

Modelling Populations of Interaction Networks via Distance Metrics

George Bolt

*STOR-i Centre for Doctoral Training
Lancaster University, Lancaster, UK, LA1 4YF*

G.BOLT@LANCASTER.AC.UK

Simón Lunagómez

*Department of Statistics
Instituto Tecnológico Autónomo de México (ITAM)
Río Hondo 1, Altabista, Álvaro Obregón, 01080 Ciudad de México, CDMX, Mexico*

SIMON.LUNAGOMEZ@ITAM.MX

Christopher Nemeth

*Department of Mathematics and Statistics
Lancaster University, Lancaster, UK, LA1 4YF*

C.NEMETH@LANCASTER.AC.UK

Editor:

Abstract

Network data arises through observation of relational information between a collection of entities. Recent work in the literature has independently considered when (i) one observes a sample of networks, connectome data in neuroscience being a ubiquitous example, and (ii) the units of observation within a network are edges or paths, such as emails between people or a series of page visits to a website by a user, often referred to as interaction network data. The intersection of these two cases, however, is yet to be considered. In this paper, we propose a new Bayesian modelling framework to analyse such data. Given a practitioner-specified distance metric between observations, we define families of models through location and scale parameters, akin to a Gaussian distribution, with subsequent inference of model parameters providing reasoned statistical summaries for this non-standard data structure. To facilitate inference, we propose specialised Markov chain Monte Carlo (MCMC) schemes capable of sampling from doubly-intractable posterior distributions over discrete and multi-dimensional parameter spaces. Through simulation studies we confirm the efficacy of our methodology and inference scheme, whilst its application we illustrate via an example analysis of a location-based social network (LSBN) data set.

Keywords: Interaction network, multiple networks, distance metrics, exchange algorithm, involutive MCMC.

1. Introduction

Network data, defined to be information regarding relations amongst some collection of entities, can appear in various guises, each with subtle idiosyncrasies warranting particular methodological considerations. This work considers the intersection of two such sub-types of network data. On the one hand, we consider when a sample of independent networks is observed. Such data, often assumed to be drawn from some population of networks, is becoming increasingly prevalent in the neuroscience literature (Behrens and Sporns, 2012; Chung et al., 2021), and has motivated recent methodological developments on multiple-

network models (Lunagomez et al., 2021; Le et al., 2018; Durante et al., 2017). On the other hand, we assume that paths or edges represent the units of observation within each network. Typically referred to as interaction networks, work has similarly been done towards defining methodologies suitable to their analysis, most notably with the proposal of so-called edge-exchangeable models (Crane and Dempsey, 2018; Cai et al., 2016; Caron and Fox, 2017). As far as we are aware, the intersection of these two cases, that is, where one observes multiple independent interaction networks, is yet to be considered in the literature.

As a motivating example, consider the Foursquare check-in data set of Yang et al. (2015). Foursquare is a location-based social network (LSBN) where users share places they have visited with their friends by ‘checking-in’ to locations they visit, such as restaurants or music venues. The data set of Yang et al. (2015) contains historical check-ins of users in New York and Tokyo. Considering a single user, we can see a day of check-ins as a path through the set of venue categories, as illustrated in Figure 1. Over an extended time period, we expect to observe check-ins on multiple days, leading to a series of paths being observed. In this way, the data of a single user can be seen as an interaction network. Furthermore, there are data for not one but possibly hundreds of users, that is, we have multiple interaction networks.

With a single observation, we are typically interested in analysing its structure. When faced with an independent sample of networks the following more familiar statistical questions are raised

- (a) What is an average network?
- (b) How variable are observations about this average?
- (c) Is there heterogeneity in the observations?

These questions are consistent with those arising more generally in object orientated data analysis (Marron and Dryden, 2021), which considers the statistical analysis of populations of complex objects. In this work, we focus on (a) and (b) in particular.

1.1 Related Work

We now review closely related work appearing in the literature. Firstly, there has been work on models suitable for multiple-network data, where observations are typically represented via graphs. Here, Lunagomez et al. (2021) construct models through graph distances, using the Frechet mean and entropy as notions of the mean and variance respectively. Le et al. (2018), Peixoto (2018) and Newman (2018) propose measurement error models which view observed networks as noisy realisations of an unknown ground truth. Along similar lines, Mantziou et al. (2021) and Young et al. (2022) have recently extended the measurement error models to capture heterogeneity, providing model-based approaches to clustering networks.

Others have considered adapting models originally proposed to analyse a single network. This includes the latent space model (LSM) (Ho et al., 2002), which has been extended by Sweet et al. (2013), who assume a hierarchical model in which each observation is drawn from an LSM with its own parameter, 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

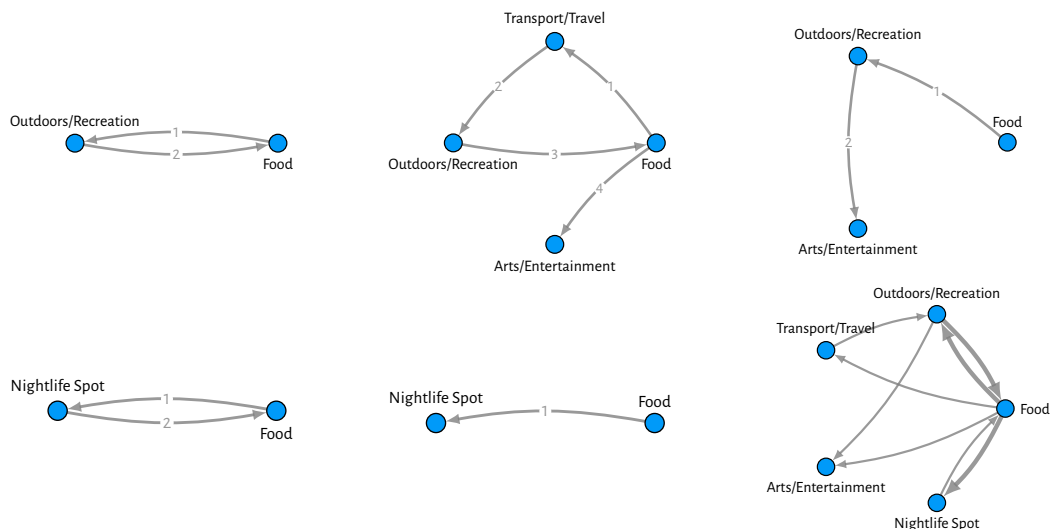


Figure 1: Data of a single user from the Foursquare data set (Yang et al., 2015). Going from left to right, the first five subplots show the observed interactions sequence \mathcal{S} , with each path representing a single day of check-ins, whilst the final plot visualises the aggregate multigraph $\mathcal{G}_{\mathcal{S}}$, where the thickness of an edge is proportional to the number of times it appears in $\mathcal{G}_{\mathcal{S}}$.

with shrinkage priors which induce removal of redundant components and unnecessary dimensions in latent coordinates. The random dot product graph (RDPG) model (Young and Scheinerman, 2007) has also been extended. Here 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. The exponential random graph model (ERGM) (Holland and Leinhardt, 1981; Frank and Strauss, 1986) 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. Finally, others have adapted the stochastic blockmodel (SBM) (Nowicki and Snijders, 2001), with Sweet et al. (2014) building upon their earlier work (Sweet et al., 2013), 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.

There has also been work on hypothesis testing for network-valued data (Ginestet et al., 2017; Durante and Dunson, 2018; Ghoshdastidar et al., 2020; Chen et al., 2021), where we note in particular Ginestet et al. (2017) make use of a distance between graphs, in similar spirit to Lunagomez et al. (2021) and this present work.

All of these works are connected by a desire to answer standard statistical questions in the context of network-valued data. However, none consider paths being the units of observation within each network. Instead, all are designed to analyse network data represented via graphs. As such, to analyse data which is truly path-observed, their use would require

rst aggregating observations to graphs in a pre-processing step; an operation which will often not be injective and hence lead to a potential loss of information.

In another direction, there is related work which has considered edge or path-observed network data. In particular, there has been development of so-called edge-exchangeable network models (Cai et al., 2016; Crane and Dempsey, 2018; Williamson, 2016; Ghaleb et al., 2019b,a). Of these we note that only Crane and Dempsey (2018) allow paths as the observational units. Closely related to the edge-exchangeable models are those based on exchangeable random measures (Caron and Fox, 2017; Veitch and Roy, 2015). These two streams of work are connected in so far as they deviate from more traditional models which are based upon assumptions of vertex exchangeability, and in doing so produce graphs which exhibit sparsity and heavy-tailed degree distributions; features often observed empirically. In another direction, others have considered models based upon higher-order Markov chains (Scholtes, 2017; Peixoto and Rosvall, 2017), where in line with this present work Scholtes (2017) considered paths as the units of observation.

The common theme in all these works is a focus on models which can capture a particular structure within a single network, such as sparsity, heavy-tailed degree distributions or high-order dependence in paths or edges. They are not, however, designed to analyse multiple observations. As such, they can only provide answers to (a)-(c) via post-hoc analysis of parameters inferred for each observation.

1.2 Summary of Contributions

In the literature, it appears there is a present lack of methods to analyse samples of interaction networks which fully respect the structure of the data; either one converts observations to graphs, possibly disregarding information, or one models each observation individually. We look to address this gap. To this end, we propose a new Bayesian modelling framework. Namely, through use of distance metrics between observations, we construct families of models via location and scale parameters, akin to a Gaussian distribution. The location parameter, itself an interaction network, admits an interpretation analogous to the mean, whilst the scale parameter can be seen as a notion of variance or precision. Conducting inference of these parameters thus provides a reasoned approach to answering questions (a) and (b). Our methodology is intended to work with any distance metric of the practitioners choosing, leading to a flexible framework which can be tailored to suit different questions of interest. We supplement our methodology with specialised Markov chain Monte Carlo (MCMC) algorithms which facilitate parameter inference. In particular, by merging the recently proposed involutive MCMC (iMCMC) framework of Neklyudov et al. (2020) with the exchange algorithm of Murray et al. (2006), we are able to not only circumvent issues pertaining to doubly-intractable posteriors, but also navigate a multi-dimensional discrete parameter space.

The remainder of this paper will be structured as follows. In Section 2, we provide background details regarding the data structure and notation. Then, in Section 3 we formally introduce our methodology, stating proposed models and providing examples of distance metrics. In Section 4, we outline our Bayesian approach to inference, discussing prior specification, stating our assumed hierarchical model and detailing our proposed MCMC scheme. In Section 5, we detail simulation studies undertaken to confirm the efficacy of our method-

ology and posterior inference scheme, whilst in Section 6 we illustrate its applicability via an analysis of the Foursquare check-in data. We finalise with conclusions and discussion in Section 7.

2. Data Representation

Due to the nature of the data, observed paths often arrive in a known order and, depending on the questions one would like to ask, it may or may not be desirable to encode this in our representation. For example, consider question (a) of Section 1. When looking to find an average, do we want to take the observed order into account, finding an average *sequence* of paths? Or do we want to disregard the order information, and instead find an average *set* of paths? To cover both situations, we propose two data representations which will be used within our framework. Namely, what we call *interaction sequences* and *interaction multisets*, covering the ordered and un-ordered cases respectively.

Adopting the vernacular of network data analysis, we formally denote the collection of entities under consideration via a *vertex set* \mathcal{V} , assumed to be some discrete set (typically the set of integers $\mathcal{V} = \{1, \dots, V\}$) where we let $V = |\mathcal{V}|$ denote the number of vertices. For example, in the Foursquare check-in data (Figure 1), \mathcal{V} would represent the venue categories. An interaction sequence will be denoted as follows

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

where the \mathcal{I}_i will be referred to as *interactions*. We consider the case where interactions are represented via *paths*, that is

$$\mathcal{I}_i = (x_{i1}, \dots, x_{in_i})$$

where $x_{ij} \in \mathcal{V}$, and thus \mathcal{S} is a sequence of paths. Returning to the Foursquare example, \mathcal{I}_i would denote a single day of check-ins, where this user started at a venue of category x_{i1} then moved to category x_{i2} and so on, with \mathcal{S} denoting all the observed check-ins of a single user over some fixed time period. Furthermore, the ordering of interactions reflects the order in which they were observed, for example, \mathcal{I}_1 appears before \mathcal{I}_2 and so on.

When the order of interaction arrival is not of interest, we instead represent the data via an interaction *multiset*. A multiset is a set with multiplicities, that is, a set where elements can occur more than once, and is the natural order-invariant generalisation of a sequence. An interaction multiset will be denoted as follows

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

where the $\{\}$ parenthesis signify this is a multiset, where \mathcal{I}_i similarly denote paths. For example, regarding the Foursquare data, this would simply represent the collection of observed check-ins for a single user, with the order of interactions as written above implying nothing with regards to the order of interaction arrival, for example, \mathcal{I}_1 was not necessarily observed before \mathcal{I}_2 . We note this representation has strong similarities with that used in Crane and Dempsey (2018). In particular, what is defined therein as an interaction network can be seen as a countably infinite interaction multiset.

In this work, we consider the case where *multiple* interaction sequences or multisets are observed. For example, in the Foursquare dataset we have check-in information on not one

but many users, and thus observe a sample of interaction sequences or multisets, one for each user. Representing the i th observation via a interaction sequence $\mathcal{S}^{(i)}$ or multiset $\mathcal{E}^{(i)}$, and letting n denote the sample size, we therefore observe

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

where the choice of representation depends on ones interest in order. Our methodology will provide a means to analyse such samples of data.

We finish this subsection by discussing aggregation. Both multisets and sequences of paths can be aggregated to form graphs, and use of any currently proposed multiple-network methodology would actually necessitate this as a pre-processing step. A *graph* $\mathcal{G} = (\mathcal{E}_{\mathcal{G}}, \mathcal{V}_{\mathcal{G}})$ consists of a set $\mathcal{V}_{\mathcal{G}}$ of *vertices* and a set $\mathcal{E}_{\mathcal{G}}$ of *edges* where $e = (i, j) \in \mathcal{E}_{\mathcal{G}}$ if there is an edge from vertex $i \in \mathcal{V}_{\mathcal{G}}$ to $j \in \mathcal{V}_{\mathcal{G}}$. Graphs can be un-directed, where $(i, j) \in \mathcal{E}_{\mathcal{G}} \iff (j, i) \in \mathcal{E}_{\mathcal{G}}$, or they can be directed, where $(i, j) \in \mathcal{E}_{\mathcal{G}}$ need not imply $(j, i) \in \mathcal{E}_{\mathcal{G}}$ (and *vice versa*). Any interaction sequence \mathcal{S} over vertices \mathcal{V} can be aggregated to form a directed graph $\mathcal{G}_{\mathcal{S}} = (\mathcal{E}_{\mathcal{S}}, \mathcal{V})$ as follows: let $(i, j) \in \mathcal{E}_{\mathcal{S}}$ if the traversal from vertex i to vertex j was observed at least once in \mathcal{S} , that is, if $x_{kl} = i$ and $x_{k(l+1)} = j$ for some $1 \leq k \leq N$ and $1 \leq l \leq n_k$.

Within an interaction sequence, a traversal between two vertices may occur more than once. Thus we can also aggregate to form a so-called *multigraph*, where edges can appear any number of times. We denote a multigraph similarly via $\mathcal{G} = (\mathcal{E}_{\mathcal{G}}, \mathcal{V}_{\mathcal{G}})$, where in this case $\mathcal{E}_{\mathcal{G}}$ is a *multiset* of edges, that is, any given edge $(i, j) \in \mathcal{V} \times \mathcal{V}$ can appear in $\mathcal{E}_{\mathcal{G}}$ multiple times. Any interaction sequence \mathcal{S} over vertices \mathcal{V} can be aggregated to form a directed multigraph $\mathcal{G}_{\mathcal{S}} = (\mathcal{E}_{\mathcal{S}}, \mathcal{V})$ by including the edge (i, j) each time the traversal from vertex i to vertex j is observed, or more formally one can let

$$\mathcal{E}_{\mathcal{S}} = \{(x_{kl}, x_{k(l+1)}) : 1 \leq k \leq N, 1 \leq l \leq (n_k - 1)\},$$

for example, an aggregate multigraph for the Foursquare check-in data of a single user can be seen in Figure 1.

One can also aggregate an interaction multiset \mathcal{E} over vertex set \mathcal{V} to form a graph or multigraph in the same manner. Suppose \mathcal{S} is an interaction sequence obtained by placing the paths of \mathcal{E} in arbitrary order, then we let

$$\mathcal{G}_{\mathcal{E}} = \mathcal{G}_{\mathcal{S}} = (\mathcal{E}_{\mathcal{S}}, \mathcal{V}),$$

which applies equally to the definition of the graph or multigraph, where in the former case $\mathcal{E}_{\mathcal{S}}$ will be a set of edges, whilst in the latter it will be a multiset of edges.

Observe that aggregation is not injective, that is, one may have $\mathcal{S} \neq \mathcal{S}'$ (respectively $\mathcal{E} \neq \mathcal{E}'$) whilst $\mathcal{G}_{\mathcal{S}} = \mathcal{G}_{\mathcal{S}'}$ (respectively $\mathcal{G}_{\mathcal{E}} = \mathcal{G}_{\mathcal{E}'}$). For interaction sequences, a trivial case would be a re-ordering of paths. This, however, is not the only example, and there can instead be interaction sequences or multisets which are structurally dissimilar but nonetheless have equivalent aggregate graphs. In this way, aggregation incurs a loss of information that will be avoided with our methodology.

3. Metric-Based Interaction-Network Models

We now introduce our proposed models for interaction sequences and multisets respectively. The core idea behind both is an assumption that observed data points are ‘noisy’ realisations

of some (unknown) ground truth, where quantification of this noise is facilitated via a pre-specified distance metric. Equivalently, they can be seen as Gaussian-like distributions over their respective discrete metric spaces, controlled by a location parameter, itself an interaction sequence or multiset, and a real-valued scale parameter.

3.1 Interaction-Sequence Models

We let \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 A. Towards eliciting a probability distribution over \mathcal{S}^* , we first endow it with a distance metric $d_S : \mathcal{S}^* \times \mathcal{S}^* \rightarrow \mathbb{R}_+$, defining a metric space (\mathcal{S}^*, d_S) . We then select an element of the space $\mathcal{S}^m \in \mathcal{S}^*$, referred to as the *mode*, upon which to center the distribution, before choosing $\gamma > 0$, referred to as the *diffusion*, controlling the concentration of probability mass in \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 1 (Spherical Interaction Sequence Family) *For a given metric $d_S(\cdot, \cdot)$ on \mathcal{S}^* , a mode $\mathcal{S}^m \in \mathcal{S}^*$ and a diffusion parameter $\gamma > 0$, the probability of observing \mathcal{S} is given by*

$$p(\mathcal{S} | \mathcal{S}^m, \gamma) \propto \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\}, \quad (1)$$

and we write

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

if we assume \mathcal{S} was sampled via (1). This we refer to as the *Spherical Interaction Sequence (SIS) family of probability distributions over \mathcal{S}^* with parameters \mathcal{S}^m and γ* .

3.2 Interaction-Multiset Models

For interaction multisets, the family of models we propose are more-or-less a carbon copy of the SIS models, albeit with slightly different notation. In this case, \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 A). Again, we endow \mathcal{E}^* with a distance metric $d_E : \mathcal{E}^* \times \mathcal{E}^* \rightarrow \mathbb{R}_+$, similarly defining a metric space (\mathcal{E}^*, d_E) . We then construct a distribution over \mathcal{E}^* via location and scale in exactly the same way, where in this case our location parameter will be an interaction *multiset* $\mathcal{E}^m \in \mathcal{E}^*$. The resultant family of probability distributions is defined as follows.

Definition 2 (Spherical Interaction Multiset Family) *Given a metric $d_E(\cdot, \cdot)$ on \mathcal{E}^* , a mode $\mathcal{E}^m \in \mathcal{E}^*$, and a diffusion parameter $\gamma > 0$, the probability of observing \mathcal{E} is given by*

$$p(\mathcal{E} | \mathcal{E}^m, \gamma) \propto \exp\{-\gamma d_E(\mathcal{E}, \mathcal{E}^m)\}, \quad (2)$$

and we write

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

if we assume \mathcal{E} was sampled via (2). This we refer to as the *Spherical Interaction Multiset (SIM) family of probability distributions over \mathcal{E}^* with parameters \mathcal{E}^m and γ* .

