixoto (2018), who similarly propose models that view observations as perturbations of an unknown ground-truth network, whilst Mantziou et al. (2021) and Young et al. (2022) have considered using the ME modelling approach as a basis for model-based clustering of networks.

2.3.4 Modelling via distances

As a final example of methods proposed to analyse samples of networks, and ones closely related to the work of this thesis, there has been the proposal of modelling

approaches utilising graph distances. Of particular relevance is the modelling framework proposed by Lunagómez et al. (2021). Assuming one has access to a distance between graphs $d_G(\cdot, \cdot)$, they propose to elicit distributions over the space of graphs via location and scale. In particular, given a graph $\mathcal{G}^m$, over the same vertex set $\mathcal{V}$ as the observed networks, and $\gamma > 0$, they assume each graph $\mathcal{G}^{(i)}$ in the sample was drawn independently with the following probability

$$p(\mathcal{G}|\mathcal{G}^m, \gamma) \propto \exp\{-\gamma\phi(d_G(\mathcal{G}, \mathcal{G}^m))\}$$

where $\phi(\cdot)$ is a monotonically increasing function such that $\phi(0) = 0$. Notice this implies the probability of observing $\mathcal{G}$ is highest when $\mathcal{G} = \mathcal{G}^m$, and thus $\mathcal{G}^m$ is also referred to as the *mode*. Notice also the parameter γ, referred to as the *dispersion*, controls how fast the probability of $\mathcal{G}$ decays as its distance from the mode increases. In this way, γ controls the scale of the distribution: when γ is higher the probability is concentrated more around the mode $\mathcal{G}^m$, representing the center of the distribution.

This defines a family of models which they refer to as the Spherical Network Family (SNF). It is worth noting this approach draws inspiration from models proposed outside of the networks literature, notably the Mallows model (Vitelli et al., 2018), which appears in the context of preference learning, and the complex Watson distribution (Mardia and Dryden, 1999), which is used in shape analysis, both of which are similarly defined by combining an exponential kernel with a distance metric between the objects of interest.

Inference for the SNF models amounts to estimation the $\mathcal{G}^m$ and γ, which Lunagómez et al. (2021) approach from a Bayesian perspective. Unfortunately, this is complicated by the normalising constant $Z(\mathcal{G}^m, \gamma) = \sum_{\mathcal{G}} p(\mathcal{G}|\mathcal{G}^m, \gamma)$ being intractable in general, involving a sum over all graphs (as with the ERGMs in Section 2.2.2). Nonetheless, they outline an MCMC algorithm which can be used to sample from the joint posterior. They also note for the special case where $d_G(\cdot, \cdot)$ is taken to be

the Hamming distance the normalising constant can be evaluated in closed form, defining what they call the centered Erdős-Rényi model (CER). With this, standard MCMC algorithms like Metropolis-Hasting (MH) can be used, leading to much faster inference.

It is also worth noting other works which have similarly considered utilising graph distances to model samples of networks. Most notably, Ginestet et al. (2017), who considered the problem of k-sample hypothesis testing for networks, deriving limit results for a specific choice of graph distance, and Josephs et al. (2023), who considered combining graph distances with a Gaussian process to facilitate prediction of network-level covariates.

2.3.5 Time series of graphs

Related to the work discussed above on samples of networks are those proposed to analyse time series of graphs. In this context, one similarly observes a sample of networks $\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(n)}$, only now it is assumed there is an ordering, in the sense that $\mathcal{G}^{(t)}$ is observed before $\mathcal{G}^{(t+1)}$, with t being an integer time index. Crucially, whilst the methods discussed in Sections 2.3.2 to 2.3.4 assumed observed networks were sampled *independently*, here one instead considers the possibility of temporal dependence between the observed networks.

In want of analysing such data, a variety of *dynamic* network models have been proposed (see the review of Kim et al., 2018). In some ways, the motivation here is similar to the other works of this section; extracting an informative summary or insight from the observed sample. However, a key difference is the desire to explore for temporal dependence between graphs, and, in one way or another, all dynamic network models that have been proposed provide some means to extract such insights.

As an example, consider the model proposed by Yang et al. (2011), who extend the SBM, as seen in the single-network case in Section 2.2, by allowing community

memberships to vary over time. More precisely, they assume the V vertices are partitioned into K communities at each point t in time, encoded via $V \times K$ matrices $Z_t = (\boldsymbol{z}_{1t}, \ldots, \boldsymbol{z}_{Vt})^{\mathrm{T}}$ where $\boldsymbol{z}_{it} = (0, \ldots, 0, 1, 0, \ldots, 0)^{\mathrm{T}}$ with $z_{ikt} = 1$ indicating the ith vertex belongs to the kth community at the tth point in time. As with the standard SBM presented in Section 2.2, one also specifies a $K \times K$ matrix B, with $B_{kl} \in (0, 1)$ representing the probability of an edge between the kth and lth communities. Introducing the shorthand notation $A^{(t)} = A^{\mathcal{G}^{(t)}}$ for the adjacency matrix of the tth graph, this is then assumed to be sampled conditional on B and its associated community memberships Z_t as follows

$$A_{ij}^{(t)} \mid B, Z_t \sim \text{Bernoulli}(\boldsymbol{z}_{it}^{\mathrm{T}} B \boldsymbol{z}_{jt}),$$

which is equivalent to the formulation of the SBM presented in Section 2.2, only now the community memberships are time-dependent. With this, given B and the full sequence of community memberships $Z_1, \ldots, Z_n$, one can sample a time series of graphs. To capture temporal dependence, Yang et al. (2011) assume community memberships follow a Markov chain. In particular, given a $K \times K$ transition matrix T, with T_{kl} denoting the probability of a vertex transitioning from the kth to the lth community, and an initial distribution $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, where π_k denotes the probability a vertex is in the kth community at the first time point, it is assumed the probability of a given sequence of community memberships is given by

$$p(Z_1, \ldots, Z_n \mid T, \boldsymbol{\pi}) = p(Z_1 \mid \boldsymbol{\pi}) \prod_{t=2}^{n} p(Z_t \mid Z_{t-1}, T),$$

where

$$p(Z_t \mid Z_{t-1}, T) = \prod_{i=1}^{V} p(\boldsymbol{z}_{it} \mid \boldsymbol{z}_{i(t-1)}, T) = \prod_{i=1}^{V} \left(\prod_{k=1}^{K} \prod_{l=1}^{K} T_{kl}^{z_{ik(t-1)} \cdot z_{ilt}} \right),$$

whilst $p(Z_1 \mid \boldsymbol{\pi}) = \prod_{i=1}^{V} \prod_{k=1}^{K} \pi_k^{z_{ik1}}$. Yang et al. (2011) then proposed an EM algorithm to estimate the model parameters B, T and π, alongside the community memberships $Z_1, \ldots, Z_n$, which are also assumed unknown *a priori*. All together, this model formulation and its associated inference procedure allow one to obtain an interpretable summary of how the graphs evolve temporally, in particular, how the vertices are partitioned into communities at each point in time, and how they move between these communities.

2.4 Samples of interaction networks

This brings us back to the focus of this thesis: analysing samples of interaction networks. As has been shown over the preceding sections, current work has either considered modelling a single interaction network, for example via the edge-exchangeable models (Section 2.2.3), or considered a sample of vertex-observed networks (Section 2.3), whereby each network is represented via a graph. It thus appears methods are yet to be proposed to deal with the scenario in which a sample of *interaction* networks are observed; a gap in the literature which this work intends to address.

2.4.1 Notation

For clarity, the central notation and vernacular that will be adopted throughout the remainder of this thesis will here be outlined. As alluded to already, an interaction network (as introduced in Chapter 1) can be represented via an *interaction sequence*,

that is

$$\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$$

where $\mathcal{I}_i = (x_{i1}, \ldots, x_{in_i})$ are *interactions* (paths), over the set of vertices $\mathcal{V}$, so that $x_{ij} \in \mathcal{V}$ for each entry thereof. Observe $\mathcal{S}$ encodes an ordering of interactions, in the sense $\mathcal{I}_i$ is assumed to have been observed before $\mathcal{I}_{i+1}$. As will be seen throughout this thesis, such information may not always be of interest. As such, we consider also a parallel representation of an interaction network via an *interaction multiset*, denoted as follows

$$\mathcal{E} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$$

where curly braces $\{\}$ are used to signify this is a multiset, so that the labels of interactions imply nothing with regards to the order in which they were observed.[3] Introducing the following notation for the space of interactions

$$\boldsymbol{\mathcal{I}}^* := \bigcup_{k=1}^{\infty} \mathcal{V}^k,$$

containing all paths over the vertex set $\mathcal{V}$, the multiset $\mathcal{E}$ can also be represented via a multiplicity function $m_{\mathcal{E}} : \boldsymbol{\mathcal{I}}^* \to \mathbb{Z}_+$, where $m_{\mathcal{E}}(\mathcal{I})$ denotes the multiplicity of $\mathcal{I}$ in $\mathcal{E}$. This also allows us to define the support of $\mathcal{E}$ in $\boldsymbol{\mathcal{I}}^*$ as follows

$$\text{Supp}(\mathcal{E}) := \{\mathcal{I} \in \boldsymbol{\mathcal{I}}^* : m_{\mathcal{E}}(\mathcal{I}) > 0\},$$

denoting the set of unique interactions in $\mathcal{E}$. As an example, we might have

$$\mathcal{E} = \{(1,2), (1,2,3), (1,2), (2,3)\},$$

[3] Observe this use of $\mathcal{E}$ clashes slightly the notation for the edge set of a graph introduced Section 2.1. However, in the remainder of this thesis this notation will be reserved for the representation of an interaction multiset.

so that

$$m_{\mathcal{E}}((1,2)) = 2 \qquad m_{\mathcal{E}}((1,2,3)) = 1 \qquad m_{\mathcal{E}}((2,3)) = 1$$

whilst $\text{Supp}(\mathcal{E}) = \{(1,2),(1,2,3),(2,3)\}$.

Another key notion is that of the aggregate graph (as in Figure 1.0.1), which collapses a given interaction sequence or multiset into a multigraph summarising observed traversals between vertices. Formally, given interaction sequence $\mathcal{S}$, its aggregate multigraph is denoted $\mathcal{G}_{\mathcal{S}} = (\mathcal{V}, \mathcal{E}_{\mathcal{S}})$ where the multiset $\mathcal{E}_{\mathcal{S}}$ is such that an edge (v,u) appears in $\mathcal{E}_{\mathcal{S}}$ each time $x_{ij} = v$ and $x_{i(j+1)} = u$ for some $1 \leq i \leq N$ and $1 \leq j \leq n_i - 1$. Moreover, observe this definition applies readily to interaction multisets, and we let $\mathcal{G}_{\mathcal{E}}$ similarly denote the multigraph obtained by aggregating the interactions of $\mathcal{E}$, an interaction multiset. We also introduce the notation $A^{\mathcal{S}}$ and $A^{\mathcal{E}}$ for the associated adjacency matrices of these aggregate multigraphs, as introduced in Section 2.1.

Since both interaction sequences and multisets represent collections of interactions among a given vertex set, we will refer to them collectively as 'interaction networks'. In this way, they are seen as two alternative representations thereof, albeit with an interaction sequence containing relatively more information through its encoding of order.

A final point of note regards alternative representations of data in this form, and why the use of interaction networks as presented above is arguably preferable. Instead of representing each interaction as a path $\mathcal{I}_i = (x_{i1}, \ldots, x_{in_i})$ one could in theory collapse this into a graph by aggregating traversals, much like for the aggregate graphs $\mathcal{G}_{\mathcal{S}}$ and $\mathcal{G}_{\mathcal{E}}$ defined above. In particular, one could construct an aggregate multigraph $\mathcal{G}_{\mathcal{I}_i} = (\mathcal{V}, \mathcal{E}_{\mathcal{I}_i})$ from the path $\mathcal{I}_i$ by letting the multiset of edges $\mathcal{E}_{\mathcal{I}_i}$ include an edge (u,v) each time $x_{ij} = v$ and $x_{i(j+1)} = u$ for some $1 \leq j \leq n_i - 1$. One could also drop the edge multiplicities, resulting in a directed graph. This would induce a

representation of an observation as either a sequence or multiset of graphs, that is

$$\mathcal{S} = (\mathcal{G}_{\mathcal{I}_1}, \ldots, \mathcal{G}_{\mathcal{I}_N}) \quad \text{or} \quad \mathcal{E} = \{\mathcal{G}_{\mathcal{I}_1}, \ldots, \mathcal{G}_{\mathcal{I}_N}\},$$

where each $\mathcal{G}_{\mathcal{I}_i}$ might be a graph or multigraph over the shared set of vertices, depending on how one has chosen to aggregate the paths. Indeed, doing so would align with the majority of other works on network data, wherein graph-based representations are often employed. Furthermore, notice a sequence of graphs would be equivalent to a graph time series, which, as was outlined in Section 2.3.5, has already seen attention in the literature; though this would regard the analysis of a single series of graphs, whilst we are concerned with analysing a *sample* of such objects.

However, if the data are truly path-observed, then, aside from being less natural, a representation via graphs as above also has the potential to be less efficient. Firstly, since it is possible for two different paths to aggregate to the same graph, there will be some information lost during this process. Secondly, notice that each graph shares the same set of vertices, which will typically be those appearing at least once in the data. In this way, as more interactions are observed, that is, as N grows, the size of the vertex set $V = |\mathcal{V}|$ would also be expected to grow. Consequently, when representing observations in this manner, their dimension will grow in two ways with the size of a dataset; both in the number of interactions and vertices. Moreover, since graph-based methods often scale in the number of vertices, any approach based upon such a representation is likely to scale poorly. In contrast, for an interaction network, the dimension of interactions will stay fixed as the number of interactions grows. As such, by considering instead a method applicable to interaction networks, there is potential to obtain an approach which scales better to larger datasets.

2.4.2 Example datasets

As justification for the applicability of this work, two example datasets which can be interpreted in this manner will be considered, featuring in the example data analyses of both Chapters 3 and 4. In particular:

1. **In-play football data:** shared by StatsBomb,[4] this dataset contains high-granularity information on events within football matches, such as passes sent and received, tackles made and shots taken. Particularly concerned with pass events, we let $\mathcal{V}$ be the set of player positions, such as "left midfielder" or "left back", constructing interactions $\mathcal{I}_i$ representing series of uninterrupted passes between players (as represented by their position), that is, until the ball was lost or went out of play. With this, a single observation $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ represents all uninterrupted pass sequences of a single team in a single match (Figure 2.4.1a). Moreover, since the dataset has nearly 2,000 matches, each with two teams, this leads to a sample of nearly 4,000 interaction networks;

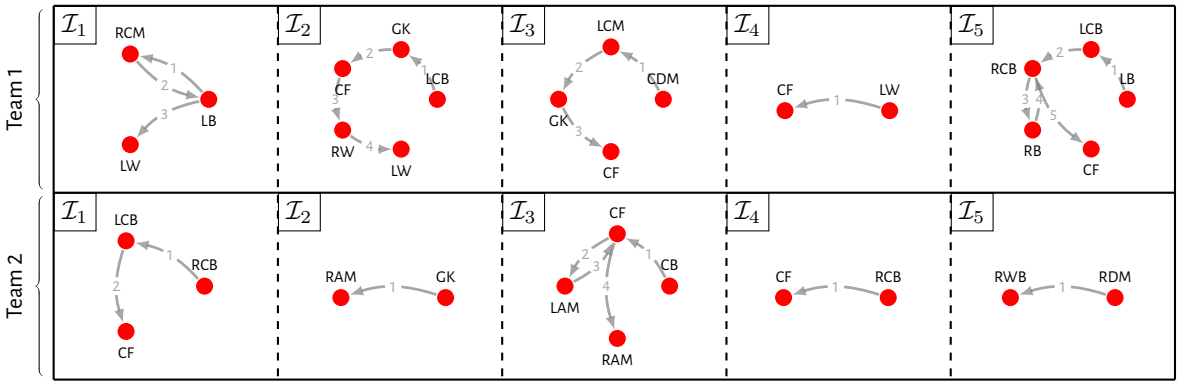
2. **User check-in data:** this is an open-source dataset containing user interactions with the app Foursquare, a location-based social network (LSBN) where users 'check-in' to various venues they visit.[5] By letting $\mathcal{V}$ denote the set of venue categories, an interaction $\mathcal{I}_i$ is here assumed to represent a single day of check-ins for a given user, for example $\mathcal{I}_i = (\text{"Coffee Shop"}, \text{"Work"}, \text{"Restaurant"})$ would imply this user checked-in at venues in these categories in this order. Over some specified time period, a single observation $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ represents all observed days of check-ins for a single user (Figure 2.4.1b). Moreover, since there is data on many users this will lead to a sample of interaction networks. This dataset is much larger than the in-play football data, with the global version (Yang et al., 2015a, 2016), for example, containing check-ins of approx-

[4] https://github.com/statsbomb/open-data
[5] https://sites.google.com/site/yangdingqi/home/foursquare-dataset

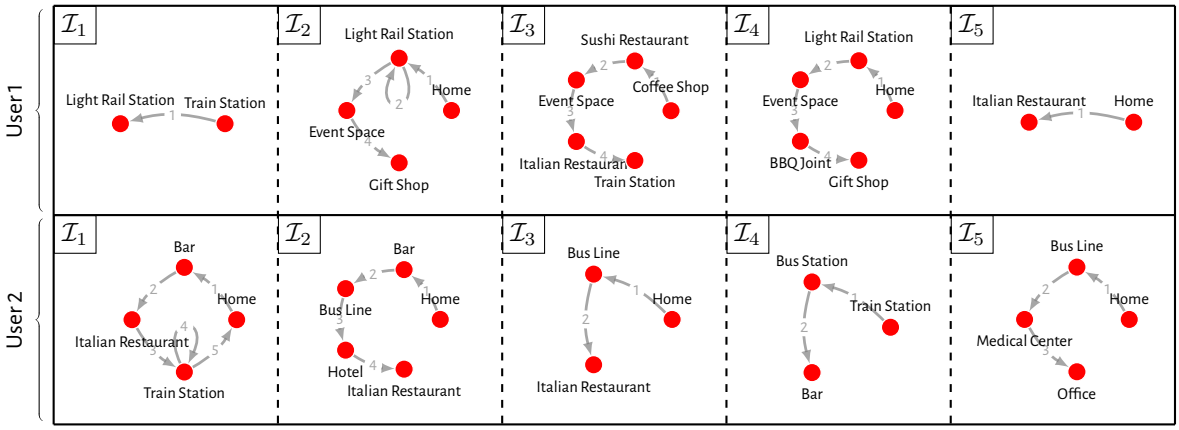
imately 260,000 users.

2.4.3 Thesis outline

The remainder of this thesis will be structured as follows. In Chapter 3, the problem of comparing two interaction networks will be considered, which is then built upon in Chapter 4, where a novel Bayesian modelling framework will be proposed designed specifically for analysing samples of interaction networks. In both, simulations studies are undertaken to further illustrate concepts and confirm the efficacy of proposed methods, whilst example analyses of data introduced in the preceding section are undertaken to highlight practical applications. These two chapters represent the main contributions of this thesis, and thus Chapter 5 will turn to the drawing of conclusions and discussions of future research directions.



(a) In-play football data



(b) Foursquare check-in data

Figure 2.4.1: Examples of interaction network data that will be considered in subsequent chapters. In (a) is shown a sample of interactions from two observations of the StatsBomb in-play football data, where vertices represent player positions (abbreviated according to Table 2.4.1), whilst (b) shows the same for two observations of the Foursquare check-in data, where vertices correspond to venue categories. In both, edges are labelled to indicate the order in which vertices were visited.

Position	Abbreviation	Position	Abbreviation
Secondary Striker	ST	Right Midfield	RM
Center Forward	CF	Center Defensive Midfield	CDM
Left Center Forward	LCF	Left Defensive Midfield	LDM
Right Center Forward	RCF	Right Defensive Midfield	RDM
Center Attacking Midfield	CAM	Left Wing Back	LWB
Right Attacking Midfield	RAM	Right Wing Back	RWB
Left Attacking Midfield	LAM	Center Back	CB
Left Wing	LW	Left Back	LB
Right Wing	RW	Right Back	RB
Center Midfield	CM	Left Center Back	LCB
Left Center Midfield	LCM	Right Center Back	RCB
Right Center Midfield	RCM	Goalkeeper	GK
Left Midfield	LM		

Table 2.4.1: Abbreviations used for player positions in the StatsBomb dataset, as introduced in Section 2.4.2.

Chapter 3

Distances for Comparing Interaction Networks

3.1 Introduction

Given observed data, of any structure, a notion of distance between data points can prove to be an incredibly useful and versatile tool. The case where data points are themselves interaction networks is no exception. Once a distance has been specified, an array of methodologies subsequently become available. These include clustering algorithms such as hierarchical clustering (Izenman, 2008, Sec. 12.3) or HDBSCAN (McInnes et al., 2017), placing networks into groups; dimension reduction or embedding techniques such as multidimensional scaling (MDS) (Kruskal, 1964) or UMAP (McInnes et al., 2018), which can be used to embed networks in Euclidean space, facilitating data visualisation; or predictive algorithms such a k-nearest neighbours regression (Hastie et al., 2009, Sec. 13.3), which can be used to predict network-level covariate information.

As discussed in Section 2.3.1, work has already been done regarding distances between networks with the recent surveys of graph distances (Donnat and Holmes,

2018; Wills and Meyer, 2020). However, it appears no attention has been paid to the comparison of *interaction* networks. With present methods, one would be required to first aggregate these interaction networks to graphs, incurring a potential loss of information.

Motivated by this apparent gap in the literature, this chapter considers different ways in which one might go about measuring the dissimilarity of two interaction networks. This starts with aggregate-based approaches, for example, comparing observations via their aggregate multigraphs. Subsequently, drawing inspiration from areas such as optimal transport and time series analysis, distances which better respect the structure of the data are proposed. Given the dual representation of interaction networks (Section 2.4.1), this reduces to the problem of eliciting distances measures between interaction sequences and multisets. For each distance, theoretical properties are stated and proved, and details regarding computation provided (summarised in Table 3.2.1). Simulation studies are also undertaken, highlighting what certain distances can and cannot capture, and illustrating the possible negative consequences of using an aggregate-based distance over a genuine distance between interaction networks. Finally, through example data analyses it is illustrated how the proposed distances can be used in practice to both cluster networks and predict network-level covariate information, providing answers to questions posed in Chapter 1.

The remainder of this chapter is structured as follows. In Section 3.2, general background on distance measures is provided. In Section 3.3, the approach of comparing interaction networks via their aggregates is then outlined. In Section 3.4, the need for a way to relate interactions of either network is motivated, with two path distances being introduced to serve this purpose. In Sections 3.5 to 3.8, four genuine distances between interaction networks are then proposed, each utilising the path distances of Section 3.4. In particular, Section 3.5 introduces two distances to compare interaction

multisets, with an associated simulation study in Section 3.6, whilst Section 3.7 and Section 3.8 similarly regard the introduction of two interaction sequence distances along with another simulation study. Finally, in Section 3.9 details of example analyses undertaken on data introduced in Section 2.4.2 are provided, before concluding with discussions in Section 3.10.

3.2 Background on distances

A distance measure over the space $\mathscr{X}$ is a function $d : \mathscr{X} \times \mathscr{X} \to \mathbb{R}_+$, taking as input two elements of the space and outputting some measure of dissimilarity between them. It is natural to require that such functions satisfy certain properties, which are formalised mathematically via the notion of a distance metric.

Definition 3.2.1 (Distance metric)**:** A function $d : \mathscr{X} \times \mathscr{X} \to \mathbb{R}_+$ is a *distance metric* over the space $\mathscr{X}$ if, for any $x, y, z \in \mathscr{X}$, the following conditions are satisfied

(i) $d(x, y) = 0 \iff x = y$ (identity of indiscernibles);

(ii) $d(x, y) = d(y, x)$ (symmetry);

(iii) $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality);

with the pair $(\mathscr{X}, d)$ being referred to as a *metric space*.

Notice this work is seeking to elicit distance measures between interaction networks, and thus here $\mathscr{X}$ of Definition 3.2.1 will correspond to the space of interaction networks, that is, the space of all interaction sequences or multisets.

We finalise these background details with discussions regarding distance normalisation. Given a distance metric d over the space $\mathscr{X}$ and some reference element $c \in \mathscr{X}$ of this space, one can transform d to form a new distance $\bar{d}$ as follows

$$\bar{d}(x, y) := \frac{2d(x, y)}{d(x, c) + d(y, c) + d(x, y)},$$

Distance	Notation	Metric Conditions (i)	(ii)	(iii)	Computational Cost
Vector Hamming & Jaccard	d_{H} & d_{J}	✖	✔	✔	$\mathcal{O}(V)$
Graph Hamming & Jaccard	d_{H} & d_{J}	✖	✔	✔	$\mathcal{O}\left(V^2\right)$
Matching distance	$d_{\mathrm{M},\delta(\cdot)}$	✔	✔	✔	$\mathcal{O}\left(N \cdot M + f(N, M)\right)$
Earth mover's distance	d_{EMD}	✖	✔	✔	$\mathcal{O}\left(\tilde{N} \cdot \tilde{M} + f(\tilde{N}, \tilde{M})\right)$
Edit distance	$d_{\mathrm{E},\delta(\cdot)}$	✔	✔	✔	$\mathcal{O}\left(N \cdot M\right)$
Dynamic time warping	d_{DTW}	✖	✔	✖	$\mathcal{O}\left(N \cdot M\right)$

Table 3.2.1: Theoretical properties and computational costs of distances. This concerns comparison of interaction networks with N and M interactions, respectively, and $\tilde{N}$ and $\tilde{M}$ unique interactions, whilst V denotes the size of the assumed vertex set, and, for the matching and EMD distances, the function $f(\cdot, \cdot)$ depends on the solver used.

which will similarly be a distance metric. Note we leave out any reference to c in this notation, though one should be aware that by definition $\bar{d}$ does depend on it. Such a transformation appears in Donnat and Holmes (2018), where it is referred to as the *Steinhaus transform*, and Deza and Deza (2009), referred to as the *biotope transform metric* (Section 4.1 therein). Observe that $\bar{d}(x, y) \geq 0$ for all $x, y \in \mathscr{X}$, being a ratio of non-negative terms. Moreover, since d is a metric, it obeys the triangle inequality (iii), implying

$$\bar{d}(x, y) = \frac{2d(x, y)}{d(x, c) + d(y, c) + d(x, y)} \leq \frac{2d(x, y)}{d(x, y) + d(x, y)} = 1$$

and hence one has $0 \leq \bar{d}(x, y) \leq 1$ for any $x, y \in \mathscr{X}$, justifying the reference to this as a normalised distance. As we will see, this transformation can be very useful when one is trying to compare objects which differ in size.

3.3 Comparing aggregates

Comparison of interaction networks in their raw form, as either sequences or multisets, is non-trivial. This is thanks in most part to requiring a way of relating the

interactions from one network with those of the other, and complicated further by the number of interactions possibly being different. However, a pragmatic solution which circumvents these issues is to first aggregate observations into another more amiable form, using distance between these as a proxy for a distance between the original observations.

There are many ways one could consider aggregating interaction networks. Here, we consider (i) vectors of vertex counts, and (ii) multigraphs of traversals between vertices. These can be seen as encoding first-order and second-order information, respectively. One could in theory go to higher-orders, considering perhaps counts of length $n > 2$ subpaths or subsequences. However, for simplicity we consider just (i) and (ii) in this work.

Given an interaction sequence $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$, with $\mathcal{I}_i = (x_{i1}, \ldots, x_{in_i})$ and $x_{ij} \in \mathcal{V}$ for vertex set $\mathcal{V}$, towards representing (i) we define $v^{\mathcal{S}} \in \mathbb{Z}_{\geq 0}^{V}$ as follows

$$v_x^{\mathcal{S}} := \sum_{i=1}^{N} \sum_{j=1}^{n_i} \mathbb{1}[x_{ij} = x]$$

so that $v_x^{\mathcal{S}}$ denotes the number of times vertex $x \in \mathcal{V}$ appears in $\mathcal{S}$. In a similar way, for (ii) we can encode the number of traversals *between* vertices with the matrix $A^{\mathcal{S}} \in \mathbb{Z}_{\geq 0}^{V \times V}$ as follows

$$A_{xy}^{\mathcal{S}} := \sum_{i=1}^{N} \sum_{j=1}^{n_i - 1} \mathbb{1}[x_{ij} = x] \cdot \mathbb{1}[x_{i(j+1)} = y]$$

so that $A_{xy}^{\mathcal{S}}$ denotes the number of times a traversal from $x \in \mathcal{V}$ to $y \in \mathcal{V}$ was observed in $\mathcal{S}$. Observe this is nothing more than the adjacency matrix of the aggregate multigraph $\mathcal{G}_{\mathcal{S}}$, as defined in Section 2.4.1. Moreover, since both definitions are invariant to the ordering of paths, an interaction multiset $\mathcal{E} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$ can be aggregated in exactly the same way, defining analogues $v^{\mathcal{E}} \in \mathbb{Z}_{\geq 0}^{V}$ and $A^{\mathcal{E}} \in \mathbb{Z}_{\geq 0}^{V \times V}$.

Given two interaction sequences $\mathcal{S}$ and $\mathcal{S}'$, in want of a distance $d(\mathcal{S}, \mathcal{S}')$, we can

use a distance between their aggregates, that is

$$d(\mathcal{S}, \mathcal{S}') = d(v^{\mathcal{S}}, v^{\mathcal{S}'}) \quad \text{or} \quad d(\mathcal{S}, \mathcal{S}') = d(\mathcal{G}_{\mathcal{S}}, \mathcal{G}_{\mathcal{S}'})$$

with $d(v^{\mathcal{S}}, v^{\mathcal{S}'})$ and $d(\mathcal{G}_{\mathcal{S}}, \mathcal{G}_{\mathcal{S}'})$ being distances between vectors of counts and multigraphs, respectively. Natural choices here are to consider analogues of the Hamming and Jaccard distances. In particular, we define the Hamming distances in these cases as follows

$$d_{\mathrm{H}}(v^{\mathcal{S}}, v^{\mathcal{S}'}) := \sum_{x \in \mathcal{V}} |v_x^{\mathcal{S}} - v_x^{\mathcal{S}'}| \qquad d_{\mathrm{H}}(\mathcal{G}_{\mathcal{S}}, \mathcal{G}_{\mathcal{S}'}) := \sum_{x \in \mathcal{V}} \sum_{y \in \mathcal{V}} |A_{xy}^{\mathcal{S}} - A_{xy}^{\mathcal{S}'}|$$

whilst the Jaccard distances are now defined by taking the Steinhaus transform of the Hamming distance, using the zero vector and empty multigraph as reference elements, respectively, leading to the following

$$d_{\mathrm{J}}(v^{\mathcal{S}}, v^{\mathcal{S}'}) := \frac{\sum_{x \in \mathcal{V}} |v_x^{\mathcal{S}} - v_x^{\mathcal{S}'}|}{\sum_{x \in \mathcal{V}} \max(v_x^{\mathcal{S}}, v_x^{\mathcal{S}'})} \qquad d_{\mathrm{J}}(\mathcal{G}_{\mathcal{S}}, \mathcal{G}_{\mathcal{S}'}) := \frac{\sum_{x \in \mathcal{V}} \sum_{y \in \mathcal{V}} |A_{xy}^{\mathcal{S}} - A_{xy}^{\mathcal{S}'}|}{\sum_{x \in \mathcal{V}} \sum_{y \in \mathcal{V}} \max(A_{xy}^{\mathcal{S}}, A_{xy}^{\mathcal{S}'})},$$

a derivation of which can be found in Appendix A.1. Again, notice such distances between interaction multisets $\mathcal{E}$ and $\mathcal{E}'$ can be defined in exactly the same way.

With these, one has a way to measure the dissimilarity of two interaction networks. However, observe the mappings $\mathcal{S} \mapsto v^{\mathcal{S}}$ and $\mathcal{S} \mapsto A^{\mathcal{S}}$ (similarly $\mathcal{E} \mapsto v^{\mathcal{E}}$ and $\mathcal{E} \mapsto A^{\mathcal{E}}$ for multisets) are not injective. As such, one will typically incur a loss of information when aggregating interaction networks in this way. Not only is this wasteful, it has theoretical consequences. Namely, one can have $d(v^{\mathcal{S}}, v^{\mathcal{S}'}) = 0$ or $d(\mathcal{G}_{\mathcal{S}}, \mathcal{G}_{\mathcal{S}'}) = 0$ when $\mathcal{S} \neq \mathcal{S}'$ (similarly $d(v^{\mathcal{E}}, v^{\mathcal{E}'}) = 0$ or $d(\mathcal{G}_{\mathcal{E}}, \mathcal{G}_{\mathcal{E}'}) = 0$ when $\mathcal{E} \neq \mathcal{E}'$ for multisets), leading to a violation of metric condition (i).

In light of this, over the coming sections distances will be proposed which can compare interaction networks directly, without the need for aggregation.

3.4 Comparing interactions

When wanting to elicit a distance between interaction networks, an appealing approach would be to consider how many interactions are shared (or not shared) between them. For example, to compare multisets $\mathcal{E}$ and $\mathcal{E}'$ we might consider

$$d_{\mathrm{H}}(\mathcal{E}, \mathcal{E}') := \sum_{\mathcal{I} \in \boldsymbol{\mathcal{I}}^*} |m_{\mathcal{E}}(\mathcal{I}) - m_{\mathcal{E}'}(\mathcal{I})|,$$

that is, the Hamming distance between multisets $\mathcal{E}$ and $\mathcal{E}'$, which would be a genuine distance metric between interaction multisets. However, this essentially views observations as categorical, whereby interactions are seen as either equal or not. This is arguably crude, as one expects some interactions to be more similar than others. Moreover, one would expect the space of interactions $\boldsymbol{\mathcal{I}}^*$ to be large, making it unlikely $\mathcal{E}$ and $\mathcal{E}'$ will contain *exactly* the same interactions, though their interactions may be quite similar.

This points to a need for eliciting distances between interaction networks which can utilise some form of relational information between interactions, and all the remaining distances that will be introduced will do exactly this. In particular, it will be assumed a distance between interactions is available, that is, $d_I : \boldsymbol{\mathcal{I}}^* \times \boldsymbol{\mathcal{I}}^* \to \mathbb{R}_+$ whereby $d_I(\mathcal{I}, \mathcal{I}')$ measures the dissimilarity of the two interactions $\mathcal{I}$ and $\mathcal{I}'$. In this section, we provide two examples of such distances.

Recalling that we consider interactions as paths, this reduces to the problem of measuring the distance between two paths. Suppose we have two paths

$$\mathcal{I} = (x_1, \ldots, x_n) \quad \text{and} \quad \mathcal{I}' = (y_1, \ldots, y_m)$$

which we would like to compare. As with the Hamming distance above, a natural approach is to consider how much these paths have or do not have in common. In

particular, one can consider common subpaths and subsequences, as illustrated in Figure 3.4.1. A *subpath* of $\mathcal{I}$ from index i to j is given by the following

$$\mathcal{I}_{i:j} = (x_i, \ldots, x_j)$$

where $1 \leq i \leq j \leq n$ (Figure 3.4.1a). More generally, assume that $\boldsymbol{v} = (v_1, \ldots, v_s)$ with $1 \leq v_1 < \cdots < v_s \leq n$, then a *subsequence* of $\mathcal{I}$ is obtained by indexing with $\boldsymbol{v}$ as follows

$$\mathcal{I}_{\boldsymbol{v}} = (x_{v_1}, \ldots, x_{v_s})$$

which will be of length s (Figure 3.4.1b). Given two paths, one can further consider *common* subpaths and subsequences. A common subpath of $\mathcal{I}$ and $\mathcal{I}'$ occurs when we have

$$\mathcal{I}_{i:j} = \mathcal{I}'_{l:k}$$

for some $1 \leq i \leq j \leq n$ and $1 \leq l \leq k \leq m$, whilst a common subsequence of $\mathcal{I}$ and $\mathcal{I}'$ occurs when

$$\mathcal{I}_{\boldsymbol{v}} = \mathcal{I}'_{\boldsymbol{u}}$$

for some $1 \leq v_1 < \cdots < v_s \leq n$ and $1 \leq u_1 < \cdots < u_s \leq m$. The more similar $\mathcal{I}$ and $\mathcal{I}'$ are, the longer we expect their common subpaths or subsequences to be. Following this rationale, a distance can be defined by finding *maximal* common subpaths or subsequences, that is, ones for which there exist none of larger size. This leads to the following

$$d_{\mathrm{LSP}}(\mathcal{I}, \mathcal{I}') := n + m - 2\delta_{\mathrm{LSP}} \quad \text{and} \quad d_{\mathrm{LCS}}(\mathcal{I}, \mathcal{I}') := n + m - 2\delta_{\mathrm{LCS}}$$

where

$$\delta_{\mathrm{LSP}} := \max\{|i:j| = |l:k| \,:\, \mathcal{I}_{i:j} = \mathcal{I}'_{l:k}\} \quad \text{and} \quad \delta_{\mathrm{LCS}} := \max\{|\boldsymbol{v}| = |\boldsymbol{u}| \,:\, \mathcal{I}_{\boldsymbol{v}} = \mathcal{I}'_{\boldsymbol{u}}\}$$

(a) Common subpath (b) Common subsequence

Figure 3.4.1: A comparison of common subpaths and subsequences. In (a) and (b) we see the same pair of paths, with (a) highlighting a common subpath, as indicated by shaded (green) entries, whilst (b) shows a common subsequence. In both cases, these are maximal.

denote the maximum size of a common subpath and subsequence between the two paths. These distances essentially count the number of entries of $\mathcal{I}$ and $\mathcal{I}'$ not included in the common subpath or subsequence. For example, since the subpaths and subsequences in Figure 3.4.1 are maximal, we have $d_{\mathrm{LSP}}(\mathcal{I}, \mathcal{I}') = 7$ and $d_{\mathrm{LCS}}(\mathcal{I}, \mathcal{I}') = 5$ in this case.

Both d_{LCS} and d_{LSP} can be shown to satisfy metric conditions (i) to (iii), making them distance metrics. For a proof, see Appendix A.4.1, whilst for details on how these distances can be computed, see Appendix A.3.1.

3.5 Comparing interaction multisets

In this section, it will be assumed there are two interaction multisets

$$\mathcal{E} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\} \quad \text{and} \quad \mathcal{E}' = \{\mathcal{I}'_1, \ldots, \mathcal{I}'_M\}$$

that are to be compared. Moreover, it is assumed a distance metric $d_I(\cdot, \cdot)$ between interactions has been specified. Over the next two subsections, two distances which can be used in this context will be proposed: the matching distance and the earth mover's distance.

3.5.1 Matching distance

At a high-level, the matching distance seeks the 'best' pairing of interactions from $\mathcal{E}$ with those from $\mathcal{E}'$, in particular, one which minimises the total distance of paired interactions. The distance between $\mathcal{E}$ and $\mathcal{E}'$ is then given by the 'cost' of this best pairing. In this way, the matching distance judges the dissimilarity based upon an optimal relation between the interactions of either multiset.

This idea builds upon distances proposed in the wider literature. In particular, similar distances have been proposed by Ramon and Bruynooghe (2001) and Eiter and Mannila (1997) for the comparison of sets within general metric spaces, though they considered genuine sets whereas we consider multisets. Of these, Ramon and Bruynooghe (2001) is most similar, considering also the notion of a matching, as will now be defined.

Given two multisets $\mathcal{E}$ and $\mathcal{E}'$ a *matching* (Figure 3.5.1a) is simply a multiset of pairs

$$\mathcal{M} = \{(\mathcal{I}, \mathcal{I}') \,:\, \mathcal{I} \in \mathcal{E}, \mathcal{I}' \in \mathcal{E}'\}$$

such that each $\mathcal{I} \in \mathcal{E}$ is matched to at most one $\mathcal{I}' \in \mathcal{E}'$, and *vice versa*. Observe by definition one must have $0 \leq |\mathcal{M}| \leq \min(|\mathcal{E}|, |\mathcal{E}'|)$, that is, we can match at most the number of interactions in the smaller multiset. A matching which achieves this upper bound we say is *complete*. For example, the matching of Figure 3.5.1a is complete. We also define the restriction of $\mathcal{M}$ to $\mathcal{E}$ as follows

$$\mathcal{M}_{\mathcal{E}} := \{\mathcal{I} \in \mathcal{E} \,:\, \exists\, \mathcal{I}' \in \mathcal{E}', \text{ with } (\mathcal{I}, \mathcal{I}') \in \mathcal{M}\}$$

so that $\mathcal{M}_{\mathcal{E}} \subseteq \mathcal{E}$ denotes the elements of $\mathcal{E}$ which are included in the matching $\mathcal{M}$. We also introduce the shorthand $\mathcal{M}^{c}_{\mathcal{E}} := \mathcal{E} \setminus \mathcal{M}_{\mathcal{E}}$ to denote the elements of $\mathcal{E}$ *not* included in the matching $\mathcal{M}$. With this notion, the matching distance is defined as follows.

Definition 3.5.1 (Matching distance)**:** Given a distance $d_I(\cdot, \cdot)$ between interactions

and a penalty $\delta : \boldsymbol{\mathcal{I}}^* \to \mathbb{R}$ for unmatched interactions, the matching distance between $\mathcal{E}$ and $\mathcal{E}'$ is given by

$$d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}') := \min_{\mathcal{M}\in\boldsymbol{\mathcal{M}}(\mathcal{E},\mathcal{E}')} \left\{ \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}} d_I(\mathcal{I},\mathcal{I}') + \sum_{\mathcal{I}\in\mathcal{M}^c_{\mathcal{E}}} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in\mathcal{M}^c_{\mathcal{E}'}} \delta(\mathcal{I}') \right\}$$

where $\boldsymbol{\mathcal{M}}(\mathcal{E},\mathcal{E}')$ denotes the set of matchings between $\mathcal{E}$ and $\mathcal{E}'$.

Notice $d_{\mathrm{M},\delta(\cdot)}$ is defined by finding a matching $\mathcal{M}$ with minimum cost, where the cost of $\mathcal{M}$ consists of (i) distances between matched interactions, and (ii) penalties for the interactions of $\mathcal{E}$ or $\mathcal{E}'$ left unmatched. Given the penalty function satisfies certain conditions, one can show that $d_{\mathrm{M},\delta(\cdot)}$ is a distance metric. We summarise this with the following result, proved in Appendix A.4.2.

Proposition 3.5.2: If $d_I(\cdot,\cdot)$ is a distance metric and the penalty function $\delta(\cdot)$ satisfies

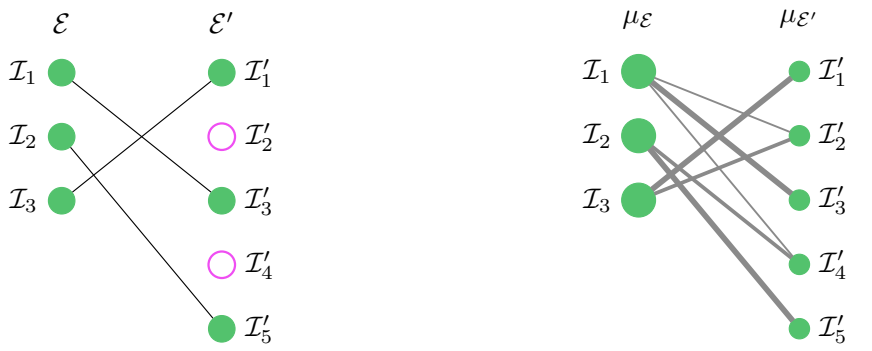
- $\delta(\mathcal{I}) > 0$ for all $\mathcal{I} \in \boldsymbol{\mathcal{I}}^*$, and
- $|\delta(\mathcal{I}) - \delta(\mathcal{I}')| \leq d_I(\mathcal{I},\mathcal{I}')$ for all $\mathcal{I},\mathcal{I}' \in \boldsymbol{\mathcal{I}}^*$

then the distance $d_{\mathrm{M},\delta(\cdot)}$ is a metric between interaction multisets, that is, it satisfies metric conditions (i) to (iii).

Given Proposition 3.5.2, this raises the question of how to specify the penalty function. Two examples which satisfy the required conditions are as follows

1. **Fixed penalty:** let $\delta(\mathcal{I}) = \rho$, where $\rho > 0$ is a chosen constant;
2. **Distance-based penalty:** let $\delta(\mathcal{I}) = d_I(\mathcal{I},\Lambda)$ where Λ is the null interaction.

Note in the case where interactions are paths a natural choice for Λ is the empty path, for which $d_I(\mathcal{I},\Lambda)$ will typically represent the size of $\mathcal{I}$. For example, with the LSP distance one has $d_{\mathrm{LSP}}(\mathcal{I},\Lambda) = n$ where n is the length of $\mathcal{I}$. It is straightforward to show that both penalties satisfy the conditions of Proposition 3.5.2, and consequently



(a) Matching of $\mathcal{E}$ and $\mathcal{E}'$

(b) Coupling of $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{E}'}$

Figure 3.5.1: Example relations found when evaluating multiset distances, with (a) showing a matching $\mathcal{M}$ of the multisets $\mathcal{E} = \{\mathcal{I}_1, \ldots, \mathcal{I}_3\}$ and $\mathcal{E}' = \{\mathcal{I}'_1, \ldots, \mathcal{I}'_5\}$, whilst (b) shows a coupling $\mathbf{P}$ of the distributions $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{E}'}$, where the edge from $\mathcal{I}_i$ to $\mathcal{I}'_j$ is proportional to $\mathbf{P}_{ij}$, the mass moved from $\mathcal{I}_i \in \boldsymbol{\mathcal{I}}^*$ to $\mathcal{I}'_j \in \boldsymbol{\mathcal{I}}^*$, and node (circle) radii of $\mathcal{I}_i$ and $\mathcal{I}'_i$ are proportional to $\mu_{\mathcal{E}}(\mathcal{I}_i)$ and $\mu_{\mathcal{E}'}(\mathcal{I}'_j)$, respectively. For simplicity, here we assume the elements of $\mathcal{E}$ and $\mathcal{E}'$ are distinct, so that within $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{E}'}$ the masses are equal.

both resultant distances will be metrics. For brevity, we introduce the following shorthand for referring to the induced distances: $d_{\mathrm{M},\rho}$ denoting the matching distance with $\delta(\mathcal{I}) = \rho$, and d_{M} denoting the matching distance with $\delta(\mathcal{I}) = d_I(\mathcal{I}, \Lambda)$. Note a slight complication with $d_{\mathrm{M},\rho}$ is the need to specify ρ. As such, in Appendix A.2 we provide guidance on how to set this in practice.

Computation of $d_{\mathrm{M},\delta(\cdot)}$ requires finding an optimal matching. Noting this is essentially an assignment problem, one can appeal to solvers thereof, such as the Hungarian algorithm (Kuhn, 1955). Further details can be found in Appendix A.3.2, where we show how to set-up a suitable assignment problem to be solved. In general, this involves two key elements (i) evaluating all pairwise distances between $\mathcal{E}$ and $\mathcal{E}'$, and then (ii) solving an assignment problem via a chosen solver. With this, the matching distance can be computed with a complexity $\mathcal{O}(N \cdot M + f(N, M))$, where $f(\cdot, \cdot)$ is a solver-dependent term. For example, if optimising over complete matchings via the Hungarian algorithm (see Appendix A.3.2) we will have $f(N, M) = \max(N, M)^3$.

We finish by noting the matching distance as defined in this section can be seen as

a generalisation of the distance proposed by Ramon and Bruynooghe (2001). In particular, whilst they considered a specific choice of penalty for unmatched elements, we have relaxed this, providing conditions on the penalty via Proposition 3.5.2 which ensure the resulting distance continues to be a metric. Moreover, the distance presented here regarded multisets, whereas Ramon and Bruynooghe (2001) considered genuine sets. We also adopt a different approach to computation, recognising this as an assignment problem, whilst Ramon and Bruynooghe (2001) instead propose a network-flow optimisation algorithm to compute their distance.

3.5.2 Earth mover's distance

Though theoretically sound, a drawback of the matching distance is that when $\mathcal{E}$ and $\mathcal{E}'$ are of different sizes the pairwise information of some paths may be ignored. For example, in Figure 3.5.1a the two unmatched paths of $\mathcal{E}'$ are related with nothing from $\mathcal{E}$. However, if these paths were also somewhat similar to those in $\mathcal{E}$, that would surely be useful information to incorporate.

Towards proposing a distance which avoids such issues, one can appeal to the literature on Optimal Transport (OT) (Peyré and Cuturi, 2019), which considers the problem of measuring the distance between probability distributions over general metric spaces. In particular, by converting multisets to distributions, an OT-based distance thereof can serve as a proxy for a distance between the original observations. Importantly, these distances make use of an underlying distance metric.

We convert a multiset $\mathcal{E}$ to a distribution $\mu_{\mathcal{E}} : \boldsymbol{\mathcal{I}}^* \to [0, 1]$ via normalisation as follows

$$\mu_{\mathcal{E}}(\mathcal{I}) := \frac{m_{\mathcal{E}}(\mathcal{I})}{|\mathcal{E}|}, \tag{3.5.1}$$

so that $\mu_{\mathcal{E}}(\mathcal{I})$ denotes the probability mass located at $\mathcal{I} \in \boldsymbol{\mathcal{I}}^*$. To measure the dissimilarity of two multisets $\mathcal{E}$ and $\mathcal{E}'$ we consider using an OT-based distance between $\mu_{\mathcal{E}}$

and $\mu_{\mathcal{E}'}$, namely, the 1-Wasserstein distance (Peyré and Cuturi, 2019, Prop. 2.2), also known as the *earth mover's distance* (EMD).

Viewing $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{E}'}$ as locations of mass within the space $\boldsymbol{\mathcal{I}}^*$, the EMD seeks an optimal transportation of the mass from one set of locations to the other, where the cost of transporting a unit of mass from one element to another is proportional to their pairwise distance. With $\mathrm{Supp}(\mathcal{E}) = \{\mathcal{I}_1, \ldots, \mathcal{I}_{\tilde{N}}\}$ and $\mathrm{Supp}(\mathcal{E}') = \{\mathcal{I}'_1, \ldots, \mathcal{I}'_{\tilde{M}}\}$ the unique paths of $\mathcal{E}$ and $\mathcal{E}'$, we let $\mathbf{P}_{ij} \in [0,1]$ denote the amount of mass to send from $\mathcal{I}_i$ to $\mathcal{I}'_j$ and collate these into the matrix $\mathbf{P} \in [0,1]^{\tilde{N}\times\tilde{M}}$ representing a complete specification of the mass transported between the two sets of locations. Observe $\mathbf{P}$ must satisfy the following constraints

$$\sum_{j=1}^{\tilde{M}} \mathbf{P}_{ij} = \mu_{\mathcal{E}}(\mathcal{I}_i) \quad (\text{for } i = 1, \ldots, \tilde{N}) \quad \text{and} \quad \sum_{i=1}^{\tilde{N}} \mathbf{P}_{ij} = \mu_{\mathcal{E}'}(\mathcal{I}'_j) \quad (\text{for } j = 1, \ldots, \tilde{M})$$

so that if we start with $\mu_{\mathcal{E}}$ and transport mass via $\mathbf{P}$ we end up with $\mu_{\mathcal{E}'}$, and *vice versa*. A matrix $\mathbf{P}$ of this form is known as a *coupling* of $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{E}'}$ (Figure 3.5.1b), and we let $\mathbf{U}(\mu_{\mathcal{E}}, \mu_{\mathcal{E}'})$ denote the set of all couplings between the two distributions, which can be defined as follows. With $\boldsymbol{\mu}_{\mathcal{E}} = (\mu_{\mathcal{E}}(\mathcal{I}_1), \ldots, \mu_{\mathcal{E}}(\mathcal{I}_{\tilde{N}}))^{\mathrm{T}}$ and $\boldsymbol{\mu}_{\mathcal{E}'} = (\mu_{\mathcal{E}'}(\mathcal{I}'_1), \ldots, \mu_{\mathcal{E}'}(\mathcal{I}'_{\tilde{M}}))^{\mathrm{T}}$ denoting vector representations of the distributions $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{E}'}$, respectively, we have

$$\mathbf{U}(\mu_{\mathcal{E}}, \mu_{\mathcal{E}'}) := \{\mathbf{P} \in [0,1]^{\tilde{N}\times\tilde{M}} \,:\, \mathbf{P} \cdot \mathbf{1}_{\tilde{M}} = \boldsymbol{\mu}_{\mathcal{E}},\ \mathbf{1}_{\tilde{N}}^{\mathrm{T}} \cdot \mathbf{P} = \boldsymbol{\mu}_{\mathcal{E}'}\},$$

where $\mathbf{1}_d$ denotes the length d column vector of ones. Collating the required pairwise distances in the matrix $\mathbf{D}$, where $\mathbf{D}_{ij} := d_I(\mathcal{I}_i, \mathcal{I}'_j)$, we define the EMD as follows.

Definition 3.5.3 (Earth mover's distance)**:** Given a distance $d_I(\cdot,\cdot)$ between interactions, the earth mover's distance (EMD) between $\mathcal{E}$ and $\mathcal{E}'$ is given by

$$d_{\mathrm{EMD}}(\mathcal{E}, \mathcal{E}') := \min_{\mathbf{P}\in\mathbf{U}(\mu_{\mathcal{E}}, \mu_{\mathcal{E}'})} \sum_{i=1}^{\tilde{N}} \sum_{j=1}^{\tilde{M}} \mathbf{P}_{ij}\mathbf{D}_{ij}$$

where $\mathbf{P}$ and $\mathbf{D}$ are defined above.

Since the EMD is known to be a distance metric between probability distributions (Peyré and Cuturi, 2019, Prop. 2.2), some properties will be naturally inherited. However, thanks to the normalisation enacted when constructing distributions via eq. (3.5.1), not all of the metric conditions will hold. We summarise this via the following result, proved in Appendix A.4.2.

Proposition 3.5.4: The earth mover's distance d_{EMD} satisfies metric conditions (ii) and (iii), but fails condition (i).

The failure of condition (i) (identity of indiscernibles) occurs when the multisets are proportional to one another, that is, when there is some $C > 0$ such that $m_{\mathcal{E}}(\mathcal{I}) = C \cdot m_{\mathcal{E}'}(\mathcal{I})$ for all $\mathcal{I} \in \boldsymbol{\mathcal{I}}^*$. However, when $\mathcal{E}$ and $\mathcal{E}'$ and the underlying space $\boldsymbol{\mathcal{I}}^*$ are all of reasonable size, the chances of this occurring are likely to be low. As such, the practical consequences are unlikely to be severe; though this will clearly depend on how one intends to use the distance.

Computation of the EMD reduces to solving a linear optimisation problem. Specifically, what is known as the *transportation problem*. As such, one can appeal to literature on solvers thereof. Details can be found in Ch. 3 of Peyré and Cuturi (2019), with packages existing in various programming languages implementing these algorithms, such at the Python Optimal Transport (POT) toolbox (Flamary et al., 2021). Generally, all one needs to do is compute the distance matrix $\mathbf{D}$ and provide this to the chosen solver. As such, the computational cost will be of the form $\mathcal{O}(\tilde{N} \cdot \tilde{M} + f(\tilde{N}, \tilde{M}))$, with $f(\cdot, \cdot)$ being a solver-dependent term.

3.6 Simulation study: multiset distances

In this section, and also later in Section 3.8, we consider using simulation to examine what different distances can and cannot capture. To do so, we parameterise M

different distributions $\mathcal{D}_1, \ldots, \mathcal{D}_M$, where each $\mathcal{D}_i$ can randomly generate multisets or sequences of paths, before assessing whether distances can identify observations sampled from the same distribution. With this, if the $\mathcal{D}_i$ are chosen in a suitable way, deficiencies of certain distances in this regard will highlight an inability capture some qualitative aspect of the observations being compared.

3.6.1 Simulation design

We framed the elicitation of distributions $\mathcal{D}_i$ around the following two questions

- Are there consequences for the information lost through aggregation? (distances of Section 3.3)
- How do distances cope when observations vary in size?

With these in mind, we consider defining such distributions via mixtures over paths. In particular, supposing we have some family of distributions $p(\mathcal{I}|\theta)$ over paths, where θ are model parameters, we parameterise K such distributions $p(\mathcal{I}|\theta_k)$ where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ denotes the K parameters of these distributions, along with the probability vector $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_K)$, where τ_k denotes the probability we sample from the kth path distribution. A single sequence or multiset with N paths is then sampled as follows

- Sample $\boldsymbol{z} \sim \text{Multinomial}(N, \boldsymbol{\tau})$ where $\boldsymbol{z} = (z_1, \ldots, z_K)$
- Construct $\boldsymbol{\theta}^{\boldsymbol{z}}$ as follows

$$\boldsymbol{\theta}^{\boldsymbol{z}} = (\underbrace{\theta_1, \ldots, \theta_1}_{z_1}, \underbrace{\theta_2, \ldots, \theta_2}_{z_2}, \ldots, \underbrace{\theta_k, \ldots, \theta_k}_{z_K})$$

- For $i = 1, \ldots, N$ sample $\mathcal{I}_i$ via $p(\mathcal{I}|\theta_i^{\boldsymbol{z}})$
- We can now obtain either

(a) **Sequence:** $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$

(b) **Multiset:** $\mathcal{E} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$

where in (b) we are essentially disregarding the order.

