Journal of Machine Learning Research 26 (2025) 1-66 Submitted 6/22; Published 6/25

Modelling Populations of Interaction Networks via Distance Metrics

George Bolt G.BOLT@LANCASTER.AC.UK
STOR-i Centre for Doctoral Training
Lancaster University, Lancaster, UK, LA1 4YF

Simón Lunagómez SIMON.LUNAGOMEZ@ITAM.MX
Department of Statistics
Instituto Tecnológico Autónomo de México (ITAM)
Río Hondo 1, Altavista, Álvaro Obregón, 01080 Ciudad de México, CDMX, Mexico

Christopher Nemeth C.NEMETH@LANCASTER.AC.UK
School of Mathematical Sciences
Lancaster University, Lancaster, UK, LA1 4YF

Editor: Debdeep Pati

Abstract

Network data arises through the observation of relational information between a collection of entities, for example, friendships (relations) amongst a sample of people (entities). Traditionally, statistical models of such data have been developed to analyse a single network, that is, a single collection of entities and relations. More recently, attention has shifted to analysing *samples* of networks. A driving force has been the analysis of connectome data, arising in neuroscience applications, where a single network is observed for each patient in a study. These models typically assume, within each network, the entities are the units of observation, that is, more data equates to including more entities. However, an alternative paradigm considers relations—such as edges or paths—as the observational units, exemplified by email exchanges or user navigations across a website. This interaction network framework has generally been applied to single networks, without extending to the case where multiple such networks are observed, for instance, analysing navigation patterns from many users. Motivated by this gap, we propose a new Bayesian modelling framework to analyse such data. Our approach is based on practitioner-specified distance metrics between networks, allowing us to parameterise models analogous to Gaussian distributions in network space, using location and scale parameters. We address the key challenge of defining meaningful distances between interaction networks, proposing two new metrics with theoretical guarantees and practical computation strategies. To enable efficient Bayesian inference, we develop specialised Markov chain Monte Carlo (MCMC) algorithms within the involutive MCMC (iMCMC) framework, tailored to the doubly-intractable and discrete nature of the induced posteriors. Through simulation studies, we demonstrate the robustness and efficiency of our approach, and we showcase its applicability with a case study on a location-based social network (LSBN) dataset.

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 form of network data comes 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 a 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 (Lunagómez 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 dataset 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 dataset of Yang et al. (2015) contains historical check-ins of users in New York and Tokyo. Consider a single user. Notice one 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 other words, the data of a single user can be seen as an interaction network, so that data on *multiple* users can thus be seen as a *sample* of interaction networks.

With a single observed network, 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, Lunagómez et al. (2021) construct models through graph distances, using the Fréchet 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

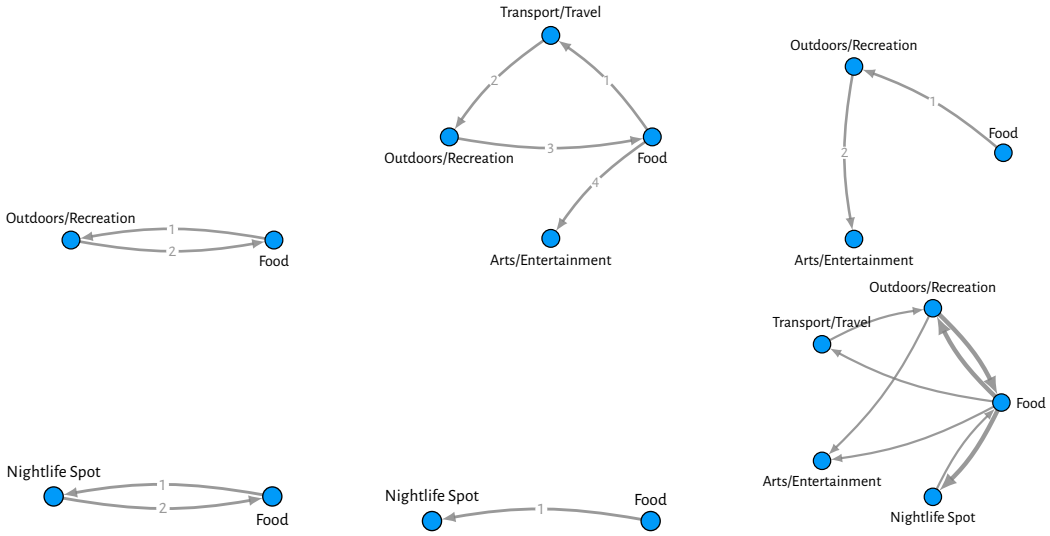


Figure 1: Data of a single user from the Foursquare dataset (Yang et al., 2015). Going from left to right, the first five subplots show the observed interactions sequence $\mathcal{S}$, with each path representing a single day of check-ins, whilst the final plot visualises the aggregate multigraph $\mathcal{G}_{\mathcal{S}}$,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

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) (Hoff 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 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 observa-

tion 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 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 to be 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 first 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 recent developments of so-called edge-exchangeable network models (Cai et al., 2016; Crane and Dempsey, 2018; Williamson, 2016; Ghalebi et al., 2019b,a). Of these, we note that only Crane and Dempsey (2018) allow paths as 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 chooses to model each observation individually. We look to address this gap. To this end, we propose a new Bayesian modelling framework. Using a practitioner-specified distance metric between interaction networks, 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 choice of distance, leading to a flexible framework which can be tailored to suit different questions of interest. However, the problem of measuring the distance between interaction networks has not yet been explicitly addressed

in the literature. As such, we propose two distance measures, borrowing ideas from the wider literature. For each distance, we prove conditions under which they will be distance metrics, and provide details on how they can be computed.

Procedures for posterior inference are also developed, where we propose specialised Markov chain Monte Carlo (MCMC) algorithms (Robert et al., 1999; Fearnhead et al., 2025). These are required to sample not only from posterior distributions over model parameters but also the models themselves. This is particularly challenging for two prominent reasons. Firstly, it requires sampling from distributions over the space of interaction networks; a non-trivial discrete space containing objects of differing dimensions. Secondly, our posterior distributions require the evaluation of intractable normalising constants that depend on the model parameters, making them doubly intractable (Murray et al., 2006). As a solution, we combine the exchange algorithm of Murray et al. (2006) with the involutive MCMC (iMCMC) framework of Neklyudov et al. (2020); the former circumvents issues pertaining to the double-intractability of our posterior distributions, whilst the latter provides added flexibility in proposal generation, aiding navigation of the sample space. The result is a generalisation of the exchange algorithm that we refer to as the *iExchange algorithm*.

The remainder of this paper will be structured as follows. In Section 2, we provide background details regarding the data structure and notation. In Section 3, we formally introduce our proposed models, before detailing our proposed distances measures for use therein in Section 4. In Section 5, we outline our Bayesian scheme, discussing prior specification, stating our assumed hierarchical model and detailing our proposed MCMC algorithms. In Section 6, we detail simulation studies undertaken to confirm the efficacy of our methodology and posterior inference scheme, whilst in Section 7 we illustrate its applicability via an analysis of the Foursquare check-in data. We finalise with conclusions and discussion in Section 8.

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: *interaction sequences* and *interaction multisets*, covering the ordered and un-ordered cases respectively.

In setting up our data representation, we build upon that used by Crane and Dempsey (2018). The collection of entities under consideration are denoted via a *vertex set* $\mathcal{V}$, assumed to be some discrete set (typically the set of integers $\mathcal{V} = \{1, \ldots, V\}$) 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. Given a vertex set $\mathcal{V}$, an interaction sequence will be denoted as follows

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

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

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

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

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

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 $\mathcal{I}_2$. Note also this interaction multiset representation is very similar to that adopted by Crane and Dempsey (2018): what they define as an interaction network can be seen as a countably infinite interaction multiset.

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

As alluded to, this work considers the case when *multiple* interaction networks are observed. For example, in the Foursquare dataset, we have check-in information on a sample of users, and thus observe a sample of interaction networks (one for each user). Representing the ith observation via a interaction sequence $\mathcal{S}^{(i)}$ or multiset $\mathcal{E}^{(i)}$, and letting n denote the sample size, we therefore observe

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

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

We finish this subsection by discussing aggregation. Both representations of interaction networks can be aggregated to form graphs, and the 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*). A slight generalisation that will be useful is that of a *multigraph*, which is a graph wherein edges can occur more than once. The definition is almost identical to a graph, only that the edge set $\mathcal{E}$ is assumed to be a multiset, rather than a set.

Considering first aggregating an interaction sequence, one can convert any $\mathcal{S}$ over vertices $\mathcal{V}$ to 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 $x_{k(l+1)}$ for some $1 \leq k \leq N$ and $1 \leq l \leq n_k$. Moreover, a traversal between two vertices may occur more than once in a given interaction sequence. Thus one can also aggregate 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 construct a multiset as edges via

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

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

An interaction mutliset $\mathcal{E}$ over vertex set $\mathcal{V}$ can be aggregated in a similar manner. Letting $\tilde{\mathcal{S}}$ be an interaction sequence obtained by placing the paths of $\mathcal{E}$ in arbitrary order, then we let

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

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

We finalise this section by noting the process of aggregation outlined above 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. However, this 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, as the interaction sequences and multisets we be handled directly.

3 Metric-Based Interaction-Network Models

In this section, our proposed models for interaction sequences and multisets will be introduced. 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 Model Definitions

Starting with our model for interaction sequences, let $\boldsymbol{\mathcal{S}}^*$ denote the infinite discrete space containing all interaction sequences over a fixed vertex set $\mathcal{V}$ (for a formal definition, see Appendix A). Towards eliciting a probability distribution over the sample space $\boldsymbol{\mathcal{S}}^*$, we first endow it with a distance metric $d_S : \boldsymbol{\mathcal{S}}^* \times \boldsymbol{\mathcal{S}}^* \to \mathbb{R}_+$, which takes as input two interaction sequences and returns a measure of their dissimilarity. Making use of two model two parameters, (i) an interaction sequence $\mathcal{S}^m \in \boldsymbol{\mathcal{S}}^*$, referred to as the *mode*, and (ii) $\gamma > 0$, referred to as the *dispersion*, we define a family of probability distributions as follows.

Definition 1 (Spherical Interaction Sequence Family)**.** *For a given distance metric $d_S(\cdot,\cdot)$ on $\boldsymbol{\mathcal{S}}^*$, a mode $\mathcal{S}^m \in \boldsymbol{\mathcal{S}}^*$ and a dispersion parameter $\gamma > 0$, the probability of observing $\mathcal{S}$ is given by*

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

and we write

$$\mathcal{S} \sim \mathrm{SIS}(\mathcal{S}^m, \gamma) \tag{2}$$

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

We defined an analogous family of models for interaction multisets as follows. In this case, we let $\boldsymbol{\mathcal{E}}^*$ denote the space containing all interaction multisets over the fixed vertex set $\mathcal{V}$ (for a formal definition, see Appendix A). To elicit a distribution over the sample space $\boldsymbol{\mathcal{E}}^*$ we again endow it with a distance metric $d_E : \boldsymbol{\mathcal{E}}^* \times \boldsymbol{\mathcal{E}}^* \to \mathbb{R}_+$, measuring the dissimilarity of any two interaction multisets. With similar model parameters (i) the mode $\mathcal{E}^m \in \boldsymbol{\mathcal{E}}^*$, in this case an interaction *multiset*, and (ii) the dispersion $\gamma > 0$, we define a family of probability distributions as follows.

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

$$p(\mathcal{E} \,|\, \mathcal{E}^m, \gamma) \propto \exp\{-\gamma d_E(\mathcal{E}, \mathcal{E}^m)\}, \tag{3}$$

and we write

$$\mathcal{E} \sim \mathrm{SIM}(\mathcal{E}^m, \gamma) \tag{4}$$

if we assume $\mathcal{E}$ was sampled via (3). This we refer to as the Spherical Interaction-Multiset (SIM) family of probability distributions over $\boldsymbol{\mathcal{E}}^*$ with parameters $\mathcal{E}^m$ and γ.

3.2 Discussion

Intuitively, the SIS and SIM models can be seen as Gaussian-like distributions over the space of interaction sequences and multiset, respectively. The mode $\mathcal{S}^m$ (resp. $\mathcal{E}^m$) plays the role of the mean, controlling the center of the distribution, whilst the dispersion γ controls the scale.[1] The role of γ can also be formalised through its impact on the entropy, which can be shown to be monotonic in γ (see Supplement S2).

Observe in both (2) and (4) no reference is made to $d_S(\cdot,\cdot)$ or $d_E(\cdot,\cdot)$, even though the respective distributions clearly depend on them. The reasoning here is these values are not intended to be model parameters but instead subjective choices made by the practitioner prior to any analysis.

This naturally raises the question of specifying distances $d_S(\cdot,\cdot)$ or $d_E(\cdot,\cdot)$, for which there appear no immediate candidates in the literature. This problem will be dealt with in the next section, where two such distances will be proposed. Example networks sampled from the models defined via these distances will also be provided, giving some intuition for the nature of such distributions.

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

Observe both distributions were presented in unnormalised form. Taking the SIS model, for example, its distribution can be normalised as follows

$$p(\mathcal{S} \,|\, \mathcal{S}^m, \gamma) = Z(\mathcal{S}^m, \gamma)^{-1} \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} \tag{5}$$

where

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

is the normalising constant, often referred to as the *partition function*. In general, this summation is intractable, which will come into play significantly when considering computational aspects in later sections. In fact, with $\boldsymbol{\mathcal{S}}^*$ being infinite, there is no guarantee that (3.2) will even exist for a given γ, and thus for some parameterisations (5) may be an improper distribution. This has pragmatic implications, motivating our recommendation to work with constrained sample spaces in practice, as defined in Appendix A.2. For further elaboration on this recommendation, see Supplement S3.

3.3 Relations to Literature

We finalise this section by relating the proposed models to others appearing in the literature. As already mentioned, Lunagómez et al. (2021) proposed a model for graphs which makes analogous use of distance metrics, though there are also examples beyond the networks literature. Most notably the Mallows model (Vitelli et al., 2018), appearing in the context of preference learning, and the complex Watson distribution (Mardia and Dryden, 1999), used in shape analysis. All these models combine the use of an exponential kernel with a notion of distance to define distributions over objects, as similarly done here. The differential factor, however, is the underlying space, be that the space of graphs (Lunagómez et al., 2021), ranks (Mallows model), shapes (complex Watson model), or as in this work, interaction networks.

Though in theory, these models are immediate extensions of one another, the different spaces typically bring their own challenges, both methodological and computational. A notable difference with our models are the flexible assumptions made regarding the size of interaction networks being considered: we do not assume observations have a fixed number of interactions, or that interactions have a fixed length. As we will see, this raises computational challenges when it comes to designing algorithms to sample from these models or conduct parameter inference. More immediately, this has consequences for the elicitation of distance measures.

4 Distances

The models proposed in Section 3 require the specification of a distance between interaction networks, represented either as sequences or multisets of interactions. Given such a problem has not been dealt with explicitly in the literature, in this section two such distances will be proposed, for the comparison of interaction sequences and multisets, respectively.

As will be seen, these two distances are closely related, both borrowing ideas from the wider literature to elicit a measure of dissimilarity in terms of an optimal pairing of paths. Their evaluation, which requires solving an optimisation problem, will be outlined. In

addition, their theoretical properties will be stated and proved, including conditions under which both distances will be metrics. Finally, to illustrate how these distances can be used within the models of Section 3 example networks sampled from our models will be presented.

4.1 Comparing Interaction Multisets

Starting with a distance to compare interaction multisets $\mathcal{E}$ and $\mathcal{E}'$, we propose the *matching distance*. At a high level, this seeks the 'best' pairing of interactions from $\mathcal{E}$ with those from $\mathcal{E}'$, in particular, one which minimises some notion cost associated to each pairing. The distance between $\mathcal{E}$ and $\mathcal{E}'$ is then given by the cost of this best pairing. In this way, the matching distance judges the dissimilarity based on an optimal relation between the interactions of either multiset.

Towards defining this distance, it is necessary to more formally define a pairing of interactions. For this, we use the notion of a *matching*. A matching between $\mathcal{E}$ and $\mathcal{E}'$ (Figure 2a) is simply a multiset of pairs

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

such that each $\mathcal{I} \in \mathcal{E}$ is matched to at most one $\mathcal{I}' \in \mathcal{E}'$, and *vice versa*. Observe by definition one must have $0 \leq |\mathcal{M}| \leq \min(|\mathcal{E}|, |\mathcal{E}'|)$, that is, we can match at most the number of interactions in the smaller multiset. A matching which achieves this upper bound we say is *complete*. For example, the matching of Figure 2a is complete. We also define the restriction of $\mathcal{M}$ to $\mathcal{E}$ as follows

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

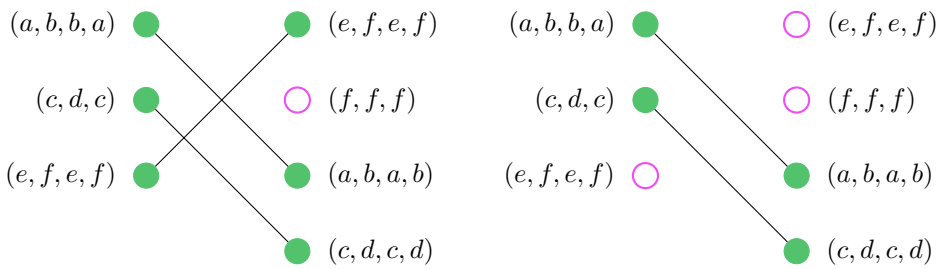
so that $\mathcal{M}_{\mathcal{E}} \subseteq \mathcal{E}$ denotes the elements of $\mathcal{E}$ which are included in the matching $\mathcal{M}$. We also introduce the shorthand $\mathcal{M}^{c}_{\mathcal{E}} := \mathcal{E} \setminus \mathcal{M}_{\mathcal{E}}$ to denote the elements of $\mathcal{E}$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

To define the matching distance, it will be necessary to have a notion of cost for each matching. For this, it will be assumed one has access to a distance between interactions $d_I(\mathcal{I}, \mathcal{I}') \geq 0$, quantifying how dissimilar a given pair of interactions $\mathcal{I}$ and $\mathcal{I}'$ are. In addition, a penalty function $\delta(\cdot)$ must be specified, with $\delta(\mathcal{I}) > 0$ denoting the cost incurred when interaction $\mathcal{I}$ is left unmatched. Examples of such distances and penalties will be provided in subsequent sections, but in principle, any choices thereof can be made. With these elements, the matching distance can be defined as follows.

Definition 3 (Matching Distance)**.** *Given a distance $d_I(\cdot,\cdot)$ between interactions and a penalty function $\delta : \boldsymbol{\mathcal{I}} \to \mathbb{R}_{>0}$ for unmatched interactions, the matching distance between $\mathcal{E}$ and $\mathcal{E}'$ is given by*

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

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



(a) general matching (b) monotone matching

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

Notice, as claimed earlier, the matching distance $d_{\mathrm{M},\delta(\cdot)}$ is defined by finding a matching $\mathcal{M}$ with minimum cost. The cost of $\mathcal{M}$ consists of (i) distances between matched interactions, and (ii) penalties for the interactions of $\mathcal{E}$ or $\mathcal{E}'$ left unmatched.

Though well-defined, the question of how to compute of $d_{\mathrm{M},\delta(\cdot)}$ remains, which will require finding an optimal matching. Noting this is essentially an assignment problem, one can appeal to solvers thereof, such as the Hungarian algorithm (Kuhn, 1955). Further details can be found in Supplement S4.1, where we show how to set up a suitable assignment problem to be solved. In general, this involves (i) evaluating all pairwise distances between $\mathcal{E}$ and $\mathcal{E}'$, and (ii) solving an assignment problem via a chosen solver. This leads to a computational complexity of $\mathcal{O}(N \cdot M + f(N, M))$, where N and M denote the number of paths in $\mathcal{E}$ and $\mathcal{E}'$, respectively, and $f(\cdot, \cdot)$ is a solver-dependent term. For example, if optimising over complete matchings via the Hungarian algorithm one will have $f(N, M) = \max(N, M)^3$ (see Supplement S4.1.2).

Regarding the wider literature, Ramon and Bruynooghe (2001) and Eiter and Mannila (1997) have proposed similar distances, both considering the problem of comparing sets within a metric space, defining distances via optimal relations between set elements. In fact, the vernacular and notion of matchings we have adopted was inspired by Ramon and Bruynooghe (2001). However, both consider comparing sets, whilst we consider *multisets*. In addition, both assume a particular form of penalty for unmatched elements. Instead of this, we define our distance in terms of a general penalty function. As will be outlined later, we also provide conditions on this penalty function which ensure the resulting distance satisfies certain theoretical properties.

4.2 Comparing Interaction Sequences

Turning now to the comparison of interaction sequences $\mathcal{S}$ and $\mathcal{S}'$, we propose the *edit distance*. Unlike when comparing multisets, there is now an ordering which must be taken into account. For this, one can naturally adapt the rationale of the matching distance: find an optimal matching which preserves order, or what we call a *monotone matching*

(Figure 2b). As with the matching distance, the cost associated with this optimal monotone matching can then be taken as a measure of dissimilarity between the two interaction sequences.

Intuitively, a monotone matching of two sequences is one in which no lines cross when this matching is drawn, as in Figure 2b. Towards a more formal treatment, recall from Section 2 that we write interaction sequences as

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

such that, for example, $\mathcal{I}_i$ is observed before $\mathcal{I}_{i+1}$. In this way, the indices of interactions encode an ordering. As for multisets, matching between the sequences $\mathcal{S}$ and $\mathcal{S}'$ is a multiset of pairs $\mathcal{M} = \{(\mathcal{I}, \mathcal{I}') : \mathcal{I} \in \mathcal{S}, \mathcal{I}' \in \mathcal{S}'\}$ such that each entry of either sequence is paired with at most one from the other. Here, one can also consider whether order is preserved by the matching. In particular, if for any $(\mathcal{I}_{i_1}, \mathcal{I}'_{j_1}) \in \mathcal{M}$ and $(\mathcal{I}_{i_2}, \mathcal{I}'_{j_2}) \in \mathcal{M}$ we have

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

which describes formally the intuition that no lines cross when the matching is drawn. A matching satisfying this condition is said to be *monotone*. Using this notion, the edit distance can be defined as follows.

Definition 4 (Edit Distance)**.** *Given a distance $d_I(\cdot, \cdot)$ between interactions and a penalty function $\delta : \boldsymbol{\mathcal{I}} \to \mathbb{R}$ for unmatched interactions, the edit distance between $\mathcal{S}$ and $\mathcal{S}'$ is given by*

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

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

Notice Definition 4 is more-or-less identical to the definition of the matching distance (Definition 3); the difference being that $\mathcal{M}$, the matching over which one is optimising, must be monotone. Again, one is free to choose $d_I(\cdot, \cdot)$ and $\delta(\cdot)$, each defining a different edit distance.

Computation of $d_{\mathrm{E},\delta(\cdot)}$ also requires solving an optimisation problem. In this case, the task turns out to be slightly less computationally costly, being possible via dynamic programming with complexity $\mathcal{O}(N \cdot M)$. Further details can be found in Supplement S4.2.

In regards to the wider literature, the edit distance can be seen as an adaptation of the string-edit distance proposed by Wagner and Fischer (1974); though our presentation via monotone matchings does differ. The string-edit distance was originally proposed to compare categorical sequences, but has since been applied in other contexts (including the present work). Most notably, with the geometric edit distance (Gold and Sharir, 2018; Fox and Li, 2019), which adapts the string edit distance for the comparison of time series.

A final point to note is the edit distance (Definition 4) and matching distance (Definition 3) are themselves closely related: the latter is essentially an unordered version of the former. As far as we are aware, this observation, which follows naturally due to our definition of both distances via matchings, has not been noted in any other applications.

4.3 Comparing Interactions

Both distances defined in Sections 4.1 and 4.2 require a distance between interactions to be specified. In this section, examples of two such distances will be given. Since interactions are assumed to be paths, this simplifies to the problem of measuring the distance between two paths, denoted as follows

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

A natural approach is to consider how much $\mathcal{I}$ and $\mathcal{I}'$ have, or do not have, in common. In particular, common subpaths and subsequences, as illustrated in Figure 3. A *subpath* of $\mathcal{I}$ from index i to j is given by the following

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

where $1 \leq i \leq j \leq n$ (Figure 3a). More generally, assuming $\boldsymbol{v} = (v_1, \dots, v_s)$ with $1 \leq v_1 < \cdots < v_s \leq n$, then a *subsequence* of $\mathcal{I}$ is obtained by indexing with $\boldsymbol{v}$ as follows

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

which will be of length s (Figure 3b). Given two paths, one can then consider *common* subpaths and subsequences. A common subpath of $\mathcal{I}$ and $\mathcal{I}'$ occurs when we have

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

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

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

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

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

where

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

denote the maximum size of a common subpath and subsequence between the two paths. Intuitively, these distances count the number of entries of $\mathcal{I}$ and $\mathcal{I}'$ not included in the common subpath or subsequence, that is, the underlined entries in Figure 3. For example, since the subpaths and subsequences in Figure 3 are maximal, we have $d_{\mathrm{LSP}}(\mathcal{I}, \mathcal{I}') = 7$ and $d_{\mathrm{LCS}}(\mathcal{I}, \mathcal{I}') = 5$ in this case.

Computation of these distances necessitates finding maximal common subsequences or subpaths, much like for the matching and edit distances themselves. In both cases, this can be achieved by dynamic programming with a complexity of $\mathcal{O}(n \cdot m)$, details of which can be found in Supplement S4.3.

(a) Common subpath (b) Common subsequence

Figure 3: A comparison of common subsequences and subpaths. Here (a) and (b) show the same pair of paths, with (a) highlighting a common subpath, as indicated by shaded (green) entries, whilst (b) shows a common subsequence. In both cases, these are maximal.

4.4 Theoretical Properties

In this section, the theoretical properties of the aforementioned distances will be discussed. In particular, we will consider the notion of a *distance metric*, providing conditions under which the proposed distances will be classified as such. In a general sense, a distance metric is defined as follows.

Definition 5 (Distance Metric)**.** *A function $d : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \to \mathbb{R}_{\geq 0}$ is a distance metric over the space $\boldsymbol{\mathcal{X}}$ if, for any $x, y, z \in \boldsymbol{\mathcal{X}}$, the following conditions are satisfied*

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

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

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

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

Notice this definition applies naturally to all distances presented in previous sections; between sequences or multisets of interactions, or between interactions themselves. Regarding why such properties are desirable, aside from being somewhat natural, recall the reason for introducing such distances: use within the models proposed in Section 3.1. From this point of view, conditions (i) and (ii) will be a strict requirement. Without (i), the resultant model is likely to be unidentifiable, making parameter inference more challenging. Without (ii), the model definitions would depend on which way round the metric is called in the probability mass functions (1) and (3), introducing some unnecessary complication. The need for (iii) is arguably less strict but is nonetheless a somewhat intuitive and natural property for a distance function.

One can show that both the matching distance $d_{\mathrm{M},\delta(\cdot)}$ and the edit distance $d_{\mathrm{E},\delta(\cdot)}$ are distance metrics, provided the penalty function $\delta(\cdot)$ satisfies certain conditions and the distance between interaction $d_I(\cdot, \cdot)$ is also a metric. This is summarised with the following result, proved in Supplements S5.1 and S5.2.

Proposition 6 (Matching and Edit Distances are Metrics)**.** *Both the matching distance $d_{\mathrm{M},\delta(\cdot)}$ and the edit distance $d_{\mathrm{E},\delta(\cdot)}$ are distance metrics, provided $d_I(\cdot, \cdot)$ is also a distance metric and the penalty function $\delta(\cdot)$ satisfies the following conditions:*

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

Examples of two penalty functions which satisfy the required conditions are as follows:

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

Note with interactions being paths, we let Λ denote the empty path. In this case, typically $d_I(\mathcal{I}, \Lambda)$ will denote the size of $\mathcal{I}$, for example, with the LSP distance $d_{\mathrm{LSP}}(\mathcal{I}, \Lambda) = n$ where n is the length of $\mathcal{I}$.

Although both induce valid distances, we find the fixed penalty results in distances which are not suitable for use with the models proposed in Section 3.1. For a justification of this claim, see Supplement S1. Instead, as will be seen in later sections, we turn to the distance-based penalty. For brevity, the short-hand notation $d_{\mathrm{M}}(\cdot, \cdot)$ and $d_{\mathrm{Edit}}(\cdot, \cdot)$ will be adopted for the matching distance and edit distance with the distance-based penalty, respectively.

Proposition 6 additionally relies on the assumption that the interaction distance $d_I(\cdot, \cdot)$ is itself a distance metric. This raises the question of whether the two distances proposed in Section 4.3 satisfy this requirement. As we detail in Supplement S5.4, it can be shown that both the LCS and LSP distances are indeed metrics.

4.5 Illustrative Model Samples

Supposing either of the previous two distances has been assumed within our models, it is natural to ask what are the features of the resulting probability distribution over the space of interaction networks. To provide some intuition in this regard, this section presents some example samples drawn from our models with different distance and parameter specifications. In particular, we will (i) illustrate the role of γ in controlling the level of noise, (ii) contrast the SIS and SIM models, showing how the assumptions regarding the order of paths manifests itself in observations, and (iii) compare models with different distance specifications.

Figure 4 summarises these sampled observations, presented via two tables, showing samples from SIS and SIM models respectively. These are further divided, showing samples from each model with different assumed distances. In particular, the edit distance d_{Edit} and matching distance d_{M} were used for the SIS and SIM model respectively, with d_{LCS} and d_{LSP} as the interaction distances, as indicated in Figure 4 via the left-hand tabs. Within each cell, we show three samples from the associated model with increasing values for the dispersion, that is, for the SIS model we show samples $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ where $\mathcal{S}_i \sim \mathrm{SIS}(\mathcal{S}^m, \gamma_i)$, whilst for the SIM model we show $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ where $\mathcal{E}_i \sim \mathrm{SIM}(\mathcal{E}^m, \gamma_i)$, where the γ_i were increasing. The mode parameter for these models was fixed throughout and is shown at the top of Figure 4, that is, $\mathcal{S}^m = ((1,1,1),(2,2,2),(3,3,3))$ for the SIS models and $\mathcal{E}^m = \{(1,1,1),(2,2,2),(3,3,3)\}$ for the SIM models. The vertex set was also assumed to be $\mathcal{V} = \{1, \ldots, 7\}$. Finally, entries have been highlighted to show how observations are associated with the mode. In particular, shaded entries indicate those shared with the mode, whilst underlined entries represent errors. These were obtained from the optimal

matchings and common subsequences or subpaths found when evaluating the distance of these samples to the mode.

Considering first the role of γ in controlling noise, this can be observed through the presence of a larger number of underlined entries for observations drawn with lower γ values, that is, those towards the bottom of each cell. Notice how this follows from the location and scale structure of the model, as discussed in Section 3.1: as γ decreases the probability becomes less concentrated about the mode, leading to a higher probability of entries *not* being shared.

Notice also, that for all models, each sampled observation contains paths with shaded entries that can be matched with exactly one in the mode. Take, for example, the sample at the very bottom. Here the second path has three shaded entries $(1, 1, 1)$ which one can see is equal to the first path of the mode. Similarly, the fourth and fifth paths of this observation can be matched with the second and third of the mode. This feature, whereby paths in the observations are matched with a path of the mode, is a consequence of using the edit and matching distances, which, as seen in Sections 4.1 and 4.2, are defined by such matchings.

