Network Inference from Co-Occurrences

Michael G. Rabbat, Mário A. T. Figueiredo, Robert D. Nowak,

Technical Report ECE-06-2
Department of Electrical and Computer Engineering
University of Wisconsin-Madison

April 16, 2006

Abstract

The study of networked systems is an emerging field, impacting almost every area of engineering and science, including the important domains of communication systems, biology, sociology, and cognitive science. The recovery of network structure from experimental data is a basic and fundamental problem. Unfortunately, experimental data often do not directly reveal network structure due to inherent measurement limitations such as imprecision in timing or other observation mechanisms. This paper considers the following problem. Suppose a number of transmissions are made between a collection of senders and receivers. We observe the subset of network elements (*e.g.*, communication links, genes, actors, neuron colonies) which carry each transmission, but the order in which these elements appear in the transmission paths is not observable. Mathematically, the network structure can be described by a graph whose vertices are the communicating elements, senders and receivers. Each transmission is a directed path through the graph, and without direct knowledge of the order in which vertices are traversed there are generally an exponentially large number of *feasible* graphs which agree with the observed data. Yet, the basic physical principles underlying most networks strongly suggest that all feasible graphs are not equally likely. Specifically, vertices that co-occur frequently are probably closely connected.

To mathematically formalize this intuition, we model paths through the graph as realizations of a random walk on the underlying graph. Each experimental observation is modelled as an independent sample of this random walk (a first-order Markov chain) subjected to a random permutation which accounts for our lack of order information. The problem of recovering network structure then reduces to estimating the parameters of this Markov chain. In particular, we derive an exact *expectation-maximization* (EM) algorithm for finding the *maximum likelihood*

M.G. Rabbat and R.D. Nowak are with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI, 53706. Email: rabbat@cae.wisc.edu, nowak@engr.wisc.edu.

M.A.T. Figueiredo is with *Instituto de Telecomunicações* and the Department of Electrical and Computer Engineering, *Instituto Superior Técnico*, Lisboa, Portugal. Email: mario.figueiredo@lx.it.pt.

(ML) or *maximum a posteriori* (MAP) estimates by treating the random permutations as missing data. For very long paths the E-step may be computationally intractable, so we also propose Monte Carlo versions of the E-step and derive conditions under which the Monte Carlo EM algorithm will converge with high probability. Simulations and experiments with Internet measurements demonstrate the promise of this approach.

1 Network Reconstruction and Co-Occurrence Observations

The study of complex networked systems is an emerging field, impacting nearly every area of engineering and science including the important domains of communication systems, biology, sociology, and cognitive science. Analysis of network structure enables:

Communication networks: A better understanding of routing, transmission patterns, and information flow which can be used to predict routes, to diagnose failures, to identify and trace back anomalous traffic, and to provision new infrastructure [6, 18];

Biological networks: A better understanding of the functional roles played by different genes and proteins in biological systems which can be used to provide insights into human diseases and to identify potential drug targets [11, 17];

Social networks: A better understanding of social interactions and dynamics which can be used to uncover the organizational structure of communities and to predict and analyze the spread of epidemics [16, 21];

Brain networks: A better understanding of how functional regions within the brain are interconnected which can be used to study brain-related illnesses and injuries and to gain new insights into the nature of brain function [1, 20].

Inferring the structure of networks from experimental data is thus a basic and fundamental task, critical to many applications. Unfortunately, measurements which directly reveal network structure are often beyond experimental capabilities or are excessively expensive. In this paper we consider a specific network inference problem: that of learning the structure of a network from indirect observations arising from a subset of simultaneously activated nodes. Mathematically, the underlying network structure is represented as a directed graph and we assume that the nodes activated during one observation form a connected subgraph. Our observations reflect which subset of vertices are activated during the measurement, but not the connectivity. Because the observed vertices simultaneously "occur", we refer to such measurements as *co-occurrence observations*. Co-occurrence observations arise naturally in each of the application areas mentioned above.

Communication networks: Transmissions over communication networks correspond to paths. The so-called *internally-sensed network tomography* problem specifically aims at recovering the network topology given a unordered lists of network elements along transmission paths [18]. It is impossible to observe order information in

practice in this setting. The sensors making observations are distributed over a wide geographic area and paths are constructed on a very short time-scale so extremely precise time synchronization is required to measure order information.

Biological networks: Signal transduction networks describe fundamental cell functions such as growth, metabolism, differentiation, and apoptosis (disintegration). High-throughput measurement techniques such as microarrays have successfully been used to identify the components of different signal transduction pathways. In particular, a single microarray experiment reflects the strength at which genes are expressed or regulated under particular environmental conditions, and these conditions are changed in different experiments. Then, cluster techniques are applied to identify groups of genes which comprise a signaling pathway [23]. The cluster analysis identifies co-occurring genes, but not their order in a pathway. Microarray data only reflects order information at a very coarse level and may be unreliable. Experimental techniques exist which provide more precise order information but they only target a few genes at a time and are both costly and time-consuming. Consequently, developing computational techniques for inferring ordering is an active research area [14].

Social networks: Co-occurrence or transactional data may arise in the context of social networks by considering which academic papers are co-cited by another paper, which web pages are linked to or from another web page, which actors co-appear at an event or are diagnosed with a common disease on the same day. Such measurements are readily available, but do not necessarily reflect the temporal or other natural order in which they appeared. Researchers in this area have considered the problem of reconstructing networks from co-occurrence data and also of using the inferred network to predict potential future co-occurrences [12].

Brain networks: *Functional magnetic resonance imaging* (fMRI) provides a mechanism for measuring activity in the brain with high spatial resolution. By observing which regions of the brain activate while a patient is performing different tasks we can obtain multiple co-occurrence observations. However, although fMRI offers high spatial resolution it comes at the cost of low temporal resolution and so it is not possible to obtain complete order information using such measurements. Magnetoencephalography and electroencephalography measure activity in the brain with higher temporal resolution but only provide coarse spatial resolution, and thus may not allow one to determine precisely which functional regions are active during a given task.

In this article we focus on observations arising from transmissions through the network. Specifically, each co-occurrence observation corresponds to a path[1] through the network. We observe the vertices comprising each path but not the order in which they appear in the path. In certain applications the endpoints (source and destination) of the path may also be observed.

Our goal is to identify which pairs of vertices are directly connected via an edge, thereby learning the structure of the network. A *feasible graph* is one which agrees with the observations; *i.e.*, a graph which contains a directed path through the vertices in each co-occurrence observation. Given a collection of co-occurrence observations a feasible graph is easily constructed by assigning an order – any order, in fact – to the

[1]Throughout this paper a "path" refers to a sequence of vertices $(x_1, x_2, \ldots, x_N)$ such that there is an edge between each adjacent pair of vertices, x_{i-1} and x_i, and no node appears more than once in the sequence.

vertices in each observation, and then inserting directed edges between vertices which are adjacent in the assigned order. Because the number of possible orders of a sequence is exponential in the sequence length, it is evident that there are generally an exponential number of feasible topologies. Without additional assumptions, side information, or prior knowledge there is no reason to prefer any one feasible topology over the others. Yet, the physical principles underlying most complex networks strongly suggest that not all feasible networks are equally likely. This motivates adopting a model or optimality criterion by which to rank the feasible topologies. Still, it may not be easy to compute an optimal solution even with a simple, well-defined criterion. For example, to the best of our knowledge, the only way to find the set of sparsest feasible graphs (*i.e.*, feasible graphs with the fewest edges) is to search the entire solution space.

Previous work on related problems has involved heuristics using frequencies of co-occurrence either to assign an order to each path [18] or to approximate the probability of transitioning from one vertex to another [12]. These approaches are simple to compute in general, but in order to achieve computational tractability they make stringent assumptions and sacrifice robustness. For example, the *frequency method* introduced in [18] is based on a model where paths from a particular source or to a particular destination form a tree. This model coincides with the shortest-path routing policy. When the network provides multiple paths between the same pair of endpoints (*e.g.*, in a load-balancing scenario) the algorithm may fail. The *cGraph* algorithm proposed in [12] inserts weighted edges between every pair of vertices which co-occur in some observation. This approach produces solutions which are typically much denser than desired. Because both of these methods are based on heuristics, the results they produce are not easily interpreted. Also, these heuristics do not readily lend themselves to incorporating side information. A different approach, introduced by Justice and Hero in [10], involves averaging over an ensemble of feasible topologies sampled uniformly from the feasible set. In general there is an enormous number of feasible topologies (exponential in the problem dimensions) exhibiting a wide variety of characteristics, and it is not clear that an average of feasible topologies will be optimal in any sense. These observations have collectively motivated our development of a more general approach to network reconstruction which we simply term *network inference from co-occurrences*, or NICO for short.

1.1 Network Inference from Co-Occurrences

This paper proposes a novel approach to estimating the structure of a network from co-occurrences which 1) generates solutions which are easy to interpret, 2) is robust to modelling assumptions, 3) admits efficient computation, and 4) provides a natural mechanism for incorporating prior knowledge or side information. Our approach is based on a generative model where paths are realizations of a random walk on the underlying graph. A co-occurrence observation is obtained by randomly shuffling the random walk realization, to account for our lack of observed order information. Based on this model, the network reconstruction problem reduces to estimating the parameters governing the random walk. Then we can use these parameter estimates to determine the most likely order for each co-occurrence and reconstruct the network accordingly.

Now, we do not necessarily expect measured paths to be generated according to a

random walk in the real system. Nonetheless, the following interpretation motivates our shuffled random walks as a robust and flexible model. Imagine sitting at a particular vertex in the network and observing a series of transmissions pass by. This vertex is only connected to a handful of other vertices in the network, so regardless of the final destination of each transmission, a transmission arriving at this vertex must pass through one of the neighboring vertices next. Over a period of time, we could record how many arriving transmissions are passed to each neighbor, and then calculate an empirical probability distribution on which neighbor an incoming transmission is passed to. The method proposed here formally develops a framework for estimating local transition probabilities from a collection of co-occurrence observations, without making any additional assumptions about routing behavior or properties of the underlying network structure. Experimental results on simulated topologies indicate that good performance is obtained for a variety of operating conditions. Also, because our method is couched in the theory of probabilistic/statistical inference it is easy to incorporate side information in the form of a prior on the inferred parameters.

The rest of the paper is organized as follows. In Section 2 we introduce notation and formally state the problem setup. Section 3 reviews the standard approach to estimating the parameters of a random walk when fully observed (ordered) samples are available. In Section 4, we derive the EM algorithm for estimating random walk parameters from shuffled observations. The Monte Carlo E-step is described in Section 5 for situations where large observations do not admit exact E-step computation. Section 6 analyzes convergence of the Monte Carlo EM algorithm. Section 7 describes how prior information can easily be incorporated into the inference procedure via a collection of independent Dirichlet priors. Simulation results are presented in Section 8 and the papers is concluded in Section 9.

2 Problem Formulation

Our goal is to reconstruct a network from co-occurrence observations. Formally, we model the network as a simple directed graph on the vertex set $S = \{1, 2, \ldots, |S|\}$. The number of vertices, $|S|$, is known ahead of time, so the network reconstruction amounts to determining the adjacency structure of the graph; *i.e.*, identifying whether or not there is an edge from i to j for every pair of vertices.

A co-occurrence observation, $\mathbf{y} \subseteq S$, is a subset of vertices in the graph which simultaneously "occur" when a particular stimulus is presented to the network. For example, when a transmission is made over a communication network, a subset of routers and switches carry the transmission from the source to the destination. This activated subset corresponds to a co-occurrence observation, with the stimulus being a transmission between that particular source-destination pair. By repeating this procedure T times with different stimuli we obtain the observation data, $\mathcal{Y} = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(T)}\}$, which will be used to infer a network.

A feasible solution contains a path coinciding with each observed co-occurrence. Let $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ denote the elements of a length-N co-occurrence, indexed in ascending order (really, any arbitrary order will do). Because we allow co-occurrences to have different lengths, in what follows we will write N_m for the number of vertices

co-occurring in the mth observation, $\mathbf{y}^{(m)}$. Formally, a directed graph G is a feasible network reconstruction if for each unordered co-occurrence, $\mathbf{y}^{(m)}$, there exists an ordered path $\mathbf{z} = (z_1, z_2, \ldots, z_{N_m})$ and a permutation $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_{N_m})$ such that $z_t = y_{\tau_t}$ for each $t = 1, \ldots, N_m$, and there is an edge from z_{t-1} to z_t in G for $t = 2, \ldots, N_m$.

Notice that a co-occurrence observation does not explicitly contain information about the order of vertices in its corresponding path, but if the order were known then network reconstruction would be trivial. Suppose we observed ordered paths $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}$. Beginning with an empty graph of $|S|$ vertices and no edges, the network reconstruction would be obtained by inserting an edge from $z_{t-1}^{(m)}$ to $z_t^{(m)}$ for each observation. Similarly, given the correct permutation $\boldsymbol{\tau}^{(m)}$ for each co-occurrence observation $\mathbf{y}^{(m)}$ we could obtain ordered observations $\mathbf{z}^{(m)}$ by inverting the permutation, and then use the same procedure.

In practice we do not make ordered observations nor do we have access to the correct permutations. However, we can obtain a feasible reconstruction by associating *any* permutation with each co-occurrence, and then following the procedure described above. There are $N_m!$ ways to permute the elements of $\mathbf{y}^{(m)}$, and so simple combinatorial calculations reveal that there may be as many as $\prod_{m=1}^{T} N_m!$ feasible reconstructions. Clearly, for large N_m and T this is a huge set to search over. Moreover, without making additional assumptions or adopting some additional criteria there is no reason to prefer one feasible reconstruction over another. This motivates making additional modelling assumptions or bringing in additional prior or side information to address the ill-posed nature of this problem.

Physical principles governing the development of many natural and man-made networks suggest that not all feasible networks are equally plausible. Intuitively, if two or more vertices appear collectively in multiple co-occurrences, we expect that their order is probably the same in the corresponding paths. Likewise, we expect that each vertex will generally only have a few neighbors. Based on these intuitions we propose the following probabilistic model. First, we model the unobserved, ordered paths, $\mathbf{z}^{(m)}$, as independent samples of a first-order Markov chain. The Markov chain is parameterized by an initial state distribution $\boldsymbol{\pi} \in [0,1]^{|S|}$ where $\pi_i = P[z_1 = i]$, and a probability transition matrix, $\mathbf{A} \in [0,1]^{|S| \times |S|}$, with $A_{i,j} = P[z_t = j | z_{t-1} = i]$. These parameters must satisfy the constraints

$$\sum_{i=1}^{|S|} \pi_i = 1 \quad \text{and} \quad \sum_{j=1}^{|S|} A_{i,j} = 1 \text{ for each } i = 1, \ldots, |S|. \tag{1}$$

In addition, we assume that the support of the transition matrix is determined by the adjacency structure of the underlying network; *i.e.*, $A_{i,j} > 0$ if and only if the network contains an edge from i to j.

A co-occurrence observation, $\mathbf{y}$, is generated by shuffling the elements of an ordered Markov chain sample, $\mathbf{z} = (z_1, \ldots, z_N)$, according to a permutation, $\boldsymbol{\tau}$, drawn uniformly from Ψ_N, the collection of all permutations of N elements. We assume that $\boldsymbol{\tau}$ is independent of the Markov chain sample, $\mathbf{z}$. Based on this model, we can write the likelihood of a co-occurrence observation $\mathbf{y}$ conditioned on a particular permutation $\boldsymbol{\tau}$

as

$$P[\mathbf{y}|\boldsymbol{\tau}, \mathbf{A}, \boldsymbol{\pi}] \quad = \quad \pi_{y_{\tau_1}} \prod_{t=2}^{N} A_{y_{\tau_{t-1}}, y_{\tau_t}}. \tag{2}$$

By marginalizing over all permutations we obtain

$$P[\mathbf{y}|\mathbf{A}, \boldsymbol{\pi}] \quad = \quad \sum_{\boldsymbol{\tau} \in \Psi_N} P[\mathbf{y}|\boldsymbol{\tau}, \mathbf{A}, \boldsymbol{\pi}] P[\boldsymbol{\tau}] \tag{3}$$

$$= \quad \frac{1}{N!} \sum_{\boldsymbol{\tau} \in \Psi_N} P[\mathbf{y}|\boldsymbol{\tau}, \mathbf{A}, \boldsymbol{\pi}]. \tag{4}$$

Finally, based on the assumption that co-occurrence observations are independent, we have

$$P[\mathcal{Y}|\mathbf{A}, \boldsymbol{\pi}] \quad = \quad \prod_{m=1}^{T} P[\mathbf{y}^{(m)}|\mathbf{A}, \boldsymbol{\pi}]. \tag{5}$$

Taking the logarithm gives

$$\log P[\mathcal{Y}|\mathbf{A}, \boldsymbol{\pi}] \quad = \quad \sum_{m=1}^{T} \left[\log \left(\sum_{\boldsymbol{\tau} \in \Psi_{N_m}} P[\mathbf{y}^{(m)}|\boldsymbol{\tau}^{(m)}, \mathbf{A}, \boldsymbol{\pi}] \right) - \log(N_m!) \right]. \tag{6}$$

Now the network reconstruction problem amounts to estimating the maximum likelihood Markov chain parameters,

$$(\mathbf{A}_{\text{ML}}, \boldsymbol{\pi}_{\text{ML}}) \quad = \quad \arg\max_{\mathbf{A}, \boldsymbol{\pi}} P[\mathcal{Y}|\mathbf{A}, \boldsymbol{\pi}]. \tag{7}$$

Then we can use $(\mathbf{A}_{\text{ML}}, \boldsymbol{\pi}_{\text{ML}})$ to compute the most likely permutation for each co-occurrence observation, and obtain a reconstruction using our procedure for ordered observations described above. Alternatively, a network reconstruction may be obtained by applying a threshold rule to the transition matrix.

Observe from (2) that each conditional likelihood involves a product of transition matrix terms, and recall the constraints $\sum_{j=1}^{|S|} A_{i,j} = 1$. We can think of node i being assigned one unit of mass, and this unit is distributed over each of its neighbors in the graph. If vertex i has more neighbors then this unit mass is being spread over a larger number of terms so the $A_{i,j}$ will be smaller. Consequently, the likelihood of co-occurrences containing i is smaller. This reasoning provides one explanation for why our model encourages sparse reconstructions.

Of course, we could try to solve the optimization (7) directly, but (6) is generally a complicated, non-convex function and so direct optimization is not a simple task. Below we derive an EM algorithm for solving this optimization by treating the permutations $\{\boldsymbol{\tau}^{(m)}\}$ shuffling each path as missing data. Before deriving the EM algorithm we review the standard approach to estimating Markov chain parameters from ordered observations.

3 Estimating Markov Chain Parameters from Direct Observations

It is convenient to introduce another representation for the Markov chain samples, $\mathbf{z} = (z_1, \ldots, z_N)$; specifically, instead of $z_t \in S$, we use the equivalent binary representation $\mathbf{w}_t = (w_{t,1}, ..., w_{t,|S|}) \in \{0,1\}^{|S|}$ with $(w_{t,i} = 1) \Leftrightarrow (z_t = i)$. One and only one entry of each vector $\mathbf{w}_t$ is equal to 1. With this notation, we can write

$$\begin{aligned} P[z_1, ..., z_N|\mathbf{A}, \boldsymbol{\pi}] &= P[\mathbf{w}_1, ..., \mathbf{w}_N|\mathbf{A}, \boldsymbol{\pi}] \quad (8) \\ &= \prod_{i=1}^{|S|} (\pi_i)^{w_{1,i}} \prod_{t=2}^{N} \prod_{i=1}^{|S|} \prod_{j=1}^{|S|} (A_{i,j})^{w_{t-1,i}\, w_{t,j}}, \quad (9) \end{aligned}$$

where, by convention $0^0 = 1$ (justifiable by continuity, since $\lim_{a\to 0} a^a = 1$). Thus,

$$\log P[\mathbf{w}_1, ..., \mathbf{w}_N|\mathbf{A}, \boldsymbol{\pi}] = \sum_{i=1}^{|S|} w_{1,i} \log \pi_i + \sum_{t=2}^{N} \sum_{i=1}^{|S|} \sum_{j=1}^{|S|} w_{t-1,i}\, w_{t,j}\, \log A_{i,j}. \quad (10)$$

Now, suppose that instead of one sequence $\mathbf{w} = (\mathbf{w}_1, ..., \mathbf{w}_N)$, we have a set $\mathcal{W}$ with a total of T sequences which are assumed to be independent realizations of this Markov process. Each sequence may have a different length, so we write $\mathcal{W} = \{\mathbf{w}^{(1)}, ..., \mathbf{w}^{(T)}\}$, where $\mathbf{w}^{(m)} = (\mathbf{w}_1^{(m)}, ..., \mathbf{w}_{N_m}^{(m)})$, for $m = 1, ..., T$. The log-likelihood for the set of sequences (due to the independence assumption) is simply

$$\begin{aligned} \log P[\mathcal{W}|\mathbf{A}, \boldsymbol{\pi}] &= \sum_{m=1}^{T} \log P[\mathbf{w}^{(m)}|\mathbf{A}, \boldsymbol{\pi}] \\ &= \sum_{m=1}^{T} \sum_{i=1}^{|S|} w_{1,i}^{(m)} \log \pi_i + \sum_{m=1}^{T} \sum_{t=2}^{N_m} \sum_{i=1}^{|S|} \sum_{j=1}^{|S|} w_{t-1,i}^{(m)}\, w_{t,j}^{(m)} \log A_{i,j}. \\ &= \sum_{i=1}^{|S|} \log \pi_i \sum_{m=1}^{T} w_{1,i}^{(m)} + \sum_{i=1}^{|S|} \sum_{j=1}^{|S|} \log A_{i,j} \sum_{m=1}^{T} \sum_{t=2}^{N_m} w_{t-1,i}^{(m)}\, w_{t,j}^{(m)}. \quad (11) \end{aligned}$$

Maximum likelihood estimates of $\boldsymbol{\pi}$ and $\mathbf{A}$ are obtained my maximizing $\log P[\mathcal{W}|\mathbf{A}, \boldsymbol{\pi}]$ under the constraints in (1). Since $\log P[\mathcal{W}|\mathbf{A}, \boldsymbol{\pi}]$ is a concave function of $\mathbf{A}$ and $\boldsymbol{\pi}$ (it is a sum of logarithms of these variables), concave constrained optimization using Lagrange multipliers leads to the following well-known estimates:

$$\begin{aligned} \widehat{A}_{i,j} &= \frac{\sum_{m=1}^{T} \sum_{t=2}^{N_m} w_{t-1,i}^{(m)}\, w_{t,j}^{(m)}}{\sum_{j=1}^{|S|} \sum_{m=1}^{T} \sum_{t=2}^{N_m} w_{t-1,i}^{(m)}\, w_{t,j}^{(m)}} \\ \widehat{\pi}_i &= \frac{1}{T} \sum_{m=1}^{T} w_{1,i}^{(m)}. \quad (12) \end{aligned}$$

4 Estimating Markov Chain Parameters from Shuffled Observations Via the EM Algorithm

We are now interested in the case where we we have co-occurrences, not ordered samples. Rather than using $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_N)$ to denote the shuffling permutation, we introduce a more convenient representation; each shuffling is represented by a *permutation matrix, i.e.*, a matrix with one and only one 1 in each row and each column. Let the shuffling matrix for sequence m be denoted as $\mathbf{r}^{(m)}$ so that $(r^{(m)}_{t,t'} = 1) \Leftrightarrow (\mathbf{x}^{(m)}_{t'} = \mathbf{w}^{(m)}_{t})$. Given both $\mathbf{r}^{(m)}$ and $\mathbf{x}^{(m)}$, we could recover the unshuffled sequence $\mathbf{w}^{(m)}$ by applying

$$w^{(m)}_{t,i} = \prod_{t'=1}^{N_m} \left(x^{(m)}_{t',i} \right)^{r^{(m)}_{t,t'}}, \tag{13}$$

adopting the convention $0^0 = 1$. For example, with $T = 2$, $N_1 = 5$, $N_2 = 4$,

$$\mathbf{r}^{(1)} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{r}^{(2)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

we have that $\mathbf{w}^{(1)} = (\mathbf{w}^{(1)}_1, \mathbf{w}^{(1)}_2, \mathbf{w}^{(1)}_3, \mathbf{w}^{(1)}_4, \mathbf{w}^{(1)}_5) = (\mathbf{x}^{(1)}_3, \mathbf{x}^{(1)}_1, \mathbf{x}^{(1)}_5, \mathbf{x}^{(1)}_2, \mathbf{x}^{(1)}_4)$, and $\mathbf{w}^{(2)} = (\mathbf{w}^{(2)}_1, \mathbf{w}^{(2)}_2, \mathbf{w}^{(2)}_3, \mathbf{w}^{(2)}_4) = (\mathbf{x}^{(2)}_2, \mathbf{x}^{(2)}_3, \mathbf{x}^{(2)}_1, \mathbf{x}^{(2)}_4)$, that is, the position of the unique 1 in row t indicates which of the shuffled samples was produced at time t.