Note that in (1) and (2) 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.

3.3 Properties of Interaction-Network Models

We now highlight some key aspects of both model families. Being almost identical, these will be shared between the two, and thus for brevity we here provide details regarding the interaction-sequence models only, noting that the same properties will hold for the multiset models, up to a change of notation.

Firstly, distributions of the form (1) agree with the intuition of location and scale. In particular, for fixed $d_S(\cdot, \cdot)$, an observation \mathcal{S} has the highest probability when $\mathcal{S} = \mathcal{S}^m$, and as \mathcal{S} moves further away from \mathcal{S}^m , that is, as $d_S(\mathcal{S}, \mathcal{S}^m)$ increases, we see a decay in probability. Thus, \mathcal{S}^m indeed represents the mode of the distribution and can be seen as controlling the location of probability within \mathcal{S}^* . Furthermore, the rate at which the probability decreases with respect to distance is controlled by γ , with larger values leading to a faster decrease, implying a greater concentration of probability mass about the mode. In this way, γ is analogous to the inverse of the variance, often referred to as the *precision*.

This latter aspect, the control of variance by γ , can be formalised as in Lunagomez et al. (2021) via the *entropy*, defined to be

$$H(\mathcal{S}^m, \gamma) := -\mathbb{E}[\log p(\mathcal{S} | \mathcal{S}^m, \gamma)], \quad (3)$$

which can be seen to quantify 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 \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 (Supplement S1). Consequently, one can say γ controls the variance of the distribution, as formalised via the entropy.

The distribution as stated in Definition 1 is normalised as follows

$$p(\mathcal{S} | \mathcal{S}^m, \gamma) = Z(\mathcal{S}^m, \gamma)^{-1} \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} \quad (4)$$

where

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

is the normalising constant, often referred to as the *partition function*. In general, 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, with \mathcal{S}^* being infinite, there is no guarantee that (3.3) will even exist for a given γ . Consequently, for some parameterisations (4) may be an improper distribution. Pragmatically, this can cause issues when sampling from these models, whereby one can observe a divergence in dimension for low values of γ . However, this will generally not be an issue provided one chooses γ sufficiently large. In any case, it can be helpful to constrain the sample space to be finite to ensure existence of normalising constants, details of which can be found in Appendix A.1.

We conclude this section by highlighting similarities of our models with those seen in other applications. Firstly, as already touched upon, both families are a direct extension of the spherical network family (SNF) proposed by Lunagomez et al. (2021) for graphs. Also, our assumption that observed data points are ‘noisy’ realisations of an unknown ground truth is very much in keeping with the measurement error models for graphs (Le et al., 2018; Peixoto, 2018; Newman, 2018). Finally, there are connections with the complex Watson distribution (Mardia and Dryden, 1999), used in the context of statistical shape analysis, and the Mallows model (Vitelli et al., 2018) used for preference learning.

3.4 Distance Metrics

A fundamental aspect of our modelling framework is the use of distance metrics. Not only are these key in facilitating the construction of both model families but they also represent an important modelling choice. In this section, examples of two distances measures usable within our framework will be given for interaction sequences and multisets, respectively.

For interaction sequences, one can consider the so-called *edit distance* (Bolt et al., 2022, Definition 4.1). This admits two interpretations: the minimum cost of transforming one sequence into the other, or the cost of an optimal pairing of paths across the two sequences. It is the latter interpretation that we formally define. Suppose \mathcal{S} and \mathcal{S}' denote two interaction sequences, then a pairing of their paths can be encoded via a *matching*, defined to be a multiset of pairs

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

satisfying certain properties. For example, each interaction of one observation may be paired with at most one of the other and *vice versa*. For comparison of sequences, it makes sense to further constrain the structure of matchings so as to ensure the preservation of order. This is achieved by allowing only so-called *monotone* matchings (see Figure 2).

The edit distance between \mathcal{S} and \mathcal{S}' is now defined as follows

$$d_{\text{Edit}}(\mathcal{S}, \mathcal{S}') := \min_{\mathcal{M} \in \mathcal{M}_m^*} \left\{ \sum_{(\mathcal{I}, \mathcal{I}') \in \mathcal{M}} d_{\mathcal{I}}(\mathcal{I}, \mathcal{I}') + \sum_{\mathcal{I} \in \mathcal{M}_{\mathcal{S}}^c} d_{\mathcal{I}}(\mathcal{I}, \emptyset) + \sum_{\mathcal{I}' \in \mathcal{M}_{\mathcal{S}'}^c} d_{\mathcal{I}'}(\emptyset, \mathcal{I}') \right\} \quad (5)$$

where \mathcal{M}_m^* is the set of monotone matchings between the two sequences; $\mathcal{M}_{\mathcal{S}}$ is the restriction of \mathcal{M} to \mathcal{S} , that is, the set of interactions in \mathcal{S} included in the matching, with $\mathcal{M}_{\mathcal{S}}^c := \mathcal{S} \setminus \mathcal{M}_{\mathcal{S}}$ the interactions of \mathcal{S} *not* included in \mathcal{M} ; $d_{\mathcal{I}}$ is some distance metric between interactions, that is, paths; and \emptyset denotes the empty interaction. That is, we sum the pairwise distances of matched interactions (equivalent to the cost of transforming interactions to one another), and then sum penalisations for unmatched interactions, where an unmatched interaction \mathcal{I} incurs penalty $d_{\mathcal{I}}(\mathcal{I}, \emptyset)$ (equivalent to the cost of inserting or deleting \mathcal{I}). Computation of d_{Edit} can be achieved via dynamic programming (Bolt et al., 2022) at a complexity of $\mathcal{O}(NM)$, where N and M are the lengths of \mathcal{S} and \mathcal{S}' respectively.

For interaction multisets, we consider the *matching distance* (Bolt et al., 2022, Definition 3.1). This has close connections with the edit distance, with the interpretation being more-or-less identical: on the one hand, the minimum cost of editing one multiset to get the other, whilst on the other an optimal matching of interactions across the two observations. The differential factor is an absence of order information, making consideration of monotonicity

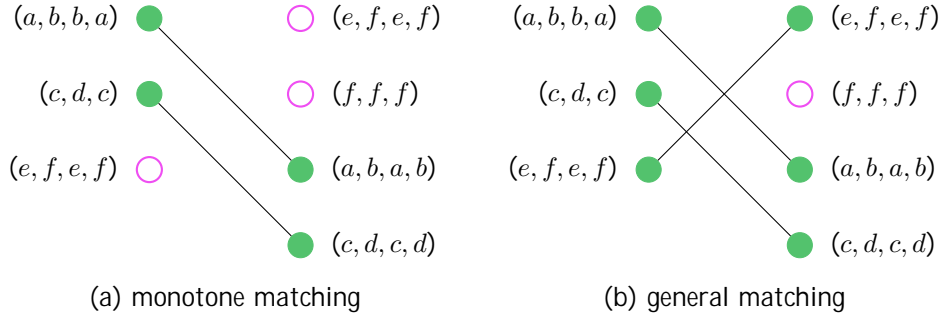


Figure 2: A comparison of matchings. In both (a) and (b) we have two interaction sequences displayed top-down. Here (a) shows a *monotone* matching of interactions, where the order is preserved, whilst (b) shows a general matching, where any two interactions can be paired together.

in the matching unnecessary. Given two interaction multisets \mathcal{E} and \mathcal{E}' , the matching distance is defined as follows

$$d_M(\mathcal{E}, \mathcal{E}') := \min_{\mathcal{M} \in \mathcal{M}^*} \left\{ \sum_{(\mathcal{I}, \mathcal{I}') \in \mathcal{M}} d_I(\mathcal{I}, \mathcal{I}') + \sum_{\mathcal{I} \in \mathcal{M}_{\mathcal{E}}^c} d_I(\mathcal{I}, \cdot) + \sum_{\mathcal{I}' \in \mathcal{M}_{\mathcal{E}'}^c} d_I(\cdot, \mathcal{I}') \right\} \quad (6)$$

where \mathcal{M}^* is the set of matchings between the two multisets (with no monotonicity constraints), whilst $\mathcal{M}_{\mathcal{E}}$ is the restriction of \mathcal{M} to \mathcal{E} , that is, the subset of interactions from \mathcal{E} included in \mathcal{M} , with $\mathcal{M}_{\mathcal{E}}^c := \mathcal{E} \setminus \mathcal{M}_{\mathcal{E}}$ again denoting the interactions not included in the matching \mathcal{M} . Computation of d_M can be achieved via the Hungarian algorithm (Kuhn, 1955), with a complexity of $\mathcal{O}(\max(N, M)^4 + NM)$, where N and M denote the cardinalities of \mathcal{E} and \mathcal{E}' respectively (Bolt et al., 2022).

Use of both d_{Edit} and d_M requires specification of a distance d_I between paths. Here we provide examples of two such distances, namely the longest common subpath (LSP) and longest common subsequence (LCS) distances. The core idea of both is to quantify the dissimilarity of two paths by considering how much they have (or do not have) in common.

For $\mathcal{I} = (x_1, \dots, x_n)$ a *subpath* of \mathcal{I} from index i to j is denoted as follows

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

where $1 \leq i \leq j \leq n$. Given another path $\mathcal{I}' = (y_1, \dots, y_m)$ one can consider finding subpaths that both \mathcal{I} and \mathcal{I}' have in common, that is

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

where $1 \leq i \leq j \leq n$ and $1 \leq k \leq l \leq m$ (Figure 3a). Alternatively, one can consider common subsequences. Assuming that $\mathbf{v} = (v_1, \dots, v_s)$ with $1 \leq v_1 < v_2 < \dots < v_s \leq n$, then the *subsequence* of \mathcal{I} indexed by \mathbf{v} is given by

$$\mathcal{I}_{\mathbf{v}} = (x_{v_1}, \dots, x_{v_s}),$$



Figure 3: A comparison of common subsequences and subpaths. 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 in fact maximal.

and one can similarly consider finding subsequences that both \mathcal{I} and \mathcal{I}' have in common, that is

$$\mathcal{I}_v = \mathcal{I}'_u$$

where $\mathbf{u} = (u_1, \dots, u_r)$ is such that $1 \leq u_1 < u_2 < \dots < u_r \leq m$ and \mathbf{v} is as above (Figure 3b).

Towards defining a distance, one can consider finding a maximal subpath or subsequence, that is, one such that there exists no other subpath or subsequence of greater length. For example, the common subpaths and subsequences of Figure 3 are maximal. Intuitively, a distance is now defined by counting the number of entries not included in the common subpath (or subsequence) from either \mathcal{I} or \mathcal{I}' .

More formally, the two distances can be defined as follows

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

where δ_{LSP} and δ_{LCS} denote the maximum common subpath and subsequence lengths respectively. This equates to counting the underlined entries of Figure 3, whereby we consider the $n + m$ total entries (of both paths), and then remove the double counting made for the entries of the common subpath or subsequence. As an example, in Figure 3 we have $\delta_{\text{LSP}} = 3$ and $\delta_{\text{LCS}} = 4$, and thus $d_{\text{LSP}}(\mathcal{I}, \mathcal{I}') = 7$ whilst $d_{\text{LCS}}(\mathcal{I}, \mathcal{I}') = 5$.

Computation of these distances reduces to finding the maximum length common subpaths and subsequences, both of which can be achieved via dynamic programming with a complexity $\mathcal{O}(nm)$. Further details on the implementation can be found in Bolt et al. (2022), Appendix C.

3.5 Modelling Choices

A key modelling choice is whether to opt for an SIS or SIM model. This should be framed around one's interest in order information, whereby if one would like to take order into account then an SIS model should be assumed, choosing an SIM model in the converse case. This can be equated with making a form of exchangeability assumption reminiscent of edge-exchangeable network models (Cai et al., 2016; Crane and Dempsey, 2018), where in assuming an SIM model one is considering observations equal up to a permutation of path order as having the same probability of being sampled.

Another important modelling decision is the choice of distance metric. This will imply certain assumptions regarding the structure of noise in the model, in turn altering the interpretations of inference. For example, taking $d_I = d_{\text{LSP}}$ within either the edit or matching distances of Section 3.4 will imply samples drawn from the SIS or SIM models will contain paths which share common subpaths with those of the mode, whilst if one took $d_I = d_{\text{LCS}}$ then samples would instead share common subsequences with the mode. As a consequence, when inferring model parameters, in particular the mode, the interpretation thereof will differ. In particular, in assuming an LSP distance between paths the inferred mode will likely consist of a multiset or sequence of subpaths often seen together in the observed data, whilst if we assume an LCS distance we will instead uncover a multiset or sequence of often-observed subsequences.

4. 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 (Section 3.1) 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 Supplement S4.

4.1 Priors, Hierarchical Model and Posterior

In specifying a prior for the mode we follow Lunagomez et al. (2021) and assume it was itself sampled from an SIS model, that is

$$\mathcal{S}^m \sim \text{SIS}(\mathcal{S}_0, \gamma_0) \tag{7}$$

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

$$\begin{aligned} \mathcal{S}^{(i)} \mid \mathcal{S}^m, \gamma &\sim \text{SIS}(\mathcal{S}^m, \gamma) \quad (\text{for } i = 1, \dots, n) \\ \mathcal{S}^m &\sim \text{SIS}(\mathcal{S}_0, \gamma_0) \\ \gamma &\sim p(\gamma). \end{aligned} \tag{8}$$

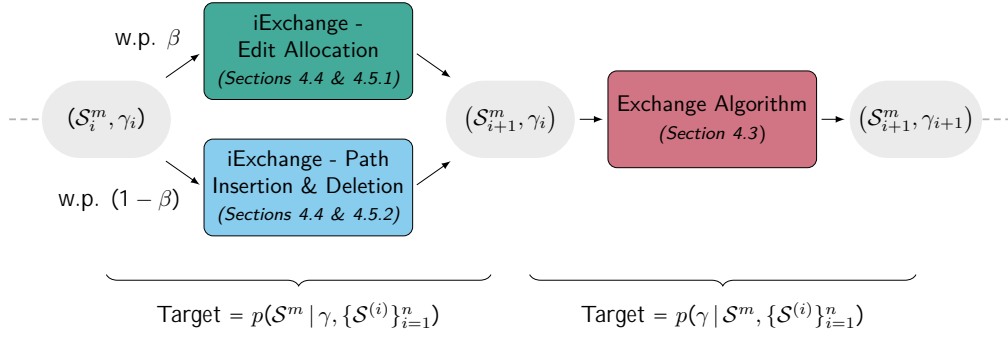


Figure 4: Summary of our MCMC scheme to sample from the SIS posterior. We first update the mode via the iExchange algorithm, doing an edit allocation move with probability β , or a path insertion and deletion move otherwise. We then update the dispersion via the exchange algorithm.

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

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

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

$$\begin{aligned} p(\mathcal{S}^m, \gamma | \{\mathcal{S}^{(i)}\}_{i=1}^n) &\propto p(\{\mathcal{S}^{(i)}\}_{i=1}^n | \mathcal{S}^m, \gamma) p(\mathcal{S}^m) p(\gamma) \\ &\propto Z(\mathcal{S}^m, \gamma)^{-n} \exp \left\{ -\gamma \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) \right\} \\ &\quad \times \exp\{-\gamma_0 d_S(\mathcal{S}^m, \mathcal{S}_0)\} p(\gamma). \end{aligned} \quad (9)$$

4.2 Sampling from the Posterior

To sample from the posterior (9), we use a component-wise MCMC algorithm which alternates between sampling from the two conditional distributions

$$p(\mathcal{S}^m | \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n) \quad \text{and} \quad p(\gamma | \mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n). \quad (10)$$

Since the normalising constant $Z(\mathcal{S}^m, \gamma)$ depends on the parameters of interest this implies (9) is doubly intractable (Murray 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. 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. 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 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

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

3. Evaluate the following probability

$$\alpha(\gamma, \gamma') = \min \left\{ 1, \frac{p(\gamma'|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n)p(\{\mathcal{S}_i^*\}_{i=1}^n|\mathcal{S}^m, \gamma)q(\gamma|\gamma')}{p(\gamma|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n)p(\{\mathcal{S}_i^*\}_{i=1}^n|\mathcal{S}^m, \gamma')q(\gamma'|\gamma)} \right\} \quad (11)$$

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 $p(\gamma|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n)$ 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 (11) and derivation thereof can be found in Supplement S3.1.

4.4 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 $f : \mathcal{S}^* \times \mathcal{U} \rightarrow \mathcal{S}^* \times \mathcal{U}$ and a conditional distribution $q(u|\mathcal{S}^m)$ over auxiliary variables. The function f must also be an *involution*, meaning that it 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

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

4. Evaluate the following probability

$$\alpha(\mathcal{S}^m, [\mathcal{S}^m]') = \min \left\{ 1, \frac{p([\mathcal{S}^m]' | \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n) p(\{\mathcal{S}_i^*\}_{i=1}^n | \mathcal{S}^m, \gamma) q(u' | [\mathcal{S}^m]')}{p(\mathcal{S}^m | \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n) p(\{\mathcal{S}_i^*\}_{i=1}^n | [\mathcal{S}^m]', \gamma) q(u | \mathcal{S}^m)} \right\} \quad (12)$$

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 1, Supplement S2). As shown in Supplement S2, 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 Supplement S2 includes a Jacobian term in the acceptance probability which we do not include above. The reasoning being that since both \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.

4.5 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.

4.5.1 EDIT ALLOCATION

Supposing $\mathcal{S}^m = (\mathcal{I}_1, \dots, \mathcal{I}_N)$ 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 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 5.

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 ,

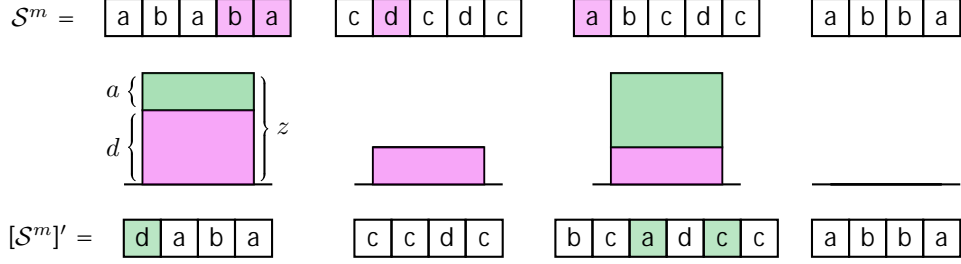


Figure 5: Illustrating the edit allocation move. Shaded entries indicate deletions and insertions, whilst bars visualise allocation of edits to paths. Bar height is proportional to the number of edits allocated to a path z , whilst the green (top) portion of the bar denotes the number of insertions a and the pink (bottom) portion represents the number of deletions d .

denoting this $\mathbf{z} = (z_1, \dots, z_N)$, where $z_i \in \mathbb{Z}_{\geq 0}$ denotes the number of edits allocated to the i th path such that $\sum_{i=1}^N z_i = \delta$. For example, in Figure 5 we have $\delta = 7$ and $\mathbf{z} = (3, 1, 3, 0)$.

Given z_i we edit the i th 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 $d_i \in \{0, \dots, \min(n_i, z_i)\}$ denote the number of deletions, where n_i denotes the length of the i th 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 $[n] = (1, \dots, n)$, we define a subsequence of $[n]$ of size m to be a vector $\mathbf{v} = (v_1, \dots, v_m)$ such that $1 \leq v_1 < v_2 < \dots < v_m \leq n$. Now, we let \mathbf{v}_i be a subsequence of $[n_i]$ of size d_i denoting the entries of \mathcal{I}_i to be deleted, whilst \mathbf{v}'_i is a 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 5 we have $\mathcal{I}_1 = (a, b, a, b, a)$ and $\mathcal{I}'_1 = (d, a, b, a)$ with $\mathbf{v}_1 = (4, 5)$ and $\mathbf{v}'_1 = (1)$ indexing the deletions and insertions respectively. The final step is to specify entries to insert, which we denote $\mathbf{y}_i = (y_{i1}, \dots, y_{ia_i})$ where $y_{ij} \in \mathcal{V}$. For example, in Figure 5 we have $\mathbf{y}_1 = (d)$.