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

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

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

5 Bayesian Inference

Given an assumed model, the goal of inference is to discern which parameters are likely to have generated the observed data, which in our case amounts to inferring the mode and dispersion parameters. We adopt a Bayesian perspective, using a specialised MCMC

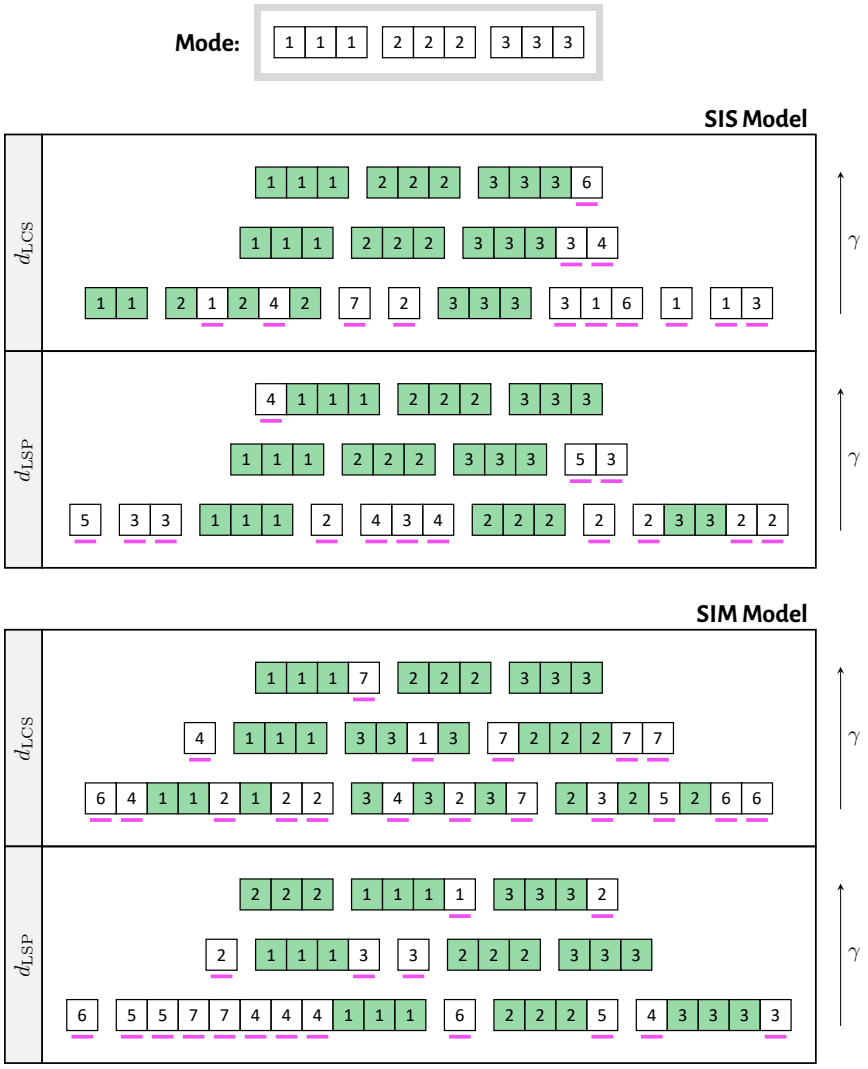


Figure 4: Example samples drawn from our models. Each table cell visualises three randomly drawn samples from a given model with the dispersion parameter γ varying, where a higher γ means a more concentrated distribution, that is, less noise. A common mode was used for each model, as displayed at the top. The edit and matching distances were assumed, for the SIS and SIM models, respectively, with different choices of path distance, as indicated on the left-hand tabs. For each sample, shaded entries indicate those matched with the mode, as implied by the optimal matchings and maximal common subsequences or subpaths found when evaluating the distances, whilst underlined entries indicate unmatched entries.

algorithm to obtain samples from the joint posterior of the mode and dispersion parameters, upon which we base our inference. In this section, details regarding this approach will be provided.

For brevity, only details regarding inference for the interaction-sequence models (Definition 1) will be provided here, delegating details for the interaction-multiset models (Definition 2) to Supplement S9. Furthermore, what follows will be mostly descriptive, with theoretical justifications found in the supplementary materials. In addition, details and guidance regarding the computational cost and mixing of our proposed algorithms can be found in Supplement S6.

5.1 Priors, Hierarchical Model and Posterior

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

$$\mathcal{S}^m \sim \text{SIS}(\mathcal{S}_0, \gamma_0) \tag{6}$$

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

$$\begin{aligned} \mathcal{S}^{(i)} \,|\, \mathcal{S}^m, \gamma &\sim \text{SIS}(\mathcal{S}^m, \gamma) \quad (\text{for } i = 1, \ldots, n) \\ \mathcal{S}^m &\sim \text{SIS}(\mathcal{S}_0, \gamma_0) \\ \gamma &\sim p(\gamma). \end{aligned} \tag{7}$$

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

$$\begin{aligned} p(\mathcal{S}^m, \gamma \,|\, \{\mathcal{S}^{(i)}\}_{i=1}^n) &\propto p(\{\mathcal{S}^{(i)}\}_{i=1}^n \,|\, \mathcal{S}^m, \gamma) p(\mathcal{S}^m) p(\gamma) \\ &\propto Z(\mathcal{S}^m, \gamma)^{-n} \exp\left\{ -\gamma \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) \right\} \\ &\quad \times \exp\{-\gamma_0 d_S(\mathcal{S}^m, \mathcal{S}_0)\} p(\gamma). \end{aligned} \tag{8}$$

5.2 Sampling from the Posterior

To sample from the posterior (8), we use a component-wise MCMC algorithm which alternates between sampling from the two conditional distributions

$$p(\mathcal{S}^m \,|\, \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n) \quad \text{and} \quad p(\gamma \,|\, \mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n). \tag{9}$$

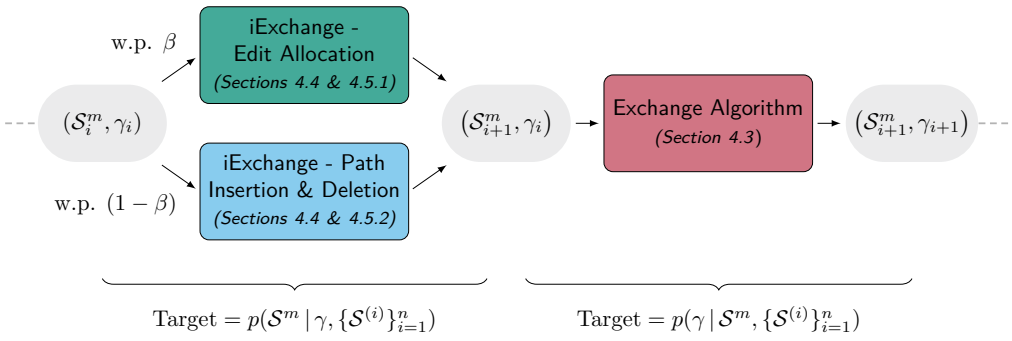


Figure 5: Summary of our MCMC scheme to sample from the SIS posterior. We first update the mode via the iExchange algorithm, doing an edit allocation move with probability β, or a path insertion and deletion move otherwise. We then update the dispersion via the exchange algorithm.

Since the normalising constant $Z(\mathcal{S}^m, \gamma)$ depends on the parameters of interest this implies (8) 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 the use of the exchange algorithm proposed by Murray et al. (2006).

A high-level summary of our scheme is visualised in Figure 5. 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.

5.3 Updating the Dispersion

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

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

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

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

3. Evaluate the following probability

$$\alpha(\gamma, \gamma') = \min\left\{1, \frac{p(\gamma'|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n)p(\{\mathcal{S}_i^*\}_{i=1}^n|\mathcal{S}^m, \gamma)q(\gamma|\gamma')}{p(\gamma|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n)p(\{\mathcal{S}_i^*\}_{i=1}^n|\mathcal{S}^m, \gamma')q(\gamma'|\gamma)}\right\} \tag{10}$$

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 resources. A closed form of (10) and derivation thereof can be found in Supplement S8.1.

5.4 Updating the Mode

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

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

4. Evaluate the following probability

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

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.

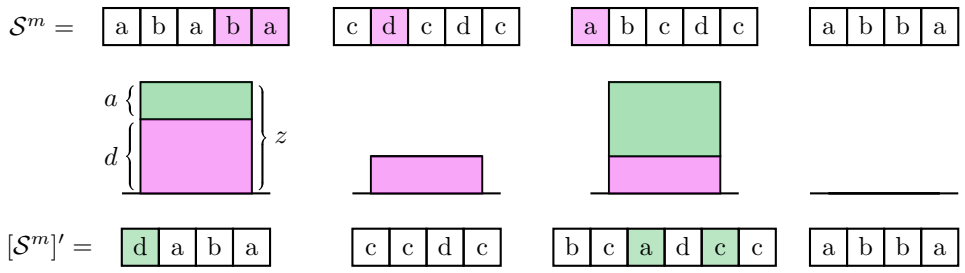


Figure 6: Illustrating the edit allocation move. Shaded entries indicate deletions and insertions, whilst bars visualise the allocation of edits to paths. Bar height is proportional to the number of edits allocated to a path z, whilst the green (top) portion of the bar denotes the number of insertions a and the pink (bottom) portion represents the number of deletions d.

This scheme represents an instance of what we call the iExchange algorithm (Algorithm 1, Supplement S7). As shown in Supplement S7, 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 S7 includes a Jacobian term in the acceptance probability which we do not include above. The reasoning is that since both $\boldsymbol{\mathcal{S}}^*$ and $\mathcal{U}$ are discrete spaces and $f(\mathcal{S}, u)$ is a one-to-one function (since it is invertible) such terms are not required.

5.5 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.

5.5.1 Edit Allocation

Supposing $\mathcal{S}^m = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ is our current state, the main idea of this move is to allocate a number of "edits" to each path in $\mathcal{S}^m$. These edits consist of inserting and 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 6.

We now give descriptive details of this proposal generation mechanism and show how it can be cast in the light of iMCMC. First, we specify the total number of edits to be made, denoting this $\delta \in \mathbb{Z}_{\geq 1}$. Next, we specify an allocation of these edits to the paths of $\mathcal{S}^m$, denoting this $\boldsymbol{z} = (z_1, \ldots, z_N)$, where $z_i \in \mathbb{Z}_{\geq 0}$ denotes the number of edits allocated to the ith path such that $\sum_{i=1}^N z_i = \delta$. For example, in Figure 6 we have $\delta = 7$ and $\boldsymbol{z} = (3, 1, 3, 0)$.

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

The penultimate step is to specify which entries to delete and where to insert new entries, which we denote via *subsequences*. Introducing the notation $[n] = (1, \ldots, n)$, we define subsequence of $[n]$ of size m to be a vector $\boldsymbol{v} = (v_1, \ldots, v_m)$ such that $1 \leq v_1 < v_2 < \cdots < v_m \leq n$. Now, we let $\boldsymbol{v}_i$ be a subsequence of $[n_i]$ of size d_i denoting the entries of $\mathcal{I}_i$ to be deleted, whilst $\boldsymbol{v}'_i$ is subsequence of $[m_i]$ of size a_i, denoting the location of entries to be inserted in $\mathcal{I}'_i$, where $m_i = n_i - d_i + a_i$ denotes the length of $\mathcal{I}'_i$. For example, considering the first path in Figure 6 we have $\mathcal{I}_1 = (a, b, a, b, a)$ and $\mathcal{I}'_1 = (d, a, b, a)$ with $\boldsymbol{v}_1 = (4, 5)$ and $\boldsymbol{v}'_1 = (1)$ indexing the deletions and insertions respectively. The final step is to specify entries to insert, which we denote $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{ia_i})$ where $y_{ij} \in \mathcal{V}$. For example, in Figure 6 we have $\boldsymbol{y}_1 = (d)$.

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

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

then in enacting the specified edit operations, we have effectively defined the first component $f_1(\mathcal{S}^m, u) = [\mathcal{S}^m]'$. Specification of the second component is more involved, and so we delegate these details to Supplement S8.3. Regarding the auxiliary distribution $q(u|\mathcal{S}^m)$, we consider the following

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

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

Further details, including full definition of the involution f, examples of possible insertion distributions $q(\boldsymbol{y}_i|\mathcal{I}_i)$ and derivations of key terms appearing the acceptance probability (11), can be found in Supplement S8.3.

5.5.2 Path Insertion and Deletion

With this move we look to vary the outer dimension, that is, the number of paths. Similar to Section 5.5.1, we consider doing so by random deletion and insertion. The difference in this case is that we delete and insert whole paths (see Figure 7).

In particular, with $\mathcal{S}^m = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ 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 $d \in$

$\{0, \ldots, \min(N, \varepsilon)\}$ denote the number of deletions, leaving $a = \varepsilon - d$ insertions. For example, in Figure 7 we have $\varepsilon = 3$, $d = 2$ and $a = 1$. Next, we choose locations of deletions and insertions. In particular, we let $\boldsymbol{v}$ be a length d subsequence of $[N]$ denoting which paths of $\mathcal{S}^m$ are to be deleted, whilst $\boldsymbol{v}'$ is a length a subsequence of $[M]$, where $M = N - d + a$, denoting where inserted paths will be located in our proposal $[\mathcal{S}^m]'$. For example, in Figure 7 we have $\boldsymbol{v} = (2, 4)$ and $\boldsymbol{v}' = (3)$. Finally, for each $i = 1, \ldots, a$ we choose some path $\mathcal{I}_i^*$ to insert into entry v_i' of $[\mathcal{S}^m]'$. For example, in Figure 7 we have a single path $\mathcal{I}_1^* = (c, b, b, a)$ which we insert into the third entry.

As in Section 5.5.1, 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 $u = (\varepsilon, d, \boldsymbol{v}, \boldsymbol{v}', \mathcal{I}_1^*, \ldots, \mathcal{I}_a^*)$ this can similarly be seen as defining the first component of the required involution, with details of the second component found in Supplement S8.4. Regarding sampling of auxiliary variables we consider the following

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

whilst we sample $\boldsymbol{v}$ and $\boldsymbol{v}'$ uniformly and assume path insertions $\mathcal{I}_i^*$ are drawn from some general distribution over paths $q(\mathcal{I}|\mathcal{S}^m)$. This leaves two tuning parameters, ν_{td} and $q(\mathcal{I}|\mathcal{S}^m)$, which in combination facilitate control over the aggressiveness of proposals. In particular, ν_{td} controls the number of deletions and insertions attempted, whilst $q(\mathcal{I}|\mathcal{S}^m)$ affects the impact of each insertion and deletion. Again, further details can be found in Supplement S8.4.

5.6 Sampling Auxiliary Data

Both algorithms to target the conditionals outlined in Sections 5.3 and 5.4 require exact sampling of auxiliary data from appropriate interaction-sequence models. Unfortunately, we cannot do this in general. Instead, we consider replacing this with approximate samples obtained via an iMCMC algorithm.

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

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

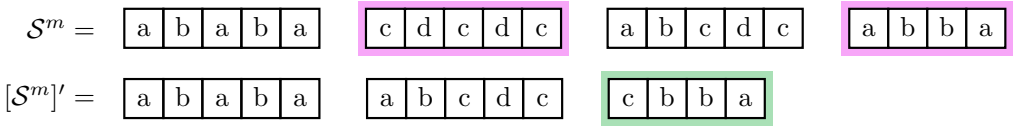


Figure 7: Illustrating path insertion and deletion move, where given the current state $\mathcal{S}^m$ the proposal $[\mathcal{S}^m]'$ is obtained by deleting and inserting the highlighted paths.

3. Evaluate the following probability

$$\alpha(\mathcal{S}, \mathcal{S}') = \min\left\{1, \frac{p(\mathcal{S}' \,|\, \mathcal{S}^m, \gamma) q(u' \,|\, \mathcal{S}')}{p(\mathcal{S} \,|\, \mathcal{S}^m, \gamma) q(u \,|\, \mathcal{S})}\right\} \tag{12}$$

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 5.4, again mixing these together with some proportion $\beta \in (0, 1)$. Note, as in Section 5.4, 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 the 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 5.3 and 5.4 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.

6 Simulation Studies

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

6.1 Posterior Concentration

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

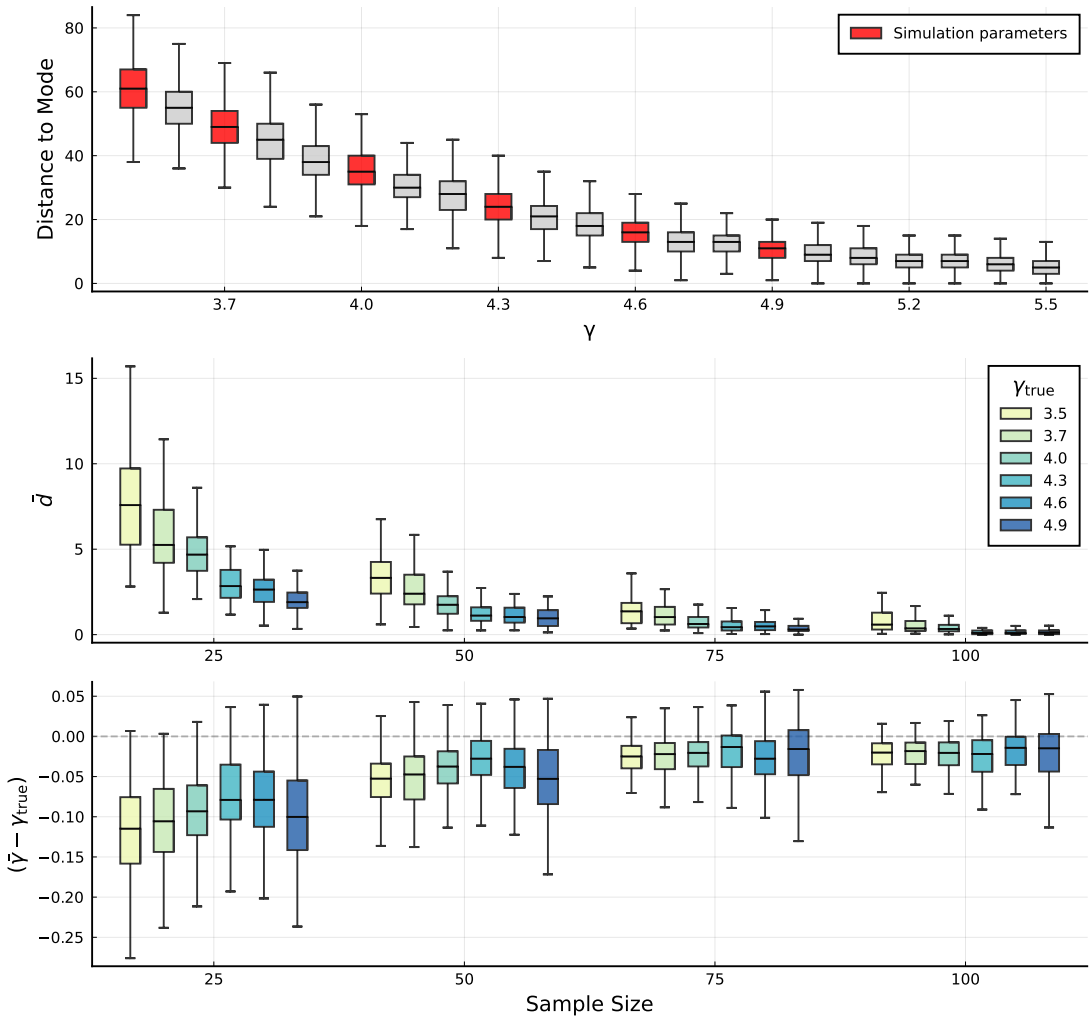


Figure 8: A summary of our first simulation (Section 6.1), where the top plot visualises the scale of the SIS model used therein, in particular, for different values of γ it shows $\{d_S(\mathcal{S}^{(i)}, \mathcal{S}^m_{\text{true}})\}_{i=1}^{1000}$ where $\mathcal{S}^{(i)} \sim \text{SIS}(\mathcal{S}^m_{\text{true}}, \gamma)$, sampled via the iMCMC scheme of Section 5.6. The remaining two plots summarise simulation outputs for each pair $(\gamma_{\text{true}}, n)$, where the middle shows distributions of $\bar{d}$, the average distance to the true mode, whilst the bottom shows $(\bar{\gamma} - \gamma_{\text{true}})$, the bias of the dispersion posterior mean relative to the truth.

The high-level approach is the following. Given true mode $\mathcal{S}^m_{\text{true}}$ and dispersion γ_{true}, we draw a sample $\{\mathcal{S}^{(i)}\}_{i=1}^n$ where

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

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

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

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

Now, recall the dispersion works inversely to the variance, in that lower values lead to more variable data (Figure 8, top). Intuitively, when the data is more variable it will be harder to discern the true mode $\mathcal{S}^m_{\text{true}}$, and thus we expect $\bar{d}$ to decrease more slowly for lower values of γ_{true}. Alternatively, as can be seen in Figure 8, when γ_{true} is smaller the difference of their parameterised distributions (as described by the distribution of distances to the mode) becomes more marked relative to neighbouring values. As such, we might also expect smaller values for the dispersion to be easier to recover.

To explore for such properties, we varied γ_{true} and n whilst keeping $\mathcal{S}^m_{\text{true}}$ fixed. In particular, we considered $\gamma_{\text{true}} = 3.5, 3.7, 4.0, 4.3, 4.6, 4.9$ (highlighted in Figure 8, 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 defined in Appendix A.2, assuming at most $L = 20$ paths in any observation, with each path being at most length $K = 10$.

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

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

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

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

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

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

Now, for each pair $(\gamma_{\text{true}}, n)$ we (i) sampled n observations from an $\text{SIS}(\mathcal{S}^m_{\text{true}}, \gamma_{\text{true}})$ model, using the iMCMC scheme outlined in Section 5.6, 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 5.2, with a burn-in period of 25,000 and taking a lag of 100 between samples[2] (iii) evaluated summary measures $\bar{d}$ and $\bar{\gamma}$.

We repeated (i)-(iii) 100 times in each case, the results of which are summarised in Figure 8. Consulting the middle plot, we observe that $\bar{d}$ 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 $\bar{\gamma}$ relative to the truth. Note this is expected since we have used approximate MCMC samples within our component-wise scheme of Section 5.2. We do, however, see a reduction in this bias as the sample size grows. Furthermore, for the larger values of n we begin to see a clearer difference in the variance of $\bar{\gamma}$ across different values of γ_{true}. In particular, the variance appears to be smaller for lower values of γ_{true}, agreeing with the intuition that these are easier to estimate.

6.2 Effect of Mode Structure

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

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

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

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

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

which thus implies, for each $\mathcal{S}$, a sample $\{k_{\mathcal{S}}(v) : v \in \mathcal{V},\, k_{\mathcal{S}}(v) > 0\}$, similar in spirit to the degree distribution. Now, α can be seen to control the heavy-tailedness of this distribution (see Figure 9), 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 $\alpha = -\tilde{\alpha}$ where $\tilde{\alpha} = 1.35, 0.75, 0.35, 0.12, 0.06, 0.03, 0.01$ and $n = 25, 50, 75, 100$. Details on how these α values were chosen can be found in Supplement S10. 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 10 summarises the output of 100 repetitions for each pair (α, n). For each α, we see values for $\bar{d}$ closer to zero as n grows, indicating concentration about the true mode. Furthermore, α shows no clear sign of impacting this concentration. Regarding the dispersion posterior mean $\bar{\gamma}$, as in the previous simulation, we observe bias relative to the truth, with this bias reducing as n grows. Furthermore, this is the same across all α, with no clear sign that α affects the inference of these values.

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

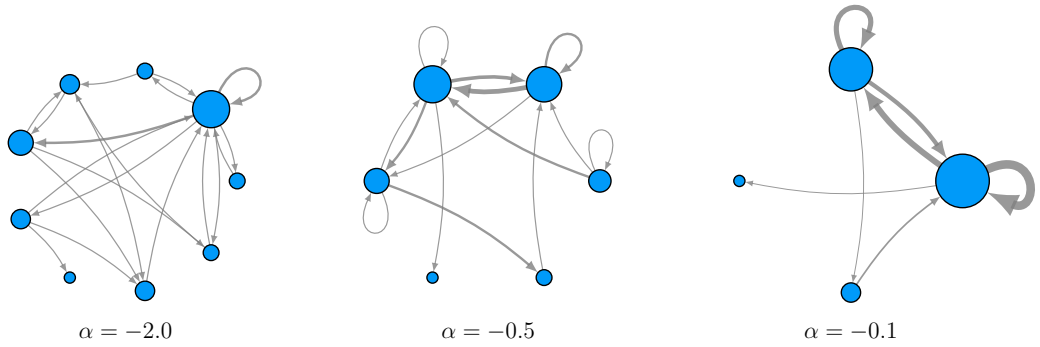


Figure 9: Visualising the role of α in the Hollywood model. Each plot shows an aggregate multigraph $\mathcal{G}_{\mathcal{S}}$ where $\mathcal{S} \sim \text{Hollywood}(\alpha, -\alpha V, \nu)$ with $V = 10$, $\nu = \text{TrPoisson}(3, 1, 10)$ and α varying. Edge thickness reflects edge multiplicity, whilst vertex size is proportional to $k_{\mathcal{S}}(v)$.

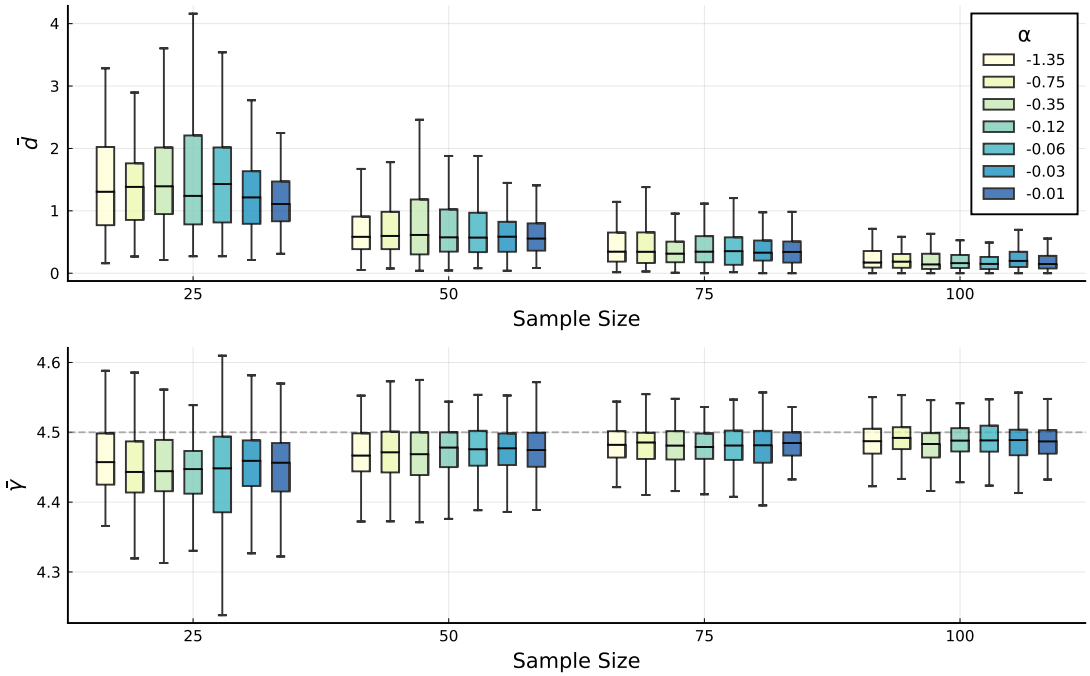


Figure 10: Summary of our second simulation (Section 6.2), where for each pair (α, n) the top subplot shows the distribution of $\bar{d}$, the average distance to the true mode, whilst the bottom shows the distribution of $\bar{\gamma}$, the posterior mean dispersion.

6.3 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

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

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

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

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

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

$$\hat{p}(x|\{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x}) := \frac{1}{m} \sum_{i=1}^m p(x|\mathcal{S}_i^m, \gamma_i, \mathcal{S}_{-x}), \tag{13}$$

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

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

In this simulation, we considered assessing the agreement of the true and posterior predictive as n grows by examining how often their predictions were equal. We adopted the same modelling set-up as Section 6.1, 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.

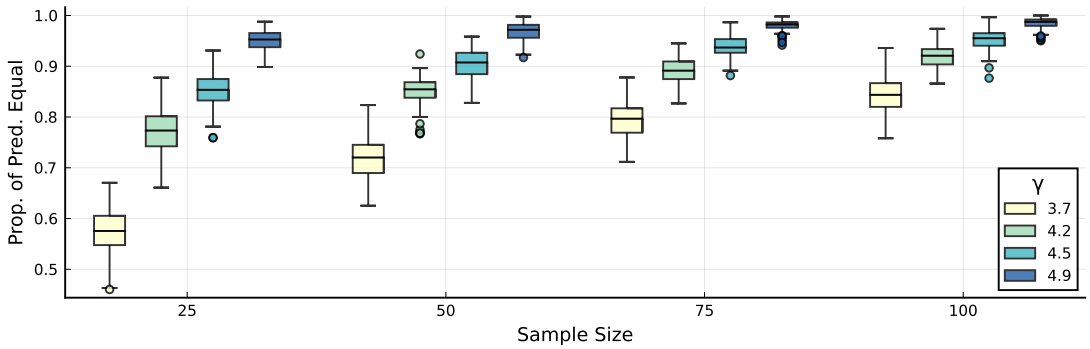


Figure 11: Summary of posterior predictive simulation (Section 6.3). Here we summarise the proportion of times the true and posterior predictions coincided when predicting missing entries of sampled test data, with boxplots showing the distribution of these proportions over 100 repetitions.

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 6.1 and 6.2) (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 $i = n+1, \ldots, n+n_{\text{test}}$ and for each entry of $\mathcal{S}^{(i)}$ (that is, each entry of each interaction) we assumed it to be missing and obtained predictions with both $\hat{p}(x | \{\mathcal{S}^{(i)}\}_{i=1}^n, \mathcal{S}_{-x})$ and $p(x | \mathcal{S}^m, \mathcal{S}_{-x})$ via MAP estimates, and finally (v) returned the proportion of times these predictions were equal.[3] For (ii) and (iii) we used the same MCMC schemes as previous simulations.

Figure 11 summarises the output of 100 repetitions for each pair $(\gamma_{\text{true}}, n)$, with $n_{\text{test}} = 100$ in each repetition. For each γ_{true}, we see the predictions of the posterior predictive are more often in accordance with those of the true predictive distribution as the number of training samples increases. Moreover, when γ is lower, that is, the observed data is more variable, the discrepancy between the true and posterior predictive tends to be larger. Observe this is expected, given the observed behaviour of the first posterior concentration simulation (Section 6.1), wherein the posterior concentrated more slowly when γ was lower. In summary, the posterior predictive appears to better resemble the true data-generating distribution as the sample size grows, as was expected.

7 Data Analysis

In this section, the applicability of the proposed methodology will be illustrated via an example analysis of the Foursquare check-in data 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

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

some graph-based estimates. Note that in aggregating observations to form graphs one implicitly assumes that the order of interaction arrival is irrelevant. Hence, for fairness, we opt to make this comparison with our SIM model.

7.1 Data Background and Processing

For this analysis, we consider a version of the Foursquare data containing check-ins for users from New York and Tokyo, focusing in particular on those in New York.[4] From there, we took a month of check-ins, over the period from 12 April to 12 May 2012. 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; "Arts & Entertainment", in the given example.

Before proceeding with our analysis, we further filtered the data. Firstly, we ignore any days when a user checks into a single venue. Since our analysis is based on interaction multisets and concerns the movements of users *between* venue categories, such observations provide little information. They would also 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. 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 50 to analyse, using a criterion based upon the distance metric used in our model fit (details in Supplement S11.1). In this final sample, the number of paths within each observed network ranged from 10 to 17, with a mean of 13.44, whilst the path lengths within each network ranged from 2 to 11, with a mean of approximately 2.97.