Denoting by $\mathcal{R} = \{\mathbf{r}^{(1)}, ..., \mathbf{r}^{(T)}\}$ the collection of sorting matrices corresponding to $\mathcal{X} = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(T)}\}$, we can write the complete log-likelihood as follows. Start by observing that

$$\log P[\mathcal{X}, \mathcal{R} | \mathbf{A}, \boldsymbol{\pi}] = \log P[\mathcal{X} | \mathcal{R}, \mathbf{A}, \boldsymbol{\pi}] + \log p[\mathcal{R}],$$

and that $p[\mathcal{R}]$ is just a constant (assuming uniform distribution over the set of all possible permutations), we have

$$\begin{aligned} \log P[\mathcal{X}, \mathcal{R} | \mathbf{A}, \boldsymbol{\pi}] &\propto \log P[\mathcal{X} | \mathcal{R}, \mathbf{A}, \boldsymbol{\pi}] && (14) \\ &= \sum_{m=1}^{T} \log P[\mathbf{x}^{(m)} | \mathbf{r}^{(m)}, \mathbf{A}, \boldsymbol{\pi}] && (15) \\ &= \sum_{m=1}^{T} \sum_{t=2}^{N_m} \sum_{t'=1}^{N_m} \sum_{t''=1}^{N_m} \sum_{i,j=1}^{|S|} r^{(m)}_{t,t'} \, r^{(m)}_{t-1,t''} \, x^{(m)}_{t'',i} \, x^{(m)}_{t',j} \, \log A_{i,j} \\ &\quad + \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \sum_{i=1}^{|S|} r^{(m)}_{1,t'} \, x^{(m)}_{t',i} \, \log \pi_i. && (16) \end{aligned}$$

Next we treat $\mathcal{R}$ as missing data and address the problem of estimating $\mathbf{A}$ and $\boldsymbol{\pi}$ via the EM algorithm. The EM algorithm proceeds by computing the expected value

of the complete log-likelihood $\log P[\mathcal{X}, \mathcal{R}|\mathbf{A}, \boldsymbol{\pi}]$ with respect to the missing data, conditioned on the observations and on the current estimate of the model parameters, $\mathbf{A}^k$ and $\boldsymbol{\pi}^k$,

$$Q\left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k\right) = E\left[\log P[\mathcal{X}, \mathcal{R}|\mathbf{A}, \boldsymbol{\pi}] \,\middle|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k\right]. \tag{17}$$

The model parameter estimates are then updated according to

$$\left(\mathbf{A}^{k+1}, \boldsymbol{\pi}^{k+1}\right) = \arg\max_{\mathbf{A}, \boldsymbol{\pi}} Q\left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k\right), \tag{18}$$

and the process in repeated cyclically until a convergence criterion is met. Equation (17) is the E-step, and (18) the M-step.

4.1 The E-step

Rearranging the order of summation in (16), we can write

$$\begin{aligned}\log P[\mathcal{X}, \mathcal{R}|\mathbf{A}, \boldsymbol{\pi}] \quad \propto \quad & \sum_{m=1}^{T} \sum_{i,j=1}^{|S|} \sum_{t',t''=1}^{N_m} \sum_{t=2}^{N_m} r_{t,t'}^{(m)} r_{t-1,t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)} \log A_{i,j} \\ & \sum_{m=1}^{T} \sum_{i=1}^{|S|} \sum_{t'=1}^{N_m} r_{1,t'}^{(m)} x_{t',i}^{(m)} \log \pi_i.\end{aligned}$$

A key observation which facilitates the derivation of the E-step is that the complete log-likelihood is linear with respect to simple functions of the missing variables:

- the first row of each matrix $\mathbf{r}^{(m)}$, that is, $r_{1,t'}^{(m)}$, for $m = 1, \ldots, T$ and $t' = 1, \ldots, N_m$;
- sums of transition indicators: $\alpha_{t',t''}^{(m)} \equiv \sum_{t=2}^{N_m} r_{t,t'}^{(m)} \; r_{t-1,t''}^{(m)}$, for $m = 1, \ldots, T$, and $t', t'' = 1, \ldots, N_m$.

Since the conditional expectation of a linear function of a random variable is simply that linear function computed at the expected value of the random variable, in the E-step we just have to compute the conditional expectations of $r_{t,t'}^{(m)}$ and $\alpha_{t',t''}^{(m)}$ and plug them into the complete log-likelihood function. Denote by

$$\bar{r}_{1,t'}^{(m)} \equiv E\left[r_{1,t'}^{(m)} \,\middle|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k\right] = P\left[r_{1,t'}^{(m)} = 1 \,\middle|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k\right] \tag{19}$$

$$\bar{\alpha}_{t',t''}^{(m)} \equiv E\left[\alpha_{t',t''}^{(m)} \,\middle|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k\right] = P\left[\alpha_{t',t''}^{(m)} = 1 \,\middle|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k\right], \tag{20}$$

where we have used the fact that these variables are all binary[2], thus their expected values coincide with the probability of being equal to one. The function $Q\left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k\right)$ is obtained by plugging $\bar{r}_{1,t'}^{(m)}$ in the place of $r_{1,t'}^{(m)}$, and $\bar{\alpha}_{t',t''}^{(m)}$ in the place of $\sum_{t=2}^{N_m} r_{t,t'}^{(m)} \; r_{t-1,t''}^{(m)}$ in the complete log-likelihood (16).

[2]Note that since $\mathbf{r}^{(m)}$ is a permutation matrix, $\alpha_{t',t''}^{(m)}$ is also a binary variable.

To conclude the derivation of the E-step, we obtain exact expressions for these conditional expectations. Let us start with $\bar{r}_{1,t'}^{(m)}$, which is given by (19). From the mutual independence among the several observed sequences,

$$\bar{r}_{1,t'}^{(m)} = P\left[r_{1,t'}^{(m)} = 1 \,\middle|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k\right] = P\left[r_{1,t'}^{(m)} = 1 \,\middle|\, \mathbf{x}^{(m)}, \mathbf{A}^k, \boldsymbol{\pi}^k\right].$$

Next, invoking Bayes law, we have that

$$\bar{r}_{1,t'}^{(m)} = \frac{P\big[\mathbf{x}^{(m)} \big| r_{1,t'}^{(m)} = 1, \mathbf{A}^k, \boldsymbol{\pi}^k\big]\; P\big[r_{1,t'}^{(m)} = 1\big]}{P\big[\mathbf{x}^{(m)} \big| \mathbf{A}^k, \boldsymbol{\pi}^k\big]}.$$

Then, under the assumption that all permutations are equally likely, marginalizing over permutations gives

$$\begin{aligned}
\bar{r}_{1,t'}^{(m)} &= \frac{\left(\frac{1}{(N_m-1)!}\sum_{\mathbf{r}\in\Psi_{N_m}:r_{1,t'}=1} P\big[\mathbf{x}^{(m)}|\mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big]\right)\left(\frac{(N_m-1)!}{N_m!}\right)}{\frac{1}{N_m!}\sum_{\mathbf{r}\in\Psi_{N_m}} P\big[\mathbf{x}^{(m)}\big|\mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big]} \\
&= \frac{\sum_{\mathbf{r}\in\Psi_{N_m}:r_{1,t'}=1} P\big[\mathbf{x}^{(m)}\big|\mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big]}{\sum_{\mathbf{r}\in\Psi_{N_m}} P\big[\mathbf{x}^{(m)}\big|\mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big]}.
\end{aligned}$$

The terms $P\big[\mathbf{x}^{(m)}\big|\mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big]$ are easily computed as

$$\begin{aligned}
P\big[\mathbf{x}^{(m)}\big|\mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big] &= P\big[\mathbf{y}^{(m)}\big|\boldsymbol{\tau}, \mathbf{A}^k, \boldsymbol{\pi}^k\big] \\
&= \pi^k_{y^{(m)}_{\tau_1}} \prod_{t=2}^{N_m} A^k_{y^{(m)}_{\tau_{t-1}}, y^{(m)}_{\tau_t}}.
\end{aligned}$$

Defining summary statistics

$$\gamma_{t'}^{(m)} \equiv \sum_{\mathbf{r}\in\Psi_{N_m}:r_{1,t'}=1} P\big[\mathbf{x}^{(m)}\big|\mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big], \tag{21}$$

for each $m = 1, \ldots, T$ and $t' = 1, \ldots, N_m$, we have

$$\bar{r}_{1,t'}^{(m)} = \frac{\gamma_{t'}^{(m)}}{\sum_{t'=1}^{N_m}\gamma_{t'}^{(m)}}.$$

We compute $\bar{\alpha}^{(m)}_{t',t''}$ in a similar fashion and obtain

$$\begin{aligned}
\bar{\alpha}^{(m)}_{t',t''} &= E\left[\sum_{t=2}^{N_m} r^{(m)}_{t,t'} r^{(m)}_{t-1,t''} \middle| \mathbf{x}^{(m)}, \mathbf{A}^k, \boldsymbol{\pi}^k\right] \\
&= \sum_{t=2}^{N_m} E\big[r^{(m)}_{t,t'} r^{(m)}_{t-1,t''} \big| \mathbf{x}^{(m)}, \mathbf{A}^k, \boldsymbol{\pi}^k\big] \\
&= \sum_{t=2}^{N_m} P\big[r^{(m)}_{t,t'} r^{(m)}_{t-1,t''} = 1 \big| \mathbf{x}^{(m)}, \mathbf{A}^k, \boldsymbol{\pi}^k\big] \\
&= \sum_{t=2}^{N_m} \frac{P\big[\mathbf{x}^{(m)} \big| r^{(m)}_{t,t'} r^{(m)}_{t-1,t''} = 1, \mathbf{A}^k, \boldsymbol{\pi}^k\big]\ P\big[r^{(m)}_{t,t'} r^{(m)}_{t-1,t''} = 1\big]}{P\big[\mathbf{x}^{(m)} \big| \mathbf{A}^k, \boldsymbol{\pi}^k\big]} \\
&= \sum_{t=2}^{N_m} \frac{\Big(\frac{1}{(N_m-2)!} \sum_{\mathbf{r}\in\Psi_{N_m}: r_{t,t'} r_{t-1,t''}=1} P\big[\mathbf{x}^{(m)} \big| \mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big]\Big) \Big(\frac{(N_m-2)!}{N_m!}\Big)}{\frac{1}{N_m!} \sum_{\mathbf{r}\in\Psi_{N_m}} P\big[\mathbf{x}^{(m)} \big| \mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big]} \\
&= \frac{\sum_{\mathbf{r}\in\Psi_{N_m}} P[\mathbf{x}^{(m)}|\mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k] \sum_{t=2}^{N_m} r_{t,t'} r_{t-1,t''}}{\sum_{\mathbf{r}\in\Psi_{N_m}} P[\mathbf{x}^{(m)}|\mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k]}.
\end{aligned}$$

Defining statistics

$$\gamma^{(m)}_{t',t''} \equiv \sum_{\mathbf{r}\in\Psi_{N_m}} P[\mathbf{x}^{(m)}|\mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k] \sum_{t=2}^{N_m} r_{t,t'} r_{t-1,t''}, \tag{22}$$

we have

$$\bar{\alpha}^{(m)}_{t',t''} = \frac{\gamma^{(m)}_{t',t''}}{\sum_{t'=1}^{N_m} \gamma^{(m)}_{t'}}. \tag{23}$$

Notice that for the mth observation, storing all the statistics, $\{\bar{r}^{(m)}_{1,t'}\}$ and $\{\bar{\alpha}^{(m)}_{t,t',t''}\}$, requires N_m^2 memory units; there are $2\binom{N_m}{2} = N_m^2 - N_m$ transition statistics, $\bar{\alpha}^{(m)}_{t',t''}$, and N_m initial state statistics, $\bar{r}^{(m)}_{1,t'}$. These quantities can be computed via the summary statistics $\{\gamma^{(m)}_{t'}\}$ and $\{\gamma^{(m)}_{t',t''}\}$ using the same memory needed to store $\{\bar{r}^{(m)}_{1,t'}\}$ and $\{\bar{\alpha}_{t,t',t''}\}$, in $O\big(N_m!\big)$ operations; *i.e.*, the number of operations required to enumerate all permutations of the co-occurring vertices in this observation. For large observations (large N_m) this can be a rather hefty load, and in Section 5 we suggest methods for computing approximations to $\bar{r}_{1,t'}$ and $\bar{\alpha}_{t',t''}$.

4.2 The M-step

As just shown in Section 4.1, the function $Q\left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k\right)$ is

$$\begin{aligned} Q\left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k\right) &= \sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \sum_{i,j=1}^{|S|} \bar{\alpha}_{t',t''}^{(m)}\, x_{t'',i}^{(m)}\, x_{t',j}^{(m)}\, \log A_{i,j} \\ &\quad + \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \sum_{i=1}^{|S|} \bar{r}_{1,t'}^{(m)}\, x_{t',i}^{(m)}\, \log \pi_i. \end{aligned} \tag{24}$$

Maximization under the constraints in (1) leads to following simple update equations:

- **Transition matrix:**

$$A_{i,j}^{k+1} = \frac{\sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)}}{\sum_{j=1}^{|S|} \sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)}}. \tag{25}$$

- **Initial probabilities:**

$$\pi_i^{k+1} = \frac{\sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)}\, x_{t',i}^{(m)}}{\sum_{i=1}^{|S|} \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)}\, x_{t',i}^{(m)}}. \tag{26}$$

4.3 Known Endpoints

We have just derived the EM algorithm for the general case where $\mathcal{X}$ is a collection of fully shuffled sequences from a Markov chain. A special case arising in some applications described in the introduction is where (one or both of) the endpoints of each path are known and only the internal nodes are unordered. This is the case in the context of communication networks (*i.e.*, internally-sensed network tomography), since the source and destination in each path are known but the connectivity within the network is unknown. For the purposes of estimating biological networks (signal transduction pathways), a physical stimulus (*e.g.*, hypotonic shock) causes a sequence of protein interactions, resulting in another observable physical response (*e.g.*, a change in cell wall structure); in this setting, the stimulus and response act as fixed endpoints and our goal is to infer the order of the sequence of protein interactions.

Our EM algorithm can easily be modified to handle known endpoints. Observe that knowledge of the endpoints of each path imposes the constraints

$$r_{1,1}^{(m)} = 1 \quad \text{and} \quad r_{N_m,N_m}^{(m)} = 1. \tag{27}$$

Under the first constraint, estimates of the initial state probabilities are simply given by

$$\widehat{\pi}_i = \frac{1}{T}\sum_{m=1}^{T} x_{1,i}^{(m)}. \tag{28}$$

Thus, the EM algorithm only needs to be used to estimate the transition matrix entries. Let

$$\widetilde{\Psi}_N = \{r \in \Psi_N \ : \ r_{1,1} = 1,\, r_{N,N} = 1\}, \tag{29}$$

denote the collection of permutations of N elements with fixed endpoints. The M-step (update for $\mathbf{A}^{k+1}$) remains exactly the same. Similar to before, the E-step can be computed using summary statistics

$$\widetilde{\gamma}^{(m)} = \sum_{r\in\widetilde{\Psi}_{N_m}} P[\mathbf{x}^{(m)}|\mathbf{r}, \widehat{\mathbf{A}}, \widehat{\boldsymbol{\pi}}] \tag{30}$$

$$\widetilde{\gamma}_{t',t''}^{(m)} = \sum_{r\in\widetilde{\Psi}_{N_m}} P[\mathbf{x}^{(m)}|\mathbf{r}, \widehat{\mathbf{A}}, \widehat{\boldsymbol{\pi}}] \sum_{t=2}^{N_m} r_{t,t'} r_{t-1,t''}, \tag{31}$$

for $t', t'' = 1, \ldots, N_m$, and setting $\bar{\alpha}_{t',t''}^{(m)} = \widetilde{\gamma}_{t',t''}^{(m)}/\widetilde{\gamma}$.

5 Monte Carlo E-Step by Importance Sampling

Implementing the exact E-step is straightforward. However, for long sequences, the combinatorial nature of (21) and (22), that is, the need to sum over all permutations of the sequence, may render exact computation impractical. In this section, we consider sampling-based approximate versions of the E-step, which avoid the combinatorial nature of its exact version. To lighten the notation in this section, we focus on a particular length-N path $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and drop the superscript (m); due to the independence of the paths, there is no loss of generality. We also drop the superscripts from $(\mathbf{A}^k,\, \boldsymbol{\pi}^k)$ and use simply $(\mathbf{A}, \boldsymbol{\pi})$ to denote the current Markov chain parameter estimates in the EM algorithm.

The E-step (see (19) and (20)) consists of computing the conditional expectations

$$\bar{r}_{1,t'} \ = \ E\big[r_{1,t'}\big|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}\big] = \sum_{\mathbf{r}\in\Psi_N} r_{1,t'}\ P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}] \tag{32}$$

$$\bar{\alpha}_{t',t''} \ = \ E\big[\alpha_{t',t''}\big|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}\big] = \sum_{\mathbf{r}\in\Psi_N} \sum_{t=2}^{N_m} r_{t,t'}\ r_{t-1,t''}\ P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]. \tag{33}$$

A naïve Monte Carlo approximation to these sums would be based on random permutations, sampled from the uniform distribution over Ψ_N (the collections of all permutations of N elements). However, the reason one may have to resort to approximation techniques in the first place is that Ψ_N is large; thus, typically only a small fraction of these random permutations will have non-negligible posterior probability,

$P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$, and so a very large number of uniform samples is needed to obtain a good approximation to $\bar{r}_{1,t'}$ and $\bar{\alpha}_{t',t''}$.

Ideally, we would sample permutations directly from the posterior distribution $P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$; however, sampling from this distribution would require determining its value for all $N!$ permutations in Ψ_N. Instead, we employ *importance sampling* (IS): we sample L permutations, $\mathbf{r}^1, \ldots, \mathbf{r}^L$, from a distribution $R[\mathbf{r}]$, from which it is easier to sample than $P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$, and then apply a corrective re-weighting to obtain approximations to $\bar{r}_{1,t'}$ and $\bar{\alpha}_{t',t''}$; see, for example, [19] or [13] for an introduction to IS. Note that we are now using the superscript on $\mathbf{r}$ to index the sample number, not to identify the path. The importance sampling estimates are given by

$$\bar{r}_{1,t'} \simeq \frac{\sum_{i=1}^{L} z_i \, r^i_{1,t'}}{\sum_{i=1}^{L} z_i}, \tag{34}$$

$$\bar{\alpha}_{t,t',t''} \simeq \frac{\sum_{i=1}^{L} z_i \; \sum_{t=2}^{N_m} r^i_{t,t'} \; r^i_{t-1,t''}}{\sum_{i=1}^{L} z_i}, \tag{35}$$

where z_i is the correction factor (or weight) for sample $\mathbf{r}^i$, given by

$$z_i = \frac{P[\mathbf{r}^i|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]}{R[\mathbf{r}^i]}, \tag{36}$$

the ratio between the desired distribution and the sampling distribution employed.

A relevant observation is that the target and sampling distributions only need to be known up to normalizing factors. Given $R'[\mathbf{r}] = Z_R \, R[\mathbf{r}]$ and $P'[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}] = Z_P \, P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$, for constants Z_R and Z_P, we can use

$$z'_i = \frac{P'[\mathbf{r}^i|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]}{R'[\mathbf{r}^i]} = \frac{Z_P}{Z_R} \, z_i, \tag{37}$$

instead of z_i in (34) and (35); these sums will remain unchanged since the factor Z_P/Z_R will appear both in the numerator and denominator, thus cancelling out.

The remainder of this section contains the description of two IS schemes, including the derivation of closed form expressions for both the sampling distribution, R, and weights, z_i, associated with each approach. In the first approach, permutations are generated sequentially by sampling the "next" element of each sequence according to our current estimate of the transition matrix, $\mathbf{A}$. The second sampler takes a more hierarchical approach, by first sampling likely transitions and then "gluing" these transitions together to form a permutation. We conclude the section by describing other sampling variants and presenting an empirical comparison of the various techniques discussed.

5.1 Causal Sampling

It is more convenient here to use the $\mathbf{y}$ and $\boldsymbol{\tau}$ representations for the shuffled paths and permutations (see Section 2). Also, we will enforce the assumption that the observed sequence $\mathbf{y} = \{y_1, \ldots, y_N\}$ is indeed a path through a network, so it contains no repeated elements.

Let us define a sequence of binary flags, $\mathbf{f} = \{f_1, f_2, \ldots, f_{|S|}\}$, with $f_i \in \{0, 1\}$. Let (for now) these variables indicate the presence/absence of state i in $\mathbf{y}$, that is,

$$f_i = \mathbb{I}_{\{i \in \mathbf{y}\}} = \begin{cases} 1 & \Leftarrow \quad i \in \mathbf{y} \\ 0 & \Leftarrow \quad i \notin \mathbf{y} \end{cases} \quad \text{for } i = 1, 2, \ldots, |S|, \tag{38}$$

where $\mathbb{I}_{\{\cdot\}}$ denotes the indicator function. Given some probability distribution $\mathbf{p} = \{p_1, p_2, \ldots, p_{|S|}\}$ on the set of states, S, denote by $\mathbf{p} \cdot \mathbf{f}$ this distribution restricted to those elements of S that have corresponding flag f_i set to 1, that is,

$$(\mathbf{p} \cdot \mathbf{f})_i = \frac{p_i f_i}{\sum_{j=1}^{|S|} p_j f_j} = \frac{p_i f_i}{\mathbf{p}^T \mathbf{f}}, \quad \text{for } i = 1, 2, \ldots, |S|. \tag{39}$$

The sequential sampling scheme, in the most general case where the endpoints are not known ahead of time, is defined as follows:

Step 1: Let $\mathbf{f}$ be initialized as in (38).

Obtain one sample from S according to the distribution $\boldsymbol{\pi} \cdot \mathbf{f}$. Let the obtained sample be denoted s; of course, one and only one element of $\mathbf{y}$ is equal to s.

Locate the position t of s in $\mathbf{y}$; that is, find t such that $y_t = s$.

Set $\tau_1 = t$.

Set $f_s = 0$ to prevent y_t from being sampled again.

Set $i = 2$.

Step 2: Let $\mathbf{p} = \{A_{s,1}, \ldots, A_{s,|S|}\}$ be the sth row of the transition matrix.

Obtain a new sample s' from S according to the distribution $\mathbf{p} \cdot \mathbf{f}$; again, one and only one element from $\mathbf{y}$ is equal to s'.

Locate the position t of s' in $\mathbf{y}$; i.e., find t such that $y_t = s'$.

Set $\tau_i = t$.

Set $f_s = 0$ to prevent y_t from being sampled again.

Step 3: If $i < N$, then set $s \leftarrow s'$, $i \leftarrow i + 1$, and go back to Step 2.