Given the information above, one can enact the specified deletions and insertions, mapping to a proposal $[\mathcal{S}^m]' = (\mathcal{I}'_1, \dots, \mathcal{I}'_N)$. This can be viewed in the iMCMC framework as follows. First, collate all this information into the auxiliary variable $u = (\delta, \mathbf{z}, u_1, \dots, u_N)$ where $u_i = (d_i, \mathbf{v}_i, \mathbf{v}'_i, \mathbf{y}_i)$. 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 $f_1(\mathcal{S}^m, u) = [\mathcal{S}^m]'$. Specification of the second component is more involved, and so we delegate these details to Supplement S3.3. Regarding the auxiliary distribution $q(u|\mathcal{S}^m)$,

we consider the following

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

whilst \mathbf{v}_i and \mathbf{v}'_i are drawn uniformly and the entry insertions \mathbf{y}_i are assumed to be sampled from some general distribution $q(\mathbf{y}_i | \mathcal{I}_i)$, 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 full definition of the involution f , examples of possible insertion distributions $q(\mathbf{y}_i | \mathcal{I}_i)$ and derivations of key terms appearing the acceptance probability (12), can be found in Supplement S3.3.

4.5.2 PATH INSERTION AND DELETION

With this move we look to vary the outer dimension, that is, the number of paths. Similar to Section 4.5.1, we consider doing so by random deletion and insertion. The difference in this case is that we delete and insert whole paths (see Figure 6).

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 6 we have $\varepsilon = 3$, $d = 2$ and $a = 1$. Next, we choose locations of deletions and insertions. In particular, we let \mathbf{v} be a length d subsequence of $[N]$ denoting which paths of \mathcal{S}^m are to be deleted, whilst \mathbf{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 6 we have $\mathbf{v} = (2, 4)$ and $\mathbf{v}' = (3)$. 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 6 we have a single path $\mathcal{I}_1^* = (c, b, b, a)$ which we insert to the third entry.

As in Section 4.5.1, 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 $u = (\varepsilon, d, \mathbf{v}, \mathbf{v}', \mathcal{I}_1^*, \dots, \mathcal{I}_a^*)$ this can similarly be seen as defining the first component of the required involution, with details of the second component found in Supplement S3.4. Regarding sampling of auxiliary variables we consider the following

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

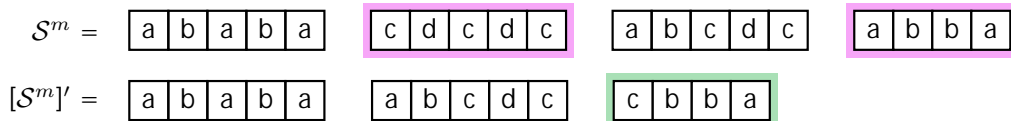


Figure 6: Illustrating path insertion and deletion move, where given current state \mathcal{S}^m the proposal $[\mathcal{S}^m]'$ is obtained by deleting and inserting the highlighted paths.

whilst we sample v and 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 can be found in Supplement S3.4.

4.6 Sampling Auxiliary Data

Both algorithms to target the conditionals outlined in Sections 4.3 and 4.4 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 $\text{SIS}(\mathcal{S}^m, \gamma)$ 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 been 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

$$\alpha(\mathcal{S}, \mathcal{S}') = \min \left\{ 1, \frac{p(\mathcal{S}'|\mathcal{S}^m, \gamma)q(u'|\mathcal{S}')}{p(\mathcal{S}|\mathcal{S}^m, \gamma)q(u|\mathcal{S})} \right\} \quad (13)$$

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 (1). Towards specifying u , $f(u, \mathcal{S})$ and $q(u|\mathcal{S})$, we now recycle the moves of Section 4.4, again mixing these together with some proportion $\beta \in (0, 1)$. Note, as in Section 4.4, 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).

A consequence of using approximate auxiliary samples within the algorithms of Sections 4.3 and 4.4 is the resulting schemes will become 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 an 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.

5. Simulation Studies

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

5.1 Posterior Concentration

If the data were generated by an SIS model at known parameters, then ideally the posterior (9) should concentrate about these as the sample size grows. The goal of the first two simulations is to confirm empirically that such behaviour is observed and to explore what factors might come into play. The high-level approach is the following. Given true mode $\mathcal{S}_{\text{true}}^m$ and dispersion γ_{true} , we draw a sample $\{\mathcal{S}^{(i)}\}_{i=1}^n$ where

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

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

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

where ideally d should be close to zero and $\gamma \approx \gamma_{\text{true}}$. 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 7, top). Intuitively, when the data is more variable it will be harder to discern the true mode $\mathcal{S}_{\text{true}}^m$, and thus we expect d to decrease more slowly for lower values of γ_{true} . Alternatively, as can be seen in Figure 7, 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 $\mathcal{S}_{\text{true}}^m$ fixed. In particular, we considered $\gamma_{\text{true}} = 3.5, 3.7, 4.0, 4.3, 4.6, 4.9$ (highlighted in Figure 7, top) and $n = 25, 50, 75, 100$. The distance we took to be $d_S = d_{\text{Edit}}$ with $d_I = d_{\text{LCS}}$ between paths. We fixed $V = 20$, and constrained the sample space as discussed in Appendix A.1, assuming at most $L = 20$ paths in any observation, with each path being at most length $K = 10$.

The mode $\mathcal{S}_{\text{true}}^m$ of length $N = 10$ we fixed throughout, sampled from the Hollywood model of Crane and Dempsey (2018). In particular, we drew $\mathcal{S}_{\text{true}}^m \sim \text{Hollywood}(\alpha, \theta, \nu)$ 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. This set-up

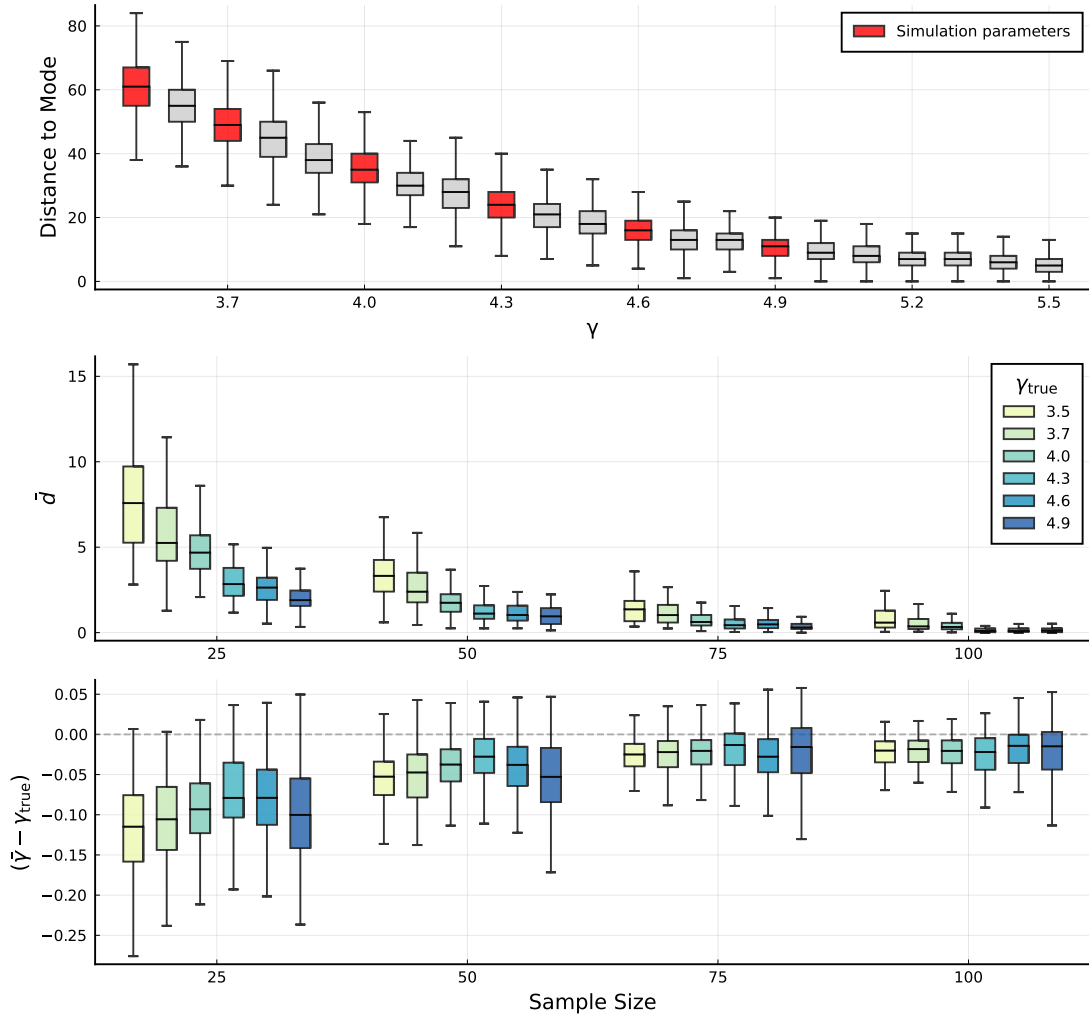


Figure 7: A summary of our first simulation (Section 5.1), where the top plot visualises the scale of the SIS model used therein, in particular, for different values of γ it shows $\{d_S(\mathcal{S}^{(i)}, \mathcal{S}_{\text{true}}^m)\}_{i=1}^{1000}$ where $\mathcal{S}^{(i)} \sim \text{SIS}(\mathcal{S}_{\text{true}}^m, \gamma)$, sampled via the iMCMC scheme of Section 4.6. The remaining two plots summarise simulation outputs for each pair $(\gamma_{\text{true}}, n)$, where the middle shows distributions of d , the average distance to the true mode, whilst the bottom shows $(\hat{\gamma} - \gamma_{\text{true}})$, the bias of the dispersion posterior mean relative to the truth.

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

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

denotes the sample Frechet 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 $\text{SIS}(\mathcal{S}_{\text{true}}^m, \gamma_{\text{true}})$ model, using the iMCMC scheme outlined in Section 4.6, 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 4.2, with a burn-in period of 25,000 and taking a lag of 100 between samples¹ (iii) evaluated summary measures \bar{d} and γ .

We repeated (i)-(iii) 100 times in each case, the results of which are summarised in Figure 7. Consulting the middle plot, we observe that d decreases with n across 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 γ relative to the truth. Note this is expected, since we have used approximate MCMC samples within our component-wise scheme of Section 4.2. 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 γ 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.

5.2 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

1. 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 $\text{SIS}(\mathcal{S}_{\text{true}}^m, \gamma_{\text{true}})$ distribution.

(see Figure 8), 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.

In this simulation, we considered $\alpha = -\alpha$ where $\alpha = 1.35, 0.75, 0.35, 0.12, 0.06, 0.03, 0.01$ (details on how these were chosen can be found in Appendix B.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 $\text{SIS}(\mathcal{S}_{\text{true}}, \gamma_{\text{true}})$ 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 9 summarises the output of 100 repetitions for each pair (α, n) . For each α , we see values for 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 γ , 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.

5.3 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 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 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

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

with $Z(\mathcal{S}^m, \gamma, \mathcal{S}_{-x}) = \sum_{x \in \mathcal{V}} \exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}_x, \mathcal{S}^m)\}$ 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

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

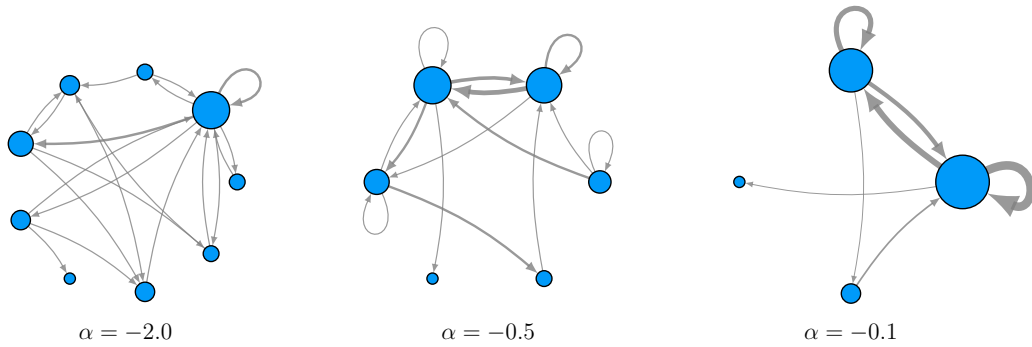


Figure 8: Visualising the role of α in the Hollywood model. Each plot shows an aggregate multigraph \mathcal{G}_S where $S \sim \text{Hollywood}(\alpha, -\alpha V, \nu)$ with $V = 10$, $\nu = \text{TrPoisson}(3, 1, 10)$ and α varying. Edge thickness reflects edge multiplicity, whilst vertex size is proportional to $k_S(v)$.

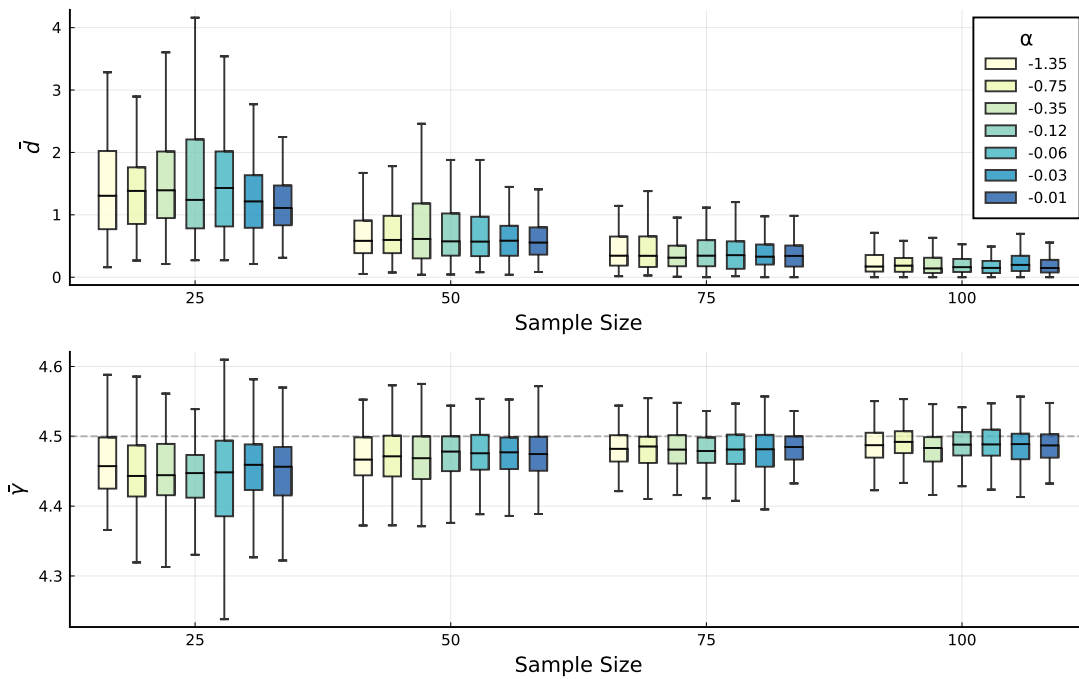


Figure 9: Summary of our second simulation (Section 5.2), where for each pair (α, n) the top subplot shows the distribution of d , the average distance to the true mode, whilst the bottom shows the distribution of γ , the posterior mean dispersion.

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. 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 $\hat{p}(x|\{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x})$ with $p(x|\mathcal{S}^m, \mathcal{S}_{-x})$ as n grows by comparing their predictive accuracy on a common test sample. We adopted the same modelling set-up as Section 5.1, 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.

Now, 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 5.1 and 5.2) (ii) sampled training and testing data $\{\mathcal{S}^{(i)}\}_{i=1}^{n+n_{\text{test}}}$ from an $\text{SIS}(\mathcal{S}_{\text{true}}, \gamma_{\text{true}})$ 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 $i = n+1, \dots, n+n_{\text{test}}$ and for each entry of $\mathcal{S}^{(i)}$ (that is, each entry of each interaction) we assumed it to be missing and predicted its value via the MAP estimate of both $\hat{p}(x|\{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x})$ and $p(x|\mathcal{S}^m, \mathcal{S}_{-x})$, and finally (v) returned the proportion of times each prediction was correct. For (ii) and (iii) we used the same MCMC schemes as previous simulations.

Figure 10 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 accuracy of the posterior predictive is typically lower than the true predictive when the number of training samples is small ($n = 25$), with this difference in accuracy diminishing as n grows, so that by $n = 100$ training samples the accuracy of the posterior predictive is almost indistinguishable from that of the true predictive. With similarity in predictive accuracy serving as a proxy for the coherence of the posterior predictive and the true data generating distribution, it thus appears the posterior predictive is more closely resembling the true data generating distribution as the number of training samples grows, as desired. One can also observe influence from variability as controlled by γ_{true} , where for larger γ_{true} the predictive accuracy is often higher (for both the posterior and the true predictive), as one might expect since the test data in such cases will look more like the true mode and thus more entries thereof will be easily predicted.

6. Real Data Analysis

In this section, we illustrate the applicability of our method by analysing the Foursquare data set of Yang et al. (2015). As mentioned in Section 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

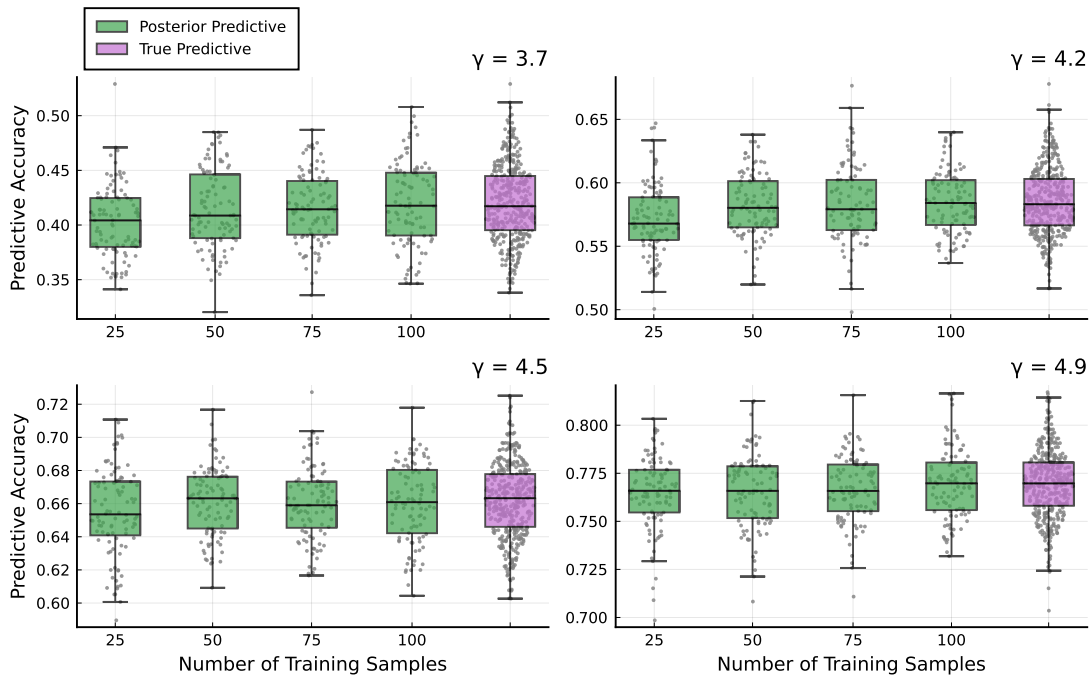


Figure 10: Summary of posterior predictive simulation (Section 5.3). In each subplot, overlaid markers show the observed predictive accuracy being summarised by box-plots, with the four leftmost boxes concerning the posterior predictive at different sample sizes, whilst the rightmost box shows the accuracy of the true predictive across all sample sizes (collated together).

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 SIM model.

6.1 Data Background and Processing

For this analysis we looked at one month of check-in data (over the period from 12 April to 12 May 2012, from the New York and Tokyo data set²), focusing in particular on those in New York. Each check-in event consists of a (i) user id, identifying which user enacted the check-in (anonymised) (ii) venue id, unique to each venue, (iii) venue category, (iv) latitude and longitude, and (v) timestamp.

As discussed in Section 1, we view this as interaction network data by seeing a day of check-ins for a single user as a path through the venue categories. Note the venue category labels have a hierarchical structure, with those given by Yang et al. (2015) being the low-level. For example, the category "Jazz Club" is a subcategory of "Music Venue", which is itself a subcategory of "Arts & Entertainment". In this analysis, we opted to use the highest-level venue categories, so for example here we would use "Arts & Entertainment".