7.2 SIM Model Fit

Following data processing, we were left with a sample of multisets $\{\mathcal{E}^{(i)}\}_{i=1}^{n}$, where each $\mathcal{E}^{(i)} = \left\{\mathcal{I}_1^{(i)}, \ldots, \mathcal{I}_{N^{(i)}}^{(i)}\right\}$ denotes the data of the ith user, with $\mathcal{I}_j^{(i)}$ denoting a single day of their check-ins. Recalling the inferential questions of interest outlined in Section 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 S9, we fit our SIM model to these data. For our distance, we made use of the matching distance d_{M}, with the longest common subpath distance d_{LSP} between paths. Consequently, our inferred mode will contain paths often appearing as subpaths in the observed data, as discussed in Section 4.5. For our priors, we assumed $\mathcal{E}^m \sim \mathrm{SIM}(\hat{\mathcal{E}}, 3.0)$, with $\hat{\mathcal{E}}$ denoting the sample Fréchet mean of the observed data $\{\mathcal{E}^{(i)}\}_{i=1}^{n}$, whilst we assumed $\gamma \sim \mathrm{Gamma}(5, 1.67)$.

4. See here `https://sites.google.com/site/yangdingqi/home/foursquare-dataset`.

Via our MCMC scheme, we then obtained a sample $\{(\mathcal{E}_i^m, \gamma_i)\}_{i=1}^M$, from the posterior $p(\mathcal{E}^m, \gamma | \{\mathcal{E}^{(i)}\}_{i=1}^n)$, obtaining a total of 100,000 samples. In each iteration, when sampling the 50 auxiliary data points, we took a lag of 50 between and discarded the first 4,000 as burn-in. From the 100,000 posterior samples, we discarded the first half as burn-in, and took a lag of 50 between samples, leaving a final $M = 1000$ samples.

The total run time for obtaining these posterior samples was approximately 18 hours, corresponding to an average of around 0.65 seconds per sample. This was implemented on a Dell Latitude 5440 laptop, with a 13th Gen Intel Core i7-1370P processor and 64 GB of RAM. As discussed in Supplement S6, a major contributor to this cost is the sampling of auxiliary data. In this present analysis, after discarding 4,000 burn-in samples and applying a thinning lag of 50, we obtained 6,500 auxiliary samples per iteration. A time of 0.65 seconds to obtain 6,500 auxiliary samples would equate to 0.1 seconds to generate 1,000 auxiliary samples, which is consistent with the results reported in Supplement S6.2, where, on the same machine, we studied the time to obtain 1,000 samples from our SIM model in a setup similar to this current analysis.

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

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

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

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

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

Figures 12 to 14 visualise the paths of $\hat{\mathcal{E}}^m$, $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$ alongside one another. In each, the paths of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$ have been aligned in accordance with the optimal matching found when evaluating their distance from $\hat{\mathcal{E}}^m$ via d_{M}. In particular, in the jth row we plot $\hat{\mathcal{I}}_j^m$ alongside $\mathcal{I}_j^{(1)}$ and $\mathcal{I}_j^{(2)}$, denoting the paths matched to $\hat{\mathcal{I}}_j^m$ when evaluating $d_{\mathrm{M}}(\hat{\mathcal{E}}^m, \mathcal{E}^{(1)})$ and $d_{\mathrm{M}}(\hat{\mathcal{E}}^m, \mathcal{E}^{(2)})$, respectively. The paths of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$ not matched to any of $\hat{\mathcal{E}}^m$ are then shown in the remaining rows.

Here one can observe paths of $\hat{\mathcal{E}}^m$ do indeed appear as subpaths within those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$. In fact, in first two rows of Figure 12, all are equivalent, that is $\hat{\mathcal{I}}_j^m = \mathcal{I}_j^{(1)} = \mathcal{I}_j^{(2)}$, whilst for the remaining rows of Figure 12 and those of Figures 13 and 14 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, and we even uncover paths of

length one. The general pattern here is that food venues seem to be a frequent appearance in many daily check-ins within this sample of data. In some cases, this is followed by a subsequent check-in to another food venue, or a shopping venue (length two paths). In other cases, the visit to a food venue appears somewhere within a series of check-ins (length one paths).

For the dispersion, we have $\hat{\gamma} \approx 2.68$, 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 15. To aid the interpretation of $\hat{\gamma}$, the right-hand plot of Figure 15 visualises the distribution of $d_{\mathrm{M}}(\mathcal{E}, \hat{\mathcal{E}}^m)$ where $\mathcal{E} \sim \mathrm{SIM}(\hat{\mathcal{E}}^m, \gamma)$ for different values of γ, each boxplot summarising 1,000 samples drawn from the respective multiset model via our iMCMC algorithm (Supplement S9.5). A comparison with our estimate $\hat{\gamma}$ shows that we expect the distance of samples to the mode to be around 25 (from $\gamma = 2.7$), 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 estimated mode has 16 entries in total (6 paths of length two, 4 of length one), this implies a reasonable amount of variability in the observed data.

7.3 Comparison with Graph-Based Inferences

As alluded to already, one can analyse these data via current network-based methods by first converting observations to graphs, before applying suitable graph-based approaches. As such, we consider striking a comparison between this approach and ours. The intention here is twofold. On one hand, to sanity-check our estimate, by confirming the graph-based inferences are not too dissimilar from ours. Whilst on the other, to illustrate how our approach can go beyond these graph-based methods, particularly in regards to the conclusions one can reasonably draw.

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 a multigraph.

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

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

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 $A^{\mathcal{G}}_{ij}$ 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 $\hat{\mathcal{G}}_{\mathrm{MV}}$, where we include an edge if it was observed in at least one-half of the observations. More formally, $\hat{\mathcal{G}}_{\mathrm{MV}}$ can be defined

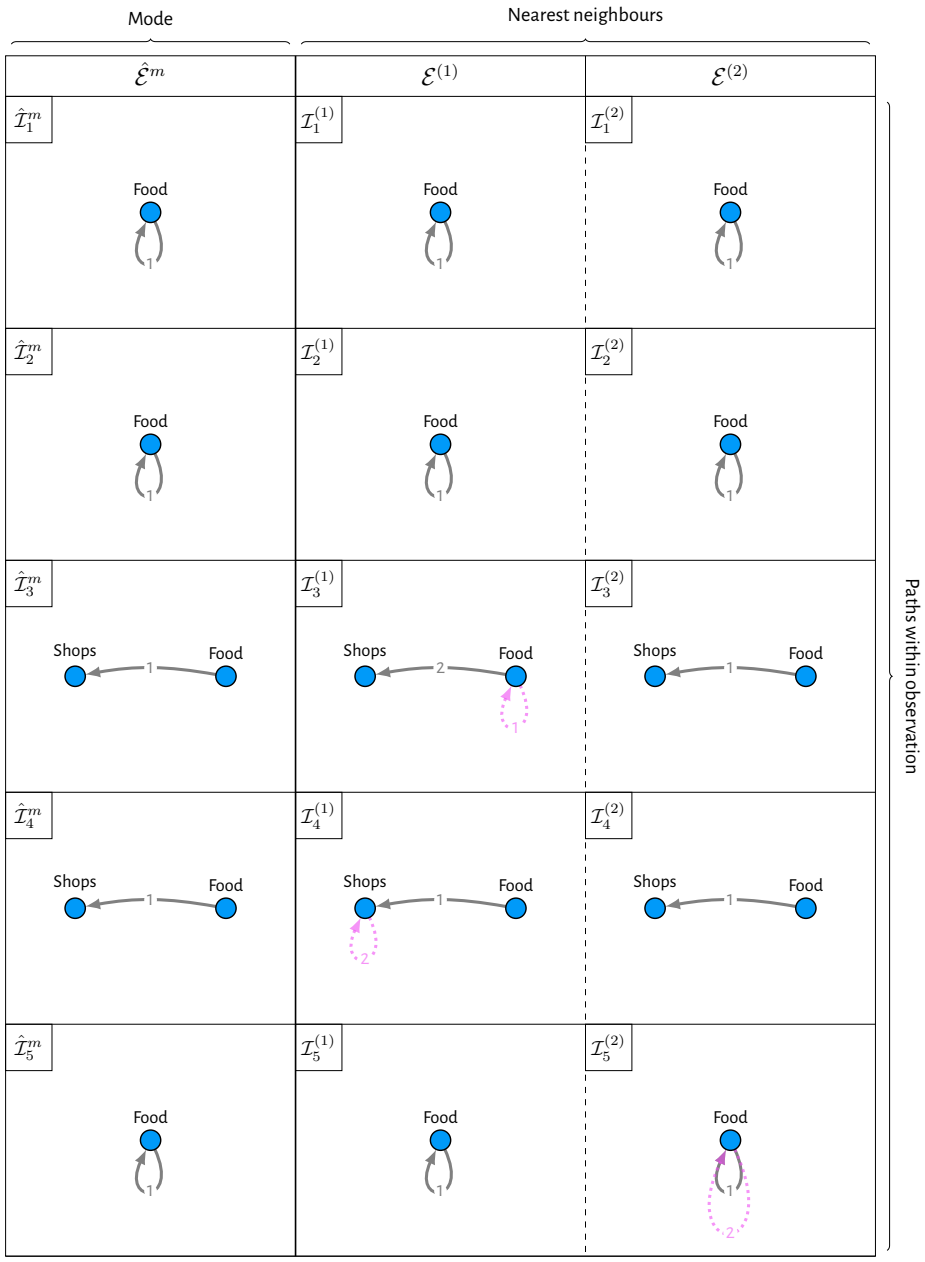


Figure 12: A subset of paths from our point estimate $\hat{\mathcal{E}}^m$ for the Foursquare data, alongside those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$, its two nearest neighbours. Paths are aligned according to the optimal matching found when evaluating $d_{\mathrm{M}}(\hat{\mathcal{E}}^m, \mathcal{E}^{(i)})$ for each neighbour $\mathcal{E}^{(i)}$. For each observed path $\mathcal{I}_j^{(i)}$, dashed pink edges and pink vertices indicate differences with $\hat{\mathcal{I}}_j^m$, with edges labels indicating the order of vertex visits. The remaining paths can be seen in Figures 13 and 14.

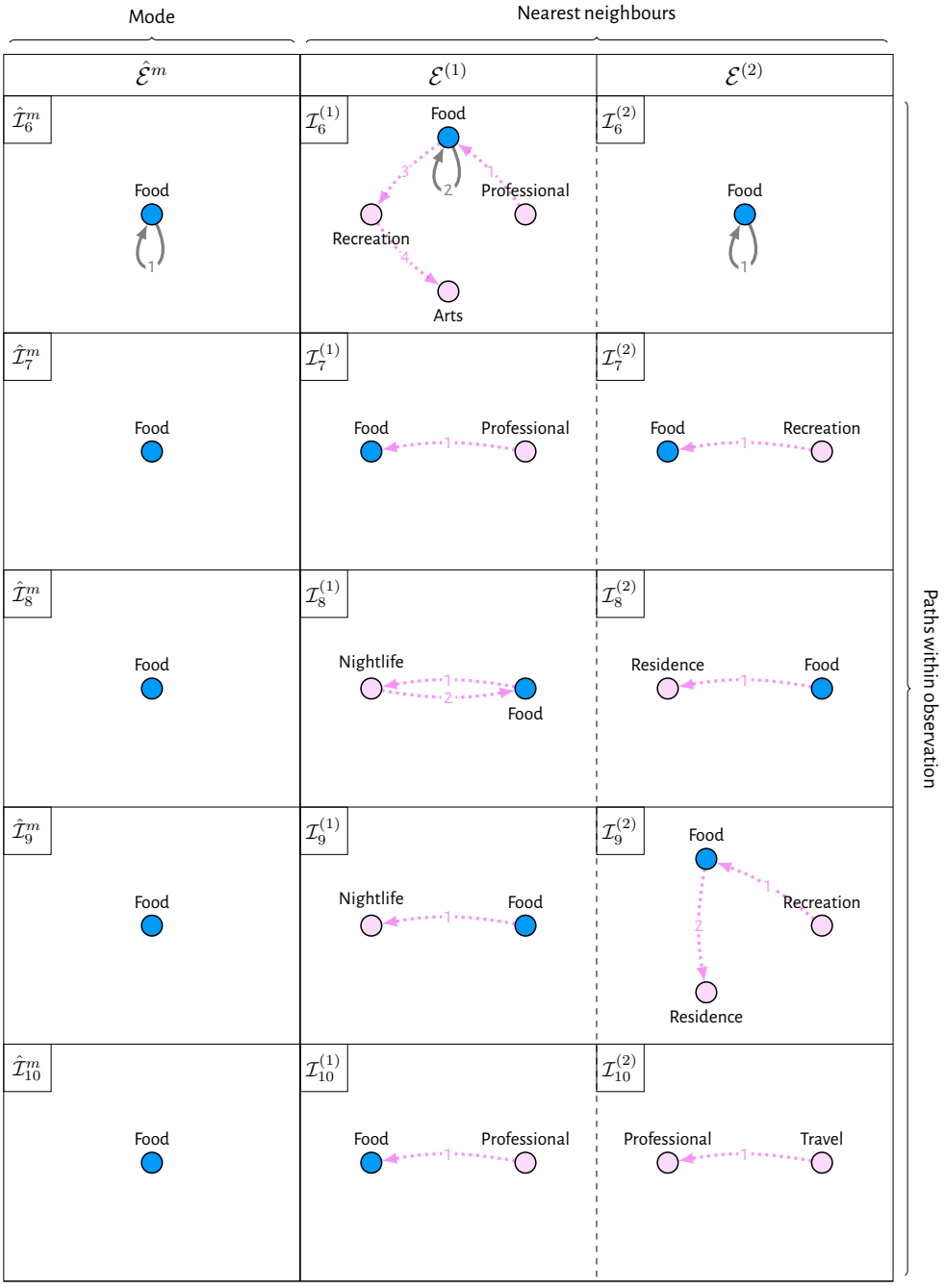


Figure 13: Paths of our point estimate $\hat{\mathcal{E}}^m$ for the Foursquare data, alongside those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$, its two nearest neighbours. The remaining paths can be seen in Figures 12 and 14.

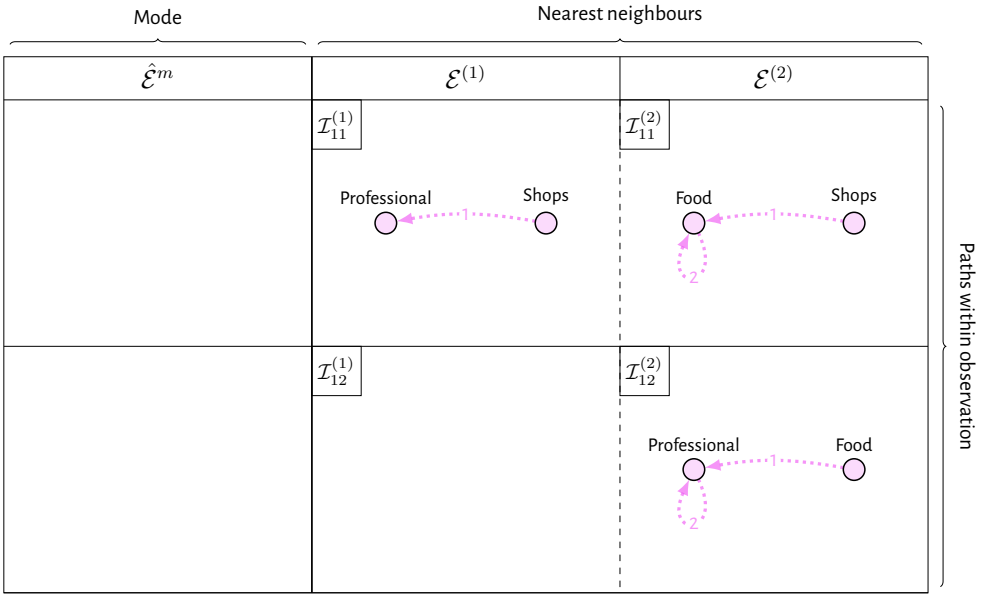


Figure 14: Paths of our point estimate $\hat{\mathcal{E}}^m$ for the Foursquare data, alongside those of $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$, its two nearest neighbours. The remaining paths can be seen in Figures 12 and 13.

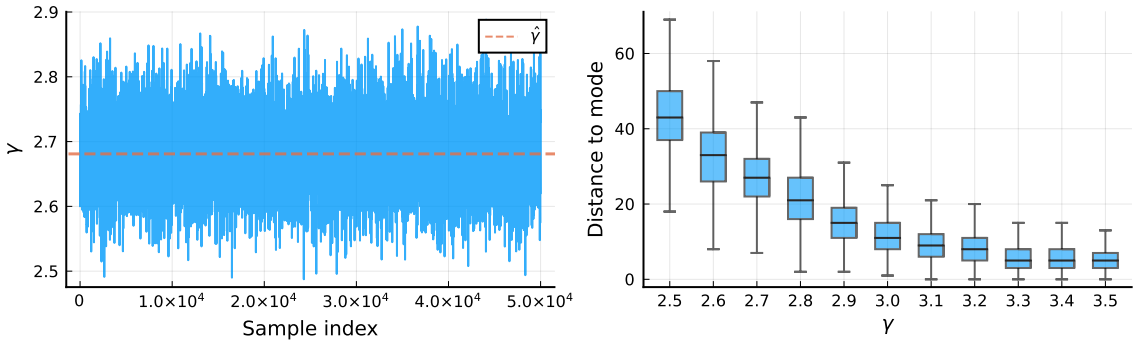


Figure 15: Summary of inference for the dispersion for the Foursquare data. Left shows a trace-plot of the posterior samples $\{\gamma_i\}_{i=1}^m$, whilst the right plot summarises the distribution of distances to the inferred mode for different values of γ, aiding interpretation of our estimate $\hat{\gamma}$.

in terms of its adjacency matrix as follows

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

where $\bar{A}$ is the real-valued matrix with entries $\bar{A}_{ij} = \frac{1}{n}\sum_{k=1}^{n} A_{ij}^{\mathcal{G}^{(k)}}$, that is, the entry-wise average of the observed adjacency matrices. As a model-based alternative, we turn to the centered Erdös-Rényi (CER) model of Lunagómez et al. (2021). 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

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

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

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

where d_{H} denotes the Hamming distance between graphs (Lunagómez et al., 2021; Donnat and Holmes, 2018). Figures 16a and 16b 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 $\bar{A}$ to the nearest integer. Referring to this as the rounded mean estimate and denoting it $\hat{\mathcal{G}}_{\text{RM}}$, it can be defined formally via its adjacency matrix as follows

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

where the notation $\lfloor x \rfloor$ for $x \in \mathbb{R}$ denotes the floor function. As a model-based approach, we consider using the SNF models proposed by Lunagómez et al. (2021). Though originally proposed to model graphs, they can be readily extended to handle multigraphs (see Supplement S11.2). Use of the SNF, like our models, requires the specification of a distance metric between multigraphs. We considered taking the absolute difference of edge multiplicities, or alternatively, the adjacency matrix entries, that is

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

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

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

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

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

Note that the posterior here will be doubly intractable, necessitating the use of a specialised MCMC algorithm. Lunagómez et al. (2021) adopted the algorithm of Møller et al. (2006), however, since here we consider multigraphs, we cannot apply their scheme directly. Instead, we took an alternative approach via the exchange algorithm (Murray et al., 2006), details of which can be found in Supplement S11.2. Visualisations of these two multigraph summaries, $\hat{\mathcal{G}}_{\text{SNF}}$ and $\hat{\mathcal{G}}_{\text{RM}}$, can be seen in Figures 16c and 16d, 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 $\hat{\mathcal{G}}_{\text{MV}}$ and $\hat{\mathcal{G}}_{\text{CER}}$, 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 $\hat{\mathcal{G}}_{\text{MV}}$.

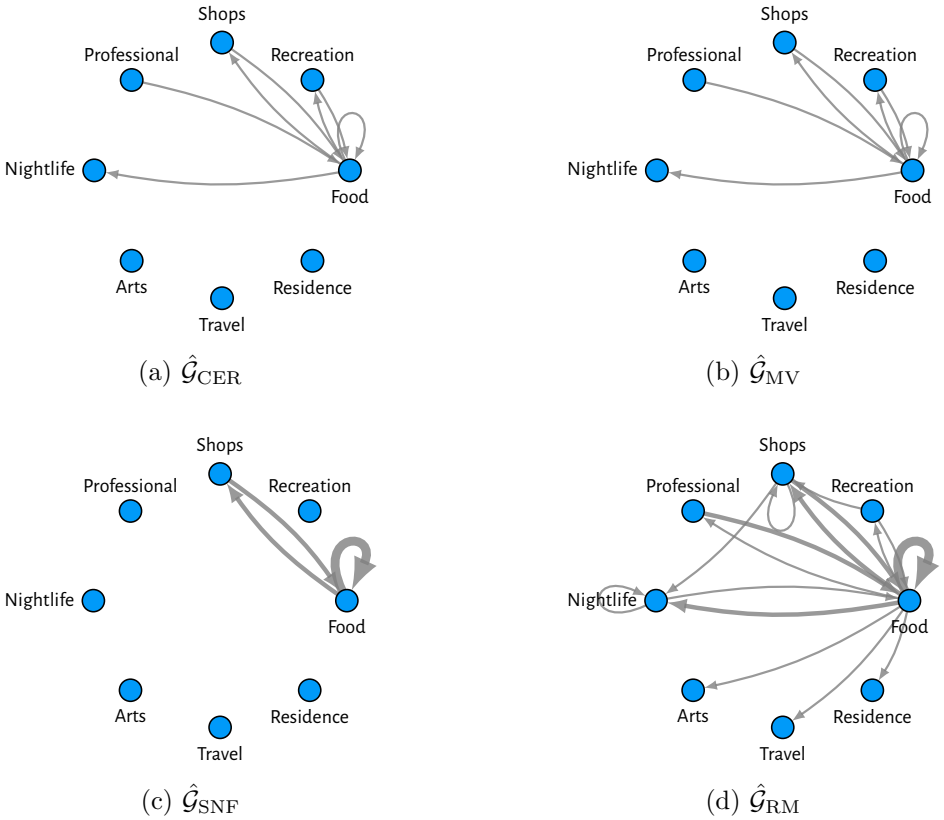
Nonetheless, a common theme does seem to appear: visits to food venues feature strongly, often followed or preceded by a visit to another food venue or some other venue category, with shopping venues being a prevalent choice. We observe that this overarching pattern is also in line with that implied by our estimate $\hat{\mathcal{E}}^m$, where we also uncovered the centrality of food venues and their precedence of visits to shopping venues.

Naturally, one might ask, if our estimate $\hat{\mathcal{E}}^m$ is not too dissimilar to those obtained via graph-based methods, then what is gained by taking our approach? The crucial point here is that our inferred object (a multiset of paths) is more complex, and consequently contains more information. Graphs by construction represent only second-order information, for example, "user x moved from venue A to venue B." In contrast, our representation can represent higher, or oven lower, orders of information, for example, via paths of length greater or less than two,

Indeed, recall in the present analysis (Figures 12 to 14), we uncovered paths of length one, highlighting how food venues often appeared within chains of check-ins. Such observations cannot be made with a graph-based approach. More subtly, consider the CER-based summary $\hat{\mathcal{G}}_{\text{CER}}$ of Figure 16a, where we see the following two edges

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

This could imply at least two things: (a) 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



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

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

Figure 16: Visualisation of graph-based inferences, as alternatives to ours in Figure 12. Note that (a) and (b) are graphs, whilst (c) and (d) are multigraphs, with edge thickness proportional to their weight.

venue, or (b) many users traced the path recreation → food → shops in a single day. From a graph-based summary, it is not possible to distinguish between these two possibilities. However, with our estimate, such distinctions can be made. For example, in this analysis, appears (a) is the case, since no paths within $\hat{\mathcal{E}}^m$ are of length greater than two.

8 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. Two distances for comparing interaction networks have also been proposed, for use within these models. Each distance is shown to be a metric, under certain conditions, and methods for their computation have been discussed. To facilitate parameter inference, specialised MCMC schemes have also been proposed. 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 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 Lunagómez 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 occupation 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).

References

Arroyo, J., Athreya, A., Cape, J., Chen, G., Priebe, C. E., and Vogelstein, J. T. (2021). Inference for multiple heterogeneous networks with a common invariant subspace. *Journal of machine learning research*, 22(142).

Behrens, T. E. and Sporns, O. (2012). Human connectomics. *Current Opinion in Neurobiology*, 22:144–153.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Cai, D., Campbell, T., and Broderick, T. (2016). Edge-exchangeable graphs and sparsity. *Advances in Neural Information Processing Systems*, 29.

Caron, F. and Fox, E. B. (2017). Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(5):1295–1366.

Chen, L., Zhou, J., and Lin, L. (2021). Hypothesis testing for populations of networks. *Communications in Statistics-Theory and Methods*, pages 1–24.

Chung, J., Bridgeford, E., Arroyo, J., Pedigo, B. D., Saad-Eldin, A., Gopalakrishnan, V., Xiang, L., Priebe, C. E., and Vogelstein, J. T. (2021). Statistical connectomics. *Annual Review of Statistics and Its Application*, 8:463–492.

Crane, H. and Dempsey, W. (2018). Edge exchangeable models for interaction networks. *Journal of the American Statistical Association*, 113:1311–1326.

Donnat, C. and Holmes, S. (2018). Tracking network dynamics: A survey using graph distances. *Annals of Applied Statistics*, 12:971–1012.

Durante, D. and Dunson, D. B. (2018). Bayesian inference and testing of group differences in brain networks. *Bayesian Analysis*, 13(1):29–58.

Durante, D., Dunson, D. B., and Vogelstein, J. T. (2017). Nonparametric Bayes modeling of populations of networks. *Journal of the American Statistical Association*, 112:1516–1530.

Eiter, T. and Mannila, H. (1997). Distance measures for point sets and their computation. *Acta informatica*, 34(2):109–133.

Fearnhead, P., Nemeth, C., Oates, C. J., and Sherlock, C. (2025). *Scalable Monte Carlo for Bayesian Learning*. Institute of Mathematical Statistics Monographs. Cambridge University Press.

Fox, K. and Li, X. (2019). Approximating the geometric edit distance. *Leibniz International Proceedings in Informatics, LIPIcs*, 149.

Frank, O. and Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association*, 81(395):832–842.

Ghalebi, E., Mahyar, H., Grosu, R., Taylor, G. W., and Williamson, S. A. (2019a). A nonparametric Bayesian model for sparse dynamic multigraphs. *arXiv preprint arXiv:1910.05098*.

Ghalebi, E., Mahyar, H., Grosu, R., and Williamson, S. (2019b). Dynamic nonparametric edge-clustering model for time-evolving sparse networks. *arXiv preprint arXiv:1905.11724*.

Ghoshdastidar, D., Gutzeit, M., Carpentier, A., and Von Luxburg, U. (2020). Two-sample hypothesis testing for inhomogeneous random graphs. *The Annals of Statistics*, 48(4):2208–2229.

Ginestet, C. E., Li, J., Balachandran, P., Rosenberg, S., and Kolaczyk, E. D. (2017). Hypothesis testing for network data in functional neuroimaging. *Annals of Applied Statistics*, 11:725–750.

Gold, O. and Sharir, M. (2018). Dynamic time warping and geometric edit distance: breaking the quadratic barrier. *ACM Transactions on Algorithms*, 14.

Gollini, I. and Murphy, T. B. (2016). Joint modeling of multiple network views. *Journal of Computational and Graphical Statistics*, 25:246–265.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098.

Holland, P. W. and Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Le, C. M., Levin, K., and Levina, E. (2018). Estimating a network from multiple noisy realizations. *Electronic Journal of Statistics*, 12:4697–4740.

Lehmann, B. and White, S. (2021). Bayesian exponential random graph models for populations of networks. *arXiv preprint arXiv:2104.05110*.

Levin, K., Athreya, A., Tang, M., Lyzinski, V., and Priebe, C. E. (2017). A central limit theorem for an omnibus embedding of multiple random dot product graphs. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 964–967. IEEE.

Liang, F. (2010). A double Metropolis-Hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computation and Simulation*, 80:1007–1022.

Lunagómez, S., Olhede, S. C., and Wolfe, P. J. (2021). Modeling network populations via graph distances. *Journal of the American Statistical Association*, 116(536):2023–2040.

Mantziou, A., Lunagomez, S., and Mitra, R. (2021). Bayesian model-based clustering for multiple network data. *arXiv preprint arXiv:2107.03431*.

Mardia, K. V. and Dryden, I. L. (1999). The complex Watson distribution and shape analysis. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 61:913–926.

Marron, J. S. and Dryden, I. L. (2021). *Object Oriented Data Analysis (1st ed.)*. Chapman and Hall/CRC.

Murray, I., Ghahramani, Z., and MacKay, D. J. (2006). Mcmc for doubly-intractable distributions. *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, UAI 2006*, pages 359–366.

Møller, J., Pettitt, A. N., Reeves, R., and Berthelsen, K. K. (2006). An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93:451–458.

Neklyudov, K., Welling, M., Egorov, E., and Vetrov, D. (2020). Involutive mcmc: A unifying framework. In *International Conference on Machine Learning*, pages 7273–7282. PMLR.

Newman, M. E. (2018). Estimating network structure from unreliable measurements. *Physical Review E*, 98(6):062321.

Nielsen, A. M. and Witten, D. (2018). The multiple random dot product graph model. *arXiv preprint arXiv:1811.12172*.

Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087.

Peixoto, T. P. (2018). Reconstructing networks with unknown and heterogeneous errors. *Physical Review X*, 8(4):041011.

Peixoto, T. P. and Rosvall, M. (2017). Modelling sequences and temporal networks with dynamic community structures. *Nature Communications*, 8(1):1–12.

Ramon, J. and Bruynooghe, M. (2001). A polynomial time computable metric between point sets. *Acta Informatica*, 37:765–780.

Reyes, P. and Rodriguez, A. (2016). Stochastic blockmodels for exchangeable collections of networks. *arXiv preprint arXiv:1606.05277*.

Robert, C. P., Casella, G., and Casella, G. (1999). *Monte Carlo statistical methods*, volume 2. Springer.

Scholtes, I. (2017). 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.

Stanley, N., Shai, S., Taylor, D., and Mucha, P. J. (2016). Clustering network layers with the strata multilayer stochastic block model. *IEEE Transactions on Network Science and Engineering*, 3(2):95–105.

Sweet, T. M., Thomas, A. C., and Junker, B. W. (2013). Hierarchical network models for education research: Hierarchical latent space models. *Journal of Educational and Behavioral Statistics*, 38(3):295–318.

Sweet, T. M., Thomas, A. C., and Junker, B. W. (2014). Hierarchical mixed membership stochastic blockmodels for multiple networks and experimental interventions. *Handbook on Mixed Membership Models and their Applications*, pages 463–488.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581.

Veitch, V. and Roy, D. M. (2015). The class of random graphs arising from exchangeable random measures. *arXiv preprint arXiv:1512.03099*.

Vitelli, V., Øystein Sørensen, Crispino, M., Frigessi, A., and Arjas, E. (2018). Probabilistic preference learning with the Mallows rank model. *Journal of Machine Learning Research*, 18:1–49.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM (JACM)*, 21:168–173.

Wang, S., Arroyo, J., Vogelstein, J. T., and Priebe, C. E. (2019). Joint embedding of graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Williamson, S. A. (2016). Nonparametric network models for link prediction. *The Journal of Machine Learning Research*, 17(1):7102–7121.

Yang, D., Zhang, D., Zheng, V. W., and Yu, Z. (2015). Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45:129–142.

Yin, F., Shen, W., and Butts, C. T. (2022). Finite mixtures of ergms for modeling ensembles of networks. *Bayesian Analysis*, 1(1):1–39.

Young, J.-G., Kirkley, A., and Newman, M. (2022). Clustering of heterogeneous populations of networks. *Physical Review E*, 105(1):014312.

Young, S. J. and Scheinerman, E. R. (2007). Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer.

Appendix A. Sample Spaces

In this section, we formally define the sample spaces of the SIS and SIM models introduced in Section 3.1. In addition, we define the some finite versions thereof obtained by bounding dimensions, which we recommend working with in practice. For further elaboration on this recommendation, including justifications of the rationale, an outline of how to alter our MCMC algorithms to ensure the bound constraints are met, and discussions on how to choose the bounds in practice, see Supplement S3.