5.1.1 Sampling Distribution

Before deriving the form of the distribution R, let us begin by writing our target distribution $P[\tau|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]$ explicitly. Using Bayes law, we have

$$P[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}] = \frac{P[\mathbf{y}|\boldsymbol{\tau}, \mathbf{A}, \boldsymbol{\pi}] P[\boldsymbol{\tau}]}{P[\mathbf{y}|\mathbf{A}, \boldsymbol{\pi}]}, \tag{40}$$

since $\boldsymbol{\tau}$ does not depend *a priori* on $\mathbf{A}$ or $\boldsymbol{\pi}$. Based on our assumption that all permutations are equiprobable we have $P[\boldsymbol{\tau}] = \mathbb{I}_{\{\boldsymbol{\tau} \in \Psi_N\}}/N!$ (any sequence $\boldsymbol{\tau}$ which is

not one of the $N!$ permutations of $\{1, ..., N\}$ has probability zero). Noticing that the denominator in (40) is just a normalizing constant independent of $\boldsymbol{\tau}$, we have

$$P[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}] \propto \mathbb{I}_{\{\boldsymbol{\tau} \in \Psi_N\}} \, P[\mathbf{y}|\boldsymbol{\tau}, \mathbf{A}, \boldsymbol{\pi}] = \mathbb{I}_{\{\boldsymbol{\tau} \in \Psi_N\}} \left(\pi_{y_{\tau_1}} \prod_{t=2}^{N} A_{y_{\tau_{t-1}}, y_{\tau_t}} \right), \quad (41)$$

since, given $\boldsymbol{\tau}$, the shuffled sequence $\mathbf{y}$ can be unshuffled, and its probability under the Markov model specified by $\mathbf{A}$ and $\boldsymbol{\pi}$ can be computed.

Next we derive the distribution $R[\boldsymbol{\tau}]$ corresponding to the sequential sampling procedure just described. Of course, this distribution also depends on $\mathbf{A}$, $\boldsymbol{\pi}$, and $\mathbf{y}$, so we should write $R[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]$; however, for the sake of simplicity, we will omit this explicit dependence from the notation. The sequential nature of the sampling scheme suggests a factorization of the form

$$R[\boldsymbol{\tau}] = R[\tau_1] \, R[\tau_2|\tau_1] \, R[\tau_3|\tau_2, \tau_1] \, \cdots \, R[\tau_N|\tau_{N-1}, \ldots, \tau_1]. \quad (42)$$

Consider Step 1 of the sampling scheme; clearly, for $s = 1, \ldots, N$,

$$R[\tau_1 = s] = \frac{\pi_{y_s}}{\sum_{t=1}^{N} \pi_{y_t}}.$$

Notice that the sum in the denominator would be incorrect if we were to allow repetitions within $\mathbf{y}$; under the assumption that measurements correspond to paths through the network this normalization is correct. In more compact notation, we have

$$R[\tau_1] \propto \pi_{y_{\tau_1}}, \quad (43)$$

since the sum in the denominator does not depend on τ_1.

Next, consider the $R[\tau_2|\tau_1]$ term. From Step 2 of the sampling procedure, we have

$$R[\tau_2|\tau_1] = A_{y_{\tau_1}, y_{\tau_2}} \, \phi_2(\tau_1) \, \mathbb{I}_{\{\tau_2 \neq \tau_1\}}, \quad (44)$$

where

$$\phi_2(\tau_1) = \frac{1}{\sum_{t \neq \tau_1} A_{y_{\tau_1}, y_t}}.$$

For the general i-th step of the sampling algorithm we have

$$R[\tau_i|\tau_{i-1}, \ldots, \tau_1] = A_{y_{\tau_{i-1}}, y_{\tau_i}} \, \phi_i(\tau_{i-1}, \ldots, \tau_1) \, \mathbb{I}_{\{\tau_i \notin \{\tau_{i-1}, \ldots, \tau_1\}\}}, \quad (45)$$

with

$$\phi_i(\tau_{i-1}, \ldots, \tau_1) = \frac{1}{\sum_{t \notin \{\tau_{i-1}, \ldots, \tau_1\}} A_{y_{\tau_{i-1}}, y_t}}.$$

Inserting (43), (44), and (45) into (42), we finally have

$$R[\boldsymbol{\tau}] \propto \left(\pi_{y_{\tau_1}} \prod_{i=2}^{N} A_{y_{\tau_{i-1}}, y_{\tau_i}} \right) \left(\prod_{i=2}^{N} \phi_i(\tau_{i-1}, \ldots, \tau_1) \right) \left(\prod_{i=2}^{N} \mathbb{I}_{\{\tau_i \notin \{\tau_{i-1}, \ldots, \tau_1\}\}} \right). \quad (46)$$

Observe that the third term in the r.h.s is simply the indicator that $\boldsymbol{\tau}$ is a permutation; i.e., for any $\boldsymbol{\tau} \in \{1, ..., N\}^N$,

$$\prod_{i=2}^{N} \mathbb{I}_{\{\tau_i \notin \{\tau_{i-1}, ..., \tau_1\}\}} = \mathbb{I}_{\{\boldsymbol{\tau} \in \Psi_N\}}. \tag{47}$$

Dividing (41) by (46) we obtain the correction factor z for a permutation sample $\boldsymbol{\tau}$ generated using this sequential scheme as

$$z = \left(\prod_{i=2}^{N} \phi_i(\tau_{i-1}, \ldots, \tau_1) \right)^{-1} = \prod_{i=2}^{N} \sum_{t \notin \{\tau_{i-1}, ..., \tau_1\}} A_{y_{\tau_{i-1}}, y_t}. \tag{48}$$

With this quantity in hand, we have all the ingredients needed to implement the sequential importance sampling procedure for computing estimates of $\bar{r}_{1,t'}$ and $\bar{\alpha}_{t,t',t''}$. Notice that computing the terms ϕ_i, and thus computing z, is easy since each of these factors are the normalization terms for the distributions $\mathbf{p} \cdot \mathbf{f}$ which we already compute while performing each iteration of Step 2. Thus, we just need to store the product of these normalizing constants as we sample sequentially to finally obtain the weight z.

5.1.2 Known Endpoints

The causal sampler can easily be modified for the situation when the path endpoints are fixed. In this case, we initialize $\tau_1 = 1$, $\tau_N = N$, set $f_1 = 0$, $f_N = 0$ in the first step, and run the remainder of the procedure as before, sampling until we have a complete permutation. Based on these constraints, the importance sampling weight takes a slightly different form:

$$z = \pi_{y_1} \left(\prod_{i=2}^{N-1} \sum_{t \notin \{\tau_{i-1}, ..., \tau_1\}} A_{y_{\tau_{i-1}}, y_t} \right) A_{y_{\tau_{N-1}}, y_N}. \tag{49}$$

5.1.3 Remarks

Recall that the motivation behind the use of IS is to focus on gathering samples which carry most of the mass of the target distribution $P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$. Simulation results reported at the end of this section indicate that the sequential sampler performs very well. However, it can still happen that the sequential sampler will draw permutations which have negligible posterior probability. This occurs when the sampler gets "stuck" at an intermediary node: the conditional distribution, $\mathbf{p} \cdot \mathbf{f}$, from which we sample in Step 2 vanishes at all the states in S. We find that this happens more frequently with longer paths and when the probability transition matrix, $\mathbf{A}$, is sparse. In some sense this illustrates that the sequential scheme is biased towards choosing more likely transitions near the beginning of the path. This observation motivated us to develop the hierarchical approach described next.

5.2 Two-Stage Hierarchical Sampling

This section describes a two-stage IS scheme which draws sample permutations in a more holistic fashion. The first stage samples from the collection of all possible transitions occurring in this path. The second stage samples from the distribution on all arrangements of these transitions, to form a permutation.

5.2.1 Stage 1: Sampling Transitions

Assume, for the moment, that the length N of the path we are considering is even. Let

$$\Gamma = \big\{(a,b) \in \{1,2,\ldots,N\} \times \{1,2,\ldots,N\} \ : \ a \neq b\big\}$$

be a collection of $|\Gamma| = N(N-1)$ pairs of distinct integers. The first stage of the sampling procedure will amount to sequentially drawing a collection $\mathcal{G}$ of $N/2$ disjoint elements from Γ, as described below. However, before introducing the sampling scheme, some notation is needed. Adopting some order (*e.g.*, lexicographic) for the elements of Γ, we can index them using the set of integers $\{1,...,|\mathcal{T}|\}$. This allows defining a vector of probabilities $\mathbf{p} = (p_1, p_2, \ldots, p_{|\mathcal{T}|})$ on $\{1,...,|\mathcal{T}|\}$, given by

$$p_i = \frac{A_{y_{a_i}, y_{b_i}}}{\sum_{j=1}^{|\mathcal{T}|} A_{y_{a_j}, y_{b_j}}}, \tag{50}$$

where (a_i, b_i) is the i-th pair in $\mathcal{T}$. Because we will design a sequential scheme, and we want the elements of the sample $\mathcal{G}$ to be disjoint, we will also need a mechanism for "masking out" transitions we do not want to sample, conditioned on the current members of $\mathcal{G}$. Let $\mathbf{f}$ denote a length-$|\mathcal{T}|$ binary vector (mask). As before, $\mathbf{p} \cdot \mathbf{f}$ will denote the distribution $\mathbf{p}$, masked by $\mathbf{f}$, as in (39).

We can now describe the first stage of the hierarchical sampling scheme.

Step 1: Set $\mathbf{p}$ as in (50). Set $f_i = 1$, for all $i = 1, \ldots, |\mathcal{T}|$. Set $j = 1$. Set $\mathcal{G} = \mathcal{U} = \emptyset$.

Step 2: Obtain a sample s_j from $\{1,...,|\mathcal{T}|\} \setminus \mathcal{U}$, according to the distribution $\mathbf{p} \cdot \mathbf{f}$.

Update $\mathcal{G} \leftarrow \mathcal{G} \cup \{(a_{s_j}, b_{s_j})\}$. Update $\mathcal{U} \leftarrow \mathcal{U} \cup \{s_j\}$.

For each $i = 1, \ldots, |\mathcal{T}|$, if $\{a_i, b_i\} \cap \{a_{s_j}, b_{s_j}\} \neq \emptyset$, set $f_i \leftarrow 0$.

Step 3: If $j < N/2$, set $j \leftarrow j + 1$ and go back to Step 2.

When the procedure terminates $\mathcal{G}$ contains exactly $N/2$ disjoint transitions which we will permute in the next stage.

In the case that N is odd, we still only need to repeat the sampling procedure while $j < N/2$, but we will end up only sampling $\lfloor N/2 \rfloor$ transitions, leaving out one integer, say k, between 1 and N, which does not appear in any of the transitions in $\mathcal{G}$. We do one final update in which this singleton is appended to $\mathcal{G}$, that is $\mathcal{G} \leftarrow \mathcal{G} \cup \{k\}$. In general, $\mathcal{G}$ will be a collection of $\lceil N/2 \rceil$ elements (pairs, with possibly one singleton).

5.2.2 Stage 2: Permuting Transitions

In stage 2, we draw a sample permutation $\boldsymbol{\tau}$ from Ψ_N, under the constraint that it must include all transitions in $\mathcal{G}$, the collection of transitions sampled in stage 1. Let $N' = \lceil N/2 \rceil$ and let $\mathcal{G} = \{(a_{s_1}, b_{s_1}), ..., (a_{s_{N'}}, b_{s_{N'}})\}$. Observe that sampling from the constrained distribution is equivalent to drawing a permutation $\boldsymbol{\tau}'$ from $\Psi_{N'}$ and then defining $\boldsymbol{\tau}$ by concatenating the elements of $\mathcal{G}$ in the order prescribed by $\boldsymbol{\tau}'$. That is, we set

$$\boldsymbol{\tau} \equiv \boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G}) = (t_{\tau'_1}, t_{\tau'_2}, \ldots, t_{\tau'_{N'}}), \tag{51}$$

where the notation $\boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G})$ is used to stress that the resulting permutation $\boldsymbol{\tau} \in \Psi_N$ is a function of the transitions sampled in the first stage, $\mathcal{G}$, and the smaller permutation $\tau' \in \Psi_{N'}$ drawn at stage 2. Here, we assume that $(N/2)!$ is small enough to allow enumeration of all permutations $\boldsymbol{\tau}' \in \Psi_{N'}$. Thus, we calculate

$$P[\boldsymbol{\tau}'|\mathcal{G}, \mathbf{y}, \mathbf{A}, \boldsymbol{\pi}] = P[\boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G})|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}] \propto \pi_{y_{\tau_1}} \prod_{t=2}^{N} A_{y_{\tau_{t-1}}, y_{\tau_t}}, \tag{52}$$

Finally, we draw a permutation $\boldsymbol{\tau}'$ according to the probability distribution defined by (52), and set $\boldsymbol{\tau} = \boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G})$.

5.2.3 Sampling Distribution

To use IS, we need the sampling distribution $R[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]$ of the two-stage hierarchical sampler. Since the first stage is independent of the second one, R decomposes into

$$R[\boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G})|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}] = R[\boldsymbol{\tau}'|\mathcal{G}, \mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]\ R[\mathcal{G}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]. \tag{53}$$

As in Section 5.1.1, we simplify the notation by omitting the explicit dependence on $\mathbf{y}$, $\mathbf{A}$, and $\boldsymbol{\pi}$. Recall that $\mathcal{U} = \{s_1, ..., s_{N'}\}$ is the sequence of indices of pairs sampled at the first stage of the algorithm; of course, knowing $\mathcal{U}$ is the same as knowing $\mathcal{G}$. The probability $R[\mathcal{G}]$ factors in a similar fashion to the causal sampling method due to the sequential nature of the procedure:

$$R[\mathcal{G}] = R[s_1, ..., s_{N'}] = R[s_1]\ R[s_2|s_1]\ \cdots R[s_{N'}|s_{N'-1}, \ldots, s_1].$$

We start with $R[s_1]$; with p_i as defined in (50), it's clear that $R[s_1] = p_{s_1}$. Next, consider $R[s_2|s_1]$; define the index set

$$\mathcal{I}_2(s_1) = \big\{i \in \{1, \ldots, |\mathcal{T}|\} : \{a_i, b_i\} \cap \{a_{s_1}, b_{s_1}\} = \emptyset\big\},$$

containing the indices of "valid" transitions after one iteration of the first stage. Then,

$$R[s_2|s_1] = \begin{cases} p_{s_2} \left(\sum_{j \in \mathcal{I}_2} p_j\right)^{-1} & \Leftarrow \quad s_2 \in \mathcal{I}_2(s_1) \\ 0 & \Leftarrow \quad s_2 \notin \mathcal{I}_2(s_1). \end{cases} \tag{54}$$

In a more compact notation,

$$R[s_2|s_1] = p_{s_2}\ \phi_2(s_1)\ \mathbb{I}_{\{s_2 \in \mathcal{I}_2(s_1)\}}, \tag{55}$$

where

$$\phi_2(s_1) = \left(\sum_{j \in \mathcal{I}_2(s_1)} p_j \right)^{-1}. \tag{56}$$

For the general k-th step of the first stage, we have

$$R[s_k | s_{k-1}, ..., s_1] = p_{s_k} \; \phi_k(s_{k-1}, ..., s_1) \; \mathbb{I}_{\{s_k \in \mathcal{I}_k(s_{k-1}, ..., s_1)\}}, \tag{57}$$

where $\mathcal{I}_k(s_{k-1}, ..., s_1)$ is the set of indices corresponding to nonzero entries of $\mathbf{f}$, prior to sampling s_k, and

$$\phi_k(s_{k-1}, ..., s_1) = \left(\sum_{j \in \mathcal{I}_k(s_{k-1}, ..., s_1)} p_j \right)^{-1}. \tag{58}$$

Finally, notice that in the case of odd N, the last step of appending the remaining node to $\mathcal{G}$ doesn't affect $R[\mathcal{G}]$ because it is a deterministic operation.

Putting this all together, we have that the sampling distribution on collections of transitions corresponding to the first stage is

$$R[\mathcal{G}] = R[s_1, ..., s_{\lfloor N/2 \rfloor}] = \frac{\prod_{k=1}^{\lfloor N/2 \rfloor} p_{s_k}}{\prod_{k=1}^{\lfloor N/2 \rfloor} \sum_{j \in \mathcal{I}_k(s_{k-1}, ..., s_1)} p_j} \prod_{k=2}^{\lfloor N/2 \rfloor} \mathbb{I}_{\{i_k \in \mathcal{I}_k(s_{k-1}, ..., s_1)\}} \tag{59}$$

Each term in the numerator comes directly from our current estimate of the probability transition matrix, $\mathbf{A}$, and the terms in the denominator correspond to $\mathbf{p} \cdot \mathbf{f}$ at each iteration of the first stage. All these quantities can easily be calculated and stored as we generate $\mathcal{G}$ in the first stage. The product of indicator functions is just the indicator function that guarantees that $\mathcal{G}$ is "valid", that is, it contains a set of disjoint pairs; thus, it will equal one for any valid $\mathcal{G}$.

In the second stage of the algorithm, a permutation $\boldsymbol{\tau}'$ is drawn with sampling probability

$$R[\boldsymbol{\tau}' | \mathcal{G}] = \frac{P[\boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G})]}{\sum_{\boldsymbol{\tau}' \in \Psi_{N'}} P[\boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G})]}, \tag{60}$$

where $P[\boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G})]$ is defined in (52). Using (53), we have

$$R[\boldsymbol{\tau} | \mathbf{y}, \mathbf{A}, \boldsymbol{\pi}] \propto \left(\frac{P[\boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G})]}{\sum_{\boldsymbol{\tau}' \in \Psi_{N'}} P[\boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G})]} \right) \left(\frac{\prod_{k=1}^{\lfloor N/2 \rfloor} p_{s_k}}{\prod_{k=1}^{\lfloor N/2 \rfloor} \sum_{j \in \mathcal{I}_k(s_{k-1}, ..., s_1)} p_j} \right). \tag{61}$$

Finally, the weight for each sampled permutation $\boldsymbol{\tau} = \boldsymbol{\tau}(\boldsymbol{\tau}', \mathcal{G})$, to be used in IS, is

$$z = \frac{P[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]}{R[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]}, \tag{62}$$

where $R[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]$ is given by (61) and $P[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]$ by (41)

5.2.4 Known Endpoints

When the endpoints of each path are known, we simply exclude these two elements in the first stage and only consider transitions among internal nodes (indices $2, \ldots, N-1$). In the second stage, we form samples $\boldsymbol{\tau}$ by permuting the internal transitions sampled in the first stage and applying the source and destination indices (1 and N) as a prefix and suffix, respectively, in the permutation. The form of the weights does not change.

5.2.5 Other Variations

The two-stage scheme just described reduces the complexity of sampling a permutation from $N!$ (required to enumerate all possible orderings) to $(N/2)!$ operations (required to enumerate all permutations of transitions in the second stage). For very long paths, this reduction may still leave too many combinations to evaluate in an acceptable amount of time. Rather than jumping directly from the first stage to the second one, a natural way to extend this idea is to include additional intermediate stages similar to the first where we sample larger 2^k-tuples constructed from the 2^{k-1}-tuples sampled in the previous stage. That is, in the first stage we sample a suitable set of transitions, say $\mathcal{G}_1$. Then, in the next stage we sample a suitable collection of pairs of elements $\mathcal{G}_1$, yielding a collection of quadruples, $\mathcal{G}_2$, and so on.

Also, rather than sampling a permutation of transitions in the second stage, one might consider using all such permutations, since we effectively need to calculate each of their posterior probabilities in order to obtain a sample ordering.

5.3 Performance Comparison

A standard error metric for comparing two distributions P and $\widehat{P}$ taking values on the finite set Ψ_N is the ℓ_1 distance, defined as

$$\|P - \widehat{P}\|_1 = \sum_{\mathbf{r} \in \Psi_N} \left| P[\mathbf{r}] - \widehat{P}[\mathbf{r}] \right|. \tag{63}$$

Given a sequence of permutations, $\mathbf{r}^1, \ldots, \mathbf{r}^L$, drawn from the importance sampling distribution R along with the corresponding weights, $z_1, \ldots, z_L$, we can compute the empirical distribution $\widehat{P}_R$ induced on Ψ_N according to

$$\widehat{P}_R[\mathbf{r}] = \frac{\sum_{i=1}^{L} z_i \mathbb{I}_{\{\mathbf{r}^i = \mathbf{r}\}}}{\sum_{i=1}^{L} z_i}. \tag{64}$$

Notice that the Monte Carlo sufficient statistics $\widehat{\alpha}^{(m)}_{t',t''}$ and $\widehat{r}^{(m)}_{1,t'}$ are just sums of certain terms $\widehat{P}_R[\mathbf{r}]$. For example, $\widehat{\alpha}_{t',t''} = \sum_{\mathbf{r}\in\Psi_N} \widehat{P}_R[\mathbf{r}] \sum_{t=2}^{N_m} r_{t-1,t''} r_{t,t'}$. Thus,

$$\left|\bar{\alpha}^{(m)}_{t',t''} - \widehat{\alpha}^{(m)}_{t',t''}\right| \leq \left\|P - \widehat{P}_R\right\|_1 \tag{65}$$

$$\left|\bar{r}^{(m)}_{1,t'} - \widehat{r}^{(m)}_{1,t'}\right| \leq \left\|P - \widehat{P}_R\right\|_1 . \tag{66}$$

If the ℓ_1 error between the true distribution on permutations and the empirical importance sampling distribution is small then all of the estimated sufficient statistics will be close to the corresponding exact value.

We have evaluated the performance of the various proposed sampling schemes via simulation. To assess performance over a varying range of conditions, we consider three scenarios: 1) the distribution over all permutations is roughly uniform, 2) the distribution is moderately peaked, and 3) the distribution is highly concentrated around just a few of the possible permutations. These scenarios were chosen based on observations made while experimenting with the EM algorithm. The first scenario is typical during the first few EM iterations, the second scenario is typical during intermediate EM iterations, and the third scenario is typical when the algorithm has nearly converged. We consider a length-8 path with known endpoints, so that there are $6! = 720$ possible path orderings. This path length is long enough to get a feel for how each sampling scheme will perform for longer paths, while still allowing us to enumerate all orderings.

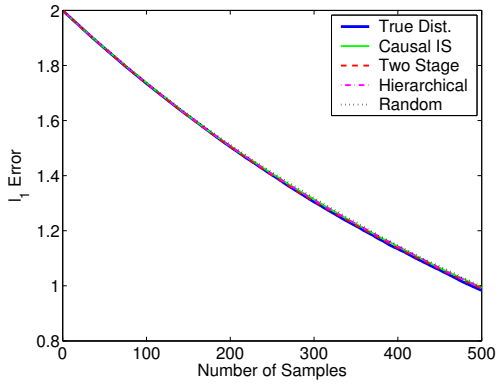
Figure 1 depicts the ℓ_1 error between the true and importance sample-induced distributions on permutations as a function of the number of samples gathered for different sampling schemes in each of the three scenarios considered. The curve labelled "True Dist." corresponds to sampling from the true distribution on permutations, is shown as a reference and is only possible when we can enumerate all permutations. The "Causal IS" curve corresponds to the causal importance sampling scheme described in Section 5.1. The "Two Stage" curve denotes performance for the two stage hierarchical scheme described in Section 5.2, "Hierarchical" corresponds to a completely hierarchical variation on this scheme, and "Random" refers to an approach where we sample from a uniform distribution on permutations, which is shown as a baseline comparison. Each curve in this figure was generated by averaging over 50 Monte Carlo simulations. Note that these curves depict performance using up to 500 samples for a path with 720 possible orderings. This is actually quite a generous helping of samples! In our experiments with Internet data we have encountered paths of up to length 27, and observed good reconstruction performance using as few as $2000 \ll 15! \approx 1.31 \times 10^{12}$ importance samples. Thus, performance for very few samples is of great interest. Note also that all of the sampling schemes except for random sampling converge more rapidly when the target distribution is more concentrated.