In this way, this procedure can sample *both* sequences and multisets. When outputting a multiset $\mathcal{E} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$ this can be seen as a sample from a standard mixture distribution $p(\mathcal{I}|\boldsymbol{\theta}, \boldsymbol{\tau}) = \sum_{k=1}^{K} \tau_k p(\mathcal{I}|\theta_k)$ over paths. In contrast, when outputting a sequence $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ the order of $\boldsymbol{\theta}$ is preserved: if $\mathcal{I}_i$ was sampled via $p(\mathcal{I}|\theta_k)$ and $\mathcal{I}_j$ via $p(\mathcal{I}|\theta_l)$ with $i < j$, then one must have $k \leq l$.

To elicit path distributions $p(\mathcal{I}|\theta)$, we consider perturbing a given path with noise. In particular, given some $\tilde{\mathcal{I}} = (\tilde{x}_1, \ldots, \tilde{x}_n)$ we sample $\mathcal{I} = (x_1, \ldots, x_m)$ by randomly deleting and inserting entries from $\tilde{\mathcal{I}}$ such that a subpath is preserved. Given parameters $p_{\text{ins}} \in (0, 1)$ and $p_{\text{del}} \in (0, 1)$, we sample $\mathcal{I}$ by applying the following random edits to $\tilde{\mathcal{I}}$

1. Sample number of deletions and additions[1]

   Deletions: $d \sim \text{TrGeometric}(p_{\text{del}}, 0, n-1)$ **Additions:** $a \sim \text{Geometric}(p_{\text{ins}})$

2. Randomly delete entries as follows

   - Let $d_1 \sim \text{Uniform}\{0, \ldots, d\}$ and let $d_2 := d - d_1$
   - Delete the first d_1 entries and last d_2 entries;

3. Randomly insert entries sampled uniformly from $\mathcal{V}$ as follows

   - Let $a_1 \sim \text{Uniform}\{0, \ldots, a\}$ and let $a_2 := a - a_1$
   - Insert a_1 entries at the front and a_2 entries at the end.

[1]Here $\text{TrGeometric}(p, a, b)$ denotes a truncated Geometric distribution where a and b represent the lower and upper bounds (inclusive) respectively.

By partitioning deletions and insertions in this way notice the sampled path $\mathcal{I}$ will share a subpath with $\tilde{\mathcal{I}}$. For example, if $\tilde{\mathcal{I}} = (1, 2, 3)$ and $(d_1, d_2) = (1, 0)$ and $(a_1, a_2) = (2, 1)$ we might have $\mathcal{I} = (4, 3, 2, 3, 5)$ where here the entries x_1, x_2 and x_5 have been sampled uniformly from $\mathcal{V}$ whilst $\mathcal{I}_{3:4} = \tilde{\mathcal{I}}_{2:3}$, that is, a length 2 subpath has been preserved. Observe this defines a distribution $p(\mathcal{I}|\tilde{\mathcal{I}}, p_{\text{ins}}, p_{\text{del}})$ centered on $\tilde{\mathcal{I}}$ and scaled by p_{inds} and p_{del}, converging to a pointmass at $\tilde{\mathcal{I}}$ as $p_{\text{ins}} \to 1$ and $p_{\text{del}} \to 1$. Finally, note that having separate parameters for insertions and deletions is a desirable since deletions will typically be more destructive, leading to paths that have less in common with $\tilde{\mathcal{I}}$, whereas insertions simply change the size of the resulting paths.

3.6.2 Study and results

In this study, we construct four distributions $\mathcal{D}_1, \ldots, \mathcal{D}_4$ over multisets using the path mixture distribution outlined in Section 3.6.1. The idea is for these mixtures to share the same components (path distributions) but have different mixing proportions. In particular, we fix four paths $\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_4$ denoting four path 'types' around which these components will be centered, as shown in Figure 3.6.1. We then parameterise four components via $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_4)$ where $\theta_k = (\tilde{\mathcal{I}}_k, p_{\text{ins}}, 0.9)$, where p_{ins} is left as a simulation parameter whilst we fix $p_{\text{del}} = 0.9$, the parameter controlling the number of deletions. For the mixing proportions, we consider $\boldsymbol{\tau}^{(1)}, \ldots, \boldsymbol{\tau}^{(4)}$ as visualised in Figure 3.6.1, where $\alpha \in (0.5, 1)$ is a further simulation parameter. Finally, to sample N, the number of paths in each observation, we consider the following

$$N \sim \text{Uniform}\{20 - \nu, 20 + \nu\}$$

leaving ν as a simulation parameter. This leaves us with three simulation parameters (i) ν controlling number of paths in each observation, (ii) p_{ins} controlling the number of insertions sampled when perturbing paths, and (iii) α controlling the entropy of

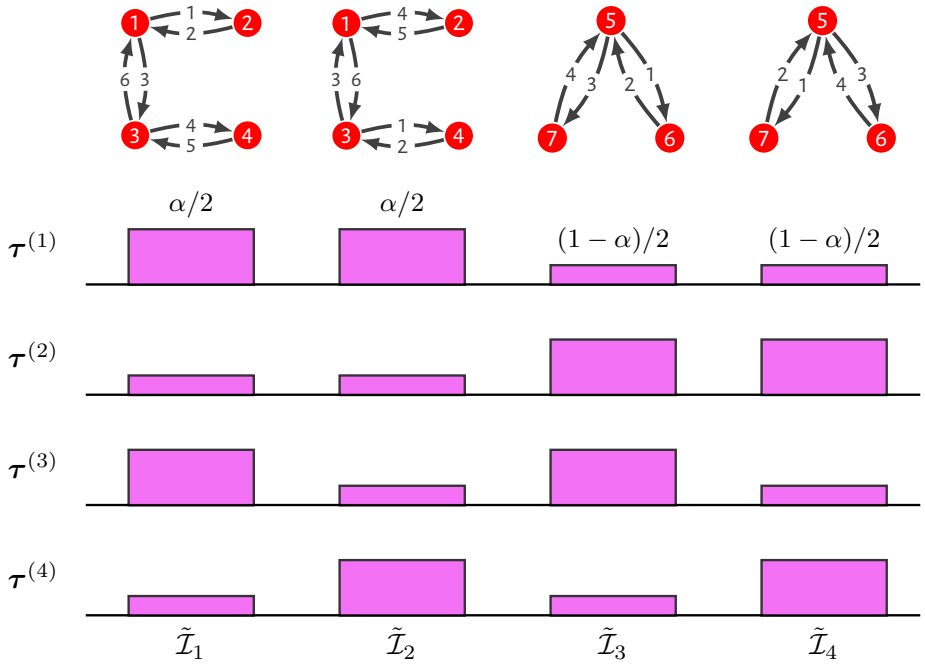


Figure 3.6.1: Visual summary of multiset simulation set-up. Here we visualise the four path types $\tilde{\mathcal{I}}_i$ for $i = 1, \ldots, 4$ (top), whilst below is shown the four different mixture proportion parameters $\boldsymbol{\tau}^{(i)}$ for $i = 1, \ldots, 4$, where the kth bar of $\boldsymbol{\tau}^{(i)}$ (from the left) represents $\tau_k^{(i)}$, the probability of sampling from $p(\mathcal{I}|\theta_k)$, the kth path distribution.

mixture proportions.

The distances we compare are stated in Table 3.6.1, considering also their normalised versions. For clarity, in the remainder of this work, the following naming conventions will be adopted when referring to distances: "n-x" denotes the normalisation of distance x, that is, via the Steinhaus transform (see Section 3.2), for example, "n-Matching" would refer the normalisation of d_{M}. Recall the Steinhaus transform is defined via a reference element of the underlying space. For all distances that will be referred to in this way, the empty element of the space over which the given distance is defined will be used as this reference, that is, for distances between interaction multisets, interaction sequences, or interactions, this will be the empty multiset, sequence or path, respectively. When distances also require specification of a distance between interactions, we signify this via "$x\{y\}$" where x denotes the distance and

Name	Description
Matching	Matching distance d_{M}
FP-Matching	Fixed-penalty matching distance $d_{\mathrm{M},\rho}$ (with $\rho = 0.5$)
EMD	Earth mover's distance d_{EMD}
Graph-Jaccard / Graph-Hamming	Hamming and Jaccard distances d_{H} and d_{J} between aggregate graphs
Vector-Jaccard / Vector-Hamming	Hamming and Jaccard distances d_{H} and d_{J} between aggregate vectors

Table 3.6.1: Distances considered in multiset simulation study (Section 3.6).

y the underlying interaction distance, for example, "Matching{LSP}" would be the matching distance d_{M} with d_{LSP} as the ground distance.

Before discussing results, let us lay out some expectations. Firstly, notice we have chosen the path types $\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_4$ (Figure 3.6.1) such that $\tilde{\mathcal{I}}_1$ and $\tilde{\mathcal{I}}_2$ are similar, and likewise $\tilde{\mathcal{I}}_3$ and $\tilde{\mathcal{I}}_4$ are similar. In particular, they are permutations of one another. As such, we expect an aggregate-based distance to be unable to make this distinction, finding $\mathcal{D}_3$ and $\mathcal{D}_4$ hard to distinguish. We also expect the EMD to fare better than the matching distances when ν is higher, since it uses all pairwise information of paths, whereas the matching distance ignores many unmatched paths. Similarly, we expect normalised distances to perform better than their un-normalised counterparts when ν is higher. Finally, distances utilising a normalised distance between interactions are likely to perform better when the path lengths are more variable, that is, when p_{ins} is lower.

Now, given a set of parameters $(\nu, p_{\mathrm{ins}}, \alpha)$ and a chosen distance, in a single simulation repetition we did the following: (i) sampled $n = 50$ observations from $\mathcal{D}_i$ for $i = 1, \ldots, 4$, leading to $4n = 200$ observations in total (ii) computed distance matrix $\mathbf{D} \in \mathbb{R}_{\geq 0}^{4n \times 4n}$, where $\mathbf{D}_{ij}$ denotes the distance between the ith and jth observation.

For a qualitative examination, given a computed matrix $\mathbf{D}$, one can consider using a dimension reduction algorithm to facilitate visualisation. These algorithms return

$(\boldsymbol{x}_i)_{i=1}^{4n}$ where $\boldsymbol{x}_i \in \mathbb{R}^d$ represents (for the ith data point) a location in Euclidean space, with the idea being that the pairwise distances or relative structure of the embedded points $(\boldsymbol{x}_i)_{i=1}^{4n}$ is in some way congruent with that of $\mathbf{D}$, the observed pairwise distances. Crucially, when $d = 2$ one can plot such embeddings, providing a visual summary of the overall structure in the data. Such algorithms generally fall into two categories: those which preserve *all* pairwise distances, that is, more global structure, such as MDS (Kruskal, 1964) or PCA (Hotelling, 1933), and those which favour preservation of more local structure, that is, distances of a given data point to its closest neighbours, with examples being t-SNE (Van der Maaten and Hinton, 2008; Van Der Maaten, 2014) and UMAP (McInnes et al., 2018). We opted to use UMAP, which, at a high-level, produces an embedding by optimising the layout of a k-nearest neighbours graph in Euclidean space. In Figure 3.6.2, we show the UMAP embeddings resulting from a single simulation run with $(\nu, p_{\text{ins}}, \alpha) = (5, 0.7, 0.9)$ for four different distances. Here one can observe both the matching and EMD distances clearly separate samples from the four distributions, whilst the aggregate-based distances struggle to distinguish those from $\mathcal{D}_3$ and $\mathcal{D}_4$, as expected.

To examine what might happen when we vary certain simulation parameters, we need a single quantitative summary that can be applied to each distance matrix $\mathbf{D}$. We propose that, in the ideal scenario, almost all of the $(n-1)$-nearest neighbours of a given observation should be sampled from its distribution. With this in mind, we consider finding the mean proportion of k-nearest neighbours in the same group, that is, letting $c_i = j$ if the ith data point was drawn from $\mathcal{D}_j$, and $\mathcal{N}_k^i$ denote the indices of the k-nearest neighbours to the ith data point, with m data points in total we consider the following

$$k\text{-MNP} := \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{k}\sum_{j\in\mathcal{N}_k^i}\mathbb{1}(c_i = c_j)\right),$$

referring to this as the k mean neighbourhood proportion (k-MNP). Thus, in our scenario, we take $(n-1)$-MNP for each distance as our performance measure.

In Figure 3.6.3 we summarise the $(n-1)$-MNP values (averaged over 25 repetitions) for different distances with the simulation parameters ν and p_{ins} varying and $\alpha = 0.9$ fixed. Here one can see the aggregate-based distances in general perform poorly relative multiset-based distances, as expected. Moreover, in all cases performance deteriorates as observations become more variable size, that is, as ν grows. However, the EMD does appear to be more robust in this regard, showing less deterioration, as expected. In addition, there appears evidence that normalised distances indeed tend to perform better than their un-normalised counterparts. Finally, in general distances using the normalised LSP between interactions performed better, with this gap widening as $p_{\text{ins}} \to 0$, that is, as paths become more variable in size, again meeting expectations.

To summarise, through this study we have shown the information loss incurred through aggregation can be detrimental for distinguishing observations. Moreover, if looking to compare multisets of quite variable sizes the EMD appears to be a good choice. Finally, where normalisation is possible it is usually helpful, particularly when comparing objects of very different size.

3.7 Comparing interaction sequences

In this section, we consider the scenario where we have two interaction sequences

$$\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N) \quad \text{and} \quad \mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_M)$$

which we would like to compare. Again, it will be assumed that some distance metric $d_I(\cdot, \cdot)$ between interactions is available. Over the next two subsections, two distances will be proposed that are applicable in this case: the edit distance and the dynamic

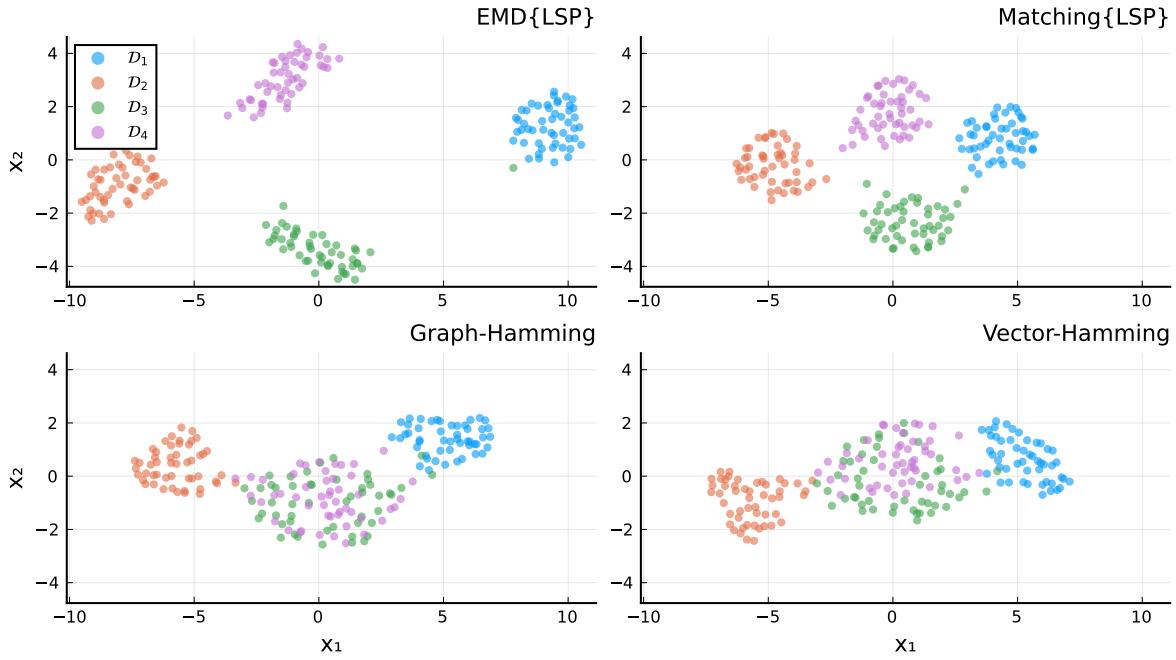


Figure 3.6.2: UMAP embeddings for a single multiset simulation run. Here one can observe the two multiset distances clearly separate samples from the four distributions, whilst aggregate-based distances struggle to distinguish $\mathcal{D}_3$ and $\mathcal{D}_4$.

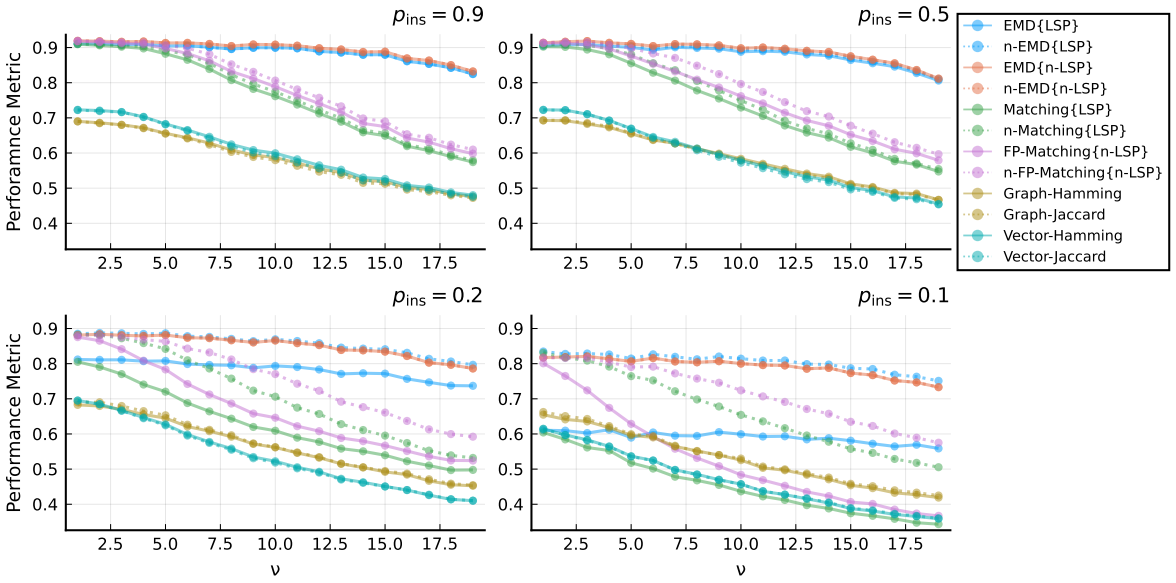


Figure 3.6.3: Comparing performance of distances at identifying samples from the same distribution as simulation parameters ν and p_{ins} are varied. Note here a solid and dashed line of the same color regard the standard and normalised versions of a given distance.

time warping distance.

3.7.1 Edit distance

The edit distance, much like the matching distance seen in Section 3.5.1, is defined via an optimal pairing between interactions of either sequence, that is, one which minimises the total distance of paired interactions plus a penalty. However, in distinction from the matching distance, the edit distance must take the ordering of the sequences into account. As such, it considers a slightly altered form of pairing: a *monotone* matching.

Observe the notion of a matching, as introduced in Section 3.5.1 for multisets, continues to make sense for sequences. In particular, a matching between $\mathcal{S}$ and $\mathcal{S}'$ is a multiset of pairs $\mathcal{M} = \{(\mathcal{I}, \mathcal{I}') : \mathcal{I} \in \mathcal{S}, \mathcal{I}' \in \mathcal{S}'\}$ such that each entry of either sequence it paired with at most one from the other. Since sequences have an ordering we can also consider whether this ordering is preserved by the matching. In particular, we say $\mathcal{M}$ is monotone if for any $(\mathcal{I}_{i_1}, \mathcal{I}'_{j_1}) \in \mathcal{M}$ and $(\mathcal{I}_{i_2}, \mathcal{I}'_{j_2}) \in \mathcal{M}$ we have

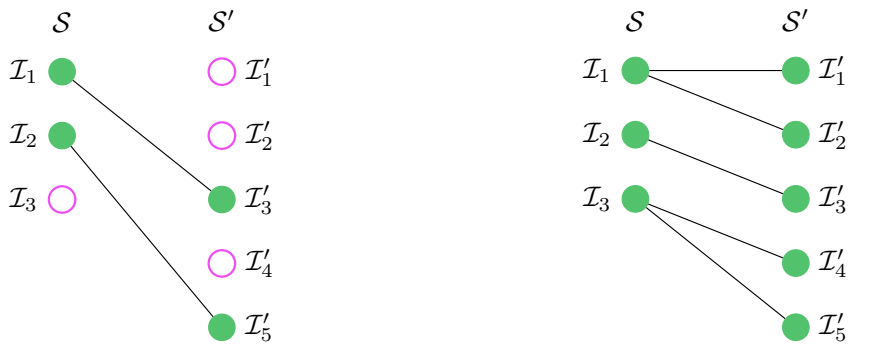
$$i_1 < i_2 \iff j_1 < j_2$$

which intuitively means no lines cross when one draws the matching (Figure 3.7.1a). With this, the edit distance is defined as follows.

Definition 3.7.1 (Edit distance)**:** Given a distance $d_I(\cdot, \cdot)$ between interactions and a penalty $\delta : \boldsymbol{\mathcal{I}}^* \to \mathbb{R}$ for unmatched interactions, the edit distance between $\mathcal{S}$ and $\mathcal{S}'$ is given by

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}') := \min_{\mathcal{M} \in \boldsymbol{\mathcal{M}}_m(\mathcal{S}, \mathcal{S}')} \left\{ \sum_{(\mathcal{I}, \mathcal{I}') \in \mathcal{M}} d_I(\mathcal{I}, \mathcal{I}') + \sum_{\mathcal{I} \in \mathcal{M}^c_{\mathcal{E}}} \delta(\mathcal{I}) + \sum_{\mathcal{I}' \in \mathcal{M}^c_{\mathcal{E}'}} \delta(\mathcal{I}') \right\}$$

where $\boldsymbol{\mathcal{M}}_m(\mathcal{S}, \mathcal{S}')$ denotes the set of monotone matchings of $\mathcal{S}$ and $\mathcal{S}'$.



(a) Montone matching (b) Sequence coupling

Figure 3.7.1: Example relations used to define sequence distances, where (a) shows an example of a monotone matching of the two sequences $\mathcal{S}$ and $\mathcal{S}'$, used to define the edit distance, whilst (b) shows a coupling, used to define the dynamic time warping distance.

Notice Definition 3.7.1 is more-or-less identical to Definition 3.5.1; the only difference being that $\mathcal{M}$, the matching over which one is optimising, must be monotone. Provided the penalty function satisfies certain conditions, one can show $d_{\mathrm{E},\delta(\cdot)}$ will be a distance metric, as summarised via the following result (proved in Appendix A.4.3).

Proposition 3.7.2: If $d_I(\cdot,\cdot)$ is a distance metric and the penalty function $\delta(\cdot)$ satisfies

- $\delta(\mathcal{I}) > 0$ for all $\mathcal{I} \in \boldsymbol{\mathcal{I}}^*$, and
- $|\delta(\mathcal{I}) - \delta(\mathcal{I}')| \leq d_I(\mathcal{I},\mathcal{I}')$ for all $\mathcal{I},\mathcal{I}' \in \boldsymbol{\mathcal{I}}^*$

then the distance $d_{\mathrm{E},\delta(\cdot)}$ is a metric between interaction sequences, that is, it satisfies metric conditions (i) to (iii).

Regarding choices for the penalty function, we propose to reuse those of Section 3.5.1 for the matching distance. Moreover, we introduce analogous short-hand notation: $d_{\mathrm{E},\rho}$ denoting the edit distance with $\delta(\mathcal{I}) = \rho$ for constant $\rho > 0$, and d_{E} denoting the edit distance with $\delta(\mathcal{I}) = d_I(\mathcal{I},\Lambda)$. For guidance on choosing ρ in practice, see Appendix A.2.

As with the matching distance, computation of $d_{\mathrm{E},\delta(\cdot)}$ requires solving an optimisation problem. However, in this case the task is slightly less computationally costly, being possible via dynamic programming at a complexity of $\mathcal{O}(N \cdot M)$. For further details, see Appendix A.3.3.

We finalise these details by noting the edit distance as presented here has close connections with those appearing elsewhere in the literature. First and foremost, the edit distance can be seen as an adaptation of the so-called string edit distance proposed by Wagner and Fischer (1974) (though our presentation via monotone matchings does differ slightly). This string edit distance was originally proposed to compare categorical sequences, but has since be applied in other contexts. Most notably, with the geometric edit distance (Gold and Sharir, 2018; Fox and Li, 2019), which adapts the string edit distance for the comparison of time series. Finally, notice the close connections between the edit distance (Definition 3.7.1) and matching distance (Definition 3.5.1), with latter essentially an unordered version of the former. As far as we are aware, this close connection has not been noted in any other applications.

3.7.2 Dynamic time warping

Though the edit distance has the theoretical benefit of being a distance metric, it suffers from the same drawback as the matching distance: when comparing sequences of different size the pairwise information of some interactions may be ignored. For example, in Figure 3.7.1a we see four unmatched interactions which will be unrelated to any interactions from the other sequence. This similarly motivates the need for a distance without such a feature. As a potential solution, one can consider adapting another distance often seen in the time series literature: the dynamic time warping (DTW) distance (Gold and Sharir, 2018).

Like the edit distance, the DTW distance is based upon finding a minimum cost relation between the two sequences. The DTW, however, considers a so-called *cou-*

pling of the two sequences (Figure 3.7.1b).[2] A coupling of $\mathcal{S}$ and $\mathcal{S}'$ is a sequence of pairs $\mathcal{C} = (p_1, \ldots, p_R)$, where each $p_r = (\mathcal{I}_i, \mathcal{I}'_j)$ for some with $1 \leq i \leq N$ and $1 \leq j \leq M$. To be a coupling, $\mathcal{C}$ must have the first and last entries paired together, that is $p_1 = (\mathcal{I}_1, \mathcal{I}'_1)$ and $p_R = (\mathcal{I}_N, \mathcal{I}'_M)$, and must satisfy the following

$$p_r = (\mathcal{I}_i, \mathcal{I}'_j) \implies p_{r+1} \in \{(\mathcal{I}_i, \mathcal{I}'_{j+1}), (\mathcal{I}_{i+1}, \mathcal{I}'_j), (\mathcal{I}_{i+1}, \mathcal{I}'_{j+1})\},$$

that is, given $\mathcal{I}_i$ and $\mathcal{I}'_j$ are paired, one can either (i) pair the next two entries $\mathcal{I}_{i+1}$ and $\mathcal{I}'_{j+1}$, or (ii) enact some *warping*, where an entry from either sequence is paired with more than one from the other. For example, in Figure 3.7.1b we see warping for the first and third entries of $\mathcal{S}$. Notice that, in contrast with the edit distance, by definition every interaction of one sequence will always be coupled with at least one interaction from the other. With this, the DTW distance is defined as follows.

Definition 3.7.3: Given a distance $d_I(\cdot, \cdot)$ between interactions, the dynamic time warping distance between $\mathcal{S}$ and $\mathcal{S}'$ is given by the following

$$d_{\mathrm{DTW}}(\mathcal{S}, \mathcal{S}') := \min_{\mathcal{C} \in \boldsymbol{\mathcal{C}}(\mathcal{S}, \mathcal{S}')} \left\{ \sum_{(\mathcal{I}, \mathcal{I}') \in \mathcal{C}} d_I(\mathcal{I}, \mathcal{I}') \right\}$$

where $\boldsymbol{\mathcal{C}}(\mathcal{S}, \mathcal{S}')$ denotes the set of couplings between the sequences $\mathcal{S}$ and $\mathcal{S}'$.

Observe, in similar spirit to the edit distance, d_{DTW} is defined by finding a coupling $\mathcal{C}$ with minimum cost, where the cost of $\mathcal{C}$ is defined by summing the pairwise distances of coupled interactions. Unfortunately, the DTW distance has certain theoretical shortcomings. Namely, it violates the identity of indiscernibles (i) and the triangle inequality (iii). This we summarise with the following result, a proof of which can be found in Appendix A.4.

[2]Note this sequence-based coupling differs from the coupling of distributions used to define the EMD (Section 3.5.2).

Proposition 3.7.4: The dynamic time warping distance d_{DTW} satisfies metric condition (ii) (symmetry), but violates conditions (i) (identity of indiscernibles) and (iii) (triangle inequality).

Regarding computation, as with the edit distance, the DTW distance can be evaluated via dynamic programming at a time complexity of $\mathcal{O}(N \cdot M)$. In fact, the algorithm is almost identical to that used to evaluate the edit distance. Further details can be found in Appendix A.3.4.

3.8 Simulation study: sequence distances

For this study, we take the set-up used for the multiset simulation and slightly alter the distributions being sampled from. In particular, we consider again four distributions $\mathcal{D}_1, \ldots, \mathcal{D}_4$ parameterised by $\boldsymbol{\theta}$ and $\boldsymbol{\tau}^{(1)}, \boldsymbol{\tau}^{(2)}, \boldsymbol{\tau}^{(3)}, \boldsymbol{\tau}^{(4)}$ as defined in Section 3.6.2, but in this case we sample sequences. Recall this implies the ordering of parameters $\boldsymbol{\theta}$ and $\boldsymbol{\tau}^{(i)}$ will be reflected in the observations sampled from $\mathcal{D}_i$. We also augment these with four distributions $\mathcal{D}_1^\sigma, \ldots, \mathcal{D}_4^\sigma$ defined by applying the permutation σ to the parameters of each distribution, that is, $\mathcal{D}_i^\sigma$ is parameterised by

$$\boldsymbol{\theta}_\sigma = \left(\theta_{\sigma(1)}, \ldots, \theta_{\sigma(4)}\right) \quad \text{and} \quad \boldsymbol{\tau}_\sigma^{(i)} = \left(\tau_{\sigma(1)}^{(i)}, \ldots, \tau_{\sigma(4)}^{(i)}\right),$$

where we consider in particular the permutation reversing order, that is, $\sigma(i) = 4 - i + 1$ in this case, or equivalently in cyclic notation $\sigma = (14)(23)$. Observe with this we expect $\mathcal{D}_i$ and $\mathcal{D}_i^\sigma$ to sample sequences with similar paths, and in similar proportions, but in a reversed order.

The distances we will compare in this case are shown in Table 3.8.1. Given observations of the previous simulation (Section 3.6.2), we again expect the aggregate distances to perform poorly relative to those comparing the complete sequence of interactions. Since DTW is capable or relating a single interaction of one sequence with

Name	Description
Edit	Edit distance d_{E}
FP-Edit	Fixed-penalty edit distance $d_{\mathrm{E},\rho}$ (with $\rho = 0.5$)
DTW	Dynamic time warping distance d_{DTW}
Graph-Jaccard / Graph-Hamming	Hamming and Jaccard distances d_{H} and d_{J} between aggregate graphs
Vector-Jaccard / Vector-Hamming	Hamming and Jaccard distances d_{H} and d_{J} between aggregate vectors

Table 3.8.1: Distances considered in sequences simulation study (Section 3.8).

more than one from another, one might also expect it to perform better than the edit distance when objects are of quite different size, as with the EMD in Section 3.6.2. Finally, we anticipate distances utilising a normalised interaction distance to fare better when paths are more variable in size, that is, when p_{ins} is smaller.

As in Section 3.6.2, we first examine qualitatively some resultant embeddings of a single simulation scenario. In particular, Figure 3.8.1 shows the UMAP embeddings resulting from a single simulation run with $(\nu, p_{\mathrm{ins}}, \alpha) = (5, 0.7, 0.9)$ for six different distances. Note here we include also two multiset distances (matching and EMD) to highlight how the sequence distances take the order within observations into account.[3] With this, one can observe the sequence distances successfully distinguish the eight distributions, whilst since both the multiset distances and aggregate-based distances are invariant to the order of paths, they do not distinguish $\mathcal{D}_i^\sigma$ from $\mathcal{D}_i$. Moreover, as in the study of Section 3.6.2, the aggregate-based distances continue to confuse samples from $\mathcal{D}_3$ and $\mathcal{D}_3^\sigma$ with those from $\mathcal{D}_4$ and $\mathcal{D}_4^\sigma$.

As in Section 3.6.2, we again ran a cross-sectional study, varying ν and p_{ins} whilst $\alpha = 0.9$ was fixed. Figure 3.8.2 summarises the $(n-1)$-MNP values (averaged over 25 repetitions) for a variety of sequence distances plus some aggregate-based distances. As expected, we again see that distances between aggregates perform the worst (though it does outperform the DTW in some cases, as we will discuss). More-

[3] In computing the matching and EMD distances in this case, we first convert sequences to multisets.

over, normalised distances generally perform better, with this being particularly evident when p_{ins} is low. Notice also of the edit distances, the normalisation of $d_{\text{E},\rho}$ performs best, with is improvement relative to the normalisation of d_{E} growing as p_{ins} decreases, highlighting the benefits of using a normalised interaction distance when paths are more variable in size.

However, the performance of the DTW in this study is somewhat curious and appears to contradict entering expectations. Though its performance matches the edit distance when p_{ins} is lower, its various versions perform very badly when p_{ins} is high, that is, when there is very little noise in the path distributions. This is a consequence of the simulation design and how the DTW treats sequences with the same paths but in different proportions. Consider the following two interaction sequences

$$\mathcal{S} = (\tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2, \tilde{\mathcal{I}}_3, \tilde{\mathcal{I}}_3, \tilde{\mathcal{I}}_4) \qquad \mathcal{S}' = (\tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2, \tilde{\mathcal{I}}_2, \tilde{\mathcal{I}}_3, \tilde{\mathcal{I}}_4, \tilde{\mathcal{I}}_4),$$

where $\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_4$ are the paths used to parameterise the distributions used in the simulation (Figure 3.6.1). Notice these contain paths in the same order but in different proportions, that is, in $\mathcal{S}$ the first and third paths are more prevalent, whilst in $\mathcal{S}'$ the same can be said for the second and fourth. Moreover, these are samples one might expect from $\mathcal{D}_3$ and $\mathcal{D}_4$, respectively, if $p_{\text{ins}} \approx 1$ and $p_{\text{del}} \approx 1$, that is, if there was almost no noise in the path distributions. Now, observe one can construct a coupling between $\mathcal{S}$ and $\mathcal{S}'$ wherein only paths that are equal are paired, implying $d_{\text{DTW}}(\mathcal{S}, \mathcal{S}') = 0$. This highlights a key feature of the DTW: it views $\mathcal{S}$ and $\mathcal{S}'$ as the same sequence of paths $\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_4$ being visited out of sync. In this way, due to the simulation set-up, with very little noise at the level of paths it is likely to see samples from $\mathcal{D}_1, \ldots, \mathcal{D}_4$ as the same. It will, however distinguish these from $\mathcal{D}_1^\sigma, \ldots, \mathcal{D}_4^\sigma$, hence the stable performance around 0.5 when $p_{\text{ins}} = 0.9$ in Figure 3.8.2. Since the cost of coupling is determined by the distance of interactions it pairs together (see Definition 3.7.3), as the level of noise increases the warping required between such

out-of-sync sequences becomes more costly. With this, it appears an increase in the noise at the level of paths allows the DTW to identify sequences sampled from the same distribution.

To summarise, this study has again emphasised the potential costs of disregarding information by use of an aggregate-based distance. It has also highlighted that of the edit distances, the fixed-penalty combined with a normalised interaction distance appears to perform best. Moreover, without normalisation some distances can perform very badly at identifying sequences from the same distribution when they vary significantly in size. Finally, for this particular scenario, the DTW struggled to distinguish some observations, highlighting in particular its treatment of observations containing similar paths, in the same order, but in different proportions.

3.9 Data analysis

In this section, the applicability of distances discussed over the preceding sections will be illustrated via an analysis of the two example datasets: (i) the StatsBomb in-play football data, and (ii) the Foursquare user check-in data. In particular, through a cluster analysis of the in-play football data we show how distances can be used for unsupervised learning purposes, whilst with the Foursquare data we consider using distances as a predictive tool, assessing their efficacy at predicting the country of a user given only their previous check-in information.

3.9.1 In-play football data

Recall here a single interaction $\mathcal{I}_i = (x_{i1}, \ldots, x_{in_i})$ represents a series of uninterrupted passes, with $x_{ij} \in \mathcal{V}$ where $\mathcal{V}$ denotes the set of all player positions. With this, a single data point is either a sequence $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ or multiset $\mathcal{E} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$, representing a full match for a single team.

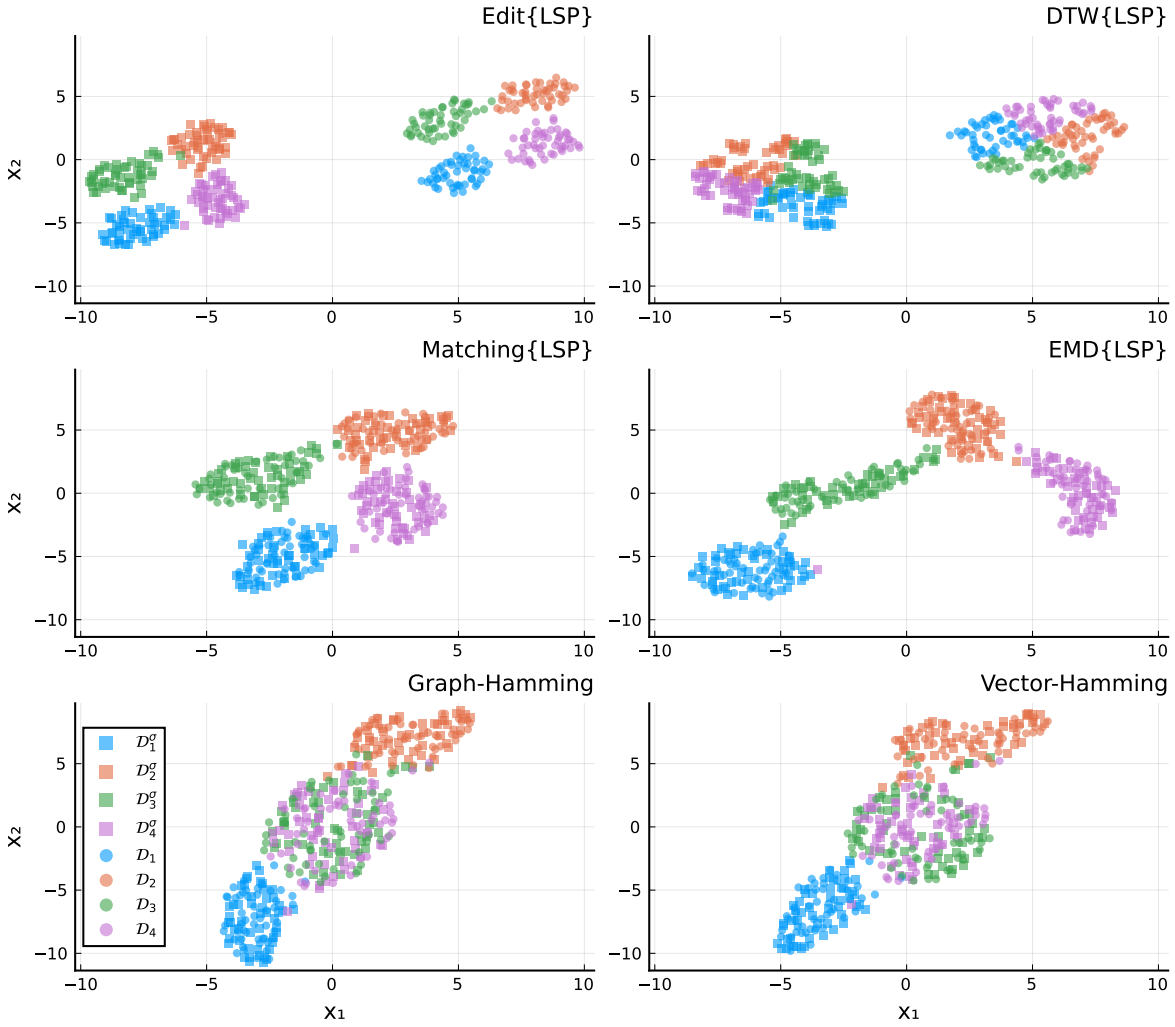


Figure 3.8.1: UMAP embeddings for a single run of the sequence simulation. Here, one can observe the two sequence distances at the top clearly separate observations from all eight distributions, with this distinction appearing slightly more marked for the edit distance. As expected, since multiset distances are order-invariant they cannot distinguish samples from $\mathcal{D}_i^g$ and $\mathcal{D}_i$, whilst aggregate distance disregard even more information and thus further confuse samples from $\mathcal{D}_3$ and $\mathcal{D}_3^g$ with those from $\mathcal{D}_4$ and $\mathcal{D}_4^g$.

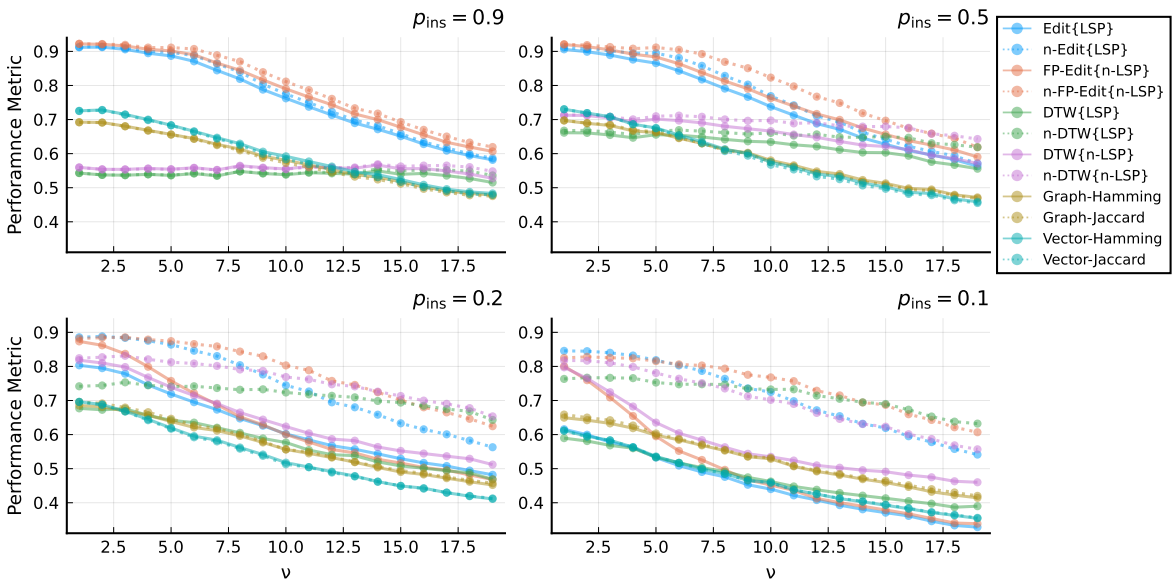


Figure 3.8.2: Comparing performance of distances at identifying samples from the same distribution as simulation parameters ν and p_{ins} are varied. Note here a solid and dashed line of the same color regard the standard and normalised versions of a given distance.

In this analysis, we consider whether it is possible to partition data points, that is, football matches, into groups, or *clusters*. As in the simulation studies of Sections 3.6 and 3.8, it is possible to get an indication such structure by visualising the data through use of an embedding method, and again we invoke UMAP for this purpose.

Initially, we compare the embeddings resulting from three different distances: the EMD (Definition 3.5.3), the normalised fixed-penalty edit distance (Definition 3.7.1), and the graph Jaccard distance (Section 3.3). Here for the EMD and edit distances, we opt to consider the normalised LSP as the interaction distance. Figure 3.9.1 shows the resultant UMAP embeddings obtained given each of these distances, wherein there appears to be strong indication of clusters.

Focusing on the EMD distance, we now run a cluster analysis. In particular, we consider applying a clustering algorithm to the embedded data, opting specifically for HDBSCAN (McInnes et al., 2017). This is a density-based algorithm which, at

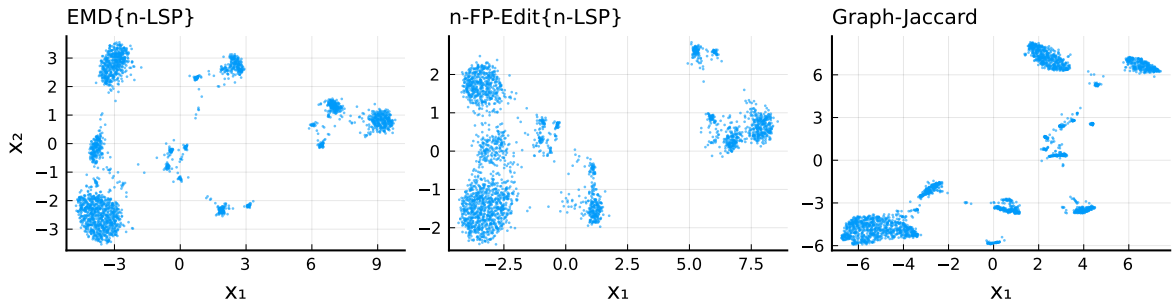


Figure 3.9.1: UMAP embeddings of in-play football data using three different distances, as indicated above each subfigure.

a high-level, uncovers groups of data points which are tightly packed together. Figure 3.9.2 visualises the cluster allocations that were obtained, showing a total of 11 clusters. Note HDBSCAN labels some data points as "noise", meaning they are not allocated to any cluster.

Towards considering what these clusters might represent, it is possible from the StatsBomb data to infer the formation a team was playing at any given time in a match. Formations represent a tactical choice made by a team, and most importantly determine the player positions that are used, that is, which vertices are likely to appear. For example, a "4-4-2" formation consists of four defenders, four midfielders and two attacking forwards. For each cluster, one can consider the proportion matches wherein a given formation was used (for the majority of the match), as visualised in the bottom of Figure 3.9.2. From this, one can see each cluster appears to be dominated by a single formation, indicating that clusters perhaps correspond to formations.

However, it appears there are exceptions to this rule: the formations predominantly appearing clusters 2 and 3 are the same, as is the case for clusters 4 and 5. The root cause can be revealed by considering what player positions (vertices) are actually being used by observations within these clusters, as we visualise in Figure 3.9.3. Here we can see, though observations in clusters 2 and 3 regard matches where teams were, according to the StatsBomb specification, playing the same formation, it appears they actually used slightly different player positions. Namely, where cluster 2

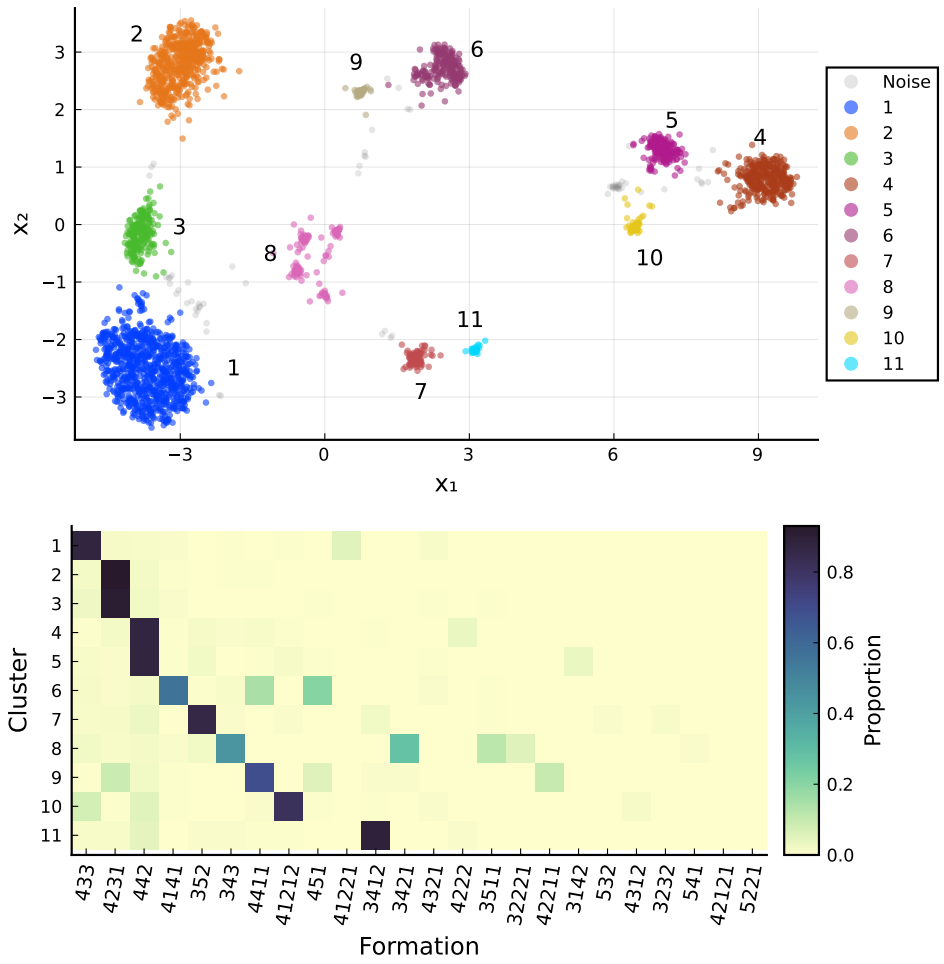


Figure 3.9.2: Summarising the clustering of in-play football data. Here the top figure shows cluster allocations obtained via HDBSCAN applied to the EMD embedding (Figure 3.9.1), whilst the bottom figure shows, for each cluster, the proportion of matches in which a given formation was used.

appears to make use of "LDM" (left defensive midfielder) and "RDM" (right defensive midfielder), cluster 3 instead uses "LCM" (left center midfielder) and "RCM" (right center midfielder). With this, it appears in fact these clusters correspond to two different *styles* of the same formation, namely a defensive and attacking version of the "4-2-3-1", respectively. Similarly, clusters 4 and 5 appear to correspond to a defensive and attacking version of the formation "4-4-2".

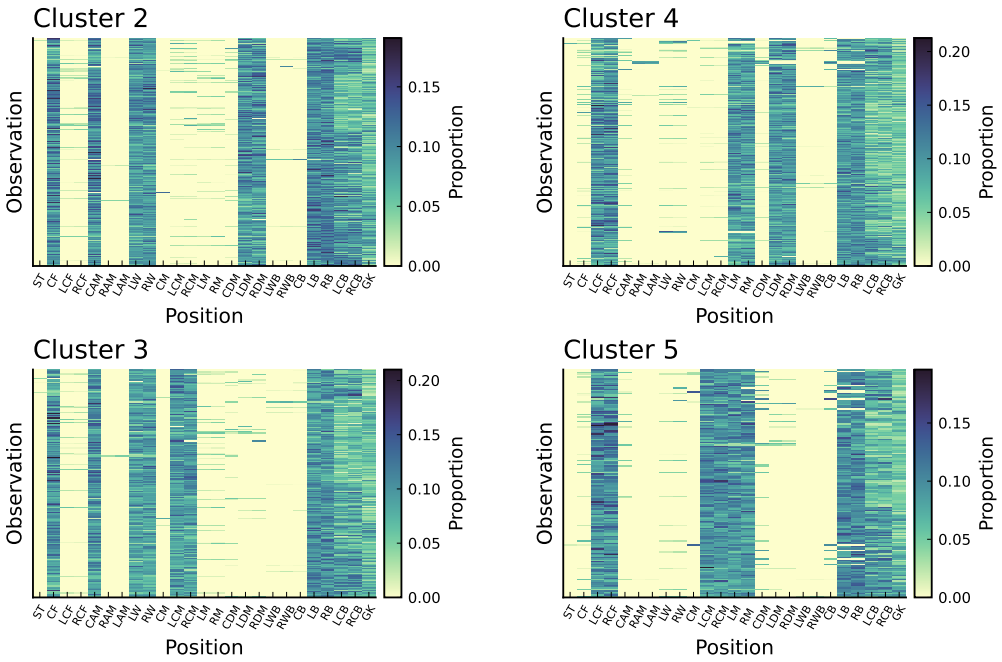


Figure 3.9.3: Examining the player positions (vertices) used by observations within different clusters. In each subplot, a single horizontal slice corresponds to a single observation, visualising the proportion of times each player position appears therein (using abbreviations of Table 2.4.1).

3.9.2 Foursquare check-in data

In this case, a single interaction $\mathcal{I}_i = (x_{i1}, \ldots, x_{in_i})$ represents a sequence of venue check-ins made by a single user, where $x_{ij} \in \mathcal{V}$ with $\mathcal{V}$ denoting the set of venue categories. As such, a single observation will represents the whole history of check-ins for a single user over a given time period.

We considered analysing the global version of the Foursquare dataset (Yang et al., 2016, 2015a). Unfortunately, this is to large to analyse completely, containing in total over 200,000 observations. Instead, we randomly sampled 500 users from each of six countries (Australia, Germany, Japan, Mexico, United Kingdom and United States), leading to a total of 3,000 observations. Note before doing so we also filtered out users with less than 10 observed interactions.

In this analysis, we consider assessing whether there is some form of relationship between the country of a user and their distance. Again, this can be examined visually

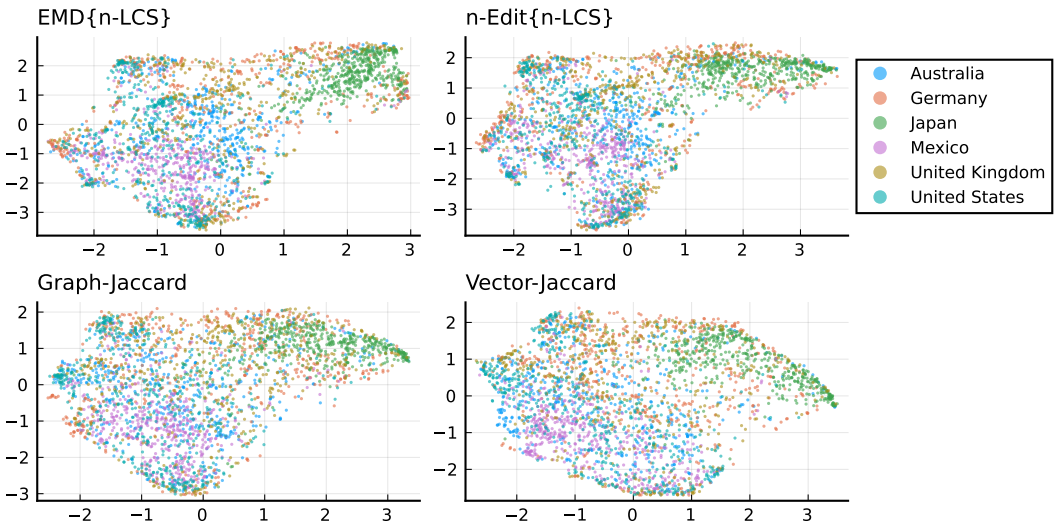


Figure 3.9.4: UMAP embeddings of Foursquare check-in data using four different distances, as indicated above each subfigure, with user country indicated.

through an embedding. Figure 3.9.4 shows UMAP embeddings obtained with four different distances with the country of each user indicated. Note in this case we opt to consider the (normalised) LCS distance to compare interactions. Though there does not appear to be a strong clustering of data points by country (as there was by formation in Section 3.9.1), there is some indication of correspondence between a user's country and their location in the embedded space, with users from Japan in particular appearing to be somewhat distinct.

To go beyond this qualitative assessment, we consider how distances outlined in Sections 3.3, 3.7 and 4.2.1 might perform at predicting the country of user. Given C countries ($C = 6$ in this case) we encode the country membership of the ith user via the vector $\boldsymbol{y}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$, where $y_{ic} = 1$ if the ith user was in the cth country. Assuming the country of the ith user was unknown, a natural approach to predict its value is via a k-nearest neighbours (k-NN) classifier, whereby one can obtain an estimate $\hat{\boldsymbol{y}}_i = (\hat{y}_{i1}, \ldots, \hat{y}_{ic})$ via

$$\hat{y}_{ic} := \frac{1}{k} \sum_{j \in \mathcal{N}_k^i} y_{jc}$$

where $\mathcal{N}_k^i$ denotes the k-nearest neighbours of the ith user (given a choice of distance), that is, we average the country membership vectors of the ith user's k-nearest neighbours. Notice $\sum_{c=1}^{C} \hat{y}_{ic} = 1$, so that $\hat{\boldsymbol{y}}_i$ is a probability vector with $\hat{y}_{ic}$ being the estimated probability of the ith user being a member of the cth country.

To estimate the error of a given k-NN classifier, we considered a leave-one-out cross-validation approach: for each user we (i) assume its country is unknown and predict its value via the k-NN classifier (given all others are known), then (ii) return a measure of prediction error. A single value is then obtained by averaging over all users. We consider measuring the error of a single prediction $\hat{\boldsymbol{y}}_i$ in two ways: (i) whether the country prediction $\hat{c}_i = \max_c \hat{y}_{ic}$ was correct, and (ii) a comparison with the true classification vector via $\sum_{c=1}^{C} (\hat{y}_{ic} - y_{ic})^2$, that is, the squared error. With this (i) leads to an estimated predictive accuracy, whilst (ii) leads to an estimated mean squared error (MSE).

Figure 3.9.5 shows estimates for the predictive accuracy and MSE obtained in this manner for various distances and choice of k, the number of neighbours. Here from the predictive accuracy values one can observe all distances do better than random guessing, which with $C = 6$ would have an accuracy of approximately 0.17 in this case. Thus there does appear in general some correspondence between a users country and their check-in patterns. It is also interesting to note the best performing distance, as judged by its best (over all k) MSE and predictive accuracy, is the Jaccard distance between vectors, an aggregate-based distance, which is followed by the EMD and the fixed-penalty matching distance. Though this may seem like a negative result, implying the use of the more complicated measures would be unjustified if prediction was the goal, one should note it could also be that these distances pick-up on finer aspects in which users from different countries are similar.