A.1 Infinite Spaces

Recall from Definitions 1 and 2 that the SIS and SIM models define distributions over the spaces of *all* interaction sequences and multisets, respectively. Given the vertex set $\mathcal{V}$ we first define the space of all interactions, that is, paths, as follows

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

with which we can define the space of interaction sequences $\boldsymbol{\mathcal{S}}^*$ in the following manner

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

moreover, with $\mathcal{E}_{\mathcal{S}}$ denoting the multiset obtained from the sequence $\mathcal{S}$ by disregarding the order of paths therein, the space of interaction multisets $\boldsymbol{\mathcal{E}}^*$ can be defined as follows

$$\boldsymbol{\mathcal{E}}^* := \{\mathcal{E}_{\mathcal{S}} \,:\, \mathcal{S} \in \boldsymbol{\mathcal{S}}^*\}$$

where here we abuse notation slightly, since we can have $\mathcal{E}_{\mathcal{S}} = \mathcal{E}_{\mathcal{S}'}$ for $\mathcal{S} \neq \mathcal{S}'$ (when equal up to a permutation of interactions), but we just assume such values have been included once and so $\boldsymbol{\mathcal{E}}^*$ is a set and not a multiset.

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

$$\mathcal{E} = \{\mathcal{S} \in \boldsymbol{\mathcal{S}}^* \,:\, \mathcal{S} \overset{p}{\sim} \tilde{\mathcal{S}}\}$$

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

A.2 Bounded Spaces

In this section, we define bounded analogues of the infinite spaces introduced the in preceding section. With regards to the objects we consider, there are two things we can bound: (i) the size of paths and (ii) the number of paths. Referring to these as the inner and outer dimensions respectively, we specify two integers K and L bounding their values and define our sample spaces accordingly. Assuming that the vertex set $\mathcal{V}$ is fixed, and $K \in \mathbb{Z}_{\geq 1}$ we let

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

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

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

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

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

where as in the definition of $\boldsymbol{\mathcal{E}}^*$ in Appendix A.1 one can have $\mathcal{E}_{\mathcal{S}} = \mathcal{E}_{\mathcal{S}'}$ for $\mathcal{S} \neq \mathcal{S}'$, but we here just assume such values have been included once, and so $\boldsymbol{\mathcal{E}}^*_{K,L}$ is indeed a set, not a multiset.

Appendix B. Posterior Predictive Approximation

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 6.3, Equation (13). First, observe that any sample $\{(\mathcal{S}^m_i, \gamma_i)\}_{i=1}^m$ from the posterior implies the following atomic approximation thereof

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

where $\delta(\cdot)$ is the Dirac delta function, and $\mathbb{1}(\cdot)$ is the identity function.

As in Section 6.3, with $\mathcal{S}_x$ denoting the observation with missing entry filled in to be x, then given some parameters $(\mathcal{S}^m, \gamma)$ we have the true predictive for x given by

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

with

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

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

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

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

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

which is exactly as stated in Section 6.3.

Supplementary Material

S1 Unsuitability of Fixed Penalties

In Section 4.4 we claimed the use of a fixed penalty within the edit and matching distances was a bad idea (when using them in our models). In this section, we provided a justification for this claim. We will here consider the case of the SIS model and the edit distance, noting a similar argument can be used regarding the matching distance and its use within the SIM model.

Suppose we have assumed the edit distance $d_{\mathrm{E},\delta(\cdot)}$ as defined in Section 4.2, with a penalty function given by $\delta(\mathcal{I}) = \rho$, where $\rho > 0$ is a fixed constant. Suppose also the mode $\mathcal{S}^m = (\mathcal{I}_1^m, \ldots, \mathcal{I}_{\tilde{N}}^m)$ and dispersion γ have been fixed. Now, consider an observation $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ drawn from the SIS model with these parameters, that is $\mathcal{S} \sim \mathrm{SIS}(\mathcal{S}^m, \gamma)$. Moreover, assume $\mathcal{S}$ is such that $N > N_m$, that is, $\mathcal{S}$ has more paths than the mode. Since $\mathcal{S}$ has more paths, at least one of these must be unmatched when evaluating $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}^m)$. Now, we assume that the ith path in $\mathcal{S}$ is such an unmatched path, and consider the conditional distribution of this path given the others. In particular, letting

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

denote $\mathcal{S}$ with the path $\mathcal{I}$ in the ith entry we would like the conditional distribution of $\mathcal{I}$ given the other paths, that is

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

where here we use the notation $\mathcal{S}_{-i}$ to denote all the paths of $\mathcal{S}$ excluding the ith. This is a distribution over the space of paths and can be obtained directly from the probability of $\mathcal{S}_{\mathcal{I}}$ implied by the model, namely

$$\begin{aligned} p(\mathcal{I} \,|\, \mathcal{S}_{-i}, \mathcal{S}^m, \gamma) &\propto \exp\{-\gamma d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{\mathcal{I}}, \mathcal{S}^m)\} \\ &\propto \exp\{-\gamma \cdot \delta(\mathcal{I})\} \\ &= \exp\{-\gamma\rho\} \\ &\propto 1 \end{aligned} \tag{15}$$

where here we use the fact that since $\mathcal{I}$ is not included in the matching it features in $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}^m)$ only via its penalty.

Now, assume we are considering the bounded case as defined in Appendix A.2, restricting the sample space to include observations with paths in $\boldsymbol{\mathcal{I}}_K^*$, that is, paths with length at most K. With this, (15) implies $\mathcal{I}$ is equally likely to be any path in this space. Though this may seem innocuous, observe that a uniform distribution over $\boldsymbol{\mathcal{I}}_K^*$ places a higher probability to paths of longer length, by virtue of their prevalence. In particular, if $\mathcal{I}$ is equally likely to be any path in $\boldsymbol{\mathcal{I}}_K^*$ then we will have

$$\begin{aligned} \mathbb{P}(\mathcal{I} \text{ has length } n) &= \frac{\#\text{paths of length } n}{|\boldsymbol{\mathcal{I}}_K^*|} \\ &\propto V^n \end{aligned}$$

recalling that $V = |\mathcal{V}|$ is the size of the vertex set, that is, the probability that $\mathcal{I}$ is of length n grows exponentially.

This is a very odd assumption to make and unlikely to hold in practice. Moreover, if we consider the infinite case $K \to \infty$ this will result in degenerative behaviour similar to that observed when sampling from the infinite versions of our model with very low values of the dispersion, as illustrated in Supplement S3. In particular, if one tried sample from such models via our MCMC algorithms then one is likely to see a divergence in path lengths.

For these reasons we recommend to not use a fixed penalty within the edit or matching distances, opting instead for one which somehow takes the size of the path being penalised into account, such as the choice $\delta(\mathcal{I}) = d_I(\mathcal{I}, \Lambda)$ mentioned in Section 4.4. This way, the distribution of probability in the underlying space can be better controlled by the induced distance, avoiding the undesirable properties illustrated above.

S2 Montonicity of the Entropy

As alluded to in Section 3.1, for both the SIS and SIS models, the dispersion γ admits an interpretation analogous with the precision in a Gaussian distribution. The impact of this model parameter can be illustrated more formally through its control of entropy. Considering the SIS model, the entropy is defined as follows

$$H(\mathcal{S}^m, \gamma) := -\mathbb{E}\left[\log p(\mathcal{S} \,|\, \mathcal{S}^m, \gamma)\right], \tag{16}$$

quantifying how evenly the distribution $p(\mathcal{S}|\mathcal{S}^m, \gamma)$ allocates probability over the sample space, whereby larger values of $H(\mathcal{S}^m, \gamma)$ imply this distribution is 'more uniform' over $\boldsymbol{\mathcal{S}}^*$, with a minimum value of $H(\mathcal{S}^m, \gamma) = 0$ attained by a pointmass.

The entropy also has an interpretation with regards to randomness or variance, whereby distributions with a higher entropy are more random, that is more variable. As will be shown, with any $d_S(\cdot, \cdot)$ and $\mathcal{S}^m$, the entropy $H(\mathcal{S}^m, \gamma)$ is a monotonic function of γ, agreeing with the intuition that γ controls the variability of the distribution. This holds similarly for the SIM model. We note a similar result was shown by Lunagómez et al. (2021) (Proposition 3.3, proved pages 41-43 of Supplementary Material).

For the SIS model, recall the entropy is given by

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

Unfortunately, as was the case for the normalising constant $Z(\mathcal{S}^m, \gamma)$ (Section 4.5), since $\boldsymbol{\mathcal{S}}^*$ is infinite we have no guarantee that $H(\mathcal{S}^m, \gamma)$ will exist. However, what we can say is that, when $H(\mathcal{S}^m, \gamma)$ exists, it is monotonic in γ. To show this, we first differentiate $H(\mathcal{S}^m, \gamma)$ with respect to γ

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

where one has

$$\begin{aligned}
\frac{\partial}{\partial\gamma}\log Z(\mathcal{S}^m,\gamma) &= \frac{\frac{\partial}{\partial\gamma}Z(\mathcal{S}^m,\gamma)}{Z(\mathcal{S}^m,\gamma)} \\
&= \frac{1}{Z(\mathcal{S}^m,\gamma)}\frac{\partial}{\partial\gamma}\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}\exp\left\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\right\}\right) \\
&= \frac{1}{Z(\mathcal{S}^m,\gamma)}\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}\frac{\partial}{\partial\gamma}\exp\left\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\right\} \\
&= \frac{1}{Z(\mathcal{S}^m,\gamma)}\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}(-d_S(\mathcal{S},\mathcal{S}^m))\exp\left\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\right\} \\
&= -\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}d_S(\mathcal{S},\mathcal{S}^m)\frac{1}{Z(\mathcal{S}^m,\gamma)}\exp\left\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\right\} \\
&= -\mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)],
\end{aligned} \tag{17}$$

thus implying

$$\frac{\partial}{\partial\gamma}H(\mathcal{S}^m,\gamma) = \frac{\partial}{\partial\gamma}\mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)].$$

Now, we have

$$\begin{aligned}
\frac{\partial}{\partial\gamma}\mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)] &= \frac{\partial}{\partial\gamma}\left(\frac{1}{Z(\mathcal{S}^m,\gamma)}\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}d_S(\mathcal{S},\mathcal{S}^m)\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}\right) \\
&= -\frac{\frac{\partial}{\partial\gamma}Z(\mathcal{S}^m,\gamma)}{Z(\mathcal{S}^m,\gamma)^2}\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}d_S(\mathcal{S},\mathcal{S}^m)\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}\right) \\
&\qquad -\frac{1}{Z(\mathcal{S}^m,\gamma)}\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}d_S(\mathcal{S},\mathcal{S}^m)^2\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}\right) \\
&= -\frac{\frac{\partial}{\partial\gamma}Z(\mathcal{S}^m,\gamma)}{Z(\mathcal{S}^m,\gamma)^2}\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}d_S(\mathcal{S},\mathcal{S}^m)\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}\right) \\
&\qquad -\frac{1}{Z(\mathcal{S}^m,\gamma)}\left(\sum_{\mathcal{S}\in\boldsymbol{\mathcal{S}}^*}d_S(\mathcal{S},\mathcal{S}^m)^2\exp\{-\gamma d_S(\mathcal{S},\mathcal{S}^m)\}\right) \\
&= -\frac{\frac{\partial}{\partial\gamma}Z(\mathcal{S}^m,\gamma)}{Z(\mathcal{S}^m,\gamma)}\mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)] - \mathbb{E}[d_S(\mathcal{S},\mathcal{S}^m)^2]
\end{aligned} \tag{18}$$

$$= \mathbb{E}[d_S(\mathcal{S}, \mathcal{S}^m)]^2 - \mathbb{E}[d_S(\mathcal{S}, \mathcal{S}^m)^2] \tag{19}$$
$$= \mathbb{V}\text{ar}[d_S(\mathcal{S}, \mathcal{S}^m)],$$

where (19) follows from (18) by applying (17). 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. This is equivalent to saying we have montonicity provided $\mathbb{V}\text{ar}[d_S(\mathcal{S}, \mathcal{S}^m)] > 0$. This result, much like that of Lunagómez 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 SIM model (Definition 2) by a simple change of notation. We do not repeat this for brevity.

S3 Bounding Dimensions

As mentioned in Section 3.1, in practice we recommend constraining the sample space to be finite, as defined in Appendix A.2. In this section, we will illustrate why this recommendation was made. We will also elaborate on how one might go about choosing the necessary bounds and discuss how our MCMC algorithms can be slightly altered to respect the imposed dimension constraints.

To illustrate the need for constraining the sample space we will show via simulation what can go wrong. In particular, we will show that one can, in certain scenarios, observe a divergence in dimension when sampling from models over an infinite space. We note the following will regard the SIS model, but analogous behaviour will be observable for the infinite version of the SIM model. Suppose we would like to sample from the SIS model of Definition 1 over the infinite space $\boldsymbol{\mathcal{S}}^*$ of all interaction sequences. As we have mentioned in Section 5.6, we cannot do this exactly, but we can obtain approximate samples via the iMCMC algorithm proposed therein. With this, for a given mode $\mathcal{S}^m$ and dispersion γ, we can obtain a chain $(\mathcal{S}_i)_{i=1}^M$ approximating a sample from the SIS model with these parameters, that is, a chain targeting the following distribution

$$p(\mathcal{S}|\mathcal{S}^m, \gamma) \propto \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\}, \tag{20}$$

over the infinite space $\boldsymbol{\mathcal{S}}^*$ of all interaction sequences over a fixed set of vertices. Figure 17 summarises three such chains drawn with different values for the dispersion, where for each sample $\mathcal{S}_i = (\mathcal{I}_1^{(i)}, \ldots, \mathcal{I}_{N_i}^{(i)})$ we plot the number of paths N_i, or what we call the outer dimension. In each case, we initialised the chains at the mode $\mathcal{S}^m$, with a lag of 1 between samples and no burn-in. Here one can clearly see the dimensions of samples tends to be larger as γ decreases. Moreover, when $\gamma = 3.5$ the dimension appears to diverge, showing a clear upward trend.

Why does this happen? Observe that as γ goes to zero $p(\mathcal{S}|\mathcal{S}^m, \gamma)$ of (20) will converge to the uniform distribution over the space $\boldsymbol{\mathcal{S}}^*$. Though this might seem innocuous, one must remember that there are far more interaction sequences with large dimensions. For example, if $\mathcal{S}$ has n entries in total across all its paths, then there are V^n possible choices thereof. As such, with a uniform distribution over $\boldsymbol{\mathcal{S}}^*$, the probability of sampling observations with large dimensions will be higher relative to those with smaller dimensions, leading to the observed divergence.

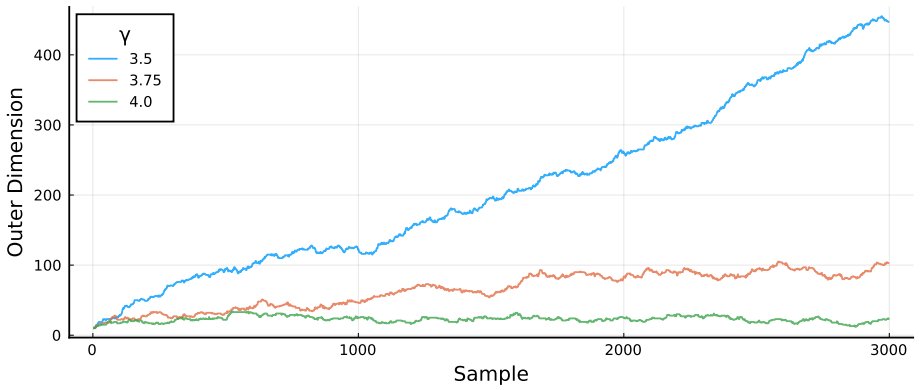


Figure 17: Illustrating divergence in dimension for the SIS model over an infinite space. Each trace summarises an MCMC chain sampling from an SIS($\mathcal{S}^m, \gamma$) model over the space $\boldsymbol{\mathcal{S}}^*$ with the dispersion γ set at different values. Here we observe, for γ low enough, the dimensions of samples diverges.

This implies there is always a chance, if γ is low enough, that the dimensions will diverge. This will inevitably cause computational issues when sampling from these models via our iMCMC algorithm. Even if one does not first run out of memory, the cost of evaluating the distance $d_S(\mathcal{S}, \mathcal{S}^m)$ is very likely to grow with the dimension of $\mathcal{S}$, significantly slowing down the sampling time. This becomes ever more significant in the context of the algorithms we proposed to sample from the posterior in Section 5. Recall that in updating the dispersion (Section 5.3) we must sample auxiliary data from the model at γ and γ', a current and proposed value for the dispersion, which we do via our iMCMC algorithm as above. Consequently, there is a chance, either for γ or γ', the dimension will blow-up when sampling auxiliary data. Moreover, obtaining such samples will generally be more computationally cumbersome, increasing the time taken to obtain the auxiliary data, in turn slowing down the time taken to obtain the posterior samples. Ultimately, the result will be a posterior sampling scheme which is unstable and unpredictable in terms of run time.

This motivates our recommendation to constrain the sample space. In particular, one can instead assume the sample space is given by $\boldsymbol{\mathcal{S}}^*_{K,L} \subseteq \boldsymbol{\mathcal{S}}^*$, as defined in Supplement S3, where K and L represent the maximum path length (inner dimension) and number of paths (outer dimension) respectively. This effectively places a lid on the possible dimension of observations, removing the possibility of divergence in dimensions. Note to sample from models over such constrained spaces we can use the exact same MCMC algorithms used in the infinite case. All one must do is set the probability of values outside of $\boldsymbol{\mathcal{S}}^*_{K,L}$ to zero, that is for each $\mathcal{S} \in \boldsymbol{\mathcal{S}}^*$ let

$$p(\mathcal{S}|\mathcal{S}^m, \gamma) \propto \begin{cases} \exp\{-\gamma d_S(\mathcal{S}, \mathcal{S}^m)\} & \text{if } \mathcal{S} \in \boldsymbol{\mathcal{S}}^*_{K,L} \\ 0 & \text{if } \mathcal{S} \notin \boldsymbol{\mathcal{S}}^*_{K,L} \end{cases}$$

defining a distribution over the infinite space $\boldsymbol{\mathcal{S}}^*$ which we can target with our MCMC algorithm. Observe that, within the MCMC algorithm, if we are currently at state $\mathcal{S} \in \boldsymbol{\mathcal{S}}^*_{K,L}$

any proposal $\mathcal{S}' \notin \boldsymbol{\mathcal{S}}^*_{K,L}$ will always be rejected, since its acceptance probability will evaluate to zero. Hence we will obtain only samples from the constrained space, as desired.

With the recommendation of bounding the sample space in this manner comes the question of how to choose bounds K and L. Observe this is only a question of interest when one is considering inference.[5] In this case, we will have observed a sample

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

which we assume was drawn i.i.d. via

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

where $\mathcal{S}^m$ and γ are some unknown model parameters. Notice assuming bounds K and L implies we must have $\mathcal{S}^{(i)} \in \boldsymbol{\mathcal{S}}^*_{K,L}$ for each of the observed samples. In this way, this informs the following thresholds for possible choices of K and L

$$K \geq \max_{i=1,\ldots,n} \left\{ \max_{j=1,\ldots,N^{(i)}} n_j^{(i)} \right\} \qquad L \geq \max_{i=1,\ldots,n} N^{(i)}$$

where $N^{(i)}$ is the number of paths in the ith observation and $n_j^{(i)}$ is the length of the jth path in the ith observation. As such, we recommend choosing bounds either at or close these thresholds, and indeed this is what we did for the data analysis of Section 7.

We finalise these discussions by noting that in constraining the sample space one can actually alter the interpretation of γ in the resulting model, in the sense that draws from the model with the same value of γ but different choices for K and L can look quite different in terms of the samples they generate. Though this might seem problematic, we note that the same applies to different choices of distance $d_S(\cdot, \cdot)$, the flexibility of which is a key feature of our proposed methodology. In this way, one must accept that the interpretation of γ is context-dependent. However, as was shown in Section 7.2, it is possible to simulate from the model as a means to interpret such inferred values.

S4 Distance Computation

This section contains details regarding the computation of the distances defined in Section 4. All involve setting-up and solving some form of optimisation problem. Guidance on both steps will here be provided.

S4.1 Matching Distance

As mentioned in Section 4.1, evaluating $d_{\text{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}')$ (Definition 3) requires finding an optimal matching. As outlined therein, we consider casting this as an *assignment problem*. Consequently, one can appeal to known solvers, such as the Hungarian algorithm (Kuhn, 1955), to find an optimal solution.

5. One may instead be simply sampling from the model, for example, to examine the behaviour of the model with a particular distance $d_S(\cdot, \cdot)$. In such cases, the bounds can be set to personal preference, or the infinite space assumed with the awareness that dimensions could diverge.

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

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

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

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

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

Towards evaluating the matching distance, we set-up a cost matrix C such that the optimal solution found via the Hungarian algorithm coincides with an optimal matching in accordance with Definition 3. Here we consider two scenarios. In the first, more general case, we will optimise over all matchings (including those which match nothing). In the second scenario, we will optimise over only complete matchings. The second case is a smaller optimisation problem, making it easier to solve and thus preferable. However, it is not guaranteed that an optimal matching will be complete. Thus the former will work in all cases, but the latter may result in a sub-optimal solution in some scenarios. To guide this, we provide a result which says when it is okay to use the latter approach.

S4.1.1 Optimising Over All Matchings

Suppose we have two interaction multisets

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

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

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

where Λ represents a dummy element, so that, if say $\mathcal{I}_i$ is paired with a Λ this will be interpreted as $\mathcal{I}_i$ being unmatched, and the same for elements of $\mathcal{E}'$. Notice also with A

there are M dummy elements added to $\mathcal{E}$, so that all M elements of $\mathcal{E}'$ could in theory be matched with a dummy element, that is, all elements of $\mathcal{E}'$ could be left unmatched. Similarly, in B there are N dummy elements added to $\mathcal{E}'$, so that all elements of $\mathcal{E}$ could be unmatched. Moreover, with this both A and B are now of the same size $n = N + M$, as required for the assignment problem.

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

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

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

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

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

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

where $d_I(\cdot,\cdot)$ is the chosen ground distance and $\delta(\cdot)$ the chosen penalty term for unmatched elements. Notice the cost of pairing two dummy elements is zero. To see why C takes this form, consider the cost it implies for a given matching $\mathcal{M}_\sigma$, that is

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

where we have simply applied the definition of C as above. Comparing this with Definition 3, one can see that C encodes the required matching cost to be minimised when evaluating $d_{\text{M},\delta(\cdot)}$. Thus, if every matching is represented by every pairing of A and B, and the costs are equivalent, then the optimal solutions will coincide. This means any optimal σ^* found for the given C will define an optimal matching $\mathcal{M}_{\sigma^*}$ which can be used to evaluate $d_{\text{M},\delta(\cdot)}$. With this, the steps to evaluate $d_{\text{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}')$ are: (i) construct C as above, (ii) pass C to a solver, such as the Hungarian algorithm, returning an optimal permutation σ^* and then finally (iii) translate σ^* to an optimal matching $\mathcal{M}_{\sigma^*}$ to evaluate the distance.

S4.1.2 Optimising Over Complete Matchings

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

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

Proposition 7. *Given two interaction multisets $\mathcal{E}$ and $\mathcal{E}'$, if the following holds*

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

for all $\mathcal{I} \in \mathcal{E}$ and $\mathcal{I}' \in \mathcal{E}'$, then there exists a complete optimal matching achieving the optimum defining the matching distance $d_{\mathrm{M},\delta(\cdot)}$ (Definition 3).

A proof of Proposition 7 can be found in Supplement S5.3. This result implies, if the conditions therein are satisfied, to compute the matching distance it suffices to find an optimal complete matching. As such, in what follows we show how an assignment problem can again be set up to enact this optimisation.

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

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

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

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

where Λ represents a dummy element such that $\mathcal{I}'_j \in \mathcal{E}'$ being paired with Λ is interpreted as this interaction being left unmatched. Observe that in comparison with the set-up of Supplement S4.1.1, we need only augment the smaller of the two multisets with dummy variables. Notice again we have A and B being of the same size, in particular $n = M$, that is, the size of the larger multiset. In this case, the interpretation of a permutation σ is as follows

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

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

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

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

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

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

where $d_I(\cdot,\cdot)$ is the chosen ground distance and $\delta(\cdot)$ the penalty for unmatched interactions. Notice, as in Supplement S4.1.1, this cost matrix C is such that

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

which, since $\mathcal{M}_\sigma$ is complete, is in accordance with the cost function being minimised in evaluating $d_{\text{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}')$. Thus any optimal σ^* found for cost matrix C of this form will map to an optimal complete matching $\mathcal{M}_{\sigma^*}$ which can be used to evaluate the matching distance, provided the conditions of Proposition 7 hold. With this, the steps to evaluate $d_{\text{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}')$ (when the necessary conditions hold) are: (i) construct C as above (ii), pass C to a solver, returning an optimal permutation σ^*, then (iii) map σ^* to an optimal complete matching $\mathcal{M}_{\sigma^*}$ to evaluate the distance.

We finalise these details by noting when the conditions of Proposition 7 will hold for the example penalty functions provided in Section 4.1. In particular, we have

- **Fixed penalty:** if $\delta(\mathcal{I}) = \rho$ for some constant $\rho > 0$, then when comparing two multisets $\mathcal{E}$ and $\mathcal{E}'$ the conditions will hold provided

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

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

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

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

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

This implies, when looking to evaluate the matching distance, if using a distance-based penalty, one only needs to optimise over complete matchings, whilst for the fixed-penalty, optimisation over complete matchings is only valid if the above bound is satisfied.

S4.2 Edit Distance

The edit distance (Definition 4) can be seen as a special case of the string edit distance introduced by Wagner and Fischer (1974). Consequently, it can be evaluated using the same dynamic programming algorithm proposed therein.

Suppose we have two interaction sequences

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

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

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:i}, \mathcal{S}'_{1:j}) = \min \begin{cases} d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:(i-1)}, \mathcal{S}'_{1:j}) + \delta(\mathcal{I}_i) \\ d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:i}, \mathcal{S}'_{1:(j-1)}) + \delta(\mathcal{I}'_j) \\ d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:(i-1)}, \mathcal{S}'_{1:(j-1)}) + d_I(\mathcal{I}_i, \mathcal{I}'_j), \end{cases} \tag{21}$$

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

Letting $C_{ij} = d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:(i-1)}, \mathcal{S}'_{1:(j-1)})$, incremental evaluation of (21) can be seen as filling up an $(N+1) \times (M+1)$ matrix C either row-by-row or column-by-column according to the following formula

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

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

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

for $i = 2, \ldots, N$ and $j = 2, \ldots, M$, which follow by seeing $\mathcal{S}_{1:0}$ and $\mathcal{S}'_{1:0}$ as empty sequences, so that when measuring the distance of these to $\mathcal{S}'_{1:j}$ and $\mathcal{S}_{1:i}$ (respectively) all entries thereof will be left unmatched, since there are no entries to be matched to. Finally, when both $i = 1$ and $j = 1$ we will have $C_{11} = d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_{1:0}, \mathcal{S}'_{1:0}) = 0$, since we can see this as the distance of the empty sequence to itself. All together, these represent initial conditions from which repeated application of (21) will take us to the desired result.

Algorithm 4 outlines pseudocode to evaluate $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}')$ by filling the matrix C in this manner. However, observe that when updating a row (or column) of C one only needs to know the previous row (or column). As such, we only need to store the current and previous row (or column), leading to an algorithm which uses less memory and is typically faster. Pseudocode of this light-memory alternative is given in Algorithm 5.

S4.3 Path distances

The LCS distance d_{LCS}, like the edit distance (Supplement S4.2), is a special case of the string edit distance proposed by Wagner and Fischer (1974). Thus, the dynamic programming algorithm proposed therein can be applied, in this case at a complexity $\mathcal{O}(n \cdot m)$ where n and m are the lengths of the paths being compared.

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

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

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

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

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

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

where the distance is then given by $d_{\text{LCS}}(\mathcal{I}, \mathcal{I}') = C_{(n+1)(m+1)}$, that is, the final entry of the constructed matrix. For pseudocode of the resulting algorithm to compute d_{LCS} see Algorithm 6, with a lighter-memory version outlined in Algorithm 7, which essentially stores only the current and previous rows of C.

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

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

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

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

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

where when common subpaths appear between $\mathcal{I}$ and $\mathcal{I}'$ one will see increments in Q diagonally. The maximum length of a subpath can thus be obtained by taking the element-wise maximum of Q, that is $\delta_{\text{LSP}} = \max_{ij} Q_{ij}$, which can then be used to evaluate d_{LSP} (see definition in Section 4.3). We summarise this in Algorithm 8, where we keep track of the maximum in Q as it is filled. A lighter-memory algorithm is also outlined in Algorithm 9, making use of the fact we only need to know the current and previous rows of Q.

S5 Distance Proofs

In this section, theoretical properties of distances introduced in Section 4 will be proved. These mostly regard showing they are *distance metrics* (Definition 5). The proofs will follow a similar structure, proving each metric condition in turn.

S5.1 Matching Distance is a Metric

This section contains a proof that the matching distance (Definition 4) is a metric, provided that certain conditions on the penalty function and the distance between interactions are satisfied. This represents one half of Proposition 6.

Proof To aid this exposition, write $d_{\text{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}')$ in terms of its cost function as follows

$$d_{\text{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}') = \min_{\mathcal{M} \in \boldsymbol{\mathcal{M}}(\mathcal{E}, \mathcal{E}')} \text{Cost}(\mathcal{M})$$

where

$$\text{Cost}(\mathcal{M}) = \sum_{(\mathcal{I}, \mathcal{I}') \in \mathcal{M}} d_I(\mathcal{I}, \mathcal{I}') + \sum_{\mathcal{I} \in \mathcal{M}^c_{\mathcal{E}}} \delta(\mathcal{I}) + \sum_{\mathcal{I}' \in \mathcal{M}^c_{\mathcal{E}'}} \delta(\mathcal{I}'),$$

denotes the cost of the matching $\mathcal{M}$. We first show metric condition (i) (identity of indiscernibles) holds. If we assume $\mathcal{E} = \mathcal{E}'$ then one can construct a matching $\mathcal{M}^*$ by pairing equivalent elements of $\mathcal{E}$ and $\mathcal{E}'$, leading to the following upper bound

$$\begin{aligned} d_{\text{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}') &\leq \text{Cost}(\mathcal{M}^*) \\ &= \sum_{(\mathcal{I}, \mathcal{I}') \in \mathcal{M}^*} d_I(\mathcal{I}, \mathcal{I}') \\ &= 0 \end{aligned}$$

where the second line follows since $\mathcal{M}^*$ includes all elements of $\mathcal{E}$ and $\mathcal{E}'$ and thus no penalty terms will appear, whilst the third line follows since $\mathcal{M}^*$ matches equivalent elements and hence, using the fact $d_I(\cdot, \cdot)$ satisfies the identity of indiscernibles, all pairwise distances will be zero. Now, since $d_I(\cdot, \cdot) \geq 0$ and $\delta(\cdot) > 0$ by assumption, $d_{\text{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}')$ is a sum of positive values, implying also that $d_{\text{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}') \geq 0$. Together these imply $d_{\text{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}') = 0$.