Before proceeding with our analysis, we further filtered the data. 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, from which we extracted a subset of 100 to analyse using a criterion based upon the distance metric used in our model (further details in Supplement S5.1).

6.2 SIM Model Fit

Following data processing, we were left with a sample of multisets $\{\mathcal{E}^{(i)}\}_{i=1}^n$, where each $\mathcal{E}^{(i)} = \{\mathcal{I}_1^{(i)}, \dots, \mathcal{I}_{N^{(i)}}^{(i)}\}$ denotes the data of the i th user, with $\mathcal{I}_j^{(i)}$ denoting a single day of their check-ins. Recalling the inferential questions of interest outlined in Section 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 Supplement S4, 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 discussed in Section 3.5, 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 $\mathcal{E}^m \sim \text{SIM}(\hat{\mathcal{E}}, 3.0)$, with $\hat{\mathcal{E}}$ denoting the sample Frechet mean of the observed data $\{\mathcal{E}^{(i)}\}_{i=1}^n$, whilst we assumed $\gamma \sim \text{Gamma}(5, 1.67)$. Via our MCMC scheme, we then obtained a sample $\{(\mathcal{E}_i^m, \gamma_i)\}_{i=1}^M$, from the posterior

2. <https://sites.google.com/site/yangdingqi/home/foursquare-dataset>.

$p(\mathcal{E}^m, \gamma | \{\mathcal{E}^{(i)}\}_{i=1}^n)$, 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

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

that is, the Frechet 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

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

Figures 11 to 13 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 j th 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 $d_M(\hat{\mathcal{E}}^m, \mathcal{E}^{(1)})$ and $d_M(\hat{\mathcal{E}}^m, \mathcal{E}^{(2)})$, 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 11 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 11 and those of Figures 12 and 13 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.

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 14. To aide interpretation of $\hat{\gamma}$, the right hand plot of Figure 14 visualises the distribution of $d_M(\mathcal{E}, \hat{\mathcal{E}}^m)$ where $\mathcal{E} \sim \operatorname{SIM}(\hat{\mathcal{E}}^m, \gamma)$ for different values of γ , each boxplot summarising 1,000 samples drawn from the respective multiset model via our iMCMC algorithm (Supplement S4.5). 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.

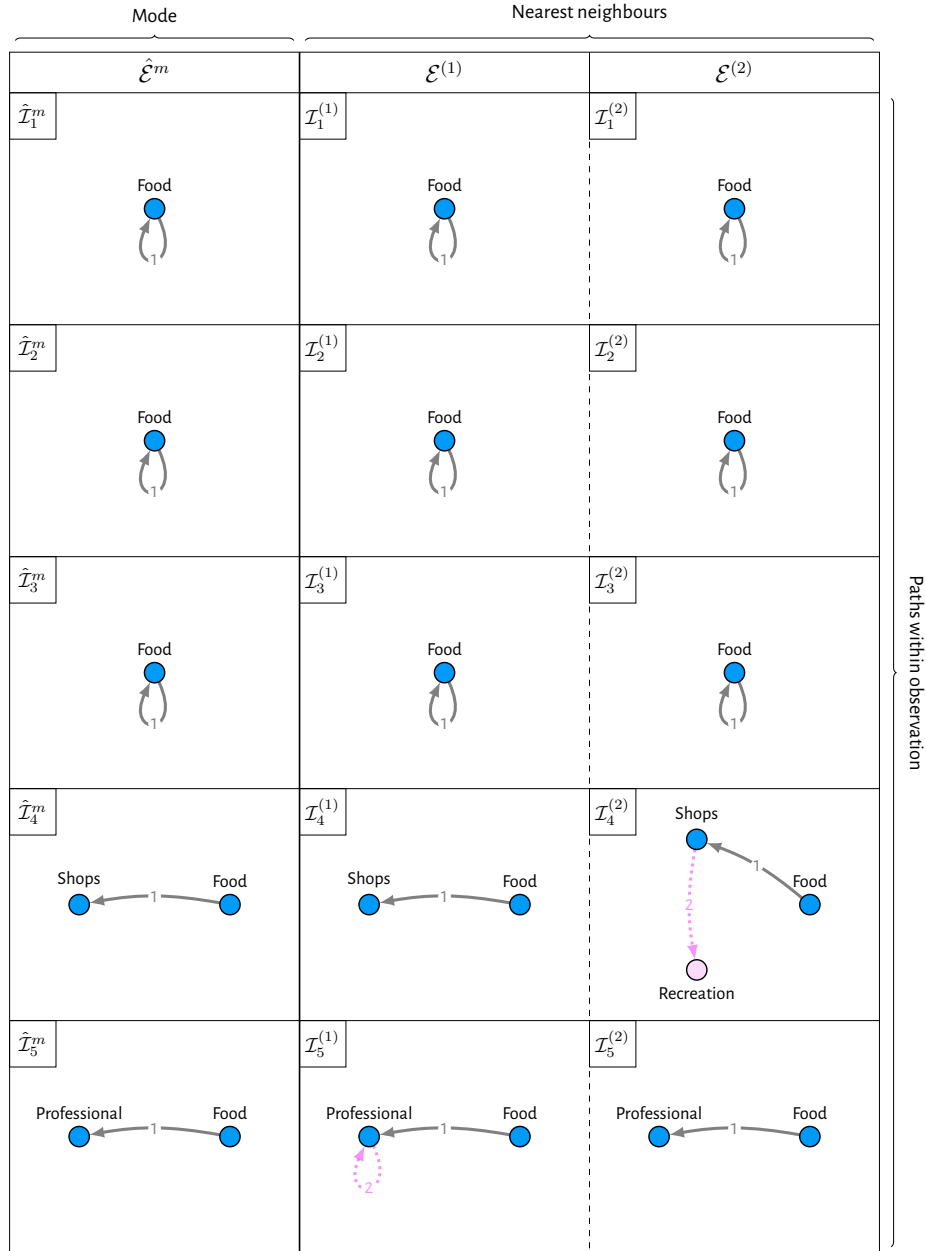


Figure 11: A subset of paths from our point estimate $\hat{\mathcal{E}}^m$ for the Foursquare data, alongside those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$, its two nearest neighbours. Paths are aligned according to the optimal matching found when evaluating $d_M(\hat{\mathcal{E}}^m, \mathcal{E}^{(i)})$ 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 12 and 13.

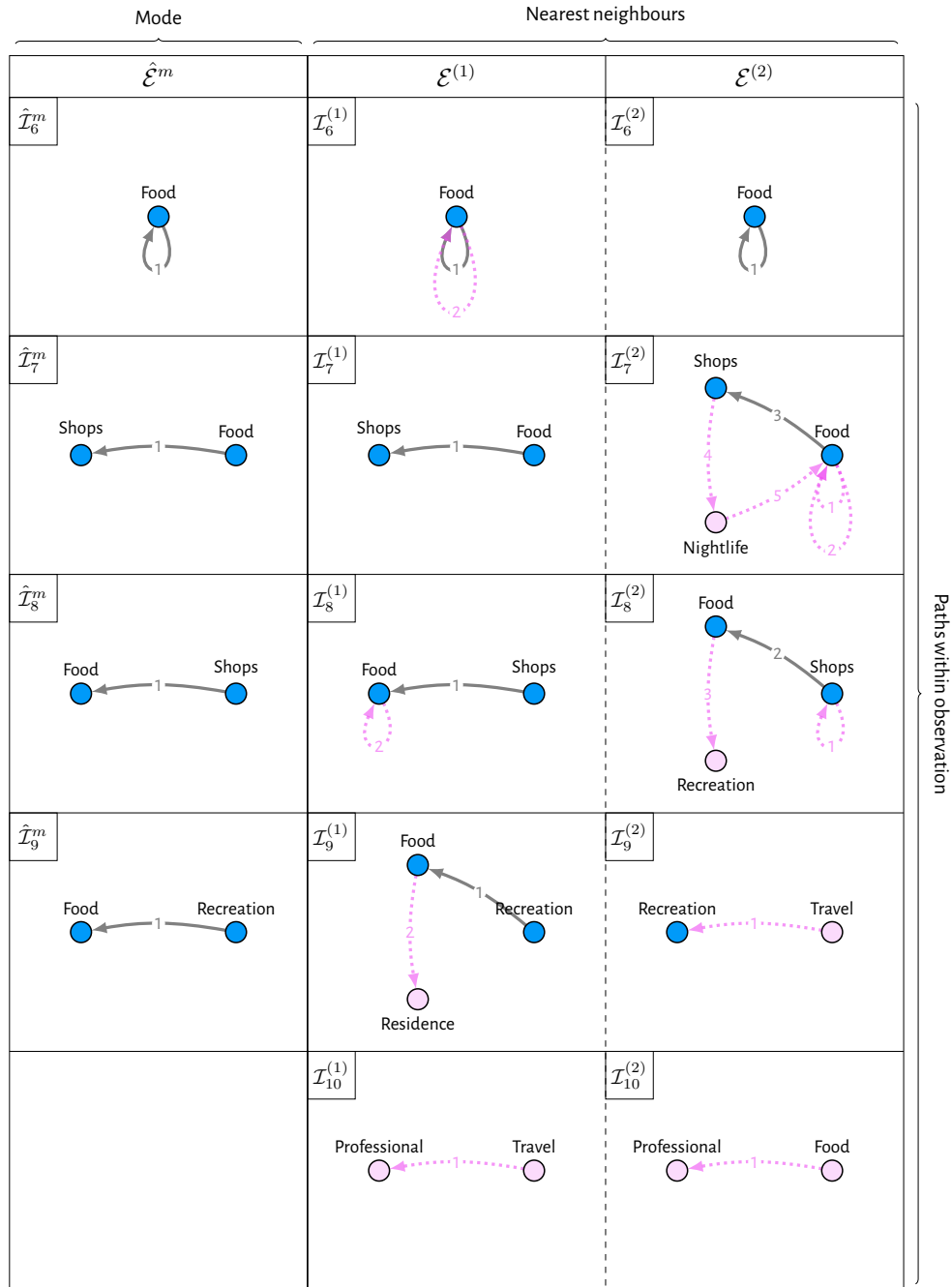


Figure 12: Paths of our point estimate $\hat{\mathcal{E}}^m$ for the Foursquare data, alongside those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$, its two nearest neighbours. The remaining paths can be seen in Figures 11 and 13.

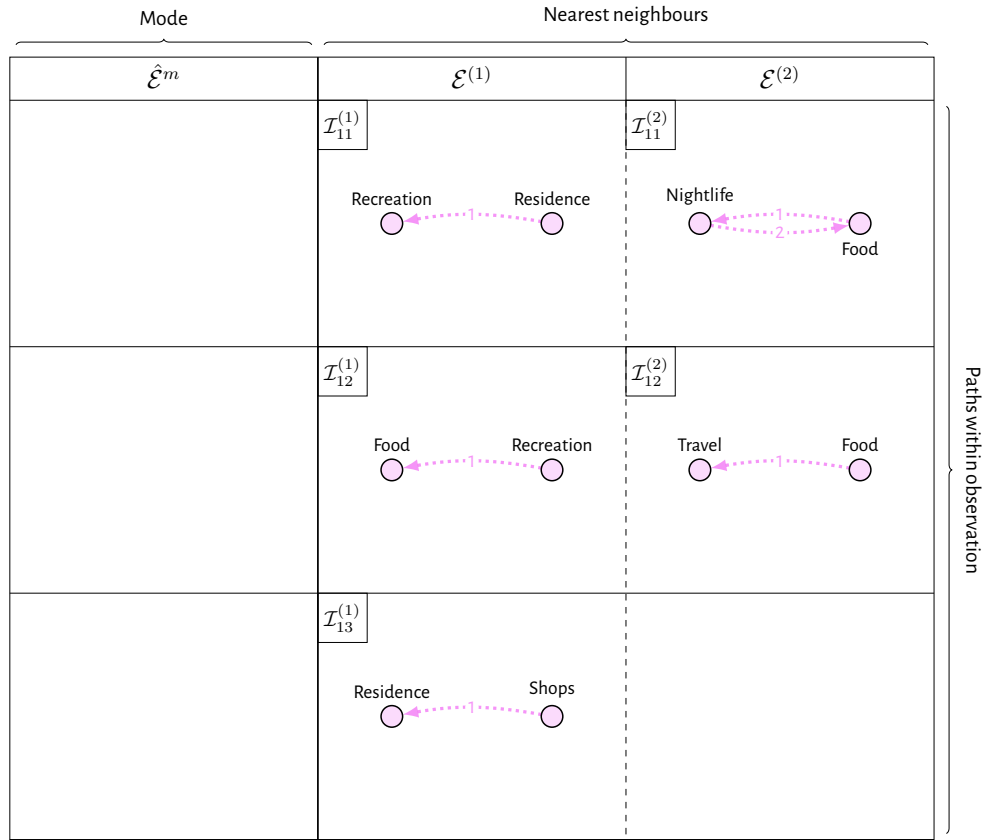


Figure 13: Paths of our point estimate $\hat{\mathcal{E}}^m$ for the Foursquare data, alongside those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$, its two nearest neighbours. The remaining paths can be seen in Figures 11 and 12.

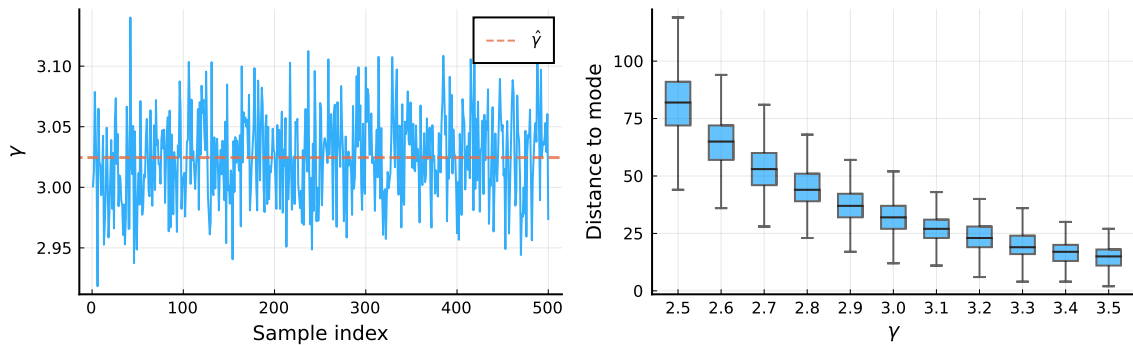


Figure 14: Summary of inference for the dispersion for the Foursquare data. Left shows a trace-plot of the posterior samples $\{\gamma_i\}_{i=1}^m$, whilst the right plot summarises the distribution of distances to the inferred mode for different values of γ , aiding interpretation of our estimate $\hat{\gamma}$.

6.3 Comparison with Graph-Based Inferences

An alternative to applying our methodology is to convert observations to graphs in a pre-processing step, before applying a suitable graph-based method. 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 graph obtained by aggregating the paths of $\mathcal{E}^{(i)}$, as outlined in Section 2. In the same way that our estimate $\hat{\mathcal{E}}^m$ summarises the sample $\{\mathcal{E}^{(i)}\}_{i=1}^n$, one can now consider obtaining a 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. We will consider both cases here. In each instance, the output summary $\hat{\mathcal{G}}$ will be either a graph or multigraph, which we then compare with $\mathcal{G}_{\hat{\mathcal{E}}^m}$, the aggregation of our estimate $\hat{\mathcal{E}}^m$.

To aide this exposition, we will make use of the graph *adjacency matrix*. For a graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, where $\mathcal{V} = \{1, \dots, V\}$, its adjacency matrix $A^{\mathcal{G}} \in \{0, 1\}^{V \times V}$ is the binary matrix with

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

whilst if $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ is a multigraph its adjacency matrix $A^{\mathcal{G}} \in \mathbb{Z}_{\geq 0}^{V \times V}$, is defined by letting $A_{ij}^{\mathcal{G}}$ equal the number of times (i, j) appears in \mathcal{E} . Note there is a one-to-one correspondence between graphs and adjacency matrices, and as such they can be used interchangeably for convenience.

In the case where each $\mathcal{G}^{(i)}$ is a graph, and thus each $A^{\mathcal{G}^{(i)}}$ is a binary matrix, a simple model-free summary is the majority vote, which we denote $\hat{\mathcal{G}}_{\text{MV}}$, where we include an edge 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

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

where A is the real-valued matrix with entries $A_{ij} = \frac{1}{n} \sum_{k=1}^n A_{ij}^{\mathcal{G}^{(k)}}$, that is, the entry-wise average of the observed adjacency matrices. As a model-based alternative, we turn to the centered Erdos-Renyi (CER) model of Lunagomez et al. (2021). Using the notation $\mathcal{G} \sim \text{CER}(\mathcal{G}^m, \alpha)$ when a graph \mathcal{G} was drawn from the CER model with mode \mathcal{G}^m (a graph), and noise parameter $0 \leq \alpha \leq 0.5$, we assumed the following hierarchical model

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

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 an uninformative set-up, letting $\mathcal{G}_0 = \hat{\mathcal{G}}_{\text{MV}}$ 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 Lunagomez 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 Frechet mean

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

where d_{H} denotes the Hamming distance between graphs (Lunagomez et al., 2021; Donnat and Holmes, 2018). Figures 15a and 15b show these two un-weighted 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 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

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

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 models proposed by Lunagomez et al. (2021). Though originally proposed to model graphs, they can be readily extended to handle multigraphs (see Supplement S5.2). Use of the SNF, like our models, requires specification of a distance metric between graphs. We considered taking the absolute difference of edge multiplicities, or alternatively, the adjacency matrix entries, that is

$$d_1(\mathcal{G}, \mathcal{G}') = \sum_{i,j} |A_{ij}^{\mathcal{G}} - A_{ij}^{\mathcal{G}'}|,$$

which can be seen as the generalisation of the Hamming distance to 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

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

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 Frechet 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 Frechet mean to obtain the desired summary

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

Note that the posterior here will be doubly-intractable, necessitating use of a specialised MCMC algorithm. Lunagomez et al. (2021) adopted the algorithm of Müller 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 Supplement S5.2. Visualisations of these two multigraph summaries, $\hat{\mathcal{G}}_{\text{SNF}}$ and $\hat{\mathcal{G}}_{\text{RM}}$, can be seen in Figures 15c and 15d, 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 $\hat{\mathcal{G}}_{\text{SNF}}$, generally involving the node corresponding to food venues. Conversely, the SNF-based estimate $\hat{\mathcal{G}}_{\text{SNF}}$ appears to instead disregard edges which appear in $\hat{\mathcal{G}}_{\text{CER}}$ and $\hat{\mathcal{G}}_{\text{RM}}$. Sitting somewhere in between is the aggregate of our estimate $\hat{\mathcal{G}}_{\hat{\mathcal{E}}^m}$, which appears to have some degree of similarity with each of the graph-based summaries, 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 $\hat{\mathcal{G}}_{\hat{\mathcal{E}}^m}$ is not too dissimilar to those obtained via analysis of the data processed into graphs or multigraphs, then what does one gain by taking our approach? Recalling that we estimated $\hat{\mathcal{E}}^m$, 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 2, 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 15a, where we see the following two edges

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

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 11 to 13), it appears we are in the former case, since no paths therein are of length greater than two.