As expected, all of the sampling schemes give the same performance when the Markov chain parameters are such that the distribution on all orderings is roughly uniform. However, as the distribution becomes more and more concentrated around just a few orderings there is a noticeable difference between the various sampling schemes, particularly in the 10-100 sample range. The uniform random sampling scheme clearly performs the worst on more concentrated distributions, as would also be expected, since

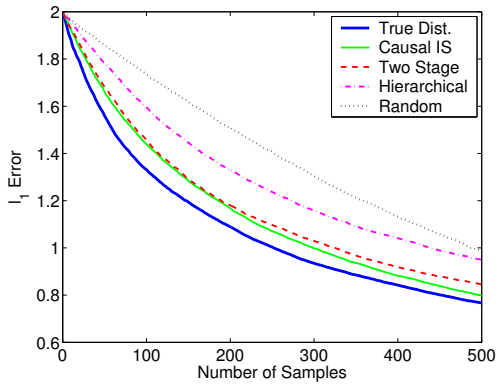
the uniform distribution is not using any information about the target distribution. Of the importance sampling schemes which are practical for long paths, our simulation results indicate that the causal sampling scheme performs the best, and is slightly better than the two-stage sampler.

In terms of computational complexity, the causal sampler is the simplest and fastest scheme to implement, requiring only $O(N)$ operations per sample, where N is the path length. The two-stage sampler converges to the target distribution nearly as fast, but requires $O\big((N/2)!\big)$ operations per sample due to the enumeration of all transition permutations in Stage 2 (see Section 5.2.2). Finally, the fully hierarchical scheme converges to the target distribution slower than the causal or two-stage samplers, but offers middle grounds as far as computational complexity at $O(N^2 \log N)$ operations per sample (required to compute the distribution on the elements of $\mathcal{G}_i$ at each stage). The upshot is that the causal sampling procedure is simple to implement, fast, and it empirically outperforms more computationally complex sampling schemes. Figures 2, 3, and 4 depict the probability transition matrix, the true distribution on permutations, and typical distributions estimated using each of the importance sampling schemes after 500 samples.

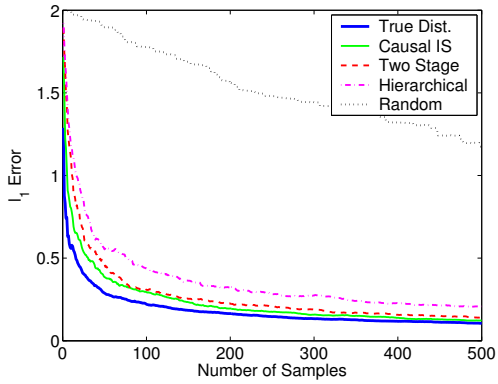
We have also compared the efficacy of each sampling scheme for estimating network parameters within the EM algorithm. In this experiment we generated a random network with 250 nodes and simulated 60 random sample paths through this network ranging in length from 4 to 10 hops. Then we estimated a probability transition matrix for the network using the EM algorithm with different approximate E-steps, assuming the endpoints of each path to be known. Figure 5 depicts the marginal log-likelihood of the data computed according to (6) using the probability transition matrices returned by the EM algorithm. In our experiments we varied the number of samples-per-path between 20 and 100, regardless of the length of the path being considered, to test the behavior of each sampling scheme over a variety of conditions ($100 \ll 10! \approx 3$ million). The horizontal dashed line across the top of the figure marks the marginal log likelihood value computed using a transition matrix derived from correctly ordered paths. In addition to the sampling schemes described above, we also included a variant of the two-stage sampling scheme which uses all permutations of the selected transitions in Stage 2. This variant did a better job of maximizing the marginal log likelihood than the other two-stage scheme for the same computational complexity, but still does not perform as well as the causal sampling scheme in this experiment.



(a) Uniform distribution on permutations

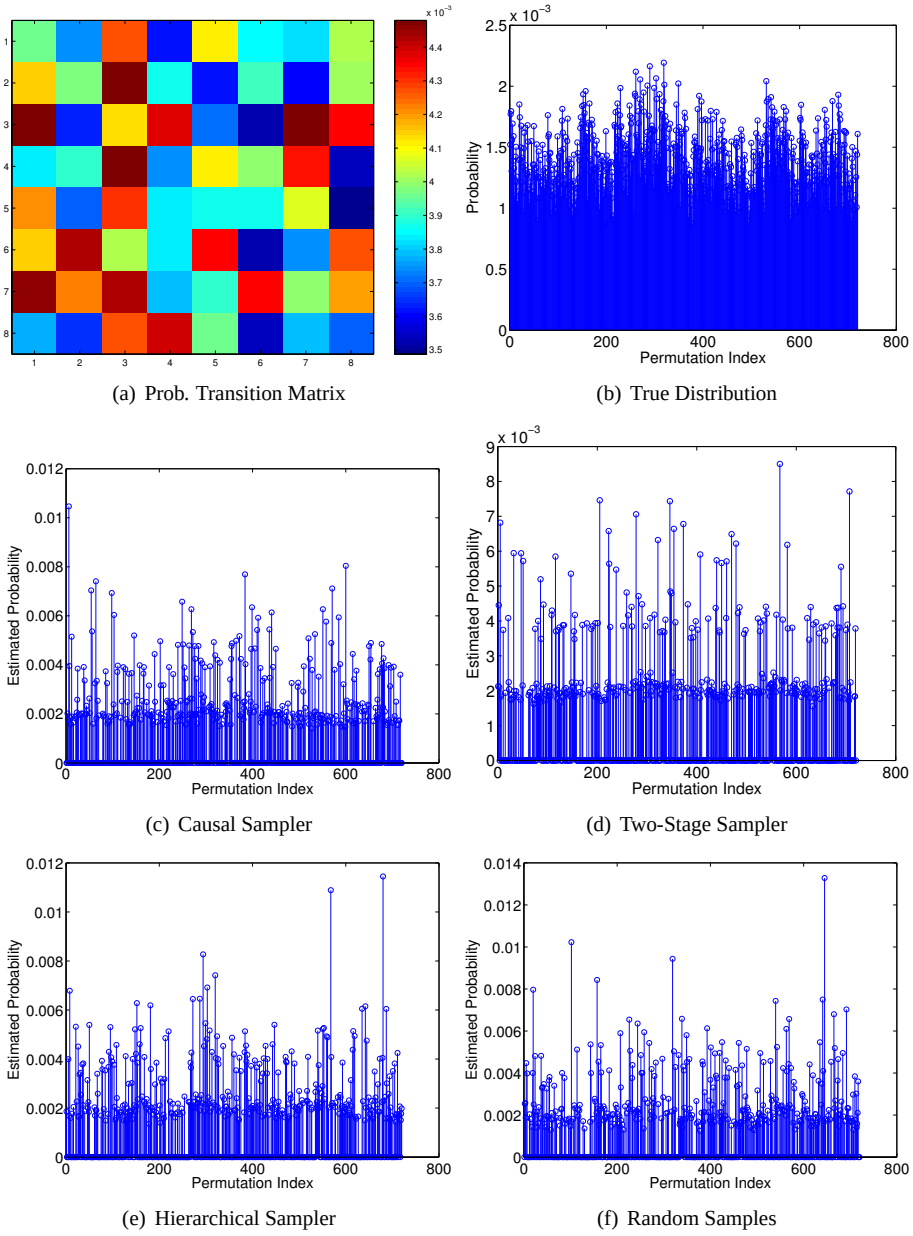


(b) Moderately peaked distribution



(c) Highly concentrated distribution

Figure 1: ℓ_1 error as a function of the number of importance samples drawn for various sampling schemes in the following scenarios: (a) a roughly uniform distribution on the permutations, (b) moderately peaked distribution, (c) highly concentrated distribution. The curves in each figure were calculated by averaging over 50 Monte Carlo simulations.

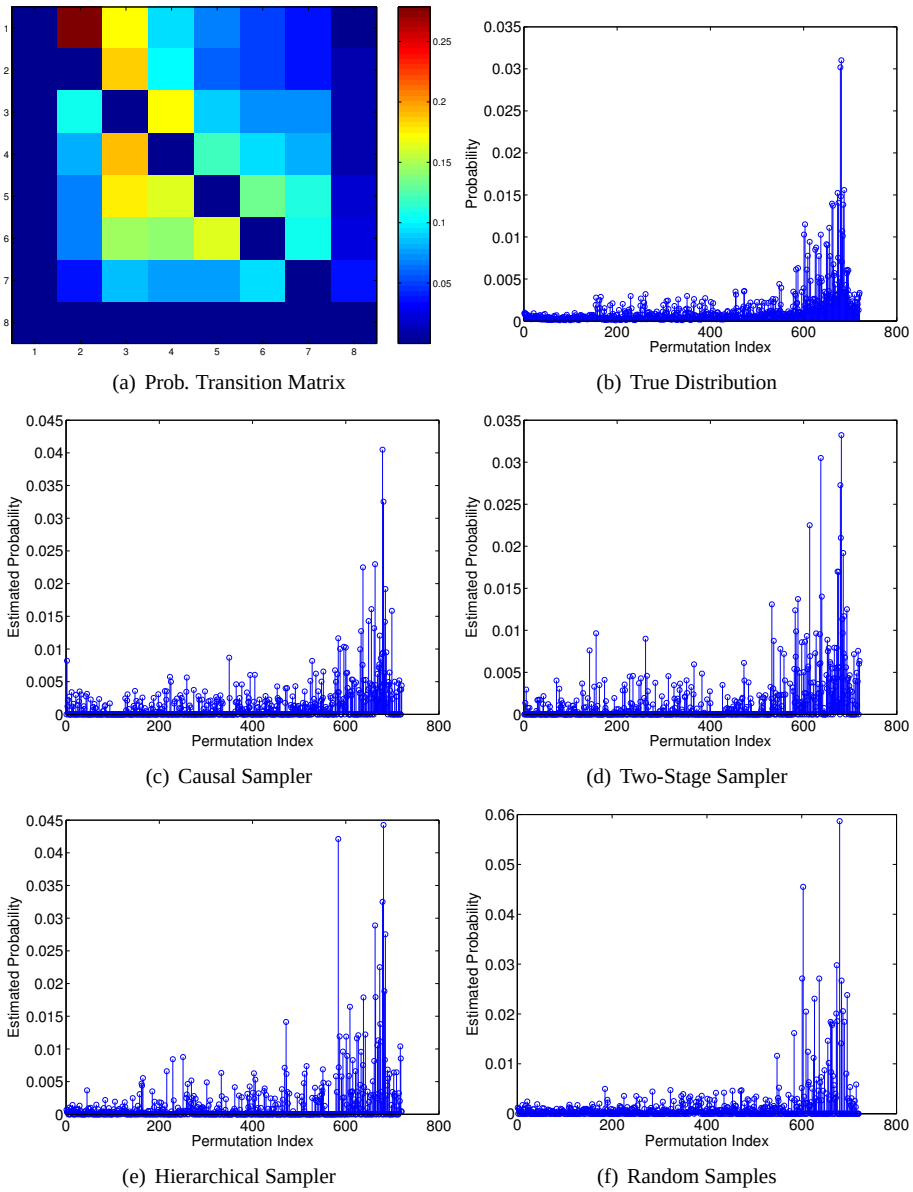


(a) Prob. Transition Matrix (b) True Distribution

(c) Causal Sampler (d) Two-Stage Sampler

(e) Hierarchical Sampler (f) Random Samples

Figure 2: Comparing sampling schemes for the scenario where all orderings are approximately uniformly distributed. This figure depicts the (a) probability transition matrix, (b) true distribution on permutations, and estimates of the distribution using 500 samples from each of the (c) causal, (d) two-stage, (e) completely hierarchical, and (f) uniform random permutation samplers.

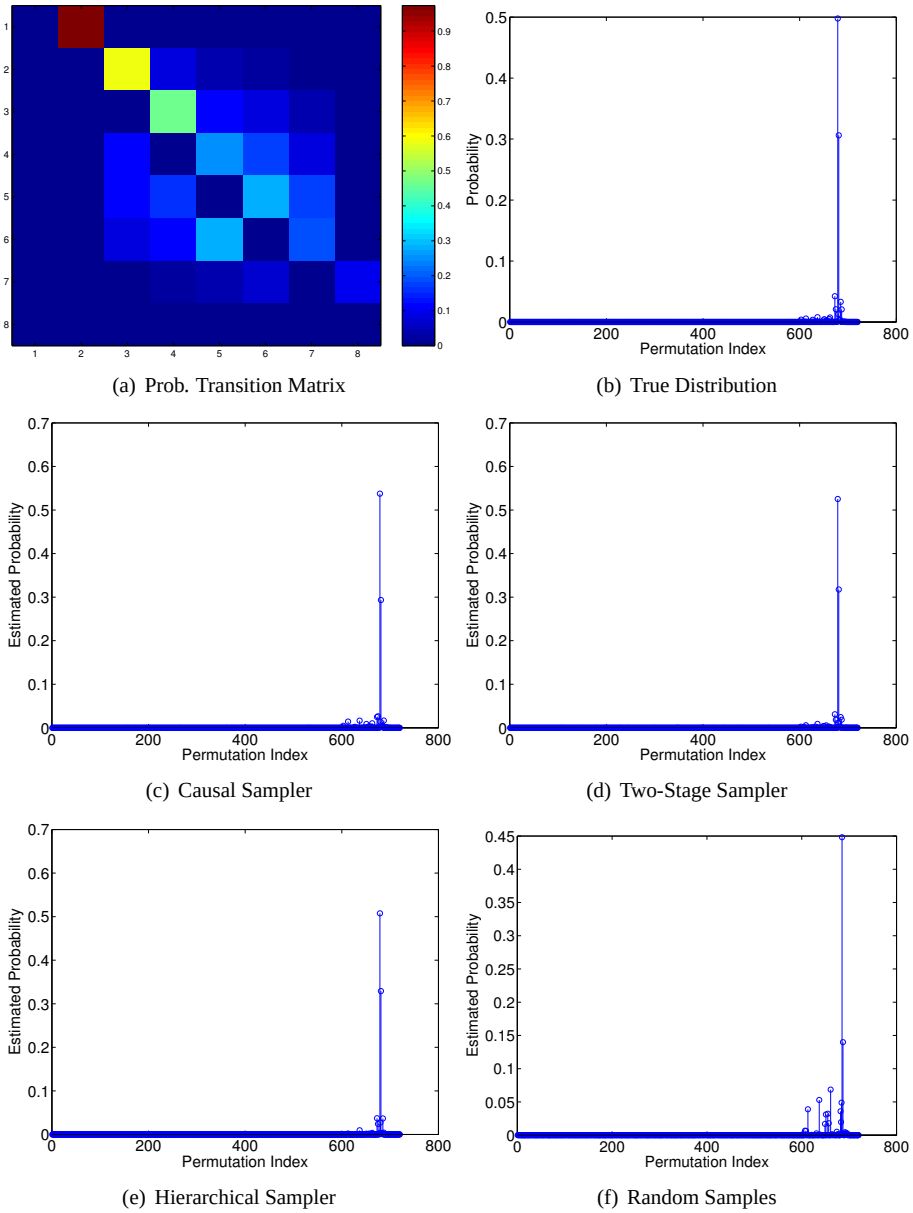


(a) Prob. Transition Matrix (b) True Distribution

(c) Causal Sampler (d) Two-Stage Sampler

(e) Hierarchical Sampler (f) Random Samples

Figure 3: Comparing sampling schemes for the scenario where the distribution on orderings is somewhat concentrated. This figure depicts the (a) probability transition matrix, (b) true distribution on permutations, and estimates of the distribution using 500 samples from each of the (c) causal, (d) two-stage, (e) completely hierarchical, and (f) uniform random permutation samplers.



(a) Prob. Transition Matrix (b) True Distribution

(c) Causal Sampler (d) Two-Stage Sampler

(e) Hierarchical Sampler (f) Random Samples

Figure 4: Comparing sampling schemes for the scenario where the distribution on orderings is somewhat concentrated. This figure depicts the (a) probability transition matrix, (b) true distribution on permutations, and estimates of the distribution using 500 samples from each of the (c) causal, (d) two-stage, (e) completely hierarchical, and (f) uniform random permutation samplers.

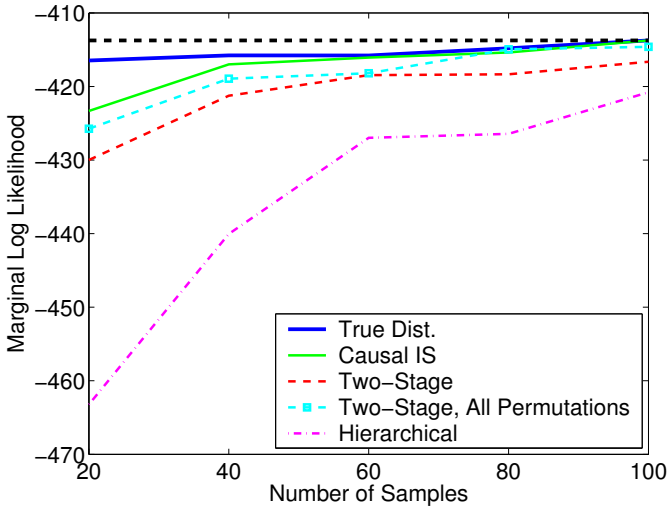


Figure 5: Using various approximate E-steps in the EM algorithm for estimating the Markov transition matrix of a simulated network. The horizontal dashed line at the top of the figure marks the marginal log likelihood of the data using the transition matrix derived using the correctly ordered paths. The curves in this figure correspond to the average over 10 Monte Carlo simulations.

6 Monte Carlo EM Convergence

When exact computation of the E-step is used, our EM algorithm is guaranteed to converge via well-known convergence results due to Wu and Boyles [4, 22]. Let $\boldsymbol{\theta}^k = \left(\mathbf{A}^k, \boldsymbol{\pi}^k\right)$ denote parameter estimates calculated at the kth EM iteration (using the exact EM expressions). Because we choose $\boldsymbol{\theta}^{k+1} = \left(\mathbf{A}^{k+1}, \boldsymbol{\pi}^{k+1}\right)$ according to (18) in the M-step, our iterates satisfy the *monotonicity property*:

$$Q\left(\boldsymbol{\theta}^{k+1}; \boldsymbol{\theta}^k\right) \quad \geq \quad Q\left(\boldsymbol{\theta}^k; \boldsymbol{\theta}^k\right). \tag{67}$$

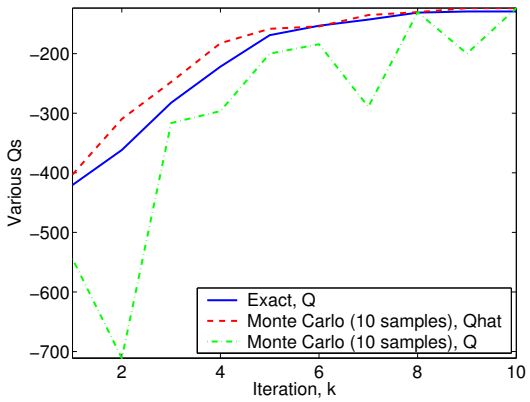
The marginal log-likelihood (6) is continuous in its parameters $\mathbf{A}$ and $\boldsymbol{\pi}$ and it is bounded above, thus the monotonicity property guarantees that the EM iterates converge monotonically to a local maximum of the marginal likelihood.

Exact calculation of sufficient statistics is not always practical for co-occurrence observations with many vertices which is why we propose the Monte Carlo schemes described in the previous section. However, when Monte Carlo calculation of the sufficient statistics is used we no longer have the monotonicity property. In particular, the M-step now becomes

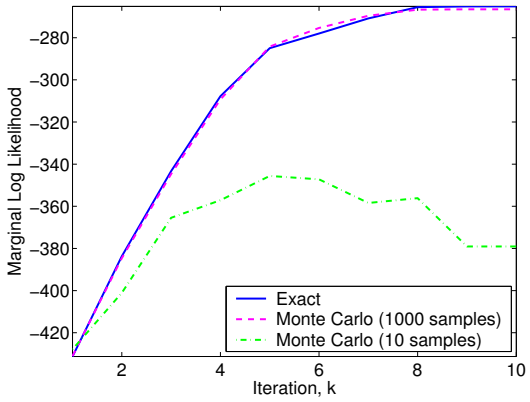
$$\widehat{\boldsymbol{\theta}}^{k+1} \equiv \left(\widehat{\mathbf{A}}^{k+1}, \widehat{\boldsymbol{\pi}}^{k+1}\right) \quad = \quad \arg\max_{\mathbf{A}, \boldsymbol{\pi}} \widehat{Q}\left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k\right),$$

where $\widehat{Q}$ is defined analogously to Q in (24), but with terms $\bar{\alpha}^{(m)}_{t',t''}$ and $\bar{r}^{(m)}_{1,t'}$ replaced by $\widehat{\alpha}^{(m)}_{t',t''}$ and $\widehat{r}^{(m)}_{1,t'}$, their corresponding importance sample approximations. Consequently, care must be taken to ensure that $\widehat{Q}$ approximates Q well enough so that the EM algorithm is not swamped with error from the Monte Carlo estimates.

Consider the following toy example using simulated observations. We begin with a network of 142 vertices and randomly generate 40 co-occurrence observations where between 4 and 8 vertices co-occur in each observation. Then we run three versions of the EM algorithm on this data set. Exact E-step computation is used in one version, and in the other two versions causal importance sampling is used with 10 and 1000 importance samples per observation. Each version of the algorithm starts from the same initial estimate. Figure 6(a) depicts $Q(\boldsymbol{\theta}^{k+1}; \boldsymbol{\theta}^k)$ for the exact EM iterates, as well as $\widehat{Q}(\widehat{\boldsymbol{\theta}}^{k+1}; \widehat{\boldsymbol{\theta}}^k)$ and $Q(\widehat{\boldsymbol{\theta}}^{k+1}; \widehat{\boldsymbol{\theta}}^k)$ for the 10 sample Monte Carlo iterates. Note that for the Monte Carlo EM algorithm, even though $\widehat{Q}$ increases monotonically by design, Q may not increase, and consequently the monotonicity property (67) may not hold. Figure 6(b) shows the marginal log-likelihood for all three versions of the algorithm. Performance of the Monte Carlo EM algorithm closely resembles that of the exact EM algorithm when enough importance samples are used, however when too few samples are used the resulting estimates may be of a much poorer quality.



(a)



(b)

Figure 6: An example with simulated observations illustrating that the Monte Carlo EM algorithm may not result in monotonic increase of the marginal log-likelihood if too few Monte Carlo samples are used. The solid line in (a) is $Q(\boldsymbol{\theta}^{k+1};\boldsymbol{\theta}^k)$ for exact EM iterations, the dashed line is $\widehat{Q}(\widehat{\boldsymbol{\theta}}^{k+1};\widehat{\boldsymbol{\theta}}^k)$ and the dash-dot line is $Q(\widehat{\boldsymbol{\theta}}^{k+1};\widehat{\boldsymbol{\theta}}^k)$ for Monte Carlo EM iterations using only 10 samples. Even though $\widehat{Q}$ increases monotonically, Q may not be monotonic for the Monte Carlo EM algorithm. Figure (b) depicts the marginal log-likelihood for exact EM iterates and for two versions of the Monte Carlo EM. Monte Carlo EM performance closely resembles that of the exact EM algorithm when sufficiently many importance samples are used.

In the recent literature researchers have considered the question of how many importance samples must be used in a Monte Carlo E-step [3, 5, 9]. These studies seek a balance between monotonicity and efficiency. We would like to use enough samples to guarantee that monotonicity holds with sufficiently high probability while not using unnecessarily many samples. Booth et al. argue that if the same number of importance samples is used at each EM iteration then the algorithm will eventually be swamped by Monte Carlo error and will not converge [3]. They also suggest requiring that a convergence criterion be satisfied on multiple successive iterations since the criterion may be met prematurely due to poor Monte Carlo approximations.