Conversely, assume that $d_{\text{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}') = 0$. This implies both the sum of pairwise distances and penalisation terms must be zero. Since by assumption $\delta(\mathcal{I}) > 0$ this implies there must be no penalty terms, that is, all elements of $\mathcal{E}$ and $\mathcal{E}'$ must be included in the

matching. Thus, with $\mathcal{M}^*$ the optimal matching, we have

$$\begin{aligned} d_{\mathrm{M},\delta(\cdot)}(\mathcal{E},\mathcal{E}') &= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^*} d_I(\mathcal{I},\mathcal{I}') \\ &= 0, \end{aligned}$$

which, since $d_I(\cdot,\cdot)$ is non-negative, implies

$$d_I(\mathcal{I},\mathcal{I}') = 0 \quad \forall\, (\mathcal{I},\mathcal{I}') \in \mathcal{M}^*,$$

which in turn implies

$$\mathcal{I} = \mathcal{I}' \quad \forall\, (\mathcal{I},\mathcal{I}') \in \mathcal{M}^*,$$

since $d_I(\cdot,\cdot)$ satisfies the identity of indiscernibles. Hence, we have $\mathcal{E} = \mathcal{E}'$, thus confirming that $d_{\mathrm{M},\delta(\cdot)}$ satisfies metric condition (i).

The symmetry condition (ii) follows trivially from the symmetry of $d_I(\cdot,\cdot)$ (since it is a metric) and the penalisation terms.

Finally, we prove that $d_{\mathrm{M},\delta(\cdot)}$ satisfies metric condition (iii) (triangle inequality). Assuming we have three multisets

$$\mathcal{E}_X = \{\mathcal{I}_1^X,\ldots,\mathcal{I}_{n_X}^X\} \quad \mathcal{E}_Y = \{\mathcal{I}_1^Y,\ldots,\mathcal{I}_{n_Y}^Y\} \quad \mathcal{E}_Z = \{\mathcal{I}_1^Z,\ldots,\mathcal{I}_{n_Z}^Z\}$$

we seek to show that

$$d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X,\mathcal{E}_Y) \le d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X,\mathcal{E}_Z) + d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_Z,\mathcal{E}_Y).$$

Let $\mathcal{M}_{XZ}^*$ and $\mathcal{M}_{ZY}^*$ denote optimal matchings for $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X,\mathcal{E}_Z)$ and $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_Z,\mathcal{E}_Y)$ respectively, so that

$$d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X,\mathcal{E}_Z) = \mathrm{Cost}(\mathcal{M}_{XZ}^*) \quad d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_Z,\mathcal{E}_Y) = \mathrm{Cost}(\mathcal{M}_{ZY}^*)$$

and observe these induce a matching $\mathcal{M}_{XY}$ of $\mathcal{E}_X$ and $\mathcal{E}_Y$ as follows

$$\mathcal{M}_{XY} := \{(\mathcal{I}^X,\mathcal{I}^Y) \,:\, (\mathcal{I}^X,\mathcal{I}^Z) \in \mathcal{M}_{XZ}^* \text{ and } (\mathcal{I}^Z,\mathcal{I}^Y) \in \mathcal{M}_{ZY}^* \text{ for some } \mathcal{I}^Z \in \mathcal{E}_Z\}$$

that is, we pair elements of $\mathcal{E}_X$ and $\mathcal{E}_Y$ if they were paired to the same elements of $\mathcal{E}_Z$. For example, Figure 18 shows two cases of optimal matchings $\mathcal{M}_{XZ}^*$ and $\mathcal{M}_{ZY}^*$ along with the matching $\mathcal{M}_{XY}$ they induce (which turns out to be the same in both cases). Notice by definition of $d_{\mathrm{M},\delta(\cdot)}$ we have

$$d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X,\mathcal{E}_Y) \le \mathrm{Cost}(\mathcal{M}_{XY}),$$

and so the triangle inequality will follow if we can show the following holds

$$\mathrm{Cost}(\mathcal{M}_{XY}) \le d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_X,\mathcal{E}_Z) + d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}_Z,\mathcal{E}_Y). \tag{22}$$

To prove (22) we show every possible term on the LHS is less than or equal to some unique terms appearing on the RHS. The key terms appearing on the LHS are (i) pairwise distances for matched elements (ii) penalisation of unmatched elements.

Considering first (i), by definition of $\mathcal{M}_{XY}$ each pair $(\mathcal{I}^X, \mathcal{I}^Y) \in \mathcal{M}_{XY}$ is associated with some *unique* $(\mathcal{I}^X, \mathcal{I}^Z) \in \mathcal{M}^*_{XZ}$ and $(\mathcal{I}^Z, \mathcal{I}^Y) \in \mathcal{M}^*_{ZY}$, that is, there is some element $\mathcal{I}^Z \in \mathcal{E}_Z$ which both $\mathcal{I}^X$ and $\mathcal{I}^Y$ are matched to. Furthermore, since $d_I(\cdot,\cdot)$ is a distance metric it satisfies the triangle inequality, and so

$$d_I(\mathcal{I}^X, \mathcal{I}^Y) \leq d_I(\mathcal{I}^X, \mathcal{I}^Z) + d_I(\mathcal{I}^Z, \mathcal{I}^Y),$$

and thus each pairwise distance of matched elements on the LHS of (22) is less than or equal to some unique terms on the RHS.

For (ii) consider first the penalisation terms for elements of $\mathcal{E}_X$ not included in the matching $\mathcal{M}_{XY}$, that is $\delta(\mathcal{I}^X)$ for $\mathcal{I}^X \in (\mathcal{M}_{XY})^c_X$. We now seek to show that each $\delta(\mathcal{I}^X)$ is less than or equal to some unique terms appearing on the RHS of (22). For $\mathcal{I}^X$ to not be in $\mathcal{M}_{XY}$ one of two things must have happened

Case 1: As illustrated in Figure 18a, one may have $(\mathcal{I}^X, \mathcal{I}^Z) \in \mathcal{M}^*_{XZ}$ for some $\mathcal{I}^Z \in \mathcal{E}_Z$ with $(\mathcal{I}^Z, \mathcal{I}^Y) \notin \mathcal{M}^*_{ZY}$ for any $\mathcal{I}^Y \in \mathcal{E}_Y$

$$\implies \text{ a term on the RHS of } d_I(\mathcal{I}^X, \mathcal{I}^Z) + \delta(\mathcal{I}^Z)$$

which will also be unique to the pair $(\mathcal{I}^X, \mathcal{I}^Z)$. Now, since by the assumption $|\delta(\mathcal{I}^X) - \delta(\mathcal{I}^Y)| \leq d_I(\mathcal{I}^X, \mathcal{I}^Y)$ for all $\mathcal{I}^X, \mathcal{I}^Y \in \boldsymbol{\mathcal{I}}$, we have that

$$\delta(\mathcal{I}^X) \leq d_I(\mathcal{I}^X, \mathcal{I}^Z) + \delta(\mathcal{I}^Z)$$

as desired;

Case 2: Alternatively, as shown in Figure 18b, we might have $(\mathcal{I}^X, \mathcal{I}^Z) \notin \mathcal{M}^*_{XZ}$ for any $\mathcal{I}^Z \in \mathcal{E}_Z$

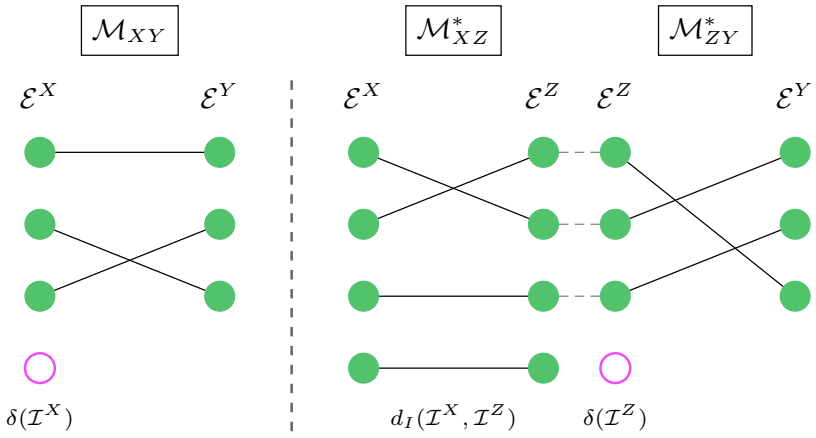
$$\implies \text{ a term on the RHS of } \delta(\mathcal{I}^X),$$

and thus in this case we trivially have

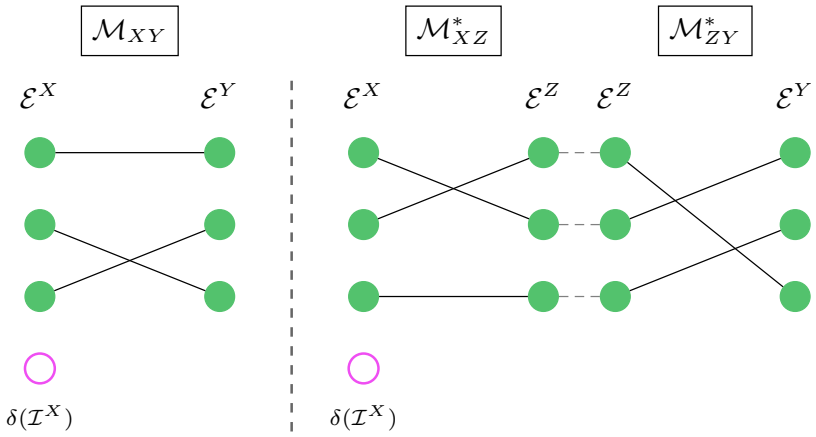
$$\delta(\mathcal{I}^X) \leq \delta(\mathcal{I}^X).$$

In both cases, we have a term on the LHS of (22) which is less than or equal to some unique terms on the RHS. Notice this argument can be applied similarly to the penalisation terms for elements of $\mathcal{E}_Y$ not in the matching $\mathcal{M}_{XY}$. For brevity we will not repeat this here, and henceforth assume all penalisation terms for $\mathcal{E}_Y$ are less than or equal to some unique terms on the RHS of (22).

All together, we have every term on the LHS of (22), both pairwise distances and penalties, being less than or equal to some unique terms on the RHS, proving the inequality holds. As a consequence, $d_{\mathrm{M},\delta(\cdot)}$ satisfies the triangle inequality. This completes the proof that if $d_I(\cdot,\cdot)$ is a distance metric and the penalty $\delta(\cdot)$ satisfies the conditions of Proposition 6 then the matching distance will be a metric. ■



(a) An element of $\mathcal{E}_X$ is left unmatched in $\mathcal{M}_{XY}$ (induced matching) because the element it was matched with in $\mathcal{E}_Z$ was left unmatched in $\mathcal{M}_{ZY}$.



(b) An element of $\mathcal{E}_X$ is left unmatched in $\mathcal{M}_{XY}$ (induced matching) because it was also unmatched in $\mathcal{M}_{XZ}$.

Figure 18: Examples of (a) **Case 1** and (b) **Case 2** appearing when proving that $d_{\mathrm{M},\delta(\cdot)}$ satisfies the triangle inequality. In each subfigure, we have three matchings of the multisets $\mathcal{E}_X$, $\mathcal{E}_Y$ and $\mathcal{E}_Z$, where the two right-most matchings are example optimal matchings which induce the left-most matching of $\mathcal{E}_X$ and $\mathcal{E}_Y$. In both cases, an element of $\mathcal{E}_X$ is left unmatched in the induced matching.

S5.2 Edit Distance is a Metric

This section contains a proof that the edit distance (Definition 4) is a metric, given certain conditions on the penalty function are satisfied. This represents one half of Proposition 6.

Proof To aid this exposition, write $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S},\mathcal{S}')$ in terms of its cost function as follows

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S},\mathcal{S}') = \min_{\mathcal{M}\in\boldsymbol{\mathcal{M}}_m(\mathcal{S},\mathcal{S}')}\{\mathrm{Cost}(\mathcal{M})\}$$

where

$$\mathrm{Cost}(\mathcal{M}) = \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}} d_I(\mathcal{I},\mathcal{I}') + \sum_{\mathcal{I}\in\mathcal{M}_{\mathcal{S}}^{c}} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in\mathcal{M}_{\mathcal{S}'}^{c}} \delta(\mathcal{I}'),$$

denotes the cost of the matching $\mathcal{M}$. First we consider metric condition (i) (identity of indiscernibles). Supposing we have $\mathcal{S} = (\mathcal{I}_1,\ldots,\mathcal{I}_n)$ and $\mathcal{S}' = (\mathcal{I}'_1,\ldots,\mathcal{I}'_m)$ with $\mathcal{S} = \mathcal{S}'$, this implies that $n = m$ and

$$\mathcal{I}_i = \mathcal{I}'_i \quad \text{for } i = 1,\ldots,n$$

that is, all interactions are equal. As such, we can trivially construct a monotone matching $\mathcal{M}^*$ by pairing equivalent interactions, that is

$$\mathcal{M}^* = \{(\mathcal{I}_1,\mathcal{I}'_1),\ldots,(\mathcal{I}_n,\mathcal{I}'_n)\}, \tag{23}$$

which leads to the following upper bound

$$\begin{aligned} d_{\mathrm{E},\delta(\cdot)}(\mathcal{S},\mathcal{S}') &\leq \mathrm{Cost}(\mathcal{M}^*) \\ &= \sum_{i=1}^{n} d_I(\mathcal{I}_i,\mathcal{I}'_i) = 0. \end{aligned}$$

Now, since $d_I(\cdot,\cdot)$ is a metric we have $d_I(\mathcal{I},\mathcal{I}') \geq 0$, whilst $\delta(\mathcal{I}) > 0$ also by assumption, which together imply $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S},\mathcal{S}') \geq 0$ for any sequences $\mathcal{S}$ and $\mathcal{S}'$. These two bounds combine to imply that when $\mathcal{S} = \mathcal{S}'$ we have $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S},\mathcal{S}') = 0$, proving one direction of metric condition (i).

For the converse case, we first assume that $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S},\mathcal{S}') = 0$, which implies both the sum of pairwise distances and penalisation terms must be zero (since all are sums of non-negative values). Moreover, since $\delta(\mathcal{I}) > 0$ this implies there must be no penalty terms. Thus if $\mathcal{M}^*$ is an optimal monotone matching then it must contain all entries of $\mathcal{S}$ and $\mathcal{S}'$. Observe this also implies $\mathcal{S}$ and $\mathcal{S}'$ must be of the same length. Furthermore, the only possible *monotone* matching which includes all entries of both sequences is that defined in (23), which implies

$$\begin{aligned} d_{\mathrm{E},\delta(\cdot)}(\mathcal{S},\mathcal{S}') &= \mathrm{Cost}(\mathcal{M}^*) \\ &= \sum_{i=1}^{n} d_I(\mathcal{I}_i,\mathcal{I}'_i) = 0, \end{aligned}$$

where we have applied the definition of $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S},\mathcal{S}')$ directly, using the fact that since $\mathcal{M}^*$ is the only possibly monotone matching of $\mathcal{S}$ and $\mathcal{S}'$ it must be optimal. Now, since $d_I(\mathcal{I},\mathcal{I}') \geq 0$ (since $d_I(\cdot,\cdot)$ is a metric), this implies

$$d_I(\mathcal{I}_i,\mathcal{I}'_i) = 0 \quad \text{for } i = 1,\ldots n,$$

and since $d_I(\cdot,\cdot)$ itself satisfies the identity of indiscernibles, this implies

$$\mathcal{I}_i = \mathcal{I}'_i \quad \text{for } i = 1, \ldots, n$$

from which we can conclude $\mathcal{S} = \mathcal{S}'$. This proves the converse case, confirming that $d_{\mathrm{E},\delta(\cdot)}$ satisfies metric condition (i).

The symmetry condition (ii) follows trivially from the symmetry of $d_I(\cdot,\cdot)$ and the penalisation terms.

Finally, we confirm metric condition (iii) (triangle inequality) is satisfied. The approach is almost identical to that applied in the proof of Supplement S5.1 (matching distance is a metric; the other half of Proposition 6) with one key difference: we must ensure all matchings are monotone. Given three interaction sequences

$$\mathcal{S}_X = (\mathcal{I}_1^X, \ldots, \mathcal{I}_{n_X}^X) \quad \mathcal{S}_Y = (\mathcal{I}_1^Y, \ldots, \mathcal{I}_{n_Y}^Y) \quad \mathcal{S}_Z = (\mathcal{I}_1^Z, \ldots, \mathcal{I}_{n_Z}^Z)$$

we seek to show that

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X, \mathcal{S}_Y) \leq d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X, \mathcal{S}_Z) + d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_Z, \mathcal{S}_Y).$$

With $\mathcal{M}^*_{XZ}$ and $\mathcal{M}^*_{ZY}$ denoting optimal monotone matchings for $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X, \mathcal{S}_Z)$ and $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_Z, \mathcal{S}_Y)$ respectively, that is

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X, \mathcal{S}_Z) = \mathrm{Cost}(\mathcal{M}^*_{XZ}) \quad d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_Z, \mathcal{S}_Y) = \mathrm{Cost}(\mathcal{M}^*_{ZY})$$

observe these induce a matching $\mathcal{M}_{XY}$ of $\mathcal{S}_X$ and $\mathcal{S}_Y$ as follows

$$\mathcal{M}_{XY} = \{(\mathcal{I}_i^X, \mathcal{I}_j^Y) \,:\, (\mathcal{I}_i^X, \mathcal{I}_k^Z) \in \mathcal{M}^*_{XZ} \text{ and } (\mathcal{I}_k^Z, \mathcal{I}_j^Y) \in \mathcal{M}^*_{ZY} \text{ for some } \mathcal{I}_k^Z \in \mathcal{S}_Z\}$$

that is, we match entries of $\mathcal{S}_X$ and $\mathcal{S}_Y$ if they were matched to the same entry of $\mathcal{S}_Z$.

We now confirm $\mathcal{M}_{XY}$ is a monotone matching. Recall that $\mathcal{M}_{XY}$ is monotone if for any pairs $(\mathcal{I}_{i_1}^X, \mathcal{I}_{j_1}^Y)$ and $(\mathcal{I}_{i_2}^X, \mathcal{I}_{j_2}^Y)$ in $\mathcal{M}_{XY}$ we have

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

To show this holds, observe by definition of $\mathcal{M}_{XY}$ there exists $\mathcal{I}_{k_1}^Z$ and $\mathcal{I}_{k_2}^Z$ in $\mathcal{S}_Z$ such that

$$\begin{aligned} (\mathcal{I}_{i_1}^X, \mathcal{I}_{k_1}^Z) \in \mathcal{M}^*_{XZ} \quad (\mathcal{I}_{k_1}^Z, \mathcal{I}_{j_1}^Y) \in \mathcal{M}^*_{ZY} \\ (\mathcal{I}_{i_2}^X, \mathcal{I}_{k_2}^Z) \in \mathcal{M}^*_{XZ} \quad (\mathcal{I}_{k_2}^Z, \mathcal{I}_{j_2}^Y) \in \mathcal{M}^*_{ZY} \end{aligned}$$

Furthermore, since $\mathcal{M}^*_{XZ}$ and $\mathcal{M}^*_{ZY}$ are monotone we have

$$i_1 < i_2 \iff k_1 < k_2 \qquad \text{and} \qquad k_1 < k_2 \iff j_1 < j_2$$

which therefore implies

$$i_1 < i_2 \iff k_1 < k_2 \iff j_1 < j_2,$$

as required. Hence $\mathcal{M}_{XY}$ is also monotone. With the induced matching being monotone, observe that by definition of $d_{\mathrm{E},\delta(\cdot)}$ we have the following

$$d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X, \mathcal{S}_Y) \leq \mathrm{Cost}(\mathcal{M}_{XY})$$

which implies the triangle inequality will hold if we can show the following inequality is satisfied

$$\mathrm{Cost}(\mathcal{M}_{XY}) \leq d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_X, \mathcal{S}_Z) + d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}_Z, \mathcal{S}_Y). \tag{24}$$

Observe this is almost identical to the scenario appearing in Supplement S5.1, where the inequality of (22) was shown to hold to prove that $d_{\mathrm{M},\delta(\cdot)}$ satisfied the triangle inequality. Since the induced matching here is the same used therein, albeit applied to sequences rather than multisets, an identical argument can be used show that (24) holds. For brevity, we will not repeat these steps here, assuming henceforth that (24) holds, implying $d_{\mathrm{E},\delta(\cdot)}$ satisfies the triangle inequality and completing the proof. ■

S5.3 Completeness of Matchings

This section contains a proof of Proposition 7, which gives conditions under which there exists a complete optimal matching between two interaction multisets (as obtained to evaluate the matching distance between them).

Proof Given two interaction multisets $\mathcal{E}$ and $\mathcal{E}'$, in accordance with Proposition 7, it will be assumed that

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

for all $\mathcal{I} \in \mathcal{E}$ and $\mathcal{I}' \in \mathcal{E}'$. To aid this exposition, write $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}')$ in terms of its cost function as follows

$$d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}') = \min_{\mathcal{M} \in \boldsymbol{\mathcal{M}}(\mathcal{E}, \mathcal{E}')} \{\mathrm{Cost}(\mathcal{M})\}$$

where

$$\mathrm{Cost}(\mathcal{M}) = \sum_{(\mathcal{I}, \mathcal{I}') \in \mathcal{M}} d_I(\mathcal{I}, \mathcal{I}') + \sum_{\mathcal{I} \in \mathcal{M}_{\mathcal{E}}^c} \delta(\mathcal{I}) + \sum_{\mathcal{I}' \in \mathcal{M}_{\mathcal{E}'}^c} \delta(\mathcal{I}'),$$

denotes the cost of the matching $\mathcal{M}$. Towards proving this result, assume that any matching $\mathcal{M}^*$ for which

$$\mathrm{Cost}(\mathcal{M}^*) = \min_{\mathcal{M} \in \boldsymbol{\mathcal{M}}(\mathcal{E}, \mathcal{E}')} \{\mathrm{Cost}(\mathcal{M})\} = d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}'),$$

is *not* complete, seeking a contradiction. There may be more than one such matching, so without loss of generality, let $\mathcal{M}^*$ denote any one of these optimal matchings. Since $\mathcal{M}^*$ is not complete, there must be a currently unmatched pair, that is, $(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}')$ such that $\tilde{\mathcal{I}} \in \mathcal{E}$ and $\tilde{\mathcal{I}}' \in \mathcal{E}'$ but $\tilde{\mathcal{I}} \notin \mathcal{M}_{\mathcal{E}}^*$ and $\tilde{\mathcal{I}}' \notin \mathcal{M}_{\mathcal{E}'}^*$. One can now define a new matching $\mathcal{M}^{**}$ by augmenting $\mathcal{M}^*$ as follows

$$\mathcal{M}^{**} = \mathcal{M}^* \cup \{(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}')\}$$

for which

$$\begin{aligned}
\text{Cost}(\mathcal{M}^{**}) &= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^{**}} d_I(\mathcal{I},\mathcal{I}') + \sum_{\mathcal{I}\in(\mathcal{M}^{**})^c_{\mathcal{E}}} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in(\mathcal{M}^{**})^c_{\mathcal{E}'}} \delta(\mathcal{I}'), \\
&= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^{*}} d_I(\mathcal{I},\mathcal{I}') + d_I(\tilde{\mathcal{I}},\tilde{\mathcal{I}}') + \sum_{\mathcal{I}\in(\mathcal{M}^{**})^c_{\mathcal{E}}} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in(\mathcal{M}^{**})^c_{\mathcal{E}'}} \delta(\mathcal{I}') \\
&\leq \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^{*}} d_I(\mathcal{I},\mathcal{I}') + \delta(\tilde{\mathcal{I}}) + \delta(\tilde{\mathcal{I}}') + \sum_{\mathcal{I}\in(\mathcal{M}^{**})^c_{\mathcal{E}}} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in(\mathcal{M}^{**})^c_{\mathcal{E}'}} \delta(\mathcal{I}') \qquad (25) \\
&= \sum_{(\mathcal{I},\mathcal{I}')\in\mathcal{M}^{*}} d_I(\mathcal{I},\mathcal{I}') + \sum_{\mathcal{I}\in(\mathcal{M}^{*})^c_{\mathcal{E}}} \delta(\mathcal{I}) + \sum_{\mathcal{I}'\in(\mathcal{M}^{*})^c_{\mathcal{E}'}} \delta(\mathcal{I}') \\
&= \text{Cost}(\mathcal{M}^{*})
\end{aligned}$$

where in the third line we invoke the assumption that

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

for all $\mathcal{I} \in \mathcal{E}$ and $\mathcal{I}' \in \mathcal{E}'$. Since $\mathcal{M}^*$ was optimal, we must also have $\text{Cost}(\mathcal{M}^*) \leq \text{Cost}(\mathcal{M})$ for all matchings $\mathcal{M}$, which combined with (25) implies $\text{Cost}(\mathcal{M}^{**}) = \text{Cost}(\mathcal{M}^*)$, that is, $\mathcal{M}^{**}$ is also an optimal matching. Moreover, we have $|\mathcal{M}^{**}| = |\mathcal{M}^*|+1$. Now, either (i) $\mathcal{M}^{**}$ is complete, or (ii) we can repeat this augmentation, increasing the matching cardinality until it is complete. Either way, we arrive at a matching which is both optimal and complete, contradicting our assumption that all optimal matchings were not complete. Consequently, by contradiction, there must exist at least one optimal matching that is complete. ■

S5.4 Path Distances are Metrics

This section contains a proof that both the interaction distances introduced in Section 4.3, namely, the longest common subpath distance, denoted d_{LSP}, and the longest common subsequence distance, denoted d_{LCS}, are distance metrics.

Proof This a proof that d_{LSP} and d_{LSP} are both distance metrics. Let us first prove that d_{LCS} is a metric. Recall the LCS distance (defined in Section 4.3) between paths $\mathcal{I}$ and $\mathcal{I}'$ is given by

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

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

$$d_{\text{LCS}}(\mathcal{I},\mathcal{I}') = n + m - 2\delta_{\text{LCS}} = 0 \qquad (26)$$

we claim this implies $n = m$. To see this, notice if we assume $n < m$ this implies $n + m > 2n \geq 2\delta_{\text{LCS}}$ where we have used the fact $\delta_{\text{LCS}} \leq n$. Notice this contradicts (26). A similar contradiction will be found if we assume $n > m$, and consequently we must have $n = m$.

Substituting this into (26) leads to $\delta_{\rm LCS} = n = m$ which implies that $\mathcal{I}$ and $\mathcal{I}'$ share a common subsequence of the same length as themselves, that is $\mathcal{I} = \mathcal{I}'$. This proves one direction. For the converse case, if $\mathcal{I} = \mathcal{I}'$ then it should be clear that the maximum common subsequence will be that including all their entries, that is $\delta_{\rm LCS} = n = m$ and hence

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

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

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

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

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

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

$$\begin{aligned} d_{\rm LCS}(\mathcal{I}^X, \mathcal{I}^Y) &= n + m - 2\delta_{XY} \quad d_{\rm LCS}(\mathcal{I}^X, \mathcal{I}^Z) = n + k - 2\delta_{XZ} \\ d_{\rm LCS}(\mathcal{I}^Z, \mathcal{I}^Y) &= m + k - 2\delta_{ZY} \end{aligned}$$

then, if the triangle inequality holds, we have

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

which is equivalent to the following

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

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

$$\delta_{XZ} + \delta_{ZY} - k \leq \delta_{XY}, \tag{27}$$

thus, if we show (27) holds the implications will trace back to show the the triangle inequality also holds. Towards doing so, we consider a finding the common subsequence between $\mathcal{I}^X$ and $\mathcal{I}^Y$ induced by that between $\mathcal{I}^X$ and $\mathcal{I}^Z$ and between $\mathcal{I}^Y$ and $\mathcal{I}^Z$, which will allow us to obtain the desired lower bound.

To aide this exposition we introduce some notation. In particular, for two subsequences $\boldsymbol{v}$ and $\boldsymbol{u}$ of $[n] = (1, \ldots, n)$ we can extend the notion of unions and intersections used for sets, that is $\boldsymbol{v} \cup \boldsymbol{u}$ and $\boldsymbol{v} \cap \boldsymbol{u}$ respectively, where if $\boldsymbol{w} = \boldsymbol{v} \cap \boldsymbol{u}$ then each entry w_i appears in both $\boldsymbol{v}$ and $\boldsymbol{u}$, whilst if $\boldsymbol{w} = \boldsymbol{v} \cup \boldsymbol{u}$ then each w_i appears in at least one of $\boldsymbol{u}$ and $\boldsymbol{v}$. For example, if we have $n = 5$ and $\boldsymbol{u} = (1, 3, 5)$ and $\boldsymbol{v} = (1, 2, 5)$ then $\boldsymbol{u} \cap \boldsymbol{v} = (1, 5)$ whilst $\boldsymbol{u} \cup \boldsymbol{v} = (1, 2, 3, 5)$. Moreover, with $|\boldsymbol{v}|$ denoting the length of subsequence $\boldsymbol{v}$, the following will hold

$$|\boldsymbol{v}| + |\boldsymbol{u}| - |\boldsymbol{v} \cap \boldsymbol{u}| = |\boldsymbol{v} \cup \boldsymbol{u}|,$$

which can be seen as analogous to the inclusion-exclusion identity for sets.

Now suppose that we have indexing subsequences $\boldsymbol{u}_{XZ}$, $\boldsymbol{v}_{XZ}$, $\boldsymbol{u}_{ZY}$ and $\boldsymbol{v}_{ZY}$ such that

$$\mathcal{I}^X_{\boldsymbol{v}_{XZ}} = \mathcal{I}^Z_{\boldsymbol{u}_{XZ}} \quad \mathcal{I}^Z_{\boldsymbol{v}_{ZY}} = \mathcal{I}^Y_{\boldsymbol{u}_{ZY}}$$

with $|\boldsymbol{u}_{XZ}|=|\boldsymbol{v}_{XZ}|=\delta_{XZ}$ and $|\boldsymbol{u}_{ZY}|=|\boldsymbol{v}_{ZY}|=\delta_{ZY}$, that is, these index maximal common subsequences. Observe the intersection $\boldsymbol{u}_{XZ}\cap\boldsymbol{v}_{ZY}$ defines a subsequence of $\mathcal{I}^Z$ which is shared with both $\mathcal{I}^X$ and $\mathcal{I}^Y$, and consequently, if we let $\boldsymbol{v}_{XY}$ and $\boldsymbol{u}_{XY}$ denote indices of the associated subsequences of $\mathcal{I}^X$ and $\mathcal{I}^Y$, respectively, we have

$$\mathcal{I}^X_{\boldsymbol{v}_{XY}}=\mathcal{I}^Y_{\boldsymbol{u}_{XY}}$$

that is, these index a common subsequence of $\mathcal{I}^X$ and $\mathcal{I}^Y$. Moreover, if we let

$$\delta^*:=|\boldsymbol{v}_{XY}|=|\boldsymbol{u}_{XY}|=|\boldsymbol{u}_{XZ}\cap\boldsymbol{v}_{ZY}|,$$

denoting the size of this induced common subsequence, then by the inclusion-exclusion identity above we have

$$\delta_{XZ}+\delta_{ZY}-\delta^*=|\boldsymbol{u}_{XZ}|+|\boldsymbol{v}_{ZY}|-|\boldsymbol{u}_{XZ}\cap\boldsymbol{v}_{ZY}|=|\boldsymbol{u}_{XZ}\cup\boldsymbol{v}_{ZY}|\leq k$$

where the inequality here follows since $\boldsymbol{u}_{ZX}\cup\boldsymbol{v}_{ZY}$ is an indexing subsequences of $\mathcal{I}^Z$, which is of length k. This rearranges to the following

$$\delta_{XZ}+\delta_{ZY}-k\leq\delta^*,$$

and finally, using the fact that $\delta^*\leq\delta_{XY}$ by definition of δ_{XY} as the *maximal* length of a common subsequence between $\mathcal{I}^X$ and $\mathcal{I}^Y$, we thus have

$$\delta_{XZ}+\delta_{ZY}-k\leq\delta_{XY},$$

confirming (27) holds, as desired. Consequently, the LCS distance satisfies metric condition (iii). This completes the proof that d_{LCS} is a distance metric.