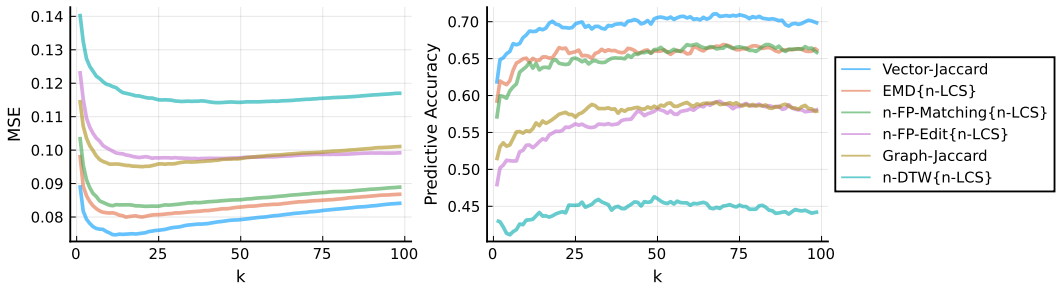


Figure 3.9.5: Estimated MSE and predictive accuracy of k-NN classifier for different choices of k and distance.

3.10 Discussion

In this chapter, the presently unconsidered problem of comparing interaction networks has been addressed. Following discussions of aggregate-based distances, four distance measures have been proposed which can be used to compare interaction networks directly, drawing inspiration from those seen in other contexts. Through simulation studies, the potential deficiencies of aggregate-based distances has been shown empirically, where distances comparing interaction multisets and sequences directly could correctly identify similar interaction networks where aggregate-based distances failed. Moreover, through example data analyses the practical applications of these distances has be illustrated, highlighting in particular their potential uses for clustering and prediction of network-level covariate information.

The simulation studies of Sections 3.6 and 3.8 have also highlighted some interesting points. In particular, it was seen in Section 3.6 that when comparing interaction multisets the EMD appears to do well even when observation have a very different number of interactions, whilst the matching distance appeared to struggle. Moreover, across both studies, it was seen that normalisation of distances improved there ability to distinguish observations sampled from the same distribution, particularly when the size of observations was more variable.

Regarding possible future work, it would be interesting to explore if a distance between interaction sequences could be proposed that would achieve performance

in the simulation of Section 3.8 matching that observed for the EMD in Section 3.6. A possible approach would be to borrow ideas used to define the EMD but introduce some consideration of interaction order.

Additionally, though we considered very similar simulation studies for sequences and multisets, one should note that sequences can differ from one another in far more ways than multisets can. With this, the simulation of Section 3.8 is arguably simplistic, essentially assessing if distances can identify interaction sequences which were 'proportional' to one another, in the sense they contain similar interactions, in a similar order, and in similar proportions. As such, if future work was done regarding such distances, it would be interesting to consider further simulation studies that might explore how well distances do at identifying other more subtle sequential patterns.

Finally, in Donnat and Holmes (2018) it was seen that graph distances can be distinguished by whether they take into account more local or global differences. With the distances proposed here to compare interaction sequences and multisets all utilising a distance between interactions, they appear to be inherently local. With this, it would be interesting to consider what 'global' differences might look like for interaction networks, and whether distances could be defined that would identify them. Of course, one could consider using a graph distance between aggregates that captures global differences. However, it would be interesting to explore whether such distances could be proposed that would respect the structure of the data. For example, one could consider alternative definitions of vertex centrality for interaction networks, using these as a basis for comparison, as in the centrality-based graph distances mentioned in Section 2.3.1.

Chapter 4

Modelling Populations of Interaction Networks via Distance Metrics

4.1 Introduction

Given one has observed a sample of interaction networks, a natural question that arises is how to summarise it? Do observed networks share anything in common? How strong is this common 'theme' throughout the sample? In this chapter, a new modelling methodology will be proposed to facilitate a reasoned statistical approach to answering such questions.

Given the dual representations, an observation of n interactions networks amounts to observing

$$\mathcal{S}^{(1)}, \ldots, \mathcal{S}^{(n)} \quad \text{or} \quad \mathcal{E}^{(1)}, \ldots, \mathcal{E}^{(n)}$$

that is, either a sample of interaction sequences or multisets. Towards summarising these samples, the following questions will be considered (as posed in Chapter 1):

(a) What is an average in this context?

(b) How can variability of these data be quantified?

To approach these questions via current multiple-network methodologies (Section 2.3) would require first aggregating observations to graphs. However, such aggregation will lead to a loss of information, restricting the insights that can be obtained. This motivates the proposal of an approach which instead respects the path-observed nature of the data, requiring no aggregation.

With this, in this chapter a novel Bayesian modelling framework will be proposed. Building upon the work of Chapter 3, this utilises distances between interaction networks to construct families of models via location and scale, akin to Gaussian distributions over the space of interaction networks. In this way, the location and scale parameters represent analogues of mean and variance, respectively, inference of which opens to the door to answering questions (a) and (b).

Inference for the proposed models is complicated somewhat by the presence of intractable normalising constants, leading to the induced posteriors being doubly-intractable. Additionally, with the location parameter being itself an interaction network, this leads to a non-trivial multi-dimensional discrete parameter space. As such, a specialised MCMC algorithm is also proposed, combining the exchange algorithm (Murray et al., 2006), which circumvents normalising constant evaluation, with the recently proposed involutive MCMC (iMCMC) framework of Neklyudov et al. (2020), which provides added flexibility in the proposal generation mechanism necessary to explore the parameter space.

The remainder of this chapter is structured as follows. In Section 4.2, the proposed models are introduced, along with discussions regarding choice of distances and model interpretation. In Section 4.3, a Bayesian approach to inference is then outlined, wherein the proposed MCMC algorithm is detailed. Section 4.4 then outlines three simulation studies confirming the efficacy of the proposed methodology and inference scheme, before practical applications are illustrated in Section 4.5 via an example analysis of the Foursquare check-in data (Section 2.4.2). Finally, the chapter

concludes with disucssions in Section 4.6.

4.2 Distance-based interaction-network models

We now introduce our proposed models for interaction sequences and multisets. In defining these models, we draw inspiration from the approach taken in Lunagómez et al. (2021) for analysing samples of networks, wherein a model for graphs was constructed via the use of distance metrics between graphs. The approach also has connections to models beyond the networks literature, including the Mallows model (Vitelli et al., 2018), proposed to analyse ranks in the context of preference learning, and the complex Watson distribution (Mardia and Dryden, 1999), which appears in shape analysis applications.

The core idea is to assume observed data points, be they multisets or sequences, are 'noisy' realisations of some unknown ground truth, with quantification of this noise being facilitated by a pre-specified distance measure. Equivalently, they can be seen as Gaussian-like distributions over their respective spaces, controlled by a location parameter, itself an interaction sequence or multiset, and a real-valued scale parameter.

4.2.1 Model definitions

Starting with our model for interaction sequences, let $\boldsymbol{\mathcal{S}}^*$ denote the infinite discrete space consisting of all interaction sequences over the fixed vertex set $\mathcal{V}$, further details of which can be found in Appendix B.1. Towards eliciting a probability distribution over $\boldsymbol{\mathcal{S}}^*$, we first endow it with a distance $d_S : \boldsymbol{\mathcal{S}}^* \times \boldsymbol{\mathcal{S}}^* \to \mathbb{R}_+$, permitting the comparison of elements therein. We then select an element of the space $\mathcal{S}^m \in \boldsymbol{\mathcal{S}}^*$, referred to as the *mode*, upon which to center the distribution, before choosing $\gamma > 0$, referred to as the *dispersion*, controlling the concentration of probability mass in $\boldsymbol{\mathcal{S}}^*$ about the

mode $\mathcal{S}^m$. In this way, $\mathcal{S}^m$ and γ can be seen as location and scale parameters, respectively, analogous to the mean and variance of a Gaussian distribution. A family of probability distributions can now be defined as follows.

Definition 4.2.1 (Spherical interaction sequence family)**:** Given a distance measure $d_S(\cdot,\cdot)$ on $\boldsymbol{\mathcal{S}}^*$, mode $\mathcal{S}^m \in \boldsymbol{\mathcal{S}}^*$ and dispersion parameter $\gamma > 0$, the probability of observing $\mathcal{S}$ is given by

$$p(\mathcal{S} \,|\, \mathcal{S}^m, \gamma) \propto \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\}, \tag{4.2.1}$$

and we write

$$\mathcal{S} \sim \text{SIS}(\mathcal{S}^m, \gamma) \tag{4.2.2}$$

if we assume $\mathcal{S}$ was sampled via (4.2.1). This we refer to as the Spherical Interaction Sequence (SIS) family of probability distributions over $\boldsymbol{\mathcal{S}}^*$ with parameters $\mathcal{S}^m$ and γ.

In a similar manner, for our interaction multiset model $\boldsymbol{\mathcal{E}}^*$ will denote the sample space, here consisting of all interaction multisets over the fixed vertex set $\mathcal{V}$ (further details provided in Appendix B.1). Again, we endow $\boldsymbol{\mathcal{E}}^*$ with a distance $d_E : \boldsymbol{\mathcal{E}}^* \times \boldsymbol{\mathcal{E}}^* \to \mathbb{R}_+$, before constructing a distribution over $\boldsymbol{\mathcal{E}}^*$ via location and scale in exactly the same way. However, in this case our location parameter will be an interaction *multiset* $\mathcal{E}^m \in \boldsymbol{\mathcal{E}}^*$. The resultant family of probability distributions is defined as follows.

Definition 4.2.2 (Spherical interaction multiset family)**:** Given a distance measure $d_E(\cdot,\cdot)$ on $\boldsymbol{\mathcal{E}}^*$, mode $\mathcal{E}^m \in \boldsymbol{\mathcal{E}}^*$, and dispersion parameter $\gamma > 0$, the probability of observing $\mathcal{E}$ is given by

$$p(\mathcal{E} \,|\, \mathcal{E}^m, \gamma) \propto \exp\{-\gamma d_E(\mathcal{E}, \mathcal{E}^m)\}, \tag{4.2.3}$$

and we write

$$\mathcal{E} \sim \text{SIM}(\mathcal{E}^m, \gamma) \tag{4.2.4}$$

if we assume $\mathcal{E}$ was sampled via (4.2.3). This we refer to as the Spherical Interaction Multiset (SIM) family of probability distributions over $\boldsymbol{\mathcal{E}}^*$ with parameters $\mathcal{E}^m$ and γ.

Note that in (4.2.2) and (4.2.4) no reference is made to $d_S(\cdot,\cdot)$ or $d_E(\cdot,\cdot)$, even though the respective distributions clearly depend on them. The reasoning here is these values are not intended to be model parameters but instead subjective choices made by the practitioner prior to any analysis. With this, it is assumed throughout that this distance is known *a priori*, that is, there is no uncertainty in this regard. In principal, one could relax this assumption. For example, in the SIS model we might have assume a distance $d_{S,\theta}(\cdot,\cdot)$ where θ are some distance-specific parameters that can be learnt alongside the model parameters $\mathcal{S}^m$ and γ. This, however, is an added layer of complexity that will not be considered here.

Observe that both models indeed agree with the intuition of location and scale. Take for example the SIS model. Examining its probability mass function (4.2.1) we see an observation $\mathcal{S}$ has the highest probability when $\mathcal{S} = \mathcal{S}^m$, so that $\mathcal{S}^m$ does indeed correspond to the mode. Moreover, as $\mathcal{S}$ moves away from $\mathcal{S}^m$ the distance $d_S(\mathcal{S}, \mathcal{S}^m)$ increases, and consequently the probability of observing $\mathcal{S}$ goes down. In this way, $\mathcal{S}^m$ represents the center of the distribution, controlling it location. Furthermore, the rate at which the probability decreases as one gets further from $\mathcal{S}^m$ is controlled by γ, with larger values leading to a faster decrease. As such, when γ is larger there will be a greater concentration of probability mass about the mode. In this way, the dispersion γ controls the scale of the distribution.[1]

This latter aspect, the control of variance by γ, can be formalised via the *entropy*.

[1] In analogy with the Gaussian distribution, γ functions like the inverse of the variance, often referred to as the precision.

Considering the SIS model, this is defined to be

$$H(\mathcal{S}^m, \gamma) := -\mathbb{E}\left[\log p(\mathcal{S} \mid \mathcal{S}^m, \gamma)\right],$$

which quantifies the uniformity of $p(\mathcal{S}|\mathcal{S}^m, \gamma)$, whereby larger values of $H(\mathcal{S}^m, \gamma)$ imply this distribution is 'more uniform' over $\boldsymbol{\mathcal{S}}^*$, with a minimum value of $H(\mathcal{S}^m, \gamma) = 0$ attained by a pointmass. The entropy also has an interpretation with regards to randomness or variance, whereby distributions with a higher entropy are more random, that is more variable. It can be shown that with any $d_S(\cdot, \cdot)$ and $\mathcal{S}^m$, the entropy $H(\mathcal{S}^m, \gamma)$ is a monotonic function of γ (Appendix B.3), agreeing with the intuition that γ controls the variability of the distribution. This holds similarly for the SIM model.

The distribution as stated in Definition 4.2.1 is normalised as follows

$$p(\mathcal{S} \mid \mathcal{S}^m, \gamma) = Z(\mathcal{S}^m, \gamma)^{-1} \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} \tag{4.2.5}$$

where

$$Z(\mathcal{S}^m, \gamma) = \sum_{\mathcal{S} \in \boldsymbol{\mathcal{S}}^*} \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} \tag{4.2.6}$$

is the normalising constant, often referred to as the *partition function*. In general, due to $\boldsymbol{\mathcal{S}}^*$ being an infinite space, this summation is intractable; an aspect which will come into play significantly when we consider the computational aspects of the methodology in later sections. In fact, there is no guarantee that $Z(\mathcal{S}^m, \gamma)$ will even exist for a given γ. Consequently, for some parameterisations (4.2.5) may be an improper distribution. This has pragmatic implications when it comes to sampling from these models. In particular, one can observe a divergence in the size of sampled interaction networks when γ is low, that is, when attempting to draw observations at random from these models via sequential sampling algorithms such as MCMC (as will be

discussed in Section 4.3.6) the size of samples, both in terms of the number of interactions and the size of individual interactions, will tend to grow with each sample. For this reason, in practice we recommend working with bounded sample spaces, which we define in Appendix B.1.2. This effectively places a lid on the possible size of interaction networks, preventing such behaviour. Note even though such bounded spaces will be finite, their size grows significantly. As such, the partition function will continue to be intractable even for moderate choices of the bounds. For further elaborations regarding this recommendation, see Appendix B.4.

We finalise these details by noting the key differences between the models proposed here and those considered elsewhere. As mentioned, the models presented above borrow ideas from the multiple-network modelling approach introduced by Lunagómez et al. (2021). However, their methodology was proposed for the scenario where networks were represented via graphs, that is, not for *interaction* networks. In contrast, the models above are applicable to interaction networks, being instead defined over the spaces of interaction sequences or multisets. We also make far more flexible assumptions regarding the dimensions of objects being considered: we do not assume observations have a fixed number of interactions, or that interactions have a fixed length[2], whilst Lunagómez et al. (2021) assume the number of vertices are fixed, so that the size of graphs are effectively bounded. As we will see, this raises computational challenges when it comes to designing algorithms to sample from the models proposed here, or when attempting to infer their parameters given observed data. Note also that in Definitions 4.2.1 and 4.2.2 little constraint has been placed on the properties of the underlying distance measure, whilst Lunagómez et al. (2021) specify that this must be a distance *metric*, which, as will be argued in the next section, is perhaps overly restrictive. Finally, as has been mentioned, the models proposed here are also similar to those appearing beyond the networks literature, such as the

[2]In theory, though, as discussed, it is recommended that in practice one constrains the space.

Mallows model (Vitelli et al., 2018) and the complex Watson distribution (Mardia and Dryden, 1999). Again, the key difference is the space upon which we define our models.

4.2.2 Distance measures

Evidently, use of the models defined in Section 4.2.1 requires specification of distances d_S and d_E between interaction sequences and multisets. In this section, we provide guidance on choosing such distances, highlighting the properties or features one should take into account, before discussing which of the distances introduced in Chapter 3 are most suitable.

Desirable properties

When considering whether a distance might be used within either of the models proposed in Section 4.2.1, one should firstly consider its theoretical properties. As discussed in the previous section, currently no restriction has been placed on the distance measures being used within the models of Definitions 4.2.1 and 4.2.2. However, considering the metric conditions (i) to (iii), having a distance is which does not satisfy the identity of indiscernibles (i) will be undesirable. The reason being that such distances are likely to result in models which are unidentifiable, complicating parameter inference and making them of little practical use. However, for the remaining two metric conditions, namely symmetry (ii) and the triangle inequality (iii), there is no theoretical reason why they need to hold for the given distance to be used within the model. Nonetheless, there are practical consequences that should be borne in mind. Regarding the symmetry condition (ii), it should be noted that the proposed models only use one 'side' of the chosen distance, and thus a different model will be defined depending on which way round the distance is evaluated. For example, considering the SIS model, notice the probability $p(\mathcal{S} \mid \mathcal{S}^m, \gamma)$ of (4.2.1) in-

cludes $d_S(\mathcal{S}, \mathcal{S}^m)$ in its evaluation. However, one could quite easily include $d_S(\mathcal{S}^m, \mathcal{S})$ instead, which, for an asymmetric distance, would parameterise a different model. It seems unlikely that an asymmetric distance would be of interest, but if in such a case, one should take care in light of this fact. For the triangle inequality (iii), to reinforce the claim it need not hold, it is worth noting that a multivariate Gaussian with no correlation effectively uses a squared Euclidean distance, which does not satisfy the triangle inequality. Nonetheless, one should be careful that such distances continue to imply sensible modelling assumptions. Moreover, distances which do not satisfy the triangle inequality may induce a geometry on the underlying space that differs somewhat from distances which do, which could potentially have consequences for the computational schemes that will be introduced in subsequent sections, wherein we will attempt to navigate such spaces via sampling algorithms.

In addition to theoretical concerns, the computational cost of the distance should also be taken into account. As will be seen in later sections, distances feature heavily in our computational schemes, both to sample from our models and to conduct inference for their parameters. As such, these algorithms will be sped-up or slowed-down quite significantly depending on how costly the chosen distance is to compute.

Finally, one should consider modelling assumptions implied by a given distance. This can at times be subtle, and distances which prove useful in other applications may be unsuitable for use with these models. As a knock-on effect, the choice of distance will influence the interpretations of model parameters (as will be illustrated in the next section), most notably the mode, that is, $\mathcal{S}^m$ for the SIS model and $\mathcal{E}^m$ for the SIM model. As such, a distance which results in an easily interpretable value thereof will be beneficial.

Example distances

Of the distances in introduced in Chapter 3, it seems the edit and matching distances (Definitions 3.5.1 and 3.7.1) are the most suitable choices, for two main reasons: (i) both are distances metrics, and thus have strong theoretical properties, in particular, they satisfy metric condition (i), the identity of indiscernibles, and (ii) they lead to natural parameter interpretations, as will be illustrated in Section 4.2.3.

Note both the edit and matching distance require specification of a distance $d_I(\cdot,\cdot)$ between interactions, and a penalty $\delta(\cdot)$ for unmatched interactions. Regarding the interaction distance, both the LSP and LCS distances (Section 3.4) can be invoked. Moreover, as will be shown in Section 4.2.3, each implies slightly different modelling assumptions, in turn providing inferences which are subtly different. In terms of the penalty $\delta(\cdot)$, recall the two proposals of Chapter 3, namely (i) a fixed-penalty, whereby $\delta(\mathcal{I}) = \rho$ for some constant $\rho > 0$, and (ii) a distance-based penalty, where $\delta(\mathcal{I}) = d_I(\mathcal{I}, \Lambda)$, where Λ represents the null interaction, which typically corresponds in some way to the size of $\mathcal{I}$. Though both are valid, we argue that (i) is not suitable for use within our proposed models, opting instead for the distance-based penalty (ii). A justification for this will now be outlined.

The following argument will regard use of the edit distance within the SIS model, but it should be noted that similar reasoning can be be applied to the use of the matching distance within the SIM model. The key point is that adoption of the fixed penalty within the edit distance will lead to a distribution of probability over the underlying space which implies unrealistic modelling assumptions. Suppose we have assumed the edit distance $d_{\mathrm{E},\delta(\cdot)}$ (Definition 3.7.1), with a penalty function given by $\delta(\mathcal{I}) = \rho$, where $\rho > 0$ is a fixed constant. Suppose also the mode $\mathcal{S}^m = (\mathcal{I}_1^m, \ldots, \mathcal{I}_{N^m}^m)$ and dispersion γ have been fixed, and consider an observation $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ drawn from the SIS model with these parameters, that is $\mathcal{S} \sim \mathrm{SIS}(\mathcal{S}^m, \gamma)$. Moreover, assume $\mathcal{S}$ is such that $N > N_m$, that is, $\mathcal{S}$ has more paths than the mode. Since $\mathcal{S}$ has more

paths, at least one of these must be unmatched when evaluating $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}^m)$ (as in Figure 3.7.1a). Assuming that the ith path in $\mathcal{S}$ is such an unmatched path, writing

$$\mathcal{S}_{\mathcal{I}} = (\mathcal{I}_1, \ldots, \mathcal{I}_{i-1}, \mathcal{I}, \mathcal{I}_{i+1}, \ldots, \mathcal{I}_N),$$

denoting $\mathcal{S}$ with the ith path given by $\mathcal{I}$, consider now the conditional distribution of this path given the others, that is

$$p(\mathcal{I} | \mathcal{S}_{-i}, \mathcal{S}^m, \gamma)$$

where here we use the notation $\mathcal{S}_{-i}$ to denote all the paths of $\mathcal{S}$ excluding the ith, making clear that we are conditioning on their values. This distribution, over the space of paths, can be obtained directly from the probability of $\mathcal{S}_{\mathcal{I}}$ implied by the model, namely

$$\begin{aligned} p(\mathcal{I} \,|\, \mathcal{S}_{-i}, \mathcal{S}^m, \gamma) &\propto \exp\{-\gamma d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{\mathcal{I}}, \mathcal{S}^m)\} \\ &\propto \exp\{-\gamma \cdot \delta(\mathcal{I})\} \\ &= \exp\{-\gamma \rho\} \\ &\propto 1 \end{aligned} \tag{4.2.7}$$

where here we use the fact that since $\mathcal{I}$ is not included in the matching it features in $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}^m)$ only via its penalty. Notice that (4.2.7) implies each path $\mathcal{I}$ is equally likely, under the assumed model. Though this may seem innocuous, notice if considering the model over the infinite space of interaction networks, this conditional distribution will be over the infinite space of all paths. In this case, (4.2.7) will actually be an improper distribution, since its normalising constant will involve an infinite sum of constants. Moreover, this necessarily implies the whole unconditional distribution (of the model) will also improper. Even if we consider bounding the space, as discussed in Section 4.2.1, in assigning equal probability to all paths therein, the

distribution (4.2.7) will imply a higher probability of $\mathcal{I}$ being of long length, by virtue of their prevalence. For example, if we have $V = |\mathcal{V}|$ vertices, there will be V^k paths of length k, and V^{k+1} paths of length $k+1$, which would imply the probability of $\mathcal{I}$ being length $k+1$ will be V times higher than the probability its being length k. This is a very odd assumption to make and unlikely to hold in practice.

For these reasons, we recommend not using a fixed penalty within the edit or matching distances, opting instead for one which somehow takes the size of the path being penalised into account, such as the distance-based penalty. This way, the distribution of probability in the underlying space can be better controlled, avoiding the undesirable properties illustrated above.

4.2.3 Model interpretation

We now look to provide some intuition for our proposed models, examining their features visually by plotting some randomly sampled observations. Here we will (i) illustrate the role of γ in controlling level of noise, (ii) strike a comparison between the SIS an SIM models, showing how the assumptions regarding order of paths manifests itself in observations, and (iii) compare models with different distance metrics, in particular, those with different choices for the distance between interactions.

Note it is somewhat non-trivial to sample from the models defined in Section 4.2.1, and we must rely on an MCMC algorithm to do so. The details of this algorithm will be outlined later (in Section 4.3.6), when we turn to the problem of parameter inference. For now, we note it is via this algorithm which the observations illustrated in this section were drawn.

Figure 4.2.1 summarises these sampled observations. Here we have two tables, showing samples from SIS and SIM models respectively. These are further divided, showing samples from each model with different assumed distances. In particular, the edit distance d_{E} and matching distance d_{M} were used for the SIS and SIM

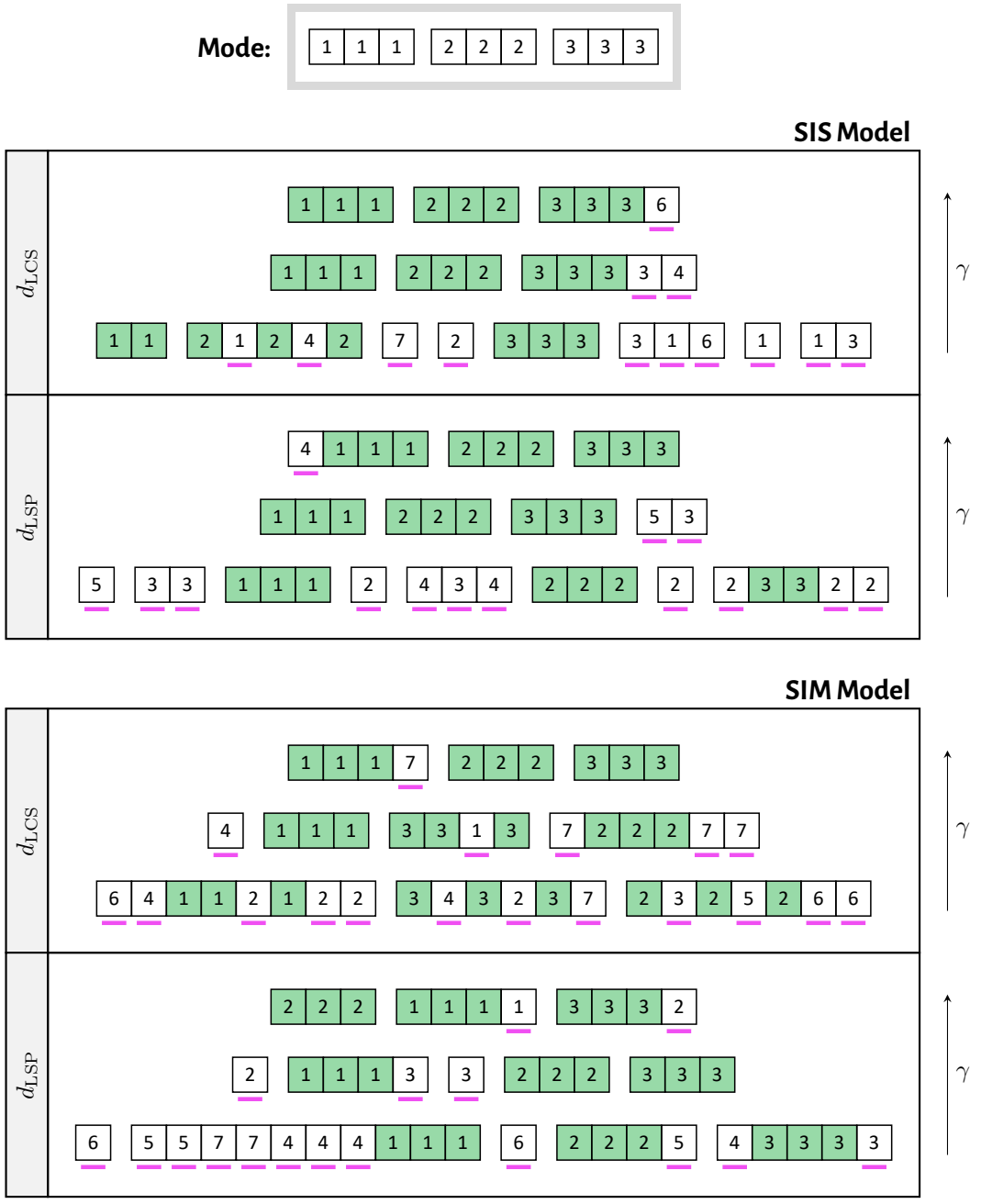


Figure 4.2.1: Example samples drawn from our models. Each table cell visualises three randomly drawn samples from a given model with the dispersion parameter γ varying. A common mode was used for each model, as displayed at the top. The edit and matching distances were assumed, for the SIS and SIM models, respectively, with different choices of path distance, as indicated on the left-hand tabs. For each sample, shaded entries indicate those matched with the mode, as implied by the optimal matchings and maximal common subsequences or subpaths found during distance evaluation, whilst underlined entries indicate unmatched entries or errors.

model respectively, with d_{LCS} and d_{LSP} as the interaction distances, as indicated in Figure 4.2.1 via the left-hand tabs. Within each cell, we show three samples from the associated model with increasing values for the dispersion, that is, for the SIS model we show samples $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ where $\mathcal{S}_i \sim \mathrm{SIS}(\mathcal{S}^m, \gamma_i)$, whilst for the SIM model we show $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ where $\mathcal{E}_i \sim \mathrm{SIM}(\mathcal{E}^m, \gamma_i)$, where the γ_i were increasing. The mode parameter for these models was fixed throughout and is shown at the top of Figure 4.2.1, that is, $\mathcal{S}^m = ((1,1,1),(2,2,2),(3,3,3))$ for the SIS models and $\mathcal{E}^m = \{(1,1,1),(2,2,2),(3,3,3)\}$ for the SIM models. The vertex set was also assumed to be $\mathcal{V} = \{1,\ldots,7\}$. Finally, entries have been highlighted to show how observations associate with the mode. In particular, shaded entries indicate those shared with the mode, whilst underlined entries represent errors. These were obtained from the optimal matchings and common subsequences or subpaths found when evaluating the distance of these samples to the mode.

Considering first the role of γ in controlling noise, this can be observed through the presence of a larger number of underlined entries for observations drawn with lower γ values, that is, those towards the bottom of each cell. Notice how this follows from the location and scale structure of the model, as discussed in Section 4.2.1: as γ decreases the probability becomes less concentrated about the mode, leading to a higher probability of entries *not* being shared.

Notice also, for all models, each sampled observation contains paths with shaded entries that can be matched with exactly one in the mode. Take, for example, the sample at the very bottom. Here the second path has three shaded entries $(1,1,1)$ which one can see is equal to the first path of the mode. Similarly, the fourth and fifth paths of this observation can be matched with the second and third of the mode. This feature, whereby paths in the observations are matched with a path of the mode, is a consequence of using the edit and matching distances, which, as seen in Definitions 3.5.1 and 3.7.1, are defined by such matchings.

Turning now to comparing the SIS and SIM models, notice how the SIS model preserves the order of paths in the mode, that is, they feature in the same order in sampled observations (albeit with some noise). In contrast, with the SIM model the order of paths within sampled observations is not necessarily consistent with the mode, for example, in the top observation in the lowest cell. Notice this is expected, since for the SIM model, being a distribution over multisets, two samples equal up to a permutation of path order would be considered the same.

A final point of note regards how the choice of distance, and the modelling assumptions this implies, manifests itself. Comparing samples drawn from both the SIS and SIM models with different choices for the distance between interactions, one can observe different structure in the error or noise, particularly evident as γ decreases. In particular, when d_{LCS} is assumed the paths of the mode appear as subsequences of those in the sampled observations, whilst when d_{LSP} is assumed they instead feature as subpaths.

Now, observe features outlined above will alter the interpretation of model parameters in each case, most notably the mode. In particular, by the reasoning above, in using the edit and matching distances, the paths of the mode will each be related to at most one path within each sample. With SIS model these paths appear (with noise) in the same order in the observed samples as they do in the mode, whilst in the SIM model the order of the mode and the samples need not be congruent. As such, for the SIS model $\mathcal{S}^m$ represents a sequence of paths often appearing in the observations, whilst for the SIM model $\mathcal{E}^m$ represents a *collection* of paths often appearing in the observations (in any order). Moreover, with d_{LCS} as the distance between paths, the paths of the mode represent subsequences appearing within the samples, whilst if using the d_{LSP} they will represent subpaths. All together, these imply a different role and interpretation for the mode in each case.

4.3 Bayesian inference

Given an assumed model, the goal of inference is to discern which parameters are likely to have generated the observed data. In our case, this amounts to inferring the mode and dispersion parameters. We approach this task via a Bayesian perspective, first assuming prior distributions for model parameters before incorporating observed data to form the posterior. Through a specialised MCMC algorithm, we subsequently obtain samples from the posterior upon which to base our inference. In this section, we provide details regarding each of these aspects.

For brevity, we give details regarding the interaction-sequence models (Definition 4.2.1) only, noting the approach taken for the multiset models is almost identical, albeit with a change of notation and some minor alterations to the MCMC algorithms. Full details of inference for the interaction-multiset models analogous to those given below can be found in Appendix B.7.

4.3.1 Priors, hierarchical model and posterior

In specifying a prior for the mode, we assume it was itself sampled from an SIS model, that is

$$\mathcal{S}^m \sim \text{SIS}(\mathcal{S}_0, \gamma_0) \tag{4.3.1}$$

where $(\mathcal{S}_0, \gamma_0)$ are specified hyperparameters. For the dispersion γ, we simply require a distribution $p(\gamma)$ whose support is a subset of the non-negative reals. For example, we typically take $\gamma \sim \text{Gamma}(\alpha_0, \beta_0)$ with (α_0, β_0) being hyperparameters. Given these specifications, an observed sample $\{\mathcal{S}^{(i)}\}_{i=1}^n$ is thus assumed to be drawn via

$$\begin{aligned} \mathcal{S}^{(i)} \,|\, \mathcal{S}^m, \gamma &\sim \text{SIS}(\mathcal{S}^m, \gamma) \quad (\text{for } i = 1, \ldots, n) \\ \mathcal{S}^m &\sim \text{SIS}(\mathcal{S}_0, \gamma_0) \\ \gamma &\sim p(\gamma). \end{aligned} \tag{4.3.2}$$

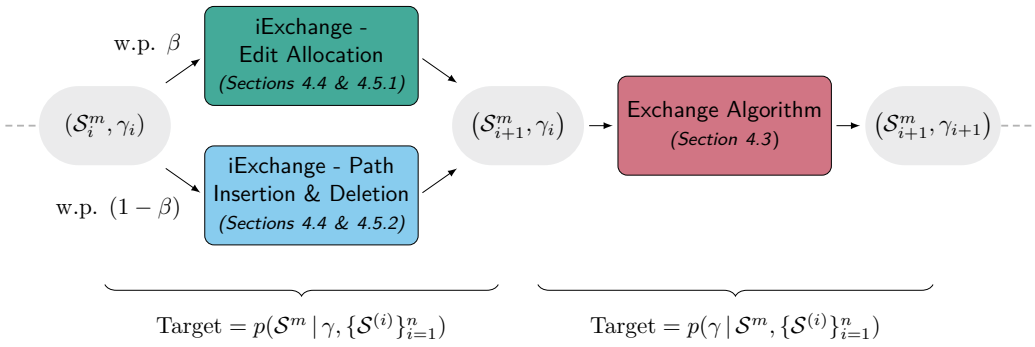


Figure 4.3.1: Summary of our MCMC scheme to sample from the SIS posterior. We first update the mode via the iExchange algorithm, doing an edit allocation move with probability β, or a path insertion and deletion move otherwise. We then update the dispersion via the exchange algorithm.

The likelihood of the sample $\{\mathcal{S}^{(i)}\}_{i=1}^n$ is given by

$$\begin{aligned} p(\{\mathcal{S}^{(i)}\}_{i=1}^n \mid \mathcal{S}^m, \gamma) &= \prod_{i=1}^n p(\mathcal{S}^{(i)} \mid \mathcal{S}^m, \gamma) \\ &= Z(\mathcal{S}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m)\right\}, \end{aligned}$$

and we have the following posterior, up to a constant of proportionality

$$\begin{aligned} p(\mathcal{S}^m, \gamma \mid \{\mathcal{S}^{(i)}\}_{i=1}^n) &\propto p(\{\mathcal{S}^{(i)}\}_{i=1}^n \mid \mathcal{S}^m, \gamma) p(\mathcal{S}^m) p(\gamma) \\ &\propto Z(\mathcal{S}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m)\right\} \\ &\quad \times \exp\{-\gamma_0 d_S(\mathcal{S}^m, \mathcal{S}_0)\} p(\gamma). \end{aligned} \tag{4.3.3}$$

4.3.2 Sampling from the posterior

To sample from the posterior (4.3.3), we use a component-wise MCMC algorithm which alternates between sampling from the two conditional distributions

$$p(\mathcal{S}^m \mid \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n) \quad \text{and} \quad p(\gamma \mid \mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n). \tag{4.3.4}$$

Since the normalising constant $Z(\mathcal{S}^m, \gamma)$ depends on the parameters of interest this implies (4.3.3) is doubly-intractable (Murray et al., 2006; Møller et al., 2006). Such terms will also persist in both conditionals above, making them also doubly-intractable. This precludes the use of standard MCMC algorithms such as Metropolis-Hastings and necessitates use of the exchange algorithm proposed by Murray et al. (2006).

A high-level summary of our scheme is visualised in Figure 4.3.1. For the dispersion conditional, being a distribution over the real line, we can apply the exchange algorithm directly. In contrast, the mode $\mathcal{S}^m$ is a discrete object, the dimensions of which can vary both in terms of the number of paths and their lengths. This makes the sample space for the mode conditional far less trivial, and so we consider merging the exchange algorithm with the involutive MCMC (iMCMC) framework of Neklyudov et al. (2020); defining what we call the iExchange algorithm (Appendix B.5). To fully explore the sample space, we mix together two iExchange moves. In particular, with probability β, we enact a move perturbing the paths currently present, whilst with probability $(1 - \beta)$ we attempt a move which varies the number of paths.

4.3.3 Updating the dispersion

Here we outline our MCMC scheme to sample from the dispersion conditional. In this instance, we suppose $\mathcal{S}^m$ is fixed and $q(\gamma'|\gamma)$ is some proposal density. In a single iteration, given current state γ we do the following

1. Sample a proposal γ' from $q(\gamma'|\gamma)$

2. Sample an auxiliary dataset $\{\mathcal{S}_i^*\}_{i=1}^n$ of size n (same as observed data) where

$$\mathcal{S}_i^* \overset{\text{i.i.d.}}{\sim} \text{SIS}(\mathcal{S}^m, \gamma'),$$

3. Evaluate the following probability

$$\alpha(\gamma, \gamma') = \min\left\{1, \frac{p(\gamma'|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n)p(\{\mathcal{S}_i^*\}_{i=1}^n|\mathcal{S}^m, \gamma)q(\gamma|\gamma')}{p(\gamma|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n)p(\{\mathcal{S}_i^*\}_{i=1}^n|\mathcal{S}^m, \gamma')q(\gamma'|\gamma)}\right\} \tag{4.3.5}$$

4. Move to state γ' with probability $\alpha(\gamma, \gamma')$, staying at γ otherwise.

For the proposal $q(\gamma'|\gamma)$ we consider sampling γ' uniformly over a ε-neighbourhood of γ with reflection at zero. More specifically, we first sample $\gamma^* \sim \text{Uniform}(\gamma - \varepsilon, \gamma + \varepsilon)$ and then let $\gamma' = \gamma^*$ if $\gamma^* > 0$ and let $\gamma' = -\gamma^*$ otherwise.

This is a direct application of the exchange algorithm (Murray et al., 2006) and as such the resultant Markov chain admits $p(\gamma \mid \mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n)$ as its stationary distribution. Moreover, this is what one might call an "exact-approximate" MCMC algorithm, in the sense that (asymptotically) samples drawn thereof will be distributed according to the desired target, meaning that one could in theory obtain exact samples given infinite resource. A closed form of (4.3.5) and derivation thereof can be found in Appendix B.6.1.

Observe this algorithm requires the ability to sample from the model. As has already been noted in Section 4.2.3, this is non-trivial for our proposed models. However, it is possible to instead sample from these approximately via MCMC, and an algorithm to do so will be outlined in Section 4.3.6. By replacing the exact sampling above with approximate MCMC-based samples, we will consequently end-up with a slightly different algorithm. Crucially, the resulting algorithm will be approximate as opposed to exact-approximate, that is to say, even in the theoretical limit, samples will not necessarily be distributed according to the desired target but instead an approximation thereof. However, as the auxiliary samples look more like an i.i.d. sample, one will get closer to the respective exact-approximate algorithm. Thus, one can in theory get arbitrarily close to the exact-approximate scheme by taking steps to reduce the bias of the MCMC-based auxiliary samples, such as introducing a burn-in

period or taking a lag between samples.

4.3.4 Updating the mode

We now outline our MCMC scheme to sample from the mode conditional. The key difference here is in the proposal generation mechanism, which follows the iMCMC algorithm (Neklyudov et al., 2020) in using a combination of random sampling and deterministic maps. Here we assume the dispersion γ is fixed and $\mathcal{S}^m$ denotes our current state. Instead of specifying a proposal density, one defines auxiliary variables $u \in \mathcal{U}$, a deterministic function $f : \boldsymbol{\mathcal{S}}^* \times \mathcal{U} \to \boldsymbol{\mathcal{S}}^* \times \mathcal{U}$ and a conditional distribution $q(u \,|\, \mathcal{S}^m)$ over auxiliary variables. The function f must also be an *involution*, meaning that is acts as its own inverse, that is, $f^{-1} = f$. A single iteration now consists of the following

1. Sample $u \sim q(u|\mathcal{S}^m)$
2. Invoke involution $f(\mathcal{S}^m, u) = ([\mathcal{S}^m]', u')$, obtaining proposal $[\mathcal{S}^m]'$
3. Sample auxiliary dataset $\{\mathcal{S}_i^*\}_{i=1}^n$ of size n where

$$\mathcal{S}_i^* \overset{\text{i.i.d.}}{\sim} \text{SIS}([\mathcal{S}^m]', \gamma)$$

4. Evaluate the following probability

$$\alpha(\mathcal{S}^m, [\mathcal{S}^m]') = \min\left\{1, \frac{p([\mathcal{S}^m]' \,|\, \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n) p(\{\mathcal{S}_i^*\}_{i=1}^n \,|\, \mathcal{S}^m, \gamma) q(u' \,|\, [\mathcal{S}^m]')}{p(\mathcal{S}^m \,|\, \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n) p(\{\mathcal{S}_i^*\}_{i=1}^n \,|\, [\mathcal{S}^m]', \gamma) q(u \,|\, \mathcal{S}^m)}\right\} \tag{4.3.6}$$

5. Move to state $[\mathcal{S}^m]'$ with probability $\alpha(\mathcal{S}^m, [\mathcal{S}^m]')$, staying at $\mathcal{S}^m$ otherwise.

Much like the proposal density of a Metropolis-Hasting or exchange algorithm, the u, $f(\mathcal{S}^m, u)$ and $q(u \,|\, \mathcal{S}^m)$ represent free choices. We consider mixing together two such specifications, details of which we provide in the next section.

This scheme represents an instance of what we call the iExchange algorithm (Algorithm 9, Appendix B.5). As shown in Appendix B.5, this can be seen as a special case of the iMCMC algorithm. As such, this represents an exact-approximate MCMC algorithm with the resultant Markov chain admitting $p(\mathcal{S}^m|\gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n)$ as its stationary distribution. Note the iExchange algorithm as defined in Appendix B.5 includes a Jacobian term in the acceptance probability which we do not include above. The reasoning being that since both $\boldsymbol{\mathcal{S}}^*$ and $\mathcal{U}$ are discrete spaces and $f(\mathcal{S}, u)$ is a one-to-one function (since it is invertible) such terms are not required.

As with the dispersion update of the previous section, notice this similarly requires the ability to obtain exact samples from the model. Again, since we cannot do this in general, we propose to draw these samples via an MCMC algorithm that will be outlined in Section 4.3.6. Consequently, the resultant algorithm will again be approximate rather than exact-approximate, but will approach the exact-approximate algorithm the closer the MCMC samples get to an i.i.d. sample.

4.3.5 Mode update moves

We now give details regarding two iExchange specifications for the mode conditional updates. In the first, we keep the number of paths fixed, varying only the path lengths or what we call the *inner dimension*. For example, in the context of the Foursquare data, this would amount to altering a particular sequence of check-ins. In the second, we look to vary the number of paths or what we call the *outer dimension*. For example, in the Foursquare data this would equate to introducing or removing a whole day of check-ins.

Edit allocation

Supposing $\mathcal{S}^m = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ is our current state, the main idea of this move is to allocate a number of "edits" to each path in $\mathcal{S}^m$. These edits consist of inserting and

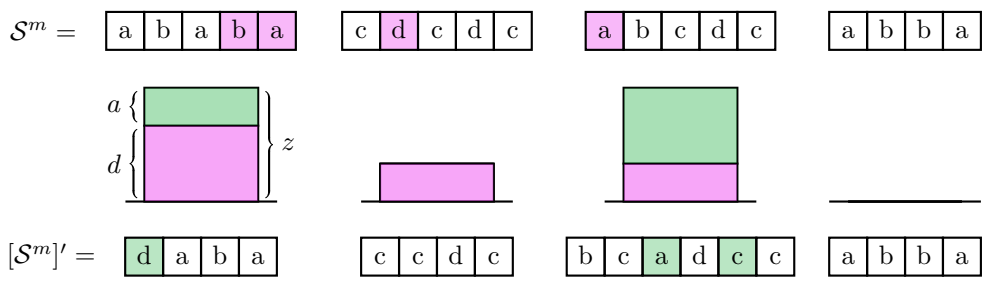


Figure 4.3.2: Illustrating the edit allocation move. Shaded entries indicate deletions and insertions, whilst bars visualise allocation of edits to paths. Bar height is proportional to the number of edits allocated to a path z, whilst the green (top) portion of the bar denotes the number of insertions a and the pink (bottom) portion represents the number of deletions d.

deleting entries, where if the number of insertions and deletions are unbalanced, paths of smaller or larger sizes relative to the current state will be proposed, thus varying the inner dimension. For an illustration, see Figure 4.3.2.

We now give descriptive details of this proposal generation mechanism and show how it can be cast in the light of iMCMC. First, we specify the total number of edits to be made, denoting this $\delta \in \mathbb{Z}_{\geq 1}$. Next, we specify an allocation of these edits to the paths of $\mathcal{S}^m$, denoting this $\boldsymbol{z} = (z_1, \ldots, z_N)$, where $z_i \in \mathbb{Z}_{\geq 0}$ denotes the number of edits allocated to the ith path such that $\sum_{i=1}^{N} z_i = \delta$. For example, in Figure 4.3.2 we have $\delta = 7$ and $\boldsymbol{z} = (3, 1, 3, 0)$.

Given z_i we edit the ith path $\mathcal{I}_i$ to obtain a corresponding proposal $\mathcal{I}_i'$ in the following manner. First, we partition the z_i edits between deletions and insertions, letting $d_i \in \{0, \ldots, \min(n_i, z_i)\}$ denote the number of deletions, where n_i denotes the length of the ith path, with $a_i = z_i - d_i$ then denoting the number of insertions. Note, we cannot delete more entries than are present, hence the restriction $d_i \leq \min(n_i, z_i)$.

The penultimate step is to specify which entries to delete and where to insert new entries, which we denote via *subsequences*. Introducing the notation $[n] = (1, \ldots, n)$, we define subsequence of $[n]$ of size m to be a vector $\boldsymbol{v} = (v_1, \ldots, v_m)$ such that $1 \leq v_1 < v_2 < \cdots < v_m \leq n$. Now, we let $\boldsymbol{v}_i$ be a subsequence of $[n_i]$ of size d_i

denoting the entries of $\mathcal{I}_i$ to be deleted, whilst $\boldsymbol{v}_i'$ is subsequence of $[m_i]$ of size a_i, denoting the location of entries to be inserted in $\mathcal{I}_i'$, where $m_i = n_i - d_i + a_i$ denotes the length of $\mathcal{I}_i'$. For example, considering the first path in Figure 4.3.2 we have $\mathcal{I}_1 = (a, b, a, b, a)$ and $\mathcal{I}_1' = (d, a, b, a)$ with $\boldsymbol{v}_1 = (4, 5)$ and $\boldsymbol{v}_1' = (1)$ indexing the deletions and insertions respectively. The final step is to specify entries to insert, which we denote $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{ia_i})$ where $y_{ij} \in \mathcal{V}$. For example, in Figure 4.3.2 we have $\boldsymbol{y}_1 = (d)$.

Given the information above, one can enact the specified deletions and insertions, mapping to a proposal $[\mathcal{S}^m]' = (\mathcal{I}_1', \ldots, \mathcal{I}_N')$. This can be viewed in the iMCMC framework as follows. First, collate all this information into the auxiliary variable $u = (\delta, \boldsymbol{z}, u_1, \ldots, u_N)$ where $u_i = (d_i, \boldsymbol{v}_i, \boldsymbol{v}_i', \boldsymbol{y}_i)$. Now, if we write the required involution as follows

$$f(\mathcal{S}^m, u) = (f_1(\mathcal{S}^m, u), f_2(\mathcal{S}^m, u)) = ([\mathcal{S}^m]', u'),$$

then in enacting the specified edit operations we have effectively defined the first component $f_1(\mathcal{S}^m, u) = [\mathcal{S}^m]'$. The second component we define as follows

$$f_2(\mathcal{S}^m, u) = (\delta, \boldsymbol{z}, u_1', \ldots, u_N')$$

where

$$u_i' = (z_i - d_i, \boldsymbol{v}_i', \boldsymbol{v}_i, (\mathcal{I}_i)_{\boldsymbol{v}_i}) \tag{4.3.7}$$

where $(\mathcal{I}_i)_{\boldsymbol{v}_i} = (x_{iv_1}, \ldots, x_{iv_{d_i}})$ is the subsequence of $\mathcal{I}_i$ indexed by $\boldsymbol{v}_i = (v_1, \ldots, v_{d_i})$. On an intuitive level, u_i' parameterises the edits to the ith path $\mathcal{I}_i$ which are exactly the opposite of those parameterised by u_i, namely we delete $z_i - d_i$ entries indexed by $\boldsymbol{v}_i'$, then insert entries $(\mathcal{I}_i)_{\boldsymbol{v}_i}$ at locations indexed by $\boldsymbol{v}_i$. In this way, enacting the operations parameterised by u' will take us back to $\mathcal{S}^m$, that is

$$f_1([\mathcal{S}^m]', u') = \mathcal{S}^m,$$

furthermore observe that reapplying the operations of (4.3.7) to u_i' itself takes us back to u_i

$$(z_i - (z_i - d_i), \boldsymbol{v}_i, \boldsymbol{v}_i', (\mathcal{I}_i')_{\boldsymbol{v}_i'}) = (d_i, \boldsymbol{v}_i, \boldsymbol{v}_i', \boldsymbol{y}_i) = u_i$$

where $\boldsymbol{y}_i = (\mathcal{I}_i')_{\boldsymbol{v}_i'}$ since $\boldsymbol{v}_i'$ indexed where the entries $\boldsymbol{y}_i$ were inserted in $\mathcal{I}_i'$. This implies

$$f_2([\mathcal{S}^m]', u') = (\delta, \boldsymbol{z}, u_1, \ldots, u_N)$$

and hence

$$f(f(\mathcal{S}^m, u)) = f([\mathcal{S}^m]', u') = (\mathcal{S}^m, u),$$

that is, $f(\mathcal{S}^m, u)$ is indeed an involution.

Regarding the auxiliary distribution $q(u|\mathcal{S}^m)$, we consider the following

$$\begin{aligned}
\delta &\sim \text{Uniform}\{1, \ldots, \nu_{\text{ed}}\} \\
\boldsymbol{z} \,|\, \delta &\sim \text{Multinomial}(\delta\,;\, 1/N, \ldots, 1/N) \\
d_i \,|\, z_i &\sim \text{Uniform}\{0, \ldots, \min(z_i, n_i)\} \qquad (\text{for } i = 1, \ldots, N)
\end{aligned}$$

whilst $\boldsymbol{v}_i$ and $\boldsymbol{v}_i'$ are drawn uniformly and the entry insertions $\boldsymbol{y}_i$ are assumed to be sampled from some general distribution $q(\boldsymbol{y}_i|\mathcal{I}_i)$, which we typically take to be the uniform distribution over $\mathcal{V}$. The only tuning parameter here is ν_{ed}, which controls the aggressiveness of proposals, with larger values leading to more edits being attempted on average.

Further details, including examples of possible insertion distributions $q(\boldsymbol{y}_i|\mathcal{I}_i)$ and derivations of key terms appearing the acceptance probability (4.3.6), can be found in Appendix B.6.3.

Path insertion and deletion

With this move we look to vary the outer dimension, that is, the number of paths. Similar to Section 4.3.5, we consider doing so by random deletion and insertion. The difference in this case is that we delete and insert whole paths (see Figure 4.3.3).

In particular, with $\mathcal{S}^m = (\mathcal{I}_1, \dots, \mathcal{I}_N)$ denoting our current state, we first choose a total number of insertions and deletions $\varepsilon \in \mathbb{Z}_{\geq 1}$. Next, we partition these, letting $d \in \{0, \dots, \min(N, \varepsilon)\}$ denote the number of deletions, leaving $a = \varepsilon - d$ insertions. For example, in Figure 4.3.3 we have $\varepsilon = 3$, $d = 2$ and $a = 1$. Next, we choose locations of deletions and insertions. In particular, we let $\boldsymbol{v}$ be a length d subsequence of $[N]$ denoting which paths of $\mathcal{S}^m$ are to be deleted, whilst $\boldsymbol{v}'$ is a length a subsequence of $[M]$, where $M = N - d + a$, denoting where inserted paths will be located in our proposal $[\mathcal{S}^m]'$. For example, in Figure 4.3.3 we have $\boldsymbol{v} = (2, 4)$ and $\boldsymbol{v}' = (3)$. Finally, for each $i = 1, \dots, a$ we choose some path $\mathcal{I}_i^*$ to insert into entry v_i' of $[\mathcal{S}^m]'$. For example, in Figure 4.3.3 we have a single path $\mathcal{I}_1^* = (c, b, b, a)$ which we insert to the third entry.

As in Section 4.3.5, given the information above we can insert and delete the corresponding paths to obtain a proposal $[\mathcal{S}^m]'$. Collating this into the auxiliary variable $u = (\varepsilon, d, \boldsymbol{v}, \boldsymbol{v}', \mathcal{I}_1^*, \dots, \mathcal{I}_a^*)$ this can similarly be seen as defining the first component of the required involution. The second component we define a follows

$$f_2(\mathcal{S}^m, u) := (\varepsilon, \varepsilon - d, \boldsymbol{v}', \boldsymbol{v}, \mathcal{I}_{v_1}, \dots, \mathcal{I}_{v_d}) = u'$$

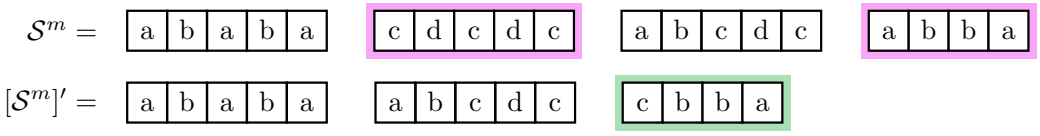


Figure 4.3.3: Illustrating path insertion and deletion move, where given current state $\mathcal{S}^m$ the proposal $[\mathcal{S}^m]'$ is obtained by deleting and inserting the highlighted paths.

which intuitively parameterises the exact opposite set of operations to u, namely where we make ε total insertions and deletions but instead delete $\varepsilon - d = a$ paths indexed by $\boldsymbol{v}'$, before inserting the paths $(\mathcal{I}_{v_1}, \ldots, \mathcal{I}_{v_d})$ (of $\mathcal{S}^m$) into locations indexed by $\boldsymbol{v}$. As such, we have the following

$$f_1([\mathcal{S}^m]', u') = \mathcal{S}^m,$$

and, furthermore, reapplying the second component just defined leads to

$$\begin{aligned} f_2([\mathcal{S}^m]', u') &= \left(\varepsilon, \varepsilon - (\varepsilon - d), \boldsymbol{v}, \boldsymbol{v}', \mathcal{I}'_{v'_1}, \ldots, \mathcal{I}'_{v'_{\varepsilon-d}}\right) \\ &= (\varepsilon, d, \boldsymbol{v}, \boldsymbol{v}', \mathcal{I}^*_1, \ldots, \mathcal{I}^*_a) \\ &= u \end{aligned}$$

using the fact that $\mathcal{I}'_{v'_i} = \mathcal{I}^*_i$, since by definition $\mathcal{I}^*_i$ was inserted to the (v'_i)th entry of $[\mathcal{S}^m]'$. Altogether this implies

$$f(f(\mathcal{S}^m, u)) = f([\mathcal{S}^m]', u') = (\mathcal{S}^m, u),$$

that is, $f(\mathcal{S}^m, u)$ is an involution.

Regarding sampling of auxiliary variables, we consider the following

$$\begin{aligned} \varepsilon &\sim \text{Uniform}\{1, \ldots, \nu_{\text{td}}\} \\ d \,|\, \varepsilon &\sim \text{Uniform}\{0, \ldots, \min(N, \varepsilon)\} \end{aligned}$$

whilst we sample $\boldsymbol{v}$ and $\boldsymbol{v}'$ uniformly and assume path insertions $\mathcal{I}^*_i$ are drawn from some general distribution over paths $q(\mathcal{I}|\mathcal{S}^m)$. This leaves two tuning parameters, ν_{td} and $q(\mathcal{I}|\mathcal{S}^m)$, which in combination facilitate control over the aggressiveness of proposals. In particular, ν_{td} controls the number of deletions and insertions attempted, whilst $q(\mathcal{I}|\mathcal{S}^m)$ affects how impactful each of these insertions and deletions are. Again,

further details, including key terms needed for evaluating acceptance probabilities, can be found in Appendix B.6.4.

4.3.6 Sampling auxiliary data

Both algorithms to target the conditionals outlined in Sections 4.3.3 and 4.3.4 require exact sampling of auxiliary data from appropriate interaction-sequence models. Unfortunately, we cannot do this in general. Instead, we consider replacing this with approximate samples obtained via an iMCMC algorithm.