In [5], Caffo et al. propose a method for automatically adapting the number of Monte Carlo samples used at each EM iteration. To lighten notation, we drop the superscripts k and $k+1$. Let $\Delta(\boldsymbol{\theta}) = Q(\boldsymbol{\theta};\boldsymbol{\theta}') - Q(\boldsymbol{\theta}';\boldsymbol{\theta}')$ and $\widehat{\Delta}(\boldsymbol{\theta}) = \widehat{Q}(\boldsymbol{\theta};\boldsymbol{\theta}') - \widehat{Q}(\boldsymbol{\theta}';\boldsymbol{\theta}')$. Furthermore, let $\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \widehat{Q}(\boldsymbol{\theta};\boldsymbol{\theta}')$, where $\boldsymbol{\theta}' = \boldsymbol{\theta}^k$ is a fixed constant, determined at the previous EM iteration. Recall that L importance samples are used to calculate $\widehat{Q}$. The algorithm of Caffo et al. is based on a Central Limit Theorem-like approximation in which they show that $\sqrt{L}\left|\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})\right|$ converges in distribution to the standard normal. Observe that the monotonicity property (67) is equivalent to the condition $\Delta(\widehat{\boldsymbol{\theta}}) \geq 0$, and although we cannot calculate this quantity without exactly computing the sufficient statistics in the E-step, we can compute $\widehat{\Delta}(\widehat{\boldsymbol{\theta}})$. The scheme proposed by Caffo et al. amounts to increasing the number of Monte Carlo samples until $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) > \epsilon$ for a user-specified $\epsilon > 0$. Then, via an asymptotic standard normal tail approximation, they obtain a statement of the form

$$\Pr\left(\left|\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})\right| \geq \epsilon\right) \quad \leq \quad \delta(\epsilon).$$

Based on this statement they claim that monotonicity holds with probability at least $1-\delta(\epsilon)$. They further remark that if ϵ_k is chosen at each iteration so that $\sum_{k=1}^{\infty} \delta(\epsilon_k) < \infty$ then by the Borel-Cantelli Lemma,

$$\Pr\left(\left|\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})\right| \geq \epsilon_k \text{ i.o.}\right) \quad = \quad 0,$$

and so eventually, $\left|\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})\right| < \epsilon_k$ with probability 1. Of course, in practice only a finite number of EM iterates are used and so we may never reach the stage where all iterates are monotonic.

Notice that for the monotonicity condition $\Delta(\widehat{\boldsymbol{\theta}}) \geq 0$ to hold in the above framework, the events

$$\left\{\left|\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})\right| \leq \epsilon\right\} \quad \text{and} \quad \left\{\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) \geq \epsilon\right\}$$

must occur simultaneously. Because the probabilistic bound above only addresses one of these events we refer to this type of result as guaranteeing an (ϵ, δ)-*probably approximately monotonic* update, or PAM for short. More generally, an (ϵ, δ)-PAM result states that with probability at least $1-\delta$, the update will be ϵ-approximately monotonic; *i.e.*, $\left|\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})\right| \leq \epsilon$ implies $\Delta(\widehat{\boldsymbol{\theta}}) \geq -\epsilon$.

Rather than resorting to asymptotic approximations to obtain such a result, we can take advantage of the specific form of Q in our problem to obtain the following finite-sample PAM result. Recall that the sufficient statistics computed in the E-step are

independently for each observation. That is, the importance samples used to compute $\{\widehat{\alpha}^{(m)}_{t',t''}\}$ are independent of those used to compute $\{\widehat{\alpha}^{(m')}_{t',t''}\}$ for $m \neq m'$. Denote by L_m the number of importance samples used to compute sufficient statistics for observation $\mathbf{x}^{(m)}$. Exact E-step computation for this observation requires $O(N_m!)$ operations. Similarly, we should expect that larger observations will require more importance samples for two reasons: 1) there are more sufficient statistics associated with this observation (N_m^2 in total), and 2) there are more ways to shuffle these observations.

In the previous section we derived closed form expressions for the importance sample weights, $z_i = \frac{P[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]}{R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]}$, where P denotes the target distribution and R the importance sampling distribution. One key assumption is that P is absolutely continuous with respect to R; that is, $P[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}] = 0$ for every permutation $\mathbf{r}$ with $R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}] = 0$. We adopt the convention $0/0 = 0$ so that $z_i = 0$ for such samples, and this guarantees that $z_i < \infty$. Because Hoeffding's inequality is used to derive the bound below, the number of importance samples required depends on the range of z_i. For the mth observation, define

$$b_m \quad = \quad \max_{\mathbf{r}\in\Psi_{N_m}} \frac{P[\mathbf{r}|\mathbf{x}^{(m)},\mathbf{A},\boldsymbol{\pi}]}{R[\mathbf{r}|\mathbf{x}^{(m)},\mathbf{A},\boldsymbol{\pi}]}. \tag{68}$$

Because the set Ψ_{N_m} is finite, P and R have finite support and the maximum is well-defined.

There is one other subtlety we will account for in our bounds. Because $\widehat{Q}(\boldsymbol{\theta};\boldsymbol{\theta}')$ has terms $\log A_{i,j}$ and $\log \pi_i$, in practice we typically bound $A_{i,j}$ and π_i away from zero to ensure that $\widehat{Q}$ does not go to $-\infty$. To this end, we will assume that $\widehat{A}_{i,j} \geq \theta_{\min}$ and $\widehat{\pi}_i \geq \theta_{\min}$ for some $0 < \theta_{\min} < |S|^{-1}$. The upper bound on $\theta_{\min}$ ensures it is still possible to satisfy the constraints (1). Generally we choose $\theta_{\min}$ very close to zero; at machine precision, for example.

We have the following finite-sample PAM result for our Monte Carlo EM algorithm. Proofs of all results reported in this section appear in the appendix.

Theorem 1. *Let $\epsilon > 0$ and $\delta > 0$ be given and assume there exists $\theta_{\min} \in (0, |S|^{-1})$ such that $A'_{i,j} \geq \theta_{\min}$ and $\pi'_i \geq \theta_{\min}$ for all i and j. If*

$$L_m \quad = \quad \frac{2T^2 N_m^4 b_m^2 \,|\log\theta_{\min}|^2}{\epsilon^2} \log\left(\frac{2N_m^2}{1-(1-\delta)^{1/T}}\right) \tag{69}$$

importance samples are used for the mth observation then $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) < \epsilon$ with probability greater than $1-\delta$.

Remark 1. Because $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) \geq 0$ by definition, this result guarantees that $\Delta(\widehat{\boldsymbol{\theta}}) > -\epsilon$ with probability greater than $1-\delta$.

Remark 2. Recall that the computational complexity of the exact E-step is $O(N_m!)$ operations for the mth observation. In contrast, $O(N_m)$ operations are required to generate one sample using the causal importance sampling scheme, and so only $O(N_m^5)$ operations are needed to have a PAM update for the mth observation using the Monte Carlo E-step. This clearly demonstrates that the computational complexity of the

Monte Carlo E-step scales polynomially in the observation size, compared to exponential scaling for the exact E-step.

Remark 3. The choice L_m is roughly a factor of T off from the number of importance samples we would expect to need, based on an asymptotic variance calculation. Observe that for fixed $\boldsymbol{\theta}$,

$$\begin{aligned}\text{Var}\left(\widehat{\Delta}(\boldsymbol{\theta})\right) &\simeq \text{Var}\left(\sum_{m=1}^{T}\sum_{t',t''=1}^{N_m}\widehat{\alpha}_{t',t''}^{(m)} + \sum_{m=1}^{T}\sum_{t'=1}^{N_m}\widehat{r}_{1,t'}^{(m)}\right) \\ &= \sum_{m=1}^{T}\text{Var}\left(\sum_{t',t''=1}^{N_m}\widehat{\alpha}_{t',t''}^{(m)} + \sum_{t'=1}^{N_m}\widehat{r}_{1,t'}^{(m)}\right),\end{aligned}$$

since independent sets of importance samples are used to calculate sufficient statistics for different observations. It is not too difficult to believe that the variance of an individual approximate statistic decays according to the parametric rate; *i.e.*, $\text{Var}(\widehat{\alpha}_{t',t''}^{(m)}) \simeq 1/L_m$. In total, there are N_m^2 sufficient statistics for the mth observation, and they are all potentially correlated since they are functions of the same set of importance samples. Then we have

$$\text{Var}\left(\widehat{\Delta}(\boldsymbol{\theta})\right) \simeq \sum_{m=1}^{T}\frac{\left(N_m^2\right)^2}{L_m}.$$

Therefore, to have $\text{Var}\left(\widehat{\Delta}(\boldsymbol{\theta})\right)$ equal to a constant which is independent of T and the N_m we need $L_m \propto TN_m^4$. The additional factor of T in our bound is essentially an artifact from our application of the union bound.

Remark 4. In practice, we do not know b_m explicitly for observations with N_m large, since calculating b_m essentially requires enumerating every permutation of the co-occurrence. However, b_m could potentially be very large. We could probably do better using Bernstein's inequality instead of Hoeffding's inequality. Then, instead of dependence on b_m, the number of importance samples required would depend on the variance of the importance sample weights which is a better measure of quality for our importance sampling distribution. If the sampling distribution R is well matched to the shape of the target distribution P, then the variance should be relatively small. Even if the distributions are well matched, b_m could still be very large in the "tails".

While PAM results are encouraging, we would really like to have *monotonicity with high probability* and not just *approximate* monotonicity. Let $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta};\boldsymbol{\theta}')$. By bounding $\Delta(\widehat{\boldsymbol{\theta}}) - \Delta(\boldsymbol{\theta}^*)$ instead of $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})$ we obtain the following stronger result guaranteeing a *probably monotonic* (PM) update. However, instead of restricting $A_{i,j}$ and $\pi_i \geq \theta_{\min}$, we need to make a stronger assumption about the values of $\bar{\alpha}_{t',t''}^{(m)}$ and $\bar{r}_{1,t'}^{(m)}$.

Theorem 2. *Let $\delta > 0$ be given and assume there exists $\lambda > 0$ such that $\bar{\alpha}^{(m)}_{t',t''} > \lambda$ and $\bar{r}^{(m)}_{1,t'} > \lambda$ for all t' and t''. If*

$$L_m \quad = \quad \frac{27 b_m}{\lambda} \left(\frac{2 \sum_{m=1}^{T} N_m + \Delta(\boldsymbol{\theta}^*)}{\Delta(\boldsymbol{\theta}^*)} \right)^2 \log \left(\frac{4 \sum_{m=1}^{T} N_m^2}{\delta} \right) \tag{70}$$

importance samples are used for the mth observation, then $\Delta(\widehat{\boldsymbol{\theta}}) \geq 0$ with probability at least $1 - \delta$.

Remark 5. Note the dependence on $\Delta(\boldsymbol{\theta}^*)$. By definition, $\Delta(\boldsymbol{\theta}^*) \geq 0$ at every iteration, and typically $\Delta(\boldsymbol{\theta}^*)$ is larger at earlier EM iterations and approaches zero as the algorithm converges. This dependence reflects the observation of Booth et al. mentioned earlier, that the number of importance samples ought to increase at each iteration.

Remark 6. The main assumption of Theorem 2 is that the sufficient statistics are bounded away from zero at each iteration. We motivate this assumption by observing that if the algorithm is properly initialized the sufficient statistics will not vanish in a finite number of iterations. The need for these assumptions arises out of the fact that $\Delta(\widehat{\boldsymbol{\theta}}) - \Delta(\boldsymbol{\theta}^*)$ contains terms involving $\log \widehat{A}_{i,j} - \log A^*_{i,j}$. If $\widehat{A}_{i,j}$ vanishes while $A^*_{i,j}$ is non-zero then $\Delta(\widehat{\boldsymbol{\theta}}) - \Delta(\boldsymbol{\theta}^*)$ diverges to $-\infty$ and we run into problems (if both $\widehat{A}_{i,j}$ and $A^*_{i,j}$ vanish at the same rate then there is no problem since $\log 1 = 0$).

Remark 7. The number of samples required for a PM increase in the marginal log-likelihood also grows polynomially in the size of the observations, in comparison to exponential computational complexity for exact E-step calculation. To gauge the quality of this result, consider the total computational complexity required for a PM update. If we define $\bar{N} = \frac{1}{T} \sum_{m=1}^{T} N_m$ to be the average observation size, then Theorem 2 dictates that, ignoring constant and log factors,

$$L^{\text{PM}}_{\text{tot}} = \sum_{m=1}^{T} L_m \quad \simeq \quad \sum_{m=1}^{T} \frac{\left(T \bar{N}\right)^2}{\lambda \Delta^2(\boldsymbol{\theta}^*)} \tag{71}$$

$$= \quad \frac{T^3 \bar{N}^2}{\lambda \Delta^2(\boldsymbol{\theta}^*)} \tag{72}$$

importance samples are required, in total, for a PM update. We do not know the precise values of the constants λ and $\Delta(\boldsymbol{\theta}^*)$ in practice, but the dependence on $T^3 \bar{N}^2$ is a useful guideline for how many importance samples to use.

Compared to the behavior required for a PAM update, $T^3 \bar{N}^2$ seems to be about as good as we could hope to do. If we approximate $N_m \approx \bar{N}$ then according to Theorem 1,

$$L^{\text{PAM}}_{\text{tot}} = \sum_{m=1}^{T} L_m \quad \simeq \quad \frac{T^3 \bar{N}^4 \, |\log \theta_{\min}|}{\epsilon^2} \tag{73}$$

importance samples are required to certify an ϵ-approximate PAM update. In general we choose ϵ and $\theta_{\min}$ very small in order to ensure an accurate, nearly monotonic update. It seems reasonable that $\lambda\Delta^2(\boldsymbol{\theta}^*)$ and $\epsilon^2/|\log\theta_{\min}|$ will be roughly on the same order.

Remark 8. Note that if we use different δ_k at each EM iteration, chosen such that $\sum_{k=1}^{\infty}\delta_k < \infty$, then by the Borel-Cantelli Lemma we can argue that $\Pr\left(\Delta(\widehat{\boldsymbol{\theta}}) < 0 \text{ i.o.}\right) = 0$. In other words, eventually all EM iterates result in a monotonic increase of the marginal log-likelihood.

Remark 9. In practical applications it may not be necessary to use Monte Carlo approximation for every observation. There may be a threshold, $N' > 0$, such that exact E-step computation is performed for observations with $N_m \leq N'$, and Monte Carlo approximation is used when $N_m > N'$. Accounting for this modification results in the following change to the expressions for L_m in each result.

- Let $\widetilde{T}$ denote the number of observations for which $N_m > N'$. Then each T in (69) can be replaced with $\widetilde{T}$.
- Let $\mathcal{M}$ denote the set of indices of observations with $N_m > N'$. Then both of the sums in (70) can be changed to sums over indices $m \in \mathcal{M}$ rather than over all $m = 1, \ldots, T$.

Remark 10. Finally, when the endpoints of each path are known (in particular, we need the destinations) then the following identity holds: $\sum_{t''=1}^{N_m}\bar{\alpha}_{t',t''}^{(m)} = 1$ for all t' not corresponding to the destination, and for t' corresponding to the destination, $\bar{\alpha}_{t',t''}^{(m)} = 0$ for all t''. Consequently, we can strengthen Theorem 2 in the following fashion. First, suppose that if t' does not correspond to the destination of the mth observation then we enforce $\widehat{\alpha}_{t',t''}^{(m)} \geq \alpha_{\min}$ for some $0 < \alpha_{\min} < N_m^{-1}$. Then it follows that $\Delta(\widehat{\boldsymbol{\theta}}) \geq 0$ with probability greater than $1-\delta$ if

$$L_m \quad = \quad \frac{27 b_m}{\alpha_{\min}}\left(\frac{\sum_{m=1}^{T} N_m + \Delta(\boldsymbol{\theta}^*)}{\Delta(\boldsymbol{\theta}^*)}\right)^2 \log\left(\frac{2\sum_{m=1}^{T} N_m^2}{\delta}\right)$$

importance samples are used for the mth observation. The main improvement here is that now $\alpha_{\min}$ is a parameter we control, and we no longer need to make assumptions about $\bar{\alpha}_{t',t''}^{(m)}$ being bounded away from zero.

In addition to demonstrating that the Monte Carlo EM algorithm has polynomial computational complexity, these bounds give a useful guideline for determining how many importance samples should be used. However, because they involve worst-case analysis, the numbers of samples dictated by these bounds tend to be on the conservative side. For example, in the Internet experiments described in Section 8, $T = 249$ and $\bar{N} = 17$. Theorem 2 suggests that roughly 72 million importance samples should be used per observation. However, in our experiments we find that the algorithm exhibits reasonable performance on this data set using as few as $2,000$ samples per observation. Of course, in practice, we do not have direct access to the b_m's, λ, or $\Delta(\boldsymbol{\theta}^*)$, so these bounds cannot be used as explicit guidelines.

7 Incorporating Prior Information

Additional side information about the Markov chain parameters $\mathbf{A}$ and $\boldsymbol{\pi}$ which we are estimating can easily be incorporated into the algorithm by applying independent Dirichlet priors to each row of the transition matrix and to the initial state distribution. Hence, we have

$$P[\boldsymbol{\pi}|\mathbf{u}] \propto \prod_{i=1}^{|S|} \pi_i^{u_i-1} \tag{74}$$

$$P[\mathbf{A}|\mathbf{v}] \propto \prod_{i=1}^{|S} \prod_{j=1}^{|S|} A_{i,j}^{v_{i,j}-1}, \tag{75}$$

where the parameters u_i and $v_{i,j}$ should be non-negative in order to have proper priors [2]. The larger that u_i is relative to the other $u_{i'}$, $i' \neq i$, the greater our prior belief that state i is an initial state rather than the others. Similarly, the larger $v_{i,j}$ relative to other $v_{i,j'}$ for $j' \neq j$, the more likely we expect, *a priori*, transitions from state i to state j relative to transitions from i to the other states.

Plugging (74) and (75) into our complete log-likelihood (16), we find that incorporating priors into the EM algorithm only results in a change to the M-step. In particular, instead of (26) we have

$$(\hat{\pi}_i)_{\text{new}} = \frac{u_i + \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} x_{t',i}^{(m)}}{\sum_{i=1}^{|S|} \left(u_i + \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} x_{t',i}^{(m)} \right)}, \tag{76}$$

and instead of (25) we have

$$\left(\widehat{A}_{i,j}\right)_{\text{new}} = \frac{v_{i,j} + \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \sum_{t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t-1,i}^{(m)} x_{t,j}^{(m)}}{\sum_{j=1}^{|S|} \left(v_{i,j} + \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \sum_{t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t-1,i}^{(m)} x_{t,j}^{(m)} \right)}. \tag{77}$$

Consider, for the moment, just the prior distribution on the initial state distribution. Applying $P[\boldsymbol{\pi}|\mathbf{u}]$ where $u_i = c > 1$ for all i will encourage all of the states to have some mass in the initial state distribution. On the other hand, setting $0 < c < 1$ in this example will have a shrinkage effect, encouraging all of the mass to go to one (or a few) of the states. Letting all u_i and $v_{i,j}$ tend uniformly to zero we are back to the original problem formulation with no additional information. We can push even more aggressively for a sparse solution by choosing negative parameters for the Dirichlet distributions as was done in [7] for Gaussian mixtures. This results in an improper prior and one must take care to threshold appropriately since π_i and $A_{i,j}$ are probabilities.

When negative Dirichlet parameters are allowed, the M-step updates become

$$(\hat{\pi}_i)_{\text{new}} = \frac{\left(u_i + \sum_{m=1}^{T}\sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)}\right)^+}{\sum_{i=1}^{|S|}\left(u_i + \sum_{m=1}^{T}\sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)}\right)^+} \tag{78}$$

$$\left(\widehat{A}_{i,j}\right)_{\text{new}} = \frac{\left(v_{i,j} + \sum_{m=1}^{T}\sum_{t'=1}^{N_m}\sum_{t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t-1,i}^{(m)} x_{t,j}^{(m)}\right)^+}{\sum_{j=1}^{|S|}\left(v_{i,j} + \sum_{m=1}^{T}\sum_{t'=1}^{N_m}\sum_{t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t-1,i}^{(m)} x_{t,j}^{(m)}\right)^+}. \tag{79}$$

where $(\cdot)^+$ retains the positive part of its argument and is equal to zero otherwise.

8 Experimental Results

In this section we evaluate the performance of our *network inference from co-occurrences* (NICO) algorithm on both simulated data sets, and on data gathered from the public Internet. In the results reported below we obtain a network reconstruction by first estimating an initial state distribution and probability transition matrix via the EM algorithm. Then we calculate the maximum likelihood ordering of elements in each path according to the inferred model, and use this ordering on each path to reconstruct a feasible network. The maximum likelihood optimization problem we are solving is not convex, and so the EM algorithm cannot be guaranteed to converge to a global solution. In general, there may also be multiple global maxima. Accordingly, we rerun the EM algorithm from multiple different random initializations and report on the collective results.

We compare the performance of our algorithm with that of the *Frequency Method* (FM), defined in [18] and mentioned in the introduction. The FM also reconstructs a network topology by estimating an order for the vertices in each transmission path. This method determines each path ordering independently by sorting the elements in the path according to a score computed from pair-wise co-occurrence frequencies involving the source and destination of the path. It is possible that within a particular path multiple vertices may receive identical FM scores, in which case the sorted order of those elements would be arbitrary (one could exchange elements with identical scores without violating the FM criteria). In fact, we observe this phenomena in many of our experiments. We resolve ties by choosing a random order for elements with identical scores, and then also perform multiple repetitions yielding different solutions.

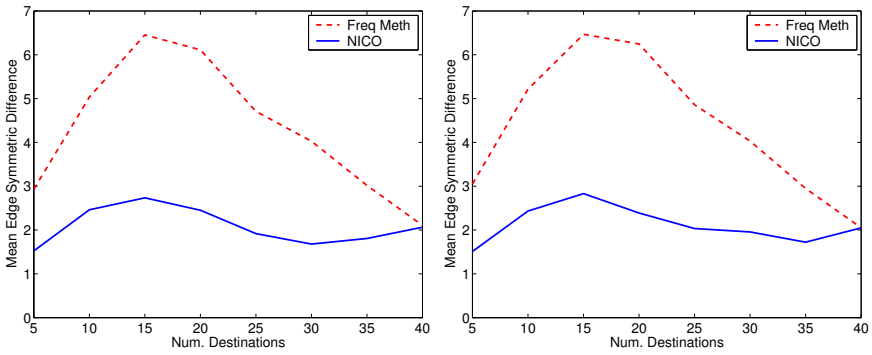
The quality of a network reconstruction is determined by a quantity we term the *edge symmetric difference* error. Because the nodes in the network have unique labels, the goal of any reconstruction scheme is to determine which nodes are connected by an edge. The edge symmetric difference error is defined as the sum of the number of false positives (edges appearing in the reconstructed network which do not exist in the true network) and the number of false negatives (edges in the true network not appearing in the reconstructed network).

8.1 Simulated Networks

In this section we study the performance of our algorithm on simulated data. A network is generated according to a random geometric graph model: 50 vertices are thrown at random in the unit square, and two vertices are connected with an edge if the Euclidean distance between them is less than or equal to $\sqrt{\log(50)/50}$. This threshold guarantees that the graph is connected with high probability. Groups of nodes are randomly chosen as sources and destinations, a subset of edges (chosen randomly) in the network are equipped with sensors, transmission paths are generated according to either a shortest path or random routing model, and then co-occurrence observations are formed from each path. The sources, destinations, and monitored edges then become the vertices in our reconstructed graph. In all the experiments reported in this section we vary the number of destination vertices between 5 and 40. This allows us to examine the effect of increasing the number of observed paths. Also, since there are only 50 vertices total in the graph, as the number of destinations increases there is more overlap between different co-occurrence observations. Each experiment is repeated over 100 randomly generated topologies, and 10 restarts of both the NICO and FM algorithms are run on each configuration. The exact E-step is used for observations with $N_m \leq 12$, and causal importance sampling is used for longer paths with 2000 Monte Carlo samples. The largest observation in any simulation has $N_m = 19$.