We now consider proving d_{LSP} is also a distance metric. Firstly, regarding the identity of indiscernibles (i), one can use exactly the same argument as for the LCS distance above. For brevity, we will avoid repeating this and henceforth assume this condition holds. Similarly, the symmetry condition (ii) again follows trivially from the symmetry of common subpaths.

To show d_{LSP} satisfies the triangle inequality (iii) we can use almost the same argument outlined above for the LCS distance. In particular, one can show that (27) holds, where in this case δ_{XZ}, δ_{ZY} and δ_{XY} denote maximal *subpath* sizes. A key difference here is that we must obtain an induced subpath rather than subsequence. If we introduce the shorthand notation $(i:j)=(i,\dots,j)$ where $1\leq i\leq j\leq n$, denoting the subpath of $[n]$ from i to j (notice this is consistent with notation used in Section 4.3), then as with subsequences we can define natural generalisations of the interaction and union of two subpaths, in particular

$$(i:j)\cap(l:k)=(\max(i,l):\min(j,k))\quad(i:j)\cup(l:k)=(\min(i,l):\max(j,k)),$$

and moreover if $|(i:j)|=j-i+1$ denotes subpath length we will again have the following inclusion-exclusion identity

$$|(i:j)|+|(l:k)|-|(i:j)\cap(l:k)|=|(i:j)\cup(l:k)|.$$

With these, one can directly adapt the argument used to show d_{LCS} satisfied the triangle inequality. In particular, any two optimal common subpaths between $\mathcal{I}^X$ and $\mathcal{I}^Z$ and

between $\mathcal{I}^Z$ and $\mathcal{I}^Y$ will induce a common subpath between $\mathcal{I}^X$ and $\mathcal{I}^Y$, in turn providing the required bound. For brevity, we do not repeat this here, assuming henceforth that metric condition (iii) holds.

In summary, above it has been shown that metric conditions (i) to (iii) hold for both the LCS and LSP distances, completing the proof that both are indeed distance metrics. ■

S6 Guidance on MCMC Scaleability and Mixing

In this section, we provide some further details and guidance regarding our MCMC scheme outlined in Section 5. In particular, we elaborate on the scaleability of our algorithms, highlighting the key features that will impact their computational cost, and discuss their mixing, illustrating how certain tuning parameters can be used to maximise efficiency.

S6.1 Posterior Sampling Cost

The details here will concern inference for the SIS model, as outlined in Section 5. However, all the points highlighted will apply equally in the context of the SIM model and its inference scheme outlined in Supplement S9. Suppose we have observed a sample of n interaction sequences

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

which we assume were drawn via the hierarchical model (7). This implies a posterior distribution $p(\mathcal{S}^m, \gamma | \{\mathcal{S}^{(i)}\}_{i=1}^n)$ as stated in (8) which we propose to sample from via our component-wise MCMC algorithm (Section 5.2), returning a sample $\{(\mathcal{S}_i^m, \gamma_i)\}_{i=1}^M$ where M is a specified desired number of posterior samples. Considerations of scaleability thus becomes a question of how long it will take to obtain these samples, and what can affect this.

For us, of most interest are cases where we evaluate $d_S(\cdot, \cdot)$, the chosen distance metric.[6] Naturally, it makes sense to think of the what happens in a single iteration. For our MCMC algorithm, there are two key elements where distance evaluations appear, featuring both in updates for the dispersion (Section 5.3) and the mode (Section 5.4), namely

1. **Likelihood evaluations** - these arise when evaluating acceptance probabilities, as seen in the closed forms for the dispersion update (37) and mode update (40). In particular, we have terms of the following form

$$\sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) \quad \text{and} \quad \sum_{i=1}^n d_S(\mathcal{S}_i^*, \mathcal{S}^m)$$

   representing evaluations on observed data $\{\mathcal{S}^{(i)}\}_{i=1}^n$ and auxiliary data $\{\mathcal{S}_i^*\}_{i=1}^n$, respectively;

2. **Auxiliary data sampling** - both updates for the dispersion and mode require sampling a single auxiliary dataset $\{\mathcal{S}_i^*\}_{i=1}^n$ (of the same size as the observed data) at

6. Of course, there are other computations and operations involved, but in general those involving d_S will contribute most to the overall computational cost.

the proposed parameters. Recall these we sample via MCMC (Section 5.6), wherein distance evaluations will occur during computation of the associated acceptance probabilities.

Given these two elements, the question is what can affect their cost? In this regard, the following three aspects will come into play:

- The computational cost of the distance metric $d_S(\cdot, \cdot)$ - a more costly distance will necessarily lead to more expensive likelihood evaluations and auxiliary sampling, in turn leading to longer posterior sampling times;
- The number of observed data points n - this will increase the number of terms in the likelihood and require sampling of more auxiliary data, both pushing up the computational cost;
- The dimensions of observed data $\{\mathcal{S}^{(i)}\}_{i=1}^n$ - recall that generally one expects the cost of evaluating d_S to grow with the size of interaction sequences being compared, for example, the edit distance $d_{\mathrm{E},\delta(\cdot)}$ has a cost $\mathcal{O}(N \cdot M)$ where N and M are the number of paths in $\mathcal{S}$ and $\mathcal{S}'$, respectively (see Section 4.2). This implies both the cost of likelihood evaluations and auxiliary sampling will grow with the dimension of observed data. The former should be evident, the latter, however, is somewhat subtle. This follows since we expect the sampled auxiliary data to resemble the observed data.[7] In particular, we expect $\{\mathcal{S}_i^*\}_{i=1}^n$ and $\{\mathcal{S}^{(i)}\}_{i=1}^n$ to be of similar dimension. Moreover, samples of larger dimension will in general take longer to obtain, since they will involve distance evaluations between larger interaction sequences, thus driving up the cost of auxiliary sampling.

S6.2 Auxiliary Sampling Cost

As noted in the previous section, sampling of auxiliary data is a key computational elements of our proposed posterior sampling algorithm. As such, we here elaborate further on its cost. Firstly, we show it is highly likely to make-up the majority of the computational cost of the overall posterior sampling algorithm, often requiring far more distance evaluations than the likelihood terms. We then go on to discuss how the nature of the distribution being targeted will alter the cost required to obtain the desired samples.

As we mention at the end of Section 5.6, one will typically want to introduce some burn-in period b and lag l, that is, dropping the first b samples and taking every lth sample thereafter. The hope is this will reduce the correlation in the chain, leading to samples which look more like an exact draw from the model, as required for the exchange and iExchange algorithms. However, doing so will clearly increase the cost of obtaining the auxiliary samples. To be more precise, since each accept-reject step involves two distance evaluations (for the current state and the proposal), the total number of evaluations involved in sampling the auxiliary data will be given by

$$2(b + l(n-1) + 1), \tag{28}$$

7. To see this, observe one expects the posterior samples to concentrate around parameter values which would generate data resembling that which was observed. Since auxiliary data is sampled at such parameter values, this implies we would expect the auxiliary and observed data to have some resemblance.

where n is the number of observed data points. Thus, as we increase b and l the cost of obtaining these samples will grow. Moreover, this will typically be much larger than the number of distance evaluations arising during evaluation of the likelihood terms, as outlined in the previous section. In particular, notice the likelihood terms appearing in the dispersion conditional (37) require $2n$ distance evaluations, whilst those in the mode conditional (40) require $2(n+1)$ distance evaluations, where the latter includes both likelihood evaluations and terms from the prior. Both will be dwarfed by (28) when b and l are of reasonable size. This leads to a key point worth emphasis: the overall cost of posterior sampling is likely to be driven predominantly by the cost of auxiliary sampling.

One might now ask what can impact the cost of auxiliary sampling? This reduces to the question of how long it takes to sample from our SIS and SIM models via the proposed MCMC algorithms. The answer: it depends on the model parameters (and the underlying distance being used). To see this, suppose we have sampled a chain $(\mathcal{S}_i)_{i=1}^M$ targeting an $\text{SIS}(\mathcal{S}^m, \gamma)$ distribution via our MCMC algorithm (Section 5.6). Observe this implies we have evaluated

$$d_S(\mathcal{S}_i, \mathcal{S}^m)$$

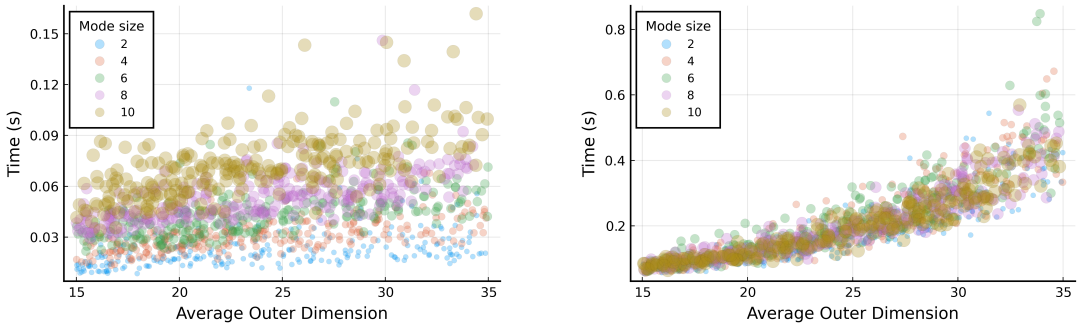
for each sample in the chain (when evaluating acceptance probabilities). Recall that in general we expect d_S to be more costly to evaluate on larger interaction sequences. As such, these evaluations will be slowed down if (i) $\mathcal{S}^m$ is large, or (ii) the samples $\mathcal{S}_i$ are large. Observe both will depend on the model parameters: clearly (i) depends on the mode $\mathcal{S}^m$, whilst (ii) will depend on dispersion γ since, as seen in Supplement S3, as γ decreases the resulting distributions will tend to focus higher probability on interaction sequences of larger size. This argument applies equally to the SIM model.

As empirical evidence, in Figure 19 we summarise the timings of samples drawn from the SIS and SIM models at different parameterisations. These simulations were run on a Dell Latitude 5440 laptop, with a 13th Gen Intel Core i7-1370P processor and 64 GB of RAM; the same machine used for the model fit of Section 7.2. With $M = 1000$ we timed how long it took to sample a chain.

- $(\mathcal{S}_i)_{i=1}^M$ targeting an $\text{SIS}(\mathcal{S}^m, \gamma)$ distribution, with distance $d_{\text{E},\delta(\cdot)}$,
- $(\mathcal{E}_i)_{i=1}^M$ targeting an $\text{SIM}(\mathcal{E}^m, \gamma)$ distribution, with distance $d_{\text{M},\delta(\cdot)}$,

where we considered different combinations of $\mathcal{S}^m$, $\mathcal{E}^m$ and γ. In each case, for a single sampled chain, we plot $(\bar{N}, t)$ where t is the time (in seconds) taken and $\bar{N} = \frac{1}{M}\sum_{i=1}^M N_i$ is the average outer dimension of the samples, where here N_i is the number of paths in the ith sample, that is $\mathcal{S}_i$ for the SIS model and $\mathcal{E}_i$ for the SIM model. Model parameters were here chosen as follows. We let $\mathcal{S}^m = (\mathcal{I}_1^m, \ldots, \mathcal{I}_{\tilde{N}}^m)$ and $\mathcal{E}^m = \{\mathcal{I}_1^m, \ldots, \mathcal{I}_{\tilde{N}}^m\}$ for $\tilde{N} = 2, 4, 6, 8, 10$, leading to 5 different modes, where the paths $\mathcal{I}_i^m$ were fixed and of equal length. For each mode, we considered a range of γ values which led to samples with $a \leq \bar{N} \leq b$, taking $a = 15$ and $b = 35$ in this case.

Consulting Figure 19, for both models one can clearly see a positive correlation between the dimension of samples and the time taken. However, there is a difference in the nature of this relationship between the two models. This can be explained via the complexity of the underlying distances as follows



(a) SIS model with $d_{\mathrm{E},\delta(\cdot)}$

(b) SIM model with $d_{\mathrm{M},\delta(\cdot)}$

Figure 19: Examining the cost of sampling from the SIS and SIM models via our MCMC algorithms. In (a) and (b) a single data point $(\bar{N}, t)$ summarises a single MCMC chain drawn from the respective model, where t is the time taken to sample this chain and $\bar{N}$ is the average number of paths within the samples, or what we call the outer dimension. To highlight choices for $\tilde{N}$, the number of paths in the mode, markers have been colored and sized proportionally.

- For the SIS model, $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}^m)$ has a complexity $\mathcal{O}(\tilde{N} \cdot N)$, where N and $\tilde{N}$ are the number of paths in $\mathcal{S}$ and $\mathcal{S}^m$ respectively. This explains the linear relationship with respect to $\bar{N}$ and changes in slope with $\tilde{N}$;
- For the SIM model, $d_{\mathrm{M},\delta(\cdot)}(\mathcal{E}, \mathcal{E}^m)$ has a complexity $\mathcal{O}(\max(\tilde{N}, N)^3 + \tilde{N} \cdot N)$, where N and $\tilde{N}$ are the number of paths in $\mathcal{E}$ and $\mathcal{E}^m$ respectively. Notice for the parameterisations above we expect $N > \tilde{N}$, so the complexity becomes $\mathcal{O}(N^3 + \tilde{N} \cdot N)$. With the N^3 term likely to dominate, this explains the non-linear relationship with respect to $\bar{N}$ and the absence of any relationship with respect to $\tilde{N}$.

S6.3 Mixing

In this section, we discuss the mixing of our proposed MCMC algorithms, highlighting how certain tuning parameters come into play. We will first discuss the mixing of our model sampling algorithms, as used to sample auxiliary data, before going on to discuss the mixing of our posterior sampling algorithms.

S6.3.1 Model Sampling

We will here discuss sampling from the SIS model, but note what follows will apply readily in the context of the SIM model and the MCMC algorithm proposed to sample from it. As outlined in Section 5.6, we propose an iMCMC algorithm which mixes together two moves

(a) **Edit allocation move** - keeps the number of paths fixed, editing to those currently present. This has the tuning parameter $\nu_{\mathrm{ed}} \in \mathbb{Z}_{\geq 1}$ representing the maximum number of edits in total;

(b) **Path insertion and deletion move** - varies the number of paths by simultaneously deleting and inserting paths. Here we have the tuning parameter $\nu_{\text{td}} \in \mathbb{Z}_{\geq 1}$ representing the maximum number of paths to be inserted and deleted;

where we do (a) with probability β and (b) with probability $(1-\beta)$, where $\beta \in (0,1)$ is further tuning parameter.

Both ν_{ed} and ν_{td} serve to control the aggressiveness of proposals, with larger values resulting in proposal which are 'more different' than the current state, on average. The suitability of choices thereof will depend on the nature of the target distribution

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

where if γ is large, so that $p(\mathcal{S}|\mathcal{S}^m, \gamma)$ will be highly concentrated about $\mathcal{S}^m$, smaller values of ν_{ed} and ν_{td} will be appropriate, whilst as γ gets smaller, and $p(\mathcal{S}|\mathcal{S}^m, \gamma)$ becomes less concentrated about $\mathcal{S}^m$, it is likely that larger values for ν_{ed} and ν_{td} would be required to ensure sufficient exploration of the space.

We illustrate this empirically with a small simulation study. Here we consider two scenarios, one where γ is large and one where it is small. In each scenario, for a grid of ν_{ed} and ν_{td} values we sample an MCMC chain $(\mathcal{S}_i)_{i=1}^M$ via out iMCMC algorithm targeting the respective distribution, before evaluating their mixing. In particular, we consider evaluating the integrated autocorrelation time (IACT) of the real-valued series $(x_i)_{i=1}^M$ where $x_i := d_S(\mathcal{S}_i, \mathcal{S}^m)$, where $\mathcal{S}^m$ is the mode of the distribution being sampled from. For a series of real-valued random variables $X_1, X_2, \ldots$ the IACT is defined to be

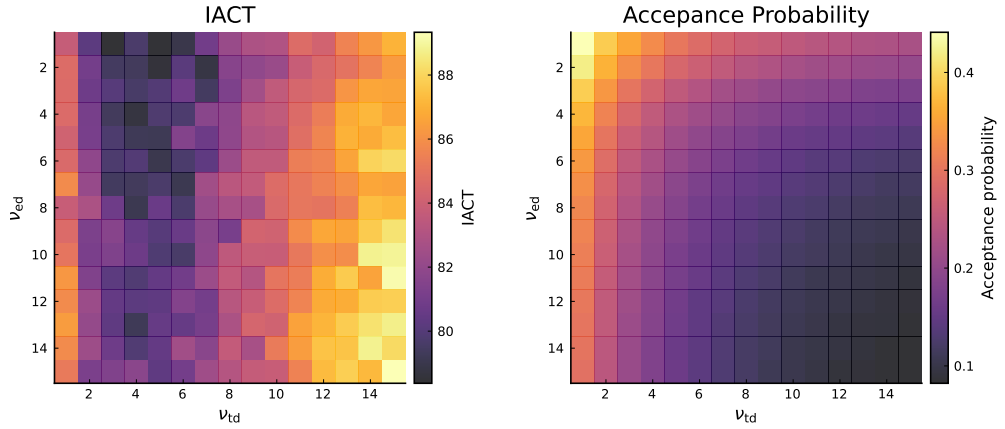
$$\text{IACT} := 1 + 2\sum_{i=1}^{\infty} \gamma_k$$

where γ_k is the lag k correlation, that is, the correlation between X_i and X_{i+k}. Typically, we cannot evaluate the IACT exactly, but given the finite realisation $(x_i)_{i=1}^M$ it can be estimated via

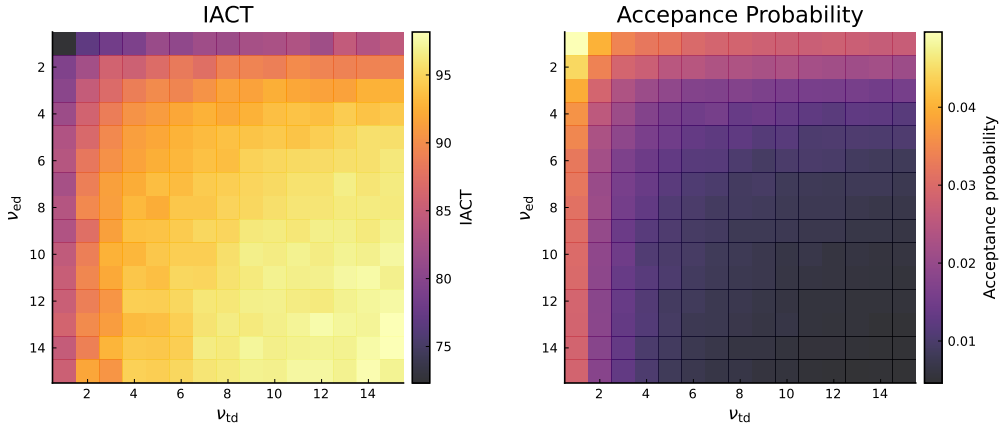
$$\text{IACT} \approx 1 + 2\sum_{i=1}^{K} \hat{\gamma}_k \tag{29}$$

where $\hat{\gamma}_k$ are estimates of the lagged correlations obtained from the sample $(x_i)_{i=1}^M$ and $K < \infty$ is some chosen truncation. The IACT is an often-used measure of efficiency for MCMC chains, since the presence of correlation within the chain is known to increase the variance of any estimates computed with its samples. In this way, a lower IACT is better, since it implies a more efficient use of the MCMC samples.

Figure 20 summarises the results. In each subfigure, for each pair $(\nu_{\text{ed}}, \nu_{\text{td}})$ we plot estimates of the IACT, obtained via (29) with $K = 50$, for a single chain sampled with these hyperparameters. Alongside, we also plot the observed acceptance probability, that is, the proportion of proposals that were accepted. Note there is some noise in the IACT values, evident particularly in Figure 20a, which is to be expected since these are estimates. Here we also let $\beta = 0.5$, attempting either move with 50-50 probability in each iteration. As expected, the concentration of the distribution influences which hyperparameters are most suitable. In particular, we can see from Figure 20a that when γ is high the lower values for ν_{ed} and ν_{td} lead to the lowest IACT values, whilst from Figure 20a we can see that when γ is low it is larger values thereof that result the best values of the IACT.



(a) Low γ



(b) High γ

Figure 20: Comparing the IACT for MCMC chains sampled with different choices of hyper-parameters ν_{ed} and ν_{td} when targeting an SIS($\mathcal{S}^m, \gamma$) model. In the right-hand subplots we show also the observed acceptance probability of each MCMC chain.

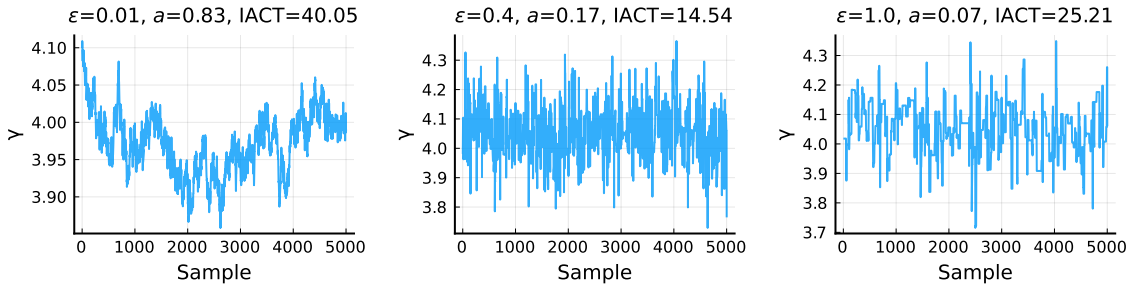


Figure 21: Illustrating how ε impacts mixing of γ when sampling from the posterior $p(\mathcal{S}^m, \gamma | \{\mathcal{S}^{(i)}\}_{i=1}^n)$. Here one can see when ε is too small or too large the correlation in the resultant chain increases.

S6.3.2 Posterior Sampling

In this section, we discuss the mixing of our MCMC algorithm proposed to sample from posterior for the SIS model, as outlined in Section 5, again noting all points made will apply equally to our scheme for the SIM model. Recall we propose to sample from the posterior $p(\mathcal{S}^m, \gamma | \{\mathcal{S}^{(i)}\}_{i=1}^n)$ via a component-wise MCMC algorithm (Section 5.2 and Figure 5), alternating between sampling from (i) $p(\mathcal{S}^m | \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n)$, a distribution over $\boldsymbol{\mathcal{S}}^*$ (or $\boldsymbol{\mathcal{S}}^*_{K,L}$ if constraining the sample space), as detailed in Section 5.4, and (ii) $p(\gamma | \mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n)$, a distribution over $\mathbb{R}_+$, outlined in Section 5.3. As such, when it comes to assessing the mixing in this context, one can consider two elements

(i) Mixing of $\mathcal{S}^m$ in $\boldsymbol{\mathcal{S}}^*$ given ν_{ed} and ν_{td};

(ii) Mixing of γ in $\mathbb{R}_+$ given ε.

Considering (i), observe we will generally expect the posterior to concentrate as the number of observations n increases, as confirmed in our simulations of Section 6. This implies in practice we are likely to be in a scenario reminiscent of sampling from the model with a high value of γ, that is, the second simulation scenario considered in Supplement S6.3.1, summarised in Figure 20b. As such, one will typically want to choose ν_{ed} and ν_{td} to be very low. Moreover, we expect acceptance probabilities to be low, and thus the introduction of a lag between samples would be appropriate, as was done in the simulation studies (Section 6) and data analysis (Section 7).

Considering (ii), the influence of ε is much like the scale parameter of a proposal within the familiar random-walk Metropolis-Hastings algorithm. We illustrate this with some simulated examples, shown in Figure 21. These show, for the same posterior but different choices of ε, marginal samples $(\gamma_i)_{i=1}^M$ obtained via our component-wise MCMC algorithm (Section 5.2), that is, by sampling from the joint and keeping only those samples for the dispersion. For each chain $(\gamma_i)_{i=1}^M$ we also report the IACT estimated via (29) with $K = 20$ and the observed acceptance probability. Here one can observe when ε is low the correlation is high, since moves are not aggressive enough, whilst if ε is high many proposals are rejected, leading to periods 'stuck' at certain values, similarly pushing up the correlation.

To maximise efficiency one must therefor find a sweet spot between these two extremes, for example, taking $\varepsilon = 0.4$, as shown in the middle subfigure of Figure 21, appears to strike a good balance.

S7 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.

Algorithm 1: Involutive exchange (iExchnage) algorithm

Input: target density $p(\theta|\mathbf{x}) \propto p(\theta)\gamma(\mathbf{x}|\theta)/Z(\theta)$
Input: auxiliary density $q(u|\theta)$
Input: involution $f(\theta, u)$, i.e. $f^{-1}(\theta, u) = f(\theta, u)$
initialise θ
for $i = 1, \ldots, n$ **do**
  sample $u \sim q(u|\theta)$
  invoke involution $(\theta', u') = f(x, u)$
  sample $\mathbf{y} \sim p(\mathbf{y}|\theta')$
  evaluate $\alpha(\theta, \theta') = \min\left\{1, \frac{p(\theta')\gamma(\mathbf{x}|\theta')\gamma(\mathbf{y}|\theta)q(u'|\theta')}{p(\theta)\gamma(\mathbf{x}|\theta)\gamma(\mathbf{y}|\theta')q(u|\theta)} \left|\frac{\partial f(\theta,u)}{\partial(\theta,u)}\right|\right\}$
  $\theta_i = \begin{cases} \theta' & \text{with probability } \alpha(\theta, \theta') \\ \theta & \text{with probability } 1 - \alpha(\theta, \theta') \end{cases}$
  $\theta \leftarrow \theta_i$
end

Let us first set the context. We have some data $\mathbf{x}$ which is assumed to have been drawn via a model $p(\mathbf{x}|\theta)$, where θ denote parameters, taking the following form

$$p(\mathbf{x}|\theta) = \frac{\gamma(\mathbf{x}|\theta)}{Z(\theta)} \tag{30}$$

where $Z(\theta) = \int \gamma(\mathbf{x}|\theta)d\mathbf{x}$ denotes its normalising constant, assumed to be *intractable*. If one is taking a Bayesian approach to inference and has specified a prior $p(\theta)$, this leads to the following posterior

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} \tag{31}$$

where $p(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta)d\theta$ is the marginal probability of the data, which in most cases is also intractable. Due to these two elements of intractability, such posteriors are often referred to as *doubly*-intractable (Murray et al., 2006). For example, the posteriors resulting from both our SIS and SIM models are doubly-intractable.

A typical approach to circumvent the intractability present in Bayesian posterior distributions is to use MCMC algorithms to sample from them, with the Metropolis-Hastings

(MH) algorithm being a prevalent choice. However, for doubly-intractable posteriors, many standard MCMC algorithms are not feasible. To illustrate, consider using the MH algorithm. Here, with θ the current state and $q(\theta'|\theta)$ some proposal density, in a single iteration one would sample proposal θ' from $q(\theta'|\theta)$ and accept this with the following probability

$$\begin{aligned}\alpha(\theta,\theta') &= \min\left\{1, \frac{p(\theta'|\mathbf{x})q(\theta|\theta')}{p(\theta|\mathbf{x})q(\theta'|\theta)}\right\} \\ &= \min\left\{1, \frac{\gamma(\mathbf{x}|\theta')/Z(\theta')p(\theta')q(\theta|\theta')}{\gamma(\mathbf{x}|\theta)/Z(\theta)p(\theta)q(\theta'|\theta)}\right\},\end{aligned} \tag{32}$$

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 (32), 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.

S7.1 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 prespecified. We also introduce an auxiliary dataset $\mathbf{y}$ 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 (30) (sample from the model)
3. Evaluate acceptance probability

$$\begin{aligned}\alpha(\theta,\theta') &= \min\left\{1, \frac{p(\theta'|\mathbf{x})q(\theta|\theta')p(\mathbf{y}|\theta)}{p(\theta|\mathbf{x})q(\theta'|\theta)p(\mathbf{y}|\theta')}\right\} \\ &= \min\left\{1, \frac{p(\theta')\gamma(\mathbf{x}|\theta')\gamma(\mathbf{y}|\theta)q(\theta|\theta')}{p(\theta)\gamma(\mathbf{x}|\theta)\gamma(\mathbf{y}|\theta')q(\theta'|\theta)}\right\}\end{aligned} \tag{33}$$

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 $i = 1, \ldots, n$ **do**
  sample θ' via $q(\theta'|\theta)$
  sample $\mathbf{y}$ via $p(\mathbf{y}|\theta')$ (from the model)
  evaluate $\alpha(\theta, \theta') = \min\left\{1, \frac{p(\theta')\gamma(\mathbf{x}|\theta')\gamma(\mathbf{y}|\theta)q(\theta|\theta')}{p(\theta)\gamma(\mathbf{x}|\theta)\gamma(\mathbf{y}|\theta')q(\theta'|\theta)}\right\}$
  $\theta_i = \begin{cases} \theta' & \text{with probability } \alpha(\theta, \theta') \\ \theta & \text{with probability } 1 - \alpha(\theta, \theta') \end{cases}$
  $\theta \leftarrow \theta_i$
end
Output: sample $\{\theta_i\}_{i=1}^n$

S7.2 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 (31) (replacing x with θ). Like all MCMC algorithms, it does so by sampling a Markov chain admitting $p(x)$ as its stationary distribution, using in particular a combination of random sampling and involutive deterministic maps. The result is a very general framework which includes many well-known MCMC algorithms as special cases.

As the name suggests, iMCMC uses a particular type of deterministic function know as an *involution*. This is a function which serves as its own inverse, that is, if $f : \mathcal{X} \to \mathcal{X}$ then one has $f^{-1}(x) = f(x)$. Equivalently, a composition f with itself leads to the identity

$$f(f(x)) = x.$$

Towards targeting $p(x)$ one introduces auxiliary variables $u \in \mathcal{U}$ with conditional density $q(u|x)$ over an auxiliary space $\mathcal{U}$ (which need not be equal to $\mathcal{X}$), augmenting the target as follows

$$p(x, u) = p(x)q(u|x)$$

which is now a distribution over $\mathcal{X} \times \mathcal{U}$. Observe this admits $p(x)$ as its marginal and hence one can obtain samples thereof by targeting $p(x, u)$ and disregarding the u samples. To do so, suppose an involution $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X} \times \mathcal{U}$ has been specified along with the auxiliary distribution $q(u|x)$. In structure reminiscent of the MH algorithm, a single iteration consists of the following. With current state (x, u), an auxiliary variable $u \in \mathcal{U}$ is first drawn from $q(u|x)$, before the involution f is invoked to get a proposal $(x', u') = f(x, u)$, which is subsequently accepted with the following probability

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

leading to a Markov chain admitting $p(x, u)$ as its stationary distribution (Neklyudov et al., 2020, Proposition 2).

Observe that since auxiliary variables u are 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.

Algorithm 3: Involutive MCMC (iMCMC)

Input: target density $p(x)$
Input: auxiliary density $q(u|x)$
Input: involution $f(x, u)$
initialise x;
for $i = 1, \ldots, n$ **do**
  sample $u \sim q(u|x)$
  invoke involution $(x', u') = f(x, u)$
  evaluate $\alpha(x, x') = \min\left\{1, \frac{p(x')q(u'|x')}{p(x)q(u|x)} \left|\frac{\partial f(x,u)}{\partial(x,u)}\right|\right\}$
  $x_i = \begin{cases} x' & \text{with probability } \alpha(x, x') \\ x & \text{with probability } 1 - \alpha(x, x') \end{cases}$
  $x \leftarrow x_i$
end
Output: sample $\{x_i\}_{i=1}^n$

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 S7.1. Moreover, we define our involution as follows

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

that is, we simply swap $\theta \leftrightarrow \theta'$. Observe we have

$$\begin{aligned} f(f(\theta, u)) &= f(f(\theta, (\theta', \mathbf{y})) \\ &= f(\theta', (\theta, \mathbf{y})) \\ &= (\theta, (\theta', \mathbf{y})) \\ &= (\theta, u) \end{aligned}$$

so that f is indeed an involution. We now derive the Jacobian term. For convenience, drop the inner parenthesis and write $(\theta, u) = (\theta, \theta', \mathbf{y})$, for which we have $f(\theta, \theta', \mathbf{y}) = (\theta', \theta, \mathbf{y})$.