7. Discussion

In this paper, we have motivated and instantiated the study of multiple interaction-network data. We have proposed a flexible Bayesian modelling framework capable of analysing such data without the need to perform any aggregation of observations. This has been supplemented with specialised MCMC schemes, facilitating inference for our proposed models. Through simulation studies, we have confirmed the efficacy of our approach and inference scheme, whilst the applicability of our methodology has been illustrated by an analysis of Foursquare check-in data, where we illustrated how our methodology can be used to answer inferential questions (a) and (b) posed in Section 1. Moreover, in comparing with

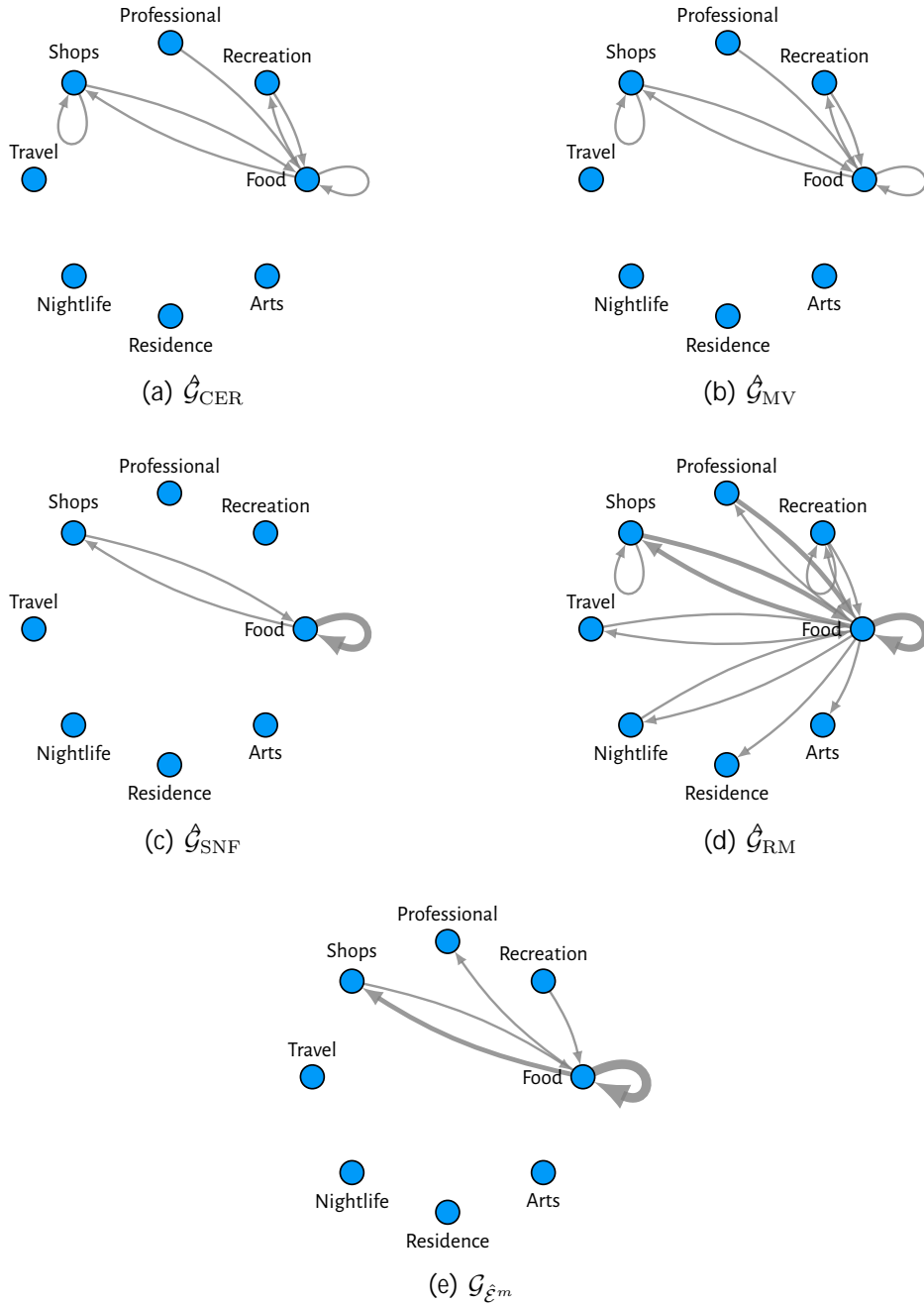


Figure 15: A comparison with graph-based inferences. Here (e) shows $\mathcal{G}_{\hat{\mathcal{E}}^m}$, the aggregate multigraph of our point estimate $\hat{\mathcal{E}}^m$ of Section 6.2, whilst (a)-(d) show alternative inferences obtained via graph-based approaches outlined in Section 6.3. Note that (a) and (b) are graphs, whilst (c)-(e) are multigraphs, with edge thickness proportional to their weight.

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 of our models 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, opening the door to answering question (c) of Section 1. 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 Lunagomez et al. (2021). Finally, 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 slightly modified model structure could be considered, reminiscent of the latent Dirichlet allocation (LDA) model (Blei et al., 2003). Namely, one could assume each observation was drawn from some mixture distribution over paths, 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 any computational benefit would be achieved.

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, it could be interesting to consider the situation where one has access to covariate information at the level of observations. For example, considering the Foursquare data, one might have additional information for each user, such as their line of work or country of residence. Interest might then be in defining a modelling framework which could be invoked to examine for a relationship between covariates and observed data. Such developments would mirror those in the wider literature on multiple-network data, such as work on hypothesis testing (Ginestet et al., 2017; Durante and Dunson, 2018; Ghoshdastidar et al., 2020; Chen et al., 2021).

Appendix A. Sample Spaces

Here we provide extra details regarding the sample spaces for our SIS and SIM models of Definition 1 and 2. Recall that distributions within the defined families constitute distributions over sequences and multisets (of paths) respectively. Thus, their respective sample spaces consist of *all* such objects.

Assuming the vertex set \mathcal{V} is fixed, we first define the space of all paths over \mathcal{V} as follows

$$\mathcal{I}^* := \{(x_1, \dots, x_n) : x_i \in \mathcal{V}, n \geq 1\},$$

subsequently defining the space of interaction sequences \mathcal{S}^* via

$$\mathcal{S}^* := \{(\mathcal{I}_1, \dots, \mathcal{I}_N) : \mathcal{I}_i \in \mathcal{I}^*, N \geq 1\},$$

whilst, with \mathcal{E}_S denoting the multiset obtained from the sequence S by disregarding the order of paths therein, the space of interaction multisets \mathcal{E}^* can be defined via

$$\mathcal{E}^* := \{\mathcal{E}_S : S \in \mathcal{S}^*\}$$

where here we are abusing notation slightly, since we can have $\mathcal{E}_S = \mathcal{E}_{S'}$ for $S \neq S'$ (when equal up to a permutation of interactions), but we just assume such values have been included once and so \mathcal{E}^* is indeed a set.

We note that \mathcal{E}^* also admits another interpretation as a partitioning of \mathcal{S}^* into equivalence classes. To see this, first define an equivalence relation on \mathcal{S}^* via permutations, in particular we write $S \stackrel{p}{\sim} S'$ if there is some permutation σ such that $S' = S^\sigma$, where $S^\sigma = (\mathcal{I}_{\sigma(1)}, \dots, \mathcal{I}_{\sigma(N)})$ is the interaction sequence obtained by permuting the interactions of S via σ . Now, observe that each $\mathcal{E} \in \mathcal{E}^*$ can be seen as an equivalence class of interaction sequences obtained via $\stackrel{p}{\sim}$, that is

$$\mathcal{E} = \{S \in \mathcal{S}^* : S \stackrel{p}{\sim} \mathcal{S}\}$$

where \mathcal{S} denotes some arbitrary ordering of the interactions of \mathcal{E} . Thus, \mathcal{E}^* is in a sense the union of such sets and partitions \mathcal{S}^* .

A.1 Bounding Dimensions

As highlighted Section 3, it can in practice be somewhat easier to work with bounded sample spaces, since in the unbounded case our models are not guaranteed to be proper, which can lead to a divergence of dimension when doing MCMC sampling for some parameterisations (typically when γ is low). This can further cause computational instabilities if one uses such MCMC sampling within other algorithms, as we do within our scheme to sample from the posterior. In this section, we state our notation for the bounded sample spaces.

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

$$\mathcal{I}_K^* := \{(x_1, \dots, x_n) : x_i \in \mathcal{V}, 1 \leq n \leq K\}$$

denote the space of paths up to length K , and then with $L \in \mathbb{Z}_{\geq 1}$ we let

$$\mathcal{S}_{K,L}^* := \{(\mathcal{I}_1, \dots, \mathcal{I}_N) : \mathcal{I}_i \in \mathcal{I}_K^*, 1 \leq N \leq L\},$$

denote the space of interaction sequences with at most L paths of length at most K . The analogous bounded space of interaction multisets is then given by

$$\mathcal{E}_{K,L}^* := \{\mathcal{E}_S : S \in \mathcal{S}_{K,L}^*\}.$$

Appendix B. Simulation Studies

This section contains supporting details for the simulation studies of Section 5. In particular, we discuss how parameters were chosen for the simulation of Section 5.2, and provide a derivation for the posterior predictive approximation used in Section 5.3.

B.1 Posterior Concentration Parameter Choices

Recall that in the simulation of Section 5.2 we re-sampled 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 5.2, 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. Now, 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 $\text{Hollywood}(\alpha, -\alpha V, \nu)$ model, taking ν and V as above, drawing a total of $N = 10$ paths in each case, then (ii) for $i = 1, \dots, n$ 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 16 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 16 by the dashed line, which allowed us to select α (red crosses) providing an even spread of expected degree-distribution 95% quantiles.

B.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 5.3. First, observe that any sample $\{(\mathcal{S}_i^m, \gamma_i)\}_{i=1}^m$ from the posterior implies the following atomic approximation thereof

$$\hat{p}(\mathcal{S}^m, \gamma | \{\mathcal{S}^{(i)}\}_{i=1}^n) = \frac{1}{m} \sum_{i=1}^m \mathbb{1}(\mathcal{S}^m = \mathcal{S}_i^m) \cdot \delta(\gamma - \gamma_i) \quad (14)$$

where $\delta(\cdot)$ is the Dirac delta function.

As in Section 5.3, with \mathcal{S}_x denoting the observation with missing entry filled in to be x , then given some parameters (\mathcal{S}^m, γ) we have the true predictive for x given by

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

with

$$Z(\mathcal{S}^m, \gamma, \mathcal{S}_{-x}) = \sum_{x \in \mathcal{V}} \exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}_x, \mathcal{S}^m)\}.$$

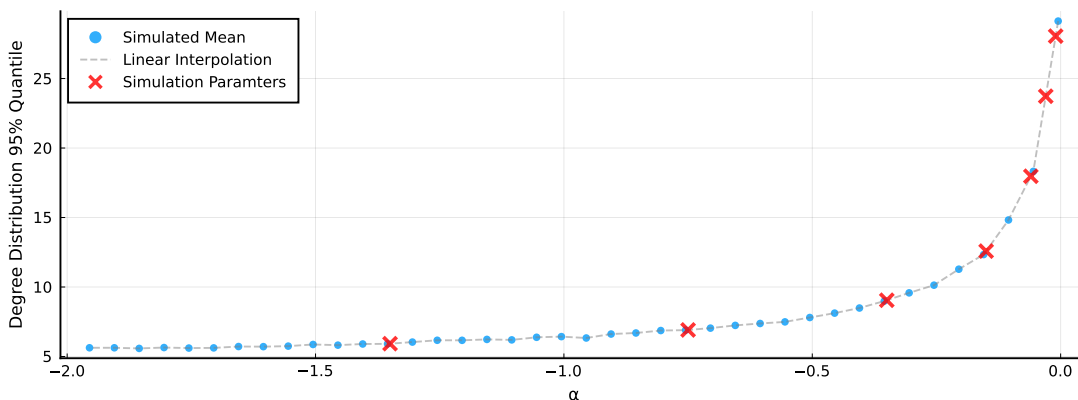


Figure 16: Summary of Hollywood model simulation used to select parameters for simulation of Section 5.2. Plot shows simulated mean degree distribution 95% quantiles for Hollywood($\alpha, -\alpha V, \nu$) model, where $V = 20$, $\nu = \text{TrPoisson}(3, 1, 10)$ and α varies. Via linear interpolation (dashed line), we choose α values (crosses) to get an even spread over the expected degree distribution quantiles.

The posterior predictive is now obtained by averaging with respect to the posterior

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

which we can now approximate by substituting in (14) as follows

$$\begin{aligned} \hat{p}(x|\{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x}) &:= \sum_{\mathcal{S}^m \in \mathcal{S}^*} \int_{\mathbb{R}_+} p(x|\mathcal{S}^m, \gamma, \mathcal{S}_{-x}) \hat{p}(\mathcal{S}^m, \gamma|\{\mathcal{S}^{(i)}\}_{i=1}^n) d\gamma \\ &= \sum_{\mathcal{S}^m \in \mathcal{S}^*} \int_{\mathbb{R}_+} p(x|\mathcal{S}^m, \gamma) \left(\frac{1}{m} \sum_{i=1}^m \mathbb{1}(\mathcal{S}^m = \mathcal{S}_i^m) \delta(\gamma - \gamma_i) \right) d\gamma \\ &= \frac{1}{m} \sum_{i=1}^m p(x|\mathcal{S}_i^m, \gamma_i), \end{aligned}$$

which is exactly as stated in Section 5.3.

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.

References

Jesus Arroyo, Avanti Athreya, Joshua Cape, Guodong Chen, Carey E Priebe, and Joshua T Vogelstein. Inference for multiple heterogeneous networks with a common invariant subspace. *Journal of machine learning research*, 22(142), 2021.

- Timothy E.J. Behrens and Olaf Sporns. Human connectomics. *Current Opinion in Neurobiology*, 22:144{153, 2012.
- David M Blei, Andrew Y Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993{1022, 2003.
- George Bolt, Simon Lunagomez, and Christopher Nemeth. Distances for comparing multi-sets and sequences. *arXiv preprint arXiv:2206.08858*, 2022.
- Diana Cai, Trevor Campbell, and Tamara Broderick. Edge-exchangeable graphs and sparsity. *Advances in Neural Information Processing Systems*, 29, 2016.
- Francois Caron and Emily B Fox. Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(5):1295{1366, 2017.
- Li Chen, Jie Zhou, and Lizhen Lin. Hypothesis testing for populations of networks. *Communications in Statistics-Theory and Methods*, pages 1{24, 2021.
- Jaewon Chung, Eric Bridgeford, Jesus Arroyo, Benjamin D Pedigo, Ali Saad-Eldin, Vivek Gopalakrishnan, Liang Xiang, Carey E Priebe, and Joshua T Vogelstein. Statistical connectomics. *Annual Review of Statistics and Its Application*, 8:463{492, 2021.
- Harry Crane and Walter Dempsey. Edge exchangeable models for interaction networks. *Journal of the American Statistical Association*, 113:1311{1326, 2018.
- Claire Donnat and Susan Holmes. Tracking network dynamics: A survey using graph distances. *Annals of Applied Statistics*, 12:971{1012, 2018.
- Daniele Durante and David B Dunson. Bayesian inference and testing of group differences in brain networks. *Bayesian Analysis*, 13(1):29{58, 2018.
- Daniele Durante, David B Dunson, and Joshua T Vogelstein. Nonparametric Bayes modeling of populations of networks. *Journal of the American Statistical Association*, 112:1516{1530, 2017.
- Ove Frank and David Strauss. Markov graphs. *Journal of the American Statistical Association*, 81(395):832{842, 1986.
- Elahe Ghalebi, Hamidreza Mahyar, Radu Grosu, Graham W Taylor, and Sinead A Williamson. A nonparametric Bayesian model for sparse dynamic multigraphs. *arXiv preprint arXiv:1910.05098*, 2019a.
- Elahe Ghalebi, Hamidreza Mahyar, Radu Grosu, and Sinead Williamson. Dynamic non-parametric edge-clustering model for time-evolving sparse networks. *arXiv preprint arXiv:1905.11724*, 2019b.
- Debarghya Ghoshdastidar, Maurilio Gutzeit, Alexandra Carpentier, and Ulrike Von Luxburg. Two-sample hypothesis testing for inhomogeneous random graphs. *The Annals of Statistics*, 48(4):2208{2229, 2020.

- Cedric E. Ginestet, Jun Li, Prakash Balachandran, Steven Rosenberg, and Eric D. Kolaczyk. Hypothesis testing for network data in functional neuroimaging. *Annals of Applied Statistics*, 11:725{750, 7 2017.
- Isabella Gollini and Thomas Brendan Murphy. Joint modeling of multiple network views. *Journal of Computational and Graphical Statistics*, 25:246{265, 2016.
- Peter D Ho , Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090{1098, 2002.
- Paul W Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33{50, 1981.
- H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83{97, 1955.
- Can M Le, Keith Levin, and Elizaveta Levina. Estimating a network from multiple noisy realizations. *Electronic Journal of Statistics*, 12:4697{4740, 2018.
- Brieuc Lehmann and Simon White. Bayesian exponential random graph models for populations of networks. *arXiv preprint arXiv:2104.05110*, 2021.
- Keith Levin, Avanti Athreya, Minh Tang, Vince Lyzinski, and Carey E Priebe. 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, 2017.
- Faming Liang. A double Metropolis-Hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computation and Simulation*, 80:1007{1022, 2010.
- Simon Lunagomez, So a C. Olhede, and Patrick J. Wolfe. Modeling network populations via graph distances. *Journal of the American Statistical Association*, 116(536):2023{2040, 2021.
- Anastasia Mantziou, Simon Lunagomez, and Robin Mitra. Bayesian model-based clustering for multiple network data. *arXiv preprint arXiv:2107.03431*, 2021.
- K. V. Mardia and I. L. Dryden. The complex Watson distribution and shape analysis. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 61:913{926, 1999.
- James Stephen Marron and Ian L. Dryden. *Object Oriented Data Analysis (1st ed.)*. Chapman and Hall/CRC, 2021.
- Iain Murray, Zoubin Ghahramani, and David J.C. MacKay. Mcmc for doubly-intractable distributions. *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, UAI 2006*, pages 359{366, 2006.

- J. M. Iler, A. N. Pettitt, R. Reeves, and K. K. Berthelsen. An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93: 451{458, 2006.
- Kirill Neklyudov, Max Welling, Evgenii Egorov, and Dmitry Vetrov. Involutive mcmc: A unifying framework. In *International Conference on Machine Learning*, pages 7273{7282. PMLR, 2020.
- Mark EJ Newman. Estimating network structure from unreliable measurements. *Physical Review E*, 98(6):062321, 2018.
- Agnes Martine Nielsen and Daniela Witten. The multiple random dot product graph model. *arXiv preprint arXiv:1811.12172*, 11 2018.
- Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic block-structures. *Journal of the American Statistical Association*, 96(455):1077{1087, 2001.
- Tiago P Peixoto. Reconstructing networks with unknown and heterogeneous errors. *Physical Review X*, 8(4):041011, 2018.
- Tiago P Peixoto and Martin Rosvall. Modelling sequences and temporal networks with dynamic community structures. *Nature Communications*, 8(1):1{12, 2017.
- Perla Reyes and Abel Rodriguez. Stochastic blockmodels for exchangeable collections of networks. *arXiv preprint arXiv:1606.05277*, 2016.
- Ingo Scholtes. When is a network a network? multi-order graphical model selection in pathways and temporal networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1037{1046, 2017.
- Natalie Stanley, Saray Shai, Dane Taylor, and Peter J Mucha. Clustering network layers with the strata multilayer stochastic block model. *IEEE Transactions on Network Science and Engineering*, 3(2):95{105, 2016.
- Tracy M Sweet, Andrew C Thomas, and Brian W Junker. Hierarchical network models for education research: Hierarchical latent space models. *Journal of Educational and Behavioral Statistics*, 38(3):295{318, 2013.
- Tracy M Sweet, Andrew C Thomas, and Brian W Junker. Hierarchical mixed membership stochastic blockmodels for multiple networks and experimental interventions. *Handbook on Mixed Membership Models and their Applications*, pages 463{488, 2014.
- Yee Whye Teh, Michael I Jordan, Matthew J. Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566{1581, 2006.
- Victor Veitch and Daniel M Roy. The class of random graphs arising from exchangeable random measures. *arXiv preprint arXiv:1512.03099*, 2015.
- Valeria Vitelli, Øystein Sørensen, Marta Crispino, Arnaldo Frigessi, and Elja Arjas. Probabilistic preference learning with the Mallows rank model. *Journal of Machine Learning Research*, 18:1{49, 2018.

- Shangsi Wang, Jesus Arroyo, Joshua T Vogelstein, and Carey E Priebe. Joint embedding of graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- Sinead A Williamson. Nonparametric network models for link prediction. *The Journal of Machine Learning Research*, 17(1):7102{7121, 2016.
- Dingqi Yang, Daqing Zhang, Vincent W. Zheng, and Zhiyong Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45:129{142, 2015.
- Fan Yin, Weining Shen, and Carter T Butts. Finite mixtures of ergms for modeling ensembles of networks. *Bayesian Analysis*, 1(1):1{39, 2022.
- Jean-Gabriel Young, Alec Kirkley, and MEJ Newman. Clustering of heterogeneous populations of networks. *Physical Review E*, 105(1):014312, 2022.
- Stephen J Young and Edward R Scheinerman. Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138{149. Springer, 2007.