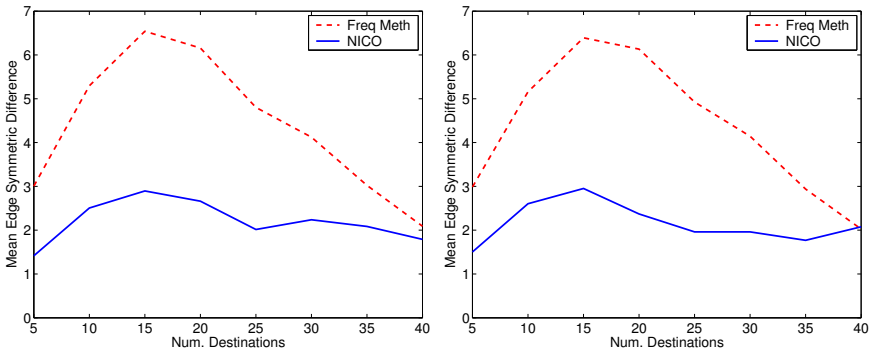
The first set of simulations reported use a shortest path routing policy (Dijkstra's algorithm) to generate transmission paths through the network. Figures 7(a-d) depict the average edge symmetric difference error for different levels of network coverage. At 25% coverage, one quarter of all the edges in the original network (chosen at random) are capable of sensing transmissions. At 50% coverage, half of the edges sense, and so on. Each data point shown is the average over 10 restarts of each algorithm on 100 different topologies. As a point of reference, at 100% coverage the typical network contains roughly 250 edges. Thus, although performance is consistent across the different levels of coverage in terms of absolute error, both algorithms actually perform worse as the coverage decreases, relative to the number of edges in the network. For a fixed level of coverage, there seems to be a general trend in that the performance of both algorithms is the worst for a moderate-to-low number of destinations (10-20), and performance improves at either extreme.

When there are very few destinations, the target network closely resembles a tree which might explain why both algorithms perform well. In tree networks the relative frequencies of co-occurrence accurately reflect the network distance of each internal vertex from the sources and destinations. At the other extreme, when there are 40 destinations the FM performs essentially as well as NICO. A possible explanation for this might be that when nearly all vertices in the network are available as destinations (recall, there are only 50 vertices total in the generating network and 5 of these act as sources), there is sufficient overlap among all of the observed co-occurrences so that pairwise co-occurrence frequencies again accurately reflect the positions of vertices within the network. Also, one reason the FM is not performing as well in this simulation as one might expect (based on the shortest path routing policy) is that, even though the routes from each source to all destinations form a tree, when the routes from different sources are combined to form the network some of this structure is lost and the



(a) 100% Coverage

(b) 75% Coverage



(c) 50% Coverage

(d) 25% Coverage

Figure 7: Average edge symmetric difference error for simulated networks with shortest path routing. The coverage level indicates what percentage of edges in the network sense transmissions. These results reflect the average over 10 random initializations of each algorithm on 100 different topologies.

tree-based model is violated. In general, the frequency method seems to be much more sensitive to the amount of data available, whereas NICO offers more consistent average performance across various settings.

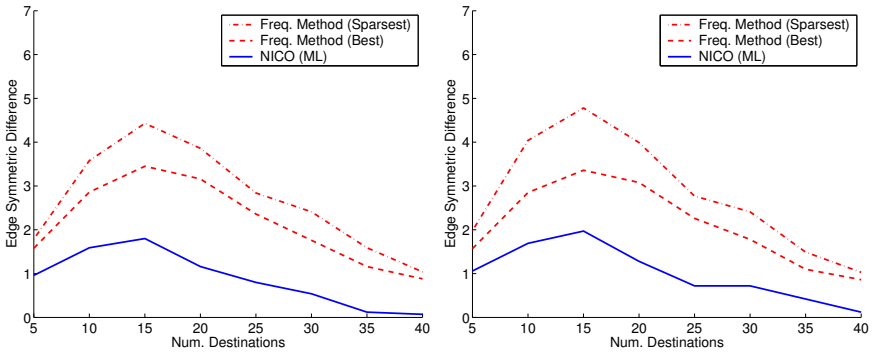
We perform multiple restarts because of the possibility that the EM algorithm will get trapped at a local maximum. For each random initialization we potentially will compute a different candidate solution. Once we have executed multiple restarts from random initializations, we decide which of the reported solutions is superior by calculating the marginal log-likelihood of the data for each solution. Figure 8 depicts the edge symmetric difference error for NICO when the most likely candidate is used for reconstruction. In this figure performance is averaged over 100 topologies, but not over restarts. Unfortunately, of the multiple solutions returned by the FM, there is no obvious way to prefer one over the other. A potential heuristic might be to choose the sparsest reconstruction candidate, however this doesn't always result in the best performance. In Figure 8 we display the error resulting from using both the sparsest candidate, and by clairvoyantly choosing the best FM candidate. For this simulation, the most likely NICO candidate always also corresponded to one with best edge symmetric difference error. Clearly, using the sparsest FM solution as a heuristic for picking one of the candidate solutions does consistently better than just choosing one at random (compare with the mean FM performance in the previous Figure), however it still is not doing as well as possible with the FM. Moreover, the most likely NICO solution is significantly more accurate than average.

We have also repeated the above experimental setup, but using a random route generation scheme rather than shortest path routes. A random route for a given source destination pair is found by first generating a random weight matrix for the edges in the graph, and then running the shortest path algorithm, taking these weights into account. Shortest path routes generated in the first set of simulations correspond to each edge having the same weight. By varying the weight matrix, the only consistent characteristics across routes are those arising from the underlying topological structure of the graph. Figure 9 shows results for the sparsest and clairvoyant best FM candidates and the maximum likelihood NICO candidate for 10 random initializations on 100 different topologies. As might be expected, NICO handles random routes much better than the FM. At the extreme numbers of destinations, performance of the FM is relatively unchanged, however FM performance degrades for intermediate values, as compared to the shortest path simulation results.

8.2 Internet Data

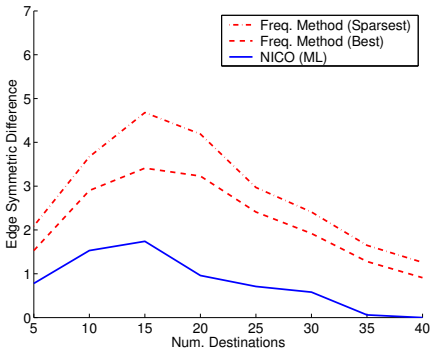
We have also studied the performance of our algorithm on co-occurrence observations gathered from the Internet. Using `traceroute` we have collected data describing roughly 250 router-level paths. Our motivation for using this type of data is two-fold. First, `traceroute` allows us to measure the true order of elements in each path so that we have a ground truth to validate our results against. Second, and more importantly, the data comes from a real network where, presumably, paths are not generated according to a first-order Markov model. This allows us to gauge the robustness of the proposed model and to evaluate how well it generalizes to realistic scenarios.

The data used in this experiment were collected on October 12, 2005.

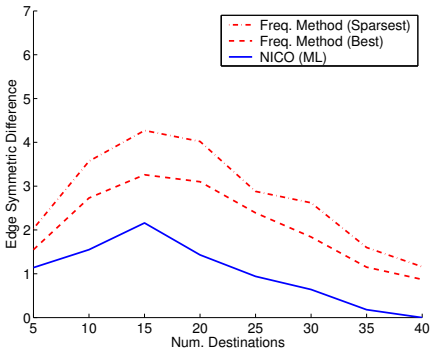


(a) 100% Coverage

(b) 75% Coverage

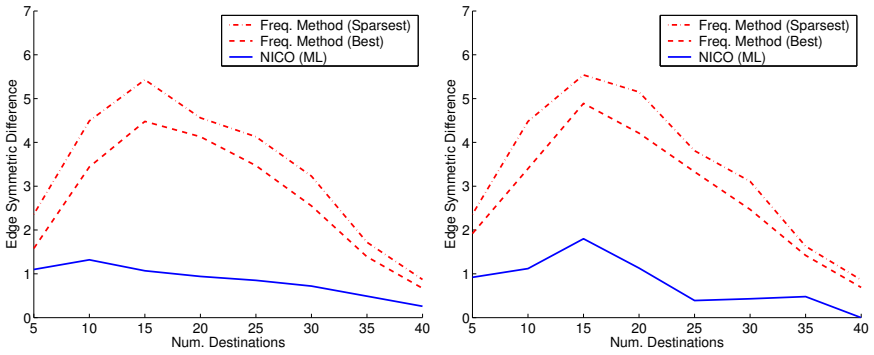


(c) 50% Coverage



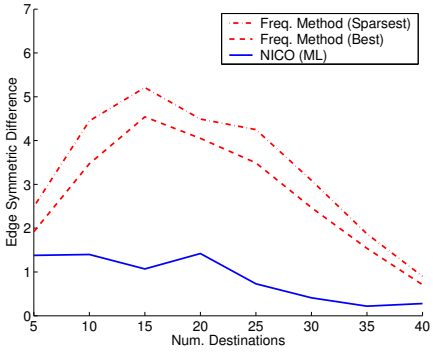
(d) 25% Coverage

Figure 8: Edge symmetric difference error for simulated networks with shortest path routing at different coverage levels. Results are averaged over 100 different topologies. Of the 10 candidate solutions corresponding to random initializations of each algorithm, the sparsest and (clairvoyant) best FM solution and most likely NICO solution are used.

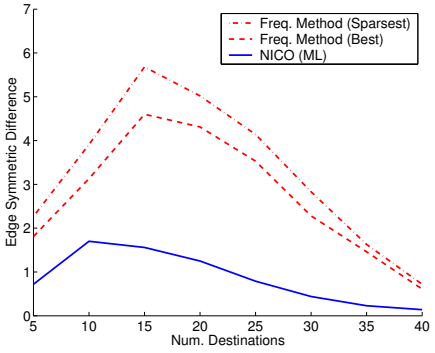


(a) 100% Coverage

(b) 75% Coverage



(c) 50% Coverage



(d) 25% Coverage

Figure 9: Edge symmetric difference error for simulated networks with randomly generated transmission paths. Of the 10 candidate solutions corresponding to random initializations of each algorithm, the sparsest and (clairvoyant) best FM solution and most likely NICO solution are used.

`Traceroute` probes were initiated from three sources located at the University of Wisconsin-Madison, the *Instituto Superior Técnico* in Lisbon, and at Rice University, and probes were transmitted to 83 web servers affiliated with a mixture of corporations, universities, and governments around the world. The shortest path was eight hops long, between two machines located in Madison, and the longest path was 27 hops long, stretching from Lisbon to a site in Australia. The exact E-step is used to compute $\bar{\alpha}$ for paths of up to 9 hops. For paths longer than 9 hops we use the causal importance sampling described in Section 5.1 to approximate the E-step calculations. The ground truth topology derived from the ordered routes is depicted in Figure 10. The network contains a total of 1105 vertices and 1317 edges.

A Matlab implementation of the EM algorithm typically runs for roughly one hour on this data set, converging after 19 iterations. A faster C implementation performs the same computation in 10 minutes. The frequency method requires two passes over the data: one to collect pair-wise co-occurrence frequency statistics, and a second to compute orders for each path. A Matlab implementation runs in under a minute for this data set.

The minimum, median, and maximum edge symmetric difference errors are shown in Figure 11. Both algorithms have seemingly high error rates, as there are roughly 1300 links in the true network. However keep in mind that each reconstruction scheme is attempting to fill in the entries of a roughly 1100×1100 matrix. For 50 networks constructed by choosing a random order for the elements of each path, the average edge symmetric difference error was 4300, so both algorithms are indeed doing considerably better than random guessing. Moreover, performance of the proposed NICO approach is noticeably better than that for the frequency method; the NICO average error is better than that of the best FM reconstruction, and the worst case NICO reconstruction is on par with the average FM performance. We also note that the number of false positives and false negatives in a reconstruction using either scheme tend to be roughly equal (each constituting half of the edge symmetric difference error). From a detection-theoretic standpoint, the Type-I and Type-II errors are more or less balanced.

Figure 12 shows statistics for the number of edges in the reconstructed networks. There is an interesting correlation between the number of edges and accuracy of reconstruction in this example. As seen above, the typical NICO reconstruction is more accurate, in terms of edge errors, than a FM reconstruction. NICO also consistently returns a sparser estimate. The median number of links in a NICO reconstruction was 1329, whereas the median number of links in a FM reconstruction was 1426. There are 1317 edges in the true network, so it seems that the reconstructions generated using NICO more accurately reflect the level of complexity in the true network.

The marginal log-likelihood values for each of the 50 NICO estimates are depicted in Figure 13. The marginal log-likelihood, given by (6), is the cost function being optimized by the EM algorithm. Often, for non-convex problems such as ours where there are local maxima, multiple runs of the EM algorithm will be performed with different random initializations. Then the solution with the maximum marginal likelihood will be used. However, in contrast to the experiments with simulated data reported above, for this example there is not an exact correlation between higher marginal likelihood values and lower edge symmetric difference error. The topology with the highest likelihood value results in an edge symmetric difference error of 627. This is better than

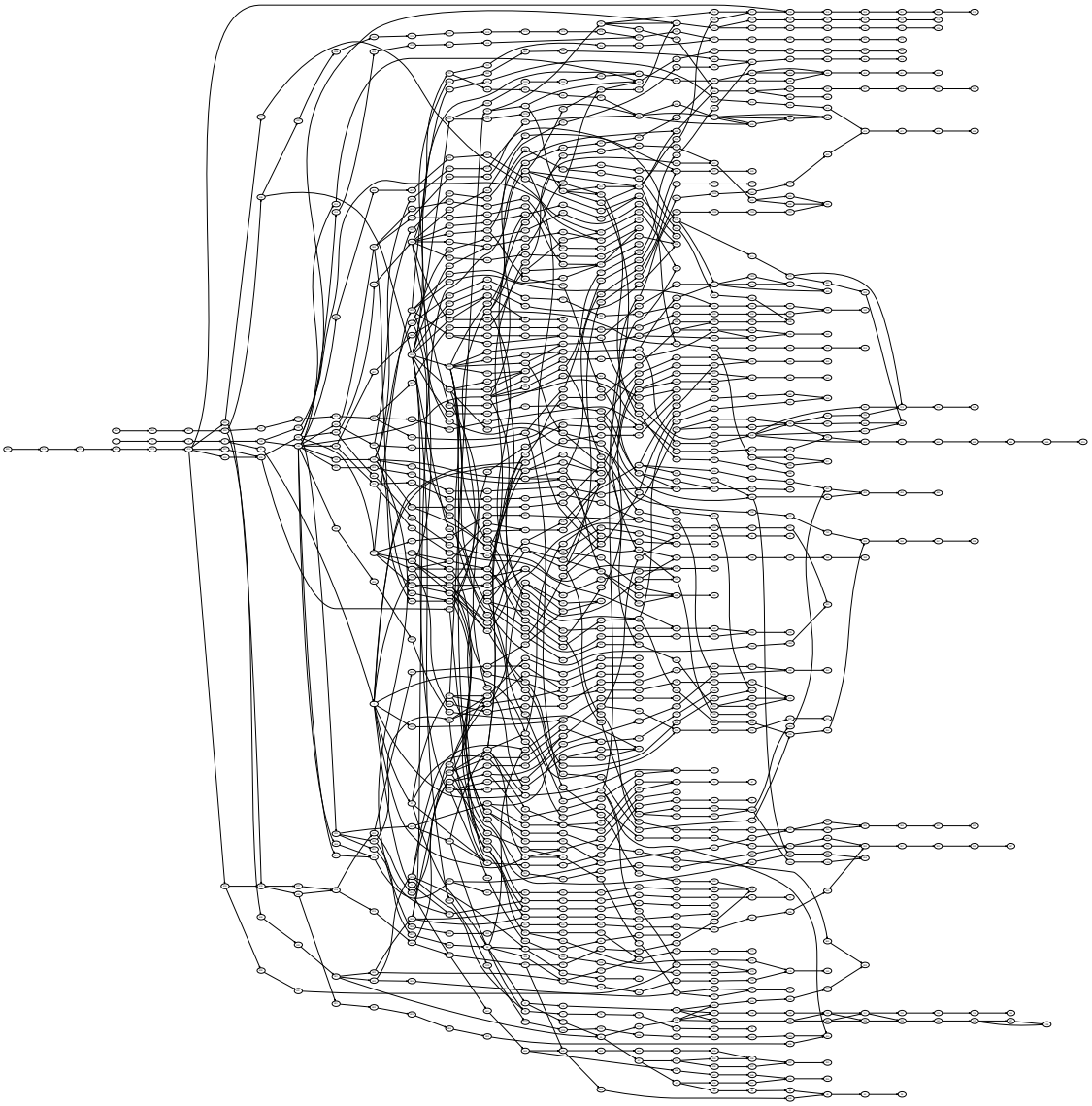


Figure 10: Internet topology obtained using `traceroute` from three sources to 83 destinations around the world.

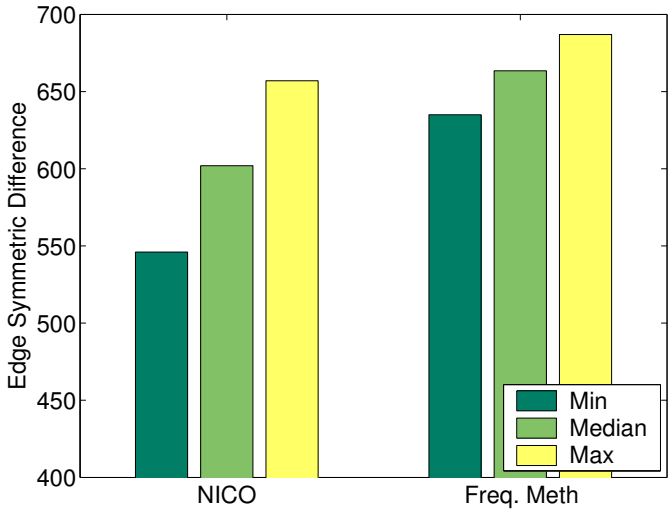


Figure 11: Edge symmetric difference error comparison of NICO and FM on Internet data. The reported values come from 50 random initializations of each algorithm.

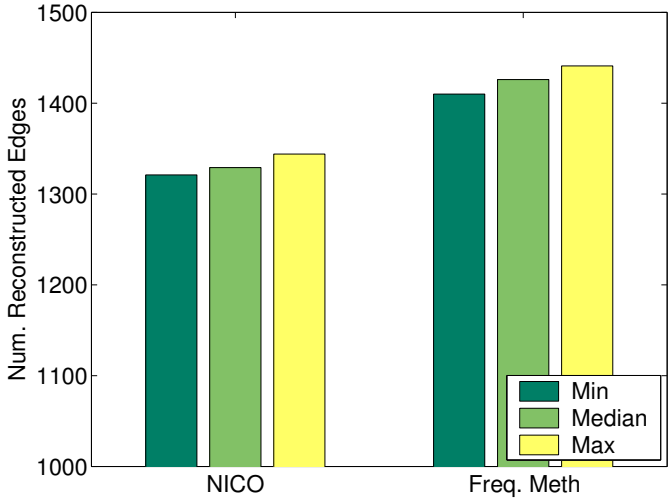


Figure 12: Number of edges in networks reconstructed using each method. The median number of edges per reconstruction is 1329 for NICO and 1426 for FM. The true network has 1317 edges, and so it appears that NICO does a better job of capturing the complexity of the true network.

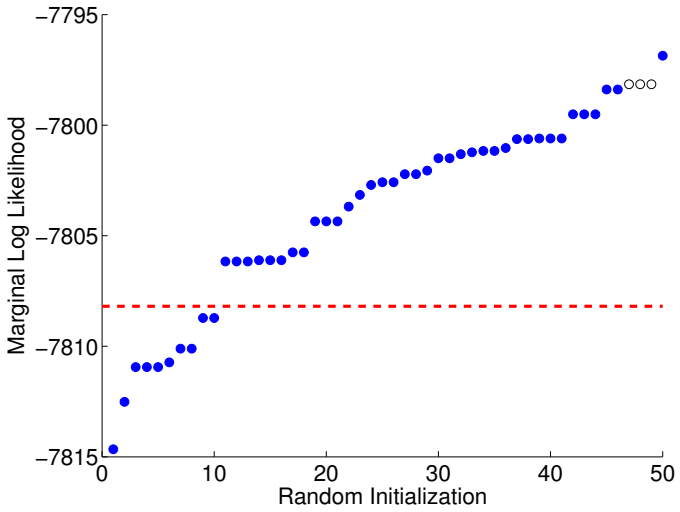


Figure 13: Marginal log likelihood values for different random initializations of NICO, sorted in ascending order. The three hollow circles correspond to the solutions which achieve the lowest edge symmetric difference error of all NICO trials. The red line shows the marginal log likelihood value computed using the true path orders to estimate a Markov transition matrix. Many candidate solutions have higher marginal log-likelihood than the true topology, suggesting that our generative model may not be the best match for Internet topology data.

the clairvoyant best FM error, but only average for NICO. The three repetitions which returned a topology with the lowest symmetric difference error had the next highest likelihood value, as indicated by the three hollow circles in the figure. The dashed line shows the likelihood value based on a transition matrix estimated using the true path orders as measured by `traceroute`. Notice that a majority of the candidate solutions returned by NICO have a higher marginal likelihood than the true topology. This suggests that our generative model may not be the best match for Internet topology data. Still the overall performance of our algorithm is encouraging.

9 Conclusion and Discussion

This paper presents a novel approach to network reconstruction from co-occurrence observations. A co-occurrence observation reflects which vertices are activated by a particular transmission through the network, but not the order in which they are activated. We model transmission paths as i.i.d. random walks on the underlying graph structure. The parameters for this model are the initial state distribution and transition matrix of a first-order Markov chain governing the random walk. Co-occurrence observations are then modelled as samples of the random walk, subjected to a random permutation which accounts for the fact that we do not observe the activation

order. Treating the random permutations as latent variables, we derive an *expectation-maximization* (EM) algorithm for efficiently computing maximum likelihood estimates of the Markov chain parameters. Because the marginal log-likelihood is not convex (in general it is multi-modal), we only guarantee that the EM algorithm will converge to a local maximum. Multiple restarts from different initializations are typically used, and of these solutions, the one with the largest marginal log-likelihood is taken as the best. Our algorithm is easily modified to compute the maximum *a posteriori* estimates, allowing a user to incorporate additional side information in a natural, Bayesian fashion.

The complexity of the EM algorithm is dominated by the E-step calculation which requires $O(N_m!)$ operations for the mth observation, where N_m is the number of vertices in that observation. If N_m is larger than 10 or 15, exact computation of the E-step may not be tractable. For such situations we describe faster approximation methods based on importance sampling and Monte Carlo techniques. Care must be taken to use enough samples, or else the EM algorithm will be swamped by Monte Carlo error and will not converge. On the other hand, one would like to avoid using too many samples and incurring greater complexity than necessary. We derive concentration-style bounds, characterizing the quality of the Monte Carlo approximation in terms of the problem dimensions and the number of samples used. Based on these bounds, we can guarantee that the EM algorithm will converge with high probability using a number of samples which depends polynomially on N_m, as opposed to exponential dependence required for exact E-step calculation.

To obtain a network reconstruction, we determine the most likely order for each co-occurrence observation according to the Markov chain parameter estimates, and then insert edges in the graph based on these ordered transmission paths. This procedure always produces a feasible reconstruction (one which is consistent with the observations). The parameter estimates can also be used to assign likelihoods to different permutations of a co-occurrence observation, guiding an expert to alternative reconstructions. The parameter estimates may be also useful for other tasks such as predicting new, unobserved, paths through the network [10]. Alternatively, one could analyze properties of an ensemble of solutions, obtained by running the EM algorithm from different initializations, and then posit a new set of experiments to be conducted based on this analysis.