Now, we have

$$\frac{\partial f(\theta,\theta',\mathbf{y})}{\partial(\theta,\theta',\mathbf{y})} = \begin{bmatrix} \frac{\partial f_1}{\partial\theta} & \frac{\partial f_1}{\partial\theta'} & \frac{\partial f_1}{\partial\mathbf{y}} \\ \frac{\partial f_2}{\partial\theta} & \frac{\partial f_2}{\partial\theta'} & \frac{\partial f_2}{\partial\mathbf{y}} \\ \frac{\partial f_3}{\partial\theta} & \frac{\partial f_3}{\partial\theta'} & \frac{\partial f_3}{\partial\mathbf{y}} \end{bmatrix} = \begin{bmatrix} \frac{\partial\theta'}{\partial\theta} & \frac{\partial\theta'}{\partial\theta'} & \frac{\partial\theta'}{\partial\mathbf{y}} \\ \frac{\partial\theta}{\partial\theta} & \frac{\partial\theta}{\partial\theta'} & \frac{\partial\theta}{\partial\mathbf{y}} \\ \frac{\partial\mathbf{y}}{\partial\theta} & \frac{\partial\mathbf{y}}{\partial\theta'} & \frac{\partial\mathbf{y}}{\partial\mathbf{y}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and taking determinants

$$\left|\frac{\partial f(\theta,\theta',\mathbf{y})}{\partial(\theta,\theta',\mathbf{y})}\right| = 1 \cdot \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} + 0 + 0 = 1.$$

Finally, with $q(\theta'|\theta)$ denoting the proposal density of the exchange algorithm, define the auxiliary distribution as follows

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

where $p(\mathbf{y}|\theta')$ is the likelihood of auxiliary data $\mathbf{y}$ under the assumed model (30). 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).

S7.3 Defining the iExchange Algorithm

We now define our extension of the exchange algorithm. We will assume that an iMCMC scheme to target $p(\theta|\mathbf{x})$ has been defined, that is, auxiliary variables u, involution $f(\theta,u) = (\theta',u')$ and conditional distribution $q(u|\theta)$ have all been specified. When the posterior is doubly-intractable, in general one will not be able to implement this algorithm due to the intractability of the acceptance probability. However, in spirit of the exchange algorithm, we can choose auxiliary variables and their conditional distribution to induce cancellation of normalising constants in the acceptance probability.

In particular, we let $\tilde{u} = (u,\mathbf{y})$, where $\mathbf{y}$ denotes an auxiliary dataset lying in the same space as $\mathbf{x}$. Now, writing $f(\theta,u) = (f_1(\theta,u), f_2(\theta,u)) = (\theta',u')$ we define an involution $g(\theta,\tilde{u})$ as follows

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

for which we have

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

that is, g is indeed an involution. Now, as in Section S7.3, drop the inner parenthesis and write $(\theta, \tilde{u}) = (\theta, u, \mathbf{y})$. The Jacobian is now given by

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

and taking determinants we get the following

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

The final element to define is the auxiliary distribution. Given current state θ we consider sampling $\tilde{u} = (u, \mathbf{y})$ as follows: (i) sample u from $q(u|\theta)$, then (ii) sample $\mathbf{y}$ from $p(\mathbf{y}|\theta')$ (the model) where $\theta' = f_1(\theta, u)$. This leads to the following auxiliary density

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

All the elements of an iMCMC algorithm have now been defined, a single iteration of which consists of the following. Given current state θ, we first sample $\tilde{u} = (u, \mathbf{y})$ via $q(\tilde{u}|\theta)$ as above. We then invoke involution $g(\theta, \tilde{u}) = (\theta', \tilde{u}') = (\theta', (u', \mathbf{y}))$, generating a proposal θ' which we accept with the following probability

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

where as in the exchange algorithm we observe cancellation of normalising constants thanks to the introduction of auxiliary data. Note the Jacobian term here concerns the involution of the original iMCMC scheme to sample from $p(\theta|\mathbf{x})$, and thus the key difference here is the introduction of auxiliary data. The result is what we call the iExchange algorithm (Algorithm 1).

S8 Bayesian Inference: Extra Details

In this section we provide extra details concerning our MCMC scheme for the interaction-sequence models outlined in Section 5, including explicit specification of proposal distributions, involutions and auxiliary distributions, derivations of closed-form acceptance probabilities and pseudocode.

S8.1 Dispersion Conditional

The dispersion conditional can be obtained directly from (8) by conditioning on the mode $\mathcal{S}^m$, in particular we have

$$p(\gamma|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n) \propto Z(\mathcal{S}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m)\right\} p(\gamma). \tag{34}$$

To target (34) we use the exchange algorithm of Murray et al. (2006) (see Supplement S7.1 for background details). As a proposal $q(\gamma'|\gamma)$ we consider sampling γ' uniformly over a ε-neighbourhood of γ with reflection at zero, this is, we first sample $\gamma^* \sim \text{Uniform}(\gamma-\varepsilon, \gamma+\varepsilon)$ and then let $\gamma' = \gamma^*$ if $\gamma^* > 0$ and let $\gamma' = -\gamma^*$ otherwise. The density is thus given by the following (for $\gamma > 0$)

$$q(\gamma'|\gamma) = \begin{cases} \frac{1}{2\varepsilon} & \text{if } \gamma' > 0 \text{ and } \gamma + \gamma' > \varepsilon \\ \frac{1}{\varepsilon} & \text{if } \gamma' > 0 \text{ and } \gamma + \gamma' < \varepsilon \\ 0 & \text{if } \gamma' \leq 0. \end{cases} \tag{35}$$

whilst $q(\gamma'|\gamma) = 0$ for $\gamma \leq 0$. Observe this proposal is symmetric, in that $q(\gamma'|\gamma) = q(\gamma|\gamma)$.

Now, a single iteration consists of the following. Assuming γ is our current state, we first sample proposal γ' from $q(\gamma'|\gamma)$. Next, we sample auxiliary data $\{\mathcal{S}_i^*\}_{i=1}^n$ i.i.d. from the appropriate model, namely

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

which we note implies

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

Finally, we accept this proposal with the following probability

$$\alpha(\gamma, \gamma') = \min\left\{1, H(\gamma, \gamma')\right\} \tag{36}$$

where

$$\begin{aligned} H(\gamma, \gamma') &= \frac{p(\gamma'|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n) p(\{\mathcal{S}_i^*\}_{i=1}^n|\mathcal{S}^m, \gamma)}{p(\gamma|\mathcal{S}^m, \{\mathcal{S}^{(i)}\}_{i=1}^n) p(\{\mathcal{S}_i^*\}_{i=1}^n|\mathcal{S}^m, \gamma')} \frac{q(\gamma|\gamma')}{q(\gamma'|\gamma)} \\ &= \exp\left\{-(\gamma'-\gamma)\left(\sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) - \sum_{i=1}^n d_S(\mathcal{S}_i^*, \mathcal{S}^m)\right)\right\} \frac{p(\gamma')}{p(\gamma)}, \end{aligned} \tag{37}$$

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 11, which details a single accept-reject step for updating the dispersion.

S8.2 Mode Conditional

By conditioning on γ in (8) we get the following form for the mode conditional posterior

$$p(\mathcal{S}^m|\gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n) \propto Z(\mathcal{S}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) - \gamma_0 d_S(\mathcal{S}^m, \mathcal{S}_0)\right\}, \tag{38}$$

which as outlined in Section 5.4 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 S7.

Supposing that auxiliary variables u, involution $f(\mathcal{S}^m, u)$ and auxiliary distribution $q(u|\mathcal{S}^m)$ have all be specified, a single iteration of the iExchange algorithm in this case consists of the following. With γ fixed and $\mathcal{S}^m$ denoting our current state we first sample auxiliary variable u according to $q(u|\mathcal{S}^m)$. We then invoke the involution $f(\mathcal{S}^m, u) = ([\mathcal{S}^m]', u')$, which generates our proposal $[\mathcal{S}^m]'$. Next, we sample auxiliary data $\{\mathcal{S}_i^*\}_{i=1}^n$ i.i.d. where

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

Finally, we accept $[\mathcal{S}^m]'$ with the following probability

$$\alpha\left(\mathcal{S}^m, [\mathcal{S}^m]'\right) = \min\left\{1, H(\mathcal{S}^m, [\mathcal{S}^m]')\right\} \tag{39}$$

where

$$\begin{aligned} H(\mathcal{S}^m, [\mathcal{S}^m]') &= \frac{p([\mathcal{S}^m]' \,|\, \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n)}{p(\mathcal{S}^m \,|\, \gamma, \{\mathcal{S}^{(i)}\}_{i=1}^n)} \frac{p(\{\mathcal{S}_i^*\}_{i=1}^n \,|\, \mathcal{S}^m, \gamma)}{p(\{\mathcal{S}_i^*\}_{i=1}^n \,|\, [\mathcal{S}^m]', \gamma)} \frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)} \\ &= \exp\left\{ -\gamma \left(\sum_{i=1}^n d_S(\mathcal{S}^{(i)}, [\mathcal{S}^m]') - \sum_{i=1}^n d_S(\mathcal{S}^{(i)}, \mathcal{S}^m) \right) \right. \\ &\quad -\gamma \left(\sum_{i=1}^n d_S(\mathcal{S}_i^*, \mathcal{S}^m) - \sum_{i=1}^n d_S(\mathcal{S}_i^*, [\mathcal{S}^m]') \right) \\ &\quad \left. -\gamma_0 \left(d_S([\mathcal{S}^m]', \mathcal{S}_0) - d_S(\mathcal{S}^m, \mathcal{S}_0) \right) \right\} \frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)} \end{aligned} \tag{40}$$

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.

S8.3 Edit Allocation Move

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

$$u = (\delta, \boldsymbol{z}, u_1, \ldots, u_N)$$

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

to the ith path, where d_i is the number of deletions, $\boldsymbol{v}_i$ and $\boldsymbol{v}_i'$ are subsequences indexing entry insertions and deletions and $\boldsymbol{y}_i$ denotes entries to be inserted.

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

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

where

$$u_i' = (z_i - d_i, \boldsymbol{v}_i', \boldsymbol{v}_i, (\mathcal{I}_i)_{\boldsymbol{v}_i}) \tag{41}$$

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

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

furthermore observe that reapplying the operations of (41) to u_i' itself takes us back to u_i

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

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

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

and hence

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

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 5.5.1

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

whilst we sample indexing subsequences $\boldsymbol{v}_i$ and $\boldsymbol{v}_i'$ uniformly. Regarding this latter assumption, recall that $\boldsymbol{v}_i$ is a length d_i (number of deletions) subsequence of $[n_i]$ (n_i is the length of $\mathcal{I}_i$), whilst $\boldsymbol{v}_i'$ is a length $a_i := z_i - d_i$ (number of insertions) subsequence of $[m_i]$, where $m_i = n_i - d_i + a_i$ (length of the ith proposed path $\mathcal{I}_i'$). Thus sampling these uniformly implies

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

Finally, regarding sampling entry insertions we for now assume these are drawn via some general distribution which may be dependent on the current state, namely we assume each $\boldsymbol{y}_i$ was drawn via $q(\boldsymbol{y}|\mathcal{I}_i)$. Together this implies the following closed form for the auxiliary distribution

$$\begin{aligned} q(u|\mathcal{S}^m) &= q(\delta)q(\boldsymbol{z}|\delta)\prod_{i=1}^{N} q(d_i)q(\boldsymbol{v}_i|d_i)q(\boldsymbol{v}_i'|d_i,z_i)q(\boldsymbol{y}_i|\mathcal{I}_i) \\ &= \frac{1}{\nu_{\text{ed}}}\left(\frac{1}{N}\right)^{\delta}\prod_{i=1}^{N}\frac{1}{\min(n_i,z_i)+1}\binom{n_i}{d_i}^{-1}\binom{m_i}{a_i}^{-1}q(\boldsymbol{y}_i|\mathcal{I}_i). \end{aligned} \tag{42}$$

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

$$\begin{aligned} q(u'|[\mathcal{S}^m]') &= q(\delta)q(\boldsymbol{z}|\delta)\prod_{i=1}^{N} q(a_i)q(\boldsymbol{v}_i'|a_i)q(\boldsymbol{v}_i|a_i,z_i)q((\mathcal{I}_i)_{\boldsymbol{v}_i}|\mathcal{I}_i') \\ &= \frac{1}{\nu_{\text{ed}}}\left(\frac{1}{N}\right)^{\delta}\prod_{i=1}^{N}\frac{1}{\min(m_i,z_i)+1}\binom{m_i}{a_i}^{-1}\binom{n_i}{d_i}^{-1}q((\mathcal{I}_i)_{\boldsymbol{v}_i}|\mathcal{I}_i'). \end{aligned} \tag{43}$$

Taking the ratio of (43) and (42) leads to the following

$$\frac{q(u'|[\mathcal{S}^m]')}{q(u|\mathcal{S}^m)} = \prod_{i=1}^{N}\frac{\min(n_i,z_i)+1}{\min(m_i,z_i)+1}\frac{q((\mathcal{I}_i)_{\boldsymbol{v}_i}|\mathcal{I}_i')}{q(\boldsymbol{y}_i|\mathcal{I}_i)}. \tag{44}$$

which is the key term appearing in the acceptance probability of this move, as seen in (40).

We finalise these details on the edit allocation move with a discussion on entry insertion distributions. The simplest option here is to sample entries uniformly over the vertex set $\mathcal{V}$. In this case, with $V = |\mathcal{V}|$, we have

$$q(\boldsymbol{y}_i|\mathcal{I}_i) = \left(\frac{1}{V}\right)^{a_i} \tag{45}$$

which implies

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

any of which can be plugged into (44).

As an alternative choice, one can consider informing the entry insertions from observed data. This approach is based on the following assumption: If two vertices have been observed in the same path across many observations then the probability of proposing one given the other is already present should be higher within the MCMC algorithm.

To reflect this assumption in a proposal, we first extract the necessary information from the observed data. Letting

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

denote the observed sample we construct a *co-occurrence matrix* $\mathbf{A} \in \mathbb{Z}_{\geq 0}^{V\times V}$, defined as follows

$$\begin{aligned} \mathbf{A}_{vv'} &= \#\text{observations with an interaction containing both } v \text{ and } v' \\ &= |\{k : \exists\, \mathcal{I} \in \mathcal{S}^{(k)} \text{ with } v, v' \in \mathcal{I}\}| \end{aligned}$$

where $v \neq v'$, whilst for $v = v'$ we let

$$\begin{aligned}\mathbf{A}_{vv} &= \#\text{observations with an interaction containing } v \text{ at least twice} \\ &= |\{k : \exists\, \mathcal{I} \in \mathcal{S}^{(k)} \text{ with } v \in \mathcal{I} \text{ at least twice}\}|,\end{aligned}$$

which can be seen as the adjacency matrix of a weighted graph describing the co-occurrence structure observed in the data. Now, given $\mathbf{A}$ we construct a probability matrix $\mathbf{P} \in \mathbb{R}^{V \times V}$ by normalising the rows, that is

$$\mathbf{P}_{vv'} = \mathbf{A}_{vv'}/Z_v$$

where $Z_v = \sum_{v' \in \mathcal{V}} \mathbf{A}_{vv'}$ is the normalising constant of the vth row. Intuitively, the entry $\mathbf{P}_{vv'}$ can be seen as the probability of observing v' in an interaction given v is known to already be present. We consider using $\mathbf{P}$ to inform entry insertions as follows. Suppose that $\mathcal{I}_i = (x_{i1}, \ldots, x_{in_i})$ denotes the path being edited, with $\boldsymbol{v}_i$ denoting the subsequence of $[n_i]$ indexing which entries are to be deleted. Introduce the notation $\boldsymbol{v}_i^c$ for the complement of $\boldsymbol{v}_i$, which is the subsequence of $[n_i]$ containing the entries not in $\boldsymbol{v}_i$. For example, with $\boldsymbol{v} = (1, 2, 5) \in [5]$ we would have $\boldsymbol{v}^c = (3, 4)$. Now, observed that $(\mathcal{I}_i)_{\boldsymbol{v}_i^c}$ denotes the entries of $\mathcal{I}_i$ *not* being deleted, that is, those being preserved. Our approach is to now propose entries which have often been observed in the data alongside those being preserved. Since each unique preserved entry has an associated distribution over $\mathcal{V}$ given by the respective row of $\mathbf{P}$, we can consider mixing these distributions together with equal weight to form an entry proposal distribution. In particular, we sample entry insertions for the ith path i.i.d. via the following

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

One can also introduce a tuning parameter to control the extent to which proposals are informed by the data. In particular, with $\alpha > 0$ first alter the probability matrix as follows

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

which normalises to

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

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

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

where as $\alpha \to \infty$ this will converge to a mixture of uniform distributions over $\mathcal{V}$, that is, also a uniform distribution. In this way, one has a proposal which is informed by the data, but becomes less informed as the tuning parameter $\alpha \to \infty$.

We finish with a note regarding evaluation of (44) for this informed proposal. Supposing that $\mathcal{I}_i'$ is ith path in the proposal $[\mathcal{S}^m]'$ (obtained by deleting d_i entries of $\mathcal{I}_i$ indexed by $\boldsymbol{v}_i$, and inserting entries $\boldsymbol{y}_i$ at locations indexed by $\boldsymbol{v}_i'$), then observe we have $(\mathcal{I}_i)_{\boldsymbol{v}_i^c} = (\mathcal{I}_i')_{(\boldsymbol{v}_i')^c}$

(preserved entries) which thus implies $q_\alpha(y|\mathcal{I}_i) = q_\alpha(y|\mathcal{I}'_i)$. Consequently we can write the following

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

and hence only the single mixed distribution $q_\alpha(y|\mathcal{I}_i)$ needs to be constructed. This is helpful to bare in mind when evaluating (44).

S8.4 Path Insertion and Deletion Move

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

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

where (i) ε denotes the total number of paths to be inserted or deleted, (ii) d denotes the number of paths to be deleted, implying $a = \varepsilon - d$ insertions, (iii) $\boldsymbol{v}$ and $\boldsymbol{v}'$ denote subsequences indexing path deletions and insertions respectively, and (iv) $(\mathcal{I}^*_1, \ldots, \mathcal{I}^*_a)$ denote the paths to be inserted.

We now define the involution of this iMCMC move. As outlined in Section 5.5.2, 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 a follows

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

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

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

furthermore, reapplying the second component just defined leads to

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

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 5.5.2

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

whilst we sample indexing subsequences $\boldsymbol{v}$ and $\boldsymbol{v}'$ uniformly and assume path insertions a drawn via some general distribution $q(\mathcal{I}|\mathcal{S}^m)$. In this instance, recall that $\boldsymbol{v}$ is a subsequence of $[N]$ of size d, whilst $\boldsymbol{v}'$ is a subsequence of $[M]$ of size a, where $M = N - d + a$ is the length of $[\mathcal{S}^m]'$. Sampling these uniformly thus implies

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

leading to the following closed form

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

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

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

Taking the ratio of these leads to the following

$$\frac{q(u'|[\mathcal{S}^m]')}{q(u|\mathcal{S}^m)} = \frac{\min(N,\varepsilon)+1}{\min(M,\varepsilon)+1}\frac{\prod_{i=1}^{d} q(\mathcal{I}_{v_i}|[\mathcal{S}^m]')}{\prod_{i=1}^{a} q(\mathcal{I}'_{v'_i}|\mathcal{S}^m)}, \tag{46}$$

which can be substituted into (40) to evaluate the acceptance probability of this move (here we again use the fact $\mathcal{I}'_{v'_i} = \mathcal{I}_i^*$).

We finalise by discussing possible choices for the path insertion distribution. The simplest approach is to combine a distribution on path length with uniform sampling of entries. In particular, to sample some path $\mathcal{I} = (x_1, \ldots, x_m)$ we (i) sample its length m via some distribution $q(m)$ (ii) sample entries x_i uniformly from $\mathcal{V}$. This implies

$$q(\mathcal{I}|\mathcal{S}^m) = q(\mathcal{I}) = q(m)\left(\frac{1}{V}\right)^m$$

where $V = |\mathcal{V}|$, which can be substituted into (46).

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

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

a sample, for each $v \in \mathcal{V}$ we let

$$c_v = |\{k : \exists \mathcal{I} \in \mathcal{S}^{(k)} \text{ with } v \in \mathcal{I}\}|$$

denote the number of observations with at least one path containing the vertex v. Normalising this leads to

$$p_v = \frac{c_v}{\sum v \in \mathcal{V} c_v}$$

which can be seen as the probability a randomly selected observation contains v. Introducing the parameter $\alpha > 0$ we let

$$q_\alpha(v) \propto p_v + \alpha$$

which normalises to

$$q_\alpha(v) = \frac{p_v + \alpha}{1 + \alpha V}.$$

One can now use this to sample path entries, namely to sample $\mathcal{I} = (x_1, \ldots, x_m)$ we (i) sample length m via some $q(m)$, (ii) sample entries x_i via $q_\alpha(x_i)$. Observe that if $\alpha = 0$ we have $q_\alpha(v) = p_v$, and the entry insertion distribution is fully informed by the data, whilst as $\alpha \to \infty$ we have $q_\alpha(v) \to 1/V$, and we recover uniform entry insertions.

S8.5 Model Sampling

In this section we provide supporting details regarding our iMCMC algorithm to sample from the SIS models outlined in Section 5.6. Recall that for the SIS model (Definition 1) the (normalised) probability of observing $\mathcal{S}$ is given by

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

implying the following closed form for the acceptance probability (12)

$$\alpha(\mathcal{S}, \mathcal{S}') = \min\left\{1, H(\mathcal{S}, \mathcal{S}')\right\} \tag{47}$$

where

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

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

As mentioned in Section 5.6, we consider re-using the iMCMC moves of our iExchange scheme used to sample from the mode conditional (Supplements S8.3 and S8.4). 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 S8.3 (replacing $\mathcal{S}^m$ with $\mathcal{S}$ and $[\mathcal{S}^m]'$ with $\mathcal{S}'$) with a uniform entry insertion distribution (45). With $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ the current state, supposing $u = (\delta, \boldsymbol{z}, u_1, \ldots, u_N)$ has been sampled via $q(u|\mathcal{S})$ mapping to $(\mathcal{S}', u') = f(\mathcal{S}, u)$ with $\mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_N)$ we will have

$$\frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} = \prod_{i=1}^{N} \frac{\min(n_i, z_i) + 1}{\min(m_i, z_i) + 1} \left(\frac{1}{V}\right)^{n_i - m_i} \tag{48}$$

  where n_i and m_i denote the lengths of the ith 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 S8.4 (again using $\mathcal{S}$ and $\mathcal{S}'$ instead of $\mathcal{S}^m$ and $[\mathcal{S}^m]'$). With $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ the current state, supposing $u = (\varepsilon, d, \boldsymbol{v}, \boldsymbol{v}', \mathcal{I}_1^*, \ldots, \mathcal{I}_a^*)$ (where $a = \varepsilon - d$) has been sampled via $q(u|\mathcal{S})$ mapping to $(\mathcal{S}', u') = f(\mathcal{S}, u)$ with $\mathcal{S}' = (\mathcal{I}_1', \ldots, \mathcal{I}_M')$ we will have

$$\frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})} = \frac{\min(N, \varepsilon) + 1}{\min(M, \varepsilon) + 1} \frac{\prod_{i=1}^{d} q(\mathcal{I}_{v_i}|\mathcal{S}')}{\prod_{i=1}^{a} q(\mathcal{I}'_{v'_i}|\mathcal{S})}. \tag{49}$$

As mentioned in Section 5.6, 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 13.

S9 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 5, 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.

S9.1 Priors, Hierarchical Model and Posterior

To specify priors, we follow Section 5.1 and assume the mode was itself sampled from an SIM model, namely

$$\mathcal{E}^m \sim \text{SIM}(\mathcal{E}_0, \gamma_0)$$

where $(\mathcal{E}_0, \gamma)$ are hyperparameters, whilst we assume the dispersion was drawn from some distribution $p(\gamma)$ whose support is a subset of the non-negative reals. Given these specifications, an observed sample $\{\mathcal{E}^{(i)}\}_{i=1}^n$ is assumed to be drawn via

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

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

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

which implies a posterior given by

$$\begin{aligned} p(\mathcal{E}^m, \gamma \,|\, \{\mathcal{E}^{(i)}\}_{i=1}^n) &\propto p(\{\mathcal{E}^{(i)}\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma) p(\mathcal{E}^m) p(\gamma) \\ &= Z(\mathcal{E}^m, \gamma)^{-n} \exp\left\{-\gamma \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m)\right\} \\ &\quad \exp\{-\gamma_0 d_E(\mathcal{E}^m, \mathcal{E}_0)\} p(\gamma). \end{aligned} \tag{50}$$

S9.2 Posterior Sampling

As for the interaction-sequence models, we consider sampling from the posterior (50) 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 (50) 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 5. 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 5 and Supplement S8) to target them.

S9.3 Dispersion Conditional

Conditioning on $\mathcal{E}^m$ in (50) 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 5.3 and Supplement S8.1 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 S8.1). 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

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

for which we have

$$p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma') = Z(\mathcal{E}^m, \gamma')^{-n} \exp\left\{-\gamma' \sum_{i=1}^n d_E(\mathcal{E}_i^*, \mathcal{E}^m)\right\}.$$

Finally, we accept this proposal with the following probability

$$\alpha(\gamma, \gamma') = \min\{1, H(\gamma, \gamma')\} \tag{51}$$

where

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

where, as in Supplement S8.1, normalising constants of the (conditional) posterior and auxiliary data cancel one another out, whilst the proposal density terms cancel due to its symmetry.

S9.4 Mode Conditional

Conditioning on γ in (50) we have the following

$$p(\mathcal{E}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n) \propto Z(\mathcal{E}^m, \gamma)^{-n} \exp\left\{ -\gamma \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m) - \gamma_0 d_E(\mathcal{E}^m, \mathcal{E}_0) \right\}, \tag{52}$$

which is a distribution over $\boldsymbol{\mathcal{E}}^*$, that is, the space of multisets. To re-use the iExchange scheme of Section 5.4 we instead need a distribution over the space of interaction sequences $\boldsymbol{\mathcal{S}}^*$. To this end, we extend (52) to a distribution over interaction sequences.

Consider the general problem of extending some distribution $\pi(\mathcal{E})$ over $\boldsymbol{\mathcal{E}}^*$ to one over $\boldsymbol{\mathcal{S}}^*$. Firstly, observe each $\mathcal{E}$ is associated with a set of sequences, obtained by placing the interactions of $\mathcal{E}$ in different orders. More formally, $\mathcal{E}$ can be seen as equivalence class of sequences (see Appendix A). As such, one can consider assigning equal probability to each unique ordering of $\mathcal{E}$. In particular, for $\mathcal{S} \in \boldsymbol{\mathcal{S}}^*$ we let

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

where $\mathcal{E}$ is the multiset obtained from $\mathcal{S}$ by disregarding the order of interactions, and $A(\mathcal{E})$ denotes the number of unique orderings of the paths in $\mathcal{E}$.

The form of $A(\mathcal{E})$ can be obtained as follows. Suppose that $\mathcal{E}$ consists of N paths, with $M \leq N$ *unique* paths. Without loss of generality label the unique paths 1 to M and let w_i denote the multiplicity of the ith path. Now, if each path of $\mathcal{E}$ is different there are $N!$ possible ways to order them. However, if there are repeated paths this will include double counting. Therefore, in general we must further divide by $(w_i)!$ leading to the familiar multinomial term

$$A(\mathcal{E}) := \binom{N}{w_1, \ldots, w_N} = \frac{N!}{w_1! \cdots w_N!}. \tag{53}$$

Through this reasoning we can extend (52) as follows

$$\tilde{p}(\mathcal{S}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n) = \frac{1}{A(\mathcal{E}^m)} p(\mathcal{E}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n) \tag{54}$$

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

We can now re-use the iExchange algorithm of Section 5.4 and Supplement S8.2 to target (54). However, note the normalising constant appearing in (52), and hence also in (54), 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 5, 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

$$\mathcal{E}_i^* \sim \mathrm{SIM}([\mathcal{E}^m]', \gamma)$$

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

$$\alpha(\mathcal{E}^m, [\mathcal{E}^m]') = \min\left\{1, H(\mathcal{E}^m, [\mathcal{E}^m]')\right\} \tag{55}$$

where

$$\begin{aligned}
H(\mathcal{E}^m, [\mathcal{E}^m]') &= \frac{\tilde{p}([\mathcal{S}^m]' \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)}{\tilde{p}(\mathcal{S}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)} \frac{p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma)}{p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, [\mathcal{E}^m]', \gamma)} \frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)} \\
&= \frac{\frac{1}{A([\mathcal{E}^m]')} p([\mathcal{E}^m]' \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)}{\frac{1}{A(\mathcal{E}^m)} p(\mathcal{E}^m \,|\, \gamma, \{\mathcal{E}^{(i)}\}_{i=1}^n)} \frac{p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, \mathcal{E}^m, \gamma)}{p(\{\mathcal{E}_i^*\}_{i=1}^n \,|\, [\mathcal{E}^m]', \gamma)} \frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)} \\
&= \frac{A(\mathcal{E}^m)}{A([\mathcal{E}^m]')} \exp\left\{ -\gamma \left(\sum_{i=1}^n d_E(\mathcal{E}^{(i)}, [\mathcal{E}^m]') - \sum_{i=1}^n d_E(\mathcal{E}^{(i)}, \mathcal{E}^m) \right) \right. \\
&\quad - \gamma \left(\sum_{i=1}^n d_E(\mathcal{E}_i^*, \mathcal{E}^m) - \sum_{i=1}^n d_E(\mathcal{E}_i^*, [\mathcal{E}^m]') \right) \\
&\quad \left. - \gamma_0 \left(d_E([\mathcal{E}^m]', \mathcal{E}_0) - d_E(\mathcal{E}^m, \mathcal{E}_0) \right) \right\} \frac{q(u' \,|\, [\mathcal{S}^m]')}{q(u \,|\, \mathcal{S}^m)}
\end{aligned} \tag{56}$$

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

$$\frac{A(\mathcal{E}^m)}{A([\mathcal{E}^m]')} = \frac{N!\,(w_1'! \cdots w_{M'}'!)}{M!\,(w_1! \cdots w_M!)} \tag{57}$$

where N and M are the cardinalities of $\mathcal{E}^m$ and $[\mathcal{E}^m]'$ respectively, w_i is the multiplicity of the ith unique path in $\mathcal{E}^m$ and w'_i is the multiplicity of the ith unique path in $[\mathcal{E}^m]'$.

Clearly, this all depends on a particular iMCMC specification (auxiliary variables, involution and auxiliary distribution). For this we can use the edit allocation (Section 5.5.1 and Supplement S8.3) and interaction insertion and deletion (Supplement S8.4) 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 16.

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

S9.5 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 5.6), 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 S9.4 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 5.6 and Supplement S8.5). Recalling that for the SIM model (Definition 2) the (normalised) probability of observing $\mathcal{E} \in \boldsymbol{\mathcal{E}}^*$ is given by

$$p(\mathcal{E}|\mathcal{E}^m, \gamma) = \frac{1}{Z(\mathcal{E}^m, \gamma)} \exp\{-\gamma d_E(\mathcal{E}, \mathcal{E}^m)\},$$

we can assign any $\mathcal{S} \in \boldsymbol{\mathcal{S}}^*$ the following probability

$$\tilde{p}(\mathcal{S}|\mathcal{E}^m, \gamma) = \frac{1}{A(\mathcal{E})} p(\mathcal{E}|\mathcal{E}^m, \gamma) \tag{58}$$

where $\mathcal{E}$ is multiset obtain from $\mathcal{S}$ by disregarding order, and $A(\mathcal{E})$ is as defined in (53), thus defining an extended distribution over $\boldsymbol{\mathcal{S}}^*$.