Supplementary Material

S1. Monotonicity of the Entropy

Here we examine the entropy for the SIS and SIM model families (Definition 1 and 2 respectively), in particular, we confirm it is monotonic with respect to the dispersion. A similar result was shown by Lunagomez et al. (2021) (Proposition 3.3, proved pages 41-43 of Supplementary Material), and as such we follow the reasoning used therein.

For the SIS model, recall the entropy is given by

$$\begin{aligned}
 H(\mathcal{S}^m, \gamma) &= -\mathbb{E}[\log p(\mathcal{S}|\mathcal{S}^m, \gamma)] \\
 &= -\sum_{\mathcal{S} \in \mathcal{S}^*} \log \left(\frac{\exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)\}}{Z(\mathcal{S}^m, \gamma)} \right) \frac{\exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)\}}{Z(\mathcal{S}^m, \gamma)} \\
 &= -\left(\sum_{\mathcal{S} \in \mathcal{S}^*} -\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m) \frac{\exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)\}}{Z(\mathcal{S}^m, \gamma)} \right. \\
 &\quad \left. - \log Z(\mathcal{S}^m, \gamma) \sum_{\mathcal{S} \in \mathcal{S}^*} \frac{\exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)\}}{Z(\mathcal{S}^m, \gamma)} \right) \\
 &= \gamma \left(\sum_{\mathcal{S} \in \mathcal{S}^*} d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m) \frac{\exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)\}}{Z(\mathcal{S}^m, \gamma)} \right) + \log Z(\mathcal{S}^m, \gamma) \\
 &= \gamma \times \mathbb{E}[d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)] + \log Z(\mathcal{S}^m, \gamma).
 \end{aligned}$$

Unfortunately, as was the case for the normalising constant $Z(\mathcal{S}^m, \gamma)$ (Section 3.3), since \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

$$\frac{\partial}{\partial \gamma} H(\mathcal{S}^m, \gamma) = \frac{\partial}{\partial \gamma} \mathbb{E}[d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)] + \mathbb{E}[d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)] + \frac{\partial}{\partial \gamma} \log Z(\mathcal{S}^m, \gamma)$$

where one has

$$\begin{aligned}
 \frac{\partial}{\partial \gamma} \log Z(\mathcal{S}^m, \gamma) &= \frac{\frac{\partial}{\partial \gamma} Z(\mathcal{S}^m, \gamma)}{Z(\mathcal{S}^m, \gamma)} \\
 &= \frac{1}{Z(\mathcal{S}^m, \gamma)} \frac{\partial}{\partial \gamma} \left(\sum_{\mathcal{S} \in \mathcal{S}^*} \exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)\} \right) \\
 &= \frac{1}{Z(\mathcal{S}^m, \gamma)} \sum_{\mathcal{S} \in \mathcal{S}^*} \frac{\partial}{\partial \gamma} \exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)\} \\
 &= \frac{1}{Z(\mathcal{S}^m, \gamma)} \sum_{\mathcal{S} \in \mathcal{S}^*} (-d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)) \exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)\} \\
 &= -\sum_{\mathcal{S} \in \mathcal{S}^*} d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m) \frac{1}{Z(\mathcal{S}^m, \gamma)} \exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)\} \\
 &= -\mathbb{E}[d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)],
 \end{aligned} \tag{15}$$

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

$$\begin{aligned} \frac{\partial}{\partial \gamma} \mathbb{E}[d_S(\mathcal{S}, \mathcal{S}^m)] &= \frac{\partial}{\partial \gamma} \left(\frac{1}{Z(\mathcal{S}^m, \gamma)} \sum_{\mathcal{S} \in \mathcal{S}^*} d_S(\mathcal{S}, \mathcal{S}^m) \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} \right) \\ &= -\frac{\frac{\partial}{\partial \gamma} Z(\mathcal{S}^m, \gamma)}{Z(\mathcal{S}^m, \gamma)^2} \left(\sum_{\mathcal{S} \in \mathcal{S}^*} d_S(\mathcal{S}, \mathcal{S}^m) \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} \right) \\ &\quad - \frac{1}{Z(\mathcal{S}^m, \gamma)} \left(\sum_{\mathcal{S} \in \mathcal{S}^*} d_S(\mathcal{S}, \mathcal{S}^m)^2 \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} \right) \\ &= -\frac{\frac{\partial}{\partial \gamma} Z(\mathcal{S}^m, \gamma)}{Z(\mathcal{S}^m, \gamma)^2} \left(\sum_{\mathcal{S} \in \mathcal{S}^*} d_S(\mathcal{S}, \mathcal{S}^m) \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} \right) \\ &\quad - \frac{1}{Z(\mathcal{S}^m, \gamma)} \left(\sum_{\mathcal{S} \in \mathcal{S}^*} d_S(\mathcal{S}, \mathcal{S}^m)^2 \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} \right) \\ &= -\frac{\frac{\partial}{\partial \gamma} Z(\mathcal{S}^m, \gamma)}{Z(\mathcal{S}^m, \gamma)} \mathbb{E}[d_S(\mathcal{S}, \mathcal{S}^m)] - \mathbb{E}[d_S(\mathcal{S}, \mathcal{S}^m)^2] \end{aligned} \quad (16)$$

$$= \mathbb{E}[d_S(\mathcal{S}, \mathcal{S}^m)]^2 - \mathbb{E}[d_S(\mathcal{S}, \mathcal{S}^m)^2] \quad (17)$$

$$= \text{Var}[d_S(\mathcal{S}, \mathcal{S}^m)],$$

where (17) follows from (16) by applying (15). 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 $\text{Var}[d_S(\mathcal{S}, \mathcal{S}^m)] > 0$. This result, identical to that of Lunagomez et al. (2021), 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 2) by a simple change of notation. For brevity, we do not repeat this here.

S2. The iExchange Algorithm

In this section, we outline the *iExchange* algorithm (Algorithm 1), 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.

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

$$p(\mathbf{x}|\theta) = \frac{\gamma(\mathbf{x}|\theta)}{Z(\theta)} \quad (18)$$

Algorithm 1: Involutional 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 $i = 1, \dots, n$ **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

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

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} \quad (19)$$

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

$$\begin{aligned} \alpha(\theta, \theta') &= \min \left\{ 1, \frac{p(\theta'|\mathbf{x})q(\theta|\theta')}{p(\theta|\mathbf{x})q(\theta'|\theta)} \right\} \\ &= \min \left\{ 1, \frac{\gamma(\mathbf{x}|\theta')/Z(\theta')p(\theta')q(\theta|\theta')}{\gamma(\mathbf{x}|\theta)/Z(\theta)p(\theta)q(\theta'|\theta)} \right\}, \end{aligned} \quad (20)$$

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 (20), 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.

S2.1 Exchange Algorithm

In this section, we give a high-level overview of the exchange algorithm (Algorithm 2), 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 $q(\theta' | \theta)$ 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 $q(\theta' | \theta)$
2. Sample auxiliary data $\mathbf{y} | \theta'$ via $p(\mathbf{y} | \theta')$ of (18) (sample from the model)
3. Evaluate acceptance probability

$$\begin{aligned} \alpha(\theta, \theta') &= \min \left\{ 1, \frac{p(\theta' | \mathbf{x}) q(\theta | \theta') p(\mathbf{y} | \theta)}{p(\theta | \mathbf{x}) q(\theta' | \theta) p(\mathbf{y} | \theta')} \right\} \\ &= \min \left\{ 1, \frac{p(\theta') \gamma(\mathbf{x} | \theta') \gamma(\mathbf{y} | \theta) q(\theta | \theta')}{p(\theta) \gamma(\mathbf{x} | \theta) \gamma(\mathbf{y} | \theta') q(\theta' | \theta)} \right\} \end{aligned} \quad (21)$$

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 2, 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 2: 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 $i = 1, \dots, n$ **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 $\{\theta_i\}_{i=1}^n$

S2.2 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 (19) (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 known as an *involution*. This is a function which serves as its own inverse, that is, if $f : \mathcal{X} \rightarrow \mathcal{X}$ 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 $f : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X} \times \mathcal{U}$ 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

$$\begin{aligned} \alpha((x, u), (x', u')) &= \min \left\{ 1, \frac{p(f(x, u))}{p(x, u)} \left| \frac{\partial f(x, u)}{\partial(x, u)} \right| \right\} \\ &= \min \left\{ 1, \frac{p(x')q(u'|x')}{p(x)q(u|x)} \left| \frac{\partial f(x, u)}{\partial(x, u)} \right| \right\}, \end{aligned}$$

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 re-sampled 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 3.

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 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 Supplement S2.1. Moreover, we define our involution as follows

$$f(\theta, u) = (\theta', (\theta, \mathbf{y})),$$

Algorithm 3: Involutive MCMC (iMCMC)

Input: target density $p(x)$
Input: auxiliary density $q(u|x)$
Input: involution $f(x, u)$
 initialise x ;
for $i = 1, \dots, n$ **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 $\{x_i\}_{i=1}^n$

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 (18). 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 2).

S2.3 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 $\mathfrak{u} = (u, \mathbf{y})$, 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 $g(\theta, \mathfrak{u})$ as follows

$$\begin{aligned}g(\theta, \mathfrak{u}) &= g(\theta, (u, \mathbf{y})) = (f_1(\theta, u), (f_2(\theta, u), \mathbf{y})) \\ &= (\theta', (u', \mathbf{y}))\end{aligned}$$

for which we have

$$\begin{aligned}g(g(\theta, \mathfrak{u})) &= g(\theta', (u', \mathbf{y})) \\ &= (f_1(\theta', u'), (f_2(\theta', u'), \mathbf{y})) \\ &= (\theta, (u, \mathbf{y})) \\ &= (\theta, \mathfrak{u})\end{aligned}$$

that is, g is indeed an involution. Now, as in Section S2.3, drop the inner parenthesis and write $(\theta, \mathfrak{u}) = (\theta, u, \mathbf{y})$. The Jacobian is now given by

$$\frac{\partial g(\theta, \mathfrak{u})}{\partial(\theta, \mathfrak{u})} = \frac{\partial g(\theta, u, \mathbf{y})}{\partial(\theta, u, \mathbf{y})} = \begin{bmatrix} \frac{\partial g_1}{\partial \theta} & \frac{\partial g_1}{\partial u} & \frac{\partial g_1}{\partial \mathbf{y}} \\ \frac{\partial g_2}{\partial \theta} & \frac{\partial g_2}{\partial u} & \frac{\partial g_2}{\partial \mathbf{y}} \\ \frac{\partial g_3}{\partial \theta} & \frac{\partial g_3}{\partial u} & \frac{\partial g_3}{\partial \mathbf{y}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial \mathbf{y}} \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial u} & \frac{\partial f_2}{\partial \mathbf{y}} \\ \frac{\partial f_3}{\partial \theta} & \frac{\partial f_3}{\partial u} & \frac{\partial f_3}{\partial \mathbf{y}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial u} & 0 \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial u} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and taking determinants we get the following

$$\left| \frac{\partial g(\theta, \mathfrak{u})}{\partial(\theta, \mathfrak{u})} \right| = 1 \cdot \left| \begin{bmatrix} \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial u} \end{bmatrix} \right| + 0 + 0 = \left| \frac{\partial f(\theta, u)}{\partial(\theta, u)} \right|.$$

The final element to define is the auxiliary distribution. Given current state θ we consider sampling $\mathfrak{u} = (u, \mathbf{y})$ 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

$$q(\mathbf{u}|\theta) = q(u|\theta)p(\mathbf{y}|\theta').$$

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 $\mathbf{u} = (u, \mathbf{y})$ via $q(\mathbf{u}|\theta)$ as above. We then invoke involution $g(\theta, \mathbf{u}) = (\theta', \mathbf{u}') = (\theta', (u', \mathbf{y}'))$, generating a proposal θ' which we accept with the following probability

$$\begin{aligned} \alpha(\theta, \theta') &= \min \left\{ 1, \frac{p(g(\theta, \mathbf{u}))}{p(\theta, \mathbf{u})} \left| \frac{\partial g(\theta, \mathbf{u})}{\partial(\theta, \mathbf{u})} \right| \right\} \\ &= \min \left\{ 1, \frac{p(\theta'|\mathbf{x})q(\mathbf{u}'|\theta')}{p(\theta|\mathbf{x})q(\mathbf{u}|\theta)} \left| \frac{\partial g(\theta, \mathbf{u})}{\partial(\theta, \mathbf{u})} \right| \right\} \\ &= \min \left\{ 1, \frac{p(\theta'|\mathbf{x})q(u'|\theta')p(\mathbf{y}|\theta)}{p(\theta|\mathbf{x})q(u|\theta)p(\mathbf{y}|\theta')} \left| \frac{\partial f(\theta, u)}{\partial(\theta, u)} \right| \right\} \\ &= \min \left\{ 1, \frac{p(\theta')\gamma(\mathbf{x}|\theta')\gamma(\mathbf{y}|\theta)q(u'|\theta')}{p(\theta)\gamma(\mathbf{x}|\theta)\gamma(\mathbf{y}|\theta)q(u|\theta)} \left| \frac{\partial f(\theta, u)}{\partial(\theta, u)} \right| \right\}, \end{aligned}$$

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 1).

S3. Bayesian Inference: Extra Details

In this section we provide extra details concerning our MCMC scheme for the interaction-sequence models outlined in Section 4, including explicit specification of proposal distributions, involutions and auxiliary distributions, derivations of closed-form acceptance probabilities and pseudocode.

S3.1 Dispersion Conditional

The dispersion conditional can be obtained directly from (9) by conditioning on the mode \mathcal{S}^m , in particular we have

$$p(\gamma|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n) \propto Z(\mathcal{S}^m, \gamma)^{-n} \exp \left\{ -\gamma \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) \right\} p(\gamma). \quad (22)$$

To target (22) we use the exchange algorithm of Murray et al. (2006) (see Supplement S2.1 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 $\gamma > 0$)

$$q(\gamma'|\gamma) = \begin{cases} \frac{1}{2\varepsilon} & \text{if } \gamma' > 0 \text{ and } \gamma + \gamma' > \varepsilon \\ \frac{1}{\varepsilon} & \text{if } \gamma' > 0 \text{ and } \gamma + \gamma' < \varepsilon \\ 0 & \text{if } \gamma' \leq 0. \end{cases} \quad (23)$$

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

$$\mathcal{S}_i^* \sim \text{SIS}(\mathcal{S}^m, \gamma') \quad (\text{for } i = 1, \dots, n),$$

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

$$\alpha(\gamma, \gamma') = \min \{1, H(\gamma, \gamma')\} \quad (24)$$

where

$$\begin{aligned} H(\gamma, \gamma') &= \frac{p(\gamma' | \mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n) p(\{\mathcal{S}_i^*\}_{i=1}^n | \mathcal{S}^m, \gamma) q(\gamma | \gamma')}{p(\gamma | \mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n) p(\{\mathcal{S}_i^*\}_{i=1}^n | \mathcal{S}^m, \gamma') q(\gamma' | \gamma)} \\ &= \exp \left\{ -(\gamma' - \gamma) \left(\sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) - \sum_{i=1}^n d_S(\mathcal{S}_i^*, \mathcal{S}^m) \right) \right\} \frac{p(\gamma')}{p(\gamma)}, \end{aligned}$$

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. This is summarised in Algorithm 5, which details a single accept-reject step for updating the dispersion.

S3.2 Mode Conditional

By conditioning on γ in (9) we get the following form for the mode conditional posterior

$$p(\mathcal{S}^m | \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n) \propto Z(\mathcal{S}^m, \gamma)^{-n} \exp \left\{ -\gamma \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) - \gamma_0 d_S(\mathcal{S}^m, \mathcal{S}_0) \right\}, \quad (25)$$

which as outlined in Section 4.4 we target via the iExchange algorithm (Algorithm 1). For further details on the iExchange algorithm, including justification as an instance of iMCMC, please see Supplement S2.

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

$$\alpha(\mathcal{S}^m, [\mathcal{S}^m]') = \min \{1, H(\mathcal{S}^m, [\mathcal{S}^m]')\} \quad (26)$$

where

$$\begin{aligned}
 H(\mathcal{S}^m, [\mathcal{S}^m]') &= \frac{p([\mathcal{S}^m]' | \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n)}{p(\mathcal{S}^m | \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n)} \frac{p(\{\mathcal{S}_i^*\}_{i=1}^n | \mathcal{S}^m, \gamma)}{p(\{\mathcal{S}_i^*\}_{i=1}^n | [\mathcal{S}^m]', \gamma)} \frac{q(u' | [\mathcal{S}^m]')}{q(u | \mathcal{S}^m)} \\
 &= \exp \left\{ -\gamma \left(\sum_{i=1}^n d_S(\mathcal{S}^{(i)}, [\mathcal{S}^m]') - \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) \right) \right. \\
 &\quad \left. - \gamma \left(\sum_{i=1}^n d_S(\mathcal{S}_i^*, \mathcal{S}^m) - \sum_{i=1}^n d_S(\mathcal{S}_i^*, [\mathcal{S}^m]') \right) \right. \\
 &\quad \left. - \gamma_0 (d_S([\mathcal{S}^m]', \mathcal{S}_0) - d_S(\mathcal{S}^m, \mathcal{S}_0)) \right\} \frac{q(u' | [\mathcal{S}^m]')}{q(u | \mathcal{S}^m)}
 \end{aligned} \tag{27}$$

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.

S3.3 Edit Allocation Move

Supposing that $\mathcal{S}^m = (\mathcal{I}_1, \dots, \mathcal{I}_N)$ denotes the current state, recall that for this move we have an auxiliary variable given by

$$u = (\delta, \mathbf{z}, u_1, \dots, u_N)$$

where (i) δ denotes the total number of edits (entry insertions and deletions), (ii) $\mathbf{z} = (z_1, \dots, z_N)$ denotes the allocation of edits to paths, that is, $z_i \in \mathbb{Z}_{\geq 0}$ is the number of edits allocated to the i th path, where $\sum_{i=1}^N z_i = \delta$, and (iii) $u_i = (d_i, \mathbf{v}_i, \mathbf{v}'_i, \mathbf{y}_i)$ describes the edits to the i th path, where d_i is the number of deletions, \mathbf{v}_i and \mathbf{v}'_i are subsequences indexing entry insertions and deletions and \mathbf{y}_i denotes entries to be inserted.

We now define the involution of this iMCMC move. Writing the required involution as follows

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

as outlined in Section 4.5.1 in enacting the operations parameterised by u we define the first component $f_1(\mathcal{S}^m, u) = [\mathcal{S}^m]' = (\mathcal{I}'_1, \dots, \mathcal{I}'_N)$. The second component we define as follows

$$f_2(\mathcal{S}^m, u) = (\delta, \mathbf{z}, u'_1, \dots, u'_N)$$

where

$$u'_i = (z_i - d_i, \mathbf{v}'_i, \mathbf{v}_i, (\mathcal{I}_i)_{\mathbf{v}_i}) \tag{28}$$

where $(\mathcal{I}_i)_{\mathbf{v}_i} = (x_{i v_1}, \dots, x_{i v_{d_i}})$ is the subsequence of \mathcal{I}_i indexed by $\mathbf{v}_i = (v_1, \dots, v_{d_i})$. On an intuitive level, u'_i parameterises the edits to the i th 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 \mathbf{v}'_i , then insert entries $(\mathcal{I}_i)_{\mathbf{v}_i}$ at locations indexed by \mathbf{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 (28) to u'_i itself takes us back to u_i

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

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

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

and hence

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

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

Turning now to the auxiliary distribution $q(u|\mathcal{S}^m)$, recall the following assumptions stated in Section 4.5.1

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

whilst we sample indexing subsequences \mathbf{v}_i and \mathbf{v}'_i uniformly. Regarding this latter assumption, recall that \mathbf{v}_i is a length d_i (number of deletions) subsequence of $[n_i]$ (n_i is the length of \mathcal{I}_i), whilst \mathbf{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 i th proposed path \mathcal{I}'_i). Thus sampling these uniformly implies

$$q(\mathbf{v}_i | d_i) = \binom{n_i}{d_i}^{-1} \quad q(\mathbf{v}'_i | d_i, z_i) = \binom{m_i}{a_i}^{-1}.$$

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 \mathbf{y}_i was drawn via $q(\mathbf{y}_i | \mathcal{I}_i)$. Together this implies the following closed form for the auxiliary distribution

$$\begin{aligned} q(u|\mathcal{S}^m) &= q(\delta)q(\mathbf{z}|\delta) \prod_{i=1}^N q(d_i)q(\mathbf{v}_i | d_i)q(\mathbf{v}'_i | d_i, z_i)q(\mathbf{y}_i | \mathcal{I}_i) \\ &= \frac{1}{\nu_{\text{ed}}} \left(\frac{1}{N}\right)^\delta \prod_{i=1}^N \frac{1}{\min(n_i, z_i) + 1} \binom{n_i}{d_i}^{-1} \binom{m_i}{a_i}^{-1} q(\mathbf{y}_i | \mathcal{I}_i). \end{aligned} \quad (29)$$

whilst if $([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)$ has been obtained by the involution above we have

$$\begin{aligned} q(u'|[\mathcal{S}^m]') &= q(\delta)q(\mathbf{z}|\delta) \prod_{i=1}^N q(a_i)q(\mathbf{v}'_i | a_i)q(\mathbf{v}_i | a_i, z_i)q((\mathcal{I}_i)_{\mathbf{v}_i} | \mathcal{I}'_i) \\ &= \frac{1}{\nu_{\text{ed}}} \left(\frac{1}{N}\right)^\delta \prod_{i=1}^N \frac{1}{\min(m_i, z_i) + 1} \binom{m_i}{a_i}^{-1} \binom{n_i}{d_i}^{-1} q((\mathcal{I}_i)_{\mathbf{v}_i} | \mathcal{I}'_i). \end{aligned} \quad (30)$$

Taking the ratio of (30) and (29) leads to the following

$$\frac{q(u'|[\mathcal{S}^m]')}{q(u|\mathcal{S}^m)} = \prod_{i=1}^N \frac{\min(n_i, z_i) + 1}{\min(m_i, z_i) + 1} \frac{q((\mathcal{I}_i)_{\mathbf{v}_i} | \mathcal{I}'_i)}{q(\mathbf{y}_i | \mathcal{I}_i)}. \quad (31)$$

which is the key term appearing in the acceptance probability of this move, as seen in (27).