In particular, suppose we would like to obtain samples from an SIS($\mathcal{S}^m, \gamma$) model. Assuming that $\mathcal{S}$ denotes the current state, and auxiliary variables u, involution $f(\mathcal{S}, u)$ and auxiliary distribution $q(u|\mathcal{S})$ have be defined, in a single iteration we do the following

1. Sample $u \sim q(u|\mathcal{S})$
2. Invoke involution $f(\mathcal{S}, u) = (\mathcal{S}', u')$
3. Evaluate the following probability

$$\alpha(\mathcal{S}, \mathcal{S}') = \min\left\{1, \frac{p(\mathcal{S}' \,|\, \mathcal{S}^m, \gamma) q(u' \,|\, \mathcal{S}')}{p(\mathcal{S}, \,|\, \mathcal{S}^m, \gamma) q(u \,|\, \mathcal{S})}\right\} \tag{4.3.8}$$

4. Move to state $\mathcal{S}'$ with probability $\alpha(\mathcal{S}, \mathcal{S}')$, staying at $\mathcal{S}$ otherwise.

where $p(\mathcal{S}|\mathcal{S}^m, \gamma)$ denotes the likelihood as given in (4.2.1). Towards specifying u, $f(u, \mathcal{S})$ and $q(u|\mathcal{S})$, we now recycle the moves of Section 4.3.4, again mixing these together with some proportion $\beta \in (0, 1)$. Note, as in Section 4.3.4, we omit the Jacobian term in the acceptance probability above since we are working with discrete spaces.

In sampling auxiliary data in this manner, we now have two MCMC-based elements: what one might call the *outer* MCMC algorithm, navigating the parameter

space, and the *inner* MCMC algorithm, sampling auxiliary data. We note this approach has been considered by others. In particular, Liang (2010) proposed the so-called double Metropolis-Hastings algorithm which replaces the exact samples of the exchange algorithm with those obtained via a Metropolis-Hasting scheme. The difference in our case is use of the more general iMCMC framework, be that in the outer MCMC scheme (as in the iExchange algorithm), or the inner MCMC scheme (as outlined above).

4.4 Simulation studies

In this section, simulation studies undertaken to confirm the efficacy of the proposed methodology and inference scheme will be outlined. In the first two, the posterior concentration is examined, exploring how this is affected by variability of observed data and structural features of the mode. In the third, convergence of the posterior predictive is assessed via a missing data problem. In each, we will be working with the interaction-sequence models.

4.4.1 Posterior concentration

Given the observed data were generated by an SIS model at known parameters, one expects the posterior to concentrate about these values as the sample size grows, that is, the posterior should be *consistent*. The next two simulation studies will serve to not only confirm this, but also, in assuming the given posterior is indeed consistent, confirm the efficacy of our proposed MCMC algorithms at approximating this posterior. In addition, we explore what can impact the rate of posterior convergence, considering both the variability of the observed data and features of the true underlying mode parameter.

The high-level approach is the following. Given true mode $\mathcal{S}^m_{\text{true}}$ and dispersion

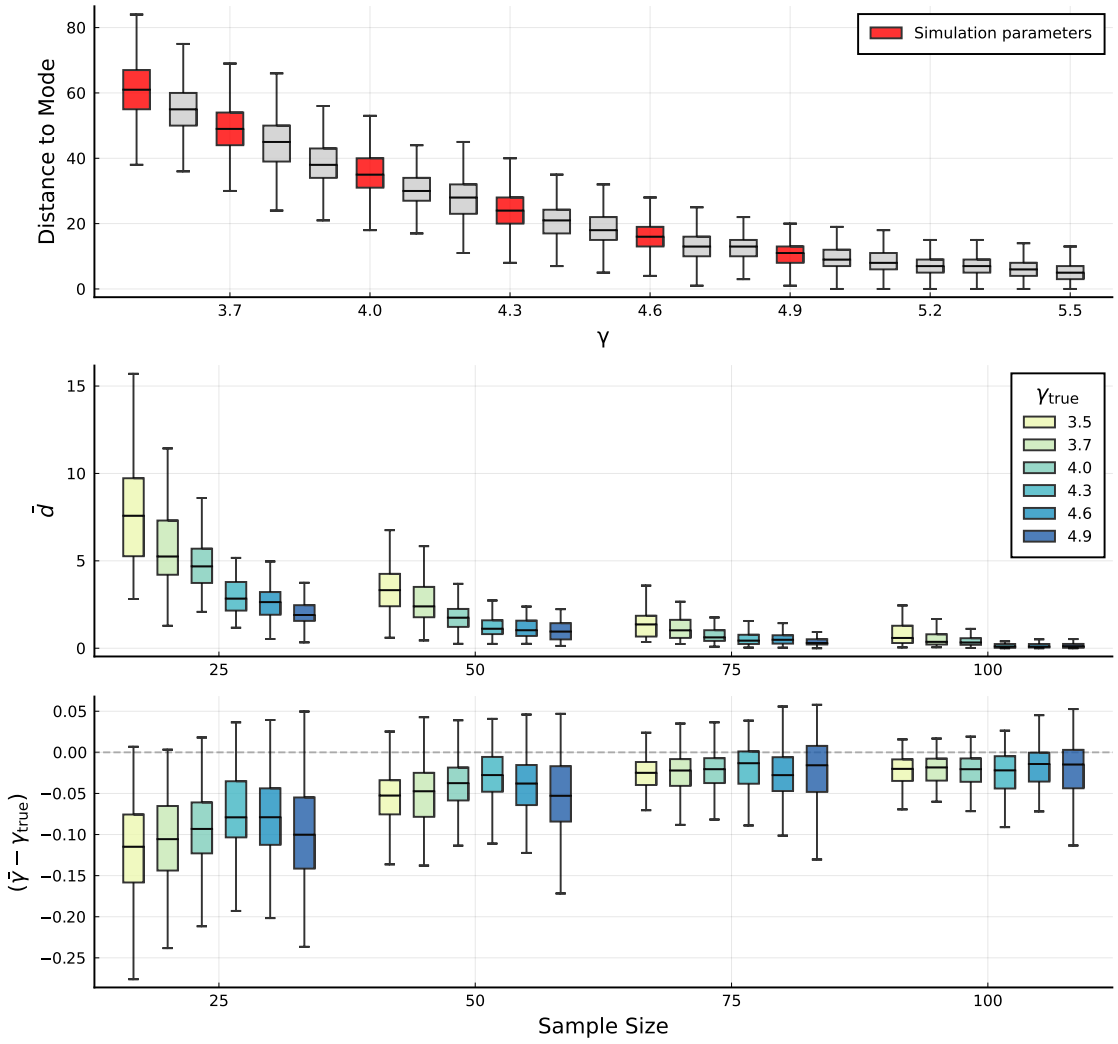


Figure 4.4.1: A summary of our first simulation (Section 4.4.1), where the top plot visualises the scale of the SIS model used therein, in particular, for different values of γ it shows $\{d_S(\mathcal{S}^{(i)}, \mathcal{S}^m_{\text{true}})\}_{i=1}^{1000}$ where $\mathcal{S}^{(i)} \sim \text{SIS}(\mathcal{S}^m_{\text{true}}, \gamma)$, sampled via the iMCMC scheme of Section 4.3.6. The remaining two plots summarise simulation outputs for each pair $(\gamma_{\text{true}}, n)$, where the middle shows distributions of $\bar{d}$, the average distance to the true mode, whilst the bottom shows $(\bar{\gamma} - \gamma_{\text{true}})$, the bias of the dispersion posterior mean relative to the truth.

γ_{true}, we draw a sample $\{\mathcal{S}^{(i)}\}_{i=1}^{n}$ where

$$\mathcal{S}^{(i)} \sim \text{SIS}(\mathcal{S}^{m}_{\text{true}}, \gamma_{\text{true}})$$

before obtaining samples $\{(\mathcal{S}^{m}_{i}, \gamma_i)\}_{i=1}^{m}$ from the posterior $p(\mathcal{S}^{m}, \gamma | \{\mathcal{S}^{(i)}\}_{i=1}^{n})$. We then assess the behaviour of these posterior samples via the following summary measures

$$\bar{d} := \frac{1}{m}\sum_{i=1}^{m} d_S(\mathcal{S}^{m}_{i}, \mathcal{S}^{m}_{\text{true}}) \qquad \bar{\gamma} := \frac{1}{m}\sum_{i=1}^{m} \gamma_i$$

where ideally $\bar{d}$ should be close to zero and $\bar{\gamma} \approx \gamma_{\text{true}}$. By repeating this a number of times for different n and evaluating these summaries we can thus get a sense of how the posterior is concentrating about the true parameters.

Now, recall the dispersion works inversely to the variance, in that lower values lead to more variable data (Figure 4.4.1, top). Intuitively, when the data is more variable it will be harder to discern the true mode $\mathcal{S}^{m}_{\text{true}}$, and thus we expect $\bar{d}$ to decrease more slowly for lower values of γ_{true}. Alternatively, as can be seen in Figure 4.4.1, when γ_{true} is smaller the difference of their parameterised distributions (as described by the distribution of distances to the mode) becomes more marked relative to neighbouring values. As such, we might also expect smaller values for the dispersion to be easier to recover.

To explore for such properties, we varied γ_{true} and n whilst keeping $\mathcal{S}^{m}_{\text{true}}$ fixed. In particular, we considered $\gamma_{\text{true}} = 3.5, 3.7, 4.0, 4.3, 4.6, 4.9$ (highlighted in Figure 4.4.1, top) and $n = 25, 50, 75, 100$. The distance we took to be $d_S = d_{\text{E}}$ with $d_I = d_{\text{LCS}}$ between paths. The number of vertices was fixed to $V = 20$, and the sample space constrained to be finite as defined in Appendix B.1.2, assuming at most $L = 20$ paths in any observation, with each path being of length at most $K = 10$.

The mode $\mathcal{S}^{m}_{\text{true}}$ of length $N = 10$ was fixed throughout, sampled from the Holly-

wood model of Crane and Dempsey (2018). In particular, we drew

$$\mathcal{S}^m_{\text{true}} \sim \text{Hollywood}(\alpha, \theta, \nu)$$

where

$$\alpha = -0.3 \quad \theta = 0.3V \quad \nu = \text{TrPoisson}(3, 1, K),$$

where $\text{TrPoisson}(\lambda, a, b)$ denotes a truncated Poisson distribution with $\lambda > 0$ the parameter of a standard Poisson, whilst $0 \leq a < b \leq \infty$ are the lower and upper bounds. Note this set-up for the Hollywood model, with $\alpha < 0$ and $\theta = -V\alpha$, corresponds to the finite setting, implying the sampled interaction sequences will have at most V vertices.

Regarding priors, we considered an uninformative set-up with $(\mathcal{S}_0, \gamma_0) = (\hat{\mathcal{S}}, 0.1)$ where

$$\hat{\mathcal{S}} := \underset{\mathcal{S} \in \{\mathcal{S}^{(i)}\}_{i=1}^n}{\text{argmin}} \sum_{i=1}^{n} d_S^2(\mathcal{S}^{(i)}, \mathcal{S})$$

denotes the sample Fréchet mean, whilst we took $\gamma \sim \text{Uniform}(0.5, 7.0)$. Here we note the sample $\{\mathcal{S}^{(i)}\}_{i=1}^n$ used to obtain $\hat{\mathcal{S}}$ will be different in each repetition of the simulation, and consequently so will $\hat{\mathcal{S}}$.

Now, for each pair $(\gamma_{\text{true}}, n)$ we (i) sampled n observations from an $\text{SIS}(\mathcal{S}^m_{\text{true}}, \gamma_{\text{true}})$ model, using the iMCMC scheme outlined in Section 4.3.6, with a burn-in period of 50,000 and taking a lag of 500 between samples (ii) obtained $m = 250$ samples from the posterior using the component-wise MCMC scheme of Section 4.3.2, with a burn-in period of 25,000 and taking a lag of 100 between samples[3] (iii) evaluated summary measures $\bar{d}$ and $\bar{\gamma}$.

We repeated (i)-(iii) 100 times in each case, the results of which are summarised in Figure 4.4.1. Consulting the middle plot, we observe that $\bar{d}$ decreases with n across

[3]One must also parameterise the MCMC algorithm used to sample the auxiliary data. These were tuned by considering acceptance probabilities observed when sampling from an $\text{SIS}(\mathcal{S}^m_{\text{true}}, \gamma_{\text{true}})$ distribution.

all cases, indicating a concentration of the posterior about the true mode. Furthermore, this decrease is more gradual for lower values of γ_{true}, agreeing with intuition. Turning to the bottom plot, the most obvious feature is bias in $\bar{\gamma}$ relative to the truth. Note this is expected, since we have used approximate MCMC samples within our component-wise scheme of Section 4.3.2. We do, however, see a reduction in this bias as the sample size grows. Furthermore, for the larger values of n we begin to see a clearer difference in the variance of $\bar{\gamma}$ across different values of γ_{true}. In particular, the variance appears to be smaller for lower values of γ_{true}, agreeing with the intuition that these are easier to estimate.

4.4.2 Effect of mode structure

Here we explored whether structural features of the mode might impact its inference. Adopting the same modelling set-up as the previous simulation, but in this case fixing the true dispersion to $\gamma_{\text{true}} = 4.5$, we re-sampled the mode in each repetition via

$$\mathcal{S}_{\text{true}} \sim \text{Hollywood}(\alpha, -\alpha V, \nu)$$

where we again take $V = 20$ and $\nu = \text{TrPoisson}(3, 1, K)$, whilst $\alpha < 0$.

The key idea underlying the Hollywood model is a 'rich get richer' assumption made when sampling vertices. This results in α admitting an interpretation regarding the heavy-tailed nature of vertex counts. In particular, for a given interaction sequence $\mathcal{S}$ and vertex $v \in \mathcal{V}$ one can define an analogue of the vertex degree (often defined for graphs) as follows

$$k_{\mathcal{S}}(v) := \text{\# times } v \text{ appears in } \mathcal{S},$$

which thus implies, for each $\mathcal{S}$, a sample $\{k_{\mathcal{S}}(v) : v \in \mathcal{V},\, k_{\mathcal{S}}(v) > 0\}$, similar in spirit to the degree distribution. Now, α can be seen to control the heavy-tailedness of

this distribution (see Figure 4.4.2), whereby when α is low one tends to see vertices appearing a similar number of times, whilst when α is larger these counts become disproportionately focused on a smaller subset of vertices.

We considered $\alpha = -\tilde{\alpha}$ where $\tilde{\alpha} = 1.35, 0.75, 0.35, 0.12, 0.06, 0.03, 0.01$ (details on how these were chosen can be found in Appendix B.2.1) and $n = 25, 50, 75, 100$. For each pair (α, n) in a single repetition we (i) sampled $\mathcal{S}_{\text{true}} \sim \text{Hollywood}(\alpha, -\alpha V, \nu)$, (ii) sampled n observations from an SIS($\mathcal{S}_{\text{true}}, \gamma_{\text{true}}$) model (iii) obtained $m = 250$ samples from the posterior, and (iv) evaluated summaries. For (ii) and (iii) we used exactly the same MCMC set-up as in the previous simulation.

Figure 4.4.3 summarises the output of 100 repetitions for each pair (α, n). For each α, we see values for $\bar{d}$ closer to zero as n grows, indicating concentration about the true mode. Furthermore, α shows no clear sign of impacting this concentration. Regarding the dispersion posterior mean $\bar{\gamma}$, as in the previous simulation we observe bias relative to the truth, with this bias reducing as n grows. Furthermore, this is the same across all α, with no clear sign that α affects the inference of these values.

4.4.3 Posterior predictive efficacy

A desirable feature of the posterior predictive is a growing resemblance of the true data generating distribution as the sample size increases. In this simulation, we considered exploring for such behaviour in the context of a missing data problem.

Suppose we have an observation $\mathcal{S}$ in which a single entry is missing, for example

$$\mathcal{S} = ((1, 2, 1, \bullet), (2, 3, 4, 3), (1, 2, 2, 1, 2, 3))$$

with $\bullet$ denoting the unknown entry. Towards predicting its value, let $\mathcal{S}_x$ denote the

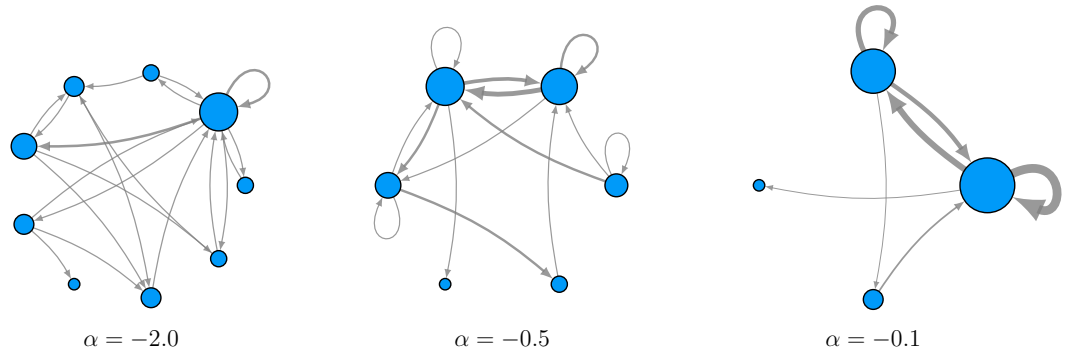


Figure 4.4.2: Visualising the role of α in the Hollywood model. Each plot shows an aggregate multigraph $\mathcal{G}_{\mathcal{S}}$ where $\mathcal{S} \sim \text{Hollywood}(\alpha, -\alpha V, \nu)$ with $V = 10$, $\nu = \text{TrPoisson}(3, 1, 10)$ and α varying. Edge thickness reflects edge multiplicity, whilst vertex size is proportional to $k_{\mathcal{S}}(v)$.

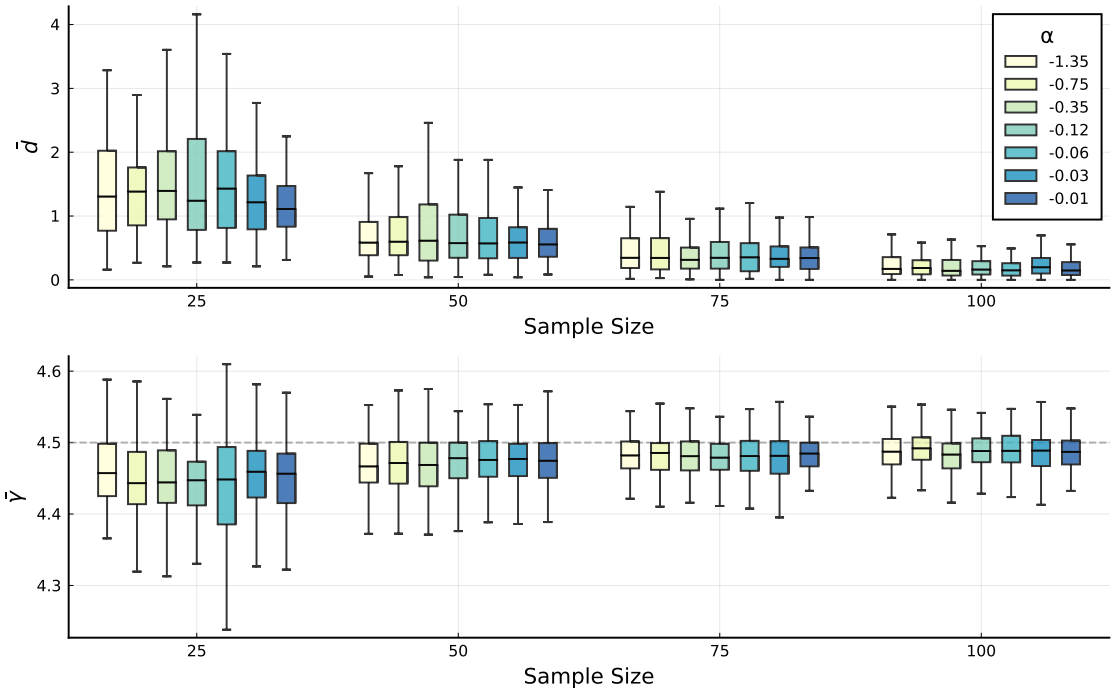


Figure 4.4.3: Summary of our second simulation (Section 4.4.2), where for each pair (α, n) the top subplot shows the distribution of $\bar{d}$, the average distance to the true mode, whilst the bottom shows the distribution of $\bar{\gamma}$, the posterior mean dispersion.

observation obtained by taking this entry to be $x \in \mathcal{V}$, that is

$$\mathcal{S}_x = ((1, 2, 1, x), (2, 3, 4, 3), (1, 2, 2, 1, 2, 3)),$$

and consider assigning a probability to each $x \in \mathcal{V}$ of being the true entry. If one knew $\mathcal{S} \sim \text{SIS}(\mathcal{S}^m, \gamma)$, then such a distribution could be obtained by comparing the relative probability of $\mathcal{S}_x$ for each $x \in \mathcal{V}$, in particular we could consider

$$p(x|\mathcal{S}^m, \gamma, \mathcal{S}_{-x}) := \frac{1}{Z(\mathcal{S}^m, \gamma, \mathcal{S}_{-x})} \exp\{-\gamma d_S(\mathcal{S}_x, \mathcal{S}^m)\}$$

with $Z(\mathcal{S}^m, \gamma, \mathcal{S}_{-x}) = \sum_{x \in \mathcal{V}} \exp\{-\gamma d_S(\mathcal{S}_x, \mathcal{S}^m)\}$ the normalising constant, where we introduce the notation $\mathcal{S}_{-x}$ to indicate that we are conditioning on the other known entries (and implicitly also on the dimensions of the observation). We refer to this as the *true predictive* for $x \in \mathcal{V}$.

In practice, with the true distribution unknown, one can instead leverage an observed sample $\{\mathcal{S}^{(i)}\}_{i=1}^n$ by averaging with respect to the posterior as follows

$$p(x|\{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x}) = \sum_{\mathcal{S}^m \in \boldsymbol{\mathcal{S}}^*} \int_{\mathbb{R}_+} p(x|\mathcal{S}^m, \gamma, \mathcal{S}_{-x}) p(\mathcal{S}^m, \gamma|\{\mathcal{S}^{(i)}\}_{i=1}^n) d\gamma,$$

defining the *posterior predictive* for $x \in \mathcal{V}$, which itself can be approximated using a sample $\{(\mathcal{S}_i^m, \gamma_i)\}_{i=1}^m$ from the posterior via

$$\hat{p}(x|\{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x}) := \frac{1}{m} \sum_{i=1}^m p(x|\mathcal{S}_i^m, \gamma_i, \mathcal{S}_{-x}),$$

a derivation of which can be found in Appendix B.2.2. To now predict x, one can for example take the maximum *a posteriori* (MAP) estimate

$$\hat{x} = \operatorname*{argmax}_{x \in \mathcal{V}} \hat{p}(x|\{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x}).$$

In this simulation, we considered assessing the agreement of the true and posterior predictive as n grows by examining how often their predictions were equal. We adopted the same modelling set-up as Section 4.4.1, jointly varying the dispersion and sample size, in this case considering $\gamma_{\text{true}} = 3.7, 4.2, 4.5, 4.9$ and $n = 25, 50, 75, 100$. However, in a slight deviation we here re-sampled the mode in each repetition from a fixed Hollywood model.

For a given pair $(\gamma_{\text{true}}, n)$ and a pre-specified number of test samples n_{test}, in a single repetition we (i) sampled mode $\mathcal{S}_{\text{true}} \sim \text{Hollywood}(\alpha, -\alpha V, \nu)$, with $\alpha = -0.35$ (V and ν as in Sections 4.4.1 and 4.4.2) (ii) sampled training and testing data $\{\mathcal{S}^{(i)}\}_{i=1}^{n+n_{\text{test}}}$ from an SIS$(\mathcal{S}_{\text{true}}, \gamma_{\text{true}})$ model, (iii) obtained a sample $\{(\mathcal{S}_i^m, \gamma_i)\}_{i=1}^m$ from the posterior $p(\mathcal{S}^m, \gamma | \{\mathcal{S}^{(i)}\}_{i=1}^n)$, that is, using the n training samples, (iv) for each $i = n+1, \ldots, n+n_{\text{test}}$ and for each entry of $\mathcal{S}^{(i)}$ (that is, each entry of each interaction) we assumed it to be missing and obtained predictions with both $\hat{p}(x | \{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x})$ and $p(x | \mathcal{S}^m, \mathcal{S}_{-x})$ via MAP estimates, and finally (v) returned the proportion of times these predictions were equal.[4] For (ii) and (iii) we used the same MCMC schemes as previous simulations.

Figure 4.4.4 summarises the output of 100 repetitions for each pair $(\gamma_{\text{true}}, n)$, with $n_{\text{test}} = 100$ in each repetition. For each γ_{true}, we see the predictions of the posterior predictive are more often in accordance with those of the true predictive distribution as the number of training samples increases. Moreover, when γ is lower, that is, the observed data is more variable, the discrepancy between the true and posterior predictive tends to be larger. Observe this is expected, given the observed behaviour of the first posterior concentration simulation (Section 4.4.1), wherein the posterior concentrated more slowly when γ was lower. In summary, the posterior predictive appears to better resemble the true data generating distribution as the sample size

[4]Note both the true and posterior predictive can have multiple values achieving the maximum defining the MAP estimate. To test for equality in these scenarios we thus compared the set of values achieving this maximum, whereby the two predictions would be considered equal if these sets were equal.

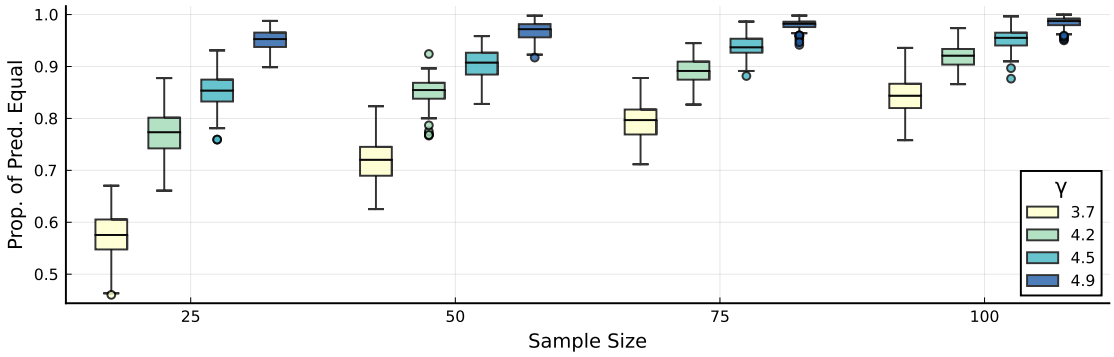


Figure 4.4.4: Summary of posterior predictive simulation (Section 4.4.3). Here we summarise the proportion of times the true and posterior predictions coincided when predicting missing entries of sampled test data, with boxplots showing the distribution of these proportions over 100 repetitions.

grows, as was expected.

4.5 Data analysis

In this section, the applicability of the proposed methodology will be illustrated via an example analysis of the Foursquare check-in data (Section 2.4.2). As mentioned in Section 4.1, an alternative approach to ours is to first aggregate observations to form graphs before applying a suitable graph-based method. As such, we compare our inference with some graph-based estimates. Note that in aggregating observations to form graphs one implicitly makes the assumption that the order of interaction arrival is irrelevant. Hence, for fairness we opt to make this comparison with our interaction-multiset model.

4.5.1 Data background and processing

For this analysis, we consider a version of the Foursquare data containing check-ins for users from New York and Tokyo (Yang et al., 2015b), focusing in particular on those in New York, looking at one month of check-in data over the period from 12 April to 12 May 2012.

As discussed in Section 2.4.2, here vertices correspond to venue categories, with a single interaction representing a sequence of check-ins made by a given user during a single day. However, compared with the analysis of Chapter 3, here a slightly different set of venue category labels was considered. In particular, note the venue category labels have a hierarchical structure, for example, the category "Jazz Club" is a subcategory of "Music Venue", which is itself a subcategory of "Arts & Entertainment". Those given by Yang et al. (2015b), and used in Chapter 3, were the lowest-level, that is, "Jazz Club" in the example above. In this analysis, the highest-level venue category was instead used, that is, "Arts & Entertainment" in this example.

The data was also further filtered prior to analysis. Firstly, it is clearly possible a user might only check-in to a single venue on a given day. Since our analysis is based on interaction multisets and concerns the movements of users *between* venue categories, such observations provide little information. Furthermore, such observations will be disregarded when aggregating to form graphs, and therefore would not feature in any of the graph-based approaches with which we intend to compare. As such, we considered only days where a user had checked-in to at least two venues. To further ensure each observation contained enough information, we considered only users with at least 10 observed days of check-ins. This left a total of 402 observations.

It was also necessary to filter these even further to avoid the inclusion of outliers which might lead to degenerative behaviour. In particular, it was seen that the inclusion of a few observations of significantly different size (for example, with many more interactions), or observations which shared little in common with the others, could result in an inferred mode that was empty, that is, an interaction network with no interactions. Clearly, such an inference provides little insight, making this an undesirable scenario. In addition, the MCMC scheme in such cases often showed poor mixing. For these reasons, we used the distance metric in the model fit (the matching distance in this case) to select a subset of 100 data points to analyse. In particular, we

used the normalised version of this distance, that is, via the Steinhaus transform (see Section 3.2), which ensured the observations that were selected shared something in common. For further details, see Appendix B.8.1.

4.5.2 SIM model fit

Following data processing, we were left with a sample of multisets $\{\mathcal{E}^{(i)}\}_{i=1}^{n}$, where each $\mathcal{E}^{(i)} = \left\{\mathcal{I}_1^{(i)}, \ldots, \mathcal{I}_{N^{(i)}}^{(i)}\right\}$ denotes the data of the ith user, with $\mathcal{I}_j^{(i)}$ denoting a single day of their check-ins. Recalling the inferential questions of interest outlined in Section 4.1, we now use our methodology to obtain (a) an average multiset of paths, and (b) a measure of variability.

In particular, using the Bayesian inference approach outlined in Appendix B.7, we fit our SIM model to these data. We made use of the matching distance d_{M}, with the LSP distance d_{LSP} between paths. As seen in Section 4.2.3, a consequence of assuming this distance is that our inferred mode will contain paths often appearing as subpaths in the observed data. For our priors, we assumed $\mathcal{E}^m \sim \mathrm{SIM}(\hat{\mathcal{E}}, 3.0)$, with $\hat{\mathcal{E}}$ denoting the sample Fréchet mean of the observed data $\{\mathcal{E}^{(i)}\}_{i=1}^{n}$, whilst we assumed $\gamma \sim \mathrm{Gamma}(5, 1.67)$. Via our MCMC scheme, we then obtained a sample $\{(\mathcal{E}_i^m, \gamma_i)\}_{i=1}^{M}$, from the posterior $p(\mathcal{E}^m, \gamma | \{\mathcal{E}^{(i)}\}_{i=1}^{n})$, obtaining a total of $M = 500$ samples with a burn-in period of 25,000 and taking a lag of 50 between samples.

Given the posterior sample $\{(\mathcal{E}_i^m, \gamma_i)\}_{i=1}^{M}$, we subsequently obtained point estimates $(\hat{\mathcal{E}}^m, \hat{\gamma})$, with the mode estimate $\hat{\mathcal{E}}^m$ functioning as our desired average, and $\hat{\gamma}$ a measure of data variability. In particular, we considered the following

$$\hat{\mathcal{E}}^m = \underset{\mathcal{E} \in \{\mathcal{E}_i^m\}_{i=1}^{M}}{\operatorname{argmin}} \sum_{i=1}^{M} d_{\mathrm{M}}^2(\mathcal{E}_i^m, \mathcal{E}) \qquad \hat{\gamma} = \frac{1}{M} \sum_{i=1}^{M} \gamma_i$$

that is, the Fréchet mean for the mode and the arithmetic mean for the dispersion, both obtained from their respective posterior samples.

As mentioned, due to our choice of distance, the inferred mode $\hat{\mathcal{E}}^m$ represents a collection of pathways frequently seen together in the observed data. To visualise this, we consider plotting the paths of $\hat{\mathcal{E}}^m$ alongside those of its two nearest observations. Supposing the data points have been labelled such that $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$ denote the first and second nearest neighbours of $\hat{\mathcal{E}}^m$ with respect to d_{M}, writing these as follows

$$\hat{\mathcal{E}}^m = \left\{\hat{\mathcal{I}}_1^m, \ldots, \hat{\mathcal{I}}_{N^m}^m\right\} \quad \mathcal{E}^{(1)} = \left\{\mathcal{I}_1^{(1)}, \ldots, \mathcal{I}_{N_1}^{(1)}\right\} \quad \mathcal{E}^{(2)} = \left\{\mathcal{I}_1^{(2)}, \ldots, \mathcal{I}_{N_2}^{(2)}\right\},$$

Figures 4.5.1 to 4.5.3 visualise the paths of $\hat{\mathcal{E}}^m$, $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$ alongside one another. In each, the paths of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$ have been aligned in accordance with the optimal matching found when evaluating their distance from $\hat{\mathcal{E}}^m$ via d_{M}. In particular, in the jth row we plot $\hat{\mathcal{I}}_j^m$ alongside $\mathcal{I}_j^{(1)}$ and $\mathcal{I}_j^{(2)}$, denoting the paths matched to $\hat{\mathcal{I}}_j^m$ when evaluating $d_{\mathrm{M}}(\hat{\mathcal{E}}^m, \mathcal{E}^{(1)})$ and $d_{\mathrm{M}}(\hat{\mathcal{E}}^m, \mathcal{E}^{(2)})$, respectively. The paths of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$ not matched to any of $\hat{\mathcal{E}}^m$ are then shown in the remaining rows.

Here one can observe paths of $\hat{\mathcal{E}}^m$ do indeed appear as subpaths within those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$. In fact, in first three rows of Figure 4.5.1 they are equivalent, that is $\hat{\mathcal{I}}_j^m = \mathcal{I}_j^{(1)} = \mathcal{I}_j^{(2)}$, whilst for the remaining rows of Figure 4.5.1 and those of Figures 4.5.2 and 4.5.3 we begin to see differences in the observed paths relative to those of the estimated mode, however, in almost all cases, the paths in the mode $\hat{\mathcal{I}}_j^m$ continue to feature as subpaths of both $\mathcal{I}_j^{(1)}$ and $\mathcal{I}_j^{(2)}$. Note also that no paths in $\hat{\mathcal{E}}^m$ are of length greater than two. At face value this might seem to imply use of this method gains nothing over a graph-based approach. However, as we illustrate in the next section, the subtle difference is that our inference is unambiguous.

It it worth noting that as an alternative to $\hat{\mathcal{E}}^m$, one could instead use the sample Fréchet mean $\hat{\mathcal{E}}$ directly as a summary. However, when the data is quite variable this sample mean, being itself an observation, is likely to differ in some ways from the other observations. In contrast, the inferred mode $\hat{\mathcal{E}}^m$ contains only paths which have appeared in some manner within many observations. In this way, our estimate

$\hat{\mathcal{E}}^m$ can be seen as a pruned version of $\hat{\mathcal{E}}$, where the common theme present in all observations has been extracted.

For the dispersion, we have $\hat{\gamma} \approx 3.02$, with a trace-plot of the posterior samples $\{\gamma_i\}_{i=1}^M$ from which this estimate was obtained shown in the left-hand plot of Figure 4.5.4. To aide interpretation of $\hat{\gamma}$, the right hand plot of Figure 4.5.4 visualises the distribution of $d_{\mathrm{M}}(\mathcal{E}, \hat{\mathcal{E}}^m)$ where $\mathcal{E} \sim \mathrm{SIM}(\hat{\mathcal{E}}^m, \gamma)$ for different values of γ, each boxplot summarising 1,000 samples drawn from the respective multiset model via our iMCMC algorithm (Appendix B.7.5). A comparison with our estimate $\hat{\gamma}$ shows that we expect the distance of samples to the mode to be around 25 (from $\gamma = 3.0$), which, since we used the matching distance, can be seen as the average number of edit operations required to transform the mode into an observation. Considering the mode has 18 entries in total (9 paths of length two), this implies a reasonable amount of variability in the observed data.

4.5.3 Comparison with graph-based inferences

As alluded to already, an alternative to applying our methodology is to apply a graph-based method to aggregated observations. As such, we consider striking a comparison between this approach and ours. The intention here is twofold. On one hand, to show the graph-based inferences are not too dissimilar from that obtained via our approach. Whilst on the other, that our approach goes beyond the graph-based methods, in so far as producing an inference which is unambiguous regarding the presence of higher-order information in the observed data.

Given the observed sample $\{\mathcal{E}^{(i)}\}_{i=1}^n$, one can obtain a sample of graphs $\{\mathcal{G}^{(i)}\}_{i=1}^n$ via aggregation, namely, by letting $\mathcal{G}^{(i)} = \mathcal{G}_{\mathcal{E}^{(i)}}$, the multigraph obtained by aggregating the paths of $\mathcal{E}^{(i)}$, as outlined in Section 2.4.1. Note some multiple-network methods require graphs, not multigraphs. However, a graph can be obtained naturally from a multigraph by simply removing edge multiplicities, including an edge

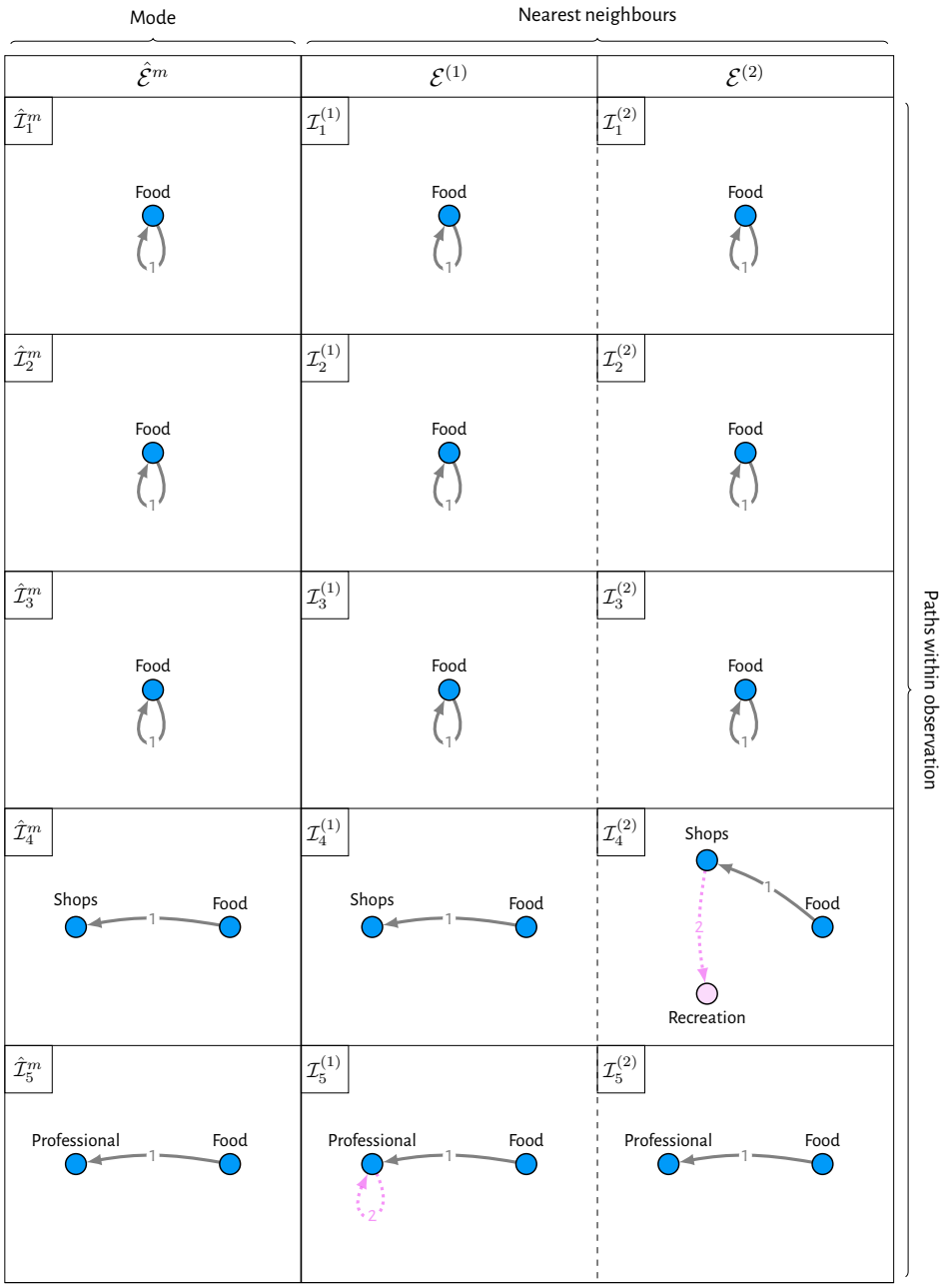


Figure 4.5.1: A subset of paths from our point estimate $\hat{\mathcal{E}}^m$ for the Foursquare data, alongside those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$, its two nearest neighbours. Paths are aligned according to the optimal matching found when evaluating $d_{\mathrm{M}}(\hat{\mathcal{E}}^m, \mathcal{E}^{(i)})$ for each neighbour $\mathcal{E}^{(i)}$. For each observed path $\mathcal{I}_j^{(i)}$, dashed pink edges and pink vertices indicate differences with $\hat{\mathcal{I}}_j^m$, with edges labels indicating the order of vertex visits. The remaining paths can be seen in Figures 4.5.2 and 4.5.3.

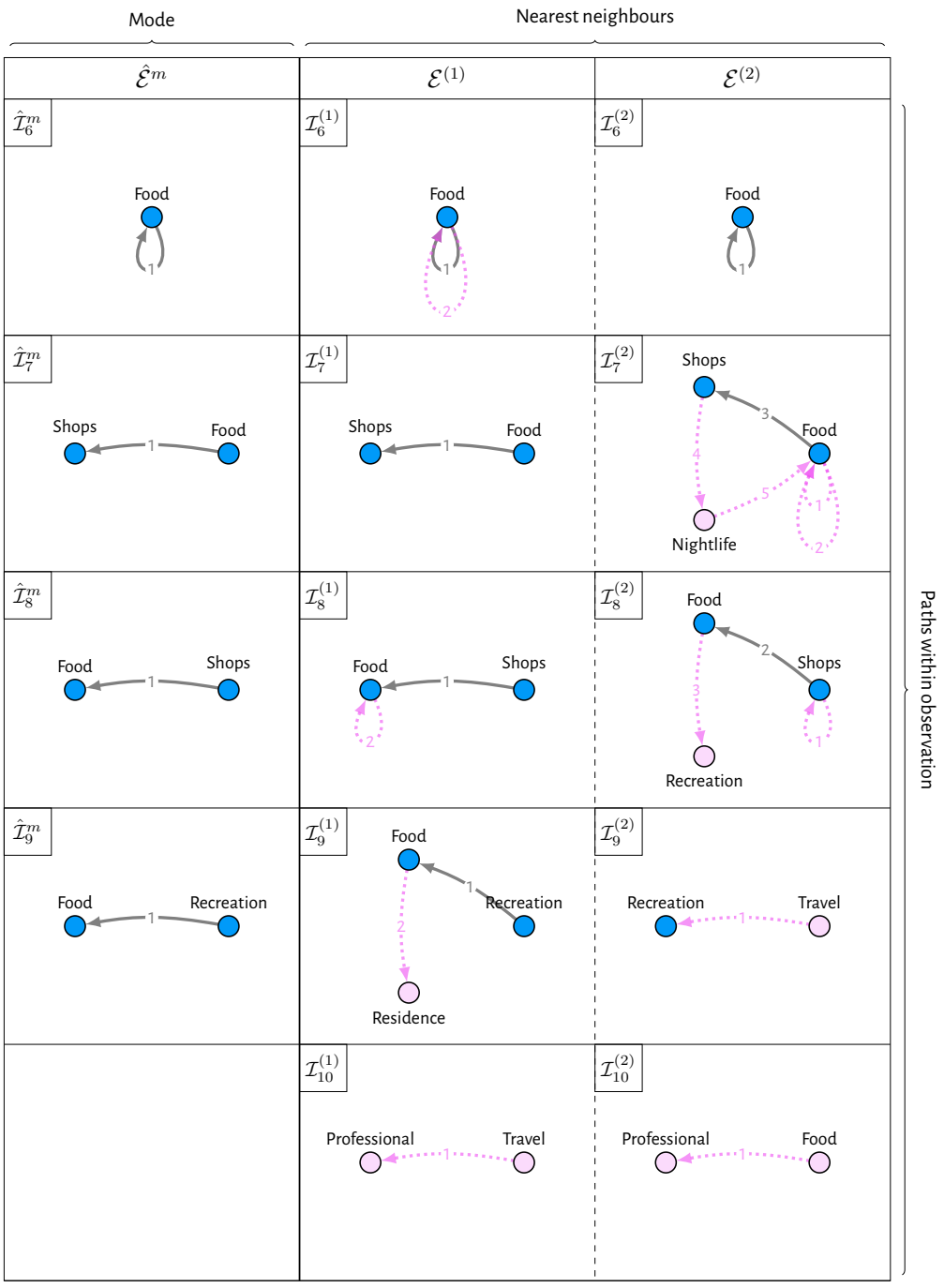


Figure 4.5.2: Paths of our point estimate $\hat{\mathcal{E}}^m$ for the Foursquare data, alongside those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$, its two nearest neighbours. The remaining paths can be seen in Figures 4.5.1 and 4.5.3.

Figure 4.5.3: Paths of our point estimate $\hat{\mathcal{E}}^m$ for the Foursquare data, alongside those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$, its two nearest neighbours. The remaining paths can be seen in Figures 4.5.1 and 4.5.2.

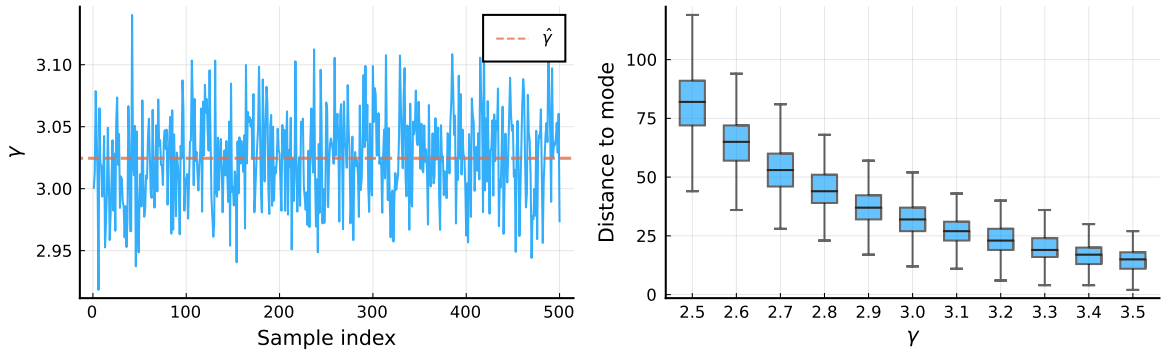


Figure 4.5.4: Summary of inference for the dispersion for the Foursquare data. Left shows a trace-plot of the posterior samples $\{\gamma_i\}_{i=1}^m$, whilst the right plot summarises the distribution of distances to the inferred mode for different values of γ, aiding interpretation of our estimate $\hat{\gamma}$.

if it was observed at least once. In what follows, both aggregation schemes will be considered.

In the same way that our estimate $\hat{\mathcal{E}}^m$ summarises the sample $\{\mathcal{E}^{(i)}\}_{i=1}^n$, one can consider obtaining a multigraph or graph $\hat{\mathcal{G}}$ which summarises the sample $\{\mathcal{G}^{(i)}\}_{i=1}^n$. This can be achieved through a variety of different approaches, the choice of which will depend on whether the $\mathcal{G}^{(i)}$ are graphs or multigraphs.

In the case where each $\mathcal{G}^{(i)}$ is a graph, each associated adjacency matrix $A^{\mathcal{G}^{(i)}}$ will be a binary matrix. With this, a simple model-free summary of this sample of graphs is the majority vote, which we denote $\hat{\mathcal{G}}_{\text{MV}}$, where an edge is included if it was observed in at least one half of the observations. More formally, $\hat{\mathcal{G}}_{\text{MV}}$ can be defined in terms of its adjacency matrix as follows

$$A_{ij}^{\hat{\mathcal{G}}_{\text{MV}}} = \mathbb{1}(\bar{A}_{ij} \geq 0.5),$$

where $\bar{A}$ is the real-valued matrix with entries $\bar{A}_{ij} = \frac{1}{n}\sum_{k=1}^n A_{ij}^{\mathcal{G}^{(k)}}$, that is, the entry-wise average of the observed adjacency matrices. As a model-based alternative, we turn to the centered Erdös-Rényi (CER) model of Lunagómez et al. (2021), which is defined as follows. Given graph $\mathcal{G}^m$ and parameter $0 < \alpha < 0.5$ we say $\mathcal{G} \sim \text{CER}(\mathcal{G}^m, \alpha)$ if the graph $\mathcal{G}$ was sampled, via its adjacency matrix, as follows

$$A_{ij}^{\mathcal{G}} \,|\, A_{ij}^{\mathcal{G}^m}, \alpha = |A_{ij}^{\mathcal{G}^m} - Z_{ij}| \quad \text{where } Z_{ij} \overset{\text{i.i.d.}}{\sim} \text{Bern}(\alpha),$$

that is, for each possible edge in $\mathcal{G}^m$ with probability α we flip it, that is, either remove it, or insert it if not present. Using this, we assumed the following hierarchical model

$$\begin{aligned}
\mathcal{G}^{(i)} \,|\, \mathcal{G}^m, \alpha &\sim \text{CER}(\mathcal{G}^m, \alpha) \quad (\text{for } i = 1, \ldots, n) \\
\mathcal{G}^m &\sim \text{CER}(\mathcal{G}_0, \alpha_0) \\
\alpha &\sim 0.5 \cdot \text{Beta}(\beta_1, \beta_2)
\end{aligned}$$

where $\mathcal{G}_0$ (a graph), $0 \leq \alpha_0 \leq 0.5$, $\beta_1 > 0$ and $\beta_2 > 0$ denote hyperparameters. For this analysis, we assumed $\mathcal{G}_0 = \hat{\mathcal{G}}_{\text{MV}}$ and $\alpha_0 = 0.5$, leading to a uniform distribution over the space of graphs for the prior on $\mathcal{G}^m$, whilst we took $\beta_1 = \beta_2 = 1$, similarly leading to the uniform distribution over the interval $(0, 0.5)$ for the prior on α. Following the scheme of Lunagómez et al. (2021), we drew a sample $\{(\mathcal{G}_i^m, \alpha_i)\}_{i=1}^M$ from the posterior $p(\mathcal{G}^m, \alpha | \{\mathcal{G}^{(i)}\}_{i=1}^n)$ via MCMC, obtaining the desired summary via the sample Fréchet mean

$$\hat{\mathcal{G}}_{\text{CER}} = \underset{\mathcal{G} \in \{\mathcal{G}_i^m\}}{\operatorname{argmin}} \sum_{i=1}^{n} d_{\text{H}}^2(\mathcal{G}, \mathcal{G}_i^m)$$

where d_{H} denotes the Hamming distance between graphs (Lunagómez et al., 2021; Donnat and Holmes, 2018). Figures 4.5.5a and 4.5.5b show these two graph summaries, $\hat{\mathcal{G}}_{\text{CER}}$ and $\hat{\mathcal{G}}_{\text{MV}}$, respectively, for the Foursquare data, where it transpires that $\hat{\mathcal{G}}_{\text{CER}} = \hat{\mathcal{G}}_{\text{MV}}$.

In the case where each $\mathcal{G}^{(i)}$ is a multigraph, and thus each $A^{\mathcal{G}^{(i)}}$ is a matrix of non-negative integers, an analogous model-free summary can be obtained by rounding the entries of $\bar{A}$ to the nearest integer. Referring to this as the rounded mean estimate and denoting it $\hat{\mathcal{G}}_{\text{RM}}$, it can be defined formally via its adjacency matrix as follows

$$A_{ij}^{\hat{\mathcal{G}}_{\text{RM}}} = \lfloor \bar{A}_{ij} \rfloor + \mathbb{1}(\bar{A}_{ij} - \lfloor \bar{A}_{ij} \rfloor \geq 0.5),$$

where the notation $\lfloor x \rfloor$ for $x \in \mathbb{R}$ denotes the floor function. As a model-based approach, we consider using the SNF model proposed by Lunagómez et al. (2021). Though originally proposed to model graphs, it can be readily extended to handle multigraphs (see Appendix B.8.2). Use of the SNF, like our models, requires specification of a distance metric between graphs. We considered the Hamming distance, as defined in Section 3.3, that is

$$d_{\text{H}}(\mathcal{G}, \mathcal{G}') = \sum_{i,j} \left| A_{ij}^{\mathcal{G}} - A_{ij}^{\mathcal{G}'} \right|,$$

which essentially counts the number of edges *not* shared by the two multigraphs. Adopting the notation $\mathcal{G} \sim \text{SNF}(\mathcal{G}^m, \gamma)$ when a graph $\mathcal{G}$ is drawn from the SNF model with mode $\mathcal{G}^m$ (a multigraph) and dispersion $\gamma > 0$, we assumed the following hierarchical model

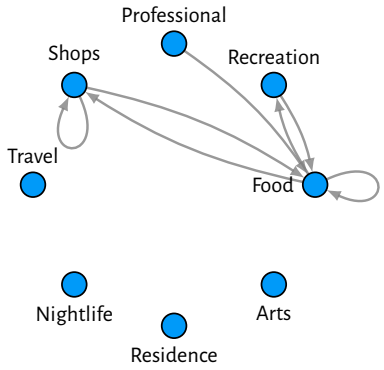
$$\begin{aligned} \mathcal{G}^{(i)} \,|\, \mathcal{G}^m, \gamma &\sim \text{SNF}(\mathcal{G}^m, \gamma) \quad (\text{for } i = 1, \ldots, n) \\ \mathcal{G}^m &\sim \text{SNF}(\mathcal{G}_0, \gamma_0) \\ \gamma &\sim \text{Gamma}(\alpha, \beta) \end{aligned}$$

where $\mathcal{G}_0$ (a multigraph), $\gamma_0 > 0$, $\alpha > 0$ and $\beta > 0$ are hyperparameters. For this analysis, we took $\mathcal{G}_0$ to be the sample Fréchet mean of the observed multigraphs $\{\mathcal{G}^{(i)}\}_{i=1}^n$ with respect to the distance d_1, whilst we let $\gamma_0 = 0.1$, $\alpha = 3$ and $\beta = 1$. Again, we obtained a sample $\{(\mathcal{G}_i^m, \gamma_i)\}_{i=1}^M$ from the posterior $p(\mathcal{G}^m, \gamma | \{\mathcal{G}^{(i)}\}_{i=1}^n)$ via MCMC, before invoking the sample Fréchet mean to obtain the desired summary

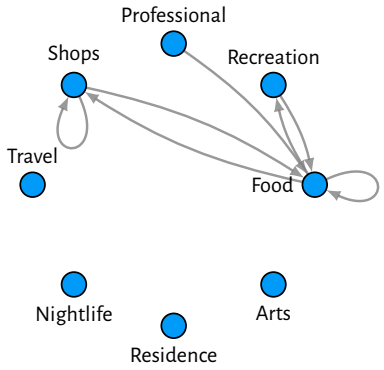
$$\hat{\mathcal{G}}_{\text{SNF}} = \underset{\mathcal{G} \in \{\mathcal{G}^{(i)}\}_{i=1}^n}{\text{argmin}} \sum_{i=1}^n d_1^2(\mathcal{G}, \mathcal{G}_i^m).$$

Note that the posterior here will be doubly-intractable, necessitating use of a specialised MCMC algorithm. Lunagómez et al. (2021) adopted the algorithm of Møller et al. (2006), however, since here we consider multigraphs, we cannot apply their scheme directly. Instead, we took an alternative approach via the exchange algorithm (Murray et al., 2006), details of which can be found in Appendix B.8.2. Visualisations of these two multigraph summaries, $\hat{\mathcal{G}}_{\text{SNF}}$ and $\hat{\mathcal{G}}_{\text{RM}}$, can be seen in Figures 4.5.5c and 4.5.5d, respectively.