We can now target (58) via iMCMC as in Section 5.6. In particular, suppose that one would like to sample from an SIM$(\mathcal{E}^m, \gamma)$ model. With u, $q(u|\mathcal{S})$ and $f(\mathcal{S}, u)$ some iMCMC specification as used therein, and $\mathcal{E}$ the current state, a single iteration of will consist of the following

1. Construct interaction sequence $\mathcal{S}$ by placing the paths of $\mathcal{E}$ in an arbitrary order
2. Sample $u \sim q(u|\mathcal{S})$
3. Invoke involution $f(\mathcal{S}, u) = (\mathcal{S}', u')$
4. Disregard order in $\mathcal{S}'$ to obtain proposed multiset $\mathcal{E}'$

5. Evaluate the following probability

$$\alpha(\mathcal{E}, \mathcal{E}') = \min\left\{1, \frac{\tilde{p}(\mathcal{S}'|\mathcal{E}^m, \gamma)}{\tilde{p}(\mathcal{S}|\mathcal{E}^m, \gamma)} \frac{q(u'|\mathcal{S}')}{q(u|\mathcal{S})}\right\} \tag{59}$$

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 5.6 and recycle the edit allocation (Section 5.5.1 and Supplement S8.3) and path insertion/deletion moves (Section 5.5.2 and Supplement S8.4), again mixing them together with proportion $\beta \in (0, 1)$, left as a tuning parameter.

A closed form for (59) can be derived as follows. Writing $\alpha(\mathcal{E}, \mathcal{E}') = \min\{1, H(\mathcal{E}, \mathcal{E}')\}$ we have

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

where

$$\frac{A(\mathcal{E})}{A(\mathcal{E}')} = \frac{N!\,(w_1'! \cdots w_{M'}'!\,)}{M!\,(w_1! \cdots w_M!\,)}$$

with N and M the cardinalities of $\mathcal{E}$ and $\mathcal{E}'$ respectively, w_i the multiplicity of the ith unique path in $\mathcal{E}$ and w_i' the multiplicity of the ith unique path in $\mathcal{E}'$. As when sampling from the interaction-sequence models (Supplement S8.5), the ratio $q(u'|\mathcal{S}')/q(u|\mathcal{S})$ will be move dependent and identical to those appearing in Supplement S8.5, namely (48) for the edit allocation move and (49) for the path insertion/deletion move. The whole procedure to sample from the SIM models is summarised in the pseudocode of Algorithm 17.

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 S9.3 and Supplement S9.4 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 17).

S10 Simulation Study Parameter Choices

This section, we discuss how parameters were chosen for the simulation of Section 6.2. Recall in this case 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. Here we will discuss how such values for α were chosen.

As mentioned in Section 6.2, 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

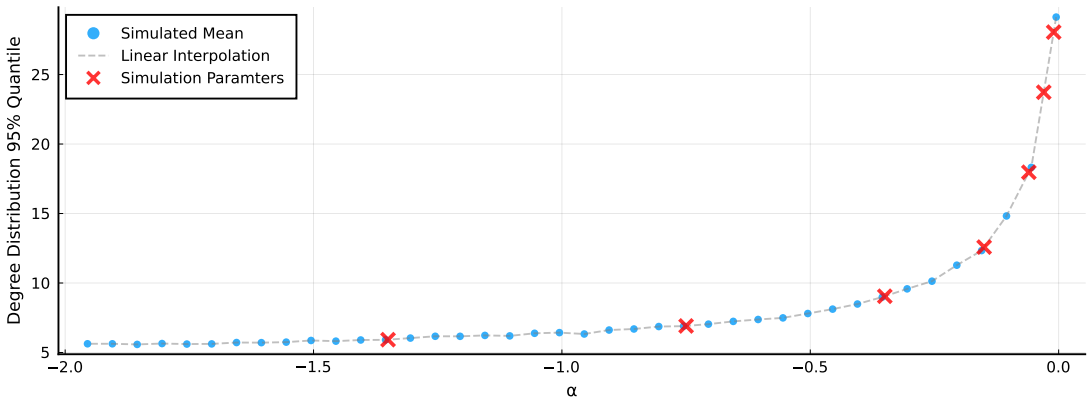


Figure 22: Summary of Hollywood model simulation used to select parameters for simulation of Section 6.2. Plot shows simulated mean degree distribution 95% quantiles for Hollywood$(\alpha, -\alpha V, \nu)$ model, where $V = 20$, $\nu = \text{TrPoisson}(3, 1, 10)$ and α varies. Via linear interpolation (dashed line), we choose α values (crosses) to get an even spread over the expected degree distribution quantiles.

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 sampled randomly from a Hollywood model). In particular, for a range of α values, we (i) drew a sample $\{\mathcal{S}^{(i)}\}_{i=1}^{n}$, from a Hollywood$(\alpha, -\alpha V, \nu)$ model, taking ν and V as above, drawing a total of $N = 10$ paths in each case, then (ii) for $i = 1, \ldots, n$ we evaluated the 95% quantile of the sample $\{k_{\mathcal{S}^{(i)}}(v) : v \in \mathcal{V}, k_{\mathcal{S}^{(i)}}(v) > 0\}$, before returning the mean value of these quantiles.

Figure 22 summarises the output with $n = 1000$ samples, where circular markers show the mean quantiles. Towards choosing simulation parameters, we then used linear interpolation to construct a function mapping all $\alpha < 0$ to an expected quantil, as shown in Figure 22 by the dashed lines. With this, we selected α values (red crosses) providing an even spread of expected degree-distribution 95% qauntiles.

S11 Real Data Analysis

In this section, we provide details supporting the data analysis of Section 7. This includes further details on the data and how it was processed, and extra information regarding the integer-weighted extension of the SNF model (Lunagómez et al., 2021) used in Section 7.3.

S11.1 Foursquare Data Processing

The data analysed in Section 7 was obtained from the New York and Tokyo dataset of Yang et al. (2015), 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).

S11.1.1 Venue Category Hierarchy

As discussed in Section 7, 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 dataset 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 dataset it appears that Foursquare have changed how they label venues, thus there is another set of venue category names (see here). However, the dataset of Yang et al. (2015) appears to be congruent with the former. Using this information we were able to essentially 'fill-in' the higher-level category labels for each category name appearing in the dataset of Yang et al. (2015), mapping their low-level labels to top-level ones.

S11.1.2 Data Filtering

As mentioned in Section 7, we analysed only a subset of 50 interaction networks. This was due to issues causes by the presence of outliers. In particular, it was seen that the

inclusion of a few observations of significantly different size, for example, with many more interactions, or observations which shared little in common with the others, could result in an inferred mode that was empty, that is, an interaction network with no interactions. Clearly, such an inference provides little insight, making this an undesirable scenario. In addition, the MCMC scheme in such cases often showed poor mixing. In this subsection, we outline exactly how this subset of data points was chosen.

Following processing of the raw data and some initial filtering, including the removal of all length one paths and observations with less than 10 paths, we were left with a sample of interaction multisets $\{\mathcal{E}^{(i)}\}_{i=1}^{n}$ with $n = 402$, from which we now select a subset. This we did using a given distance $d_E(\cdot,\cdot)$ between interaction multisets. In particular, a subset of size m was chosen as follows: find the data point which has the smallest total distance to its m nearest neighbours, taking this neighbourhood as the subset. More formally, introducing the notation $\mathcal{N}_m(\mathcal{E})$ for the indices of the m nearest neighbours of $\mathcal{E}$ with respect to d_E in the sample, we let

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

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 took that used in the model-fit, that is, 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, took the following

$$\bar{d}_{\mathrm{M}}(\mathcal{E},\mathcal{E}') = \frac{2d_{\mathrm{M}}(\mathcal{E},\mathcal{E}')}{d_{\mathrm{M}}(\mathcal{E},\emptyset) + d_{\mathrm{M}}(\mathcal{E}',\emptyset) + d_{\mathrm{M}}(\mathcal{E},\mathcal{E}')}$$

where here $\emptyset$ denotes the empty multiset, which functions as our reference element in the space of interaction multisets. This transformation appears in Donnat and Holmes (2018), where it is also referred to as the *Steinhaus transform*, and Deza and Deza (2009), who refer to it as the *biotope transform metric* (Section 4.1 therein).

Note that $d_{\mathrm{M}}(\mathcal{E},\emptyset)$ 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 $d_{\mathrm{LSP}}(\mathcal{I},\Lambda)$, where Λ denotes empty path, which for the LSP distance is equivalent to the the path length. It can be shown that for this new distance one has $0 \leq \bar{d}_{\mathrm{M}}(\mathcal{E},\mathcal{E}') \leq 1$, where $\bar{d}_{\mathrm{M}}(\mathcal{E},\mathcal{E}') = 1$ 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

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

which appears to contradict this intuition. The problem here is the difference in the observation sizes; though $\mathcal{E}^{(2)}$ is more similar to $\mathcal{E}$ it is also larger, hence pushing up its distance.

However, by taking sizes into account, the normalised distances evaluate to

$$\bar{d}_{\mathrm{M}}(\mathcal{E}, \mathcal{E}^{(1)}) = \frac{2 \times 6}{3+3+6} = 1 \qquad \bar{d}_{\mathrm{M}}(\mathcal{E}, \mathcal{E}^{(2)}) = \frac{2 \times 6}{3+9+6} = \frac{2}{3}$$

which better agrees with the intuition that $\mathcal{E}^{(2)}$ is closer to $\mathcal{E}$. As such, if we use the normalised distance we are likely to select a sample of data 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.

S11.2 Multigraph SNF Model

Here we provide extra details regarding the generalisation of the SNF models (Lunagómez et al., 2021) used in Section 7.3. 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 $A^{\mathcal{G}}_{ij} \in \mathbb{Z}_{\geq 0}$ denotes the multiplicity of the edge (i, j) in $\mathcal{E}$.

To define a model, we place a probability distribution over *all* multigraphs (over the vertex set $\mathcal{V}$). This space, which we denote $\mathscr{G}$, can be defined via the one-to-one correspondence with adjacency matrices as follows

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

so that we seek to assign each $\mathcal{G} \in \mathscr{G}$ a probability. Following the same rationale as the SNF models (and the models of this paper), we construct this model via location and scale. Moreover, this is done with the use of distance metrics, this time between multigraphs. We have two parameters, the mode $\mathcal{G}^m \in \mathscr{G}$ (location) and the dispersion $\gamma > 0$ (scale). We also assume that a distance metric has been pre-specified $d_G(\mathcal{G}, \mathcal{G}')$, quantifying the dissimilarity of any two multigraphs $\mathcal{G}$ and $\mathcal{G}'$. Given this, we assume the probability of $\mathcal{G} \in \mathscr{G}$ is, up to proportionality, the following

$$p(\mathcal{G} | \mathcal{G}^m, \gamma) \propto \exp\{-\gamma \phi(d_G(\mathcal{G}, \mathcal{G}^m))\} \tag{60}$$

where $\phi(\cdot)$ is a non-negative strictly increasing function with $\phi(0) = 0$. The notation $\mathcal{G} \sim \mathrm{SNF}(\mathcal{G}^m, \gamma)$ is used when $\mathcal{G}$ is assumed to have been sampled from this probability distribution. The normalising constant of (60) is given by the following

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

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

Note this is more-or-less identical the SNF models seen in Lunagómez et al. (2021), Definition 3.4. The only differences being (i) the sample space $\mathscr{G}$ is now all muligraphs over $\mathcal{V}$, and (ii) the distance metrics $d_G(\cdot, \cdot)$ are between multigraphs.

Supposing that a sample of multigraphs $\{\mathcal{G}^{(i)}\}_{i=1}^n$ has been observed, as discussed in Section 7.3, we can use this mutligraph-based SNF to construct the following hierarchical model

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

where $\mathcal{G}_0 \in \mathscr{G}$ and $\gamma_0 > 0$ are hyperparameters, and $p(\gamma)$ denotes a prior for the dispersion. The goal of inference is to now estimate $\mathcal{G}^m$ and γ, representing notations of average and precision, respectively, and can be achieved by sampling from the posterior via MCMC. The posterior in this case is given by the following

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

which, since $Z(\mathcal{G}^m, \gamma)$ is intractable and depends on the parameters being sampled, is doubly-intractable (Murray et al., 2006). As such, to sample from it one must use a specialised MCMC algorithm. Since we are dealing with multigraphs, we cannot apply the scheme proposed by Lunagómez et al. (2021) directly, and instead propose an alternative approach via the exchange algorithm (Murray et al., 2006). In particular, we considered a component-wise MCMC algorithm which alternates between sampling from the two conditionals (i) $p(\gamma | \mathcal{G}^m, \{\mathcal{G}^{(i)}\}_{i=1}^n)$, and (ii) $p(\mathcal{G}^m | \gamma, \{\mathcal{G}^{(i)}\}_{i=1}^n)$. For (i) we apply the exchange algorithm directly, whilst for (ii) do an exchange-within-Gibbs step, updating each edge in turn in a single repetition.

We first outline the procedure to update the dispersion. Assume that $q(\gamma' | \gamma)$ denotes a suitable proposal density. With $\mathcal{G}^m$ fixed and current state γ, first sample proposal γ' from $q(\gamma' | \gamma)$. Next, sample auxiliary data $\{\mathcal{G}_i^*\}_{i=1}^n$ i.i.d. where $\mathcal{G}_i^* \sim \text{SNF}(\mathcal{G}^m, \gamma')$ and then accept γ' with the following probability

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

where

$$\begin{aligned}H(\gamma', \gamma) = \exp\Bigg\{ &- (\gamma' - \gamma) \left(\sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^m)) - \sum_{i=1}^n \phi(d_G(\mathcal{G}_i^*, \mathcal{G}^m))\right) \Bigg\}\\ &\times \frac{p(\gamma')}{p(\gamma)} \frac{q(\gamma | \gamma')}{q(\gamma' | \gamma)}.\end{aligned} \tag{61}$$

For the proposal $q(\gamma' | \gamma)$ we consider sampling uniformly over a ε-neighbourhood of γ with reflection at zero, as defined in (35), for which one has $q(\gamma' | \gamma) = q(\gamma | \gamma')$.

To update the mode, we consider a exchange-within-Gibbs scheme, whereby we scan through all edges, propose new multiplicities and accept these with some probability. Assume one has defined a proposal $q(x'|x)$, which proposes a new edge multiplicity $x' \in \mathbb{Z}_{\geq 0}$ given current value $x \in \mathbb{Z}_{\geq 0}$. With γ fixed and current state $\mathcal{G}^m$, with A^m its adjacency matrix (abbreviating notation for readability), we first generate proposal $\mathcal{G}^{m'}$ by proposing a new multiplicity for (i,j). More precisely, letting $x = A^m_{ij}$ 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

$$A^{m'}_{kl} = \begin{cases} x' & \text{if } (k,l) = (i,j) \\ A^m_{kl} & \text{else} \end{cases}$$

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 $A^{m'}_{ij} = A^{m'}_{ij} = x'$, since the adjacency matrices must be symmetric. Here we will assumed 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

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

where

$$\begin{aligned} H(\mathcal{G}^{m'}, \mathcal{G}^m) = \exp\Bigg\{ &- \gamma\left(\sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^{m'})) - \sum_{i=1}^n \phi(d_G(\mathcal{G}^{(i)}, \mathcal{G}^m))\right) \\ &- \gamma\left(\sum_{i=1}^n \phi(d_G(\mathcal{G}^*_i, \mathcal{G}^m)) - \sum_{i=1}^n \phi(d_G(\mathcal{G}^*_i, \mathcal{G}^{m'}))\right) \\ &- \gamma_0\left(\phi(d_G(\mathcal{G}^{m'}, \mathcal{G}_0)) - \phi(d_G(\mathcal{G}^m, \mathcal{G}_0))\right)\Bigg\} \frac{q(x|x')}{q(x'|x)}. \end{aligned} \tag{62}$$

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 (62).

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 $(\mathcal{G}^m, \gamma)$. Unfortunately, this cannot be done in general. However, we can replace this with approximate MCMC-based samples, exactly as we did for our interaction-sequence and interaction-multiset models (Section 5.6). To do so, we re-use the scheme above (without auxiliary sampling).

In particular, with current state $\mathcal{G}$, we update edge (i, j) as follows. Letting $x = A_{ij}^{\mathcal{G}}$, we sample x' from $q(x'|x)$ (via ν-neighbourhood as above), constructing proposal $\mathcal{G}'$ via its adjacency matrix

$$A_{kl}^{\mathcal{G}'} = \begin{cases} x' & \text{if } (k, l) = (i, j) \\ A_{kl}^{\mathcal{G}} & \text{else} \end{cases}$$

that is, $\mathcal{G}'$ is equivalent to $\mathcal{G}$ with the multiplicity of edge (i, j) flipped from x to x'. We then accept $\mathcal{G}'$ with the following probability

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

Note this will update a single edge (i, j). One could now follow the approach used to update the mode $\mathcal{G}^m$, looping over all edges in turn. However, in this case we opt to instead choose a single edge at random to update. That is, in a single iteration, we choose (i, j) uniformly from $\mathcal{V} \times \mathcal{V}$, and update it as above. This can be seen as a Gibbs sampler with a randomised sweep strategy (Levine and Casella, 2006).

References

Deza, M. M. and Deza, E. (2009). *Encyclopedia of Distances*. Springer.

Donnat, C. and Holmes, S. (2018). Tracking network dynamics: A survey using graph distances. *Annals of Applied Statistics*, 12:971–1012.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Levine, R. A. and Casella, G. (2006). Optimizing random scan gibbs samplers. *Journal of Multivariate Analysis*, 97(10):2071–2100.

Lunagómez, S., Olhede, S. C., and Wolfe, P. J. (2021). Modeling network populations via graph distances. *Journal of the American Statistical Association*, 116(536):2023–2040.

Murray, I., Ghahramani, Z., and MacKay, D. J. (2006). Mcmc for doubly-intractable distributions. *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, UAI 2006*, pages 359–366.

Neklyudov, K., Welling, M., Egorov, E., and Vetrov, D. (2020). Involutive mcmc: A unifying framework. In *International Conference on Machine Learning*, pages 7273–7282. PMLR.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM (JACM)*, 21:168–173.

Yang, D., Zhang, D., Zheng, V. W., and Yu, Z. (2015). Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45:129–142.

Algorithm 4: Evaluating edit distance $d_{\mathrm{E},\delta(\cdot)}$

Data: Interaction sequences $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ and $\mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_M)$
Result: $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}')$ (Definition 4)
$C \in \mathbb{R}^{(N+1)\times(M+1)}$;
$C_{11} = 0$;
$C_{(i+1)1} = C_{i1} + \delta(\mathcal{I}_i)$ (for $i = 1, \ldots, N$);
$C_{1(j+1)} = C_{1j} + \delta(\mathcal{I}'_j)$ (for $j = 1, \ldots, M$);
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, m$ **do**

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

  end
end
return $C_{(N+1)(M+1)}$

Algorithm 5: Evaluating edit distance $d_{\mathrm{E},\delta(\cdot)}$ (light memory)

Data: Sequences $\mathcal{S} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)$ and $\mathcal{S}' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_M)$
Result: $d_{\mathrm{E},\delta(\cdot)}(\mathcal{S}, \mathcal{S}')$ (Definition 4)
$Z^{\mathrm{prev}}, Z^{\mathrm{curr}} \in \mathbb{R}^{(M+1)}$;
$Z_1^{\mathrm{prev}} = 0$, $Z_1^{\mathrm{curr}} = 0$;
$Z_{i+1}^{\mathrm{prev}} = Z_i^{\mathrm{prev}} + \delta(\mathcal{I}'_i)$ (for $i = 1, \ldots, M$);
for $i = 1, \ldots, n$ **do**
  $Z_1^{\mathrm{curr}} = Z_1^{\mathrm{curr}} + \delta(\mathcal{I}_i)$;
  for $j = 1, \ldots, m$ **do**

$$Z_{j+1}^{\mathrm{curr}} = \min \begin{cases} Z_{j+1}^{\mathrm{prev}} + \delta(\mathcal{I}_i) \\ Z_j^{\mathrm{curr}} + \delta(\mathcal{I}'_j) \\ Z_j^{\mathrm{prev}} + d_I(\mathcal{I}_i, \mathcal{I}'_j) \end{cases}$$

  end
  $Z^{\mathrm{prev}} = Z^{\mathrm{curr}}$
end
return Z_{M+1}^{curr}

Algorithm 6: Evaluating LCS distance d_{LCS}

Data: Paths $\mathcal{I} = (x_1, \ldots, x_n)$ and $\mathcal{I}' = (y_1, \ldots, y_m)$
Result: $d_{\mathrm{LCS}}(\mathcal{I}, \mathcal{I}')$ (Section 4.3)
$C \in {\mathbb{Z}_+}^{(n+1)\times(m+1)}$;
$C_{11} = 0$;
$C_{(i+1)1} = i$ (for $i = 1, \ldots, n$);
$C_{1(j+1)} = j$ (for $j = 1, \ldots, m$);
$\delta = 0$;
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, m$ **do**

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

  end
end
return $C_{(n+1)(m+1)}$

Algorithm 7: Evaluating LCS distance d_{LCS} (light memory)

Data: Paths $\mathcal{I} = (x_1, \ldots, x_n)$ and $\mathcal{I}' = (y_1, \ldots, y_m)$
Result: $d_{\mathrm{LSP}}(\mathcal{I}, \mathcal{I}')$ (Section 4.3)
$Z^{\mathrm{prev}}, Z^{\mathrm{curr}} \in \mathbb{Z}_+^{(m+1)}$;
$Z^{\mathrm{prev}}_{i+1} = Z^{\mathrm{curr}}_{i+1} = i$ (for $i = 0, \ldots, m$);
for $i = 1, \ldots, n$ **do**
  $Z^{\mathrm{curr}}_1 = Z^{\mathrm{curr}}_1 + 1$;
  for $j = 1, \ldots, m$ **do**

$$Z^{\mathrm{curr}}_{j+1} = \min \begin{cases} Z^{\mathrm{prev}}_{j+1} + 1 \\ Z^{\mathrm{curr}}_{j} + 1 \\ Z^{\mathrm{prev}}_{j} + 2 \cdot \mathbb{1}(x_i \neq y_i) \end{cases}$$

  end
  $Z^{\mathrm{prev}} = Z^{\mathrm{curr}}$;
end
return Z^{curr}_{m+1}

Algorithm 8: Evaluating LSP distance d_{LSP}

Data: Paths $\mathcal{I} = (x_1, \ldots, x_n)$ and $\mathcal{I}' = (y_1, \ldots, y_m)$
Result: $d_{\mathrm{LSP}}(\mathcal{I}, \mathcal{I}')$ (Section 4.3)
$Q \in {\mathbb{Z}_+}^{(n+1)\times(m+1)}$;
$Q_{11} = 0$;
$Q_{(i+1)1} = 0$ (for $i = 1, \ldots, n$);
$Q_{1(j+1)} = 0$ (for $j = 1, \ldots, m$);
$\delta = 0$;
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, m$ **do**
    if $x_i = y_j$ **then**
      $Q_{(i+1)(j+1)} = Q_{ij} + 1$
      $\delta = \max\left(z, Q_{(i+1)(j+1)}\right)$
    else
      $Q_{(i+1)(j+1)} = 0$
    end
  end
end
return $n + m - 2\delta$

Algorithm 9: Evaluating LSP distance d_{LSP} (light memory)

Data: Paths $\mathcal{I} = (x_1, \ldots, x_n)$ and $\mathcal{I}' = (y_1, \ldots, y_m)$
Result: $d_{\mathrm{LSP}}(\mathcal{I}, \mathcal{I}')$ (Section 4.3)
$Z^{\mathrm{prev}}, Z^{\mathrm{curr}} \in \mathbb{Z}_+^{(m+1)}$;
$Z^{\mathrm{prev}}_{i+1} = Z^{\mathrm{curr}}_{i+1} = 0$ (for $i = 0, \ldots, m$);
$\delta = 0$;
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, m$ **do**
    if $x_i = y_j$ **then**
      $Z^{\mathrm{curr}}_{j+1} = Z^{\mathrm{prev}}_{j} + 1$
      $\delta = \max\left(z, Z^{\mathrm{curr}}_{j+1}\right)$
    else
      $Z^{\mathrm{curr}}_{j+1} = 0$
    end
  end
  $Z^{\mathrm{prev}} = Z^{\mathrm{curr}}$
end
return $n + m - 2\delta$

Algorithm 10: SIS posterior component-wise MCMC

Input: observed data $\{\mathcal{S}^{(i)}\}_{i=1}^{n}$
initialise $(\mathcal{S}_0^m, \gamma_0)$
for $i = 1, \ldots, m$ **do**
  `// Update gamma`
  γ_i = `dispersion_update`$(\mathcal{S}_{i-1}^m, \gamma_{i-1})$ `// (Algorithm 11)`
  `// Update mode`
  $\mathcal{S}_i^m$ = `mode_update`$(\mathcal{S}_{i-1}^m, \gamma_i)$ `// (Algorithm 12)`
end
Output: sample $\{(\mathcal{S}_i^m, \gamma_i)\}_{i=1}^{m}$

Algorithm 11: SIS posterior dispersion conditional accept-reject

Input: $(\mathcal{S}_i^m, \gamma_i)$
Output: γ_{i+1}
Function `dispersion_update`$(\mathcal{S}_i^m, \gamma_i)$**:**
  let $(\mathcal{S}^m, \gamma) = (\mathcal{S}_i^m, \gamma_i)$
  sample γ' via $q(\gamma'|\gamma)$ of (35) `// Sample proposal`
  sample $\{\mathcal{S}_i^*\}_{i=1}^{n}$ i.i.d. from SIS$(\mathcal{S}^m, \gamma')$ `// Sample auxiliary data`
  evaluate $\alpha = \alpha(\gamma, \gamma')$ of (36) `// 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 12: SIS posterior mode conditional accept-reject

Input: $(\mathcal{S}_i^m, \gamma_i)$
Output: $\mathcal{S}_{i+1}^m$
function `mode_update(`$\mathcal{S}_i^m$, γ_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 S8.3
sample u via $p(u|\mathcal{S}^m)` // Sample auxiliary variable`
$([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)` // Invoke involution`
sample $\{\mathcal{S}_i^*\}_{i=1}^n$ i.i.d. from SIS$([\mathcal{S}^m]', \gamma)` // Sample auxiliary data`
$\alpha = \alpha(\mathcal{S}^m, [\mathcal{S}^m]')$ of (39), using ratio (44) `// Acceptance probability`
else
`// Path insertion & deletion move`
let u, $f(u, \mathcal{S}^m)$ and $p(u|\mathcal{S}^m)$ be as in Supplement S8.4
sample u via $p(u|\mathcal{S}^m)` // Sample auxiliary variable`
$([\mathcal{S}^m]', u') = f(\mathcal{S}^m, u)` // Invoke involution`
sample $\{\mathcal{S}_i^*\}_{i=1}^n$ i.i.d. from SIS$([\mathcal{S}^m]', \gamma)` // Sample auxiliary data`
$\alpha = \alpha(\mathcal{S}^m, [\mathcal{S}^m]')$ of (39), using ratio (46) `// Acceptance probability`
end

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

return $\mathcal{S}_{i+1}^m$
end

Algorithm 13: SIS model iMCMC sampling

Input: $(\mathcal{S}^m, \gamma)$ (model parameters)
Input: ν_{ed}, ν_{td}, $p(\mathcal{I}|\mathcal{S})$, β (MCMC tuning parameters)
Input: m (sample size), b (burn-in), l (lag)
initialise $\mathcal{S}$;
initialise $i = 1$;
while $i \leq m$ **do**
  sample $z \sim \text{Bernoulli}(\beta)$
  if $z = 1$ **then**
    // Edit allocation move
    let u, $f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Supplement S8.3
    sample u via $p(u|\mathcal{S})$
    $(\mathcal{S}', u') = f(\mathcal{S}, u)$
    evaluate $\alpha = \alpha(\mathcal{S}, \mathcal{S}')$ of (47) using (48)
  else
    // Path insertion & deletion move
    let u, $f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Supplement S8.4
    sample u via $p(u|\mathcal{S})$
    $(\mathcal{S}', u') = f(\mathcal{S}, u)$
    evaluate $\alpha = \alpha(\mathcal{S}, \mathcal{S}')$ of (47) using (49)
  end
  // Accept/reject proposal

$$\mathcal{S} = \begin{cases} \mathcal{S}' & \text{with probability } \alpha \\ \mathcal{S} & \text{with probability } (1 - \alpha) \end{cases}$$

  // Store sample (accounting for lag and burn-in)
  if $(i > b)$ *and* $(i \mod l = 1)$ **then**
    $\mathcal{S}_i \leftarrow \mathcal{S}$
    $i = i + 1$
  end
end
Output: $\{\mathcal{S}_i\}_{i=1}^m$

Algorithm 14: SIM posterior component-wise MCMC

Input: observed data $\{\mathcal{E}^{(i)}\}_{i=1}^n$
initialise $(\mathcal{E}_0^m, \gamma_0)$
for $i = 1, \ldots, m$ **do**
  // Update gamma
  $\gamma_i =$ `dispersion_update`$(\mathcal{E}_{i-1}^m, \gamma_{i-1})$ // (Algorithm 15)
  // Update mode
  $\mathcal{E}_i^m =$ `mode_update`$(\mathcal{E}_{i-1}^m, \gamma_i)$ // (Algorithm 12)
end
Output: sample $\{(\mathcal{E}_i^m, \gamma_i)\}_{i=1}^m$

Algorithm 15: 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 (35) `// Sample proposal`

sample $\{\mathcal{E}_i^*\}_{i=1}^n$ i.i.d. from SIM$(\mathcal{E}^m, \gamma')$ `// Sample auxiliary data`

evaluate $\alpha = \alpha(\gamma, \gamma')$ of (51) `// 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 16: SIM posterior mode conditional accept-reject

Input: $(\mathcal{E}_i^m, \gamma_i)$
Output: $\mathcal{E}_{i+1}^m$
function `mode_update(`$\mathcal{E}_i^m$, γ_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 S8.3
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 (55), using ratio (44) `// Acceptance probability`
else
`// Path insertion & deletion move`
let u, $f(u, \mathcal{S}^m)$ and $p(u|\mathcal{S}^m)$ be as in Supplement S8.4
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 (55), using ratio (46) `// 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 17: SIM model iMCMC sampling

Input: $(\mathcal{E}^m, \gamma)$ (model parameters)
Input: ν_{ed}, ν_{td}, $p(\mathcal{I}|\mathcal{S})$, β (MCMC tuning parameters)
Input: m (sample size), b (burn-in), l (lag)
initialise $\mathcal{E}$;
initialise $i = 1$;
while $i \leq m$ **do**
  obtain $\mathcal{S}$ from $\mathcal{E}$ `// Place paths in arbitrary order`
  sample $z \sim \text{Bernoulli}(\beta)$
  if $z = 1$ **then**
    `// Edit allocation move`
    let u, $f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Supplement S8.3
    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 (59) using (48) `// Acceptance probability`
  else
    `// Path insertion & deletion move`
    let u, $f(u, \mathcal{S})$ and $p(u|\mathcal{S})$ be as in Supplement S8.4
    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 (59) using (49) `// Acceptance probability`
  end
  `// Accept/reject proposal`
  $$\mathcal{E} = \begin{cases} \mathcal{E}' & \text{with probability } \alpha \\ \mathcal{E} & \text{with probability } (1-\alpha) \end{cases}$$
  `// Store sample (accounting for lag and burn-in)`
  if $(i > b)$ *and* $(i \mod l = 1)$ **then**
    $\mathcal{E}_i \leftarrow \mathcal{E}$
    $i = i + 1$
  end
end
Output: $\{\mathcal{E}_i\}_{i=1}^m$