The transition matrix parameter $A_{i,j}$ can be interpreted as estimates of the probability a transmission will be passed from vertex i to j, conditioned on the path passing reaching i; that is, $A_{i,j} = P[Z_{k+1} = j | Z_k = i]$. In particular, they *are not* estimates of the probability of a link existing from i to j. Since $\mathbf{A}$ is a stochastic matrix, each row must sum to 1, and so if vertex i is connected to many other nodes then the unit mass is being spread over more entries. We can obtain joint probabilities, $P[Z_k = i, Z_{k+1} = j]$, via Bayes theorem,

$$P[Z_{k+1} = j | Z_k = i] = \frac{P[Z_k = i, Z_{k+1} = j]}{P[Z_k = i]},$$

where $P[Z_k = i]$ is the stationary distribution of the chain (not necessarily equal to the initial state distribution). These joint probabilities (appropriately scaled versions of the transition matrix entries) more accurately reflect the likelihood of there being an edge from i to j, based on our estimates.

Our future work involves extending and generalizing both algorithmic and theoretical aspects of this work. Co-occurrence observations naturally arise from transmission *paths* in communication network applications and, to a degree, in biological, social, and brain networks as well. However the physical mechanisms driving interactions in the latter three applications may also correspond to more general connected subgraph structures such as trees or directed acyclic graphs. Extending our methods in this fashion is easily accomplished in theory, however the computational complexity is significantly amplified when more general structures are considered. In this paper we have also restricted our attention to noise-free observations. We are also extending our algorithm to handle the case where observations reflect a soft probability that a given vertex occurred in the path rather than hard, "active" or "not active", binary observations. This extension is relevant in many applications including the inference of signal transduction networks (in systems biology) where co-occurrence observations are themselves the result of inference procedures run on experimental data.

A Proofs of Monotonicity Theorems

A.1 Proof of Theorem 1

There are two main steps to the proof of Theorem 1. First, we derive a concentration inequality for the importance sample approximations, $\widehat{\alpha}^{(m)}_{t',t''}$ and $\widehat{r}^{(m)}_{1,t'}$. Then we use these concentration inequalities to construct a bound for $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\boldsymbol{\theta})$.

Recall the expressions (35) and (34) of importance sample approximations calculated in the Monte Carlo E-step. More generally, we will consider self-normalizing sums of the form

$$\widehat{\mu}_L \quad = \quad \frac{\sum_{i=1}^{L} Z(\mathbf{r}^i) X(\mathbf{r}^i)}{\sum_{i=1}^{L} Z(\mathbf{r}^i)}, \tag{80}$$

where $Z : \Psi_N \rightarrow [0, b]$ correspond to the importance sample weights, and $X : \Psi_N \rightarrow \{0, 1\}$ indicates whether or not the ith importance sample exhibits the event of interest. For example, if we are approximating $\bar{r}_{1,t'}$ then $X(\mathbf{r}^i) = r^i_{1,t'}$. Denoting by P the target distribution and by R the importance sampling distribution, we have $Z(\mathbf{r}^i) = P[\mathbf{r}^i|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]/R[\mathbf{r}^i|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$. Now, we are trying to approximate

$$\mu \quad = \quad \sum_{\mathbf{r} \in \Psi_N} X(\mathbf{r}) P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]. \tag{81}$$

Note that $\mathbb{E}[\widehat{\mu}_L] \neq \mu$, so we cannot directly apply standard concentration results such as Hoeffding's inequality or McDiarmid's bounded-differences inequality. To see why this is true, consider the case $L = 1$:

$$\mathbb{E}\left[\frac{Z(\mathbf{r}^1)X(\mathbf{r}^1)}{Z(\mathbf{r}^1)}\right] \quad = \quad \sum_{\mathbf{r} \in \Psi_N} X(\mathbf{r}) R[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}] \tag{82}$$

$$\neq \quad \sum_{\mathbf{r} \in \Psi_N} X(\mathbf{r}) P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]. \tag{83}$$

We can, however, show that the approximation $\widehat{\mu}_L$ is asymptotically consistent. Observe that

$$\begin{aligned}
\mathbb{E}[Z(\mathbf{r}^i)] &= \sum_{\mathbf{r}\in\Psi_N} \frac{P[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]}{R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]} R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}] && (84)\\
&= \sum_{\mathbf{r}\in\Psi_N} P[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}] && (85)\\
&= 1, && (86)
\end{aligned}$$

since P is a probability distribution on Ψ_N, and

$$\begin{aligned}
\mathbb{E}[Z(\mathbf{r}^i)X(\mathbf{r}^i)] &= \sum_{\mathbf{r}\in\Psi_N} \frac{P[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]}{R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]} X(\mathbf{r}) R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}] && (87)\\
&= \sum_{\mathbf{r}\in\Psi_N} X(\mathbf{r}) P[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}] && (88)\\
&= \mu. && (89)
\end{aligned}$$

Then, by a strong law of large numbers argument, it follows that $\widehat{\mu}_L \to \mu$ as $L \to \infty$.

We have the following finite-sample concentration inequality demonstrating that the approximation error, $\widehat{\mu}_N - \mu$ decays exponentially with the number of importance samples, L.

Proposition 1. *Let $\{(X_i, Z_i)\}$ be a sequence of independent and identically distributed random variables with $X_i \in \{0,1\}$ and $Z_i \in [0,b]$. Assume that $\mathbb{E}[Z_i] = 1$ and $\mathbb{E}[Z_i X_i] = \mu$, and define*

$$\widehat{\mu}_L = \frac{\sum_{i=1}^{L} Z_i X_i}{\sum_{i=1}^{L} Z_i}. \tag{90}$$

Then for any $\epsilon > 0$,

$$\Pr\left(\widehat{\mu}_L - \mu \geq \epsilon\right) \leq 2\exp\left\{\frac{-2L}{b^2}\left(\frac{\epsilon}{1+\mu+\epsilon}\right)^2\right\}. \tag{91}$$

Proof. We have $Z_i \in [0,b]$, and $X_i \in \{0,1\}$ so $Z_i X_i \in [0,b]$ also. Then by Hoeffding's inequality [8], for any $t > 0$,

$$\Pr\left(\sum_{i=1}^{L} Z_i X_i - L\mu \geq Lt\right) \leq e^{-2Lt^2/b^2}, \tag{92}$$

and for any $t > 0$,

$$\Pr\left(\sum_{i=1}^{L} Z_i - L \leq -Lt\right) \leq e^{-2Lt^2/b^2}. \tag{93}$$

Define the event,

$$E_t = \left\{\sum_{i=1}^{L} Z_i X_i - L\mu \geq Lt\right\} \bigcup \left\{\sum_{i=1}^{L} Z_i - L \leq -Lt\right\}. \quad (94)$$

Then by the union bound, $\Pr(E_t) \leq 2e^{-2Lt^2/b^2}$ for any $t > 0$. Next consider the complement,

$$\bar{E}_t = \left\{\sum_{i=1}^{L} Z_i X_i - L\mu < Lt\right\} \bigcap \left\{\sum_{i=1}^{L} Z_i > L(1-t)\right\}. \quad (95)$$

The event $\bar{E}_t$ implies that

$$\widehat{\mu}_L - \mu = \frac{\sum_{i=1}^{L} Z_i X_i - L\mu}{\sum_{i=1}^{L} Z_i} + \frac{L\mu}{\sum_{i=1}^{L} Z_i} - \mu \quad (96)$$

$$< \frac{Lt}{L(1-t)} + \frac{L\mu}{L(1-t)} - \mu \quad (97)$$

$$= \frac{t(1+\mu)}{1-t}. \quad (98)$$

It follows that

$$\left\{\widehat{\mu}_L - \mu \geq \frac{t(1+\mu)}{1-t}\right\} \subseteq E_t, \quad (99)$$

and so

$$\Pr\left(\widehat{\mu}_L - \mu \geq \frac{t(1+\mu)}{1-t}\right) \leq \Pr(E_t) \quad (100)$$

$$\leq 2e^{-2Lt^2/b^2}. \quad (101)$$

Set $\epsilon = t(1+\mu)/(1-t)$ to obtain the desired result. □

Before proceeding we slightly weaken the result of Proposition 1 to simplify computations below. We note that this relaxation only effects the constants and does not change the rate of convergence. Since $\widehat{\mu}_L \leq 1$ and $\mu \geq 0$, $\widehat{\mu}_L - \mu \leq 1$ with probability 1. That is, if $\epsilon > 1 - \mu$, then

$$\Pr\left(\widehat{\mu}_L - \mu \geq \epsilon\right) = 0, \quad (102)$$

and Proposition 1 holds trivially. Thus, it suffices to consider $\mu + \epsilon \leq 1$ in which case $1 + \mu + \epsilon \leq 2$. Let

$$\delta = 2\exp\left\{\frac{-L\epsilon^2}{2b^2}\right\}. \quad (103)$$

Then with probability at least $1 - \delta$,

$$\widehat{\mu}_L - \mu < \sqrt{\frac{2b^2 \log\frac{2}{\delta}}{L}}. \quad (104)$$

Next, we will use this result to construct a bound for $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})$.

Consider the collection of Monte Carlo sufficient statistics for the mth observation, $\{\widehat{\alpha}^{(m)}_{t',t''}\}$ and $\{\widehat{r}^{(m)}_{1,t'}\}$. By assumption, we have

$$b_m \;=\; \max_{\mathbf{r}\in\Psi_{N_m}} \frac{P[\mathbf{r}|\mathbf{x}^{(m)},\mathbf{A}',\boldsymbol{\pi}']}{R[\mathbf{r}|\mathbf{x}^{(m)},\mathbf{A}',\boldsymbol{\pi}']}, \tag{105}$$

which exists because the collection Ψ_{N_m} is finite, and P is absolutely continuous with respect to R by assumption. Define

$$\begin{aligned} B^m_{\delta',L_m} \;=\; & \left(\bigcup_{t',t''}\left\{\widehat{\alpha}^{(m)}_{t',t''} - \bar{\alpha}^{(m)}_{t',t''} \geq \sqrt{\frac{2b_m^2\log\frac{2}{\delta'}}{L_m}}\right\}\right) \\ & \bigcup\left(\bigcup_{t'}\left\{\widehat{r}^{(m)}_{1,t'} - \bar{r}^{(m)}_{1,t'} \geq \sqrt{\frac{2b_m^2\log\frac{2}{\delta'}}{L_m}}\right\}\right), \end{aligned} \tag{106}$$

which is a union of $2\binom{N_m}{2} + N_m = N_m^2$ events, each of which holds with probability at most δ'. By the union bound it follows that $\Pr(B^m_{\delta',L_m}) \leq N_m^2\delta'$. Next, define

$$C^m_{\delta',L_m} = \left\{\sum_{t',t''=1}^{N_m}\left(\widehat{\alpha}^{(m)}_{t',t''} - \bar{\alpha}^{(m)}_{t',t''}\right) + \sum_{t'=1}^{N_m}\left(\widehat{r}^{(m)}_{1,t'} - \bar{r}^{(m)}_{1,t'}\right) \geq N_m^2\sqrt{\frac{2b_m^2\log\frac{2}{\delta'}}{L_m}}\right\}. \tag{107}$$

Observe that $\bar{B}^m_{\delta',L_m}$ implies $\bar{C}^m_{\delta',L_m}$. Therefore, $C^m_{\delta',L_m} \subseteq B^m_{\delta',L_m}$ and $\Pr(C^m_{\delta',L_m}) \leq \Pr(B^m_{\delta',L_m}) \leq N_m^2\delta'$. Let $\delta'' = N_m^2\delta'$. Also let $L > 0$ be a value to be determined later, and for each $m = 1,\ldots,T$, set

$$L_m \;=\; \frac{2LN_m^4 b_m^2\log\frac{2N_m^2}{\delta''}}{\log\frac{1}{\delta''}}, \tag{108}$$

so that

$$N_m^2\sqrt{\frac{2b_m^2\log\frac{2}{\delta'}}{L_m}} \;=\; N_m^2\sqrt{\frac{2b_m^2\log\frac{2N_m^2}{\delta''}}{L_m}} \;=\; \sqrt{\frac{\log\frac{1}{\delta''}}{L}}. \tag{109}$$

Then with probability greater than $1-\delta''$,

$$\sum_{t',t''=1}^{N_m}\left(\widehat{\alpha}^{(m)}_{t',t''} - \bar{\alpha}^{(m)}_{t',t''}\right) + \sum_{t'=1}^{N_m}\left(\widehat{r}^{(m)}_{1,t'} - \bar{r}^{(m)}_{1,t'}\right) \;<\; \sqrt{\frac{\log\frac{1}{\delta''}}{L}}. \tag{110}$$

By the independence of importance sample estimates for different observations, with probability greater than $(1-\delta'')^T$,

$$\bigcap_{m=1}^{T}\left\{\sum_{t',t''=1}^{N_m}\left(\widehat{\alpha}^{(m)}_{t',t''} - \bar{\alpha}^{(m)}_{t',t''}\right) + \sum_{t'=1}^{N_m}\left(\widehat{r}^{(m)}_{1,t'} - \bar{r}^{(m)}_{1,t'}\right) < \sqrt{\frac{\log\frac{1}{\delta''}}{L}}\right\}, \tag{111}$$

which implies

$$\sum_{m=1}^{T}\sum_{t',t''=1}^{N_m}\left(\widehat{\alpha}_{t',t''}^{(m)}-\bar{\alpha}_{t',t''}^{(m)}\right)+\sum_{m=1}^{T}\sum_{t'=1}^{N_m}\left(\widehat{r}_{1,t'}^{(m)}-\bar{r}_{1,t'}^{(m)}\right) \quad < \quad T\sqrt{\frac{\log\frac{1}{\delta''}}{L}}. \tag{112}$$

Recall that the variables $x_{t',i}^{(m)}$ are indicators such that for fixed t', $x_{t',i}^{(m)}=1$ for exactly one value of i, and $x_{t',j}^{(m)}=0$ for all other $j\neq i$. Thus, $\sum_{i,j=1}^{|S|}x_{t'',i}^{(m)}x_{t',j}^{(m)}=1$ and $\sum_{i=1}^{|S|}x_{t',i}^{(m)}=1$, and so with probability greater than $(1-\delta'')^T$,

$$\begin{aligned}
&T\sqrt{\frac{\log\frac{1}{\delta''}}{L}}\\
&> \sum_{m=1}^{T}\sum_{t',t''=1}^{N_m}\left(\widehat{\alpha}_{t',t''}^{(m)}-\bar{\alpha}_{t',t''}^{(m)}\right)+\sum_{m=1}^{T}\sum_{t'=1}^{N_m}\left(\widehat{r}_{1,t'}^{(m)}-\bar{r}_{1,t'}^{(m)}\right)\\
&= \sum_{m=1}^{T}\sum_{t',t''=1}^{N_m}\left(\widehat{\alpha}_{t',t''}^{(m)}-\bar{\alpha}_{t',t''}^{(m)}\right)\sum_{i,j=1}^{|S|}x_{t'',i}^{(m)}x_{t',j}^{(m)}+\sum_{m=1}^{T}\sum_{t'=1}^{N_m}\left(\widehat{r}_{1,t'}^{(m)}-\bar{r}_{1,t'}^{(m)}\right)\sum_{i=1}^{|S|}x_{t',i}^{(m)}\\
&= \sum_{m=1}^{T}\sum_{i,j=1}^{|S|}\sum_{t',t''=1}^{N_m}\left(\widehat{\alpha}_{t',t''}^{(m)}-\bar{\alpha}_{t',t''}^{(m)}\right)x_{t'',i}^{(m)}x_{t',j}^{(m)}+\sum_{m=1}^{T}\sum_{i=1}^{|S|}\sum_{t'=1}^{N_m}\left(\widehat{r}_{1,t'}^{(m)}-\bar{r}_{1,t'}^{(m)}\right)x_{t',i}^{(m)}.
\end{aligned}$$

Finally, set $1-\delta=(1-\delta'')^T$ and multiply through by $|\log\theta_{\min}|>0$. Then with probability greater than $1-\delta$,

$$\begin{aligned}
&T\,|\log\theta_{\min}|\sqrt{\frac{\log\frac{1}{1-(1-\delta)^{1/T}}}{L}}\\
&\quad> \sum_{m=1}^{T}\sum_{i,j=1}^{|S|}\sum_{t',t''=1}^{N_m}\left(\widehat{\alpha}_{t',t''}^{(m)}-\bar{\alpha}_{t',t''}^{(m)}\right)x_{t'',i}^{(m)}x_{t',j}^{(m)}\,|\log\theta_{\min}|\\
&\qquad+\sum_{m=1}^{T}\sum_{i=1}^{|S|}\sum_{t'=1}^{N_m}\left(\widehat{r}_{1,t'}^{(m)}-\bar{r}_{1,t'}^{(m)}\right)x_{t',i}^{(m)}\,|\log\theta_{\min}|.
\end{aligned} \tag{113}$$

To complete the proof, observe that

$$\begin{aligned}
&\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) \\
&= \left(\sum_{m=1}^{T} \sum_{i,j=1}^{|S|} \sum_{t',t''=1}^{N_m} \widehat{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)} \left(\log \widehat{A}_{i,j} - \log A'_{i,j} \right) \right. \\
&\quad \left. + \sum_{m=1}^{T} \sum_{i=1}^{|S|} \sum_{t'=1}^{N_m} \widehat{r}_{1,t'}^{(m)} x_{t',i}^{(m)} \left(\log \widehat{\pi}_i - \log \pi'_i \right) \right) \\
&\quad - \left(\sum_{m=1}^{T} \sum_{i,j=1}^{|S|} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)} \left(\log \widehat{A}_{i,j} - \log A'_{i,j} \right) \right. \\
&\quad \left. + \sum_{m=1}^{T} \sum_{i=1}^{|S|} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} x_{t',i}^{(m)} \left(\log \widehat{\pi}_i - \log \pi'_i \right) \right) \qquad (114) \\
&= \sum_{m=1}^{T} \sum_{i,j=1}^{|S|} \sum_{t',t''}^{N_m} \left(\widehat{\alpha}_{t',t''}^{(m)} - \bar{\alpha}_{t',t''}^{(m)} \right) x_{t'',i}^{(m)} x_{t',j}^{(m)} \left(\log \widehat{A}_{i,j} - \log A'_{i,j} \right) \\
&\quad \sum_{m=1}^{T} \sum_{i=1}^{|S|} \sum_{t'=1}^{N_m} \left(\widehat{r}_{1,t'}^{(m)} - \bar{r}_{1,t'}^{(m)} \right) x_{t',i}^{(m)} \left(\log \widehat{\pi}_i - \log \pi'_i \right). \qquad (115)
\end{aligned}$$

Since $\theta_{\min} \leq \widehat{A}_{i,j}, A'_{i,j} \leq 1$ by assumption, for $i, j = 1, \ldots, |S|$,

$$\begin{aligned}
\log \widehat{A}_{i,j} - \log A'_{i,j} &\leq -\log \theta_{\min} \qquad (116) \\
&= |\log \theta_{\min}|. \qquad (117)
\end{aligned}$$

Similarly, $\log \widehat{\pi}_i - \log \pi'_i \leq |\log \theta_{\min}|$ for $i = 1, \ldots, |S|$. Thus,

$$\begin{aligned}
\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) &\leq \sum_{m=1}^{T} \sum_{i,j=1}^{|S|} \sum_{t',t''}^{N_m} \left(\widehat{\alpha}_{t',t''}^{(m)} - \bar{\alpha}_{t',t''}^{(m)} \right) x_{t'',i}^{(m)} x_{t',j}^{(m)} \, |\log \theta_{\min}| \\
&\quad \sum_{m=1}^{T} \sum_{i=1}^{|S|} \sum_{t'=1}^{N_m} \left(\widehat{r}_{1,t'}^{(m)} - \bar{r}_{1,t'}^{(m)} \right) x_{t',i}^{(m)} \, |\log \theta_{\min}|, \qquad (118)
\end{aligned}$$

and by (113), with probability greater than $1 - \delta$,

$$\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) < T \, |\log \theta_{\min}| \sqrt{\frac{\log \frac{1}{1-(1-\delta)^{1/T}}}{L}}. \qquad (119)$$

Set

$$\epsilon = T \, |\log \theta_{\min}| \sqrt{\frac{\log \frac{1}{1-(1-\delta)^{1/T}}}{L}}, \qquad (120)$$

solve for L, and plug the resulting value back into (108) with $\delta'' = 1 - (1-\delta)^{1/T}$ to obtain the desired result.

A.2 Proof of Theorem 2

Theorem 1, our probably *approximately* monotonic result, was based on showing that $\left|\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})\right| \leq \epsilon$ with high probability. In order to remove the "approximately" and obtain a *probably monotonic* result we will show that $\Delta(\widehat{\boldsymbol{\theta}})$ concentrates, in a relative sense, about $\Delta(\boldsymbol{\theta}^*)$; *i.e.*, our goal is to show that with high probability,

$$\Delta(\widehat{\boldsymbol{\theta}}) > (1-\epsilon)\Delta(\boldsymbol{\theta}^*).$$

Recall that $\Delta(\boldsymbol{\theta}^*) \geq 0$ by definition, so the relative bound implies that $\Delta(\widehat{\boldsymbol{\theta}}) \geq 0$ with high probability.

We need two preliminary results before we can get to the proof of the theorem. First, we again need to derive concentration inequalities for the Monte Carlo sufficient statistics. Then we use these bounds to show that the corresponding M-step parameter estimates, $\widehat{A}_{i,j}$ and $\widehat{\pi}_i$, concentrate about their asymptotic means, $A^*_{i,j}$ and $\pi^*_{i,j}$. At that point we have everything we need to construct the desired bound on $\Delta(\widehat{\boldsymbol{\theta}}) > (1-\epsilon)\Delta(\boldsymbol{\theta}^*)$.

The proof of Theorem 1 made use of *additive* concentration inequalities, bounding the probability of deviations of the form $\widehat{\mu}_L - \mu \geq t$. In this proof we will need *relative* concentration inequalities to ensure that $\widehat{\mu}_L > (1+\epsilon)\mu$ with high probability.