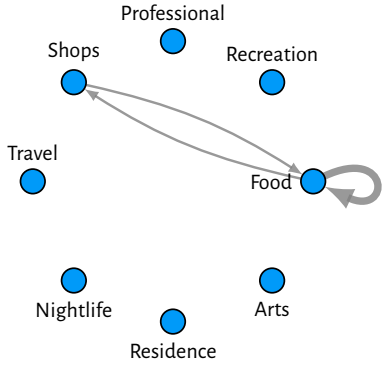
Comparing the graph-based methods amongst themselves, we see a slight variation in the signal they uncover. For example, in taking edge multiplicities into account, the multigraph-based estimate $\hat{\mathcal{G}}_{\text{RM}}$ introduces edges which did not appear in either of the graphs $\hat{\mathcal{G}}_{\text{CER}}$ and $\hat{\mathcal{G}}_{\text{MV}}$, generally involving the node corresponding to food venues. Conversely, the SNF-based estimate $\hat{\mathcal{G}}_{\text{SNF}}$ appears to instead disregard



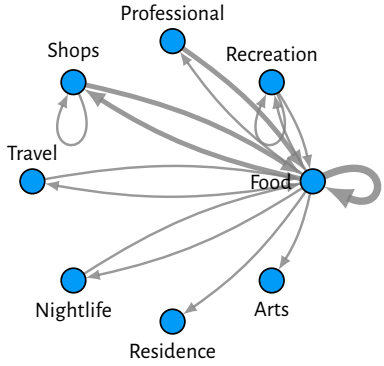
(a) $\hat{\mathcal{G}}_{\text{CER}}$



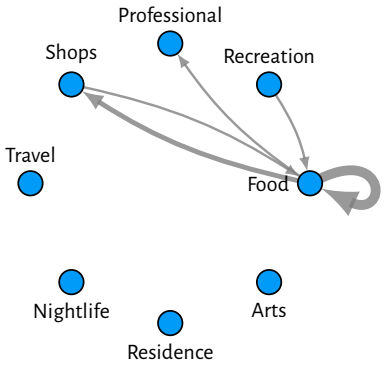
(b) $\hat{\mathcal{G}}_{\text{MV}}$



(c) $\hat{\mathcal{G}}_{\text{SNF}}$



(d) $\hat{\mathcal{G}}_{\text{RM}}$



(e) $\mathcal{G}_{\hat{\mathcal{E}}^m}$

Figure 4.5.5: A comparison with graph-based inferences. Here (e) shows $\mathcal{G}_{\hat{\mathcal{E}}^m}$, the aggregate multigraph of our point estimate $\hat{\mathcal{E}}^m$ of Section 4.5.2, whilst (a)-(d) show alternative inferences obtained via graph-based approaches outlined in Section 4.5.3. Note that (a) and (b) are graphs, whilst (c)-(e) are multigraphs, with edge thickness proportional to their weight.

edges which appear in $\hat{\mathcal{G}}_{\text{CER}}$ and $\hat{\mathcal{G}}_{\text{MV}}$.

To compare these summaries with our estimate $\hat{\mathcal{E}}^m$ one can consider *its* aggregate multigraph $\mathcal{G}_{\hat{\mathcal{E}}^m}$, as shown in Figure 4.5.5e. Observe this appears to sit somewhere in between the multigraph and graph summaries, appearing to have some degree of similarity with each, as we intended to confirm. Moreover, a common theme seems to appear (for all summaries). Namely, that visits to food venues feature strongly, often followed or preceded by a visit to another food venue or some other venue category, with shopping venues being a prevalent choice.

Naturally, one is inclined to ask if the aggregation of our estimate $\mathcal{G}_{\hat{\mathcal{E}}^m}$ is not too dissimilar to these other graph-based summaries, then what does one gain by taking our approach? Recalling that $\hat{\mathcal{E}}^m$ denotes a multiset of paths, we argue this contains more information regarding the signals present in the observed data than any graph-based summary, assuming the data were truly path-observed. This comes back to a point made in Section 4.1, namely that when one aggregates paths a loss of information is incurred, which will in turn limit the conclusions one can draw concerning the original data. For example, consider the CER-based summary $\hat{\mathcal{G}}_{\text{CER}}$ of Figure 4.5.5a, where we see the following two edges

$$e_1 = (\text{recreation}, \text{food}) \quad e_2 = (\text{food}, \text{shops}).$$

This could imply at least two things. Perhaps many users went from a recreation venue to a food venue, and separately, that is, on a different day, from a food venue to a shopping venue. Alternatively, maybe many users traced the path recreation $\rightarrow$ food $\rightarrow$ shops in a single day. Both are possibilities, and from a graph-based summary there is no way of knowing which is the case. However, in directly estimating a collection of paths, we can make such distinctions. For example, considering our estimate $\hat{\mathcal{E}}^m$ for the Foursquare data (Figures 4.5.1 to 4.5.3), it appears we are in the former case, since no paths therein are of length greater than two.

4.6 Discussion

To summarise, in this chapter a novel Bayesian modelling framework capable of analysing samples of interaction networks has been proposed, without the need to perform any aggregation of observations. This has been supplemented with specialised MCMC schemes, facilitating inference for the proposed models. Through simulation studies, efficacy of the methodology and inference scheme have been confirmed, whilst its applicability has been illustrated via an analysis of the Foursquare check-in data, highlighting how answers to inferential questions (a) and (b) posed in Section 4.1 can be obtained. Moreover, in comparing with graph-based methods we highlighted the extra information one subtly gains by taking our approach.

Regarding future work, there are a few ways one might consider building upon what has been proposed here. Firstly, a natural extension is to consider a mixture model, with our SIS or SIM models functioning as mixture components, which would allow one to capture heterogeneity in the observations and provide an model-based approach to clustering interaction networks.

Secondly, on a more pragmatic note, one could also take steps to scale-up our approach computationally. For example, one might be able to circumvent the need to use the exchange algorithm if the normalising constant for a particular distance metric was derived, as was the case for the CER model in Lunagómez et al. (2021).

One could also consider alternative modelling approaches. For example, if one is able to make an exchangeability assumption for each observation, that is, the order in which paths arrive is not of interest, then a model reminiscent of the latent Dirichlet allocation (LDA) model (Blei et al., 2003) would be a possibility. This would assume each observation was drawn from some mixture distribution over paths (as in the simulation studies of Chapter 3), with mixture components being shared between observations but mixture proportions differing. This would also have a natural non-parametric extension via the hierarchical Dirichlet process (HDP) (Teh et al., 2006).

It would be interesting to see how the inferences from such an approach compare with ours, at least qualitatively, and whether there would be any computational benefits. More tangentially, one could also follow the path laid in the wider literature on multiple networks and consider extending models designed to analyse a single interaction network, for example, the models of Crane and Dempsey (2018) or Williamson (2016).

Finally, recall from Section 4.5.1 that to avoid unwanted degenerative behaviour it was necessary to remove outliers from the data before using the proposed methodology, in particular, those of vastly different size or that shared little in common with the other observations. This is a somewhat crude approach, and relies heavily on how one determines which observations are outliers. As such, it would be interesting to explore whether a more reasoned approach could be taken by instead handling these outliers within the modelling framework. An approach might be to assume that observations were either drawn from an SIS or SIM model with some probability β, or from some other outlier model with probability $(1-\beta)$, where this outlier model need not be an SIS or SIM model. Alternatively, perhaps a different modelling approach, such as those mentioned in the previous paragraph, is better suited to handling such outliers.

Chapter 5

Conclusions

In this thesis, the currently unconsidered problem of analysing samples of interaction networks has been addressed. In particular, through the work outlined in Chapters 3 and 4, methods have been proposed which can provide answers to the questions posed in Chapter 1. Namely, in Chapter 3 a variety of distance measures which can be used to compare interaction networks were proposed, which, as was illustrated via example data analyses, can be used to both cluster networks and predict network-level covariate information. Building upon this work, a novel modelling framework was then proposed in Chapter 4, leading to a statistically reasoned approach to the problem of summarising a sample of interaction networks.

5.1 Limitations

Of course, any method is not without its limitations, with those proposed here being no exception. Of prominence in this regard would be the complexity that has been introduced, both conceptually and computationally. In terms of computational cost, this includes both the distances proposed in Chapter 3, all of which involved solving some form of optimisation problem during evaluation, and the computationally intensive MCMC scheme required to fit the models proposed in Chapter 4. This limits

the size of datasets that can be analysed. In addition, the model introduced in Chapter 4 is itself arguably conceptually complex and hard to interpret, in turn making communication of results a challenge. A driver for this is the manner in which assumptions of a given model are often subtly wrapped-up in the choice of distance, with distances that may serve uses in other tasks being unsuitable. For example, as alluded to in Section 4.2.2, it is not recommend to use the fixed-penalty matching and edit distances, though they appeared to perform well in the simulation studies of Chapter 3.

A second limitation regards sensitivity to outliers. As seen in the data analysis of Chapter 4, before using the proposed methodology care had to be taken to correctly filter the data (see Section 4.5.1), necessary since this was sensitive to the inclusion of observations which differed significantly from the rest. This included the presence of observations which were of very different size, perhaps with only a few or very many interactions, and observations which shared very little in common with the others. Such sensitivities thus limit the applicability of the proposed approach.

5.2 Future work

Aside from the proposals made in the discussions of Chapters 3 and 4, there are some further ways in which the work of this thesis could be built upon.

Firstly, related to the limitations alluded to above regarding sensitivity to outliers, it would be interesting to consider modelling approaches which can suitably handle the scenario where observations have a very different number of interactions. For example, considering the Foursquare data, there may be very many users with only a few observed interactions (days of check-ins), alongside users with very many. In a way, one would expect to know more about the latter, since one has more data, whilst there would be more uncertainty for users with only a few interactions. Considering

how one might handle having such varying observation-wise uncertainty would be an interesting avenue to explore.

One could also consider modelling approaches that take into account covariate information at the level of networks. This would, for example, provide an alternative to the distance-based approach seen in the Foursquare analysis of Chapter 3. Methods that could provide deeper insights or be more interpretable than such a non-parametric approach would be particularly useful.

Finally, note for both datasets considered in this thesis it would be possible to obtain the time at which each interaction was observed. For example, in the Foursquare data this might be the specific day on which it was observed, whilst in the football data this would be the time during the match. This could motivate more involved modelling approaches that take into account such temporal information. With this, one could then consider questions regarding the interdependence between time and the structure of observed interactions.

Appendix A

Appendix to Chapter 3

A.1 Deriving Jaccard distances

In this section, we derive the Jaccard distance between vectors of counts and multigraphs defined in Section 3.3. Recall, given an interaction network $\mathcal{S}$ over vertex set $\mathcal{V}$ with $V = |\mathcal{V}|$, we have $v^{\mathcal{S}} \in \mathbb{Z}_{\geq 0}^{V}$ denoting the vector of vertex counts, $A^{\mathcal{S}} \in \mathbb{Z}_{\geq 0}^{V \times V}$ the matrix of vertex traversals, and $\mathcal{G}_{\mathcal{S}}$ the multigraph defined by $A^{\mathcal{S}}$.

We first derive $d_{\mathrm{J}}(v^{\mathcal{S}}, v^{\mathcal{S}'})$ by applying the Steinhaus transform (see Section 3.2) to the Hamming distance d_{H} defined in Section 3.3, using $c = \mathbf{0}_V$, where $\mathbf{0}_V \in \mathbb{Z}_{\geq 0}^{V}$ denotes the vector of zeros, as the reference element

$$\begin{aligned} d_{\mathrm{J}}(v^{\mathcal{S}}, v^{\mathcal{S}'}) &= \frac{2d_{\mathrm{H}}(v^{\mathcal{S}}, v^{\mathcal{S}'})}{d_{\mathrm{H}}(v^{\mathcal{S}}, \mathbf{0}_V) + d_{\mathrm{H}}(v^{\mathcal{S}'}, \mathbf{0}_V) + d_{\mathrm{H}}(v^{\mathcal{S}}, v^{\mathcal{S}'})} \\ &= \frac{2\sum_{x \in \mathcal{V}} |v_x^{\mathcal{S}} - v_x^{\mathcal{S}'}|}{\sum_{x \in \mathcal{V}} |v_x^{\mathcal{S}}| + \sum_{x \in \mathcal{V}} |v_x^{\mathcal{S}'}| + \sum_{x \in \mathcal{V}} |v_x^{\mathcal{S}} - v_x^{\mathcal{S}'}|} \\ &= \frac{\sum_{x \in \mathcal{V}} |v_x^{\mathcal{S}} - v_x^{\mathcal{S}'}|}{\sum_{x \in \mathcal{V}} \max(v_x^{\mathcal{S}}, v_x^{\mathcal{S}'})} \end{aligned}$$

where here we have used the identity $|x| + |y| + |x - y| = 2 \cdot \max(x, y)$ for any $x, y \in \mathbb{R}$.

The same approach can be applied to derive $d_{\mathrm{J}}(\mathcal{G}_{\mathcal{S}}, \mathcal{G}_{\mathcal{S}'})$, the Jaccard distance between multigraphs. The reference element in this case will be the empty multigraph

$\mathcal{G}_0$, that is, the multigraph with no edges, which is encoded via the matrix of zeros $\mathbf{0}_{V,V} \in \mathbb{Z}_{\geq 0}^{V \times V}$. This leads to following

$$\begin{aligned} d_{\mathrm{J}}(\mathcal{G}_{\mathcal{S}}, \mathcal{G}_{\mathcal{S}'}) &= \frac{2d_{\mathrm{H}}(\mathcal{G}_{\mathcal{S}}, \mathcal{G}_{\mathcal{S}'})}{d_{\mathrm{H}}(\mathcal{G}_{\mathcal{S}}, \mathcal{G}_0) + d_{\mathrm{H}}(\mathcal{G}_{\mathcal{S}'}, \mathcal{G}_0) + d_{\mathrm{H}}(\mathcal{G}_{\mathcal{S}}, \mathcal{G}_{\mathcal{S}'})} \\ &= \frac{2\sum_{x\in\mathcal{V}}\sum_{y\in\mathcal{V}} |A^{\mathcal{S}}_{xy} - A^{\mathcal{S}'}_{xy}|}{\sum_{x\in\mathcal{V}}\sum_{y\in\mathcal{V}} |A^{\mathcal{S}}_{xy}| + \sum_{x\in\mathcal{V}}\sum_{y\in\mathcal{V}} |A^{\mathcal{S}'}_{xy}| + \sum_{x\in\mathcal{V}}\sum_{y\in\mathcal{V}} |A^{\mathcal{S}}_{xy} - A^{\mathcal{S}'}_{xy}|} \\ &= \frac{\sum_{x\in\mathcal{V}}\sum_{y\in\mathcal{V}} |A^{\mathcal{S}}_{xy} - A^{\mathcal{S}'}_{xy}|}{\sum_{x\in\mathcal{V}}\sum_{y\in\mathcal{V}} \max(A^{\mathcal{S}}_{xy}, A^{\mathcal{S}'}_{xy})} \end{aligned}$$

where we again invoke the identity $|x|+|y|+|x-y| = 2\cdot\max(x,y)$ for any $x, y \in \mathbb{R}$.

A.2 Fixed penalties for the matching and edit distances

Both $d_{\mathrm{M},\rho}$ and $d_{\mathrm{E},\rho}$ require specifying the parameter $\rho > 0$, denoting the fixed penalty to be incurred for any interactions which are left unmatched. In this section, we provide guidance on how one can consider setting this parameter in practice.

Our recommendation is the same for both the fixed-penalty edit and matching distances. Observe, within either distance, when optimising over matchings one will compare the following two scenarios

that is, for some pair of interactions $\mathcal{I}$ and $\mathcal{I}'$, we could choose to match them or leave them both unmatched, where the highlighted terms below show the cost contribution in each case. With this, we can see if

$$d_I(\mathcal{I}, \mathcal{I}') > 2\rho$$

then it would actually be better, that is, less costly, to leave these two interactions

unmatched. Following the rationale that we want to maximise the pairwise information of interactions, we would ideally like to avoid scenarios where interactions are left unmatched. As such, in want of avoiding the case above, this implies the following lower bound on ρ

$$\rho \geq \frac{d_I(\mathcal{I}, \mathcal{I}')}{2},$$

which we would ideally like to hold for any $\mathcal{I} \in \boldsymbol{\mathcal{I}}^*$ and $\mathcal{I}' \in \boldsymbol{\mathcal{I}}^*$. Note for the fixed-penalty matching distance $d_{\mathrm{M},\rho}$ this bound can also be obtained by considering those values of ρ for which a complete optimal matching is guaranteed to exist (see Proposition A.3.1).

Notice achieving this lower bound on ρ is possible provided the distance $d_I(\cdot, \cdot)$ is bounded. In particular, if there exists some $K > 0$ such that $d_I(\mathcal{I}, \mathcal{I}') \leq K$ for all $\mathcal{I} \in \boldsymbol{\mathcal{I}}^*$ and $\mathcal{I}' \in \boldsymbol{\mathcal{I}}^*$, then we will have the bound $\rho \geq K/2$. Moreover, observe that as $\rho \to \infty$ the distances $d_{\mathrm{M},\rho}$ and $d_{\mathrm{E},\rho}$ will be driven more by penalisation terms and less by the pairwise distances of matched interactions. Again appealing to the rationale that we want to maximise the use of pairwise information, this implies adopting $\rho = K/2$ would the best choice in this scenario.

What if $d_I(\cdot, \cdot)$ is not bounded? In this case, $d_{\mathrm{M},\rho}$ and $d_{\mathrm{E},\rho}$ are still applicable (and still distance metrics) with ρ essentially representing a threshold: only interactions with a distance that is less than 2ρ will matched. In this way, there is no explicit recommendation for how to set ρ in this case, being a slightly more subjective choice. However, a pragmatic solution would be to use the observed distribution of pairwise distances to inform this choice.

A.3 Distance computation

A.3.1 Path distances

The LCS distance d_{LCS}, like the edit distance (Appendix A.3.3), is a special case of the string edit distance proposed by Wagner and Fischer (1974). Thus, the dynamic programming algorithm proposed therein can be applied, in this case at a complexity $\mathcal{O}(n \cdot m)$ where n and m are the lengths of the paths being compared.

In particular, suppose we are comparing $\mathcal{I} = (x_1, \ldots, x_n)$ and $\mathcal{I}' = (y_1, \ldots, y_m)$. Using the subpath notation $\mathcal{I}_{k:l} = (x_k, \ldots, x_l)$, to compute the LCS distance we incrementally evaluate $d_{\mathrm{LCS}}(\mathcal{I}_{1:i}, \mathcal{I}'_{1:j})$, that is, the LCS distance between truncations of the two paths, until $i = n$ and $j = m$. This is done via the following recursive formula

$$d_{\mathrm{LCS}}(\mathcal{I}_{1:i}, \mathcal{I}'_{1:j}) = \min \begin{cases} d_{\mathrm{LCS}}(\mathcal{I}_{1:(i-1)}, \mathcal{I}'_{1:j}) + 1 \\ d_{\mathrm{LCS}}(\mathcal{I}_{1:i}, \mathcal{I}'_{1:(j-1)}) + 1 \\ d_{\mathrm{LCS}}(\mathcal{I}_{1:(i-1)}, \mathcal{I}'_{1:(j-1)}) + 2 \cdot \mathbb{1}(x_i \neq y_j), \end{cases}$$

where $\mathbb{1}(\cdot)$ is the identity function, which follows directly from the definition of the LCS distance. Letting

$$C_{ij} = d_{\mathrm{LCS}}(\mathcal{I}_{1:(i-1)}, \mathcal{I}'_{1:(j-1)})$$

this equates to filling up an $(n+1) \times (m+1)$ matrix C via the following formula

$$C_{(i+1)(j+1)} = \min \begin{cases} C_{i(j+1)} + 1 \\ C_{(i+1)j} + 1 \\ C_{ij} + 2 \cdot \mathbb{1}(x_i \neq y_j), \end{cases}$$

where the distance is then given by $d_{\mathrm{LCS}}(\mathcal{I}, \mathcal{I}') = C_{(n+1)(m+1)}$, that is, the final entry of the constructed matrix. For pseudocode of the resulting algorithm to compute

d_{LCS} see Algorithm 5, with a lighter-memory version outlined in Algorithm 6, which essentially stores only the current and previous rows of C.

Note, in Wagner and Fischer (1974) they set-up the problem in terms of substitution costs, whereby $\gamma(a \to b)$ denotes the cost of substituting entry a for b, whilst $\gamma(a \to \Lambda)$ denotes the cost of deleting a, with Λ denoting the null entry, so that similarly $\gamma(\Lambda \to a)$ denotes the cost of insertion. In this notation, the LCS distance as we define it equates to

$$\gamma(a \to b) = \begin{cases} 0 & \text{if } a = b \\ 2 & \text{otherwise} \end{cases}$$

whilst $\gamma(a \to \Lambda) = \gamma(\Lambda \to a) = 1$ for all entries a.

The approach we use to evaluate d_{LSP} is slightly different, though its complexity continues to be $\mathcal{O}(n{\cdot}m)$. In this case, we essentially scan over $\mathcal{I} = (x_1, \ldots, x_n)$ and $\mathcal{I}' = (y_1, \ldots, y_m)$ and keep track of the common subpaths seen. Formally, we construct an $n \times m$ matrix Q incrementally via the following recursive formula

$$Q_{(i+1)(j+1)} = \begin{cases} Q_{ij} + 1 & \text{if } x_i = y_j \\ 0 & \text{otherwise} \end{cases},$$

where when common subpaths appear between $\mathcal{I}$ and $\mathcal{I}'$ one will see increments in Q diagonally. The maximum length of a subpath can thus be obtained by taking the element-wise maximum of Q, that is $\delta_{\mathrm{LSP}} = \max_{ij} Q_{ij}$, which can then be used to evaluate d_{LSP} (see definition in Section 3.4). We summarise this in Algorithm 7, where we keep track of the maximum in Q as it is filled. A lighter-memory algorithm is also outlined in Algorithm 8, making use of the fact we only need to know the current and previous rows of Q.

A.3.2 Matching distance

As mentioned in Section 3.5.1, evaluating $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}')$ (Definition 3.5.1) requires solving an optimisation problem. In particular, finding an optimal matching. As outlined therein, we consider casting this as an *assignment problem*, which can then be solved with known algorithms, such as the Hungarian algorithm (Kuhn, 1955).

The assignment problem is as follows. Supposing that one has two sets

$$A = \{a_1, \ldots, a_n\} \quad \text{and} \quad B = \{b_1, \ldots, b_n\},$$

both of size n, one considers pairing elements of set A with those of set B in an 'optimal' way, where the objective is defined by assigning a cost to each possible pairing. Note the labelling of elements here is arbitrary but will serve a purpose in what follows, allowing us to index set elements. The cost of all possible pairings is summarised via the $n \times n$ matrix C, where $C_{ij} > 0$ denotes the cost incurred when $a_i \in A$ is paired with $b_j \in B$. A specific pairing of set elements can be encoded via a permutation $\sigma \in S_n$, where S_n denotes the set of all permutations on n symbols, with $\sigma(i) = j$ implying that $a_i \in A$ has been paired with $b_j \in B$. With this, the assignment problem seeks a permutation with minimal cost, that is

$$\min_{\sigma \in S_n} \sum_{i=1}^{n} C_{i,\sigma(i)},$$

the solution of which may not be unique. Observe that though A and B are typically assumed to be sets, this formulation works equally well if they are multisets (as we will consider).

Towards evaluating the matching distance, we set-up a cost matrix C such that the optimal solution found via the Hungarian algorithm coincides with an optimal matching in accordance with Definition 3.5.1. Here we consider two scenarios. In the first, more general case, we will optimise over all matchings (including those

which match nothing). In the second scenario, we will optimise over only complete matchings. The second case is a smaller optimisation problem, making it easier to solve and thus preferable. However, it is not guaranteed that an optimal matching will be complete. Thus the former will work in all cases, but the latter may result in a sub-optimal solution in some scenarios. To guide this, we provide a result which says when it is okay to use the latter approach.

Optimising over all matchings

Suppose we have two interaction multisets

$$\mathcal{E} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\} \quad \mathcal{E}' = \{\mathcal{I}'_1, \ldots, \mathcal{I}'_M\}$$

and we are seeking to evaluate $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}')$. If we see $\mathcal{E}$ and $\mathcal{E}'$ as the sets of the assignment problem, it is somewhat natural to represent the matching of set elements: we let $\sigma(i) = j$ if $(\mathcal{I}_i, \mathcal{I}'_j) \in \mathcal{M}$. However, we also need to encode the possibility for an element of either set to be left unmatched. This can be handled by effectively augmenting each set with some dummy elements which, if paired to, will represent an interaction being unmatched. Using the notation above we would assume

$$\begin{aligned} A &= \{a_1, \ldots, a_n\} & B &= \{b_1, \ldots, b_n\} \\ &= \{\mathcal{I}_1, \ldots, \mathcal{I}_N, \underbrace{\Lambda, \ldots, \Lambda}_{M}\} & &= \{\mathcal{I}'_1, \ldots, \mathcal{I}'_M, \underbrace{\Lambda, \ldots, \Lambda}_{N}\} \end{aligned}$$

where Λ represents a dummy element, so that, if say $\mathcal{I}_i$ is paired with a Λ this will be interpreted as $\mathcal{I}_i$ being unmatched, and the same for elements of $\mathcal{E}'$. Notice also with A there are M dummy elements added to $\mathcal{E}$, so that all M elements of $\mathcal{E}'$ could in theory be matched with a dummy element, that is, all elements of $\mathcal{E}'$ could be left unmatched. Similarly, in B there are N dummy elements added to $\mathcal{E}'$, so that all elements of $\mathcal{E}$ could be unmatched. Moreover, with this both A and B are now of the

same size $n = N + M$, as required for the assignment problem.

With this, the interpretation of a permutation $\sigma \in S_n$ in terms of a matching between $\mathcal{E}$ and $\mathcal{E}'$ is a follows

- If $\sigma(i) = j \leq M$ for $i \leq N$ then $\mathcal{I}_i \in \mathcal{E}$ has been matched with $\mathcal{I}'_j \in \mathcal{E}'$;
- If $\sigma(i) > M$ for $i \leq N$ then $\mathcal{I}_i \in \mathcal{E}$ has been unmatched;
- If $\sigma(i) = j \leq M$ for $i > N$ then $\mathcal{I}'_j \in \mathcal{E}'$ has been left unmatched;
- If $\sigma(i) > M$ for $i > N$ then an dummy element has been paired with a dummy element.

With this, each σ encodes a matching $\mathcal{M}_\sigma$ of $\mathcal{E}$ and $\mathcal{E}'$ given by the following

$$\mathcal{M}_\sigma = \{(\mathcal{I}_i, \mathcal{I}'_{\sigma(i)}) \,:\, 1 \leq i \leq N,\, \sigma(i) \leq M\}.$$

With the sets to be paired defined, all that remains is to lay out the correct $(N + M) \times (N + M)$ cost matrix, which in this case is defined as follows

$$C_{ij} = \begin{cases} d_I(\mathcal{I}_i, \mathcal{I}'_j) & \text{if } i \leq N \text{ and } j \leq M \\ \delta(\mathcal{I}'_j) & \text{if } i > N \text{ and } j \leq M \\ \delta(\mathcal{I}_i) & \text{if } i \leq N \text{ and } j > M \\ 0 & \text{if } i > N \text{ and } j > M \end{cases}$$

where $d_I(\cdot, \cdot)$ is the chosen ground distance and $\delta(\cdot)$ the chosen penalty term for unmatched elements. Notice the cost of pairing two dummy elements is zero. To see

why C takes this form, consider the cost it implies for a given matching $\mathcal{M}_\sigma$, that is

$$\begin{aligned}\text{Cost}(\mathcal{M}_\sigma) &= \sum_{i=1}^{N+M} C_{i,\sigma(i)} \\ &= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}_\sigma} d_I(\mathcal{I},\mathcal{I}') + \sum_{\mathcal{I}\in(\mathcal{M}_\sigma)^{\mathrm{c}}_{\mathcal{E}}} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in(\mathcal{M}_\sigma)^{\mathrm{c}}_{\mathcal{E}'}} \delta(\mathcal{I}')\end{aligned}$$

where we have simply applied the definition of C as above. Comparing this with Definition 3.5.1, one can see that C encodes the required matching cost to be minimised when evaluating $d_{\mathrm{M},\delta(\cdot)}$. Thus, if every matching is represented by every pairing of A and B, and the costs are equivalent, then the optimal solutions will coincide. This means any optimal σ^* found for the given C will define an optimal matching $\mathcal{M}_{\sigma^*}$ which can be used to evaluate $d_{\mathrm{M},\delta(\cdot)}$. With this, the steps to evaluate $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}')$ are: (i) construct C as above, (ii) pass C to a solver, such as the Hungarian algorithm, returning an optimal permutation σ^* and then finally (iii) translate σ^* to an optimal matching $\mathcal{M}_{\sigma^*}$ to evaluate the distance.

Optimising over complete matchings

In the previous section, we set-up an assignment problem which optimise over all matchings, including those which match no elements. However, there are scenarios where this is unnecessary. In particular, in some cases one actually needs to only optimise over *complete* matchings. This allows us to set-up a slightly smaller assignment problem, which will typically be quicker to solve.

Recall a matching $\mathcal{M}$ of the two multisets $\mathcal{E}$ and $\mathcal{E}'$ is complete if all elements of the smaller set are included, that is, if $|\mathcal{M}| = \min(|\mathcal{E}|, |\mathcal{E}'|)$. Now, the main motivation for this second evaluation approach is the following result (proved in Appendix A.4.2).

Proposition A.3.1: Given two interaction multisets $\mathcal{E}$ and $\mathcal{E}'$, if the following holds

$$\delta(\mathcal{I}) + \delta(\mathcal{I}') \geq d_I(\mathcal{I},\mathcal{I}')$$

for all $\mathcal{I} \in \mathcal{E}$ and $\mathcal{I}' \in \mathcal{E}'$, then there exists a complete optimal matching achieving the optimum defining the matching distance $d_{\mathrm{M},\delta(\cdot)}$ (Definition 3.5.1).

As a consequence of Proposition A.3.1, if the conditions therein are satisfied then it suffices to find an optimal complete matching. As such, in what follows we show how an assignment problem can again be set up to enact this optimisation.

Suppose that, without loss of generality, the two multisets to be compared

$$\mathcal{E} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\} \quad \text{and} \quad \mathcal{E}' = \{\mathcal{I}'_1, \ldots, \mathcal{I}'_M\}$$

are such that $N \leq M$, that is, $\mathcal{E}$ is the smaller of the two (when they are of different size). As such, a complete matching between $\mathcal{E}$ and $\mathcal{E}'$ will match all elements of $\mathcal{E}$ to a unique element of $\mathcal{E}'$, whilst some elements of $\mathcal{E}'$ may be left unmatched. With this, we set-up the following sets for the assignment problem

$$\begin{aligned} A &= \{a_1, \ldots, a_n\} & B &= \{b_1, \ldots, b_n\} \\ &= \{\mathcal{I}_1, \ldots, \mathcal{I}_N, \underbrace{\Lambda, \ldots, \Lambda}_{M-N}\} & &= \{\mathcal{I}'_1, \ldots, \mathcal{I}'_M\} \end{aligned}$$

where Λ represents a dummy element such that $\mathcal{I}'_j \in \mathcal{E}'$ being paired with Λ is interpreted as this interaction being left unmatched. Observe that in comparison with the set-up of Appendix A.3.2, we need only augment the smaller of the two multisets with dummy variables. Notice again we have A and B being of the same size, in particular $n = M$, that is, the size of the larger multiset. In this case, the interpretation of a permutation σ is as follows

- If $\sigma(i) = j$ for $i \leq N$ then $\mathcal{I}_i \in \mathcal{E}$ has been matched with $\mathcal{I}'_j \in \mathcal{E}'$;
- If $\sigma(i) = j$ for $i > N$ then $\mathcal{I}'_j \in \mathcal{E}'$ has been left unmatched.

which again encodes a matching $\mathcal{M}_\sigma$ of $\mathcal{E}$ and $\mathcal{E}'$ given by the following

$$\mathcal{M}_\sigma = \{(\mathcal{I}_i, \mathcal{I}'_{\sigma(i)}) \,:\, 1 \leq i \leq N\},$$

which in this case will be complete, since all elements of $\mathcal{E}$ are included in $\mathcal{M}_\sigma$.

In this case, we construct the $M \times M$ cost matrix as follows

$$C_{ij} = \begin{cases} d_I(\mathcal{I}_i, \mathcal{I}'_j) & \text{if } i \leq N \\ \delta(\mathcal{I}'_i) & \text{if } i > N \end{cases}$$

where $d_I(\cdot,\cdot)$ is the chosen ground distance and $\delta(\cdot)$ the penalty for unmatched interactions. Notice, as in Appendix A.3.2, this cost matrix C is such that

$$\begin{aligned} \text{Cost}(\mathcal{M}_\sigma) &= \sum_{i=1}^{M} C_{i,\sigma(i)} \\ &= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}_\sigma} d_I(\mathcal{I},\mathcal{I}') + \sum_{\mathcal{I}'\in(\mathcal{M}_\sigma)^c_{\mathcal{E}'}} \delta(\mathcal{I}'), \end{aligned}$$

which, since $\mathcal{M}_\sigma$ is complete, is in accordance with the cost function being minimised in evaluating $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}')$. Thus any optimal σ^* found for cost matrix C of this form will map to an optimal complete matching $\mathcal{M}_{\sigma^*}$ which can be used to evaluate the matching distance, provided the conditions of Proposition A.3.1 hold. With this, the steps to evaluate $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}')$ (when the necessary conditions hold) are: (i) construct C as above (ii), pass C to a solver, returning an optimal permutation σ^*, then (iii) map σ^* to an optimal complete matching $\mathcal{M}_{\sigma^*}$ to evaluate the distance.

We finalise these details by noting when the conditions of Proposition A.3.1 will hold for the example penalty functions provided in Section 3.5.1. In particular, we have

- **Fixed penalty:** if $\delta(\mathcal{I}) = \rho$ for some constant $\rho > 0$, then when comparing two

multisets $\mathcal{E}$ and $\mathcal{E}'$ the conditions will hold provided

$$\rho \geq \frac{1}{2}\left(\max_{\mathcal{I}\in\mathcal{E},\mathcal{I}'\in\mathcal{E}'} d_I(\mathcal{I},\mathcal{I}')\right),$$

thus placing a lower bound of ρ values which will result in complete matchings;

- **Distance-based penalty:** if $\delta(\mathcal{I}) = d_I(\mathcal{I},\Lambda)$, where Λ represents the null interaction, then the conditions will always hold since

$$d_I(\mathcal{I},\Lambda) + d_I(\Lambda,\mathcal{I}') \geq d_I(\mathcal{I},\mathcal{I}'),$$

following since $d_I(\cdot,\cdot)$ satisfies the triangle inequality, as it is a distance metric.

This implies, d_{M} can always be evaluated by optimising over complete matchings, whilst $d_{\mathrm{M},\rho}$ can be evaluated in this manner only if the above bound on ρ is satisfied.

A.3.3 Edit distance

The edit distance (Definition 3.7.1) can be seen as a special case of the so-called string edit distance introduced by Wagner and Fischer (1974). As such, to evaluate it the dynamic programming algorithm proposed therein can be invoked.

Suppose we have two interaction sequences

$$\mathcal{S} = (\mathcal{I}_1,\ldots,\mathcal{I}_N) \quad \text{and} \quad \mathcal{S}' = (\mathcal{I}'_1,\ldots,\mathcal{I}'_M)$$

and are seeking to evaluate $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S},\mathcal{S}')$. Introducing the notation $\mathcal{S}_{k:l} = (\mathcal{I}_k,\ldots,\mathcal{I}_l)$, the approach is to incrementally evaluate $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:i},\mathcal{S}'_{1:j})$, that is, the distance between truncations of $\mathcal{S}$ and $\mathcal{S}'$, repeating this until $i = |\mathcal{S}|$ and $j = |\mathcal{S}'|$. This is done

via the following recursive result

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:i}, \mathcal{S'}_{1:j}) = \min \begin{cases} d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:(i-1)}, \mathcal{S'}_{1:j}) + \delta(\mathcal{I}_i) \\ d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:i}, \mathcal{S'}_{1:(j-1)}) + \delta(\mathcal{I}'_j) \\ d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:(i-1)}, \mathcal{S'}_{1:(j-1)}) + d_I(\mathcal{I}_i, \mathcal{I}'_j), \end{cases} \tag{A.3.1}$$

which relates the distance between $\mathcal{S}_{1:i}$ and $\mathcal{S'}_{1:j}$ to distances between slight truncations thereof. The key point here is this recursive result comes straight from the definition of the edit distance, where the three cases correspond to three different scenarios: (i) the ith entry of $\mathcal{S}$ is unmatched, (ii) the jth entry of $\mathcal{S}'$ is unmatched, and (iii) the ith entry of $\mathcal{S}$ is matched with the jth entry of $\mathcal{S}'$.

Letting $C_{ij} = d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:(i-1)}, \mathcal{S'}_{1:(j-1)})$, incremental evaluation of eq. (A.3.1) can be seen as filling up an $(N+1) \times (M+1)$ matrix C either row-by-row or column-by-column according to the following formula

$$C_{(i+1)(j+1)} = \min \begin{cases} C_{i(j+1)} + \delta(\mathcal{I}_i) \\ C_{(i+1)j} + \delta(\mathcal{I}'_j) \\ C_{ij} + d_I(\mathcal{I}_i, \mathcal{I}'_j), \end{cases}$$

where the final entry $C_{(N+1)(M+1)}$ corresponds to the desired distance. Note the first column and row can be specified as follows

$$\begin{aligned} C_{i1} &= d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:i}, \mathcal{S'}_{1:0}) & C_{1j} &= d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:0}, \mathcal{S'}_{1:j}) \\ &= \sum_{k=1}^{i} \delta(\mathcal{I}_k) & &= \sum_{k=1}^{j} \delta(\mathcal{I}'_k) \end{aligned}$$

for $i = 2, \ldots, N$ and $j = 2, \ldots, M$, which follow by seeing $\mathcal{S}_{1:0}$ and $\mathcal{S'}_{1:0}$ as empty sequences, so that when measuring the distance of these to $\mathcal{S'}_{1:j}$ and $\mathcal{S}_{1:i}$ (respectively)

all entries thereof will be left unmatched, since there are no entries to be matched to. Finally, when both $i = 1$ and $j = 1$ we will have $C_{11} = d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:0}, \mathcal{S}'_{1:0}) = 0$, since we can see this as the distance of the empty sequence to itself. All together, these represent initial conditions from which repeated application of eq. (A.3.1) will take us to the desired result.

Algorithm 1 outlines pseudocode for an algorithm to evaluate $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}')$ by filling the matrix C in this manner. However, observe that when updating a row (or column) of C one only needs to know the previous row (or column). As such, we only need to store the current and previous row (or column), leading to an algorithm which uses less memory and is typically faster. Pseudocode of this light-memory alternative is given in Algorithm 2.

A.3.4 Dynamic time warping distance

Evaluation of the dynamic time warping distance (Definition 3.7.3) can be achieved, as with the edit distance (Appendix A.3.3), via dynamic programming. In fact, the algorithm is almost identical to that used for the edit distance, differing only in the underlying recursive formula used. We note the implementation we use here was drawn from Gold and Sharir (2018), Section 3.

Suppose we have two interaction sequences

$$\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N) \quad \text{and} \quad \mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_M)$$

and are seeking to evaluate $d_{\mathrm{DTW}}(\mathcal{S}, \mathcal{S}')$. Adopting the notation $\mathcal{S}_{k:l} = (\mathcal{I}_k, \ldots, \mathcal{I}_l)$, one evaluates $d_{\mathrm{DTW}}(\mathcal{S}_{1:i}, \mathcal{S}'_{1:j})$ incrementally until $i = N$ and $j = M$ via the following

recursive result

$$d_{\mathrm{DTW}}(\mathcal{S}_{1:i}, \mathcal{S'}_{1:j}) = d_I(\mathcal{I}_i, \mathcal{I}'_j) + \min \begin{cases} d_{\mathrm{DTW}}(\mathcal{S}_{1:(i-1)}, \mathcal{S'}_{1:j}) \\ d_{\mathrm{DTW}}(\mathcal{S}_{1:i}, \mathcal{S'}_{1:(j-1)}) \\ d_{\mathrm{DTW}}(\mathcal{S}_{1:(i-1)}, \mathcal{S'}_{1:(j-1)}) \end{cases}$$

which follows directly from the definition of the DTW distance. Note here one is essentially comparing three possibilities: (i) warping on the jth entry of $\mathcal{S}'$, that is, $\mathcal{I}'_j$ being paired with more than one element of $\mathcal{S}$, (ii) warping on the ith entry of $\mathcal{S}$, and (iii) no warping, with $\mathcal{I}_i$ and $\mathcal{I}'_j$ being paired *only* with each other. Note the $d_I(\mathcal{I}_i, \mathcal{I}'_j)$ term comes out front of the $\min$ term since by definition $\mathcal{I}_i$ and $\mathcal{I}'_j$ (the last entries of these truncated sequences) must be paired.

Introducing the notation $C_{ij} = d_{\mathrm{DTW}}(\mathcal{S}_{1:(i-1)}, \mathcal{S'}_{1:(j-1)})$, the incremental computation can be seen as filling-up the $(N+1) \times (M+1)$ matrix C either row-by-row or column-by-column via the following recursive formula

$$C_{(i+1)(j+1)} = d_I(\mathcal{I}_i, \mathcal{I}'_j) + \min \begin{cases} C_{i(j+1)} \\ C_{(i+1)j} \\ C_{ij} \end{cases}$$

with $d_{\mathrm{DTW}}(\mathcal{S}, \mathcal{S}') = C_{(N+1)(M+1)}$. As with evaluating the edit distances (Appendix A.3.3), we must also pre-specify the first row and column of C. Here we again assume $C_{11} = 0$ whilst

$$C_{(i+1)1} = \infty \quad (\text{for } i = 1, \ldots, N) \qquad C_{1(j+1)} = \infty \quad (\text{for } j = 1, \ldots, M).$$

To see why these conditions are used, consider obtaining the second column entries,

that is

$$C_{i2} = d_{\mathrm{DTW}}(\mathcal{S}_{1:(i-1)}, \mathcal{S}'_{1:1}).$$

Observe that since $\mathcal{S}'_{1:1} = (\mathcal{I}'_1)$ is a sequence with a single entry, the only valid coupling here is where $\mathcal{I}'_1$ is paired with every entry of $\mathcal{S}_{1:(i-1)}$. By opting for the above choice of initial values within C one ensures this occurs via the recursive formula. In particular, if one considers filling the second column of C, one has

$$C_{22} = d_I(\mathcal{I}_1, \mathcal{I}'_1) + \min \begin{cases} \infty \\ \infty \\ 0 \end{cases}$$

so we choose the third option, paring the first two entries with no warping, whilst for $i > 2$ we have

$$C_{i2} = d_I(\mathcal{I}_1, \mathcal{I}'_1) + \min \begin{cases} C_{(i-1)2} \\ \infty \\ \infty \end{cases}$$

where here we choose the first option, which corresponds to warping on the first entry of $\mathcal{S}$, that is, $\mathcal{I}'_1$ being paired with more than one element of $\mathcal{S}$. The same reasoning can be used to justify the initial values of the first row by considering filling the second row of C.

Algorithm 3 outlines the algorithm which fills the matrix C via this recursive formula to obtain the desired distance. As with the edit distance, this procedure only requires knowledge of the previous and current row, and hence a lighter memory alternative can also be used, as detailed in Algorithm 4.

A.4 Proofs

A.4.1 Path distances

Proof that d_{LCS} and d_{LSP} are distances are metrics. Let us first prove that d_{LCS} is a metric. Recall the LCS distance (defined in Section 3.4) between paths $\mathcal{I}$ and $\mathcal{I}'$ is given by

$$d_{\mathrm{LCS}}(\mathcal{I}, \mathcal{I}') = n + m - \delta_{\mathrm{LCS}}$$

where n and m are the lengths of $\mathcal{I}$ and $\mathcal{I}'$, and δ_{LCS} is the maximum length of a common sequence between them. Consider now the first metric condition (i) (identity of indiscernibles). Here we will use the following fact: $\delta_{\mathrm{LCS}} \leq n$ and $\delta_{\mathrm{LCS}} \leq m$, following since a common subsequence cannot include more entries than are present in either path. Now, assuming that

$$d_{\mathrm{LCS}}(\mathcal{I}, \mathcal{I}') = n + m - 2\delta_{\mathrm{LCS}} = 0 \tag{A.4.1}$$

we claim this implies $n = m$. To see this, notice if we assume $n < m$ this implies

$$n + m > 2n \geq 2\delta_{\mathrm{LCS}}$$

where we have used the fact $\delta_{\mathrm{LCS}} \leq n$. Notice this contradicts eq. (A.4.1). A similar contradiction will be found if we assume $n > m$, and consequently we must have $n = m$. Substituting this into eq. (A.4.1) leads to $\delta_{\mathrm{LCS}} = n = m$ which implies that $\mathcal{I}$ and $\mathcal{I}'$ share a common subsequence of the same length as themselves, that is $\mathcal{I} = \mathcal{I}'$. This proves one direction. For the converse case, if $\mathcal{I} = \mathcal{I}'$ then it should be clear that the maximum common subsequence will be that including all their entries, that is

$\delta_{\text{LCS}} = n = m$ and hence

$$d_{\text{LCS}}(\mathcal{I}, \mathcal{I}') = n + m - 2\delta_{\text{LCS}} = 0,$$

thus proving condition (i) holds for the LCS distance.

It should be clear the symmetry condition (ii) follows trivially from the inherent symmetry in the definition of a common subsequence.

Finally, we turn to the triangle inequality (iii). Assume we have three paths

$$\mathcal{I}^X = (x_1, \ldots, x_n) \quad \mathcal{I}^Y = (y_1, \ldots, y_m) \quad \mathcal{I}^Z = (z_1, \ldots, z_k)$$

and that δ_{XY}, δ_{ZY} and δ_{XY} are such that

$$d_{\text{LCS}}(\mathcal{I}^X, \mathcal{I}^Y) = n + m - 2\delta_{XY} \qquad d_{\text{LCS}}(\mathcal{I}^X, \mathcal{I}^Z) = n + k - 2\delta_{XZ}$$
$$d_{\text{LCS}}(\mathcal{I}^Z, \mathcal{I}^Y) = m + k - 2\delta_{ZY}$$

then, if the triangle inequality holds, we have

$$d_{\text{LCS}}(\mathcal{I}^X, \mathcal{I}^Y) \leq d_{\text{LCS}}(\mathcal{I}^X, \mathcal{I}^Z) + d_{\text{LCS}}(\mathcal{I}^Z, \mathcal{I}^Y)$$

which is equivalent to the following

$$n + m - 2\delta_{XY} \leq (n + k - 2\delta_{XZ}) + (m + k - 2\delta_{ZY})$$

which is true if and only if (by rearranging terms)

$$\delta_{XZ} + \delta_{ZY} - k \leq \delta_{XY}, \tag{A.4.2}$$

thus, if we show eq. (A.4.2) holds the implications will trace back to show the the

triangle inequality also holds. Towards doing so, we consider a finding the common subsequence between $\mathcal{I}^X$ and $\mathcal{I}^Y$ induced by that between $\mathcal{I}^X$ and $\mathcal{I}^Z$ and between $\mathcal{I}^Y$ and $\mathcal{I}^Z$, which will allow us to obtain the desired lower bound.

To aide this exposition we introduce some notation. In particular, for two subsequences $\boldsymbol{v}$ and $\boldsymbol{u}$ of $[n] = (1, \ldots, n)$ we can extend the notion of unions and intersections used for sets, that is $\boldsymbol{v} \cup \boldsymbol{u}$ and $\boldsymbol{v} \cap \boldsymbol{u}$ respectively, where if $\boldsymbol{w} = \boldsymbol{v} \cap \boldsymbol{u}$ then each entry w_i appears in both $\boldsymbol{v}$ and $\boldsymbol{u}$, whilst if $\boldsymbol{w} = \boldsymbol{v} \cup \boldsymbol{u}$ then each w_i appears in at least one of $\boldsymbol{u}$ and $\boldsymbol{v}$. For example, if we have $n = 5$ and $\boldsymbol{u} = (1, 3, 5)$ and $\boldsymbol{v} = (1, 2, 5)$ then $\boldsymbol{u} \cap \boldsymbol{v} = (1, 5)$ whilst $\boldsymbol{u} \cup \boldsymbol{v} = (1, 2, 3, 5)$. Moreover, with $|\boldsymbol{v}|$ denoting the length of subsequence $\boldsymbol{v}$, the following will hold

$$|\boldsymbol{v}| + |\boldsymbol{u}| - |\boldsymbol{v} \cap \boldsymbol{u}| = |\boldsymbol{v} \cup \boldsymbol{u}|,$$

which can be seen as analogous to the inclusion-exclusion identity for sets.

Now suppose that we have indexing subsequences $\boldsymbol{u}_{XZ}$, $\boldsymbol{v}_{XZ}$, $\boldsymbol{u}_{ZY}$ and $\boldsymbol{v}_{ZY}$ such that

$$\mathcal{I}^X_{\boldsymbol{v}_{XZ}} = \mathcal{I}^Z_{\boldsymbol{u}_{XZ}} \qquad\qquad \mathcal{I}^Z_{\boldsymbol{v}_{ZY}} = \mathcal{I}^Y_{\boldsymbol{u}_{ZY}}$$

with $|\boldsymbol{u}_{XZ}| = |\boldsymbol{v}_{XZ}| = \delta_{XZ}$ and $|\boldsymbol{u}_{ZY}| = |\boldsymbol{v}_{ZY}| = \delta_{ZY}$, that is, these index maximal common subsequences. Observe the intersection $\boldsymbol{u}_{XZ} \cap \boldsymbol{v}_{ZY}$ defines a subsequence of $\mathcal{I}^Z$ which is shared with both $\mathcal{I}^X$ and $\mathcal{I}^Y$, and consequently, if we let $\boldsymbol{v}_{XY}$ and $\boldsymbol{u}_{XY}$ denote indices of the associated subsequences of $\mathcal{I}^X$ and $\mathcal{I}^Y$, respectively, we have

$$\mathcal{I}^X_{\boldsymbol{v}_{XY}} = \mathcal{I}^Y_{\boldsymbol{u}_{XY}}$$

that is, these index a common subsequence of $\mathcal{I}^X$ and $\mathcal{I}^Y$. Moreover, if we let

$$\delta^* := |\boldsymbol{v}_{XY}| = |\boldsymbol{u}_{XY}| = |\boldsymbol{u}_{XZ} \cap \boldsymbol{v}_{ZY}|,$$

denoting the size of this induced common subsequence, then by the inclusion-exclusion identity above we have

$$\delta_{XZ} + \delta_{ZY} - \delta^* = |\boldsymbol{u}_{XZ}| + |\boldsymbol{v}_{ZY}| - |\boldsymbol{u}_{XZ} \cap \boldsymbol{v}_{ZY}| = |\boldsymbol{u}_{XZ} \cup \boldsymbol{v}_{ZY}| \leq k$$

where the inequality here follows since $\boldsymbol{u}_{ZX} \cup \boldsymbol{v}_{ZY}$ is an indexing subsequences of $\mathcal{I}^Z$, which is of length k. This rearranges to the following

$$\delta_{XZ} + \delta_{ZY} - k \leq \delta^*,$$

and finally, using the fact that $\delta^* \leq \delta_{XY}$ by definition of δ_{XY} as the *maximal* length of a common subsequence between $\mathcal{I}^X$ and $\mathcal{I}^Y$, we thus have

$$\delta_{XZ} + \delta_{ZY} - k \leq \delta_{XY},$$

confirming eq. (A.4.2) holds, as desired. Consequently, the LCS distance satisfies metric condition (iii). This completes the proof that d_{LCS} is a distance metric.

We now consider proving d_{LSP} is also a distance metric. Firstly, regarding the identity of indiscernibles (i), one can use exactly the same argument as for the LCS distance above. For brevity, we will avoid repeating this and henceforth assume this condition holds. Similarly, the symmetry condition (ii) again follows trivially from the symmetry of common subpaths.

To show d_{LSP} satisfies the triangle inequality (iii) we can use almost the same argument outlined above for the LCS distance. In particular, one can show that eq. (A.4.2) holds, where in this case δ_{XZ}, δ_{ZY} and δ_{XY} denote maximal *subpath* sizes. A key dif-

ference here is that we must obtain an induced subpath rather than subsequence. If we introduce the shorthand notation $(i : j) = (i, \ldots, j)$ where $1 \leq i \leq j \leq n$, denoting the subpath of $[n]$ from i to j (notice this is consistent with notation used in Section 3.4), then as with subsequences we can define natural generalisations of the interaction and union of two subpaths, in particular

$$(i : j) \cap (l : k) = (\max(i, l) : \min(j, k)) \qquad (i : j) \cup (l : k) = (\min(i, l) : \max(j, k)),$$

and moreover if $|(i : j)| = j - i + 1$ denotes subpath length we will again have the following inclusion-exclusion identity

$$|(i : j)| + |(l : k)| - |(i : j) \cap (l : k)| = |(i : j) \cup (l : k)|.$$

With these, one can directly adapt the argument used to show d_{LCS} satisfied the triangle inequality. In particular, any two optimal common subpaths between $\mathcal{I}^X$ and $\mathcal{I}^Z$ and between $\mathcal{I}^Z$ and $\mathcal{I}^Y$ will induce a common subpath between $\mathcal{I}^X$ and $\mathcal{I}^Y$, in turn providing the required bound. For brevity, we do not repeat this here, assuming henceforth that metric condition (iii) holds.

Thus conditions (i) to (iii) hold for both the LCS and LSP distances, completing the proof.

□

A.4.2 Multiset distances

Proof of Proposition 3.5.2 (*Matching distance is a metric*). To aid this exposition, write $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}')$ (Definition 3.5.1) in terms of its cost function as follows

$$d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}') = \min_{\mathcal{M} \in \boldsymbol{\mathcal{M}}(\mathcal{E}, \mathcal{E}')} \mathrm{Cost}(\mathcal{M})$$

where

$$\mathrm{Cost}(\mathcal{M}) = \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}} d_I(\mathcal{I},\mathcal{I}') + \sum_{\mathcal{I}\in\mathcal{M}^c_{\mathcal{E}}} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in\mathcal{M}^c_{\mathcal{E}'}} \delta(\mathcal{I}'),$$

denotes the cost of the matching $\mathcal{M}$. We first show metric condition (i) (identity of indiscernibles) holds. If we assume $\mathcal{E} = \mathcal{E}'$ then one can construct a matching $\mathcal{M}^*$ by pairing equivalent elements of $\mathcal{E}$ and $\mathcal{E}'$, leading to the following upper bound

$$\begin{aligned} d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}') &\leq \mathrm{Cost}(\mathcal{M}^*) \\ &= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^*} d_I(\mathcal{I},\mathcal{I}') \\ &= 0 \end{aligned}$$

where the second line follow since $\mathcal{M}^*$ includes all elements of $\mathcal{E}$ and $\mathcal{E}'$ and thus no penalty terms will appear, whilst the third line follows since $\mathcal{M}^*$ matches equivalent elements and hence, using the fact $d_I(\cdot,\cdot)$ satisfies the identity of indiscernibles, all pairwise distances will be zero. Now, since $d_I(\cdot,\cdot) \geq 0$ and $\delta(\cdot) > 0$ by assumption, $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}')$ is a sum of positive values, implying also that $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}') \geq 0$. Together these imply $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}') = 0$.

Conversely, assume that $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}') = 0$. This implies both the sum of pairwise distances and penalisation terms must be zero. Since by assumption $\delta(\mathcal{I}) > 0$ this implies there must be no penalty terms, that is, all elements of $\mathcal{E}$ and $\mathcal{E}'$ must be included in the matching. Thus, with $\mathcal{M}^*$ the optimal matching, we have

$$\begin{aligned} d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}') &= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^*} d_I(\mathcal{I},\mathcal{I}') \\ &= 0, \end{aligned}$$

which, since $d_I(\cdot,\cdot)$ is non-negative, implies

$$d_I(\mathcal{I},\mathcal{I}') = 0 \quad \forall\, (\mathcal{I},\mathcal{I}') \in \mathcal{M}^*,$$

which in turn implies

$$\mathcal{I} = \mathcal{I}' \quad \forall\, (\mathcal{I}, \mathcal{I}') \in \mathcal{M}^*,$$

since $d_I(\cdot, \cdot)$ satisfies the identity of indiscernibles. Hence, we have $\mathcal{E} = \mathcal{E}'$, thus confirming that $d_{\mathrm{M},\delta(\cdot)}$ satisfies metric condition (i).

The symmetry condition (ii) follows trivially from the symmetry of $d_I(\cdot, \cdot)$ (since it is a metric) and the penalisation terms.

Finally, we prove that $d_{\mathrm{M},\delta(\cdot)}$ satisfies metric condition (iii) (triangle inequality). Assuming we have three multisets

$$\mathcal{E}_X = \{\mathcal{I}_1^X, \ldots, \mathcal{I}_{n_X}^X\} \quad \mathcal{E}_Y = \{\mathcal{I}_1^Y, \ldots, \mathcal{I}_{n_Y}^Y\} \quad \mathcal{E}_Z = \{\mathcal{I}_1^Z, \ldots, \mathcal{I}_{n_Z}^Z\}$$

we seek to show that

$$d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X, \mathcal{E}_Y) \leq d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X, \mathcal{E}_Z) + d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_Z, \mathcal{E}_Y).$$

Let $\mathcal{M}_{XZ}^*$ and $\mathcal{M}_{ZY}^*$ denote optimal matchings for $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X, \mathcal{E}_Z)$ and $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_Z, \mathcal{E}_Y)$ respectively, so that

$$d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X, \mathcal{E}_Z) = \mathrm{Cost}(\mathcal{M}_{XZ}^*) \qquad d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_Z, \mathcal{E}_Y) = \mathrm{Cost}(\mathcal{M}_{ZY}^*)$$

and observe these induce a matching $\mathcal{M}_{XY}$ of $\mathcal{E}_X$ and $\mathcal{E}_Y$ as follows

$$\mathcal{M}_{XY} := \{(\mathcal{I}^X, \mathcal{I}^Y) \,:\, (\mathcal{I}^X, \mathcal{I}^Z) \in \mathcal{M}_{XZ}^* \text{ and } (\mathcal{I}^Z, \mathcal{I}^Y) \in \mathcal{M}_{ZY}^* \text{ for some } \mathcal{I}^Z \in \mathcal{E}_Z\}$$

that is, we pair elements of $\mathcal{E}_X$ and $\mathcal{E}_Y$ if they were paired to the same elements of $\mathcal{E}_Z$. For example, Figure A.4.1 shows two cases of optimal matchings $\mathcal{M}_{XZ}^*$ and $\mathcal{M}_{ZY}^*$ along with the matching $\mathcal{M}_{XY}$ they induce (which turns out to be the same in both

cases). Notice by definition of $d_{\mathrm{M},\delta(\cdot)}$ we have

$$d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X, \mathcal{E}_Y) \leq \mathrm{Cost}(\mathcal{M}_{XY}),$$

and so the triangle inequality will follow if we can show the following holds

$$\mathrm{Cost}(\mathcal{M}_{XY}) \leq d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X, \mathcal{E}_Z) + d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_Z, \mathcal{E}_Y). \tag{A.4.3}$$

To prove eq. (A.4.3) we show every possible term on the LHS is less than or equal to some unique terms appearing on the RHS. The key terms appearing on the LHS are (i) pairwise distances for matched elements (ii) penalisation of unmatched elements.