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

$$q(\mathbf{y}_i | \mathcal{I}_i) = \left(\frac{1}{V}\right)^{a_i} \quad (32)$$

which implies

$$\frac{q((\mathcal{I}_i)_{v_i} | \mathcal{I}_i)}{q(\mathbf{y}_i | \mathcal{I}_i)} = \left(\frac{1}{V}\right)^{d_i - a_i} = \left(\frac{1}{V}\right)^{2d_i - z_i} = \left(\frac{1}{V}\right)^{n_i - m_i}$$

any of which can be plugged into (31).

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

$$\mathcal{S}^{(1)}, \dots, \mathcal{S}^{(n)}$$

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 v th 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 $\mathcal{I}_i = (x_{i1}, \dots, x_{in_i})$ denotes the path being edited, with \mathbf{v}_i denoting the subsequence of $[n_i]$ indexing which entries are to be deleted. Introduce the notation \mathbf{v}_i^c for the complement of \mathbf{v}_i , which is the subsequence of $[n_i]$ containing the entries not in \mathbf{v}_i . For example, with $\mathbf{v} = (1, 2, 5) \in [5]$ we would have $\mathbf{v}^c = (3, 4)$. Now, observed that $(\mathcal{I}_i)_{\mathbf{v}_i^c}$ 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 i th path i.i.d. via the following

$$q(y|\mathcal{I}_i) \propto \sum_{v \in (\mathcal{I}_i)_{\mathbf{v}_i^c}} \mathbf{P}_{vy}.$$

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

$$\mathbf{P}_{vv'}^\alpha \propto \mathbf{P}_{vv'} + \alpha$$

which normalises to

$$\mathbf{P}_{vv'}^\alpha = \frac{\mathbf{P}_{vv'} + \alpha}{1 + V\alpha},$$

for which $\mathbf{P}_{vv'}^\alpha \rightarrow 1/V$ as $\alpha \rightarrow \infty$, that is, the rows converge to the uniform distribution over \mathcal{V} . We can now define an analogous insertion distribution

$$q_\alpha(y|\mathcal{I}_i) \propto \sum_{v \in (\mathcal{I}_i)_{\mathbf{v}_i^c}} \mathbf{P}_{vy}^\alpha$$

where as $\alpha \rightarrow \infty$ 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 $\alpha \rightarrow \infty$.

We finish with a note regarding evaluation of (31) for this informed proposal. Supposing that \mathcal{I}'_i is i th path in the proposal $[\mathcal{S}^m]'$ (obtained by deleting d_i entries of \mathcal{I}_i indexed by \mathbf{v}_i , and inserting entries \mathbf{y}_i at locations indexed by \mathbf{v}'_i), then observe we have $(\mathcal{I}_i)_{\mathbf{v}_i^c} = (\mathcal{I}'_i)_{(\mathbf{v}'_i)^c}$ (preserved entries) which thus implies $q_\alpha(y|\mathcal{I}_i) = q_\alpha(y|\mathcal{I}'_i)$. Consequently we can write the following

$$\frac{q_\alpha((\mathcal{I}_i)_{\mathbf{v}_i}|\mathcal{I}'_i)}{q_\alpha(\mathbf{y}_i|\mathcal{I}_i)} = \frac{q_\alpha((\mathcal{I}_i)_{\mathbf{v}_i}|\mathcal{I}_i)}{q_\alpha(\mathbf{y}_i|\mathcal{I}_i)}$$

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 (31).

S3.4 Path Insertion and Deletion Move

Supposing that $\mathcal{S}^m = (\mathcal{I}_1, \dots, \mathcal{I}_N)$ denotes the current state, recall that for this move we have an auxiliary variable given by

$$u = (\varepsilon, d, \mathbf{v}, \mathbf{v}', \mathcal{I}_1^*, \dots, \mathcal{I}_a^*)$$

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) \mathbf{v} and \mathbf{v}' denote subsequences indexing path deletions and insertions respectively, and (iv) $(\mathcal{I}_1^*, \dots, \mathcal{I}_a^*)$ denote the paths to be inserted.

We now define the involution of this iMCMC move. As outlined in Section 4.5.2, if we decompose the the 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 enacting the path insertions and deletions parameterised by u defines the first component $f_2(\mathcal{S}^m, u) = [\mathcal{S}^m]'$. The second component we define as follows

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

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 \mathbf{v}' , before inserting the paths $(\mathcal{I}_{v_1}, \dots, \mathcal{I}_{v_d})$ (of \mathcal{S}^m) into locations indexed by \mathbf{v} . As such, we have the following

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

furthermore, reapplying the second component just defined leads to

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

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 the auxiliary distribution $q(u|\mathcal{S}^m)$, recall the following assumptions stated in Section 4.5.2

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

whilst we sample indexing subsequences \mathbf{v} and \mathbf{v}' uniformly and assume path insertions are drawn via some general distribution $q(\mathcal{I}|\mathcal{S}^m)$. In this instance, recall that \mathbf{v} is a subsequence of $[N]$ of size d , whilst \mathbf{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

$$q(\mathbf{v}|d) = \binom{N}{d}^{-1} \quad q(\mathbf{v}'|\varepsilon, d) = \binom{M}{a}^{-1}$$

leading to the following closed form

$$\begin{aligned} q(u|\mathcal{S}^m) &= q(\varepsilon)q(d|\varepsilon)q(\mathbf{v}|d)q(\mathbf{v}'|\varepsilon, d) \prod_{i=1}^a q(\mathcal{I}_i^*|\mathcal{S}^m) \\ &= \frac{1}{\nu_{\text{td}}} \frac{1}{\min(N, \varepsilon) + 1} \binom{N}{d}^{-1} \binom{M}{a}^{-1} \prod_{i=1}^a q(\mathcal{I}_i^*|\mathcal{S}^m) \end{aligned}$$

whilst, if $([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)$ has been obtained by the involution above, we have

$$\begin{aligned} q(u'|[\mathcal{S}^m]') &= q(\varepsilon)q(a|\varepsilon)q(\mathbf{v}'|a)q(\mathbf{v}|\varepsilon, a) \prod_{i=1}^d q(\mathcal{I}_{v_i}|\mathcal{S}^m) \\ &= \frac{1}{\nu_{\text{td}}} \frac{1}{\min(M, \varepsilon) + 1} \binom{M}{a}^{-1} \binom{N}{d}^{-1} \prod_{i=1}^d q(\mathcal{I}_{v_i}|\mathcal{S}^m). \end{aligned}$$

Taking the ratio of these leads to the following

$$\frac{q(u'|[\mathcal{S}^m]')}{q(u|\mathcal{S}^m)} = \frac{\min(N, \varepsilon) + 1 \prod_{i=1}^d q(\mathcal{I}_{v_i} | [\mathcal{S}^m]')}{\min(M, \varepsilon) + 1 \prod_{i=1}^a q(\mathcal{I}'_{v'_i} | \mathcal{S}^m)}, \quad (33)$$

which can be substituted into (27) to evaluate the acceptance probability of this move (here we again use the fact $\mathcal{I}'_{v'_i} = \mathcal{I}_i^*$).

We normalise 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 $\mathcal{I} = (x_1, \dots, x_m)$ 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 (33).

One can also consider informing entry insertions from observed data. With

$$\mathcal{S}^{(1)}, \dots, \mathcal{S}^{(n)}$$

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 $\mathcal{I} = (x_1, \dots, x_m)$ 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 $\alpha \rightarrow \infty$ we have $q_\alpha(v) \rightarrow 1/V$, and we recover uniform entry insertions.

S3.5 Model Sampling

In this section we provide supporting details regarding our iMCMC algorithm to sample from the SIS models outlined in Section 4.6. Recall that for the SIS model (Definition 1) the (normalised) probability of observing \mathcal{S} is given by

$$p(\mathcal{S} | \mathcal{S}^m, \gamma) = \frac{\exp\{-\gamma d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m)\}}{Z(\mathcal{S}^m, \gamma)}.$$

implying the following closed form for the acceptance probability (13)

$$\alpha(\mathcal{S}, \mathcal{S}') = \min \{1, H(\mathcal{S}, \mathcal{S}')\} \quad (34)$$

where

$$\begin{aligned} H(\mathcal{S}, \mathcal{S}') &= \frac{p(\mathcal{S}'|\mathcal{S}^m, \gamma) q(u'|\mathcal{S}')}{p(\mathcal{S}|\mathcal{S}^m, \gamma) q(u|\mathcal{S})} \\ &= \exp \left\{ -\gamma \left(d_{\mathcal{S}}(\mathcal{S}', \mathcal{S}^m) - d_{\mathcal{S}}(\mathcal{S}, \mathcal{S}^m) \right) \right\} \frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})}, \end{aligned}$$

where the value of $q(u'|\mathcal{S}')/q(u|\mathcal{S})$ will depend on the iMCMC specification.

As mentioned in Section 4.6, we consider re-using the iMCMC moves of our iExchange scheme used to sample from the mode conditional (Supplements S3.3 and S3.4). 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 Supplement S3.3 (replacing \mathcal{S}^m with \mathcal{S} and $[\mathcal{S}^m]'$ with \mathcal{S}') with a uniform entry insertion distribution (32). With $\mathcal{S} = (\mathcal{I}_1, \dots, \mathcal{I}_N)$ the current state, supposing $u = (\delta, z, u_1, \dots, u_N)$ has been sampled via $q(u|\mathcal{S})$ mapping to $(\mathcal{S}', u') = f(\mathcal{S}, u)$ with $\mathcal{S}' = (\mathcal{I}'_1, \dots, \mathcal{I}'_N)$ we will have

$$\frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} = \prod_{i=1}^N \frac{\min(n_i, z_i) + 1}{\min(m_i, z_i) + 1} \left(\frac{1}{V} \right)^{n_i - m_i} \quad (35)$$

where n_i and m_i denote the lengths of the i th path in \mathcal{S}^m and $[\mathcal{S}^m]'$ respectively;

- **Path insertion and deletion** - suppose that u , $f(u, \mathcal{S})$ and $q(u|\mathcal{S})$ are defined as in Supplement S3.4 (again using \mathcal{S} and \mathcal{S}' instead of \mathcal{S}^m and $[\mathcal{S}^m]'$). With $\mathcal{S} = (\mathcal{I}_1, \dots, \mathcal{I}_N)$ the current state, supposing $u = (\varepsilon, d, \mathbf{v}, \mathbf{v}', \mathcal{I}_1^*, \dots, \mathcal{I}_a^*)$ (where $a = \varepsilon - d$) has been sampled via $q(u|\mathcal{S})$ mapping to $(\mathcal{S}', u') = f(\mathcal{S}, u)$ with $\mathcal{S}' = (\mathcal{I}'_1, \dots, \mathcal{I}'_M)$ we will have

$$\frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} = \frac{\min(N, \varepsilon) + 1 \prod_{i=1}^d q(\mathcal{I}_{v_i}|\mathcal{S}')}{\min(M, \varepsilon) + 1 \prod_{i=1}^a q(\mathcal{I}'_{v'_i}|\mathcal{S})}. \quad (36)$$

As mentioned in Section 4.6, 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. Pseudocode of the resultant algorithm can be found in Algorithm 7.

S4. Bayesian Inference for Multiset Models

Here we detail the approach to inference for the interaction-multiset models (Definition 2). This is very similar to the interaction-sequence models outlined in Section 4, 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.

S4.1 Priors, Hierarchical Model and Posterior

To specify priors, we follow Section 4.1 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, γ) 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

$$\begin{aligned} \mathcal{E}^{(i)} | \mathcal{E}^m, \gamma &\sim \text{SIM}(\mathcal{E}^m, \gamma) \quad (\text{for } i = 1, \dots, n) \\ \mathcal{E}^m &\sim \text{SIM}(\mathcal{E}_0, \gamma_0) \\ \gamma &\sim p(\gamma). \end{aligned}$$

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

$$\begin{aligned} p(\mathcal{E}^m, \gamma | \{\mathcal{E}^{(i)}\}_{i=1}^n) &\propto p(\{\mathcal{E}^{(i)}\}_{i=1}^n | \mathcal{E}^m, \gamma) p(\mathcal{E}^m) p(\gamma) \\ &= Z(\mathcal{E}^m, \gamma)^{-n} \exp \left\{ -\gamma \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m) \right\} \\ &\quad \exp\{-\gamma_0 d_E(\mathcal{E}^m, \mathcal{E}_0)\} p(\gamma). \end{aligned} \tag{37}$$

S4.2 Posterior Sampling

As for the interaction-sequence models, we consider sampling from the posterior (37) 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 (37) will persist, making them doubly intractable (Murray 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. 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 and Supplement S3) to target them.

S4.3 Dispersion Conditional

Conditioning on \mathcal{E}^m in (37) 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 and Supplement S3.1 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 Supplement S3.1). 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

$$\mathcal{E}_i^* \sim \text{SIM}(\mathcal{E}^m, \gamma') \quad (\text{for } i = 1, \dots, n),$$

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

$$\alpha(\gamma, \gamma') = \min\{1, H(\gamma, \gamma')\} \tag{38}$$

where

$$\begin{aligned} H(\gamma, \gamma') &= \frac{p(\gamma' | \mathcal{E}^m, \{\mathcal{E}^{(i)}\}_{i=1}^n) p(\{\mathcal{E}_i^*\}_{i=1}^n | \mathcal{E}^m, \gamma') q(\gamma|\gamma')}{p(\gamma | \mathcal{E}^m, \{\mathcal{E}^{(i)}\}_{i=1}^n) p(\{\mathcal{E}_i^*\}_{i=1}^n | \mathcal{E}^m, \gamma') q(\gamma'|\gamma)} \\ &= \exp \left\{ -(\gamma' - \gamma) \left(\sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m) - \sum_{i=1}^n d_E(\mathcal{E}_i^*, \mathcal{E}^m) \right) \right\} \frac{p(\gamma')}{p(\gamma)}, \end{aligned}$$

where, as in Supplement S3.1, normalising constants of the (conditional) posterior and auxiliary data cancel one another out, whilst the proposal density terms cancel due to its symmetry.

S4.4 Mode Conditional

Conditioning on γ in (37) we have the following

$$p(\mathcal{E}^m | \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n) \propto Z(\mathcal{E}^m, \gamma)^{-n} \exp \left\{ -\gamma \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m) - \gamma_0 d_E(\mathcal{E}^m, \mathcal{E}_0) \right\}, \tag{39}$$

which is a distribution over \mathcal{E}^* , that is, the space of multisets. To re-use the iExchange scheme of Section 4.4 we instead need a distribution over the space of interaction sequences \mathcal{S}^* . To this end, we extend (39) to a distribution over interaction sequences.

Consider the general problem of extending some distribution $\pi(\mathcal{E})$ over \mathcal{E}^* to one over \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 A). As such, one can consider assigning equal probability to each unique ordering of \mathcal{E} . In particular, for $\mathcal{S} \in \mathcal{S}^*$ we let

$$\pi(\mathcal{S}) = \frac{1}{A(\mathcal{E})} \pi(\mathcal{E})$$

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 $M \leq N$ unique paths. Without loss of generality label the unique paths 1 to M and let w_i denote the multiplicity of the i th 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

$$A(\mathcal{E}) := \binom{N}{w_1, \dots, w_N} = \frac{N!}{w_1! \cdots w_N!}. \quad (40)$$

Through this reasoning we can extend (39) as follows

$$p(\mathcal{S}^m | \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n) = \frac{1}{A(\mathcal{E}^m)} p(\mathcal{E}^m | \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n) \quad (41)$$

where now $\mathcal{S}^m \in \mathcal{S}^*$ and \mathcal{E}^m is the multiset obtained from \mathcal{S}^m by disregarding the order of paths.

We can now re-use the iExchange algorithm of Section 4.4 and Supplement S3.2 to target (41). However, note the normalising constant appearing in (39), and hence also in (41), 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, 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

$$\alpha(\mathcal{E}^m, [\mathcal{E}^m]') = \min \{1, H(\mathcal{E}^m, [\mathcal{E}^m]')\} \quad (42)$$

where

$$\begin{aligned}
 H(\mathcal{E}^m, [\mathcal{E}^m]') &= \frac{p([\mathcal{S}^m]' | \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)}{p(\mathcal{S}^m | \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)} \frac{p(\{\mathcal{E}_i^*\}_{i=1}^n | \mathcal{E}^m, \gamma)}{p(\{\mathcal{E}_i^*\}_{i=1}^n | [\mathcal{E}^m]', \gamma)} \frac{q(u' | [\mathcal{S}^m]')}{q(u | \mathcal{S}^m)} \\
 &= \frac{\frac{1}{A([\mathcal{E}^m]')} p([\mathcal{E}^m]' | \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)}{\frac{1}{A(\mathcal{E}^m)} p(\mathcal{E}^m | \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)} \frac{p(\{\mathcal{E}_i^*\}_{i=1}^n | \mathcal{E}^m, \gamma)}{p(\{\mathcal{E}_i^*\}_{i=1}^n | [\mathcal{E}^m]', \gamma)} \frac{q(u' | [\mathcal{S}^m]')}{q(u | \mathcal{S}^m)} \\
 &= \frac{A(\mathcal{E}^m)}{A([\mathcal{E}^m]')} \exp \left\{ -\gamma \left(\sum_{i=1}^n d_E(\mathcal{E}^{(i)}, [\mathcal{E}^m]') - \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m) \right) \right. \\
 &\quad \left. - \gamma \left(\sum_{i=1}^n d_E(\mathcal{E}_i^*, \mathcal{E}^m) - \sum_{i=1}^n d_E(\mathcal{E}_i^*, [\mathcal{E}^m]') \right) \right. \\
 &\quad \left. - \gamma_0 (d_E([\mathcal{E}^m]', \mathcal{E}_0) - d_E(\mathcal{E}^m, \mathcal{E}_0)) \right\} \frac{q(u' | [\mathcal{S}^m]')}{q(u | \mathcal{S}^m)} \tag{43}
 \end{aligned}$$

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

$$\frac{A(\mathcal{E}^m)}{A([\mathcal{E}^m]')} = \frac{N! (w_1'! \cdots w_{M'}!)}{M! (w_1! \cdots w_M!)} \tag{44}$$

where N and M are the cardinalities of \mathcal{E}^m and $[\mathcal{E}^m]'$ respectively, w_i is the multiplicity of the i th unique path in \mathcal{E}^m and w_i' is the multiplicity of the i th 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 4.5.1 and Supplement S3.3) and interaction insertion and deletion (Supplement S3.4) 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 10.