Proposition 2. *Let $\{(X_i, Z_i)\}$ be a sequence of independent and identically distributed random variables with $X_i \in \{0,1\}$ and $Z_i \in [0,b]$. Assume that $\mathbb{E}[Z_i] = 1$ and $\mathbb{E}[Z_i X_i] = \mu$, and define $\widehat{\mu}_L = (\sum_{i=1}^L Z_i X_i)/(\sum_{i=1}^L Z_i)$, as before. Then for $\epsilon \in (0,1)$,*

$$\Pr\left(\widehat{\mu}_L \geq (1+\epsilon)\mu\right) \quad \leq \quad 2\exp\left\{\frac{-L\mu}{3b}\left(\frac{\epsilon}{1+\sqrt{\frac{2}{3}\mu}+\epsilon\sqrt{\frac{2}{3}\mu}}\right)^2\right\},$$

and for $\epsilon \in (0,1)$,

$$\Pr\left(\widehat{\mu}_L \leq (1-\epsilon)\mu\right) \quad \leq \quad 2\exp\left\{\frac{-L\mu}{3b}\left(\frac{\epsilon}{1+\sqrt{\mu}-\epsilon\sqrt{\mu}}\right)^2\right\}.$$

Proof. Since $X_i \in \{0,1\}$ and $Z_i \in [0,b]$, $Z_i X_i \in [0,b]$ also. Applying the relative form of Hoeffding's inequality (see, *e.g.*, Theorem 2.3 in [15]), we have that for any $\beta > 0$,

$$\Pr\left(\sum_{i=1}^L Z_i X_i \geq (1+\beta)L\mu\right) \quad \leq \quad \exp\left\{\frac{-L\mu\beta^2}{2b(1+\beta/3)}\right\}. \tag{121}$$

If $\beta \leq 1$ then $2(1+\beta/3) < 3$, and so for $\beta \in (0,1]$,

$$\Pr\left(\sum_{i=1}^L Z_i X_i \geq (1+\beta)L\mu\right) \quad \leq \quad \exp\left\{\frac{-L\mu\beta^2}{3b}\right\}, \tag{122}$$

which suffices for our application. Also, for any $\gamma > 0$,

$$\Pr\left(\sum_{i=1}^{L} Z_i \leq (1-\gamma)L\right) \quad \leq \quad \exp\left\{\frac{-L\gamma^2}{2b}\right\}. \tag{123}$$

Suppose the events

$$\left\{\sum_{i=1}^{L} Z_i X_i < (1+\beta)L\mu\right\} \quad \text{and} \quad \left\{\sum_{i=1}^{L} Z_i > (1-\gamma)L\right\} \tag{124}$$

occur simultaneously. Then

$$\widehat{\mu}_L \quad > \quad \left(\frac{1+\beta}{1-\gamma}\right)\mu. \tag{125}$$

Since we will apply the union bound, we balance the exponential rates in (122) and (123) by setting $\gamma = \beta\sqrt{\frac{2}{3}\mu}$. Solving

$$\frac{1+\beta}{1-\gamma} = \frac{1+\beta}{1-\beta\sqrt{\frac{2}{3}\mu}} = 1+\epsilon \tag{126}$$

for β in terms of ϵ results in

$$\beta \quad = \quad \frac{\epsilon}{1+\sqrt{\frac{2}{3}\mu}+\epsilon\sqrt{\frac{2}{3}\mu}}. \tag{127}$$

In order to ensure that $\beta \leq 1$ we restrict

$$\epsilon \quad \leq \quad \frac{1+1}{1-\sqrt{\frac{2}{3}\mu}} - 1 \tag{128}$$

$$= \quad \frac{1+\sqrt{\frac{2}{3}\mu}}{1-\sqrt{\frac{2}{3}\mu}}. \tag{129}$$

Note that the right hand side of the expression above is greater than or equal to 1 for all $\mu \in [0,1]$. Apply the union bound with the complements of the events in (124) using (127) in the exponent to obtain the first result.

The second part proceeds in a similar fashion. Applying the relative Hoeffding bounds yields that for any $\beta > 0$,

$$\Pr\left(\sum_{i=1}^{L} Z_i X_i \leq (1-\beta)L\mu\right) \quad \leq \quad \exp\left\{\frac{-L\mu\beta^2}{2b}\right\} \tag{130}$$

$$\leq \quad \exp\left\{\frac{-L\mu\beta^2}{3b}\right\}, \tag{131}$$

and for any $\gamma \in (0, 1]$,

$$\Pr\left(\sum_{i=1}^{L} Z_i \geq (1+\gamma)L\right) \quad \leq \quad \exp\left\{\frac{-L\gamma^2}{3b}\right\}. \tag{132}$$

Suppose the events

$$\left\{\sum_{i=1}^{L} Z_i X_i > (1-\beta)L\mu\right\} \quad \text{and} \quad \left\{\sum_{i=1}^{L} Z_i < (1+\gamma)L\right\} \tag{133}$$

occur simultaneously. Then

$$\widehat{\mu}_L \quad > \quad \left(\frac{1-\beta}{1+\gamma}\right)\mu. \tag{134}$$

Because we will apply the union bound, we set $\gamma = \beta\sqrt{\mu}$ to balance the rates in (131) and (132), and we restrict $\beta \leq \frac{1}{\sqrt{\mu}}$ so that $\gamma \leq 1$. Solving

$$\frac{1-\beta}{1+\gamma} = \frac{1-\beta}{1+\beta\sqrt{\mu}} = 1-\epsilon \tag{135}$$

for β in terms of ϵ yields

$$\beta \quad = \quad \frac{\epsilon}{1+\sqrt{\mu}-\epsilon\sqrt{\mu}}, \tag{136}$$

and to ensure $\beta \leq \frac{1}{\sqrt{\mu}}$ we restrict

$$\epsilon \quad \leq \quad 1 - \frac{1-\frac{1}{\sqrt{\mu}}}{1+1} \tag{137}$$

$$= \quad \frac{1+\frac{1}{\sqrt{\mu}}}{2}. \tag{138}$$

The right hand side of this expression is also greater than or equal to 1 for all $\mu \in [0, 1]$. Apply the union bound with the complements of the events in (133) using (136) in the exponent to obtain the second result. □

Before proceeding we make some minor simplifications to the bounds just derived. These relaxations only effect constants and do not change the rate of convergence. Observe that $1 + \sqrt{\frac{2}{3}\mu} + \epsilon\sqrt{\frac{2}{3}\mu} \leq 3$ for all $\mu \in [0, 1]$ and $\epsilon \in (0, 1)$. Thus, for any $\epsilon \in (0, 1)$,

$$\Pr\left(\widehat{\mu}_L \geq (1+\epsilon)\mu\right) \quad \leq \quad 2\exp\left\{\frac{-L\mu\epsilon^2}{27b}\right\}. \tag{139}$$

Similarly, $1 + \sqrt{\mu} - \epsilon'\sqrt{\mu} \leq 3$ for all $\mu \in [0, 1]$ and $\epsilon' \in (0, 1)$, and so

$$\Pr\left(\widehat{\mu}_L \leq (1-\epsilon')\mu\right) \quad \leq \quad 2\exp\left\{\frac{-L\mu(\epsilon')^2}{27b}\right\}. \tag{140}$$

Set

$$\delta \quad = \quad 4\exp\left\{\frac{-L\mu\epsilon^2}{27b}\right\}. \tag{141}$$

Then with probability greater than $1-\delta$,

$$\left(1-\sqrt{\frac{27b\log\frac{4}{\delta}}{L\mu}}\right)\mu \;<\; \widehat{\mu}_L \;<\; \left(1+\sqrt{\frac{27b\log\frac{4}{\delta}}{L\mu}}\right)\mu. \tag{142}$$

Next, we will apply our concentration bounds for the individual sufficient statistics, $\widehat{r}_{1,t'}^{(m)}$ and $\widehat{\alpha}_{t',t''}^{(m)}$, to show that each $\widehat{\theta}_i$ is not too far away from θ_i^* with high probability. Recall the exact M-step expressions for π_i^* and $A_{i,j}^*$:

$$\pi_i^* \quad = \quad \frac{\sum_{m=1}^{T}\sum_{t'=1}^{N_m}\bar{r}_{1,t'}^{(m)}x_{t',i}^{(m)}}{\sum_{k=1}^{|S|}\sum_{m=1}^{T}\sum_{t'=1}^{N_m}\bar{r}_{1,t'}^{(m)}x_{t',k}^{(m)}} \tag{143}$$

$$A_{i,j}^* \quad = \quad \frac{\sum_{m=1}^{T}\sum_{t',t''=1}^{N_m}\bar{\alpha}_{t',t''}^{(m)}x_{t'',i}^{(m)}x_{t',j}^{(m)}}{\sum_{k=1}^{|S|}\sum_{m=1}^{T}\sum_{t',t''=1}^{N_m}\bar{\alpha}_{t',t''}^{(m)}x_{t'',i}^{(m)}x_{t',k}^{(m)}}, \tag{144}$$

The corresponding expressions for $\widehat{\pi}_i$ and $\widehat{A}_{i,j}$ are found by replacing each $\bar{r}_{1,t'}^{(m)}$ with $\widehat{r}_{1,t'}^{(m)}$ and $\bar{\alpha}_{t',t''}^{(m)}$ with $\widehat{\alpha}_{t',t''}^{(m)}$. We obtain the following proposition by bounding both the numerators and denominators of these expressions using the two-sided relative bound (142).

Proposition 3. *Let $L>0$ and $\delta>0$ be given. Assume that there exists $\lambda>0$ such that $\bar{r}_{1,t'}^{(m)}\geq\lambda$ and $\bar{\alpha}_{t',t''}^{(m)}\geq\lambda$ for all $m=1,\ldots,T$ and $t',t''=1,\ldots,N_m$. If*

$$L_m \quad \geq \quad \frac{27b_mL}{\lambda}, \tag{145}$$

then with probability at least $1-(\sum_{m=1}^{T}N_m^2)\delta$,

$$\begin{aligned}\left(\bigcap_{i,j=1}^{|S|}\left\{\widehat{A}_{i,j} > \left(\frac{1-\sqrt{\frac{\log\frac{4}{\delta}}{L}}}{1+\sqrt{\frac{\log\frac{4}{\delta}}{L}}}\right)A_{i,j}^*\right\}\right)\\ \bigcap \quad \left(\bigcap_{i=1}^{|S|}\left\{\widehat{\pi}_i > \left(\frac{1-\sqrt{\frac{\log\frac{4}{\delta}}{L}}}{1+\sqrt{\frac{\log\frac{4}{\delta}}{L}}}\right)\pi_i^*\right\}\right).\end{aligned} \tag{146}$$

Proof. First recall that there are $2\binom{N_m}{2}+N_m=N_m^2$ sufficient statistics associated with the mth observation: one for each of the $2\binom{N_m}{2}$ possible transitions and one for each possible initial state. Then, in total there are $\sum_{m=1}^{T}N_m^2$ sufficient statistics to

calculate in the E-step. Applying the union bound in conjunction with (142) we have that with probability greater than $1 - (\sum_{m=1}^{T} N_m^2)\delta$,

$$\bar{\alpha}_{t',t''}^{(m)} - \sqrt{\frac{27 b_m \bar{\alpha}_{t',t''}^{(m)} \log \frac{4}{\delta}}{L_m}} \;<\; \widehat{\alpha}_{t',t''}^{(m)} \;<\; \bar{\alpha}_{t',t''}^{(m)} + \sqrt{\frac{27 b_m \bar{\alpha}_{t',t''}^{(m)} \log \frac{4}{\delta}}{L_m}}, \qquad (147)$$

for all $m = 1, \ldots, T$ and $t', t'' = 1, \ldots, N_m$, and

$$\bar{r}_{1,t'}^{(m)} - \sqrt{\frac{27 b_m \bar{r}_{1,t'}^{(m)} \log \frac{4}{\delta}}{L_m}} \;<\; \widehat{r}_{1,t'}^{(m)} \;<\; \bar{r}_{1,t'}^{(m)} + \sqrt{\frac{27 b_m \bar{r}_{1,t'}^{(m)} \log \frac{4}{\delta}}{L_m}}, \qquad (148)$$

for all $m = 1, \ldots, T$ and $t' = 1, \ldots, N_m$. Based on the assumption that $\bar{\alpha}_{t',t''}^{(m)} \geq \lambda$ and $\bar{r}_{1,t'}^{(m)} \geq \lambda$, taking $L_m \geq 27 b_m L / \lambda$ guarantees that

$$L_m \;\geq\; \max\left(\max_{t',t''=1,\ldots,N_m} \frac{27 b_m L}{\bar{\alpha}_{t',t''}^{(m)}}; \; \max_{t'=1,\ldots,N_m} \frac{27 b_m L}{\bar{r}_{1,t'}^{(m)}} \right). \qquad (149)$$

Then with probability greater than $1 - (\sum_{m=1}^{T} N_m^2)\delta$,

$$\left(1 - \sqrt{\frac{\log \frac{4}{\delta}}{L}}\right) \bar{\alpha}_{t',t''}^{(m)} \;<\; \widehat{\alpha}_{t',t''}^{(m)} \;<\; \left(1 + \sqrt{\frac{\log \frac{4}{\delta}}{L}}\right) \bar{\alpha}_{t',t''}^{(m)}, \qquad (150)$$

for all $m = 1, \ldots, T$ and $t', t'' = 1, \ldots, N_m$, and

$$\left(1 - \sqrt{\frac{\log \frac{4}{\delta}}{L}}\right) \bar{r}_{1,t'}^{(m)} \;<\; \widehat{r}_{1,t'}^{(m)} \;<\; \left(1 + \sqrt{\frac{\log \frac{4}{\delta}}{L}}\right) \bar{r}_{1,t'}^{(m)}, \qquad (151)$$

for all $m = 1, \ldots, T$ and $t' = 1, \ldots, N_m$.

Now, (150) implies that for each i and j,

$$\sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \widehat{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)} \;>\; \left(1 - \sqrt{\frac{\log \frac{4}{\delta}}{L}}\right) \sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)},$$

and for each i,

$$\sum_{k=1}^{|S|} \sum_{m=1}^{T} \sum_{t',t''}^{(m)} \widehat{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',k}^{(m)} \;<\; \left(1 + \sqrt{\frac{\log \frac{4}{\delta}}{L}}\right) \sum_{k=1}^{|S|} \sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',k}^{(m)}.$$

Taking the ratio of these two expressions yields the desired result for $\widehat{A}_{i,j}$ and $A_{i,j}^*$.

Similarly, (151) implies that for each i,

$$\sum_{m=1}^{T} \sum_{t'=1}^{N_m} \widehat{r}_{1,t'}^{(m)} x_{t',i}^{(m)} \;>\; \left(1 - \sqrt{\frac{\log \frac{4}{\delta}}{L}}\right) \sum_{m=1}^{T} \sum_{t'}^{N_m} \bar{r}_{1,t'}^{(m)} x_{t',i}^{(m)}, \qquad (152)$$

and for each i,

$$\sum_{m=1}^{T}\sum_{t'=1}^{N_m}\widehat{r}_{1,t'}^{(m)}x_{t',i}^{(m)} \quad < \quad \left(1+\sqrt{\frac{\log\frac{4}{\delta}}{L}}\right)\sum_{m=1}^{T}\sum_{t'}^{N_m}\bar{r}_{1,t'}^{(m)}x_{t',i}^{(m)}. \tag{153}$$

Taking the ratio of these two expressions yields the desired result for $\widehat{\pi}_i$ and π_i^*. □

The remainder of the proof of Theorem 2 is now fairly straightforward. Let $\delta > 0$ be the value given in the statement of Theorem 2. Monotonicity of the logarithm in conjunction with Proposition 3 implies that with probability greater than $1-\delta$,

$$\log\widehat{A}_{i,j} \quad > \quad \log A_{i,j}^* + \log\left(\frac{1-\sqrt{\frac{\log\frac{4\sum_{m=1}^{T}N_m^2}{\delta}}{L}}}{1+\sqrt{\frac{\log\frac{4\sum_{m=1}^{T}N_m^2}{\delta}}{L}}}\right), \tag{154}$$

for every i and j, and

$$\log\widehat{\pi}_i \quad > \quad \log\pi_i^* + \log\left(\frac{1-\sqrt{\frac{\log\frac{4\sum_{m=1}^{T}N_m^2}{\delta}}{L}}}{1+\sqrt{\frac{\log\frac{4\sum_{m=1}^{T}N_m^2}{\delta}}{L}}}\right), \tag{155}$$

for every i. Multiplying each term by

$$\sum_{m=1}^{T}\sum_{t',t''=1}^{N_m}\bar{\alpha}_{t',t''}^{(m)}x_{t'',i}^{(m)}x_{t',j}^{(m)} \quad > \quad 0, \tag{156}$$

or

$$\sum_{m=1}^{T}\sum_{t'=1}^{N_m}\bar{\alpha}_{t',t''}^{(m)}x_{t',i}^{(m)} \quad > \quad 0, \tag{157}$$

as appropriate, and then summing over i and j, we obtain that

$$\begin{aligned}Q(\widehat{\boldsymbol{\theta}};\boldsymbol{\theta}') \;>\; & Q(\boldsymbol{\theta}^*;\boldsymbol{\theta}') \\ & + \left(\sum_{i,j=1}^{|S|}\sum_{m=1}^{T}\sum_{t',t''=1}^{N_m}\bar{\alpha}_{t',t''}^{(m)}x_{t'',i}^{(m)}x_{t',j}^{(m)} + \sum_{i=1}^{|S|}\sum_{m=1}^{T}\sum_{t'=1}^{N_m}\bar{r}_{1,t'}^{(m)}x_{t',i}^{(m)}\right)\log\left(\frac{1-\sqrt{\frac{\log\frac{4\sum_{m=1}^{T}N_m^2}{\delta}}{L}}}{1+\sqrt{\frac{\log\frac{4\sum_{m=1}^{T}N_m^2}{\delta}}{L}}}\right).\end{aligned}$$

It follows from the definitions of $\bar{r}_{1,t'}^{(m)}$ and $\bar{\alpha}_{t',t''}^{(m)}$ that $\sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} = 1$ and $\sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} = N_m - 1$. Thus,

$$\begin{aligned}
&\sum_{i,j=1}^{|S|} \sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)} + \sum_{i=1}^{|S|} \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} x_{t',i}^{(m)} \\
&\quad = \sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} + \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} \qquad (158) \\
&\quad = \sum_{m=1}^{T} (N_m - 1) + \sum_{m=1}^{T} 1 \qquad (159) \\
&\quad = \sum_{m=1}^{T} N_m, \qquad (160)
\end{aligned}$$

and with probability greater than $1 - \delta$,

$$Q(\widehat{\boldsymbol{\theta}}; \boldsymbol{\theta}') > Q(\boldsymbol{\theta}^*; \boldsymbol{\theta}') + \left(\sum_{m=1}^{T} N_m \right) \log \left(\frac{1 - \sqrt{\frac{\log \frac{4 \sum_{m=1}^{T} N_m^2}{\delta}}{L}}}{1 + \sqrt{\frac{\log \frac{4 \sum_{m=1}^{T} N_m^2}{\delta}}{L}}} \right). \quad (161)$$

Subtract $Q(\boldsymbol{\theta}'; \boldsymbol{\theta}')$ from both sides to obtain that with probability greater than $1 - \delta$,

$$\Delta(\widehat{\boldsymbol{\theta}}) > \Delta(\boldsymbol{\theta}^*) + \left(\sum_{m=1}^{T} N_m \right) \log \left(\frac{1 - \sqrt{\frac{\log \frac{4 \sum_{m=1}^{T} N_m^2}{\delta}}{L}}}{1 + \sqrt{\frac{\log \frac{4 \sum_{m=1}^{T} N_m^2}{\delta}}{L}}} \right). \quad (162)$$

Next, let $\epsilon > 0$ be the value given in the statement of Theorem 2 and set

$$\left(\sum_{m=1}^{T} N_m \right) \log \left(\frac{1 - \sqrt{\frac{\log \frac{4 \sum_{m=1}^{T} N_m^2}{\delta}}{L}}}{1 + \sqrt{\frac{\log \frac{4 \sum_{m=1}^{T} N_m^2}{\delta}}{L}}} \right) = -\epsilon \Delta(\boldsymbol{\theta}^*). \quad (163)$$

Solving for L yields

$$L = \left(\frac{1 + \exp\left\{ \frac{-\epsilon \Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m} \right\}}{1 - \exp\left\{ \frac{-\epsilon \Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m} \right\}} \right)^2 \log \left(\frac{4 \sum_{m=1}^{T} N_m^2}{\delta} \right). \quad (164)$$

Recall the well known inequality:

Lemma 1. *$u \geq \log(1 + u)$ for $u \geq 0$.*

Applying Lemma 1 with $u = \frac{\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^T N_m} \geq 0$ gives

$$\frac{\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^T N_m} \quad \geq \quad 1 + \frac{\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^T N_m}. \tag{165}$$

Take the exponential, which is a monotonic transformation, and then invert the resulting expression to obtain

$$\exp\left\{\frac{-\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^T N_m}\right\} \quad \leq \quad \left(1 + \frac{\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^T N_m}\right)^{-1}. \tag{166}$$

It follows that

$$\frac{1 + \exp\left\{\frac{-\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^T N_m}\right\}}{1 - \exp\left\{\frac{-\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^T N_m}\right\}} \quad \leq \quad \frac{2\sum_{m=1}^T N_m + \epsilon\Delta(\boldsymbol{\theta}^*)}{\epsilon\Delta(\boldsymbol{\theta}^*)}. \tag{167}$$

Using this last result in (164), together with the choice of L_m from Proposition 3, we find that if we use

$$L_m \quad = \quad \frac{27 b_m}{\lambda}\left(\frac{2\sum_{m=1}^T N_m + \epsilon\Delta(\boldsymbol{\theta}^*)}{\epsilon\Delta(\boldsymbol{\theta}^*)}\right)^2 \log\left(\frac{4\sum_{m=1}^T N_m^2}{\delta}\right) \tag{168}$$

importance samples for the mth observation in the Monte Carlo E-step, then $\Delta(\widehat{\boldsymbol{\theta}}) \geq (1-\epsilon)\Delta(\boldsymbol{\theta}^*)$ with probability greater than $1-\delta$. Since $\Delta(\boldsymbol{\theta}^*) \geq 0$ by definition we may take $\epsilon = 1$. Then $\Delta(\widehat{\boldsymbol{\theta}}^*) \geq 0$ with probability greater than $1-\delta$, and this is exactly what we wanted to show.

References

[1] *International Workshop on Brain Connectivity*, 2005. `http://www.ccs.fau.edu/~bc2005/welcome.html`.

[2] J. Bernardo and A. Smith. *Bayesian Theory*. John Wiley & Sons, 1994.

[3] J. Booth, J. Hobert, and W. Jank. A survey of monte carlo algorithms for maximizing the likelihood of a two-stage hierarchical model. *Statistical Modelling*, 1:333–349, 2001.

[4] R. Boyles. On the convergence of the EM algorithm. *Journal of the Royal Statistical Society B*, 45(1):47–50, 1983.

[5] B. Caffo, W. Jank, and G. Jones. Ascent-based Monte Carlo EM. *Journal of the Royal Statistical Society B*, 67(2):235–252, 2005.

[6] M. Coates, A. Hero, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal Processing Magazine*, 19(3):47–65, 2002.

[7] M. Figueiredo and A. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, March 2002.

[8] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:713–721, 1963.

[9] W. Jank. Stochastic variants of the em algorithm: Monte carlo, quasi-monte carlo and more. In *Proc. of the American Statistical Association*, Minneapolis, Minnesota, August 2005.

[10] D. Justice and A. Hero. Estimation of message source and destination from link intercepts. Submitted to *IEEE Trans. on Information Forensics and Security*, April 2005.

[11] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach. *Systems Biology in Practice: Concepts, Implementation and Application*. John Wiley and Sons, 2005.

[12] J. Kubica, A. Moore, D. Cohn, and J. Schneider. cGraph: A fast graph-based method for link analysis and queries. In *Proc. IJCAI Text-Mining and Link-Analysis Workshop*, Acapulco, Mexico, August 2003.

[13] J. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[14] Y. Liu and H. Zhao. A computational approach for ordering signal transduction pathway components from genomics and proteomics data. *BMC Bioinformatics*, 5(158), October 2004.

[15] C. McDiarmid. Concentration. In M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 195–248. Springer-Verlag, New York, 1998.

[16] M. Newman, A. Barabasi, and D. Watts. *The Structure and Dynamics of Networks*. Princeton University Pres, 2006.

[17] B. Palsson. *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, 2006.

[18] M. Rabbat, J. Treichler, S. Wood, and M. Larimore. Understanding the topology of a telephone network via internally-sensed network tomography. In *Proc. IEEE International Confernece on Acoustics, Speech, and Signal Processing*, volume 3, pages 977–980, Philadelphia, PA, March 2005.

[19] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Verlag, New York, 1999.

[20] O. Sporns, D. Chialvo, M. Kaiser, and C. Hilgetag. Organization, development and function of complex brain networks. *Trends in Cognitive Science*, 8(9), 2004.

[21] S. Wasserman, K. Faust, D. Iacobucci, and M. Granovetter. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[22] C. Wu. On the convergence properties of the EM algorithm. *Annals of Statistics*, 11(1):95–103, 1983.

[23] D. Zhu, A. Hero, H. Cheng, R. Khanna, and A. Swaroop. Network constrained clustering for gene microarray data. *Bioinformatics*, 21(21):4014–4020, 2005.