Considering first (i), by definition of $\mathcal{M}_{XY}$ each pair $(\mathcal{I}^X, \mathcal{I}^Y) \in \mathcal{M}_{XY}$ is associated with some *unique* $(\mathcal{I}^X, \mathcal{I}^Z) \in \mathcal{M}^*_{XZ}$ and $(\mathcal{I}^Z, \mathcal{I}^Y) \in \mathcal{M}^*_{ZY}$, that is, there is some element $\mathcal{I}^Z \in \mathcal{E}_Z$ which both $\mathcal{I}^X$ and $\mathcal{I}^Y$ are matched to. Furthermore, since $d_I(\cdot,\cdot)$ is a distance metric it satisfies the triangle inequality, and so

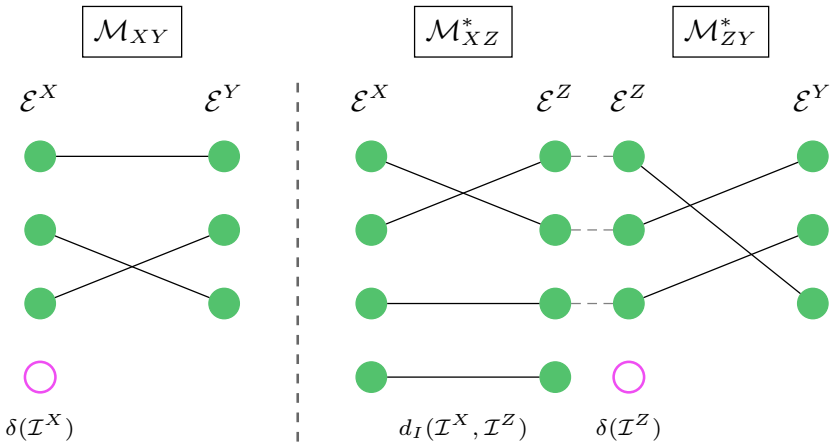
$$d_I(\mathcal{I}^X, \mathcal{I}^Y) \leq d_I(\mathcal{I}^X, \mathcal{I}^Z) + d_I(\mathcal{I}^Z, \mathcal{I}^Y),$$

and thus each pairwise distance of matched elements on the LHS of eq. (A.4.3) is less than or equal to some unique terms on the RHS.

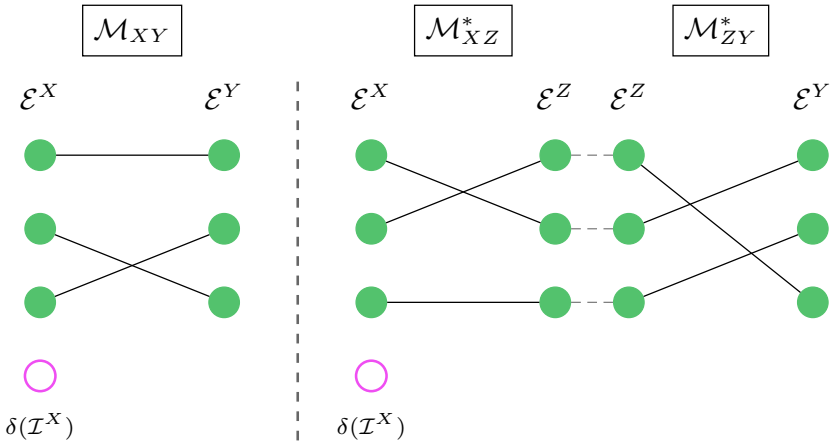
For (ii) consider first the penalisation terms for elements of $\mathcal{E}_X$ not included in the matching $\mathcal{M}_{XY}$, that is $\delta(\mathcal{I}^X)$ for $\mathcal{I}^X \in (\mathcal{M}_{XY})^c_X$. We now seek to show that each $\delta(\mathcal{I}^X)$ is less than or equal to some unique terms appearing on the RHS of eq. (A.4.3). For $\mathcal{I}^X$ to not be in $\mathcal{M}_{XY}$ one of two things must have happened

Case 1: As illustrated in Figure A.4.1a, one may have $(\mathcal{I}^X, \mathcal{I}^Z) \in \mathcal{M}^*_{XZ}$ for some $\mathcal{I}^Z \in \mathcal{E}_Z$ with $(\mathcal{I}^Z, \mathcal{I}^Y) \notin \mathcal{M}^*_{ZY}$ for any $\mathcal{I}^Y \in \mathcal{E}_Y$

$$\implies \text{ a term on the RHS of } d_I(\mathcal{I}^X, \mathcal{I}^Z) + \delta(\mathcal{I}^Z)$$



(a) An element of $\mathcal{E}_X$ is left unmatched in $\mathcal{M}_{XY}$ (induced matching) because the element it was matched with in $\mathcal{E}_Z$ was left unmatched in $\mathcal{M}_{ZY}$.



(b) An element of $\mathcal{E}_X$ is left unmatched in $\mathcal{M}_{XY}$ (induced matching) because it was also unmatched in $\mathcal{M}_{XZ}$.

Figure A.4.1: Examples of (a) **Case 1** and (b) **Case 2** appearing when proving that $d_{\mathrm{M},\delta(\cdot)}$ satisfies the triangle inequality (Proposition 3.5.2). In each subfigure, we have three matchings of the multisets $\mathcal{E}_X$, $\mathcal{E}_Y$ and $\mathcal{E}_Z$, where the two right-most matchings are example optimal matchings which induce the left-most matching of $\mathcal{E}_X$ and $\mathcal{E}_Y$. In both cases, an element of $\mathcal{E}_X$ is left unmatched in the induced matching.

which will also be unique to the pair $(\mathcal{I}^X, \mathcal{I}^Z)$. Now, since by the assumption $|\delta(\mathcal{I}^X) - \delta(\mathcal{I}^Y)| \leq d_I(\mathcal{I}^X, \mathcal{I}^Y)$ for all $\mathcal{I}^X, \mathcal{I}^Y \in \boldsymbol{\mathcal{I}}^*$, we have that

$$\delta(\mathcal{I}^X) \leq d_I(\mathcal{I}^X, \mathcal{I}^Z) + \delta(\mathcal{I}^Z)$$

as desired;

Case 2: Alternatively, as shown in Figure A.4.1b, we might have $(\mathcal{I}^X, \mathcal{I}^Z) \notin \mathcal{M}^*_{XZ}$ for any $\mathcal{I}^Z \in \mathcal{E}_Z$

$$\implies \text{ a term on the RHS of } \delta(\mathcal{I}^X),$$

and thus in this case we trivially have

$$\delta(\mathcal{I}^X) \leq \delta(\mathcal{I}^X).$$

In both cases, we have a term on the LHS of eq. (A.4.3) which is less than or equal to some unique terms on the RHS. Notice this argument can be applied similarly to the penalisation terms for elements of $\mathcal{E}_Y$ not in the matching $\mathcal{M}_{XY}$. For brevity we will not repeat this here, and henceforth assume all penalisation terms for $\mathcal{E}_Y$ are less than or equal to some unique terms on the RHS of eq. (A.4.3).

All together, we have every term on the LHS of eq. (A.4.3), both pairwise distances and penalties, being less than or equal to some unique terms on the RHS, proving the inequality holds. As a consequence, $d_{\mathrm{M},\delta(\cdot)}$ satisfies the triangle inequality, completing the proof. □

Proof of Proposition A.3.1 (*Completeness of matchings*). To aid this exposition, write $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}')$ in terms of its cost function as follows

$$d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}') = \min_{\mathcal{M} \in \boldsymbol{\mathcal{M}}(\mathcal{E}, \mathcal{E}')} \{\mathrm{Cost}(\mathcal{M})\}$$

where

$$\mathrm{Cost}(\mathcal{M}) = \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}} d_I(\mathcal{I},\mathcal{I}') + \sum_{\mathcal{I}\in\mathcal{M}_{\mathcal{E}}^c} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in\mathcal{M}_{\mathcal{E}'}^c} \delta(\mathcal{I}'),$$

denotes the cost of the matching $\mathcal{M}$. Towards proving this result, assume that any matching $\mathcal{M}^*$ for which

$$\mathrm{Cost}(\mathcal{M}^*) = \min_{\mathcal{M}\in\boldsymbol{\mathcal{M}}(\mathcal{E},\mathcal{E}')} \{\mathrm{Cost}(\mathcal{M})\} = d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}'),$$

is *not* complete, seeking a contradiction. There may be more than one such matching, so without loss of generality, let $\mathcal{M}^*$ denote any one of these optimal matchings. Since $\mathcal{M}^*$ is not complete, there must be a currently unmatched pair, that is, $(\tilde{\mathcal{I}},\tilde{\mathcal{I}}')$ such that $\tilde{\mathcal{I}}\in\mathcal{E}$ and $\tilde{\mathcal{I}}'\in\mathcal{E}'$ but $\tilde{\mathcal{I}}\notin\mathcal{M}_{\mathcal{E}}^*$ and $\tilde{\mathcal{I}}'\notin\mathcal{M}_{\mathcal{E}'}^*$. One can now define a new matching $\mathcal{M}^{**}$ by augmenting $\mathcal{M}^*$ as follows

$$\mathcal{M}^{**} = \mathcal{M}^* \cup \{(\tilde{\mathcal{I}},\tilde{\mathcal{I}}')\}$$

for which

$$\begin{aligned}
\mathrm{Cost}(\mathcal{M}^{**}) &= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^{**}} d_I(\mathcal{I},\mathcal{I}') + \sum_{\mathcal{I}\in(\mathcal{M}^{**})_{\mathcal{E}}^c} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in(\mathcal{M}^{**})_{\mathcal{E}'}^c} \delta(\mathcal{I}'),\\
&= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^{*}} d_I(\mathcal{I},\mathcal{I}') + d_I(\tilde{\mathcal{I}},\tilde{\mathcal{I}}') + \sum_{\mathcal{I}\in(\mathcal{M}^{**})_{\mathcal{E}}^c} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in(\mathcal{M}^{**})_{\mathcal{E}'}^c} \delta(\mathcal{I}'),\\
&\leq \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^{*}} d_I(\mathcal{I},\mathcal{I}') + \delta(\tilde{\mathcal{I}}) + \delta(\tilde{\mathcal{I}}') + \sum_{\mathcal{I}\in(\mathcal{M}^{**})_{\mathcal{E}}^c} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in(\mathcal{M}^{**})_{\mathcal{E}'}^c} \delta(\mathcal{I}'),\\
&= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^{*}} d_I(\mathcal{I},\mathcal{I}') + \sum_{\mathcal{I}\in(\mathcal{M}^{*})_{\mathcal{E}}^c} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in(\mathcal{M}^{*})_{\mathcal{E}'}^c} \delta(\mathcal{I}'),\\
&= \mathrm{Cost}(\mathcal{M}^*)
\end{aligned} \tag{A.4.4}$$

where in the third line we invoke the assumption that

$$\delta(\mathcal{I}) + \delta(\mathcal{I}') \geq d_I(\mathcal{I},\mathcal{I}')$$

for all $\mathcal{I} \in \mathcal{E}$ and $\mathcal{I}' \in \mathcal{E}'$ (as in Proposition A.3.1). Since $\mathcal{M}^*$ was optimal, we must also have $\mathrm{Cost}(\mathcal{M}^*) \leq \mathrm{Cost}(\mathcal{M})$ for all matchings $\mathcal{M}$, which combined with eq. (A.4.4) implies $\mathrm{Cost}(\mathcal{M}^{**}) = \mathrm{Cost}(\mathcal{M}^*)$, that is, $\mathcal{M}^{**}$ is also an optimal matching. Moreover, we have $|\mathcal{M}^{**}| = |\mathcal{M}^*| + 1$. Now, either (i) $\mathcal{M}^{**}$ is complete, or (ii) we can repeat this augmentation, increasing the matching cardinality until it is complete. Either way, we arrive at a matching with is both optimal and complete, contradicting our assumption that all optimal matchings were not complete. The result now follows by contradiction.

□

Proof of Proposition 3.5.4 (*EMD metric conditions*). In what follows we will use the notation $d_{\mathrm{W}_1}(\mu_{\mathcal{E}}, \mu_{\mathcal{E}'})$ for the 1-Wasserstein distance between the *distributions* $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{E}'}$, which is known to be a distance metric (Peyré and Cuturi, 2019, Prop. 2.2). Observe that by our definition of the EMD between multisets (Definition 3.5.3) we have $d_{\mathrm{EMD}}(\mathcal{E}, \mathcal{E}') = d_{\mathrm{W}_1}(\mu_{\mathcal{E}}, \mu_{\mathcal{E}'})$.

The conditions (ii) and (iii) are inherited naturally. Firstly, we have

$$\begin{aligned} d_{\mathrm{EMD}}(\mathcal{E}, \mathcal{E}') &= d_{\mathrm{W}_1}(\mu_{\mathcal{E}}, \mu_{\mathcal{E}'}) \\ &= d_{\mathrm{W}_1}(\mu_{\mathcal{E}'}, \mu_{\mathcal{E}}) \\ &= d_{\mathrm{EMD}}(\mathcal{E}', \mathcal{E}) \end{aligned}$$

where the second line follows since d_{W_1} is a metric between distributions, verifying that condition (ii) holds. Secondly, for any multisets $\mathcal{E}$, $\mathcal{E}'$ and $\mathcal{E}''$ we have

$$\begin{aligned} d_{\mathrm{EMD}}(\mathcal{E}, \mathcal{E}') &= d_{\mathrm{W}_1}(\mu_{\mathcal{E}}, \mu_{\mathcal{E}'}) \\ &\leq d_{\mathrm{W}_1}(\mu_{\mathcal{E}}, \mu_{\mathcal{E}''}) + d_{\mathrm{W}_1}(\mu_{\mathcal{E}''}, \mu_{\mathcal{E}'}) \\ &= d_{\mathrm{EMD}}(\mathcal{E}, \mathcal{E}'') + d_{\mathrm{EMD}}(\mathcal{E}'', \mathcal{E}') \end{aligned}$$

where again the second line follows since d_{W_1} is a metric. Thus condition (iii) also

holds.

We now show metric condition (i) does not hold. To do so, let $\mathcal{E}$ be a multiset and define $\mathcal{E}'$ via its multiplicity function as follows

$$m_{\mathcal{E}'}(\mathcal{I}) = C \cdot m_{\mathcal{E}}(\mathcal{I})$$

where $C \in \mathbb{Z}_+$, so that $\mathcal{E}'$ and $\mathcal{E}$ are proportional. Observe that if $C > 1$ then $\mathcal{E} \neq \mathcal{E}'$ whilst

$$\mu_{\mathcal{E}'}(\mathcal{I}) = \frac{m_{\mathcal{E}'}(\mathcal{I})}{|\mathcal{E}'|} = \frac{C \cdot m_{\mathcal{E}}(\mathcal{I})}{C \cdot |\mathcal{E}|} = \mu_{\mathcal{E}}(\mathcal{I})$$

for any $\mathcal{I} \in \boldsymbol{\mathcal{I}}^*$, that is, $\mu_{\mathcal{E}} = \mu_{\mathcal{E}'}$. Consequently, we have $\mathcal{E} \neq \mathcal{E}'$ and

$$d_{\mathrm{EMD}}(\mathcal{E}, \mathcal{E}') = d_{\mathrm{W}_1}(\mu_{\mathcal{E}}, \mu_{\mathcal{E}'}) = 0,$$

thus providing a counterexample, confirming that condition (i) does not hold. This completes the proof. □

A.4.3 Sequence distances

Proof of Proposition 3.7.2 (*Edit distance is a metric*). To aid this exposition, write $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}')$ (Definition 3.7.1) in terms of its cost function as follows

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}') = \min_{\mathcal{M} \in \boldsymbol{\mathcal{M}}_m(\mathcal{S},\mathcal{S}')} \{\mathrm{Cost}(\mathcal{M})\}$$

where

$$\mathrm{Cost}(\mathcal{M}) = \sum_{(\mathcal{I},\mathcal{I}') \in \mathcal{M}} d_I(\mathcal{I}, \mathcal{I}') + \sum_{\mathcal{I} \in \mathcal{M}_{\mathcal{S}}^c} \delta(\mathcal{I}) + \sum_{\mathcal{I}' \in \mathcal{M}_{\mathcal{S}'}^c} \delta(\mathcal{I}'),$$

denotes the cost of the matching $\mathcal{M}$. First we consider metric condition (i) (identity of indiscernibles). Supposing we have $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_n)$ and $\mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_m)$ with

$\mathcal{S} = \mathcal{S}'$, this implies that $n = m$ and

$$\mathcal{I}_i = \mathcal{I}'_i \quad \text{for } i = 1, \ldots, n$$

that is, all interactions are equal. As such, we can trivially construct a monotone matching $\mathcal{M}^*$ by pairing equivalent interactions, that is

$$\mathcal{M}^* = \{(\mathcal{I}_1, \mathcal{I}'_1), \ldots, (\mathcal{I}_n, \mathcal{I}'_n)\}, \tag{A.4.5}$$

which leads to the following upper bound

$$\begin{aligned} d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}') &\leq \mathrm{Cost}(\mathcal{M}^*) \\ &= \sum_{i=1}^{n} d_I(\mathcal{I}_i, \mathcal{I}'_i) = 0. \end{aligned}$$

Now, since $d_I(\cdot, \cdot)$ is a metric we have $d_I(\mathcal{I}, \mathcal{I}') \geq 0$, whilst $\delta(\mathcal{I}) > 0$ also by assumption, which together imply $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}') \geq 0$ for any sequences $\mathcal{S}$ and $\mathcal{S}'$. These two bounds combine to imply that when $\mathcal{S} = \mathcal{S}'$ we have $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}') = 0$, proving one direction of metric condition (i).

For the converse case, we first assume that $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}') = 0$, which implies both the sum of pairwise distances and penalisation terms must be zero (since all are sums of non-negative values). Moreover, since $\delta(\mathcal{I}) > 0$ this implies there must be no penalty terms. Thus if $\mathcal{M}^*$ is an optimal monotone matching then it must contain all entries of $\mathcal{S}$ and $\mathcal{S}'$. Observe this also implies $\mathcal{S}$ and $\mathcal{S}'$ must be of the same length. Furthermore, the only possible *monotone* matching which includes all entries of both sequences is that defined in eq. (A.4.5), which implies

$$\begin{aligned} d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}') &= \mathrm{Cost}(\mathcal{M}^*) \\ &= \sum_{i=1}^{n} d_I(\mathcal{I}_i, \mathcal{I}'_i) = 0, \end{aligned}$$

where we have applied the definition of $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S},\mathcal{S}')$ directly, using the fact that since $\mathcal{M}^*$ is the only possibly monotone matching of $\mathcal{S}$ and $\mathcal{S}'$ it must be optimal. Now, since $d_I(\mathcal{I},\mathcal{I}') \geq 0$ (since $d_I(\cdot,\cdot)$ is a metric), this implies

$$d_I(\mathcal{I}_i,\mathcal{I}'_i) = 0 \quad \text{for } i = 1,\dots n,$$

and since $d_I(\cdot,\cdot)$ itself satisfies the identity of indiscernibles, this implies

$$\mathcal{I}_i = \mathcal{I}'_i \quad \text{for } i = 1,\dots,n$$

from which we can conclude $\mathcal{S} = \mathcal{S}'$. This proves the converse case, confirming that $d_{\mathrm{E},\delta(\cdot)}$ satisfies metric condition (i).

The symmetry condition (ii) follows trivially from the symmetry of $d_I(\cdot,\cdot)$ and the penalisation terms.

Finally, we confirm metric condition (iii) (triangle inequality) is satisfied. The approach is almost identical to that applied in the proof of Proposition 3.5.2 (Appendix A.4.2) with one key difference: we must ensure all matchings are monotone. Given three interaction sequences

$$\mathcal{S}_X = (\mathcal{I}_1^X,\dots,\mathcal{I}_{n_X}^X) \quad \mathcal{S}_Y = (\mathcal{I}_1^Y,\dots,\mathcal{I}_{n_Y}^Y) \quad \mathcal{S}_Z = (\mathcal{I}_1^Z,\dots,\mathcal{I}_{n_Z}^Z)$$

we seek to show that

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X,\mathcal{S}_Y) \leq d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X,\mathcal{S}_Z) + d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_Z,\mathcal{S}_Y).$$

With $\mathcal{M}^*_{XZ}$ and $\mathcal{M}^*_{ZY}$ denoting optimal monotone matchings for $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X,\mathcal{S}_Z)$ and

$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_Z, \mathcal{S}_Y)$ respectively, that is

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X, \mathcal{S}_Z) = \mathrm{Cost}(\mathcal{M}^*_{XZ}) \qquad d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_Z, \mathcal{S}_Y) = \mathrm{Cost}(\mathcal{M}^*_{ZY})$$

observe these induce a matching $\mathcal{M}_{XY}$ of $\mathcal{S}_X$ and $\mathcal{S}_Y$ as follows

$$\mathcal{M}_{XY} = \{(\mathcal{I}^X_i, \mathcal{I}^Y_j) \,:\, (\mathcal{I}^X_i, \mathcal{I}^Z_k) \in \mathcal{M}^*_{XZ} \text{ and } (\mathcal{I}^Z_k, \mathcal{I}^Y_j) \in \mathcal{M}^*_{ZY} \text{ for some } \mathcal{I}^Z_k \in \mathcal{S}_Z\}$$

that is, we match entries of $\mathcal{S}_X$ and $\mathcal{S}_Y$ if they were matched to the same entry of $\mathcal{S}_Z$.

We now confirm $\mathcal{M}_{XY}$ is a monotone matching. Recall that $\mathcal{M}_{XY}$ is monotone if for any pairs $(\mathcal{I}^X_{i_1}, \mathcal{I}^Y_{j_1})$ and $(\mathcal{I}^X_{i_2}, \mathcal{I}^Y_{j_2})$ in $\mathcal{M}_{XY}$ we have

$$i_1 < i_2 \iff j_1 < j_2.$$

To show this holds, observe by definition of $\mathcal{M}_{XY}$ there exists $\mathcal{I}^Z_{k_1}$ and $\mathcal{I}^Z_{k_2}$ in $\mathcal{S}_Z$ such that

$$(\mathcal{I}^X_{i_1}, \mathcal{I}^Z_{k_1}) \in \mathcal{M}^*_{XZ} \quad (\mathcal{I}^Z_{k_1}, \mathcal{I}^Y_{j_1}) \in \mathcal{M}^*_{ZY}$$
$$(\mathcal{I}^X_{i_2}, \mathcal{I}^Z_{k_2}) \in \mathcal{M}^*_{XZ} \quad (\mathcal{I}^Z_{k_2}, \mathcal{I}^Y_{j_2}) \in \mathcal{M}^*_{ZY}$$

Furthermore, since $\mathcal{M}^*_{XZ}$ and $\mathcal{M}^*_{ZY}$ are monotone we have

$$i_1 < i_2 \iff k_1 < k_2 \qquad \text{and} \qquad k_1 < k_2 \iff j_1 < j_2$$

which therefore implies

$$i_1 < i_2 \iff k_1 < k_2 \iff j_1 < j_2,$$

as required. Hence $\mathcal{M}_{XY}$ is also monotone. With the induced matching being mono-

tone, observe that by definition of $d_{\mathrm{E},\delta(\cdot)}$ we have the following

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X, \mathcal{S}_Y) \leq \mathrm{Cost}(\mathcal{M}_{XY})$$

which implies the triangle inequality will hold if we can show the following inequality is satisfied

$$\mathrm{Cost}(\mathcal{M}_{XY}) \leq d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X, \mathcal{S}_Z) + d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_Z, \mathcal{S}_Y). \tag{A.4.6}$$

Observe this is almost identical to the scenario in the proof of Proposition 3.5.2 (Appendix A.4.2), where the inequality of eq. (A.4.3) was shown to hold to prove that $d_{\mathrm{M},\delta(\cdot)}$ satisfied the triangle inequality. Since the induced matching here is the same used therein, albeit applied to sequences rather than multisets, an identical argument can be used show that eq. (A.4.6) holds. For brevity, we will not repeat these steps here, assuming henceforth that eq. (A.4.6) holds, implying $d_{\mathrm{E},\delta(\cdot)}$ satisfies the triangle inequality and completing the proof.

□

Proof of Proposition 3.7.4. For ease of reference, recall the DTW distance between sequences $\mathcal{S}$ and $\mathcal{S}'$ is given by

$$d_{\mathrm{DTW}}(\mathcal{S}, \mathcal{S}') := \min_{\mathcal{C} \in \boldsymbol{\mathcal{C}}(\mathcal{S}, \mathcal{S}')} \left\{ \sum_{(\mathcal{I}, \mathcal{I}') \in \mathcal{C}} d_I(\mathcal{I}, \mathcal{I}') \right\}$$

where $\boldsymbol{\mathcal{C}}(\mathcal{S}, \mathcal{S}')$ denotes the set of couplings between $\mathcal{S}$ and $\mathcal{S}'$.

Observe the symmetry condition (ii) follows trivially from the symmetry of the ground distance $d_I(\cdot, \cdot)$, by virtue of it being a metric.

All that remains is to show both conditions (i) and (iii) are violated. In both cases, we do so by providing counterexamples. Beginning with (i) (identity of indis-

cernibles), consider the following two sequences

$$\begin{aligned} \mathcal{S}_X &= (\mathcal{I}_1^X) & \mathcal{S}_Y &= (\mathcal{I}_1^Y, \mathcal{I}_2^Y) \\ &= (\tilde{\mathcal{I}}) & &= (\tilde{\mathcal{I}}, \tilde{\mathcal{I}}) \end{aligned}$$

where $\tilde{\mathcal{I}} \in \boldsymbol{\mathcal{I}}^*$ denotes an arbitrary interaction. Clearly, we have $\mathcal{S}_X \neq \mathcal{S}_Y$. However, there is only one valid coupling of $\mathcal{S}_X$ and $\mathcal{S}_Y$, namely $\mathcal{C}_{XY} = ((\mathcal{I}_1^X, \mathcal{I}_1^Y), (\mathcal{I}_1^X, \mathcal{I}_2^Y))$, implying the DTW distance between $\mathcal{S}_X$ and $\mathcal{S}_Y$ is given by

$$\begin{aligned} d_{\mathrm{DTW}}(\mathcal{S}_X, \mathcal{S}_Y) &= \sum_{(\mathcal{I}^X, \mathcal{I}^Y) \in \mathcal{C}_{XY}} d_I(\mathcal{I}^X, \mathcal{I}^Y) \\ &= d_I(\mathcal{I}_1^X, \mathcal{I}_1^Y) + d_I(\mathcal{I}_1^X, \mathcal{I}_2^Y) \\ &= d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}) + d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}) = 0 \end{aligned}$$

thus violating condition (i).

Turning now to condition (iii) (triangle inequality), consider the following three sequences

$$\begin{aligned} \mathcal{S}_X &= (\mathcal{I}_1^X, \mathcal{I}_2^X) & \mathcal{S}_Y &= (\mathcal{I}_1^Y) & \mathcal{S}_Z &= (\mathcal{I}_1^Z) \\ &= (\tilde{\mathcal{I}}, \tilde{\mathcal{I}}) & &= (\tilde{\mathcal{I}}') & &= (\tilde{\mathcal{I}}) \end{aligned}$$

where $\tilde{\mathcal{I}}, \tilde{\mathcal{I}}' \in \boldsymbol{\mathcal{I}}^*$ with $\tilde{\mathcal{I}} \neq \tilde{\mathcal{I}}'$. Now, the only valid coupling of $\mathcal{S}_X$ and $\mathcal{S}_Z$ is given by $\mathcal{C}_{XZ} = ((\mathcal{I}_1^X, \mathcal{I}_1^Z), (\mathcal{I}_2^X, \mathcal{I}_1^Z))$, similarly the only coupling of $\mathcal{S}_Z$ and $\mathcal{S}_Y$ is given by $\mathcal{C}_{ZY} = ((\mathcal{I}_1^Z, \mathcal{I}_1^Y))$, whilst for $\mathcal{S}_X$ and $\mathcal{S}_Y$ this will be $\mathcal{C}_{XY} = ((\mathcal{I}_1^X, \mathcal{I}_1^Y), (\mathcal{I}_2^X, \mathcal{I}_1^Y))$. This therefore implies

$$\begin{aligned} d_{\mathrm{DTW}}(\mathcal{S}_X, \mathcal{S}_Y) &= \sum_{(\mathcal{I}^X, \mathcal{I}^Y) \in \mathcal{C}_{XY}} d_I(\mathcal{I}^X, \mathcal{I}^Y) \\ &= d_I(\mathcal{I}_1^X, \mathcal{I}_1^Y) + d_I(\mathcal{I}_2^X, \mathcal{I}_1^Y) \\ &= d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}') + d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}') \\ &= 2d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}') \end{aligned}$$

whilst

$$\begin{aligned} d_{\mathrm{DTW}}(\mathcal{S}_X, \mathcal{S}_Z) + d_{\mathrm{DTW}}(\mathcal{S}_Z, \mathcal{S}_Y) &= \sum_{(\mathcal{I}^X, \mathcal{I}^Z) \in \mathcal{C}_{XZ}} d_I(\mathcal{I}^X, \mathcal{I}^Z) + \sum_{(\mathcal{I}^Z, \mathcal{I}^Y) \in \mathcal{C}_{ZY}} d_I(\mathcal{I}^Z, \mathcal{I}^Y) \\ &= \left[d_I(\mathcal{I}_1^X, \mathcal{I}_1^Z) + d_I(\mathcal{I}_2^X, \mathcal{I}_1^Z)\right] + \left[d_I(\mathcal{I}_1^Z, \mathcal{I}_1^Y)\right] \\ &= \left[d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}) + d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}})\right] + \left[d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}')\right] \\ &= d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}') \end{aligned}$$

which, since $d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}') > 0$ (as $\tilde{\mathcal{I}} \neq \tilde{\mathcal{I}}'$ and $d_I(\cdot, \cdot)$ is a metric) implies

$$d_{\mathrm{DTW}}(\mathcal{S}_X, \mathcal{S}_Y) = 2d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}') > d_I(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}') = d_{\mathrm{DTW}}(\mathcal{S}_X, \mathcal{S}_Z) + d_{\mathrm{DTW}}(\mathcal{S}_Z, \mathcal{S}_Y)$$

and thus (iii) is violated, as desired. This completes the proof. □

A.4.4 Pseudocode

Algorithm 1: Evaluating edit distance $d_{\mathrm{E},\delta(\cdot)}$

Data: Interaction sequences $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ and $\mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_M)$
Result: $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}')$ (Definition 3.7.1)
$C \in \mathbb{R}^{(N+1)\times(M+1)}$;
$C_{11} = 0$;
$C_{(i+1)1} = C_{i1} + \delta(\mathcal{I}_i)$ (for $i = 1, \ldots, N$);
$C_{1(j+1)} = C_{1j} + \delta(\mathcal{I}'_j)$ (for $j = 1, \ldots, M$);
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, m$ **do**

$$C_{(i+1)(j+1)} = \min \begin{cases} C_{ij} + d_I(\mathcal{I}_i, \mathcal{I}'_j) \\ C_{i(j+1)} + \delta(\mathcal{I}_i) \\ C_{(i+1)j} + \delta(\mathcal{I}'_j) \end{cases}$$

  end
end
return $C_{(N+1)(M+1)}$

Algorithm 2: Evaluating edit distance $d_{\mathrm{E},\delta(\cdot)}$ (light memory)

Data: Sequences $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ and $\mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_M)$
Result: $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}')$ (Definition 3.7.1)
$Z^{\mathrm{prev}}, Z^{\mathrm{curr}} \in \mathbb{R}^{(M+1)}$;
$Z_1^{\mathrm{prev}} = 0$, $Z_1^{\mathrm{curr}} = 0$;
$Z_{i+1}^{\mathrm{prev}} = Z_i^{\mathrm{prev}} + \delta(\mathcal{I}'_i)$ (for $i = 1, \ldots, M$);
for $i = 1, \ldots, n$ **do**
  $Z_1^{\mathrm{curr}} = Z_1^{\mathrm{curr}} + \delta(\mathcal{I}_i)$;
  for $j = 1, \ldots, m$ **do**
    $$Z_{j+1}^{\mathrm{curr}} = \min \begin{cases} Z_j^{\mathrm{prev}} + d_I(\mathcal{I}_i, \mathcal{I}'_j) \\ Z_{j+1}^{\mathrm{prev}} + \delta(\mathcal{I}_i) \\ Z_j^{\mathrm{curr}} + \delta(\mathcal{I}'_j) \end{cases}$$
  end
  $Z^{\mathrm{prev}} = Z^{\mathrm{curr}}$
end
return Z_{M+1}^{curr}

Algorithm 3: Evaluating dynamic time warping distance d_{DTW}

Data: Sequences $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ and $\mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_M)$
Result: $d_{\mathrm{DTW}}(\mathcal{S}, \mathcal{S}')$ (Definition 3.7.3)
$C \in \mathbb{R}^{(N+1)\times(M+1)}$;
$C_{11} = 0$;
$C_{(i+1)1} = \infty$ (for $i = 1, \ldots, N$);
$C_{1(j+1)} = \infty$ (for $j = 1, \ldots, M$);
for $i = 1, \ldots, N$ **do**
  for $j = 1, \ldots, M$ **do**
    $C_{(i+1)(j+1)} = d_I(\mathcal{I}_i, \mathcal{I}'_j) + \min\{C_{ij}, C_{(i+1)j}, C_{i(j+1)}\}$
  end
end
return $C_{(N+1)(M+1)}$

Algorithm 4: Evaluating dynamic time warping distance d_{DTW} (light memory)

Data: Sequences $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ and $\mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_M)$
Result: $d_{\mathrm{DTW}}(\mathcal{S}, \mathcal{S}')$ (Definition 3.7.3)
$Z^{\mathrm{prev}}, Z^{\mathrm{curr}} \in \mathbb{R}^{(M+1)}$;
$Z_1^{\mathrm{prev}} = 0$, $Z_1^{\mathrm{curr}} = \infty$;
$Z_{i+1}^{\mathrm{prev}} = \infty$ (for $i = 1, \ldots, M$);
for $i = 1, \ldots, N$ **do**
  for $j = 1, \ldots, M$ **do**
    $Z_{j+1}^{\mathrm{curr}} = d_I(\mathcal{I}_i, \mathcal{I}'_j) + \min\{Z_j^{\mathrm{prev}}, Z_j^{\mathrm{curr}}, Z_{j+1}^{\mathrm{prev}}\}$
  end
  $Z^{\mathrm{prev}} = Z^{\mathrm{curr}}$
end
return Z_{M+1}^{curr}

Algorithm 5: Evaluating LCS distance d_{LCS}

Data: Paths $\mathcal{I} = (x_1, \ldots, x_n)$ and $\mathcal{I}' = (y_1, \ldots, y_m)$
Result: $d_{\mathrm{LCS}}(\mathcal{I}, \mathcal{I}')$ (Section 3.4)
$C \in {\mathbb{Z}_+}^{(n+1)\times(m+1)}$;
$C_{11} = 0$;
$C_{(i+1)1} = i$ (for $i = 1, \ldots, n$);
$C_{1(j+1)} = j$ (for $j = 1, \ldots, m$);
$\delta = 0$;
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, m$ **do**
    $$C_{(i+1)(j+1)} = \min \begin{cases} C_{i(j+1)} + 1 \\ C_{(i+1)j} + 1 \\ C_{ij} + 2 \cdot \mathbb{1}(x_i \neq y_j), \end{cases}$$
  end
end
return $C_{(n+1)(m+1)}$

Algorithm 6: Evaluating LCS distance d_{LCS} (light memory)

Data: Paths $\mathcal{I} = (x_1, \ldots, x_n)$ and $\mathcal{I}' = (y_1, \ldots, y_m)$
Result: $d_{\mathrm{LSP}}(\mathcal{I}, \mathcal{I}')$ (Section 3.4)
$Z^{\mathrm{prev}}, Z^{\mathrm{curr}} \in \mathbb{Z}_+^{(m+1)}$;
$Z^{\mathrm{prev}}_{i+1} = Z^{\mathrm{curr}}_{i+1} = i$ (for $i = 0, \ldots, m$);
for $i = 1, \ldots, n$ **do**
  $Z^{\mathrm{curr}}_1 = Z^{\mathrm{curr}}_1 + 1$;
  for $j = 1, \ldots, m$ **do**

$$Z^{\mathrm{curr}}_{j+1} = \min \begin{cases} Z^{\mathrm{prev}}_{j+1} + 1 \\ Z^{\mathrm{curr}}_{j} + 1 \\ Z^{\mathrm{prev}}_{j} + 2 \cdot \mathbb{1}(x_i \neq y_i) \end{cases}$$

  end
  $Z^{\mathrm{prev}} = Z^{\mathrm{curr}}$;
end
return Z^{curr}_{m+1}

Algorithm 7: Evaluating LSP distance d_{LSP}

Data: Paths $\mathcal{I} = (x_1, \ldots, x_n)$ and $\mathcal{I}' = (y_1, \ldots, y_m)$
Result: $d_{\mathrm{LSP}}(\mathcal{I}, \mathcal{I}')$ (Section 3.4)
$Q \in {\mathbb{Z}_+}^{(n+1)\times(m+1)}$;
$Q_{11} = 0$;
$Q_{(i+1)1} = 0$ (for $i = 1, \ldots, n$);
$Q_{1(j+1)} = 0$ (for $j = 1, \ldots, m$);
$\delta = 0$;
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, m$ **do**
    if $x_i = y_j$ **then**
      $Q_{(i+1)(j+1)} = Q_{ij} + 1$
      $\delta = \max\left(z, Q_{(i+1)(j+1)}\right)$
    else
      $Q_{(i+1)(j+1)} = 0$
    end
  end
end
return $n + m - 2\delta$

Algorithm 8: Evaluating LSP distance d_{LSP} (light memory)

Data: Paths $\mathcal{I} = (x_1, \ldots, x_n)$ and $\mathcal{I}' = (y_1, \ldots, y_m)$
Result: $d_{\mathrm{LSP}}(\mathcal{I}, \mathcal{I}')$ (Section 3.4)
$Z^{\mathrm{prev}}, Z^{\mathrm{curr}} \in \mathbb{Z}_+^{(m+1)}$;
$Z_{i+1}^{\mathrm{prev}} = Z_{i+1}^{\mathrm{curr}} = 0$ (for $i = 0, \ldots, m$);
$\delta = 0$;
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, m$ **do**
    if $x_i = y_j$ **then**
      $Z_{j+1}^{\mathrm{curr}} = Z_j^{\mathrm{prev}} + 1$
      $\delta = \max\left(z, Z_{j+1}^{\mathrm{curr}}\right)$
    else
      $Z_{j+1}^{\mathrm{curr}} = 0$
    end
  end
  $Z^{\mathrm{prev}} = Z^{\mathrm{curr}}$
end
return $n + m - 2\delta$

Appendix B

Appendix to Chapter 4

B.1 Sample spaces

In this section, we formally define the sample spaces of the SIS and SIM models introduced in Section 4.2.1. In addition, we define the some finite versions thereof obtained by bounding dimensions, which we recommend working with in practice. For further elaboration on this recommendation, including justifications of the rationale, an outline of how to alter our MCMC algorithms to ensure the bound constraints are met, and discussions on how to choose the bounds in practice, see Appendix B.4.

B.1.1 Infinite spaces

Recall from Definitions 4.2.1 and 4.2.2 that the SIS and SIM models define distributions over the spaces of *all* interaction sequences and multisets, respectively. Given the vertex set $\mathcal{V}$ we first define the space of all interactions, that is, paths, as follows

$$\boldsymbol{\mathcal{I}}^* := \{(x_1, \ldots, x_n) \,:\, x_i \in \mathcal{V}, n \geq 1\},$$

with which we can define the space of interaction sequences $\boldsymbol{\mathcal{S}}^*$ in the following manner

$$\boldsymbol{\mathcal{S}}^* := \{(\mathcal{I}_1, \ldots, \mathcal{I}_N) \,:\, \mathcal{I}_i \in \boldsymbol{\mathcal{I}}^*, N \geq 1\},$$

moreover, with $\mathcal{E}_{\mathcal{S}}$ denoting the multiset obtained from the sequence $\mathcal{S}$ by disregarding the order of paths therein, the space of interaction multisets $\boldsymbol{\mathcal{E}}^*$ can be defined as follows

$$\boldsymbol{\mathcal{E}}^* := \{\mathcal{E}_{\mathcal{S}} \,:\, \mathcal{S} \in \boldsymbol{\mathcal{S}}^*\}$$

where here we abuse notation slightly, since we can have $\mathcal{E}_{\mathcal{S}} = \mathcal{E}_{\mathcal{S}'}$ for $\mathcal{S} \neq \mathcal{S}'$ (when equal up to a permutation of interactions), but we just assume such values have been included once and so $\boldsymbol{\mathcal{E}}^*$ is a set and not a multiset.

Note that $\boldsymbol{\mathcal{E}}^*$ also admits another interpretation as a partitioning of $\boldsymbol{\mathcal{S}}^*$ into equivalence classes. To see this, first define an equivalence relation on $\boldsymbol{\mathcal{S}}^*$ via permutations, in particular we write $\mathcal{S} \overset{p}{\sim} \mathcal{S}'$ if there is some permutation σ such that $\mathcal{S}' = \mathcal{S}^\sigma$, where $\mathcal{S}^\sigma = (\mathcal{I}_{\sigma(1)}, \ldots, \mathcal{I}_{\sigma(N)})$ is the interaction sequence obtained by permuting the interactions of $\mathcal{S}$ via σ. Now, observe that each $\mathcal{E} \in \boldsymbol{\mathcal{E}}^*$ can be seen as an equivalence class of interaction sequences obtained via $\overset{p}{\sim}$, that is

$$\mathcal{E} = \{\mathcal{S} \in \boldsymbol{\mathcal{S}}^* \,:\, \mathcal{S} \overset{p}{\sim} \tilde{\mathcal{S}}\}$$

where $\tilde{\mathcal{S}}$ denotes some arbitrary ordering of the interactions of $\mathcal{E}$. Thus, $\boldsymbol{\mathcal{E}}^*$ is in a sense the union of such sets and partitions $\boldsymbol{\mathcal{S}}^*$.

B.1.2 Bounded spaces

In this section, we define bounded analogues of the infinite spaces introduced the in preceding section. With regards to the objects we consider, there are two things we can bound: (i) the size of paths and (ii) the number of paths. Referring to these as the

inner and outer dimensions respectively, we specify two integers K and L bounding their values and define our sample spaces accordingly. Assuming that the vertex set $\mathcal{V}$ is fixed, and $K \in \mathbb{Z}_{\geq 1}$ we let

$$\boldsymbol{\mathcal{I}}^*_K := \{(x_1, \ldots, x_n) \,:\, x_i \in \mathcal{V}, 1 \leq n \leq K\}$$

denote the space of paths up to length K, and then with $L \in \mathbb{Z}_{\geq 1}$ we let

$$\boldsymbol{\mathcal{S}}^*_{K,L} := \{(\mathcal{I}_1, \ldots, \mathcal{I}_N) \,:\, \mathcal{I}_i \in \boldsymbol{\mathcal{I}}^*_K, 1 \leq N \leq L\},$$

denote the space of interaction sequences with at most L paths of length at most K. The analogous bounded space of interaction multisets in then given by

$$\boldsymbol{\mathcal{E}}^*_{K,L} := \{\mathcal{E}_{\mathcal{S}} \,:\, \mathcal{S} \in \boldsymbol{\mathcal{S}}^*_{K,L}\},$$

where as in the definition of $\boldsymbol{\mathcal{E}}^*$ in Appendix B.1.1 one can have $\mathcal{E}_{\mathcal{S}} = \mathcal{E}_{\mathcal{S}'}$ for $\mathcal{S} \neq \mathcal{S}'$, but we here just assume such values have been included once, and so $\boldsymbol{\mathcal{E}}^*_{K,L}$ is indeed a set, not a multiset.

B.2 Simulation studies: extra details

This section contains supporting details for the simulation studies of Section 4.4. In particular, we discuss how parameters were chosen for the simulation of Section 4.4.2, and provide a derivation for the posterior predictive approximation used in Section 4.4.3.

B.2.1 Posterior concentration parameter choices

Recall that in the simulation of Section 4.4.2 we resampled the true mode via

$$\mathcal{S}_{\text{true}} \sim \text{Hollywood}(\alpha, -\alpha V, \nu)$$

where $V = 20$ and $\nu = \text{TrPoisson}(3, 1, 10)$, whilst $\alpha < 0$ we varied. As mentioned in Section 4.4.2, the parameter α can be seen to control the tail of the vertex count distribution. As such, rather than choosing α on an even grid we instead consult a summary measure quantifying the 'heavy-tailedness' of the degree distribution, before choosing values so as to evenly represent different structures for $\mathcal{S}_{\text{true}}$ (as quantified by this degree distribution).

For a given observation $\mathcal{S}$, recall the following definition

$$k_{\mathcal{S}}(v) := \text{\# times } v \text{ appears in } \mathcal{S},$$

which for each $\mathcal{S}$ implies a sample $\{k_{\mathcal{S}}(v) \,:\, v \in \mathcal{V},\, k_{\mathcal{S}}(v) > 0\}$, similar to the degree distribution. The summary measure we considered was the 95% quantile of this sample.

Through simulation, we examined how α controls the expected value of this 95% quantile (expected since $\mathcal{S}$ is sample randomly from a Hollywood model). In particular, for a range of α values, we (i) drew a sample $\{\mathcal{S}^{(i)}\}_{i=1}^{n}$, from a Hollywood$(\alpha, -\alpha V, \nu)$ model, taking ν and V as above, drawing a total of $N = 10$ paths in each case, then (ii) for $i = 1, \ldots, n$ we evaluated the 95% quantile of the sample $\{k_{\mathcal{S}^{(i)}}(v) \,:\, v \in \mathcal{V},\, k_{\mathcal{S}^{(i)}}(v) > 0\}$, before returning the mean value of these quantiles.

Figure B.2.1 summarises the output with $n = 1000$ samples, where circular markers show the mean quantiles. Towards choosing simulation parameters, we next constructed a function mapping all $\alpha < 0$ to an expected quantile via a linear interpolation, as shown in Figure B.2.1 by the dashed line, which allowed us to select α (red

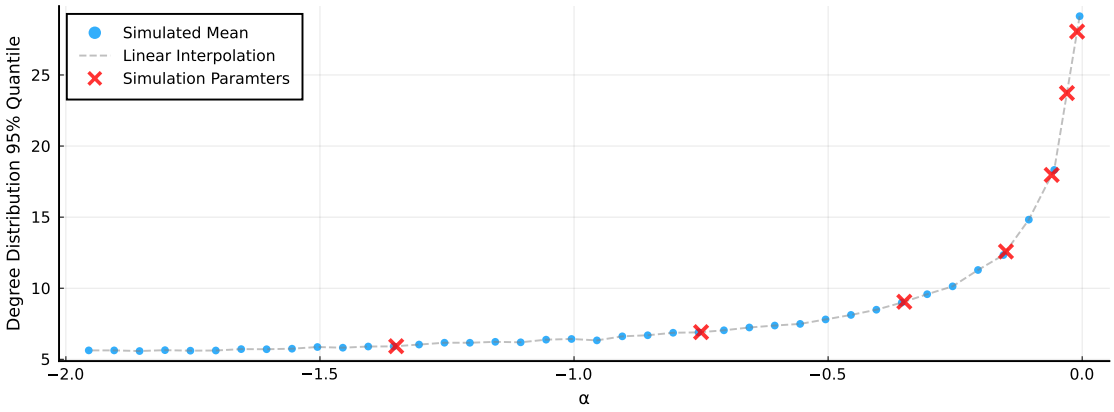


Figure B.2.1: Summary of Hollywood model simulation used to select parameters for simulation of Section 4.4.2. Plot shows simulated mean degree distribution 95% quantiles for Hollywood$(\alpha, -\alpha V, \nu)$ model, where $V = 20$, $\nu = \text{TrPoisson}(3, 1, 10)$ and α varies. Via linear interpolation (dashed line), we choose α values (crosses) to get an even spread over the expected degree distribution quantiles.

crosses) providing an even spread of expected degree-distribution 95% qauntiles.

B.2.2 Posterior predictive for missing entries

Here we show how one can obtain an approximation for the missing-entry posterior predictive using a sample from the posterior, as used in Section 4.4.3. First, observe that any sample $\{(\mathcal{S}_i^m, \gamma_i)\}_{i=1}^m$ from the posterior implies the following atomic approximation thereof

$$\hat{p}(\mathcal{S}^m, \gamma | \{\mathcal{S}^{(i)}\}_{i=1}^n) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}(\mathcal{S}^m = \mathcal{S}_i^m) \cdot \delta(\gamma - \gamma_i) \tag{B.2.1}$$

where $\delta(\cdot)$ is the Dirac delta function.

As in Section 4.4.3, with $\mathcal{S}_x$ denoting the observation with missing entry filled in to be x, then given some parameters $(\mathcal{S}^m, \gamma)$ we have the true predictive for x given by

$$p(x | \mathcal{S}^m, \gamma, \mathcal{S}_{-x}) := \frac{1}{Z(\mathcal{S}^m, \gamma, \mathcal{S}_{-x})} \exp\{-\gamma d_S(\mathcal{S}_x, \mathcal{S}^m)\}$$

with

$$Z(\mathcal{S}^m, \gamma, \mathcal{S}_{-x}) = \sum_{x \in \mathcal{V}} \exp\{-\gamma d_S(\mathcal{S}_x, \mathcal{S}^m)\}.$$

The posterior predictive is now obtained by averaging with respect to the posterior

$$p(x|\{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x}) = \sum_{\mathcal{S}^m \in \boldsymbol{\mathcal{S}}^*} \int_{\mathbb{R}_+} p(x|\mathcal{S}^m, \gamma, \mathcal{S}_{-x}) p(\mathcal{S}^m, \gamma|\{\mathcal{S}^{(i)}\}_{i=1}^n) d\gamma,$$

which we can now approximate by substituting in (B.2.1) as follows

$$\begin{aligned}
\hat{p}(x|\{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x}) &:= \sum_{\mathcal{S}^m \in \boldsymbol{\mathcal{S}}^*} \int_{\mathbb{R}_+} p(x|\mathcal{S}^m, \gamma, \mathcal{S}_{-x}) \hat{p}(\mathcal{S}^m, \gamma|\{\mathcal{S}^{(i)}\}_{i=1}^n) d\gamma \\
&= \sum_{\mathcal{S}^m \in \boldsymbol{\mathcal{S}}^*} \int_{\mathbb{R}_+} p(x|\mathcal{S}^m, \gamma) \left(\frac{1}{m} \sum_{i=1}^m \mathbb{1}\left(\mathcal{S}^m = \mathcal{S}_i^m\right) \delta(\gamma - \gamma_i) \right) d\gamma \\
&= \frac{1}{m} \sum_{i=1}^m p(x|\mathcal{S}_i^m, \gamma_i),
\end{aligned}$$

which is exactly as stated in Section 4.4.3.

A pragmatic note here is that as the posterior concentrates the number of unique values in the sample $\{\mathcal{S}_i^m\}_{i=1}^m$ will typically not be too large. Since we need only evaluate the distance metric (which is typically quite costly) at these values, this predictive is feasible to evaluate.

B.3 Montonicity of the entropy

Here we examine the entropy for the SIS and SIM model families (Definitions 4.2.1 and 4.2.2 respectively), in particular, we confirm it is monotonic with respect to the dispersion.

For the SIS model, recall the entropy is given by

$$\begin{aligned}
H(\mathcal{S}^m, \gamma) &= -\mathbb{E}[\log p(\mathcal{S}|\mathcal{S}^m, \gamma)] \\
&= -\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*} \log\left(\frac{\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}}{Z(\mathcal{S}^m,\gamma)}\right)\frac{\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}}{Z(\mathcal{S}^m,\gamma)} \\
&= -\Bigg(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*} -\gamma d_S(\mathcal{S},\mathcal{S}^m)\frac{\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}}{Z(\mathcal{S}^m,\gamma)} \\
&\qquad\qquad - \log Z(\mathcal{S}^m,\gamma)\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}\frac{\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}}{Z(\mathcal{S}^m,\gamma)}\Bigg) \\
&= \gamma\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*} d_S(\mathcal{S},\mathcal{S}^m)\frac{\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}}{Z(\mathcal{S}^m,\gamma)}\right) + \log Z(\mathcal{S}^m,\gamma) \\
&= \gamma \times \mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)] + \log Z(\mathcal{S}^m,\gamma).
\end{aligned}$$

Unfortunately, as was the case for the normalising constant $Z(\mathcal{S}^m,\gamma)$ (Section 4.2.3), since $\boldsymbol{\mathcal{S}}^*$ is infinite we have no guarantee that $H(\mathcal{S}^m,\gamma)$ will exist. However, what we can say is that, when $H(\mathcal{S}^m,\gamma)$ exists, it is monotonic in γ. To show this, we first differentiate $H(\mathcal{S}^m,\gamma)$ with respect to γ

$$\frac{\partial}{\partial\gamma}H(\mathcal{S}^m,\gamma) = \frac{\partial}{\partial\gamma}\mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)] + \mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)] + \frac{\partial}{\partial\gamma}\log Z(\mathcal{S}^m,\gamma)$$

where one has

$$\begin{aligned}
\frac{\partial}{\partial\gamma}\log Z(\mathcal{S}^m,\gamma) &= \frac{\frac{\partial}{\partial\gamma}Z(\mathcal{S}^m,\gamma)}{Z(\mathcal{S}^m,\gamma)} \\
&= \frac{1}{Z(\mathcal{S}^m,\gamma)}\frac{\partial}{\partial\gamma}\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}\exp\left\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\right\}\right) \\
&= \frac{1}{Z(\mathcal{S}^m,\gamma)}\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}\frac{\partial}{\partial\gamma}\exp\left\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\right\} \\
&= \frac{1}{Z(\mathcal{S}^m,\gamma)}\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}(-d_S(\mathcal{S},\mathcal{S}^m))\exp\left\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\right\} \\
&= -\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*} d_S(\mathcal{S},\mathcal{S}^m)\frac{1}{Z(\mathcal{S}^m,\gamma)}\exp\left\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\right\} \\
&= -\mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)],
\end{aligned} \tag{B.3.1}$$

thus implying

$$\frac{\partial}{\partial\gamma}H(\mathcal{S}^m,\gamma) = \frac{\partial}{\partial\gamma}\mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)].$$

Now, we have

$$\begin{aligned}
\frac{\partial}{\partial\gamma}\mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)] &= \frac{\partial}{\partial\gamma}\left(\frac{1}{Z(\mathcal{S}^m,\gamma)}\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*} d_S(\mathcal{S},\mathcal{S}^m)\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}\right) \\
&= -\frac{\frac{\partial}{\partial\gamma}Z(\mathcal{S}^m,\gamma)}{Z(\mathcal{S}^m,\gamma)^2}\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*} d_S(\mathcal{S},\mathcal{S}^m)\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}\right) \\
&\qquad - \frac{1}{Z(\mathcal{S}^m,\gamma)}\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*} d_S(\mathcal{S},\mathcal{S}^m)^2\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}\right) \\
&= -\frac{\frac{\partial}{\partial\gamma}Z(\mathcal{S}^m,\gamma)}{Z(\mathcal{S}^m,\gamma)^2}\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*} d_S(\mathcal{S},\mathcal{S}^m)\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}\right) \\
&\qquad - \frac{1}{Z(\mathcal{S}^m,\gamma)}\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*} d_S(\mathcal{S},\mathcal{S}^m)^2\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}\right) \\
&= -\frac{\frac{\partial}{\partial\gamma}Z(\mathcal{S}^m,\gamma)}{Z(\mathcal{S}^m,\gamma)}\mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)] - \mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)^2] \qquad \text{(B.3.2)} \\
&= \mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)]^2 - \mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)^2] \qquad \text{(B.3.3)}
\end{aligned}$$

$$= \mathbb{V}\text{ar}[d_S(\mathcal{S}, \mathcal{S}^m)],$$

where (B.3.3) follows from (B.3.2) by applying (B.3.1). Now, observe that if $\frac{\partial}{\partial\gamma} H(\mathcal{S}^m, \gamma) > 0$ this implies $H(\mathcal{S}^m, \gamma)$ is monotonic in γ, as desired. By the derivations above, this is equivalent to saying we have monotonicity provided $\mathbb{V}\text{ar}[d_S(\mathcal{S}, \mathcal{S}^m)] > 0$. This result essentially says we have monotonicity of the entropy with respect to γ provided our distribution is not a point mass.

Similar derivations can be obtained for the multiset models (Definition 4.2.2) by a simple change of notation. For brevity, we do not repeat this here.

B.4 Bounding dimensions

As we mentioned in Section 4.2.1, in practice we recommend constraining the sample spaces to be finite, as defined in Appendix B.1.2. In this section, we will illustrate why we make this recommendation. We will also elaborate on how one might go about choosing the necessary bounds and discuss how our MCMC algorithms can be slightly altered to respect the imposed dimension constraints.