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 10, and in this way the whole algorithm can be enacted on vectors of paths, simply interpreting the output samples as multisets of paths.

S4.5 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 4.6), 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 Supplement S4.4 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 4.6 and Supplement S3.5). Recalling that for the SIM model (Definition 2) the (normalised) probability of observing $\mathcal{E} \in \mathcal{E}^*$ 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 $\mathcal{S} \in \mathcal{S}^*$ the following probability

$$p(\mathcal{S}|\mathcal{E}^m, \gamma) = \frac{1}{A(\mathcal{E})} p(\mathcal{E}|\mathcal{E}^m, \gamma) \quad (45)$$

where \mathcal{E} is multiset obtain from \mathcal{S} by disregarding order, and $A(\mathcal{E})$ is as defined in (40), thus defining an extended distribution over \mathcal{S}^* .

We can now target (45) via iMCMC as in Section 4.6. In particular, suppose that one would like to sample from an $\text{SIM}(\mathcal{E}^m, \gamma)$ 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

$$\alpha(\mathcal{E}, \mathcal{E}') = \min \left\{ 1, \frac{p(\mathcal{S}'|\mathcal{E}^m, \gamma) q(u'|\mathcal{S}')}{p(\mathcal{S}|\mathcal{E}^m, \gamma) q(u|\mathcal{S})} \right\} \quad (46)$$

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 4.6 and recycle the edit allocation (Section 4.5.1 and Supplement S3.3) and path insertion/deletion moves (Section 4.5.2 and Supplement S3.4), again mixing them together with proportion $\beta \in (0, 1)$, left as a tuning parameter.

A closed form for (46) can be derived as follows. Writing $\alpha(\mathcal{E}, \mathcal{E}') = \min\{1, H(\mathcal{E}, \mathcal{E}')\}$ we have

$$\begin{aligned} H(\mathcal{E}, \mathcal{E}') &= \frac{p(\mathcal{S}'|\mathcal{E}^m, \gamma) q(u'|\mathcal{S}')}{p(\mathcal{S}|\mathcal{E}^m, \gamma) q(u|\mathcal{S})} \\ &= \frac{\frac{1}{A(\mathcal{E}')} p(\mathcal{E}'|\mathcal{E}^m, \gamma) q(u'|\mathcal{S}')}{\frac{1}{A(\mathcal{E})} p(\mathcal{E}|\mathcal{E}^m, \gamma) q(u|\mathcal{S})} \\ &= \frac{A(\mathcal{E})}{A(\mathcal{E}')} \exp \left\{ -\gamma \left(d_E(\mathcal{E}', \mathcal{E}^m) - d_E(\mathcal{E}, \mathcal{E}^m) \right) \right\} \frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} \end{aligned}$$

where

$$\frac{A(\mathcal{E})}{A(\mathcal{E}')} = \frac{N! (w'_1! \cdots w'_M!)}{M! (w_1! \cdots w_M!)}$$

with N and M the cardinalities of \mathcal{E} and \mathcal{E}' respectively, w_i the multiplicity of the i th unique path in \mathcal{E} and w'_i the multiplicity of the i th unique path in \mathcal{E}' . As when sampling from the interaction-sequence models (Supplement S3.5), the ratio $q(u'|\mathcal{S}')/q(u|\mathcal{S})$ will be move dependent and identical to those appearing in Supplement S3.5, namely (35) for the edit allocation move and (36) for the path insertion/deletion move. The whole procedure to sample from the SIM models is summarised in the pseudocode of Algorithm 11.

Finally we note that, as for the interaction-sequence models, by using approximate as opposed to exact sampling in the exchange-based algorithms of Supplement S4.3 and Supplement S4.4 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 11).

S5. Real Data Analysis

In this section, we provide details supporting the data analysis of Section 6. This includes further details on the data and how it was processed, and extra information regarding the integer-weighted extension of the SNF model (Lunagomez et al., 2021) used in Section 6.3.

S5.1 Foursquare Data Processing

The data analysed in Section 6 was obtained from the New York and Tokyo data set of Yang et al. (2015)³, 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

As outlined in Section 1, 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).

3. https://sites.google.com/site/yangdingqi/home/foursquare-dataset#h.p_ID_46

S5.1.1 VENUE CATEGORY HIERARCHY

As discussed in Section 6, 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. (2015) uses low-level category names ("Tram Station"), whilst we consider the highest-level ("Travel & Transport"). However, we do note that Yang et al. (2015) 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. (2015) appears to be congruent with the former. Using this information we were able to essentially 'll-in' the higher-level category labels for each category name appearing in the data set of Yang et al. (2015), mapping their low-level labels to top-level ones.

S5.1.2 DATA FILTERING

As mentioned in Section 6, we analysed only a subset of 100 data points. This was due to issues caused 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 6.1, 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

$$\mathcal{E}^* = \operatorname{argmin}_{\mathcal{E} \in \{\mathcal{E}^{(i)}\}_{i=1}^n} \left[\sum_{i \in \mathcal{N}_m(\mathcal{E})} d_E(\mathcal{E}, \mathcal{E}^{(i)}) \right],$$

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 itself, namely the matching distance with an LSP distance between paths. Moreover, since the observations were of quite different sizes, it made sense to also normalise this distance. In particular, we used the so-called *Steinhaus transform* (Bolt et al., 2022, Sec. 2) which defines a new distance d_M as follows

$$d_M(\mathcal{E}, \mathcal{E}') = \frac{2d_M(\mathcal{E}, \mathcal{E}')}{d_M(\mathcal{E}, \emptyset) + d_M(\mathcal{E}', \emptyset) + d_M(\mathcal{E}, \mathcal{E}')}$$

where here \emptyset denotes the empty multiset of paths, which functions as our reference element in the space of interaction multisets. Note that $d_M(\mathcal{E}, \emptyset)$ is equivalent to the sum of path

lengths in \mathcal{E} , since each path \mathcal{I} in \mathcal{E} is un-matched and hence penalised by $d_{\text{LSP}}(\mathcal{I}, \cdot)$. It can be shown that this new distance takes values in the unit interval, that is $d_{\text{M}}(\mathcal{E}, \mathcal{E}') \in [0, 1]$ (Bolt et al., 2022), where if $d_{\text{M}}(\mathcal{E}, \mathcal{E}') = 1$ this implies \mathcal{E} and \mathcal{E}' are more-or-less completely different. 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

$$d_{\text{M}}(\mathcal{E}, \mathcal{E}^{(1)}) = 6 \quad d_{\text{M}}(\mathcal{E}, \mathcal{E}^{(2)}) = 6$$

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 evaluates to

$$\begin{aligned} d_{\text{M}}(\mathcal{E}, \mathcal{E}^{(1)}) &= \frac{2 \times 6}{3 + 3 + 6} \\ &= 1 \end{aligned} \quad \begin{aligned} d_{\text{M}}(\mathcal{E}, \mathcal{E}^{(2)}) &= \frac{2 \times 6}{3 + 9 + 6} \\ &= \frac{2}{3} \end{aligned}$$

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 things 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.

S5.2 Multigraph SNF Model

Here we provide extra details regarding the generalisation of the SNF models (Lunagomez et al., 2021) used in Section 6.3. In particular, we extend the SNF to model multigraphs. Let $\mathcal{V} = \{1, \dots, V\}$ denote the fixed set of vertices, and let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a multigraph (directed or un-directed, 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 $A_{ij}^{\mathcal{G}} \in \mathbb{Z}_{\geq 0}$ 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 \mathcal{G} , can be defined via the one-to-one correspondence with adjacency matrices as follows

$$\mathcal{G} = \{\mathcal{G} : A^{\mathcal{G}} \in \mathbb{Z}_{\geq 0}^{V \times V}\},$$

so that we seek to assign each $\mathcal{G} \in \mathcal{G}$ 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 $\mathcal{G}^m \in \mathcal{G}$ (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 $\mathcal{G} \in \mathcal{G}$ is, up to proportionality, the following

$$p(\mathcal{G}|\mathcal{G}^m, \gamma) \propto \exp\{-\gamma\phi(d_G(\mathcal{G}, \mathcal{G}^m))\} \quad (47)$$

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 (47) is given by the following

$$Z(\mathcal{G}^m, \gamma) = \sum_{\mathcal{G} \in \mathcal{G}} \exp\{-\gamma\phi(d_G(\mathcal{G}, \mathcal{G}^m))\},$$

which, with \mathcal{G} being an infinite space, will in general be intractable.

Note this is more-or-less identical the SNF models seen in Lunagomez et al. (2021), Definition 3.4. The only differences being (i) the sample space \mathcal{G} is now all multigraphs 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 6.3, we can use this multigraph-based SNF to construct the following hierarchical model

$$\begin{aligned} \mathcal{G}^{(i)} &\sim \text{SNF}(\mathcal{G}^m, \gamma) \quad (\text{for } i = 1, \dots, n) \\ \mathcal{G}^m &\sim \text{SNF}(\mathcal{G}_0, \gamma_0) \\ \gamma &\sim p(\gamma) \end{aligned}$$

where $\mathcal{G}_0 \in \mathcal{G}$ 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

$$\begin{aligned} p(\mathcal{G}^m, \gamma | \{\mathcal{G}^{(i)}\}_{i=1}^n) &\propto \left(\prod_{i=1}^n p(\mathcal{G}^{(i)} | \mathcal{G}^m, \gamma) \right) p(\mathcal{G}^m) p(\gamma) \\ &= Z(\mathcal{G}^m, \gamma)^{-n} \exp \left\{ -\gamma \sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^m)) \right\} \\ &\quad \times \exp \{ -\gamma_0 \phi(d_G(\mathcal{G}^m, \mathcal{G}_0)) \} p(\gamma), \end{aligned}$$

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 Lunagomez 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 $\mathcal{G}_i^* \sim \text{SNF}(\mathcal{G}^m, \gamma')$ 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

$$\begin{aligned} H(\gamma', \gamma) &= \exp \left\{ -(\gamma' - \gamma) \left(\sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^m)) - \sum_{i=1}^n \phi(d_G(\mathcal{G}_i^*, \mathcal{G}^m)) \right) \right\} \\ &\quad \times \frac{p(\gamma') q(\gamma|\gamma')}{p(\gamma) q(\gamma|\gamma)}. \end{aligned} \quad (48)$$

For the proposal $q(\gamma'|\gamma)$ we consider sampling uniformly over a ε -neighbourhood of γ with reflection at zero (see Supplement S3.1, Eq. 23), for which one has $q(\gamma'|\gamma) = q(\gamma|\gamma')$.

To update the mode, we consider an 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 (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 un-directed graphs, where we must let $A_{ij}^{m'} = A_{ji}^{m'} = x'$, since the adjacency matrices must be symmetric. Here we will assume graphs to be 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

$$\begin{aligned} \alpha(\mathcal{G}^{m'}, \mathcal{G}^m) &= \min \left\{ 1, \frac{p(\mathcal{G}^{m'}|\gamma, \{\mathcal{G}^{(i)}\}_{i=1}^n) \prod_{i=1}^n p(\mathcal{G}_i^*|\mathcal{G}^m, \gamma) q(x|x')}{p(\mathcal{G}^m|\gamma, \{\mathcal{G}^{(i)}\}_{i=1}^n) \prod_{i=1}^n p(\mathcal{G}_i^*|\mathcal{G}^{m'}, \gamma) q(x'|x)} \right\} \\ &= \min \{1, H(\mathcal{G}^{m'}, \mathcal{G}^m)\} \end{aligned}$$

where

$$\begin{aligned} H(\mathcal{G}^{m'}, \mathcal{G}^m) &= \exp \left\{ -\gamma \left(\sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^{m'})) - \sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^m)) \right) \right. \\ &\quad \left. - \gamma \left(\sum_{i=1}^n \phi(d_G(\mathcal{G}_i^*, \mathcal{G}^m)) - \sum_{i=1}^n \phi(d_G(\mathcal{G}_i^*, \mathcal{G}^{m'})) \right) \right. \\ &\quad \left. - \gamma_0 \left(\phi(d_G(\mathcal{G}^{m'}, \mathcal{G}_0)) - \phi(d_G(\mathcal{G}^m, \mathcal{G}_0)) \right) \right\} \frac{q(x|x')}{q(x'|x)}. \end{aligned} \quad (49)$$

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 (49).

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 $\mathcal{G}_i^* \sim \text{SNF}(\mathcal{G}^m, \gamma)$ for some given (\mathcal{G}^m, γ) . 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 4.6). 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

$$\begin{aligned} \alpha(\mathcal{G}, \mathcal{G}') &= \min \left\{ 1, \frac{p(\mathcal{G}' | \mathcal{G}^m, \gamma) q(x|x')}{p(\mathcal{G} | \mathcal{G}^m, \gamma) q(x'|x)} \right\} \\ &= \min \left\{ 1, \exp \left\{ -\gamma (\phi(d_G(\mathcal{G}', \mathcal{G}^m)) - \phi(d_G(\mathcal{G}, \mathcal{G}^m))) \right\} \frac{q(x|x')}{q(x'|x)} \right\}. \end{aligned}$$

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).

References

- George Bolt, Simon Lunagomez, and Christopher Nemeth. Distances for comparing multi-sets and sequences. *arXiv preprint arXiv:2206.08858*, 2022.
- Richard A Levine and George Casella. Optimizing random scan gibbs samplers. *Journal of Multivariate Analysis*, 97(10):2071{2100, 2006.
- Simon Lunagomez, So a C. Olhede, and Patrick J. Wolfe. Modeling network populations via graph distances. *Journal of the American Statistical Association*, 116(536):2023{2040, 2021.
- Iain Murray, Zoubin Ghahramani, and David J.C. MacKay. Mcmc for doubly-intractable distributions. *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, UAI 2006*, pages 359{366, 2006.
- Kirill Neklyudov, Max Welling, Evgenii Egorov, and Dmitry Vetrov. Involutive mcmc: A unifying framework. In *International Conference on Machine Learning*, pages 7273{7282. PMLR, 2020.
- Dingqi Yang, Daqing Zhang, Vincent W. Zheng, and Zhiyong Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45:129{142, 2015.

Algorithm 4: SIS posterior component-wise MCMC

Input: observed data $\{\mathcal{S}^{(i)}\}_{i=1}^n$
 initialise $(\mathcal{S}_0^m, \gamma_0)$
for $i = 1, \dots, m$ **do**
 // Update gamma
 $\gamma_i = \text{dispersion_update}(\mathcal{S}_{i-1}^m, \gamma_{i-1})$ // (Algorithm 5)
 // Update mode
 $\mathcal{S}_i^m = \text{mode_update}(\mathcal{S}_{i-1}^m, \gamma_i)$ // (Algorithm 6)
end
Output: sample $\{(\mathcal{S}_i^m, \gamma_i)\}_{i=1}^m$

Algorithm 5: SIS posterior dispersion conditional accept-reject

Input: $(\mathcal{S}_i^m, \gamma_i)$
Output: γ_{i+1}
Function $\text{dispersion_update}(\mathcal{S}_i^m, \gamma_i)$:
 let $(\mathcal{S}^m, \gamma) = (\mathcal{S}_i^m, \gamma_i)$
 sample γ' via $q(\gamma'|\gamma)$ of (23) // Sample proposal
 sample $\{\mathcal{S}_i^*\}_{i=1}^n$ i.i.d. from $\text{SIS}(\mathcal{S}^m, \gamma')$ // Sample auxiliary data
 evaluate $\alpha = \alpha(\gamma, \gamma')$ of (24) // 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 6: 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 Supplement S3.3 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 $\text{SIS}([\mathcal{S}^m]', \gamma)$ // Sample auxiliary data $\alpha = \alpha(\mathcal{S}^m, [\mathcal{S}^m]')$ of (26), using ratio (31) // Acceptance probability **else**

// Path insertion & deletion move

 let $u, f(u, \mathcal{S}^m)$ and $p(u|\mathcal{S}^m)$ be as in Supplement S3.4 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 $\text{SIS}([\mathcal{S}^m]', \gamma)$ // Sample auxiliary data $\alpha = \alpha(\mathcal{S}^m, [\mathcal{S}^m]')$ of (26), using ratio (33) // Acceptance probability **end**

$$\mathcal{S}_{i+1}^m = \begin{cases} [\mathcal{S}^m]' & \text{with probability } \alpha \\ \mathcal{S}^m & \text{with probability } (1 - \alpha) \end{cases} \quad // \text{Accept/reject proposal}$$
 return \mathcal{S}_{i+1}^m **end**

Algorithm 7: SIS model iMCMC sampling

Input: (\mathcal{S}^m, γ) (model parameters)
Input: $\nu_{ed}, \nu_{td}, p(\mathcal{I}|\mathcal{S}), \beta$ (MCMC tuning parameters)
Input: m (sample size), b (burn-in), l (lag)
 initialise \mathcal{S} ;
 initialise $i = 1$;
while $i \leq m$ **do**
 sample $z \sim \text{Bernoulli}(\beta)$
 if $z = 1$ **then**
 // Edit allocation move
 let $u, f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Supplement S3.3
 sample u via $p(u|\mathcal{S})$
 $(\mathcal{S}', u') = f(\mathcal{S}, u)$
 evaluate $\alpha = \alpha(\mathcal{S}, \mathcal{S}')$ of (34) using (35)
 else
 // Path insertion & deletion move
 let $u, f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Supplement S3.4
 sample u via $p(u|\mathcal{S})$
 $(\mathcal{S}', u') = f(\mathcal{S}, u)$
 evaluate $\alpha = \alpha(\mathcal{S}, \mathcal{S}')$ of (34) using (36)
 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 $(i \bmod l = 1)$ **then**
 $\mathcal{S}_i \leftarrow \mathcal{S}$
 $i = i + 1$
 end
end
Output: $\{\mathcal{S}_i\}_{i=1}^m$

Algorithm 8: SIM posterior component-wise MCMC

Input: observed data $\{\mathcal{E}^{(i)}\}_{i=1}^n$
 initialise $(\mathcal{E}_0^m, \gamma_0)$
for $i = 1, \dots, m$ **do**
 // Update gamma
 $\gamma_i = \text{dispersion_update}(\mathcal{E}_{i-1}^m, \gamma_{i-1})$ // (Algorithm 9)
 // Update mode
 $\mathcal{E}_i^m = \text{mode_update}(\mathcal{E}_{i-1}^m, \gamma_i)$ // (Algorithm 6)
end
Output: sample $\{(\mathcal{E}_i^m, \gamma_i)\}_{i=1}^m$

Algorithm 9: 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 (23)*// Sample proposal* sample $\{\mathcal{E}_i^*\}_{i=1}^n$ i.i.d. from SIM(\mathcal{E}^m, γ')*// Sample auxiliary data* evaluate $\alpha = \alpha(\gamma, \gamma')$ of (38)*// 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 10: 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  $z \sim \text{Bernoulli}(\beta)$ 
    if  $z = 1$  then
        // Edit allocation move
        let  $u, f(u, \mathcal{S}^m)$  and  $p(u|\mathcal{S}^m)$  be as in Supplement S3.3
        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  $\text{SIM}([\mathcal{E}^m]', \gamma)$  // Sample auxiliary data
         $\alpha = \alpha(\mathcal{E}^m, [\mathcal{E}^m]')$  of (42), using ratio (31) // Acceptance probability
    else
        // Path insertion & deletion move
        let  $u, f(u, \mathcal{S}^m)$  and  $p(u|\mathcal{S}^m)$  be as in Supplement S3.4
        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  $\text{SIM}([\mathcal{E}^m]', \gamma)$  // Sample auxiliary data
         $\alpha = \alpha(\mathcal{E}^m, [\mathcal{E}^m]')$  of (42), using ratio (33) // 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 11: SIM model iMCMC sampling

Input: (\mathcal{E}^m, γ) (model parameters)
Input: $\nu_{\text{ed}}, \nu_{\text{td}}, p(\mathcal{I}|\mathcal{S}), \beta$ (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 Supplement S3.3
 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 (46) using (35) // Acceptance probability
 else
 // Path insertion & deletion move
 let $u, f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Supplement S3.4
 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 (46) using (36) // 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 $(i \bmod l = 1)$ **then**
 $\mathcal{E}_i \leftarrow \mathcal{E}$
 $i = i + 1$
 end
end
Output: $\{\mathcal{E}_i\}_{i=1}^m$