To illustrate the need for constraining the sample space we will show via simulation what can go wrong. In particular, we will show that one can, in certain scenarios, observe a divergence in dimension when sampling from models over an infinite space. We note the following will regard the SIS model, but analogous behaviour will be observable for the infinite version of the SIM model. Suppose we would like to sample from the SIS model of Definition 4.2.1 over the infinite space $\boldsymbol{\mathcal{S}}^*$ of all interaction sequences. As we have mentioned in Section 4.3.6, we cannot do this exactly, but we can obtain approximate samples via our iMCMC algorithm proposed therein. With this, for a given mode $\mathcal{S}^m$ and dispersion γ, we can obtain a chain $(\mathcal{S}_i)_{i=1}^M$ approximating a sample from the SIS model with these parameters, that is, a chain targeting

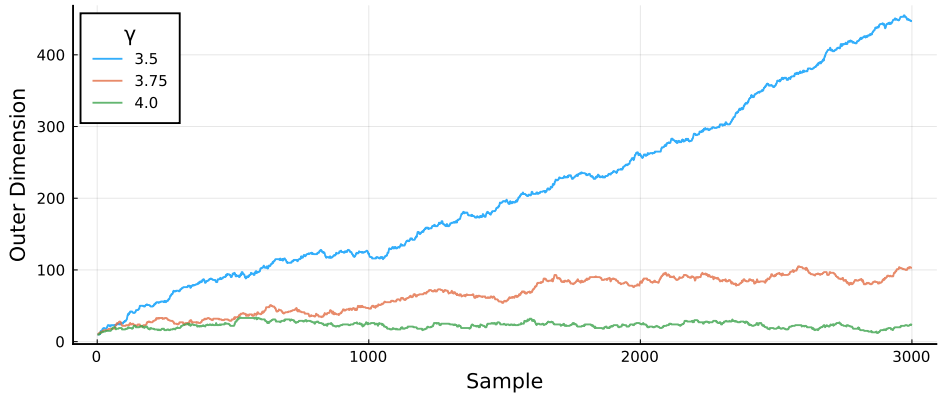


Figure B.4.1: Illustrating divergence in dimension for the SIS model over an infinite space. Each trace summarises an MCMC chain sampling from an SIS($\mathcal{S}^m, \gamma$) model over the space $\boldsymbol{\mathcal{S}}^*$ with the dispersion γ set at different values. Here we observe, for γ low enough, the number of paths (outer dimension) diverges.

the following

$$p(\mathcal{S}|\mathcal{S}^m, \gamma) \propto \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\}. \tag{B.4.1}$$

Figure B.4.1 summarises three such chains drawn with different values for the dispersion, where for each sample $\mathcal{S}_i = (\mathcal{I}_1^{(i)}, \ldots, \mathcal{I}_{N_i}^{(i)})$ we plot the number of paths N_i, or what we call the outer dimension. In each case, we initialised the chains at the mode $\mathcal{S}^m$, with a lag of 1 between samples and no burn-in. Here one can clearly see the dimensions of samples tends to be larger as γ decreases. Moreover, when $\gamma = 3.5$ the dimension appears to diverge, showing a clear upward trend.

Why does this happen? Observe that as γ goes to zero $p(\mathcal{S}|\mathcal{S}^m, \gamma)$ of (B.4.1) will converge to the uniform distribution over the space $\boldsymbol{\mathcal{S}}^*$. Though this might seem innocuous, one must remember that there are far more interaction sequences with large dimensions. For example, if $\mathcal{S}$ has n entries in total across all its paths, then there are V^n possible choices thereof. As such, with a uniform distribution over $\boldsymbol{\mathcal{S}}^*$, the probability of sampling an observation with large dimensions will be higher than those with smaller dimensions, leading to the observed divergence.

This implies there is always a chance, if γ is low enough, that the dimensions will diverge. This will inevitably cause computational issues when sampling from these

models via our iMCMC algorithm. Even if one does not first run out of memory, the cost of evaluating the distance $d_S(\mathcal{S}, \mathcal{S}^m)$ is very likely to grow with the dimension of $\mathcal{S}$, significantly slowing down the sampling time. This becomes ever more significant in the context of the algorithms we proposed to sample from the posterior in Section 4.3. Recall that in updating the dispersion (Section 4.3.3) we must sample auxiliary data from the model at γ and γ', the current value and the proposal, which we do via our iMCMC algorithm as above. Consequently, there is the chance one may have proposed a γ' for which the dimension will blow up when sampling the auxiliary data. Moreover, obtaining such samples will generally be more computationally cumbersome, increasing the time taken to obtain the auxiliary data, in turn slowing down the time taken to obtain the posterior samples. Ultimately, the result will be a posterior sampling scheme which is unstable and unpredictable in terms of run time.

This motivates our recommendation to constrain the sample space to $\boldsymbol{\mathcal{S}}^*_{K,L} \subseteq \boldsymbol{\mathcal{S}}^*$, as defined in Appendix B.4, where K and L represent the maximum path length and number of paths respectively. This will effectively place a lid on the possible dimension of samples, removing the possibility of divergence in dimensions. Note to sample from models over such constrained spaces we can use the exact same MCMC algorithms used in the infinite case. All one must do is set the probability of values outside of $\boldsymbol{\mathcal{S}}^*_{K,L}$ to zero, that is for each $\mathcal{S} \in \boldsymbol{\mathcal{S}}^*$ let

$$p(\mathcal{S}|\mathcal{S}^m, \gamma) \propto \begin{cases} \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} & \text{if } \mathcal{S} \in \boldsymbol{\mathcal{S}}^*_{K,L} \\ 0 & \text{if } \mathcal{S} \notin \boldsymbol{\mathcal{S}}^*_{K,L} \end{cases}$$

defining a distribution over the infinite space $\boldsymbol{\mathcal{S}}^*$ which we can target with our MCMC algorithm. Observe that, within the MCMC algorithm, if we are currently at state $\mathcal{S} \in \boldsymbol{\mathcal{S}}^*_{K,L}$ any proposal $\mathcal{S}' \notin \boldsymbol{\mathcal{S}}^*_{K,L}$ will always be rejected, since its acceptance probability will evaluate to zero. Hence we will obtain only samples from the constrained space,

as desired.

With the recommendation of bounding sample spaces comes the question of how to choose these bounds. We note this is only a question of interest when one is considering inference.[1] In this case, we will have observed a sample

$$\mathcal{S}^{(1)}, \ldots, \mathcal{S}^{(n)}$$

which we assume was drawn i.i.d. via

$$\mathcal{S}^{(i)} \sim \mathrm{SIS}(\mathcal{S}^m, \gamma)$$

where $\mathcal{S}^m$ and γ are some unknown model parameters. Notice assuming bounds K and L implies we must have $\mathcal{S}^{(i)} \in \boldsymbol{\mathcal{S}}^*_{K,L}$ for each of the observed samples. In this way, this informs the following thresholds for possible choices of K and L

$$K \geq \max_{i=1,\ldots,n} \left\{ \max_{j=1,\ldots,N^{(i)}} n_j^{(i)} \right\} \qquad L \geq \max_{i=1,\ldots,n} N^{(i)}$$

where $N^{(i)}$ is the number of paths in the ith observation and $n_j^{(i)}$ is the length of the jth path in the ith observation. As such, we recommend choosing bounds either at or close these thresholds, and indeed this is what we did for the data analysis of Section 4.5.

We finalise these discussions by noting that in constraining the sample space one can actually alter the interpretation of γ in the resulting model, in the sense that draws from the model with the same value of γ but different choices for K and L can look quite different in terms of the samples they generate. Though this might seem problematic, we note that the same applies to different choices of distance $d_S(\cdot, \cdot)$, the flex-

[1] If one is instead just sampling from the model, for example to examine the behaviour of the model with a particular distance $d_S(\cdot, \cdot)$, then the bounds can be set to personal preference, or the infinite space assumed, with the awareness that dimensions could diverge.

ibility of which is a key feature of our proposed methodology. In this way, one must accept that the interpretation of γ is context dependent. In any case, a pragmatic way to interpret an inferred value thereof is to instead use simulations from the model as we did in Section 4.5.2.

B.5 The iExchange algorithm

In this section, we outline the *iExchange* algorithm (Algorithm 9), a generalisation of exchange algorithm (Murray et al., 2006) obtained by incorporating the proposal generating mechanism of the iMCMC algorithm (Neklyudov et al., 2020). As we show, the iExchange algorithm is itself an iMCMC algorithm (with a particular form of involution), providing the necessary theoretical justification. For completeness, we give background details regarding both the exchange and iMCMC algorithms, before showing how they can be combined.

Algorithm 9: Involutive exchange (iExchnage) algorithm

Input: target density $p(\theta|\mathbf{x}) \propto p(\theta)\gamma(\mathbf{x}|\theta)/Z(\theta)$
Input: auxiliary density $q(u|\theta)$
Input: involution $f(\theta, u)$, i.e. $f^{-1}(\theta, u) = f(\theta, u)$
initialise θ
for $i = 1, \ldots, n$ **do**
  sample $u \sim q(u|\theta)$
  invoke involution $(\theta', u') = f(x, u)$
  sample $\mathbf{y} \sim p(\mathbf{y}|\theta')$
  evaluate $\alpha(\theta, \theta') = \min\left\{1, \frac{p(\theta')\gamma(\mathbf{x}|\theta')\gamma(\mathbf{y}|\theta)q(u'|\theta')}{p(\theta)\gamma(\mathbf{x}|\theta)\gamma(\mathbf{y}|\theta')q(u|\theta)} \left|\frac{\partial f(\theta,u)}{\partial(\theta,u)}\right|\right\}$
  $\theta_i = \begin{cases} \theta' & \text{with probability } \alpha(\theta, \theta') \\ \theta & \text{with probability } 1 - \alpha(\theta, \theta') \end{cases}$
  $\theta \leftarrow \theta_i$
end

Let us first set the context. We have some data $\mathbf{x}$ which is assumed to have been

drawn via a model $p(\mathbf{x}|\theta)$, where θ denote parameters, taking the following form

$$p(\mathbf{x}|\theta) = \frac{\gamma(\mathbf{x}|\theta)}{Z(\theta)} \tag{B.5.1}$$

where $Z(\theta) = \int \gamma(\mathbf{x}|\theta)d\mathbf{x}$ denotes its normalising constant, assumed to be *intractable*. If one is taking a Bayesian approach to inference and has specified a prior $p(\theta)$, this leads to the following posterior

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} \tag{B.5.2}$$

where $p(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta)d\theta$ is the marginal probability of the data, which in most cases is also intractable. Due to these two elements of intractability, such posteriors are often referred to as *doubly*-intractable (Murray et al., 2006). For example, the posteriors resulting from both our SIS and SIM models are doubly-intractable.

A typical approach to circumvent the intractability present in Bayesian posterior distributions is to use MCMC algorithms to sample from them, with the Metropolis-Hastings (MH) algorithm being a prevalent choice. However, for doubly-intractable posteriors, many standard MCMC algorithms are not feasible. To illustrate, consider using the MH algorithm. Here, with θ the current state and $q(\theta'|\theta)$ some proposal density, in a single iteration one would sample proposal θ' from $q(\theta'|\theta)$ and accept this with the following probability

$$\begin{aligned} \alpha(\theta, \theta') &= \min\left\{1, \frac{p(\theta'|\mathbf{x})q(\theta|\theta')}{p(\theta|\mathbf{x})q(\theta'|\theta)}\right\} \\ &= \min\left\{1, \frac{\gamma(\mathbf{x}|\theta')/Z(\theta')p(\theta')q(\theta|\theta')}{\gamma(\mathbf{x}|\theta)/Z(\theta)p(\theta)q(\theta'|\theta)}\right\}, \end{aligned} \tag{B.5.3}$$

so that, starting from some initial state θ_0 one obtains a sample $\{\theta_i\}_{i=1}^m$ which is (approximately) distributed according to $p(\theta|\mathbf{x})$. However, though the marginal probability $p(\mathbf{x})$ cancels out in (B.5.3), the normalising constants $Z(\theta)$ and $Z(\theta')$ do not.

Moreover, since these are by assumption intractable, $\alpha(\theta, \theta')$ cannot be evaluated, ruling out use of the MH algorithm.

This necessitates the proposal of specialised MCMC algorithms to sample from doubly-intractable posterior distributions, and herein lies the motivation for the exchange and iExchange algorithms.

B.5.1 Exchange algorithm

In this section, we give a high-level overview of the exchange algorithm (Algorithm 10), proposed by Murray et al. (2006). This is similar in structure to MH algorithm, but with some extra sampling in each iteration. Namely, one samples so-called *auxiliary data*, which subsequently appears in the acceptance probability, inducing cancellation of intractable normalising constants. Effectively, it targets an augmented distribution which admits the posterior of interest as its marginal (Murray et al., 2006).

As in the MH algorithm, we have some proposal distribution $q(\theta' \mid \theta)$ which is pre-specified. We also introduce an auxiliary data set $\mathbf{y}$ which lies in the same space as the observed data $\mathbf{x}$. Now, given current state θ a single iteration consists of the following

1. Sample proposal θ' via $q(\theta' \mid \theta)$
2. Sample auxiliary data $\mathbf{y} \mid \theta'$ via $p(\mathbf{y} \mid \theta')$ of (B.5.1) (sample from the model)
3. Evaluate acceptance probability

$$\begin{aligned} \alpha(\theta, \theta') &= \min \left\{ 1, \frac{p(\theta'|\mathbf{x}) q(\theta|\theta') p(\mathbf{y}|\theta)}{p(\theta|\mathbf{x}) q(\theta'|\theta) p(\mathbf{y}|\theta')} \right\} \\ &= \min \left\{ 1, \frac{p(\theta') \gamma(\mathbf{x}|\theta') \gamma(\mathbf{y}|\theta) q(\theta|\theta')}{p(\theta) \gamma(\mathbf{x}|\theta) \gamma(\mathbf{y}|\theta') q(\theta'|\theta)} \right\} \end{aligned} \tag{B.5.4}$$

4. With probability $\alpha(\theta, \theta')$ we move to state θ', otherwise we stay at θ.

Observe the absence of normalising constants here makes $\alpha(\theta, \theta')$ tractable. Repeating this a number of times, as summarised in Algorithm 10, produces a Markov chain admitting $p(\theta \,|\, \mathbf{x})$ as its stationary distribution (Murray et al., 2006). An alternative justification to that given by Murray et al. (2006) comes by viewing this as an instance of iMCMC, which we detail in the next section.

Algorithm 10: Exchange algorithm

Input: target density $p(\theta|\mathbf{x}) \propto p(\theta)\gamma(\mathbf{x}|\theta)/Z(\theta)$
Input: proposal distribution $q(\theta'|\theta)$
initialise θ;
for $i = 1, \ldots, n$ **do**
  sample θ' via $q(\theta'|\theta)$
  sample $\mathbf{y}$ via $p(\mathbf{y}|\theta')$ (from the model)
  evaluate $\alpha(\theta, \theta') = \min\left\{1, \frac{p(\theta')\gamma(\mathbf{x}|\theta')\gamma(\mathbf{y}|\theta)q(\theta|\theta')}{p(\theta)\gamma(\mathbf{x}|\theta)\gamma(\mathbf{y}|\theta')q(\theta'|\theta)}\right\}$
  $\theta_i = \begin{cases} \theta' & \text{with probability } \alpha(\theta, \theta') \\ \theta & \text{with probability } 1 - \alpha(\theta, \theta') \end{cases}$
  $\theta \leftarrow \theta_i$
end
Output: sample $\{\theta_i\}_{i=1}^n$

B.5.2 Involutive MCMC (iMCMC)

The iMCMC algorithm of Neklyudov et al. (2020) considers the problem of sampling from a general target distribution $p(x)$ over some space $\mathcal{X}$, for example, this might be our posterior from (B.5.2) (replacing x with θ). Like all MCMC algorithms, it does so by sampling a Markov chain admitting $p(x)$ as its stationary distribution, using in particular a combination of random sampling and involutive deterministic maps. The result is a very general framework which includes many well-known MCMC algorithms as special cases.

As the name suggests, iMCMC uses a particular type of deterministic function know as an *involution*. This is a function which serves as its own inverse, that is, if $f : \mathcal{X} \to \mathcal{X}$ then one has $f^{-1}(x) = f(x)$. Equivalently, a composition f with itself

leads to the identity

$$f(f(x)) = x.$$

Towards targeting $p(x)$ one introduces auxiliary variables $u \in \mathcal{U}$ with conditional density $q(u|x)$ over an auxiliary space $\mathcal{U}$ (which need not be equal to $\mathcal{X}$), augmenting the target as follows

$$p(x, u) = p(x)q(u|x)$$

which is now a distribution over $\mathcal{X} \times \mathcal{U}$. Observe this admits $p(x)$ as its marginal and hence one can obtain samples thereof by targeting $p(x, u)$ and disregarding the u samples. To do so, suppose an involution $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X} \times \mathcal{U}$ has been specified along with the auxiliary distribution $q(u|x)$. In structure reminiscent of the MH algorithm, a single iteration consists of the following. With current state (x, u), an auxiliary variable $u \in \mathcal{U}$ is first drawn from $q(u|x)$, before the involution f is invoked to get a proposal $(x', u') = f(x, u)$, which is subsequently accepted with the following probability

$$\begin{aligned}\alpha\left((x, u), (x', u')\right) &= \min\left\{1, \frac{p(f(x, u))}{p(x, u)}\left|\frac{\partial f(x, u)}{\partial(x, u)}\right|\right\} \\ &= \min\left\{1, \frac{p(x')q(u'|x')}{p(x)q(u|x)}\left|\frac{\partial f(x, u)}{\partial(x, u)}\right|\right\},\end{aligned}$$

leading to a Markov chain admitting $p(x, u)$ as its stationary distribution (Neklyudov et al., 2020, Proposition 2).

Observe that since auxiliary variables u are resampled in each iteration they do not need to be stored, and can instead be discarded as the algorithm proceeds. In this way, one may also drop their reference in the acceptance probability denoting this simply $\alpha(x, x')$. This leads to the algorithm to target $p(x)$ as outlined in Algorithm 11.

As mentioned, many known MCMC algorithms can be written in this form. For example, if one assumes $\mathcal{U} = \mathcal{X}$, with $q(x'|x)$ the auxiliary distribution and $f(x, x') = (x', x)$ the involution defined by swapping entries, then one obtains the Metropolis-Hastings algorithm with proposal distribution $q(x'|x)$. Further examples of MCMC

Algorithm 11: Involutive MCMC (iMCMC)

Input: target density $p(x)$
Input: auxiliary density $q(u|x)$
Input: involution $f(x, u)$
initialise x;
for $i = 1, \ldots, n$ **do**
  sample $u \sim q(u|x)$
  invoke involution $(x', u') = f(x, u)$
  evaluate $\alpha(x, x') = \min\left\{1, \frac{p(x')q(u'|x')}{p(x)q(u|x)} \left|\frac{\partial f(x,u)}{\partial(x,u)}\right|\right\}$
  $x_i = \begin{cases} x' & \text{with probability } \alpha(x, x') \\ x & \text{with probability } 1 - \alpha(x, x') \end{cases}$
  $x \leftarrow x_i$
end
Output: sample $\{x_i\}_{i=1}^n$

algorithms which can be cast in the iMCMC framework are given in Neklyudov et al. (2020), Appendix B.

Another iMCMC special case which is of relevance to us is the exchange algorithm. To see this, we let $u = (\theta', \mathbf{y})$, where $\mathbf{y}$ denotes the auxiliary data, as seen in Appendix B.5.1. Moreover, we define our involution as follows

$$f(\theta, u) = (\theta', (\theta, \mathbf{y})),$$

that is, we simply swap $\theta \leftrightarrow \theta'$. Observe we have

$$\begin{aligned} f(f(\theta, u)) &= f(f(\theta, (\theta', \mathbf{y})) \\ &= f(\theta', (\theta, \mathbf{y})) \\ &= (\theta, (\theta', \mathbf{y})) \\ &= (\theta, u) \end{aligned}$$

so that f is indeed an involution. We now derive the Jacobian term. For convenience, drop the inner parenthesis and write $(\theta, u) = (\theta, \theta', \mathbf{y})$, for which we have $f(\theta, \theta', \mathbf{y}) =$

$(\theta', \theta, \mathbf{y})$. Now, we have

$$\frac{\partial f(\theta, \theta', \mathbf{y})}{\partial(\theta, \theta', \mathbf{y})} = \begin{bmatrix} \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial \theta'} & \frac{\partial f_1}{\partial \mathbf{y}} \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial \theta'} & \frac{\partial f_2}{\partial \mathbf{y}} \\ \frac{\partial f_3}{\partial \theta} & \frac{\partial f_3}{\partial \theta'} & \frac{\partial f_3}{\partial \mathbf{y}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \theta'}{\partial \theta} & \frac{\partial \theta'}{\partial \theta'} & \frac{\partial \theta'}{\partial \mathbf{y}} \\ \frac{\partial \theta}{\partial \theta} & \frac{\partial \theta}{\partial \theta'} & \frac{\partial \theta}{\partial \mathbf{y}} \\ \frac{\partial \mathbf{y}}{\partial \theta} & \frac{\partial \mathbf{y}}{\partial \theta'} & \frac{\partial \mathbf{y}}{\partial \mathbf{y}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and taking determinants

$$\left| \frac{\partial f(\theta, \theta', \mathbf{y})}{\partial(\theta, \theta', \mathbf{y})} \right| = 1 \cdot \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} + 0 + 0 = 1.$$

Finally, with $q(\theta'|\theta)$ denoting the proposal density of the exchange algorithm, define the auxiliary distribution as follows

$$q(u|\theta) = q(\theta'|\theta)p(\mathbf{y}|\theta')$$

where $p(\mathbf{y}|\theta')$ is the likelihood of auxiliary data $\mathbf{y}$ under the assumed model (B.5.1). With these elements, an iMCMC algorithm targeting $p(\theta|\mathbf{x})$ would (i) sample u from $q(u|\theta)$, which amounts to first sampling θ' from $q(\theta'|\theta)$, before drawing $\mathbf{y}$ from $p(\mathbf{y}|\theta')$, and (ii) accept θ' with probability

$$\begin{aligned} \alpha(\theta, \theta') &= \min\left\{1, \frac{p(\theta'|\mathbf{x})p(u'|\theta')}{p(\theta|\mathbf{x})p(u|\theta)} \left|\frac{\partial f(\theta, u)}{\partial(\theta, u)}\right|\right\} \\ &= \min\left\{1, \frac{p(\theta')\gamma(\mathbf{x}|\theta')\gamma(\mathbf{y}|\theta)q(\theta|\theta')}{p(\theta)\gamma(\mathbf{x}|\theta)\gamma(\mathbf{y}|\theta')q(\theta'|\theta)} \left|\frac{\partial f(\theta, \theta', \mathbf{y})}{\partial(\theta, \theta', \mathbf{y})}\right|\right\} \\ &= \min\left\{1, \frac{p(\theta')\gamma(\mathbf{x}|\theta')\gamma(\mathbf{y}|\theta)q(\theta|\theta')}{p(\theta)\gamma(\mathbf{x}|\theta)\gamma(\mathbf{y}|\theta')q(\theta'|\theta)}\right\}, \end{aligned}$$

which is nothing more than the exchange algorithm (Algorithm 10).

B.5.3 Defining the iExchange algorithm

We now define our extension of the exchange algorithm. We will assume that an iMCMC scheme to target $p(\theta|\mathbf{x})$ has been defined, that is, auxiliary variables u, involution $f(\theta, u) = (\theta', u')$ and conditional distribution $q(u|\theta)$ have all been specified. When the posterior is doubly-intractable, in general one will not be able to implement this algorithm due to the intractability of the acceptance probability. However, in spirit of the exchange algorithm, we can choose auxiliary variables and their conditional distribution to induce cancellation of normalising constants in the acceptance probability.

In particular, we let $\tilde{u} = (u, \mathbf{y})$, where $\mathbf{y}$ denotes an auxiliary dataset lying in the same space as $\mathbf{x}$. Now, writing $f(\theta, u) = (f_1(\theta, u), f_2(\theta, u)) = (\theta', u')$ we define an involution $g(\theta, \tilde{u})$ as follows

$$\begin{aligned} g(\theta, \tilde{u}) = g(\theta, (u, \mathbf{y})) &= (f_1(\theta, u), (f_2(\theta, u), \mathbf{y})) \\ &= (\theta', (u', \mathbf{y})) \end{aligned}$$

for which we have

$$\begin{aligned} g(g(\theta, \tilde{u})) &= g(\theta', (u', \mathbf{y})) \\ &= (f_1(\theta', u'), (f_2(\theta', u'), \mathbf{y})) \\ &= (\theta, (u, \mathbf{y})) \\ &= (\theta, \tilde{u}) \end{aligned}$$

that is, g is indeed an involution. Now, as in Section B.5.3, drop the inner parenthesis and write $(\theta, \tilde{u}) = (\theta, u, \mathbf{y})$. The Jacobian is now given by

$$\frac{\partial g(\theta, \tilde{u})}{\partial(\theta, \tilde{u})} = \frac{\partial g(\theta, u, \mathbf{y})}{\partial(\theta, u, \mathbf{y})} = \begin{bmatrix} \frac{\partial g_1}{\partial \theta} & \frac{\partial g_1}{\partial u} & \frac{\partial g_1}{\partial \mathbf{y}} \\ \frac{\partial g_2}{\partial \theta} & \frac{\partial g_2}{\partial u} & \frac{\partial g_2}{\partial \mathbf{y}} \\ \frac{\partial g_3}{\partial \theta} & \frac{\partial g_3}{\partial u} & \frac{\partial g_3}{\partial \mathbf{y}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial \mathbf{y}} \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial u} & \frac{\partial f_2}{\partial \mathbf{y}} \\ \frac{\partial f_3}{\partial \theta} & \frac{\partial f_3}{\partial u} & \frac{\partial f_3}{\partial \mathbf{y}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial u} & 0 \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial u} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and taking determinants we get the following

$$\left|\frac{\partial g(\theta,\tilde{u})}{\partial(\theta,\tilde{u})}\right| = 1 \cdot \left|\begin{bmatrix} \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial u} \end{bmatrix}\right| + 0 + 0 = \left|\frac{\partial f(\theta,u)}{\partial(\theta,u)}\right|.$$

The final element to define is the auxiliary distribution. Given current state θ we consider sampling $\tilde{u} = (u, \mathbf{y})$ as follows: (i) sample u from $q(u|\theta)$, then (ii) sample $\mathbf{y}$ from $p(\mathbf{y}|\theta')$ (the model) where $\theta' = f_1(\theta, u)$. This leads to the following auxiliary density

$$q(\tilde{u}|\theta) = q(u|\theta)p(\mathbf{y}|\theta').$$

All the elements of an iMCMC algorithm have now been defined, a single iteration of which consists of the following. Given current state θ, we first sample $\tilde{u} = (u, \mathbf{y})$ via $q(\tilde{u}|\theta)$ as above. We then invoke involution $g(\theta, \tilde{u}) = (\theta', \tilde{u}') = (\theta', (u', \mathbf{y}))$, generating a proposal θ' which we accept with the following probability

$$\begin{aligned}
\alpha(\theta,\theta') &= \min\left\{1, \frac{p(g(\theta,\tilde{u}))}{p(\theta,\tilde{u})}\left|\frac{\partial g(\theta,\tilde{u})}{\partial(\theta,\tilde{u})}\right|\right\} \\
&= \min\left\{1, \frac{p(\theta'|\mathbf{x})q(\tilde{u}'|\theta')}{p(\theta|\mathbf{x})q(\tilde{u}|\theta)}\left|\frac{\partial g(\theta,\tilde{u})}{\partial(\theta,\tilde{u})}\right|\right\} \\
&= \min\left\{1, \frac{p(\theta'|\mathbf{x})q(u'|\theta')p(\mathbf{y}|\theta)}{p(\theta|\mathbf{x})q(u|\theta)p(\mathbf{y}|\theta')}\left|\frac{\partial f(\theta,u)}{\partial(\theta,u)}\right|\right\} \\
&= \min\left\{1, \frac{p(\theta')\gamma(\mathbf{x}|\theta')\gamma(\mathbf{y}|\theta)q(u'|\theta')}{p(\theta)\gamma(\mathbf{x}|\theta)\gamma(\mathbf{y}|\theta')q(u|\theta)}\left|\frac{\partial f(\theta,u)}{\partial(\theta,u)}\right|\right\},
\end{aligned}$$

where as in the exchange algorithm we observe cancellation of normalising constants thanks to the introduction of auxiliary data. Note the Jacobian term here concerns the involution of the original iMCMC scheme to sample from $p(\theta|\mathbf{x})$, and thus the key difference here is the introduction of auxiliary data. The result is what we call the iExchange algorithm (Algorithm 9).

B.6 Bayesian inference: extra details

In this section we provide extra details concerning our MCMC scheme for the interaction-sequence models outlined in Section 4.3, including explicit specification of proposal distributions, involutions and auxiliary distributions, derivations of closed-form acceptance probabilities and pseudocode.

B.6.1 Dispersion conditional

The dispersion conditional can be obtained directly from (4.3.3) by conditioning on the mode $\mathcal{S}^m$, in particular we have

$$p(\gamma|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n) \propto Z(\mathcal{S}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m)\right\} p(\gamma). \tag{B.6.1}$$

To target (B.6.1) we use the exchange algorithm of Murray et al. (2006) (see Appendix B.5.1 for background details). As a proposal $q(\gamma'|\gamma)$ we consider sampling γ' uniformly over a ε-neighbourhood of γ with reflection at zero, this is, we first sample $\gamma^* \sim \text{Uniform}(\gamma - \varepsilon, \gamma + \varepsilon)$ and then let $\gamma' = \gamma^*$ if $\gamma^* > 0$ and let $\gamma' = -\gamma^*$ otherwise. The density is thus given by the following (for $\gamma > 0$)

$$q(\gamma'|\gamma) = \begin{cases} \frac{1}{2\varepsilon} & \text{if } \gamma' > 0 \text{ and } \gamma + \gamma' > \varepsilon \\ \frac{1}{\varepsilon} & \text{if } \gamma' > 0 \text{ and } \gamma + \gamma' < \varepsilon \\ 0 & \text{if } \gamma' \leq 0. \end{cases} \tag{B.6.2}$$

whilst $q(\gamma'|\gamma) = 0$ for $\gamma \leq 0$. Observe this proposal is symmetric, in that $q(\gamma'|\gamma) = q(\gamma|\gamma)$.

Now, a single iteration consists of the following. Assuming γ is our current state, we first sample proposal γ' from $q(\gamma'|\gamma)$. Next, we sample auxiliary data $\{\mathcal{S}_i^*\}_{i=1}^n$ i.i.d.

from the appropriate model, namely

$$\mathcal{S}_i^* \sim \text{SIS}(\mathcal{S}^m, \gamma') \quad (\text{for } i = 1, \ldots, n),$$

which we note implies

$$p(\{\mathcal{S}_i^*\}_{i=1}^n \,|\, \mathcal{S}^m, \gamma') = Z(\mathcal{S}^m, \gamma')^{-n} \exp\left\{-\gamma' \sum_{i=1}^n d_S(\mathcal{S}_i^*, \mathcal{S}^m)\right\}.$$

Finally, we accept this proposal with the following probability

$$\alpha(\gamma, \gamma') = \min\{1, H(\gamma, \gamma')\}$$

where

$$\begin{aligned} H(\gamma, \gamma') &= \frac{p(\gamma'|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n) p(\{\mathcal{S}_i^*\}_{i=1}^n | \mathcal{S}^m, \gamma)}{p(\gamma|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n) p(\{\mathcal{S}_i^*\}_{i=1}^n | \mathcal{S}^m, \gamma')} \frac{q(\gamma|\gamma')}{q(\gamma'|\gamma)} \\ &= \exp\left\{-(\gamma' - \gamma)\left(\sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) - \sum_{i=1}^n d_S(\mathcal{S}_i^*, \mathcal{S}^m)\right)\right\} \frac{p(\gamma')}{p(\gamma)}, \end{aligned}$$

where we note normalising constants of the (conditional) posterior and auxiliary data cancel one another out, whilst the proposal density terms cancel due to its symmetry.

B.6.2 Mode conditional

By conditioning on γ in (4.3.3) we get the following form for the mode conditional posterior

$$p(\mathcal{S}^m|\gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n) \propto Z(\mathcal{S}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) - \gamma_0 d_S(\mathcal{S}^m, \mathcal{S}_0)\right\},$$

which as outlined in Section 4.3.4 we target via the iExchange algorithm (Algorithm 9). For further details on the iExchange algorithm, including justification as an instance of iMCMC, please see Appendix B.5.

Supposing that auxiliary variables u, involution $f(\mathcal{S}^m, u)$ and auxiliary distribution $q(u|\mathcal{S}^m)$ have all be specified, a single iteration of the iExchange algorithm in this case consists of the following. With γ fixed and $\mathcal{S}^m$ denoting our current state we first sample auxiliary variable u according to $q(u|\mathcal{S}^m)$. We then invoke the involution $f(\mathcal{S}^m, u) = ([\mathcal{S}^m]', u')$, which generates our proposal $[\mathcal{S}^m]'$. Next, we sample auxiliary data $\{\mathcal{S}_i^*\}_{i=1}^n$ i.i.d. where

$$\mathcal{S}_i^* \sim \text{SIS}([\mathcal{S}^m]', \gamma).$$

Finally, we accept $[\mathcal{S}^m]'$ with the following probability

$$\alpha\left(\mathcal{S}^m, [\mathcal{S}^m]'\right) = \min\left\{1, H(\mathcal{S}^m, [\mathcal{S}^m]')\right\}$$

where

$$\begin{aligned} H(\mathcal{S}^m, [\mathcal{S}^m]') &= \frac{p([\mathcal{S}^m]' \,|\, \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n)}{p(\mathcal{S}^m \,|\, \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n)} \frac{p(\{\mathcal{S}_i^*\}_{i=1}^n \,|\, \mathcal{S}^m, \gamma)}{p(\{\mathcal{S}_i^*\}_{i=1}^n \,|\, [\mathcal{S}^m]', \gamma)} \frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)} \\ &= \exp\left\{ -\gamma \left(\sum_{i=1}^n d_S(\mathcal{S}^{(i)}, [\mathcal{S}^m]') - \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) \right) \right. \\ &\quad - \gamma \left(\sum_{i=1}^n d_S(\mathcal{S}_i^*, \mathcal{S}^m) - \sum_{i=1}^n d_S(\mathcal{S}_i^*, [\mathcal{S}^m]') \right) \\ &\quad \left. - \gamma_0 \left(d_S([\mathcal{S}^m]', \mathcal{S}_0) - d_S(\mathcal{S}^m, \mathcal{S}_0) \right) \right\} \frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)} \end{aligned} \tag{B.6.3}$$

where the ratio $q(u' \,|\, [\mathcal{S}^m]')/q(u \,|\, \mathcal{S}^m)$ is move-dependent. Again, we have the normalising constants of the conditional posterior and auxiliary data cancelling one another out.

B.6.3 Edit allocation move

Supposing that $\mathcal{S}^m = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ denotes the current state, recall that for this move we have an auxiliary variable given by

$$u = (\delta, \boldsymbol{z}, u_1, \ldots, u_N)$$

where (i) δ denotes the total number of edits (entry insertions and deletions), (ii) $\boldsymbol{z} = (z_1, \ldots, z_N)$ denotes the allocation of edits to paths, that is, $z_i \in \mathbb{Z}_{\geq 0}$ is the number of edits allocated to the ith path, where $\sum_{i=1}^{N} z_i = \delta$, and (iii) $u_i = (d_i, \boldsymbol{v}_i, \boldsymbol{v}'_i, \boldsymbol{y}_i)$ describes the edits to the ith path, where d_i is the number of deletions, $\boldsymbol{v}_i$ and $\boldsymbol{v}'_i$ are subsequences indexing entry insertions and deletions and $\boldsymbol{y}_i$ denotes entries to be inserted. Given these auxiliary variables and some current state $\mathcal{S}^m$, as outlined in Section 4.3.5, this move has involution

$$f(\mathcal{S}^m, u) = ([\mathcal{S}^m]', u')$$

returning (i) $[\mathcal{S}^m]' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_N)$, denoting the proposed new state and (ii) $u' = (z_i - d_i, \boldsymbol{v}'_i, \boldsymbol{v}_i, (\mathcal{I}_i)_{\boldsymbol{v}_i})$, denoting the auxiliary variables parameterising the reverse move back to $\mathcal{S}^m$.

A key term appearing in the acceptance probability of this move, as seen in (B.6.3), is the following ratio

$$\frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)}$$

where $q(u|\mathcal{S}^m)$ denotes the assumed distribution of auxiliary variables u given current state $\mathcal{S}^m$. Towards deriving this ratio, recall the following assumptions stated in

Section 4.3.5

$$\begin{aligned}
\delta &\sim \text{Uniform}\{1, \dots, \nu_{\text{ed}}\} \\
\boldsymbol{z} \,|\, \delta &\sim \text{Multinomial}(\delta\,;\, 1/N, \dots, 1/N) \\
d_i \,|\, z_i &\sim \text{Uniform}\{0, \dots, \min(z_i, n_i)\} \qquad (\text{for } i = 1, \dots, N),
\end{aligned}$$

whilst it is assumed the indexing subsequences $\boldsymbol{v}_i$ and $\boldsymbol{v}'_i$ are sampled uniformly. Regarding this latter assumption, recall that $\boldsymbol{v}_i$ is a length d_i (number of deletions) subsequence of $[n_i]$ (n_i is the length of $\mathcal{I}_i$), whilst $\boldsymbol{v}'_i$ is a length $a_i := z_i - d_i$ (number of insertions) subsequence of $[m_i]$, where $m_i = n_i - d_i + a_i$ (length of the ith proposed path $\mathcal{I}'_i$). Thus sampling these uniformly implies

$$q(\boldsymbol{v}_i|d_i) = \binom{n_i}{d_i}^{-1} \quad q(\boldsymbol{v}'_i|d_i, z_i) = \binom{m_i}{a_i}^{-1}.$$

Finally, regarding sampling entry insertions we for now assume these are drawn via some general distribution which may be dependent on the current state, namely we assume each $\boldsymbol{y}_i$ was drawn via $q(\boldsymbol{y}|\mathcal{I}_i)$. Together this implies the following closed form for the auxiliary distribution

$$\begin{aligned}
q(u|\mathcal{S}^m) &= q(\delta)q(\boldsymbol{z}|\delta)\prod_{i=1}^{N} q(d_i)q(\boldsymbol{v}_i|d_i)q(\boldsymbol{v}'_i|d_i, z_i)q(\boldsymbol{y}_i|\mathcal{I}_i) \\
&= \frac{1}{\nu_{\text{ed}}}\left(\frac{1}{N}\right)^{\delta}\prod_{i=1}^{N}\frac{1}{\min(n_i, z_i)+1}\binom{n_i}{d_i}^{-1}\binom{m_i}{a_i}^{-1} q(\boldsymbol{y}_i|\mathcal{I}_i).
\end{aligned} \tag{B.6.4}$$

whilst if $([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)$ has been obtained by the involution of this move we have

$$\begin{aligned}
q(u'|[\mathcal{S}^m]') &= q(\delta)q(\boldsymbol{z}|\delta)\prod_{i=1}^{N} q(a_i)q(\boldsymbol{v}'_i|a_i)q(\boldsymbol{v}_i|a_i, z_i)q((\mathcal{I}_i)_{\boldsymbol{v}_i}|\mathcal{I}'_i) \\
&= \frac{1}{\nu_{\text{ed}}}\left(\frac{1}{N}\right)^{\delta}\prod_{i=1}^{N}\frac{1}{\min(m_i, z_i)+1}\binom{m_i}{a_i}^{-1}\binom{n_i}{d_i}^{-1} q((\mathcal{I}_i)_{\boldsymbol{v}_i}|\mathcal{I}'_i),
\end{aligned} \tag{B.6.5}$$

and thus the ratio of (B.6.5) and (B.6.4) is given by the following

$$\frac{q(u'|[\mathcal{S}^m]')}{q(u|\mathcal{S}^m)} = \prod_{i=1}^{N} \frac{\min(n_i, z_i) + 1}{\min(m_i, z_i) + 1} \frac{q((\mathcal{I}_i)_{\boldsymbol{v}_i}|\mathcal{I}_i')}{q(\boldsymbol{y}_i|\mathcal{I}_i)}. \tag{B.6.6}$$

We finalise these details on the edit allocation move with a discussion on entry insertion distributions. The simplest option here is to sample entries uniformly over the vertex set $\mathcal{V}$. In this case, with $V = |\mathcal{V}|$, we have

$$q(\boldsymbol{y}_i|\mathcal{I}_i) = \left(\frac{1}{V}\right)^{a_i} \tag{B.6.7}$$

which implies

$$\frac{q((\mathcal{I}_i)_{\boldsymbol{v}_i}|\mathcal{I}_i')}{q(\boldsymbol{y}_i|\mathcal{I}_i)} = \left(\frac{1}{V}\right)^{d_i - a_i} = \left(\frac{1}{V}\right)^{2d_i - z_i} = \left(\frac{1}{V}\right)^{n_i - m_i}$$

any of which can be plugged into (B.6.6).

As an alternative choice, one can consider informing the entry insertions from observed data. This approach is based on the following assumption: If two vertices have been observed in the same path across many observations then the probability of proposing one given the other is already present should be higher within the MCMC algorithm.

To reflect this assumption in a proposal, we first extract the necessary information from the observed data. Letting

$$\mathcal{S}^{(1)}, \ldots, \mathcal{S}^{(n)}$$

denote the observed sample we construct a *co-occurrence matrix* $\mathbf{A} \in \mathbb{Z}_{\geq 0}^{V \times V}$, defined

as follows

$$\begin{aligned}\mathbf{A}_{vv'} &= \#\text{observations with an interaction containing both } v \text{ and } v' \\ &= |\{k : \exists\, \mathcal{I} \in \mathcal{S}^{(k)} \text{ with } v, v' \in \mathcal{I}\}|\end{aligned}$$

where $v \neq v'$, whilst for $v = v'$ we let

$$\begin{aligned}\mathbf{A}_{vv} &= \#\text{observations with an interaction containing } v \text{ at least twice} \\ &= |\{k : \exists\, \mathcal{I} \in \mathcal{S}^{(k)} \text{ with } v \in \mathcal{I} \text{ at least twice}\}|,\end{aligned}$$

which can be seen as the adjacency matrix of a weighted graph describing the co-occurrence structure observed in the data. Now, given $\mathbf{A}$ we construct a probability matrix $\mathbf{P} \in \mathbb{R}^{V \times V}$ by normalising the rows, that is

$$\mathbf{P}_{vv'} = \mathbf{A}_{vv'}/Z_v$$

where $Z_v = \sum_{v' \in \mathcal{V}} \mathbf{A}_{vv'}$ is the normalising constant of the vth row. Intuitively, the entry $\mathbf{P}_{vv'}$ can be seen as the probability of observing v' in an interaction given v is known to already be present. We consider using $\mathbf{P}$ to inform entry insertions as follows. Suppose that $\mathcal{I}_i = (x_{i1}, \ldots, x_{in_i})$ denotes the path being edited, with $\boldsymbol{v}_i$ denoting the subsequence of $[n_i]$ indexing which entries are to be deleted. Introduce the notation $\boldsymbol{v}_i^c$ for the complement of $\boldsymbol{v}_i$, which is the subsequence of $[n_i]$ containing the entries not in $\boldsymbol{v}_i$. For example, with $\boldsymbol{v} = (1, 2, 5) \in [5]$ we would have $\boldsymbol{v}^c = (3, 4)$. Now, observed that $(\mathcal{I}_i)_{\boldsymbol{v}_i^c}$ denotes the entries of $\mathcal{I}_i$ *not* being deleted, that is, those being preserved. Our approach is to now propose entries which have often been observed in the data alongside those being preserved. Since each unique preserved entry has an associated distribution over $\mathcal{V}$ given by the respective row of $\mathbf{P}$, we can consider mixing these distributions together with equal weight to form an entry proposal distribution. In particular, we sample entry insertions for the ith path i.i.d. via

the following

$$q(y|\mathcal{I}_i) \propto \sum_{v \in (\mathcal{I}_i)_{\boldsymbol{v}_i^c}} \mathbf{P}_{vy}.$$

One can also introduce a tuning parameter to control the extent to which proposals are informed by the data. In particular, with $\alpha > 0$ first alter the probability matrix as follows

$$\mathbf{P}^{\alpha}_{vv'} \propto \mathbf{P}_{vv'} + \alpha$$

which normalises to

$$\mathbf{P}^{\alpha}_{vv'} = \frac{\mathbf{P}_{vv'} + \alpha}{1 + V\alpha},$$

for which $\mathbf{P}^{\alpha}_{vv'} \to 1/V$ as $\alpha \to \infty$, that is, the rows converge to the uniform distribution over $\mathcal{V}$. We can now define an analogous insertion distribution

$$q_\alpha(y|\mathcal{I}_i) \propto \sum_{v \in (\mathcal{I}_i)_{\boldsymbol{v}_i^c}} \mathbf{P}^{\alpha}_{vy}$$

where as $\alpha \to \infty$ this will converge to a mixture of uniform distributions over $\mathcal{V}$, that is, also a uniform distribution. In this way, one has a proposal which is informed by the data, but becomes less informed as the tuning parameter $\alpha \to \infty$.

We finish with a note regarding evaluation of (B.6.6) for this informed proposal. Supposing that $\mathcal{I}'_i$ is ith path in the proposal $[\mathcal{S}^m]'$ (obtained by deleting d_i entries of $\mathcal{I}_i$ indexed by $\boldsymbol{v}_i$, and inserting entries $\boldsymbol{y}_i$ at locations indexed by $\boldsymbol{v}'_i$), then observe we have $(\mathcal{I}_i)_{\boldsymbol{v}_i^c} = (\mathcal{I}'_i)_{(\boldsymbol{v}'_i)^c}$ (preserved entries) which thus implies $q_\alpha(y|\mathcal{I}_i) = q_\alpha(y|\mathcal{I}'_i)$. Consequently we can write the following

$$\frac{q_\alpha((\mathcal{I}_i)_{\boldsymbol{v}_i}|\mathcal{I}'_i)}{q_\alpha(\boldsymbol{y}_i|\mathcal{I}_i)} = \frac{q_\alpha((\mathcal{I}_i)_{\boldsymbol{v}_i}|\mathcal{I}_i)}{q_\alpha(\boldsymbol{y}_i|\mathcal{I}_i)}$$

and hence only the single mixed distribution $q_\alpha(y|\mathcal{I}_i)$ needs to be constructed. This is helpful to bare in mind when evaluating (B.6.6).

B.6.4 Path insertion and deletion move

Supposing that $\mathcal{S}^m = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ denotes the current state, recall that for this move we have an auxiliary variable given by

$$u = (\varepsilon, d, \boldsymbol{v}, \boldsymbol{v}', \mathcal{I}_1^*, \ldots, \mathcal{I}_a^*)$$

where (i) ε denotes the total number of paths to be inserted or deleted, (ii) d denotes the number of paths to be deleted, implying $a = \varepsilon - d$ insertions, (iii) $\boldsymbol{v}$ and $\boldsymbol{v}'$ denote subsequences indexing path deletions and insertions respectively, and (iv) $(\mathcal{I}_1^*, \ldots, \mathcal{I}_a^*)$ denote the paths to be inserted. Given these auxiliary variables and some current state $\mathcal{S}^m$, as outlined in Section 4.3.5, this move has involution

$$f(\mathcal{S}^m, u) = ([\mathcal{S}^m]', u')$$

returning (i) $[\mathcal{S}^m]' = (\mathcal{I}_1', \ldots, \mathcal{I}_M')$, denoting the proposed new state and (ii) $u' = (\varepsilon, \varepsilon - d, \boldsymbol{v}', \boldsymbol{v}, \mathcal{I}_{v_1}, \ldots, \mathcal{I}_{v_d})$, denoting the auxiliary variables parameterising the reverse move back to $\mathcal{S}^m$.

As seen in the acceptance probability (B.6.3), a key move-dependent term is the following ratio

$$\frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)}$$

where $q(u|\mathcal{S}^m)$ denotes the assumed distribution of auxiliary variables u given current state $\mathcal{S}^m$. Towards deriving this ratio for this move, recall the following assumptions stated in Section 4.3.5

$$\varepsilon \sim \text{Uniform}\{1, \ldots, \nu_{\text{td}}\}$$
$$d \,|\, \varepsilon \sim \text{Uniform}\{0, \ldots, \min(N, \varepsilon)\}$$

whilst we sample indexing subsequences $\boldsymbol{v}$ and $\boldsymbol{v}'$ uniformly and assume path inser-

tions are drawn via some general distribution $q(\mathcal{I}|\mathcal{S}^m)$. In this instance, recall that $\boldsymbol{v}$ is a subsequence of $[N]$ of size d, whilst $\boldsymbol{v}'$ is a subsequence of $[M]$ of size a, where $M = N - d + a$ is the length of $[\mathcal{S}^m]'$. Sampling these uniformly thus implies

$$q(\boldsymbol{v}|d) = \binom{N}{d}^{-1} \quad q(\boldsymbol{v}'|\varepsilon, d) = \binom{M}{a}^{-1}$$

leading to the following closed form

$$\begin{aligned} q(u|\mathcal{S}^m) &= q(\varepsilon)q(d|\varepsilon)q(\boldsymbol{v}|d)q(\boldsymbol{v}'|\varepsilon, d)\prod_{i=1}^{a} q(\mathcal{I}_i^*|\mathcal{S}^m) \\ &= \frac{1}{\nu_{\text{td}}}\frac{1}{\min(N,\varepsilon)+1}\binom{N}{d}^{-1}\binom{M}{a}^{-1}\prod_{i=1}^{a} q(\mathcal{I}_i^*|\mathcal{S}^m) \end{aligned}$$

whilst, if $([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)$ has been obtained by the involution above, we have

$$\begin{aligned} q(u'|[\mathcal{S}^m]') &= q(\varepsilon)q(a|\varepsilon)q(\boldsymbol{v}'|a)q(\boldsymbol{v}|\varepsilon, a)\prod_{i=1}^{d} q(\mathcal{I}_{v_i}|[\mathcal{S}^m]') \\ &= \frac{1}{\nu_{\text{td}}}\frac{1}{\min(M,\varepsilon)+1}\binom{M}{a}^{-1}\binom{N}{d}^{-1}\prod_{i=1}^{d} q(\mathcal{I}_{v_i}|[\mathcal{S}^m]'). \end{aligned}$$

Taking the ratio of these leads to the following

$$\frac{q(u'|[\mathcal{S}^m]')}{q(u|\mathcal{S}^m)} = \frac{\min(N,\varepsilon)+1}{\min(M,\varepsilon)+1}\frac{\prod_{i=1}^{d} q(\mathcal{I}_{v_i}|[\mathcal{S}^m]')}{\prod_{i=1}^{a} q(\mathcal{I}'_{v'_i}|\mathcal{S}^m)}, \tag{B.6.8}$$

which can be substituted into (B.6.3) to evaluate the acceptance probability of this move (here we again use the fact $\mathcal{I}'_{v'_i} = \mathcal{I}_i^*$).

We finalise by discussing possible choices for the path insertion distribution. The simplest approach is to combine a distribution on path length with uniform sampling of entries. In particular, to sample some path $\mathcal{I} = (x_1, \ldots, x_m)$ we (i) sample its length

m via some distribution $q(m)$ (ii) sample entries x_i uniformly from $\mathcal{V}$. This implies

$$q(\mathcal{I}|\mathcal{S}^m) = q(\mathcal{I}) = q(m)\left(\frac{1}{V}\right)^m$$

where $V = |\mathcal{V}|$, which can be substituted into (B.6.8).

One can also consider informing entry insertions from observed data. With

$$\mathcal{S}^{(1)}, \ldots, \mathcal{S}^{(n)}$$

a sample, for each $v \in \mathcal{V}$ we let

$$c_v = |\{k : \exists \mathcal{I} \in \mathcal{S}^{(k)} \text{ with } v \in \mathcal{I}\}|$$

denote the number of observations with at least one path containing the vertex v. Normalising this leads to

$$p_v = \frac{c_v}{\sum v \in \mathcal{V} c_v}$$

which can be seen as the probability a randomly selected observation contains v. Introducing the parameter $\alpha > 0$ we let

$$q_\alpha(v) \propto p_v + \alpha$$

which normalises to

$$q_\alpha(v) = \frac{p_v + \alpha}{1 + \alpha V}.$$

One can now use this to sample path entries, namely to sample $\mathcal{I} = (x_1, \ldots, x_m)$ we (i) sample length m via some $q(m)$, (ii) sample entries x_i via $q_\alpha(x_i)$. Observe that if $\alpha = 0$ we have $q_\alpha(v) = p_v$, and the entry insertion distribution is fully informed by the data, whilst as $\alpha \to \infty$ we have $q_\alpha(v) \to 1/V$, and we recover uniform entry insertions.

B.6.5 Model sampling

In this section we provide supporting details regarding our iMCMC algorithm to sample from the SIS models outlined in Section 4.3.6. Recall that for the SIS model (Definition 4.2.1) the (normalised) probability of observing $\mathcal{S}$ is given by

$$p(\mathcal{S}|\mathcal{S}^m, \gamma) = \frac{\exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\}}{Z(\mathcal{S}^m, \gamma)},$$

implying the following closed form for the acceptance probability (4.3.8)

$$\alpha(\mathcal{S}, \mathcal{S}') = \min\{1, H(\mathcal{S}, \mathcal{S}')\} \tag{B.6.9}$$

where

$$\begin{aligned} H(\mathcal{S}, \mathcal{S}') &= \frac{p(\mathcal{S}'|\mathcal{S}^m, \gamma)}{p(\mathcal{S}|\mathcal{S}^m, \gamma)} \frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} \\ &= \exp\left\{-\gamma\left(d_S(\mathcal{S}', \mathcal{S}^m) - d_S(\mathcal{S}, \mathcal{S}^m)\right)\right\} \frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})}, \end{aligned}$$

where the value of $q(u'|\mathcal{S}')/q(u|\mathcal{S})$ will depend on the iMCMC specification.

As mentioned in Section 4.3.6, we consider re-using the iMCMC moves of our iExchange scheme used to sample from the mode conditional (Appendices B.6.3 and B.6.4). For ease of reference, we summarise the corresponding ratios for each move:

- **Edit allocation** - suppose that u, $f(u, \mathcal{S})$ and $q(u|\mathcal{S})$ are defined as in Appendix B.6.3 (replacing $\mathcal{S}^m$ with $\mathcal{S}$ and $[\mathcal{S}^m]'$ with $\mathcal{S}'$) with a uniform entry insertion distribution (B.6.7). With $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ the current state, supposing $u = (\delta, \boldsymbol{z}, u_1, \ldots, u_N)$ has been sampled via $q(u|\mathcal{S})$ mapping to $(\mathcal{S}', u') = f(\mathcal{S}, u)$ we will have

$$\frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} = \prod_{i=1}^{N} \frac{\min(n_i, z_i) + 1}{\min(m_i, z_i) + 1} \left(\frac{1}{V}\right)^{n_i - m_i} \tag{B.6.10}$$

  where n_i and m_i denote the lengths of the ith path in $\mathcal{S}$ and $\mathcal{S}'$ respectively;

- **Path insertion and deletion** - suppose that u, $f(u, \mathcal{S})$ and $q(u|\mathcal{S})$ are defined

as in Appendix B.6.4 (again using $\mathcal{S}$ and $\mathcal{S}'$ instead of $\mathcal{S}^m$ and $[\mathcal{S}^m]'$). With $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ the current state, supposing $u = (\varepsilon, d, \boldsymbol{v}, \boldsymbol{v}', \mathcal{I}_1^*, \ldots, \mathcal{I}_a^*)$ (where $a = \varepsilon - d$) has been sampled via $q(u|\mathcal{S})$ mapping to $(\mathcal{S}', u') = f(\mathcal{S}, u)$ with $\mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_M)$ we will have

$$\frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} = \frac{\min(N, \varepsilon) + 1}{\min(M, \varepsilon) + 1} \frac{\prod_{i=1}^{d} q(\mathcal{I}_{v_i}|\mathcal{S}')}{\prod_{i=1}^{a} q(\mathcal{I}'_{v'_i}|\mathcal{S})}. \tag{B.6.11}$$

As mentioned in Section 4.3.6, we follow the approach used for the posterior mode conditional and consider mixing together these two iMCMC moves with some proportion $\beta \in (0, 1)$, left as a tuning parameter.

B.7 Bayesian inference for multiset models

Here we detail the approach to inference for the interaction-multiset models (Definition 4.2.2). This is very similar to the interaction-sequence models outlined in Section 4.3, with priors, hierarchical model and posterior are all being essentially the same (albeit with different notation). Computationally, we again use MCMC to sample from the posterior, adapting the scheme proposed for the interaction-sequence models.

B.7.1 Priors, hierarchical model and posterior

To specify priors, we follow Section 4.3.1 and assume the mode was itself sampled from an SIM model, namely

$$\mathcal{E}^m \sim \text{SIM}(\mathcal{E}_0, \gamma_0)$$

where $(\mathcal{E}_0, \gamma_0)$ are hyperparameters, whilst we assume the dispersion was drawn from some distribution $p(\gamma)$ whose support is a subset of the non-negative reals. Given

these specifications, an observed sample $\{\mathcal{E}^{(i)}\}_{i=1}^n$ is assumed to be drawn via

$$\begin{aligned}\mathcal{E}^{(i)} \,|\, \mathcal{E}^m, \gamma &\sim \text{SIM}(\mathcal{E}^m, \gamma) \quad (\text{for } i = 1, \ldots, n)\\ \mathcal{E}^m &\sim \text{SIM}(\mathcal{E}_0, \gamma_0)\\ \gamma &\sim p(\gamma).\end{aligned}$$

The likelihood of $\{\mathcal{E}^{(i)}\}_{i=1}^n$ is given by

$$\begin{aligned}p(\{\mathcal{E}^{(i)}\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma) &= \prod_{i=1}^n p(\mathcal{E}^{(i)} \,|\, \mathcal{E}^m, \gamma)\\ &= Z(\mathcal{E}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m)\right\}\end{aligned}$$

which implies a posterior given by

$$\begin{aligned}p(\mathcal{E}^m, \gamma \,|\, \{\mathcal{E}^{(i)}\}_{i=1}^n) &\propto p(\{\mathcal{E}^{(i)}\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma) p(\mathcal{E}^m) p(\gamma)\\ &= Z(\mathcal{E}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m)\right\} \qquad \text{(B.7.1)}\\ &\quad \exp\{-\gamma_0 d_E(\mathcal{E}^m, \mathcal{E}_0)\} p(\gamma).\end{aligned}$$

B.7.2 Posterior sampling

As for the interaction-sequence models, we consider sampling from the posterior (B.7.1) via component-wise MCMC algorithm, alternating between sampling from the two conditionals

$$p(\mathcal{E}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n) \quad \text{and} \quad p(\gamma \,|\, \mathcal{E}^m, \{\mathcal{E}^{(i)}\}_{i=1}^n)$$

in both of which the normalising constant of (B.7.1) will persist, making them doubly-intractable (Murray et al., 2006; Møller et al., 2006) and motivating the use of the exchange and iExchange algorithms.

There are two key differences here compared with the setting of Section 4.3. Firstly, the mode in this instance is a multiset, implying the mode conditional is a distribution over multisets rather than sequences. Secondly, to induce the required cancellation of normalising constants, sampling of auxiliary data in the exchange (or iExchange) algorithms must be from the multiset models.

In both cases, the challenge lies in sampling from distributions over multisets (of paths). As will be seen in subsequent sections, a solution can be found by first extending these to distributions over sequences, before using the iMCMC-based algorithms proposed for the interaction-sequence models (Section 4.3 and appendix B.6) to target them.

B.7.3 Dispersion conditional

Conditioning on $\mathcal{E}^m$ in (B.7.1) we have the following

$$p(\gamma \,|\, \mathcal{E}^m, \{\mathcal{E}^{(i)}\}_{i=1}^n) \propto Z(\mathcal{E}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m)\right\} p(\gamma)$$

which to target we follow Section 4.3.3 and appendix B.6.1 and use the exchange algorithm (Murray et al., 2006). For the proposal $q(\gamma'|\gamma)$ we again consider sampling γ' uniformly over a ε-neighbourhood of γ with reflection at zero (see Appendix B.6.1). With this choice of proposal, a single iteration consists of the following. Assuming γ is the current state, we first sample proposal γ' via $q(\gamma'|\gamma)$. Next, we sample auxiliary data $\{\mathcal{E}_i^*\}_{i=1}^n$ i.i.d. from the appropriate multiset model, namely

$$\mathcal{E}_i^* \sim \text{SIM}(\mathcal{E}^m, \gamma') \quad (\text{for } i = 1, \ldots, n),$$

for which we have

$$p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma') = Z(\mathcal{E}^m, \gamma')^{-n} \exp\left\{-\gamma' \sum_{i=1}^n d_E(\mathcal{E}_i^*, \mathcal{E}^m)\right\}.$$

Finally, we accept this proposal with the following probability

$$\alpha(\gamma, \gamma') = \min\{1, H(\gamma, \gamma')\} \tag{B.7.2}$$

where

$$\begin{aligned} H(\gamma, \gamma') &= \frac{p(\gamma' \,|\, \mathcal{E}^m, \{\mathcal{E}^{(i)}\}_{i=1}^n) p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma)}{p(\gamma \,|\, \mathcal{E}^m, \{\mathcal{E}^{(i)}\}_{i=1}^n) p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma')} \frac{q(\gamma|\gamma')}{q(\gamma'|\gamma)} \\ &= \exp\left\{-(\gamma' - \gamma)\left(\sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m) - \sum_{i=1}^n d_E(\mathcal{E}_i^*, \mathcal{E}^m)\right)\right\} \frac{p(\gamma')}{p(\gamma)}, \end{aligned}$$

where, as in Appendix B.6.1, normalising constants of the (conditional) posterior and auxiliary data cancel one another out, whilst the proposal density terms cancel due to its symmetry.

B.7.4 Mode conditional

Conditioning on γ in (B.7.1) we have the following

$$p(\mathcal{E}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n) \propto Z(\mathcal{E}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m) - \gamma_0 d_E(\mathcal{E}^m, \mathcal{E}_0)\right\}, \tag{B.7.3}$$

which is a distribution over $\boldsymbol{\mathcal{E}}^*$, that is, the space of multisets. To re-use the iExchange scheme of Section 4.3.4 we instead need a distribution over the space of interaction sequences $\boldsymbol{\mathcal{S}}^*$. To this end, we extend (B.7.3) to a distribution over interaction sequences.

Consider the general problem of extending some distribution $\pi(\mathcal{E})$ over $\boldsymbol{\mathcal{E}}^*$ to one over $\boldsymbol{\mathcal{S}}^*$. Firstly, observe each $\mathcal{E}$ is associated with a set of sequences, obtained by

placing the interactions of $\mathcal{E}$ in different orders. More formally, $\mathcal{E}$ can be seen as equivalence class of sequences (see Appendix B.1). As such, one can consider assigning equal probability to each unique ordering of $\mathcal{E}$. In particular, for $\mathcal{S} \in \boldsymbol{\mathcal{S}}^*$ we let

$$\tilde{\pi}(\mathcal{S}) = \frac{1}{A(\mathcal{E})}\pi(\mathcal{E})$$

where $\mathcal{E}$ is the multiset obtained from $\mathcal{S}$ by disregarding the order of interactions, and $A(\mathcal{E})$ denotes the number of unique orderings of the paths in $\mathcal{E}$.

The form of $A(\mathcal{E})$ can be obtained as follows. Suppose that $\mathcal{E}$ consists of N paths, with $\tilde{N} \leq N$ *unique* paths. Without loss of generality label the unique paths 1 to $\tilde{N}$ and let w_i denote the multiplicity of the ith path. Now, if each path of $\mathcal{E}$ is different there are $N!$ possible ways to order them. However, if there are repeated paths this will include double counting. Therefore, in general we must further divide by $(w_i)!$ leading to the familiar multinomial term

$$A(\mathcal{E}) := \binom{N}{w_1, \ldots, w_{\tilde{N}}} = \frac{N!}{w_1! \cdots w_{\tilde{N}}!}. \tag{B.7.4}$$

Through this reasoning we can extend (B.7.3) as follows

$$\tilde{p}(\mathcal{S}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n) = \frac{1}{A(\mathcal{E}^m)} p(\mathcal{E}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n) \tag{B.7.5}$$

where now $\mathcal{S}^m \in \boldsymbol{\mathcal{S}}^*$ and $\mathcal{E}^m$ is the multiset obtained from $\mathcal{S}^m$ by disregarding the order of paths.

We can now reuse the iExchange algorithm of Section 4.3.4 and appendix B.6.2 to target (B.7.5). However, note the normalising constant appearing in (B.7.3), and hence also in (B.7.5), is that of an SIM model. Thus, for the iExchange algorithm to induce the necessary cancellation auxiliary data must be sampled from an SIM model.

A single iteration of the resultant algorithm consists of the following. Suppose that $\mathcal{E}^m$ denotes our current state and γ is fixed. We first construct an interaction sequence $\mathcal{S}^m$ by placing the interactions of $\mathcal{E}^m$ in an arbitrary order. Now, assuming u, $q(u|\mathcal{S}^m)$ and $f(\mathcal{S}^m, u)$ is some iMCMC specification as used in Section 4.3, we sample auxiliary variables u via $q(u|\mathcal{S}^m)$, before invoking the involution to obtain $([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)$, where $[\mathcal{S}^m]'$ denotes our proposal. By now disregarding the order of interactions in $[\mathcal{S}^m]'$, we obtain a proposal $[\mathcal{E}^m]'$. We then sample auxiliary data $\{\mathcal{E}_i^*\}_{i=1}^n$ i.i.d. where

$$\mathcal{E}_i^* \sim \text{SIM}([\mathcal{E}^m]', \gamma)$$

which implies

$$p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, [\mathcal{E}^m]', \gamma) = Z([\mathcal{E}^m]', \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_E(\mathcal{E}_i^*, [\mathcal{E}^m]')\right\},$$

before accepting $[\mathcal{E}^m]'$ with the following probability

$$\alpha(\mathcal{E}^m, [\mathcal{E}^m]') = \min\{1, H(\mathcal{E}^m, [\mathcal{E}^m]')\} \tag{B.7.6}$$

where

$$\begin{aligned}
H(\mathcal{E}^m, [\mathcal{E}^m]') &= \frac{\tilde{p}([\mathcal{S}^m]' \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)}{\tilde{p}(\mathcal{S}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)} \frac{p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma)}{p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, [\mathcal{E}^m]', \gamma)} \frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)} \\
&= \frac{\frac{1}{A([\mathcal{E}^m]')} p([\mathcal{E}^m]' \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)}{\frac{1}{A(\mathcal{E}^m)} p(\mathcal{E}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)} \frac{p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma)}{p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, [\mathcal{E}^m]', \gamma)} \frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)} \\
&= \frac{A(\mathcal{E}^m)}{A([\mathcal{E}^m]')} \exp\left\{ -\gamma \left(\sum_{i=1}^n d_E(\mathcal{E}^{(i)}, [\mathcal{E}^m]') - \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m) \right) \right. \\
&\quad - \gamma \left(\sum_{i=1}^n d_E(\mathcal{E}_i^*, \mathcal{E}^m) - \sum_{i=1}^n d_E(\mathcal{E}_i^*, [\mathcal{E}^m]') \right) \\
&\quad \left. - \gamma_0 \left(d_E([\mathcal{E}^m]', \mathcal{E}_0) - d_E(\mathcal{E}^m, \mathcal{E}_0) \right) \right\} \frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)}
\end{aligned} \tag{B.7.7}$$

where here $\mathcal{S}^m$ and $[\mathcal{S}^m]'$ correspond to those used above to generate the proposal $[\mathcal{E}^m]'$. Again, we observe cancellation of normalising constants due to the introduction of auxiliary data. We also see the introduction of a combinatorial term, namely

$$\frac{A(\mathcal{E}^m)}{A([\mathcal{E}^m]')} = \frac{N!(w_1'! \cdots w_{\tilde{M}}'!)}{M!(w_1! \cdots w_{\tilde{N}}!)} \tag{B.7.8}$$

where N and M are the cardinalities of $\mathcal{E}^m$ and $[\mathcal{E}^m]'$ with $\tilde{N}$ and $\tilde{M}$ unique paths, respectively, where w_i is the multiplicity of the ith unique path in $\mathcal{E}^m$ and w_i' is the multiplicity of the ith unique path in $[\mathcal{E}^m]'$.

Clearly, this all depends on a particular iMCMC specification (auxiliary variables, involution and auxiliary distribution). For this we can use the edit allocation (Section 4.3.5 and appendix B.6.3) and interaction insertion and deletion (Appendix B.6.4) moves, which we again mix together with proportion $\beta \in (0, 1)$, left as a tuning parameter. A pseudocode summary of the resulting algorithm to update the mode can be seen in Algorithm 18.

One pragmatic note to be made here is that computationally it is often easier to work with sequences than multisets, since the former can be stored as a vector. To this end, one can store observations as sequences of paths but interpret them as multisets of paths. Furthermore, we can take the order in which they are stored as the 'arbitrary order' referred to in Algorithm 18, and in this way the whole algorithm can be enacted on vectors of paths, simply interpreting the output samples as multisets of paths.

B.7.5 Model sampling

The exchange-based algorithms to update $\mathcal{E}^m$ and γ both require exact sampling of auxiliary data from the SIM models. As for the interaction-sequence models (Section 4.3.6), this is not possible in general. As such, we replace this with approximate samples obtained via an MCMC algorithm.

Towards proposing a suitable MCMC algorithm, we follow the reasoning of Appendix B.7.4 and extend the target distribution (over multisets of paths) to one over sequences of paths, before appealing to the iMCMC scheme proposed to sample from the SIS models (Section 4.3.6 and appendix B.6.5). Recalling that for the SIM model (Definition 4.2.2) the (normalised) probability of observing $\mathcal{E} \in \boldsymbol{\mathcal{E}}^*$ is given by

$$p(\mathcal{E}|\mathcal{E}^m, \gamma) = \frac{1}{Z(\mathcal{E}^m, \gamma)} \exp\{-\gamma d_E(\mathcal{E}, \mathcal{E}^m)\},$$

we can assign any $\mathcal{S} \in \boldsymbol{\mathcal{S}}^*$ the following probability

$$\tilde{p}(\mathcal{S}|\mathcal{E}^m, \gamma) = \frac{1}{A(\mathcal{E})} p(\mathcal{E}|\mathcal{E}^m, \gamma) \tag{B.7.9}$$

where $\mathcal{E}$ is multiset obtain from $\mathcal{S}$ by disregarding order, and $A(\mathcal{E})$ is as defined in (B.7.4), thus defining an extended distribution over $\boldsymbol{\mathcal{S}}^*$.

We can now target (B.7.9) via iMCMC as in Section 4.3.6. In particular, suppose that one would like to sample from an SIM$(\mathcal{E}^m, \gamma)$ model. With u, $q(u|\mathcal{S})$ and $f(\mathcal{S}, u)$ some iMCMC specification as used therein, and $\mathcal{E}$ the current state, a single iteration of will consist of the following

1. Construct interaction sequence $\mathcal{S}$ by placing the paths of $\mathcal{E}$ in an arbitrary order
2. Sample $u \sim q(u|\mathcal{S})$
3. Invoke involution $f(\mathcal{S}, u) = (\mathcal{S}', u')$
4. Disregard order in $\mathcal{S}'$ to obtain proposed multiset $\mathcal{E}'$
5. Evaluate the following probability

$$\alpha(\mathcal{E}, \mathcal{E}') = \min\left\{1, \frac{\tilde{p}(\mathcal{S}'|\mathcal{E}^m, \gamma)}{\tilde{p}(\mathcal{S}|\mathcal{E}^m, \gamma)} \frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})}\right\} \tag{B.7.10}$$

6. Move to state $\mathcal{E}'$ with probability $\alpha(\mathcal{E}, \mathcal{E}')$, staying at $\mathcal{E}$ otherwise.

Clearly, this is conditional upon the choice of iMCMC specification. Here, we follow Section 4.3.6 and recycle the edit allocation (Section 4.3.5 and Appendix B.6.3) and path insertion/deletion moves (Section 4.3.5 and Appendix B.6.4), again mixing them together with proportion $\beta \in (0, 1)$, left as a tuning parameter.

A closed form for (B.7.10) can be derived as follows. Writing $\alpha(\mathcal{E}, \mathcal{E}') = \min\{1, H(\mathcal{E}, \mathcal{E}')\}$ we have

$$\begin{aligned} H(\mathcal{E}, \mathcal{E}') &= \frac{\tilde{p}(\mathcal{S}'|\mathcal{E}^m, \gamma)}{\tilde{p}(\mathcal{S}|\mathcal{E}^m, \gamma)} \frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} \\ &= \frac{\frac{1}{A(\mathcal{E}')} p(\mathcal{E}'|\mathcal{E}^m, \gamma)}{\frac{1}{A(\mathcal{E})} p(\mathcal{E}|\mathcal{E}^m, \gamma)} \frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} \\ &= \frac{A(\mathcal{E})}{A(\mathcal{E}')} \exp\left\{-\gamma\Big(d_E(\mathcal{E}', \mathcal{E}^m) - d_E(\mathcal{E}, \mathcal{E}^m)\Big)\right\} \frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} \end{aligned}$$

where

$$\frac{A(\mathcal{E}^m)}{A([\mathcal{E}^m]')} = \frac{N!(w_1'! \cdots w_{\tilde{M}}'!)}{M!(w_1! \cdots w_{\tilde{N}}!)}$$

with N and M the cardinalities of $\mathcal{E}$ and $\mathcal{E}'$ with $\tilde{N}$ and $\tilde{M}$ unique paths, respectively, where w_i the multiplicity of the ith unique path in $\mathcal{E}$ and w_i' the multiplicity of the ith unique path in $\mathcal{E}'$. As when sampling from the interaction-sequence models (Appendix B.6.5), the ratio $q(u'|\mathcal{S}')/q(u|\mathcal{S})$ will be move dependent and identical to those appearing in Appendix B.6.5, namely (B.6.10) for the edit allocation move and (B.6.11) for the path insertion/deletion move. The whole procedure to sample from the SIM models is summarised in the pseudocode of Algorithm 19.

Finally we note that, as for the interaction-sequence models, by using approximate as opposed to exact sampling in the exchange-based algorithms of Appendix B.7.3 and Appendix B.7.4 we will no longer target the true posterior, but instead an approximation thereof. This approximation can be improved, however, by obtaining samples which look 'more exact', often achievable by increasing the burn-in period and/or introducing a lag between samples (b and l of Algorithm 19).

B.8 Data analysis

In this section, we provide details supporting the data analysis of Section 4.5. This includes further details on the data and how it was processed, and extra information regarding the integer-weighted extension of the SNF model (Lunagómez et al., 2021) used in Section 4.5.3.

B.8.1 Foursquare data processing

The data analysed in Section 4.5 was obtained from the New York and Tokyo data set of Yang et al. (2015b)[2], which contains a total of 10 months of check-in activity (from 12 April 2012 to 16 February 2013). Each check-in has an associated time stamp, GPS location and venue category information. In particular, for each city, there is a tsv file containing the following columns

1. User ID - unique identifier for the user, e.g. `479`
2. Venue ID - unique identifier for the venue, e.g. `49bbd6c0f964a520f4531fe3`
3. Venue category ID - unique identifier for the venue category, e.g. `4bf58dd8d48988d127951735`
4. Venue category name - name for venue category, e.g. `Arts & Crafts`
5. Latitude & longitude - geographical location for venue, e.g. `(40.41,-74.00)`
6. UTC time - time of check-in, to the second, e.g. `Tue Apr 03 18:00:09 +0000 2012`
7. Time zone offset - the offset of local time from UTC for venue (in minutes), e.g. `-240`

[2] https://sites.google.com/site/yangdingqi/home/foursquare-dataset#h.p_ID_46

As outlined in Section 2.4.2, we converted this raw data to a sequence or multiset of paths. In particular, we let the vertices $\mathcal{V}$ denote venue categories with a path then representing a day of check-ins for a given user. Notice not all of the information above is required to enact this operation. In particular, all one requires are user IDs, venue category names (or IDs) and local time (a function of UTC and time zone offset).

Venue category hierarchy

As discussed in Section 4.5, the venue categories have a hierarchical structure. For example a venue of category "Tram Station" is a sub-category of "Train Station", which is itself a sub-category of "Travel & Transport", implying a hierarchical label given by "Travel & Transport $>$ Train Station $>$ Tram Station". As it comes, the data set of Yang et al. (2015b) uses low-level category names ("Tram Station"), whilst we consider the highest-level ("Travel & Transport"). However, we do note that Yang et al. (2015b) do not appear to have used the *lowest* level in all cases.

To get the hierarchical category names we made use of information on the Foursquare site (see here). Note that since the release of this data set it appears that Foursquare have changed how they label venues, thus there is another set of venue category names (see here). However, the data set of Yang et al. (2015b) appears to be congruent with the former. Using this information we were able to essentially 'fill-in' the higher-level category labels for each category name appearing in the data set of Yang et al. (2015b), mapping their low-level labels to top-level ones.

Data filtering

As mentioned in Section 4.5, we analysed only a subset of 100 data points. This was due to issues causes by the presence of outliers. In this subsection, we outline exactly how this subset of data points was chosen.

Following processing of the raw data we were left with a sample of interaction multisets $\{\mathcal{E}^{(i)}\}_{i=1}^{n}$ with $n = 928$. As we discussed in Section 4.5.1, after some initial filtering, including the removal of all length one paths and observations with less than 10 paths, we were left with $n = 402$ observations. To get the final $n = 100$ data points we further subset these data by making use of a distance metric between observations.

Suppose that d_E is some distance between interaction multisets, then one can choose a subset of size m as follows: find the data point which has the smallest total distance to its m nearest neighbours, taking this neighbourhood as the subset. More formally, introducing the notation $\mathcal{N}_m(\mathcal{E})$ for the indices of the m nearest neighbours of $\mathcal{E}$ with respect to d_E in the sample, we let

$$\mathcal{E}^* = \underset{\mathcal{E}\in\{\mathcal{E}^{(i)}\}_{i=1}^{n}}{\operatorname{argmin}} \left[\sum_{i\in\mathcal{N}_m(\mathcal{E})} d_E(\mathcal{E}, \mathcal{E}^{(i)}) \right],$$

with the desired subset then being given by $\{\mathcal{E}^{(i)}\}_{i\in\mathcal{N}_m(\mathcal{E}^*)}$.

Regarding the choice of distance d_E, we opted for that used in the model-fit, namely the matching distance with an LSP distance between paths. Moreover, since the observations were of quite different sizes, we used the normalised version thereof (via the Steinhaus transform, as seen in Section 3.2). To see why using this normalised distance is sensible an example is helpful. Consider comparing $\mathcal{E} = \{(1,1,1)\}$ with the following two observations

$$\mathcal{E}^{(1)} = \{(2,2,2)\} \quad \mathcal{E}^{(2)} = \{(1,1,1),(2,2,2),(2,2,2)\}.$$

Observe that $\mathcal{E}^{(1)}$ shares nothing in common with $\mathcal{E}$ whilst $\mathcal{E}^{(2)}$ and $\mathcal{E}$ share a common path, namely $(1,1,1)$. As such, intuitively we might say $\mathcal{E}^{(2)}$ is more similar to $\mathcal{E}$ than

$\mathcal{E}^{(1)}$ is, that is, its distance should be lower. However, in this case we will have

$$d_{\mathrm{M}}(\mathcal{E}, \mathcal{E}^{(1)}) = 6 \quad d_{\mathrm{M}}(\mathcal{E}, \mathcal{E}^{(2)}) = 6$$

which appears to contradict this intuition. The problem here is the difference in the observation sizes; though $\mathcal{E}^{(2)}$ is more similar to $\mathcal{E}$ it is also larger, hence pushing up its distance. However, by taking sizes into account, the normalised distances evaluate to

$$\bar{d}_{\mathrm{M}}(\mathcal{E}, \mathcal{E}^{(1)}) = \frac{2 \times 6}{3 + 3 + 6} = 1 \qquad \bar{d}_{\mathrm{M}}(\mathcal{E}, \mathcal{E}^{(2)}) = \frac{2 \times 6}{3 + 9 + 6} = \frac{2}{3}$$

which better agrees with the intuition that $\mathcal{E}^{(2)}$ is closer to $\mathcal{E}$. As such, if we use the normalised distance we are likely to select a sample of data points which share aspects in common, hence providing an underlying signal which our method can uncover. If we instead used the regular distance it is possible we may choose a sample of data which has no such common signal, causing our method to output inferences of little interest.

B.8.2 Multigraph SNF model

Here we provide extra details regarding the generalisation of the SNF models (Lunagómez et al., 2021) used in Section 4.5.3. In particular, we extend the SNF to model multigraphs. Let $\mathcal{V} = \{1, \ldots, V\}$ denote the fixed set of vertices, and let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a multigraph (directed or undirected, and possibly with self-loops), where $\mathcal{E}$ is a *multiset* of edges, so that an edge (i, j) can appear more than once in $\mathcal{E}$. A multigraph $\mathcal{G}$ can also be represented uniquely by its adjacency matrix $A^{\mathcal{G}} \in \mathbb{Z}_{\geq 0}^{V \times V}$, where $A^{\mathcal{G}}_{ij} \in \mathbb{Z}_{\geq 0}$ denotes the multiplicity of the edge (i, j) in $\mathcal{E}$.

To define a model, we place a probability distribution over *all* multigraphs (over the vertex set $\mathcal{V}$). This space, which we denote $\mathscr{G}$, can be defined via the one-to-one correspondence with adjacency matrices as follows

$$\mathscr{G} = \{\mathcal{G} \,:\, A^{\mathcal{G}} \in \mathbb{Z}_{\geq 0}^{V \times V}\},$$

so that we seek to assign each $\mathcal{G} \in \mathscr{G}$ a probability. Following the same rationale as the SNF models (and the models of this paper), we construct this model via location and scale. Moreover, this is done with the use of distance metrics, this time between multigraphs. We have two parameters, the mode $\mathcal{G}^m \in \mathscr{G}$ (location) and the dispersion $\gamma > 0$ (scale). We also assume that a distance metric has been pre-specified $d_G(\mathcal{G}, \mathcal{G}')$, quantifying the dissimilarity of any two multigraphs $\mathcal{G}$ and $\mathcal{G}'$. Given this, we assume the probability of $\mathcal{G} \in \mathscr{G}$ is, up to proportionality, the following

$$p(\mathcal{G}|\mathcal{G}^m, \gamma) \propto \exp\{-\gamma\phi(d_G(\mathcal{G}, \mathcal{G}^m))\} \tag{B.8.1}$$

where $\phi(\cdot)$ is a non-negative strictly increasing function with $\phi(0) = 0$. The notation $\mathcal{G} \sim \text{SNF}(\mathcal{G}^m, \gamma)$ is used when $\mathcal{G}$ is assumed to have been sampled from this probability distribution. The normalising constant of (B.8.1) is given by the following

$$Z(\mathcal{G}^m, \gamma) = \sum_{\mathcal{G} \in \mathscr{G}} \exp\{-\gamma\phi(d_G(\mathcal{G}, \mathcal{G}^m))\},$$

which, with $\mathscr{G}$ being an infinite space, will in general be intractable.

Note this is more-or-less identical the SNF models seen in Lunagómez et al. (2021), Definition 3.4. The only differences being (i) the sample space $\mathscr{G}$ is now all muligraphs over $\mathcal{V}$, and (ii) the distance metrics $d_G(\cdot, \cdot)$ are between multigraphs.

Supposing that a sample of multigraphs $\{\mathcal{G}^{(i)}\}_{i=1}^n$ has been observed, as discussed in Section 4.5.3, we can use this mutligraph-based SNF to construct the following

hierarchical model

$$\begin{aligned}\mathcal{G}^{(i)} &\sim \text{SNF}(\mathcal{G}^m, \gamma) \quad (\text{for } i = 1, \ldots, n)\\ \mathcal{G}^m &\sim \text{SNF}(\mathcal{G}_0, \gamma_0)\\ \gamma &\sim p(\gamma)\end{aligned}$$

where $\mathcal{G}_0 \in \mathscr{G}$ and $\gamma_0 > 0$ are hyperparameters, and $p(\gamma)$ denotes a prior for the dispersion. The goal of inference is to now estimate $\mathcal{G}^m$ and γ, representing notations of average and variance, respectively, and can be achieved by sampling from the posterior via MCMC. The posterior in this case is given by the following

$$\begin{aligned}p(\mathcal{G}^m, \gamma | \{\mathcal{G}^{(i)}\}_{i=1}^n) &\propto \left(\prod_{i=1}^n p(\mathcal{G}^{(i)} | \mathcal{G}^m, \gamma)\right) p(\mathcal{G}^m) p(\gamma)\\ &= Z(\mathcal{G}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^m))\right\}\\ &\quad \times \exp\left\{-\gamma_0 \phi(d_G(\mathcal{G}^m, \mathcal{G}_0))\right\} p(\gamma),\end{aligned}$$

which, since $Z(\mathcal{G}^m, \gamma)$ is intractable and depends on the parameters being sampled, is doubly-intractable (Murray et al., 2006). As such, to sample from it one must use a specialised MCMC algorithm. Since we are dealing with multigraphs, we cannot apply the scheme proposed by Lunagómez et al. (2021) directly, and instead propose an alternative approach via the exchange algorithm (Murray et al., 2006). In particular, we considered a component-wise MCMC algorithm which alternates between sampling from the two conditionals (i) $p(\gamma | \mathcal{G}^m, \{\mathcal{G}^{(i)}\}_{i=1}^n)$, and (ii) $p(\mathcal{G}^m | \gamma, \{\mathcal{G}^{(i)}\}_{i=1}^n)$. For (i) we apply the exchange algorithm directly, whilst for (ii) do an exchange-within-Gibbs step, updating each edge in turn in a single repetition.

We first outline the procedure to update the dispersion. Assume that $q(\gamma' | \gamma)$ denotes a suitable proposal density. With $\mathcal{G}^m$ fixed and current state γ, first sample proposal γ' from $q(\gamma' | \gamma)$. Next, sample auxiliary data $\{\mathcal{G}_i^*\}_{i=1}^n$ i.i.d. where $\mathcal{G}_i^* \sim$

SNF$(\mathcal{G}^m, \gamma')$ and then accept γ' with the following probability

$$\begin{aligned}\alpha(\gamma', \gamma) &= \min\left\{1, \frac{p(\gamma'|\mathcal{G}^m, \{\mathcal{G}^{(i)}\}_{i=1}^n)\prod_{i=1}^n p(\mathcal{G}_i^*|\mathcal{G}^m, \gamma)q(\gamma|\gamma')}{p(\gamma|\mathcal{G}^m, \{\mathcal{G}^{(i)}\}_{i=1}^n)\prod_{i=1}^n p(\mathcal{G}_i^*|\mathcal{G}^m, \gamma')q(\gamma'|\gamma)}\right\} \\ &= \min\{1, H(\gamma', \gamma)\}\end{aligned}$$

where

$$\begin{aligned}H(\gamma', \gamma) = \exp\left\{-(\gamma' - \gamma)\left(\sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^m)) - \sum_{i=1}^n \phi(d_G(\mathcal{G}_i^*, \mathcal{G}^m))\right)\right\} \\ \times \frac{p(\gamma')}{p(\gamma)}\frac{q(\gamma|\gamma')}{q(\gamma'|\gamma)}.\end{aligned} \tag{B.8.2}$$

For the proposal $q(\gamma'|\gamma)$ we consider sampling uniformly over a ε-neighbourhood of γ with reflection at zero (see Appendix B.6.1, Eq. B.6.2), for which one has $q(\gamma'|\gamma) = q(\gamma|\gamma')$.

To update the mode, we consider a exchange-within-Gibbs scheme, whereby we scan through all edges, propose new multiplicities and accept these with some probability. Assume one has defined a proposal $q(x'|x)$, which proposes a new edge multiplicity $x' \in \mathbb{Z}_{\geq 0}$ given current value $x \in \mathbb{Z}_{\geq 0}$. With γ fixed and current state $\mathcal{G}^m$, with A^m its adjacency matrix (abbreviating notation for readability), we first generate proposal $\mathcal{G}^{m'}$ by proposing a new multiplicity for edge (i, j). More precisely, letting $x = A_{ij}^m$ denote the current multiplicity, we sample x' from $q(x'|x)$, then construct proposal $\mathcal{G}^{m'}$ via its adjacency matrix $A^{m'}$, defined to be

$$A_{kl}^{m'} = \begin{cases} x' & \text{if } (k, l) = (i, j) \\ A_{kl}^m & \text{else} \end{cases}$$

that is, $A^{m'}$ is equal to A^m with the (ij)th entry altered from x to x'. Note this step will alter if we are considering undirected multigraphs, where we must let $A_{ij}^{m'} = A_{ij}^{m'} = x'$, since the adjacency matrices must be symmetric. However, in the remainder of

these details it will be assumed the multigraphs are directed. Given proposal $\mathcal{G}^{m'}$, we next sample auxiliary data $\{\mathcal{G}_i^*\}_{i=1}^n$ i.i.d. where $\mathcal{G}_i^* \sim \text{SNF}(\mathcal{G}^{m'}, \gamma)$ and then accept $\mathcal{G}^{m'}$ with the following probability

$$\begin{aligned}\alpha(\mathcal{G}^{m'}, \mathcal{G}^m) &= \min\left\{1, \frac{p(\mathcal{G}^{m'}|\gamma, \{\mathcal{G}^{(i)}\}_{i=1}^n)\prod_{i=1}^n p(\mathcal{G}_i^*|\mathcal{G}^m, \gamma)q(x|x')}{p(\mathcal{G}^m|\gamma, \{\mathcal{G}^{(i)}\}_{i=1}^n)\prod_{i=1}^n p(\mathcal{G}_i^*|\mathcal{G}^{m'}, \gamma)q(x'|x)}\right\} \\ &= \min\Big\{1, H(\mathcal{G}^{m'}, \mathcal{G}^m)\Big\}\end{aligned}$$

where

$$\begin{aligned}H(\mathcal{G}^{m'}, \mathcal{G}^m) = \exp\Bigg\{ &-\gamma\left(\sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^{m'})) - \sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^m))\right) \\ &-\gamma\left(\sum_{i=1}^n \phi(d_G(\mathcal{G}_i^*, \mathcal{G}^m)) - \sum_{i=1}^n \phi(d_G(\mathcal{G}_i^*, \mathcal{G}^{m'}))\right) \\ &-\gamma_0\Big(\phi(d_G(\mathcal{G}^{m'}, \mathcal{G}_0)) - \phi(d_G(\mathcal{G}^m, \mathcal{G}_0))\Big)\Bigg\}\frac{q(x|x')}{q(x'|x)}.\end{aligned} \tag{B.8.3}$$

The steps above update the multiplicity of a single edge (i, j). In a single iteration of updating the mode $\mathcal{G}^m$, we consider looping over each $(i, j) \in \mathcal{V} \times \mathcal{V}$, updating their multiplicity in this manner, leading to what can be seen as an exchange-within-Gibbs step for sampling from $p(\mathcal{G}^m|\gamma, \{\mathcal{G}^{(i)}\}_{i=1}^n)$.

For the proposal $q(x'|x)$, we consider uniform sampling over a ν-neighbourhood of x with reflection as zero. More precisely, given current state $x \in \mathbb{Z}_{\geq 0}$, sample proposal x' via

1. Sample $x^* \sim \text{Uniform}(A)$ where

$$A = \{j \in \mathbb{Z} : x - \nu \leq j \leq x + \nu\} \setminus \{x\}$$

   is the ν-neighbourhood of x in $\mathbb{Z}$, excluding x, then

2. If $x^* \geq 0$ let $x' = x^*$, else let $x' = -x^*$,

for which one has

$$q(x'|x) = \begin{cases} 0 & \text{if } x = x' \\ \frac{1}{\nu} & \text{if } x + x' \leq \nu \\ \frac{1}{2\nu} & \text{else} \end{cases}$$

and hence $q(x'|x) = q(x|x')$, which will lead to cancellation of such terms in (B.8.3).

Finally, we note that both of these schemes to sample from $p(\mathcal{G}^m|\gamma, \{\mathcal{G}_i^*\}_{i=1}^n)$ and $p(\gamma|\mathcal{G}^m, \{\mathcal{G}^{(i)}\}_{i=1}^n)$ require the ability to obtain an i.i.d. sample $\{\mathcal{G}_i^*\}_{i=1}^n$ where $\mathcal{G}_i^* \sim$ SNF$(\mathcal{G}^m, \gamma)$ for some given $(\mathcal{G}^m, \gamma)$. Unfortunately, this cannot be done in general. However, we can replace this with approximate MCMC-based samples, exactly as we did for our interaction-sequence and interaction-multiset models (Section 4.3.6). To do so, we re-use the scheme above (without auxiliary sampling).

In particular, with current state $\mathcal{G}$, we update edge (i, j) as follows. Letting $x = A_{ij}^{\mathcal{G}}$, we sample x' from $q(x'|x)$ (via ν-neighbourhood as above), constructing proposal $\mathcal{G}'$ via its adjacency matrix

$$A_{kl}^{\mathcal{G}'} = \begin{cases} x' & \text{if } (k, l) = (i, j) \\ A_{kl}^{\mathcal{G}} & \text{else} \end{cases}$$

that is, $\mathcal{G}'$ is equivalent to $\mathcal{G}$ with the multiplicity of edge (i, j) flipped from x to x'. We then accept $\mathcal{G}'$ with the following probability

$$\begin{aligned} \alpha(\mathcal{G}, \mathcal{G}') &= \min\left\{1, \frac{p(\mathcal{G}'|\mathcal{G}^m, \gamma)q(x|x')}{p(\mathcal{G}|\mathcal{G}^m, \gamma)q(x'|x)}\right\} \\ &= \min\left\{1, \exp\left\{-\gamma\left(\phi(d_G(\mathcal{G}', \mathcal{G}^m)) - \phi(d_G(\mathcal{G}, \mathcal{G}^m))\right)\right\}\frac{q(x|x')}{q(x'|x)}\right\}. \end{aligned}$$

Note this will update a single edge (i, j). One could now follow the approach used to update the mode $\mathcal{G}^m$, looping over all edges in turn. However, in this case we opt to instead choose a single edge at random to update. That is, in a single iteration,

we choose (i, j) uniformly from $\mathcal{V} \times \mathcal{V}$, and update it as above. This can be seen as a Gibbs sampler with a randomised sweep strategy (Levine and Casella, 2006).

B.9 Pseudocode

Algorithm 12: SIS posterior component-wise MCMC

Input: observed data $\{\mathcal{S}^{(i)}\}_{i=1}^{n}$
initialise $(\mathcal{S}_0^m, \gamma_0)$
for $i = 1, \ldots, m$ **do**
    // Update gamma
    $\gamma_i =$ `dispersion_update`$(\mathcal{S}_{i-1}^m, \gamma_{i-1})$ // (Algorithm 13)
    // Update mode
    $\mathcal{S}_i^m =$ `mode_update`$(\mathcal{S}_{i-1}^m, \gamma_i)$ // (Algorithm 14)
end
Output: sample $\{(\mathcal{S}_i^m, \gamma_i)\}_{i=1}^{m}$

Algorithm 13: SIS posterior dispersion conditional accept-reject

Input: $(\mathcal{S}_i^m, \gamma_i)$
Output: γ_{i+1}
Function `dispersion_update`$(\mathcal{S}_i^m, \gamma_i)$**:**
    let $(\mathcal{S}^m, \gamma) = (\mathcal{S}_i^m, \gamma_i)$
    sample γ' via $q(\gamma'|\gamma)$ of (B.6.2) // Sample proposal
    sample $\{\mathcal{S}_i^*\}_{i=1}^{n}$ i.i.d. from SIS$(\mathcal{S}^m, \gamma')$ // Sample auxiliary data
    evaluate $\alpha = \alpha(\gamma, \gamma')$ of (B.6.1) // Acceptance probability
    $$\gamma_{i+1} = \begin{cases} \gamma' & \text{with probability } \alpha \\ \gamma & \text{with probability } (1-\alpha) \end{cases}$$
    return γ_{i+1} // Accept/reject proposal
end

Algorithm 14: SIS posterior mode conditional accept-reject

Input: $(\mathcal{S}_i^m, \gamma_i)$
Output: $\mathcal{S}_{i+1}^m$
function `mode_update`$(\mathcal{S}_i^m, \gamma_i)$**:**

let $(\mathcal{S}^m, \gamma) = (\mathcal{S}_i^m, \gamma_i)$
sample $z \sim \text{Bernoulli}(\beta)$
if $z = 1$ **then**
// Edit allocation move
let u, $f(u, \mathcal{S}^m)$ and $p(u|\mathcal{S}^m)$ be as in Appendix B.6.3
sample u via $p(u|\mathcal{S}^m)$ // Sample auxiliary variable
$([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)$ // Invoke involution
sample $\{\mathcal{S}_i^*\}_{i=1}^n$ i.i.d. from SIS$([\mathcal{S}^m]', \gamma)$ // Sample auxiliary data
$\alpha = \alpha(\mathcal{S}^m, [\mathcal{S}^m]')$ of (B.6.2), using ratio (B.6.6) // Acceptance probability
else
// Path insertion & deletion move
let u, $f(u, \mathcal{S}^m)$ and $p(u|\mathcal{S}^m)$ be as in Appendix B.6.4
sample u via $p(u|\mathcal{S}^m)$ // Sample auxiliary variable
$([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)$ // Invoke involution
sample $\{\mathcal{S}_i^*\}_{i=1}^n$ i.i.d. from SIS$([\mathcal{S}^m]', \gamma)$ // Sample auxiliary data
$\alpha = \alpha(\mathcal{S}^m, [\mathcal{S}^m]')$ of (B.6.2), using ratio (B.6.8) // Acceptance probability
end

$$\mathcal{S}_{i+1}^m = \begin{cases} [\mathcal{S}^m]' & \text{with probability } \alpha \\ \mathcal{S}^m & \text{with probability } (1-\alpha) \end{cases}$$ // Accept/reject proposal

return $\mathcal{S}_{i+1}^m$
end

Algorithm 15: SIS model iMCMC sampling

Input: $(\mathcal{S}^m, \gamma)$ (model parameters)
Input: ν_{ed}, ν_{td}, $p(\mathcal{I}|\mathcal{S})$, β (MCMC tuning parameters)
Input: m (sample size), b (burn-in), l (lag)
initialise $\mathcal{S}$;
initialise $i = 1$;
while $i \leq m$ **do**
  sample $z \sim$ Bernoulli(β)
  if $z = 1$ **then**
    `// Edit allocation move`
    let u, $f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Appendix B.6.3
    sample u via $p(u|\mathcal{S})$
    $(\mathcal{S}', u') = f(\mathcal{S}, u)$
    evaluate $\alpha = \alpha(\mathcal{S}, \mathcal{S}')$ of (B.6.9) using (B.6.10)
  else
    `// Path insertion & deletion move`
    let u, $f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Appendix B.6.4
    sample u via $p(u|\mathcal{S})$
    $(\mathcal{S}', u') = f(\mathcal{S}, u)$
    evaluate $\alpha = \alpha(\mathcal{S}, \mathcal{S}')$ of (B.6.9) using (B.6.11)
  end
  `// Accept/reject proposal`
  $$\mathcal{S} = \begin{cases} \mathcal{S}' & \text{with probability } \alpha \\ \mathcal{S} & \text{with probability } (1 - \alpha) \end{cases}$$
  `// Store sample (accounting for lag and burn-in)`
  if $(i > b)$ *and* $(i \mod l = 1)$ **then**
    $\mathcal{S}_i \leftarrow \mathcal{S}$
    $i = i + 1$
  end
end
Output: $\{\mathcal{S}_i\}_{i=1}^m$

Algorithm 16: SIM posterior component-wise MCMC

Input: observed data $\{\mathcal{E}^{(i)}\}_{i=1}^n$
initialise $(\mathcal{E}_0^m, \gamma_0)$
for $i = 1, \ldots, m$ **do**
  `// Update gamma`
  $\gamma_i =$ `dispersion_update`$(\mathcal{E}_{i-1}^m, \gamma_{i-1})$ `// (Algorithm 17)`
  `// Update mode`
  $\mathcal{E}_i^m =$ `mode_update`$(\mathcal{E}_{i-1}^m, \gamma_i)$ `// (Algorithm 14)`
end
Output: sample $\{(\mathcal{E}_i^m, \gamma_i)\}_{i=1}^m$

Algorithm 17: SIM posterior dispersion conditional accept-reject

Input: $(\mathcal{S}_i^m, \gamma_i)$

Output: γ_{i+1}

Function `dispersion_update`$(\mathcal{E}_i^m, \gamma_i)$**:**

let $(\mathcal{E}^m, \gamma) = (\mathcal{E}_i^m, \gamma_i)$

sample γ' via $q(\gamma'|\gamma)$ of (B.6.2) // Sample proposal

sample $\{\mathcal{E}_i^*\}_{i=1}^n$ i.i.d. from SIM$(\mathcal{E}^m, \gamma')$ // Sample auxiliary data

evaluate $\alpha = \alpha(\gamma, \gamma')$ of (B.7.2) // Acceptance probability

$$\gamma_{i+1} = \begin{cases} \gamma' & \text{with probability } \alpha \\ \gamma & \text{with probability } (1-\alpha) \end{cases}$$

return γ_{i+1} // Accept/reject proposal

end

Algorithm 18: SIM posterior mode conditional accept-reject

Input: $(\mathcal{E}_i^m, \gamma_i)$
Output: $\mathcal{E}_{i+1}^m$
function `mode_update`$(\mathcal{E}_i^m, \gamma_i)$**:**

let $(\mathcal{E}^m, \gamma) = (\mathcal{E}_i^m, \gamma_i)$
obtain $\mathcal{S}^m$ from $\mathcal{E}^m$ // Place paths in arbitrary order
sample $z \sim$ Bernoulli(β)
if $z = 1$ **then**
// Edit allocation move
let u, $f(u, \mathcal{S}^m)$ and $p(u|\mathcal{S}^m)$ be as in Appendix B.6.3
sample u via $p(u|\mathcal{S}^m)$ // Sample auxiliary variable
$([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)$ // Invoke involution
obtain $[\mathcal{E}^m]'$ from $[\mathcal{S}^m]'$ // Disregard order
sample $\{\mathcal{E}_i^*\}_{i=1}^n$ i.i.d. from SIM$([\mathcal{E}^m]', \gamma)$ // Sample auxiliary data
$\alpha = \alpha(\mathcal{E}^m, [\mathcal{E}^m]')$ of (B.7.6), using ratio (B.6.6) // Acceptance probability
else
// Path insertion & deletion move
let u, $f(u, \mathcal{S}^m)$ and $p(u|\mathcal{S}^m)$ be as in Appendix B.6.4
sample u via $p(u|\mathcal{S}^m)$ // Sample auxiliary variable
$([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)$ // Invoke involution
obtain $[\mathcal{E}^m]'$ from $[\mathcal{S}^m]'$ // Disregard order
sample $\{\mathcal{E}_i^*\}_{i=1}^n$ i.i.d. from SIM$([\mathcal{E}^m]', \gamma)$ // Sample auxiliary data
$\alpha = \alpha(\mathcal{E}^m, [\mathcal{E}^m]')$ of (B.7.6), using ratio (B.6.8) // Acceptance probability
end

$$\mathcal{E}_{i+1}^m = \begin{cases} [\mathcal{E}^m]' & \text{with probability } \alpha \\ \mathcal{E}^m & \text{with probability } (1-\alpha) \end{cases}$$ // Accept/reject proposal

return $\mathcal{E}_{i+1}^m$
end

Algorithm 19: SIM model iMCMC sampling

Input: $(\mathcal{E}^m, \gamma)$ (model parameters)
Input: ν_{ed}, ν_{td}, $p(\mathcal{I}|\mathcal{S})$, β (MCMC tuning parameters)
Input: m (sample size), b (burn-in), l (lag)
initialise $\mathcal{E}$;
initialise $i = 1$;
while $i \leq m$ **do**
  obtain $\mathcal{S}$ from $\mathcal{E}$ `// Place paths in arbitrary order`
  sample $z \sim \text{Bernoulli}(\beta)$
  if $z = 1$ **then**
    `// Edit allocation move`
    let u, $f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Appendix B.6.3
    sample u via $p(u|\mathcal{S})` `// Sample auxiliary variable`
    $(\mathcal{S}', u') = f(\mathcal{S}, u)$ `// Invoke involution`
    obtain $\mathcal{E}'$ from $\mathcal{S}'$ `// Disregard order`
    $\alpha = \alpha(\mathcal{E}, \mathcal{E}')$ of (B.7.10) using (B.6.10) `// Acceptance probability`
  else
    `// Path insertion & deletion move`
    let u, $f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Appendix B.6.4
    sample u via $p(u|\mathcal{S})` `// Sample auxiliary variable`
    $(\mathcal{S}', u') = f(\mathcal{S}, u)$ `// Invoke involution`
    obtain $\mathcal{E}'$ from $\mathcal{S}'$ `// Disregard order`
    $\alpha = \alpha(\mathcal{E}, \mathcal{E}')$ of (B.7.10) using (B.6.11) `// Acceptance probability`
  end
  `// Accept/reject proposal`
  $$\mathcal{E} = \begin{cases} \mathcal{E}' & \text{with probability } \alpha \\ \mathcal{E} & \text{with probability } (1 - \alpha) \end{cases}$$
  `// Store sample (accounting for lag and burn-in)`
  if $(i > b)$ *and* $(i \mod l = 1)$ **then**
    $\mathcal{E}_i \leftarrow \mathcal{E}$
    $i = i + 1$
  end
end
Output: $\{\mathcal{E}_i\}_{i=1}^m$

Bibliography

Arroyo, J., Athreya, A., Cape, J., Chen, G., Priebe, C. E., and Vogelstein, J. T. (2021). Inference for multiple heterogeneous networks with a common invariant subspace. *Journal of machine learning research*, 22(142).

Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T., Levin, K., Lyzinski, V., and Qin, Y. (2017). Statistical inference on random dot product graphs: a survey. *The Journal of Machine Learning Research*, 18(1):8393–8484.

Behrens, T. E. and Sporns, O. (2012). Human connectomics. *Current Opinion in Neurobiology*, 22:144–153.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Cai, D., Campbell, T., and Broderick, T. (2016). Edge-exchangeable graphs and sparsity. *Advances in Neural Information Processing Systems*, 29.

Caron, F. and Fox, E. B. (2017). Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(5):1295–1366.

Chung, J., Bridgeford, E., Arroyo, J., Pedigo, B. D., Saad-Eldin, A., Gopalakrishnan, V., Xiang, L., Priebe, C. E., and Vogelstein, J. T. (2021). Statistical connectomics. *Annual Review of Statistics and Its Application*, 8:463–492.

Crane, H. and Dempsey, W. (2018). Edge exchangeable models for interaction networks. *Journal of the American Statistical Association*, 113:1311–1326.

Deza, M. M. and Deza, E. (2009). *Encyclopedia of Distances*. Springer.

Donnat, C. and Holmes, S. (2018). Tracking network dynamics: A survey using graph distances. *Annals of Applied Statistics*, 12:971–1012.

Durante, D., Dunson, D. B., and Vogelstein, J. T. (2017). Nonparametric Bayes modeling of populations of networks. *Journal of the American Statistical Association*, 112:1516–1530.

Eiter, T. and Mannila, H. (1997). Distance measures for point sets and their computation. *Acta informatica*, 34(2):109–133.

Erdős, P., Rényi, A., et al. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. (2021). Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8.

Fox, K. and Li, X. (2019). Approximating the geometric edit distance. *Leibniz International Proceedings in Informatics, LIPIcs*, 149.

Ginestet, C. E., Li, J., Balachandran, P., Rosenberg, S., and Kolaczyk, E. D. (2017). Hypothesis testing for network data in functional neuroimaging. *Annals of Applied Statistics*, 11:725–750.

Gold, O. and Sharir, M. (2018). Dynamic time warping and geometric edit distance: breaking the quadratic barrier. *ACM Transactions on Algorithms*, 14.

Gollini, I. and Murphy, T. B. (2016). Joint modeling of multiple network views. *Journal of Computational and Graphical Statistics*, 25:246–265.

Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098.

Holland, P. W. and Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Izenman, A. J. (2008). Modern multivariate statistical techniques. *Regression, classification and manifold learning*, 10:978–0.

Josephs, N., Lin, L., Rosenberg, S., and Kolaczyk, E. D. (2023). Bayesian classification, anomaly detection, and survival analysis using network inputs with application to the microbiome. *The Annals of Applied Statistics*, 17(1):199–224.

Kim, B., Lee, K. H., Xue, L., and Niu, X. (2018). A review of dynamic network models with latent variables. *Statistics surveys*, 12:105.

Kolaczyk, E. D. and Csárdi, G. (2014). *Statistical analysis of network data with R*, volume 65. Springer.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Le, C. M., Levin, K., and Levina, E. (2018). Estimating a network from multiple noisy realizations. *Electronic Journal of Statistics*, 12:4697–4740.

Lee, C. and Wilkinson, D. J. (2019). A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1):1–50.

Lehmann, B. and White, S. (2021). Bayesian exponential random graph models for populations of networks. *arXiv preprint arXiv:2104.05110*.

Levin, K., Athreya, A., Tang, M., Lyzinski, V., and Priebe, C. E. (2017). A central limit theorem for an omnibus embedding of multiple random dot product graphs. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 964–967. IEEE.

Levine, R. A. and Casella, G. (2006). Optimizing random scan gibbs samplers. *Journal of Multivariate Analysis*, 97(10):2071–2100.

Liang, F. (2010). A double Metropolis-Hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computation and Simulation*, 80:1007–1022.

Lunagómez, S., Olhede, S. C., and Wolfe, P. J. (2021). Modeling network populations via graph distances. *Journal of the American Statistical Association*, 116(536):2023–2040.

Mantziou, A., Lunagomez, S., and Mitra, R. (2021). Bayesian model-based clustering for multiple network data. *arXiv preprint arXiv:2107.03431*.

Mardia, K. V. and Dryden, I. L. (1999). The complex Watson distribution and shape analysis. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 61:913–926.

McInnes, L., Healy, J., and Astels, S. (2017). hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205.

McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Murray, I., Ghahramani, Z., and MacKay, D. J. (2006). Mcmc for doubly-intractable distributions. *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, UAI 2006*, pages 359–366.

Møller, J., Pettitt, A. N., Reeves, R., and Berthelsen, K. K. (2006). An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93:451–458.

Neklyudov, K., Welling, M., Egorov, E., and Vetrov, D. (2020). Involutive mcmc: A unifying framework. In *International Conference on Machine Learning*, pages 7273–7282. PMLR.

Newman, M. E. (2018). Estimating network structure from unreliable measurements. *Physical Review E*, 98(6):062321.

Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

Nielsen, A. M. and Witten, D. (2018). The multiple random dot product graph model. *arXiv preprint arXiv:1811.12172*.

Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087.

Orbanz, P. and Roy, D. M. (2014). Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):437–461.

Peixoto, T. P. (2018). Reconstructing networks with unknown and heterogeneous errors. *Physical Review X*, 8(4):041011.

Peyré, G. and Cuturi, M. (2019). Computational optimal transport. *Foundations and Trends in Machine Learning*, 11:1–257.

Raftery, A. E., Niu, X., Hoff, P. D., and Yeung, K. Y. (2012). Fast inference for the latent space network model using a case-control approximate likelihood. *Journal of computational and graphical statistics*, 21(4):901–919.

Ramon, J. and Bruynooghe, M. (2001). A polynomial time computable metric between point sets. *Acta Informatica*, 37:765–780.

Reyes, P. and Rodriguez, A. (2016). Stochastic blockmodels for exchangeable collections of networks. *arXiv preprint arXiv:1606.05277*.

Salter-Townshend, M. and Murphy, T. B. (2013). Variational bayesian inference for the latent position cluster model for network data. *Computational Statistics Data Analysis*, 57(1):661–671.

Salter-Townshend, M., White, A., Gollini, I., and Murphy, T. B. (2012). Review of statistical network analysis: models, algorithms, and software. *Statistical Analysis and Data Mining*, 5(4):243–264.

Sharrock, L., Dodd, D., and Nemeth, C. (2023). Coinem: Tuning-free particle-based variational inference for latent variable models. *arXiv preprint arXiv:2305.14916*.

Stanley, N., Shai, S., Taylor, D., and Mucha, P. J. (2016). Clustering network layers with

the strata multilayer stochastic block model. *IEEE Transactions on Network Science and Engineering*, 3(2):95–105.

Sweet, T. M., Thomas, A. C., and Junker, B. W. (2013). Hierarchical network models for education research: Hierarchical latent space models. *Journal of Educational and Behavioral Statistics*, 38(3):295–318.

Sweet, T. M., Thomas, A. C., and Junker, B. W. (2014). Hierarchical mixed membership stochastic blockmodels for multiple networks and experimental interventions. *Handbook on Mixed Membership Models and their Applications*, pages 463–488.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581.

Van Der Maaten, L. (2014). Accelerating t-sne using tree-based algorithms. *The journal of machine learning research*, 15(1):3221–3245.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Vitelli, V., Øystein Sørensen, Crispino, M., Frigessi, A., and Arjas, E. (2018). Probabilistic preference learning with the Mallows rank model. *Journal of Machine Learning Research*, 18:1–49.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM (JACM)*, 21:168–173.

Wang, S., Arroyo, J., Vogelstein, J. T., and Priebe, C. E. (2019). Joint embedding of graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Williamson, S. A. (2016). Nonparametric network models for link prediction. *The Journal of Machine Learning Research*, 17(1):7102–7121.

Wills, P. and Meyer, F. G. (2020). Metrics for graph comparison: a practitioner's guide. *Plos one*, 15(2):e0228728.

Yang, D., Zhang, D., Chen, L., and Qu, B. (2015a). Nationtelescope: Monitoring and visualizing large-scale collective behavior in lbsns. *Journal of Network and Computer Applications*, 55:170–180.

Yang, D., Zhang, D., and Qu, B. (2016). Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(3):1–23.

Yang, D., Zhang, D., Zheng, V. W., and Yu, Z. (2015b). Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45:129–142.

Yang, T., Chi, Y., Zhu, S., Gong, Y., and Jin, R. (2011). Detecting communities and their evolutions in dynamic social networks—a bayesian approach. *Machine learning*, 82:157–189.

Yin, F., Shen, W., and Butts, C. T. (2022). Finite mixtures of ergms for modeling ensembles of networks. *Bayesian Analysis*, 1(1):1–39.

Young, J.-G., Kirkley, A., and Newman, M. (2022). Clustering of heterogeneous populations of networks. *Physical Review E*, 105(1):014312.

Young, S. J. and Scheinerman, E. R. (2007). Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer.