

# Network Inference From Co-Occurrences

Michael G. Rabbat, *Member, IEEE*, Mário A. T. Figueiredo, *Senior Member, IEEE*, and Robert D. Nowak, *Senior Member, IEEE*

**Abstract**—The discovery of networks is a fundamental problem arising in numerous fields of science and technology, including communication systems, biology, sociology, and neuroscience. Unfortunately, it is often difficult, or impossible, to obtain data that directly reveal network structure, and so one must infer a network from incomplete data. This paper considers inferring network structure from “co-occurrence” data: observations that identify which network components (e.g., switches, routers, genes) carry each transmission but do not indicate the order in which they handle the transmission. Without order information, the number of networks that are consistent with the data grows exponentially with the size of the network (i.e., the number of nodes). Yet, the basic engineering/evolutionary principles underlying most networks strongly suggest that not all data-consistent networks are equally likely. In particular, nodes that co-occur in many observations are probably closely connected. With this in mind, we model the co-occurrence observations as independent realizations of a random walk on the network, subjected to a random permutation to account for the lack of order information. Treating permutations as missing data, we derive an expectation–maximization (EM) algorithm for estimating the random walk parameters. The model and EM algorithm significantly simplify the problem, but the computational complexity of the reconstruction process does grow exponentially in the length of each transmission path. For networks with long paths, the exact E-step may be computationally intractable. We propose a polynomial-time Monte Carlo EM algorithm based on importance sampling and derive conditions that ensure convergence of the algorithm with high probability. Simulations and experiments with Internet measurements demonstrate the promise of this approach.

**Index Terms**—Expectation–maximization (EM) algorithm, graphical models, importance sampling, Markov models, network inference, network tomography.

Manuscript received May 22, 2006; revised February 18, 2008. Published August 27, 2008 (projected). This work was supported in part by Fundação para a Ciência e Tecnologia, Portuguese Ministry of Science and Higher Education, under Project POSC/EEA-CPS/61924/2004, and the National Science Foundation under Grant CCF-0728767. The material in this paper was presented in part at the Conference on Neural Information Processing Systems, Vancouver, BC, Canada, December 2006.

M. G. Rabbat is with the Department of Electrical and Computer Engineering, McGill University, Montréal, QC H3A 2A7, Canada (e-mail: michael.rabbat@mcgill.ca).

M. A. T. Figueiredo is with the Instituto de Telecomunicações and the Department of Electrical and Computer Engineering, Instituto Superior Técnico, Lisboa 1049-001, Portugal (e-mail: mario.figueiredo@lx.it.pt).

R. D. Nowak is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA (e-mail: nowak@engr.wisc.edu).

Communicated by E. Modiano, Associate Editor for Communication Networks.

Color versions of Figures 1–5 in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIT.2008.926315

## I. NETWORK INFERENCE 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. The analysis of communication networks enables a better understanding of routing, transmission patterns, and information flow [1], [2]. Characteristics of biological networks provide insight into the functional roles played by different genes, proteins, and metabolites in biological systems [3], [4]. Social network analysis can be used to gain a deeper understanding of interactions, dynamics, and the structure of organizations [5], [6]. Functional connectivity networks of brain regions are studied to better understand coupling and interaction between different neuronal colonies [7], [8]. Obtaining or inferring the structure of networks from experimental data precedes any such analysis and is thus a basic and fundamental task, critical to many applications.

Unfortunately, measurements that directly reveal network structure are often beyond experimental capabilities or are excessively expensive. This paper considers inferring network structure from observations that identify which network components (e.g., switches, routers, genes) carry each transmission but do not indicate the order in which they handle the transmissions. Mathematically, the underlying network structure can be represented as a directed graph, and the vertices involved in each transmission form a connected subgraph. The observations only reflect which subset of vertices are involved, or “co-occur,” in each transmission; not their interconnectivity. We refer to such observations as *co-occurrences*. Co-occurrence observations arise naturally in each of the application areas mentioned above.

Transmissions over telecommunication networks are carried by links and routers/switches that form a path between the source and terminal nodes. In some cases, it is impossible to directly observe the order in which the routers/switches handle each transmission, since sensors are geographically distributed, making precise time-synchronization impractical. The so-called internally sensed network tomography problem specifically aims at recovering network structure from unordered lists of network elements along transmission paths [1].

Biological signal transduction networks describe fundamental cell functions such as growth, metabolism, differentiation, and apoptosis (disintegration) [4]. Although it is possible to test for individual, localized interactions between protein pairs, such experiments are expensive and time-consuming. High-throughput measurement techniques such as microarrays have successfully been used to identify the components of different signal transduction pathways [9]. However, microarray

data only reflect order information at a very coarse, unreliable level. Developing computational techniques for inferring pathway orders is an active research area [10].

Co-occurrence or transactional data also appear in the context of social networks, e.g., by considering which academic papers are cocited by another paper, which web pages are linked to or from another web page, or which people were 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 of occurrence. Researchers in this area have considered the problems of reconstructing networks from co-occurrence data and of using the inferred network to predict potential future co-occurrences [11].

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 coactivate while a patient is performing different tasks, we can obtain multiple co-occurrence observations. Although fMRI offers high spatial resolution, its limited temporal resolution makes it impractical to obtain complete order information. Magnetoencephalography and electroencephalography measure activity in the brain with higher temporal resolution but only provide coarse spatial resolution. Consequently, these techniques do not allow a precise determination of which functional regions are active during a given task. Existing techniques for obtaining functional coactivation networks either involve brute-force measurement or use crude correlation methods (see [7] and references therein).

In this paper, we focus on observations arising from transmissions in a network. Specifically, each co-occurrence observation corresponds to a path<sup>1</sup> through the network. We observe the vertices comprising each path but not the order in which they appear along 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 vertices in each observation, and then inserting directed edges between vertices, which are adjacent in the assigned order. In light of the many possible orders for each co-occurrence observation, the number of feasible topologies grows exponentially in the number and size of observations. Without additional assumptions, side information, or prior knowledge, there is no reason to prefer one feasible topology over the others.

Previous work on related problems has involved heuristics using frequencies of co-occurrence either to assign an order to each path [1] or to approximate the probability of transitioning from one vertex to another [11]. These approaches make stringent assumptions and sacrifice flexibility in order to achieve computational tractability and systematically identify a unique solution. The *frequency method* introduced in [1] 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., for load balancing) the algorithm may fail. The *cGraph* algorithm of Kubica *et al.* [11] inserts weighted edges between every pair of vertices, which co-occur in some observation. This approach produces solutions that 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 [12], 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 (NICO).

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 each path to account for lack of observed order information. Based on this model, network inference reduces to estimating the parameters governing the random walk. Then, these parameter estimates determine the most likely order for each co-occurrence.

The following interpretation motivates our shuffled random walk 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 its final destination, a transmission arriving at this vertex must pass through one of the neighboring vertices next. By recording how many arriving transmissions are passed to each neighbor over a period of time, it is possible to calculate the empirical probability of transmission to each neighbor. Obtaining such probabilities at each vertex would provide a tremendous amount of information about the network. Unfortunately, co-occurrence observations do not directly reveal transition probabilities and we, therefore, face a challenging inverse problem. This paper develops a formal 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.

A particularly novel aspect of the problem and approach introduced in this article is the counterintuitive idea of recovering temporal dynamics from nontemporal data. Models of information flow and causal signaling are quite common in problems where one is able to measure both where and when events occur. However, in the problem considered here, we have no knowledge of the temporal chain of events associated with each observation. We leverage correlation among the different activity patterns observed across the network, together with the notion that similar activity patterns are presumably caused by a common

<sup>1</sup>Throughout this paper, a “path” refers to a sequence of vertices  $(x_1, x_2, \dots, x_T)$  such that there is an edge between each adjacent pair of vertices,  $x_{i-1}$  and  $x_i$ , and no vertex appears more than once in the sequence.

stimulus, to recover the dynamics of information flow, a seemingly impossible task. Indeed, it is impossible to glean information about dynamics from just one co-occurrence observation, and it is only through an ensemble of patterns, which we assume are generated by the same network process, that information flow dynamics are recovered.

It is also worth mentioning that the approach discussed in this paper differs considerably from that of learning the structure of a directed graphical model or Bayesian network, a graph where nodes correspond to random variables and edges indicate conditional independence relationships [13], [14]. A typical aim of graphical modeling is to find a graph corresponding to a factorization of a high-dimensional distribution that predicts the observations well. In turn, these probabilistic models do not directly reflect physical structures, and applying such an approach in the context of this problem would ignore physical constraints inherent to the observations: that co-occurring vertices must lie along a path in the network. We note that, although the Bayesian network paradigm does not directly fit our problem setup, Teyssier and Koller [15] describe an approach to Bayesian network structure learning, which is similar to the network inference algorithm presented in this paper. In [15], rather than searching over all Bayesian network structures, a search is performed over all orderings of random variables in the model. This simplifies the search procedure since it is easier to determine the mostly likely directed acyclic graph that is consistent with a fixed ordering.

The rest of this paper is organized as follows. In Section II, we introduce notation and formulate the problem setup. Section III reviews the standard approach to estimating the parameters of a random walk when fully observed (ordered) samples are available and presents an expectation–maximization (EM) algorithm for estimating random walk parameters from shuffled observations. A Monte Carlo variant of the EM algorithm is described in Section IV for situations where long transmission paths make the E-step computationally prohibitive. Section V analyzes convergence of the Monte Carlo EM (MCEM) algorithm. Simulation results are presented in Section VI and the paper is concluded in Section VII, where ongoing work is also briefly described.

## II. PROBLEM FORMULATION

We model the network as a simple directed graph  $G = (V, E)$ , where  $V = \{1, 2, \dots, |V|\}$  is the set of vertices and  $E \subseteq V \times V$  is the set of edges. The number of vertices  $|V|$  is considered known, so network inference amounts to determining the adjacency structure of the graph; that is, identifying whether  $(i, j) \in E$ , for every pair of vertices  $(i, j) \in V \times V$ .

A co-occurrence observation  $\mathbf{x} \subset V$  is a subset of vertices in the graph that 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  $N$  times with different stimuli, we obtain observations  $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\}$ , where  $\mathbf{x}^{(n)} = (x_1^{(n)}, x_2^{(n)}, \dots, x_{T_n}^{(n)})$  is a length- $T_n$  co-occurrence, indexed in an arbitrary order.

A directed graph  $G = (V, E)$  is said to be data consistent with respect to observations  $\mathbf{X}$  if for each co-occurrence  $\mathbf{x}^{(n)} \in \mathbf{X}$  there exists an ordered path  $\mathbf{w}^{(n)} = (w_1^{(n)}, w_2^{(n)}, \dots, w_{T_n}^{(n)})$  and a permutation  $\pi^{(n)} = (\pi^{(n)}(1), \dots, \pi^{(n)}(T_n))$  such that  $w_t^{(n)} = x_{\pi^{(n)}(t)}^{(n)}$  for each  $t$ , and there is an edge from  $w_{t-1}^{(n)}$  to  $w_t^{(n)}$  in the graph for  $t = 2, \dots, T_n$ , that is,  $(w_{t-1}^{(n)}, w_t^{(n)}) \in E$ .

Notice that network inference from ordered paths is trivial. We can begin with an empty graph  $G_0 = (V, E)$  with  $E = \{\}$ . Then, for each ordered observation  $\mathbf{w}^{(n)}$ , we update the set of edges via  $E \leftarrow E \cup (w_{t-1}^{(n)}, w_t^{(n)})$  for  $t = 2, \dots, T_n$ . Even if we do not observe ordered paths, if we observe the permutation  $\pi^{(n)}$  along with each co-occurrence  $\mathbf{x}^{(n)}$ , we can use the permutation to recover the correctly ordered observation and apply the same procedure.

In practice, we do not make ordered observations and we do not have access to the correct permutations. However, we can obtain a feasible reconstruction by associating *any* permutation (of the appropriate length) with each co-occurrence, and then following the procedure described above. There are  $T_n!$  ways to permute the elements of  $\mathbf{x}^{(n)}$ , so there may be as many as  $\prod_{n=1}^N T_n!$  feasible reconstructions. Clearly, for large  $T_n$  and  $N$ , 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.

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 together in many co-occurrences, we expect that they are close in the underlying network topology. Likewise, we expect that most vertices will only be directly connected to a small fraction of the other vertices. Based on this intuition, we propose the following probabilistic model. First, we model the unobserved, ordered paths  $\mathbf{w}^{(n)}$  as independent samples of a first-order Markov chain. The Markov chain is parameterized by transition probabilities  $\theta_{i,j} = \mathbb{P}[w_t = j | w_{t-1} = i]$  and initial state probabilities  $\theta_{0,i} = \mathbb{P}[w_1 = i]$ ; we denote by  $\boldsymbol{\theta}$  the entire collection of Markov chain parameters. Of course, these parameters must satisfy the normalization constraints

$$\sum_{j=1}^{|V|} \theta_{i,j} = 1, \quad \text{for each } i = 0, 1, \dots, |V|. \quad (1)$$

In addition, we assume that the support of the transition matrix is determined by the adjacency structure of the underlying network; i.e.,  $\theta_{i,j} > 0$  if and only if  $(i, j) \in E$ .

A co-occurrence observation  $\mathbf{x}$  is generated by shuffling the elements of an ordered Markov chain sample  $\mathbf{w} = (w_1, \dots, w_T)$  via a permutation  $\pi$  drawn uniformly from  $\mathbb{S}_T$ , the collection of all permutations of  $T$  objects. Thus, for each  $t = 1, \dots, T$ ,  $x_{\pi(t)} = w_t$ . We assume the random permutation  $\pi$  is independent of the Markov chain sample  $\mathbf{w}$ .

Based on this model, we can write the likelihood of a co-occurrence observation  $\mathbf{x}$  conditioned on the permutation  $\pi$  as

$$\mathbb{P}[\mathbf{x}|\pi, \boldsymbol{\theta}] = \theta_{0, x_{\pi(1)}} \prod_{t=2}^T \theta_{x_{\pi(t-1)}, x_{\pi(t)}}. \quad (2)$$

Since  $\mathbb{P}[\pi] = 1/(T!)$ , for all  $\pi \in \mathbb{S}_T$ , marginalization over all permutations leads to

$$\mathbb{P}[\mathbf{x}|\boldsymbol{\theta}] = \frac{1}{T!} \sum_{\pi \in \mathbb{S}_T} \mathbb{P}[\mathbf{x}|\pi, \boldsymbol{\theta}]. \quad (3)$$

Finally, assuming that co-occurrence observations are independent, and taking the logarithm, gives

$$\log \mathbb{P}[\mathbf{X}|\boldsymbol{\theta}] = \sum_{n=1}^N \left[ \log \left( \sum_{\pi \in \mathbb{S}_{T_n}} \mathbb{P}[\mathbf{x}^{(n)}|\pi, \boldsymbol{\theta}] \right) - \log(T_n!) \right]. \quad (4)$$

Under this model, network inference consists in computing an estimate for the Markov chain parameters  $\boldsymbol{\theta}$ ; given a prior  $\mathbb{P}[\boldsymbol{\theta}]$ , a natural criterion for estimation is the maximum a posteriori criterion

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \log \mathbb{P}[\mathbf{X}|\boldsymbol{\theta}] + \log \mathbb{P}[\boldsymbol{\theta}]. \quad (5)$$

Of course, when  $\mathbb{P}[\boldsymbol{\theta}]$  is a constant, independent of  $\boldsymbol{\theta}$ , this reduces to the maximum-likelihood criterion. With the estimate  $\hat{\boldsymbol{\theta}}$  in hand, we may determine the most likely permutation for each co-occurrence observation according to  $\hat{\boldsymbol{\theta}}$ , and obtain a feasible reconstruction using our procedure for ordered observations described above.

For nontrivial observations,  $\log \mathbb{P}[\mathbf{X}|\boldsymbol{\theta}]$  is a complicated, non-concave function of  $\boldsymbol{\theta}$ , so solving (5) is not a simple task. In the next section, we derive an EM algorithm for finding local maxima of this optimization problem, by treating the set of permutations,  $\{\pi^{(1)}, \dots, \pi^{(N)}\}$  as missing data.

### III. EM ALGORITHM FOR ESTIMATING MARKOV CHAIN PARAMETERS FROM SHUFFLED OBSERVATIONS

#### A. Fully Observed Markov Chains: Notation and Estimation

Let  $\mathbf{W} = \{\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(T)}\}$  be a set of sample paths  $\mathbf{w}^{(n)} = (w_1^{(n)}, \dots, w_{T_n}^{(n)})$ , independently generated by a Markov chain with parameters  $\boldsymbol{\theta}$  [see (1)]. For later use, it is convenient to introduce the equivalent binary representation  $\boldsymbol{\omega}^{(n)} \in \{0, 1\}^{T_n \times |V|}$  for each sample  $\mathbf{w}^{(n)}$ , defined such that  $\omega_{t,i}^{(n)} = \mathbb{1}_{\{w_t^{(n)}=i\}}$  for  $i = 1, \dots, |V|$ , where  $\mathbb{1}_{\{\cdot\}}$  is the indicator function. Since  $\mathbf{w}^{(n)}$  and  $\boldsymbol{\omega}^{(n)}$  are equivalent representations of the same information, we will also write  $\mathbf{W} \equiv \{\boldsymbol{\omega}^{(1)}, \dots, \boldsymbol{\omega}^{(N)}\}$ , with a slight abuse of notation. With this notation, we can write

$$\begin{aligned} \log \mathbb{P}[\mathbf{W}|\boldsymbol{\theta}] &= \sum_{n=1}^N \sum_{i=1}^{|V|} \omega_{1,i}^{(n)} \log \theta_{0,i} \\ &+ \sum_{n=1}^N \sum_{t=2}^{T_n} \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \omega_{t-1,i}^{(n)} \omega_{t,j}^{(n)} \log \theta_{i,j} \end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^{|V|} \log \theta_{0,i} \sum_{n=1}^N \omega_{1,i}^{(n)} \\ &+ \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \log \theta_{i,j} \sum_{n=1}^N \sum_{t=2}^{T_n} \omega_{t-1,i}^{(n)} \omega_{t,j}^{(n)}. \end{aligned}$$

Maximum-likelihood estimates of  $\boldsymbol{\theta}$  can be obtained from  $\mathbf{W}$  by maximizing  $\log \mathbb{P}[\mathbf{W}|\boldsymbol{\theta}]$  under the constraints in (1); the solution is well known

$$\begin{aligned} \hat{\theta}_{i,j} &= \frac{\sum_{n=1}^N \sum_{t=2}^{T_n} \omega_{t-1,i}^{(n)} \omega_{t,j}^{(n)}}{\sum_{j=1}^{|V|} \sum_{n=1}^N \sum_{t=2}^{T_n} \omega_{t-1,i}^{(n)} \omega_{t,j}^{(n)}} \\ \hat{\theta}_{0,i} &= \frac{1}{N} \sum_{n=1}^N \omega_{1,i}^{(n)}. \end{aligned} \quad (6)$$

#### B. Shufflings, Permutations, and the EM Algorithm

To address the case where we have a set of co-occurrences  $\mathbf{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ , not ordered samples, we define the equivalent binary representation  $\mathbf{X} \equiv \{\boldsymbol{\chi}^{(1)}, \dots, \boldsymbol{\chi}^{(N)}\}$  in a similar way as above:  $\boldsymbol{\chi}^{(n)} \in \{0, 1\}^{T_n \times |V|}$  and  $\chi_{t,i}^{(n)} = \mathbb{1}_{\{x_t^{(n)}=i\}}$ .

Equivalent to using  $\pi \in \mathbb{S}_T$  to denote a length- $T$  permutation/shuffling, we introduce a more convenient (binary) representation; each shuffling is represented by a *permutation matrix*, which we also refer to as a *shuffling matrix*. Let the shuffling matrix corresponding to a permutation  $\pi$  be denoted as  $\mathbf{A}^\pi$ , where  $A_{i,j}^\pi = \mathbb{1}_{\{\pi(i)=j\}}$ . Thus, an ordered sequence  $\boldsymbol{\omega}$ , permutation  $\pi$ , and corresponding co-occurrence  $\boldsymbol{\chi}$  are related via  $\boldsymbol{\omega} = \mathbf{A}^\pi \boldsymbol{\chi}$ .

Let  $\Pi = \{\pi^{(1)}, \dots, \pi^{(N)}\}$  be the collection of permutations that recover the ordered paths  $\mathbf{W} = \{\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(N)}\}$  from the corresponding shuffled co-occurrences  $\mathbf{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ . Recall that the permutations  $\pi^{(m)}$  are assumed to be independent of the Markov chain parameters  $\boldsymbol{\theta}$ . We can write the complete log-likelihood  $\log \mathbb{P}[\mathbf{X}, \Pi|\boldsymbol{\theta}]$  as follows:

$$\begin{aligned} \log \mathbb{P}[\mathbf{X}, \Pi|\boldsymbol{\theta}] &= \log \mathbb{P}[\mathbf{X}|\Pi, \boldsymbol{\theta}] + \log \mathbb{P}[\Pi] \\ &= \sum_{n=1}^N \log \mathbb{P}[\boldsymbol{\chi}^{(n)}|\pi^{(n)}, \boldsymbol{\theta}] + \log \mathbb{P}[\Pi] \\ &= \sum_{n=1}^N \sum_{t=2}^{T_n} \sum_{t', t''=1}^{T_n} \sum_{i,j=1}^{|V|} A_{t-1,t'}^{\pi^{(n)}} A_{t,t''}^{\pi^{(n)}} \chi_{t',i}^{(n)} \chi_{t'',j}^{(n)} \log \theta_{i,j} \\ &+ \sum_{n=1}^N \sum_{t'=1}^{T_n} \sum_{i=1}^{|V|} A_{1,t'}^{\pi^{(n)}} \chi_{t',i}^{(n)} \log \theta_{0,i} + \log \mathbb{P}[\Pi] \end{aligned} \quad (7)$$

where  $\mathbb{P}[\Pi]$  is the probability of the set of permutations  $\Pi$ , which is constant in our model since all permutations are equiprobable.

To estimate  $\boldsymbol{\theta}$  from  $\mathbf{X}$ , we treat  $\Pi$  as missing data, opening the door to the use of the EM algorithm. Notice that if we had the complete data  $(\mathbf{X}, \Pi)$ , we could recover  $\mathbf{W}$  via  $\boldsymbol{\omega}^{(n)} = \mathbf{A}^{\pi^{(n)}} \boldsymbol{\chi}^{(n)}$  and obtain the closed-form estimates (6). The EM algorithm proceeds by computing the expected value

of  $\log \mathbb{P}[\mathbf{X}, \Pi | \boldsymbol{\theta}]$  (w.r.t.  $\Pi$ ), conditioned on the observations and on the current model estimate  $\boldsymbol{\theta}^k$  (the E-step)

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^k) = \mathbb{E}_{\Pi} [\log \mathbb{P}[\mathbf{X}, \Pi | \boldsymbol{\theta}] | \mathbf{X}, \boldsymbol{\theta}^k] \quad (9)$$

where we write  $\mathbb{E}_{\Pi}$  to denote expectation with respect to the missing permutation variables  $\Pi$ . The model parameter estimates are then updated as follows (the M-step):

$$\boldsymbol{\theta}^{k+1} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^k). \quad (10)$$

These two steps are repeated cyclically until a convergence criterion is met.

### C. The E-Step

1) *Sufficient Statistics*: Rearranging (8), and dropping  $\log \mathbb{P}[\Pi]$  (a constant), we can write

$$\begin{aligned} & \log \mathbb{P}[\mathbf{X}, \Pi | \boldsymbol{\theta}] \\ & \propto \sum_{n=1}^N \sum_{i,j=1}^{|V|} \sum_{t',t''=1}^{T_n} \sum_{t=2}^{T_n} A_{t-1,t'}^{\pi(n)} A_{t,t''}^{\pi(n)} \chi_{t',i}^{(n)} \chi_{t'',j}^{(n)} \log \theta_{i,j} \\ & + \sum_{n=1}^N \sum_{i=1}^{|V|} \sum_{t'=1}^{T_n} A_{1,t'}^{\pi(n)} \chi_{t',i}^{(n)} \log \theta_{0,i}, \\ & = \sum_{n=1}^N \sum_{i,j=1}^{|V|} \sum_{t',t''=1}^{T_n} \alpha_{t',t''}^n \chi_{t',i}^{(n)} \chi_{t'',j}^{(n)} \log \theta_{i,j} \\ & + \sum_{n=1}^N \sum_{i=1}^{|V|} \sum_{t'=1}^{T_n} \alpha_{0,t'}^n \chi_{t',i}^{(n)} \log \theta_{0,i}, \end{aligned} \quad (11)$$

revealing that  $\log \mathbb{P}[\mathbf{X}, \Pi | \boldsymbol{\theta}]$  is linear with respect to the following simple functions.

- The first entry of each permutation  $\pi^{(n)}$ :  $\alpha_{0,t'}^n = A_{1,t'}^{\pi(n)} = \mathbb{1}_{\{\pi^{(n)}(1)=t'\}}$ , for  $n = 1, \dots, N$  and  $t' = 1, \dots, T_n$ ;
- Transition indicators

$$\alpha_{t',t''}^n = \sum_{t=2}^{T_n} A_{t-1,t'}^{\pi(n)} A_{t,t''}^{\pi(n)} = \sum_{t=2}^{T_n} \mathbb{1}_{\{\pi^{(n)}(t-1)=t'\}} \mathbb{1}_{\{\pi^{(n)}(t)=t''\}}$$

for  $n = 1, \dots, N$ , and  $t', t'' = 1, \dots, T_n$ .

The E-step reduces to computing the conditional expectations of  $\alpha_{0,t'}^n$  and  $\alpha_{t',t''}^n$  given  $\boldsymbol{\theta}^k$  (denoted  $\bar{\alpha}_{0,t'}^{n,k}$  and  $\bar{\alpha}_{t',t''}^{n,k}$ ), since the expectation is a linear operator and hence it commutes with linear functions. Noticing that  $\alpha_{0,t'}^n$  and  $\alpha_{t',t''}^n$  are binary-valued yields

$$\bar{\alpha}_{0,t'}^{n,k} = \mathbb{E} [\alpha_{0,t'}^n | \mathbf{X}, \boldsymbol{\theta}^k] = \mathbb{P} [\alpha_{0,t'}^n = 1 | \mathbf{X}, \boldsymbol{\theta}^k] \quad (12)$$

$$\bar{\alpha}_{t',t''}^{n,k} = \mathbb{E} [\alpha_{t',t''}^n | \mathbf{X}, \boldsymbol{\theta}^k] = \mathbb{P} [\alpha_{t',t''}^n = 1 | \mathbf{X}, \boldsymbol{\theta}^k]. \quad (13)$$

Finally,  $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^k)$  is obtained simply by plugging  $\bar{\alpha}_{0,t'}^{n,k}$  and  $\bar{\alpha}_{t',t''}^{n,k}$  in the places of  $\alpha_{0,t'}^n$  and  $\alpha_{t',t''}^n$  in (11).

2) *Computing  $\bar{\alpha}_{0,t'}^{n,k}$* : Since the permutations are (a priori) equiprobable, for  $\pi \in \mathbb{S}_{T_n}$ , we have  $\mathbb{P}[\pi] = 1/(T_n!)$ ,  $\mathbb{P}[\pi(1) =$

$t'] = ((T_n - 1)!/T_n!) = 1/T_n$ , and  $\mathbb{P}[\pi(\pi(1) = t') = 1/((T_n - 1)!)]$ . Using these facts, together with the mutual independence among the several sequences, and Bayes law, yields

$$\begin{aligned} \bar{\alpha}_{0,t'}^{n,k} &= \mathbb{P} [\alpha_{0,t'}^n = 1 | \mathbf{x}^{(n)}, \boldsymbol{\theta}^k] \\ &= \frac{\mathbb{P}[\mathbf{x}^{(n)} | \pi^{(n)}(1) = t', \boldsymbol{\theta}^k] \mathbb{P}[\pi^{(n)}(1) = t']}{\mathbb{P}[\mathbf{x}^{(n)} | \boldsymbol{\theta}^k]} \\ &= \frac{\sum_{\pi \in \mathbb{S}_{T_n}: \pi(1)=t'} \mathbb{P}[\mathbf{x}^{(n)} | \pi, \boldsymbol{\theta}^k]}{\sum_{\pi \in \mathbb{S}_{T_n}} \mathbb{P}[\mathbf{x}^{(n)} | \pi, \boldsymbol{\theta}^k]} \end{aligned} \quad (14)$$

where each term  $\mathbb{P}[\mathbf{x}^{(n)} | \pi, \boldsymbol{\theta}^k]$  is easily computed after using  $\pi$  to unshuffle  $\mathbf{x}^{(n)}$

$$\mathbb{P}[\mathbf{x}^{(n)} | \pi, \boldsymbol{\theta}^k] = \theta_{0, x_{\pi(1)}^{(n)}}^k \prod_{t=2}^{T_n} \theta_{x_{\pi(t-1)}^{(n)}, x_{\pi(t)}^{(n)}}^k \cdot$$

3) *Computing  $\bar{\alpha}_{t',t''}^{n,k}$* : The computation of  $\bar{\alpha}_{t',t''}^{n,k}$  follows a similar path as that of  $\bar{\alpha}_{0,t'}^{n,k}$ ; since all permutations are equiprobable, for  $\pi \in \mathbb{S}_{T_n}$ ,  $\mathbb{P}[\pi(t-1) = t', \pi(t) = t''] = (T_n - 2)!/(T_n!)$  and  $\mathbb{P}[\pi(\pi(t-1) = t', \pi(t) = t'') = 1/((T_n - 2)!)]$ , thus

$$\begin{aligned} \bar{\alpha}_{t',t''}^{n,k} &= \sum_{t=2}^{T_n} \mathbb{P}[\pi^{(n)}(t-1) = t', \pi^{(n)}(t) = t'' | \mathbf{x}^{(n)}, \boldsymbol{\theta}^k] \\ &= \sum_{t=2}^{T_n} \left[ \frac{\mathbb{P}[\mathbf{x}^{(n)} | \pi^{(n)}(t-1) = t', \pi^{(n)}(t) = t'', \boldsymbol{\theta}^k]}{\mathbb{P}[\mathbf{x}^{(n)} | \boldsymbol{\theta}^k]} \right. \\ & \quad \left. \times \mathbb{P}[\pi^{(n)}(t-1) = t', \pi^{(n)}(t) = t''] \right] \\ &= \sum_{t=2}^{T_n} \frac{\left( \frac{1}{(T_n-2)!} \sum_{\pi \in \mathbb{S}_{T_n}: \pi(t-1)=t', \pi(t)=t''} \mathbb{P}[\mathbf{x}^{(n)} | \pi, \boldsymbol{\theta}^k] \right)}{\frac{1}{T_n!} \sum_{\pi \in \mathbb{S}_{T_n}} \mathbb{P}[\mathbf{x}^{(n)} | \pi, \boldsymbol{\theta}^k]} \\ & \quad \times \left( \frac{(T_n-2)!}{T_n!} \right) \\ &= \frac{\sum_{\pi \in \mathbb{S}_{T_n}} \mathbb{P}[\mathbf{x}^{(n)} | \pi, \boldsymbol{\theta}^k] \sum_{t=2}^{T_n} \mathbb{1}_{\{\pi(t-1)=t'\}} \mathbb{1}_{\{\pi(t)=t''\}}}{\sum_{\pi \in \mathbb{S}_{T_n}} \mathbb{P}[\mathbf{x}^{(n)} | \pi, \boldsymbol{\theta}^k]}. \end{aligned} \quad (15)$$

Exact computation of the sufficient statistics  $\{\bar{\alpha}_{0,t'}^{n,k}\}_{t'=1}^{T_n}$  and  $\{\bar{\alpha}_{t',t''}^{n,k}\}_{t',t''=1}^{T_n}$  via (14) and (15) requires enumerating all permutations of  $\mathbf{x}^{(n)}$ . For large  $T_n$ , this is a heavy load; Section IV describes a Monte Carlo sampling approach for computing approximations to  $\bar{\alpha}_{0,t'}^{n,k}$  and  $\bar{\alpha}_{t',t''}^{n,k}$ .

### D. The M-Step

Recall that the function  $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^k)$  is obtained by plugging  $\bar{\alpha}_{0,t'}^{n,k}$  and  $\bar{\alpha}_{t',t''}^{n,k}$  in the places of  $\alpha_{0,t'}^n$  and  $\alpha_{t',t''}^n$ , respectively, in

(11). Maximization w.r.t.  $\theta$ , under the constraints in (1), leads to following simple update equations:

$$\theta_{i,j}^{k+1} = \frac{\sum_{n=1}^N \sum_{t',t''=1}^{T_n} \bar{\alpha}_{t',t''}^{n,k} \chi_{t',i}^{(n)} \chi_{t'',j}^{(n)}}{\sum_{j=1}^{|V|} \sum_{n=1}^N \sum_{t',t''=1}^{T_n} \bar{\alpha}_{t',t''}^{n,k} \chi_{t',i}^{(n)} \chi_{t'',j}^{(n)}} \quad (16)$$

and

$$\theta_{0,i}^{k+1} = \frac{\sum_{n=1}^N \sum_{t'=1}^{T_n} \bar{\alpha}_{0,t'}^{n,k} \chi_{t',i}^{(n)}}{\sum_{i=1}^{|V|} \sum_{n=1}^N \sum_{t'=1}^{T_n} \bar{\alpha}_{0,t'}^{n,k} \chi_{t',i}^{(n)}}.$$

### E. Handling Known Endpoints

In some applications, (one or both of) the endpoints of each path are known and only the internal nodes are shuffled. This is the case in communication networks (i.e., internally sensed network tomography), since the sources and destinations are known, but not the connectivity within the network. In estimation of 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) [3]; in this case, the stimulus and response act as fixed endpoints, and our goal is to infer the order of the sequence of protein interactions.

Observe that knowledge of the endpoints of each path imposes the constraints  $\pi^{(n)}(1) = 1$  and  $\pi^{(n)}(T_n) = T_n$ . Under the first constraint, estimates of the initial state probabilities are simply given by (for all  $k$ )

$$\theta_{0,i}^k = \frac{1}{N} \sum_{n=1}^N \chi_{1,i}^{(n)}.$$

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

$$\tilde{\mathfrak{S}}_T = \{\pi \in \mathfrak{S}_T : \pi(1) = 1, \pi(T) = T\}$$

denote the set of permutations of  $T$  elements with fixed endpoints. As in the general case, the E-step can be computed using summary statistics (for  $t', t'' = 1, \dots, T_n$ )

$$\tilde{\gamma}^{n,k} = \sum_{\pi \in \tilde{\mathfrak{S}}_{T_n}} \mathbb{P}[\mathbf{x}^{(m)} | \pi, \theta^k]$$

$$\tilde{\gamma}_{t',t''}^{n,k} = \sum_{\pi \in \tilde{\mathfrak{S}}_{T_n}} \mathbb{P}[\mathbf{x}^{(m)} | \pi, \theta^k] \sum_{t=2}^{N_m} \mathbb{1}_{\{\pi(t-1)=t'\}} \mathbb{1}_{\{\pi(t)=t''\}}$$

and setting  $\bar{\alpha}_{t',t''}^{n,k} = \tilde{\gamma}_{t',t''}^{n,k} / \tilde{\gamma}$ . The M-step (update for  $\theta_{i,j}^{k+1}$ ) remains unchanged.

### F. Incorporating Prior Information

The EM algorithm can be easily modified to incorporate conjugate priors; these are Dirichlet priors for each row  $\theta_i = (\theta_{i,1}, \dots, \theta_{i,|V|})$ ,  $i = 0, \dots, |V|$  of  $\theta$

$$\mathbb{P}[\theta_i | \mathbf{u}_i] \propto \prod_{j=1}^{|V|} \theta_{i,j}^{u_{i,j}-1} \quad (17)$$

which are proper priors if and only if the parameters  $u_{i,j}$  are nonnegative [16]. The larger  $u_{0,j}$  relative to the other  $u_{0,j'}$ ,  $j' \neq j$ , the greater our prior belief is that state  $j$  is an initial state rather than the others; equivalently, the expected value of  $\theta_0$  under the Dirichlet distribution is given by  $\mathbb{E}[\theta_{0,j} | \mathbf{u}_0] = u_{0,j} / \sum_{i=1}^{|V|} u_{0,i}$ . Similarly, the larger  $u_{i,j}$  relative to other  $u_{i,j'}$  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. Consider the prior distribution on the initial state distribution; taking  $u_{0,j} = c > 1$ , for all  $j$ , has a *smoothing* effect, as if all of the states had some mass, regardless of the observations, in the initial state distribution.

Due to the conjugacy of the Dirichlet priors, their incorporation into the EM algorithm only results in a minor change to the M-step. Incorporating the prior leads to the following modified version of the function  $Q(\theta; \theta^k)$ :

$$\begin{aligned} Q(\theta; \theta^k) \propto & \sum_{n=1}^N \sum_{i,j=1}^{|V|} \sum_{t',t''=1}^{T_n} \bar{\alpha}_{t',t''}^{n,k} \chi_{t',i}^{(n)} \chi_{t'',j}^{(n)} \log \theta_{i,j} \\ & + \sum_{n=1}^N \sum_{i=1}^{|V|} \sum_{t'=1}^{T_n} \bar{\alpha}_{0,t'}^{n,k} \chi_{t',i}^{(n)} \log \theta_{0,i} \\ & + \sum_{i,j=1}^{|V|} (u_{i,j} - 1) \log \theta_{i,j} + \sum_{i=1}^{|V|} (u_{0,i} - 1) \log \theta_{0,i}. \end{aligned} \quad (18)$$

Note that the prior does not involve the missing data (permutations) and thus does not effect the E-step calculation. The M-step updates become

$$\theta_{0,i}^{k+1} = \frac{u_{0,i} - 1 + \sum_{n=1}^N \sum_{t'=1}^{T_n} \bar{\alpha}_{0,t'}^{n,k} \chi_{t',i}^{(n)}}{\sum_{i=1}^{|V|} \left( u_{0,i} - 1 + \sum_{n=1}^N \sum_{t'=1}^{T_n} \bar{\alpha}_{0,t'}^{n,k} \chi_{t',i}^{(n)} \right)} \quad (19)$$

$$\theta_{i,j}^{k+1} = \frac{u_{i,j} - 1 + \sum_{n=1}^N \sum_{t',t''=1}^{T_n} \bar{\alpha}_{t',t''}^{n,k} \chi_{t',i}^{(n)} \chi_{t'',j}^{(n)}}{\sum_{j=1}^{|V|} \left( u_{i,j} - 1 + \sum_{n=1}^N \sum_{t',t''=1}^{T_n} \bar{\alpha}_{t',t''}^{n,k} \chi_{t',i}^{(n)} \chi_{t'',j}^{(n)} \right)}. \quad (20)$$

## IV. MONTE CARLO E-STEP BY IMPORTANCE SAMPLING

For long sequences, the combinatorial nature of (14) and (15) (involving sums over all permutations of each sequence) may render exact computation impractical. In this section, we consider Monte Carlo approximate versions of the E-step, which

avoid the combinatorial nature of its exact version. The MCEM algorithm, based on an MC version of the E-step, was originally proposed in [17], and used ever since by many authors (recent work can be found in [18]–[20] and references therein).

To lighten the notation in this section, we drop the superscript from  $\theta^k$ , using simply  $\theta$  as the current parameter estimates. Moreover, we focus on a particular length- $T$  co-occurrence  $\mathbf{x} = (x_1, \dots, x_T) \in V^T$  and drop the superscript  $(n)$ ; due to the independence of the paths, there is no loss of generality. Recall that  $\mathbf{x}$  is a (shuffled) path, and thus has no repeated elements.

The E-step [see (12) and (13)] consists of computing the conditional expectations  $\bar{\alpha}_{0,t'} = \mathbb{E}[\alpha_{0,t'} | \mathbf{x}, \theta]$  and  $\bar{\alpha}_{t',t''} = \mathbb{E}[\alpha_{t',t''} | \mathbf{x}, \theta]$ . A naive Monte Carlo approximation would be based on random permutations, sampled from the uniform distribution over  $\mathcal{S}_T$ . However, the reason to resort to approximation techniques in the first place is that  $\mathcal{S}_T$  is large, with only a small fraction of these random permutations having nonnegligible posterior probability,  $\mathbb{P}[\pi | \mathbf{x}, \theta]$ ; a very large number of uniform samples is thus needed to obtain a good approximation to  $\bar{\alpha}_{0,t'}$  and  $\bar{\alpha}_{t',t''}$ .

Ideally, we would sample permutations directly from the posterior  $\mathbb{P}[\pi | \mathbf{x}, \theta]$ ; however, this would require determining its value for all  $T!$  permutations. Instead, we employ importance sampling (IS) (see, e.g., [21] and [22], for an introduction to IS): we sample  $L$  permutations,  $\pi_1, \dots, \pi_L$ , from a distribution  $\mathbb{R}[\pi | \mathbf{x}, \theta]$ , from which it is easier to sample than  $\mathbb{P}[\pi | \mathbf{x}, \theta]$ , then apply a corrective reweighting to obtain approximations to  $\bar{\alpha}_{0,t'}$  and  $\bar{\alpha}_{t',t''}$ . The IS estimates are given by

$$\hat{\alpha}_{0,t'} = \frac{\sum_{i=1}^L z_i \mathbb{1}_{\{\pi_i(1)=t'\}}}{\sum_{i=1}^L z_i} \quad (21)$$

$$\hat{\alpha}_{t',t''} = \frac{\sum_{i=1}^L z_i \sum_{t=2}^T \mathbb{1}_{\{\pi_i(t-1)=t'\}} \mathbb{1}_{\{\pi_i(t)=t''\}}}{\sum_{i=1}^L z_i} \quad (22)$$

where  $z_i$ , the correction factor (or weight) for sample  $\pi_i$ , is given by

$$z_i = \frac{\mathbb{P}[\pi_i | \mathbf{x}, \theta]}{\mathbb{R}[\pi_i | \mathbf{x}, \theta]} \quad (23)$$

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  $R[\pi] = Z_R \mathbb{R}[\pi | \mathbf{x}, \theta]$  and  $P[\pi] = Z_P \mathbb{P}[\pi | \mathbf{x}, \theta]$ , for constants  $Z_R$  and  $Z_P$ , we can use

$$z'_i = \frac{P[\pi_i]}{R[\pi_i]} = \frac{Z_P}{Z_R} z_i \quad (24)$$

instead of  $z_i$  in (21) and (22); the approximations will remain unchanged since the factor  $Z_P/Z_R$  will appear both in the numerator and denominator of (21) and (22), thus canceling out.

The IS framework just described is general, and the performance of this approach is closely tied to the particular sampling scheme employed. In particular, the more closely the shape of the sampling distribution  $\mathbb{R}$  matches the shape of the target distribution  $\mathbb{P}$ , the better the quality of the estimate will be. Although our goal is to accurately approximate the E-step sufficient statistics, our primary motivation for using IS is to speed up calculation of the E-step. The remainder of this section describes an IS scheme that is both simple to implement (fast) and closely mimics the generative Markov model for ordered paths. Next, we describe the IS scheme, including the derivation of closed-form expressions for both the sampling distribution  $\mathbb{R}$  and the sample weights  $z_i$ . We conclude the section by mentioning other sampling variants.

### A. Causal Sampling Scheme

Let  $\mathbf{f} = (f_1, \dots, f_{|V|}) \in \{0, 1\}^{|V|}$  be a sequence of binary flags. Given a probability distribution  $\mathbf{p} = (p_1, p_2, \dots, p_{|V|})$  on the set of states  $V$ , denote by  $\mathbf{p} | \mathbf{f}$  the restriction of  $\mathbf{p}$  to those elements of  $V$  that have corresponding flag  $f_i$  set to 1, that is

$$(\mathbf{p} | \mathbf{f})_i = \frac{p_i f_i}{\sum_{j=1}^{|V|} p_j f_j}, \quad \text{for } i = 1, 2, \dots, |V|. \quad (25)$$

The proposed sampling scheme is defined as follows.

- Step 1) Let  $\mathbf{f} = (f_1, \dots, f_{|V|})$  be initialized according to  $f_i = \mathbb{1}_{\{i \in \mathbf{x}\}}$ . Obtain one sample from  $V$  according to the distribution  $\theta_0 | \mathbf{f}$ . Let the obtained sample be denoted  $s$ ; of course, one and only one element of  $\mathbf{x}$  is equal to  $s$ . Locate the position  $t$  of  $s$  in  $\mathbf{x}$ ; that is, find  $t$  such that  $x_t = s$ . Set  $\pi(1) = t$ . Set  $f_s = 0$  (preventing  $x_t$  from being sampled again). Set  $i = 2$ .
- Step 2) Obtain a sample  $s'$  from  $S$ , according to the distribution  $\theta_s | \mathbf{f}$ , where  $\theta_s$  denotes the  $s$ th row of the transition matrix. Find  $t$  such that  $x_t = s'$ . Set  $\pi(i) = t$ . Set  $f_{s'} = 0$ .
- Step 3) If  $i < T$ , then set  $s \leftarrow s'$ , set  $i \leftarrow i + 1$ ; go back to Step 2); otherwise, stop.

1) *Sampling Distribution:* Before deriving the form of the distribution  $\mathbb{R}$ , let us begin by writing the target distribution  $\mathbb{P}[\pi | \mathbf{x}, \theta]$  explicitly. Using Bayes law,

$$\mathbb{P}[\pi | \mathbf{x}, \theta] = \frac{\mathbb{P}[\mathbf{x} | \pi, \theta] \mathbb{P}[\pi]}{\mathbb{P}[\mathbf{x} | \theta]} \quad (26)$$

since  $\pi$  does not depend *a priori* on  $\theta$ . Based on our assumption that all permutations are equiprobable, we have  $P[\pi] = \mathbb{1}_{\{\pi \in \mathcal{S}_T\}} / T!$ . Noticing that the denominator in (26) is just a normalizing constant independent of  $\pi$ , we have

$$\begin{aligned} \mathbb{P}[\pi | \mathbf{x}, \theta] &\propto \mathbb{1}_{\{\pi \in \mathcal{S}_T\}} \mathbb{P}[\mathbf{x} | \pi, \theta] \\ &= \mathbb{1}_{\{\pi \in \mathcal{S}_T\}} \left( \theta_{0, x_{\pi(1)}} \prod_{t=2}^T \theta_{x_{\pi(t-1)}, x_{\pi(t)}} \right). \end{aligned} \quad (27)$$

For the sake of notational economy, we will write simply  $\mathbb{R}[\pi]$  to represent  $\mathbb{R}[\pi|\mathbf{x}, \boldsymbol{\theta}]$ . The sequential nature of the sampling scheme suggests a factorization of the form

$$\mathbb{R}[\pi] = \mathbb{R}[\pi(1)] \mathbb{R}[\pi(2)|\pi(1)] \mathbb{R}[\pi(3)|\pi(2), \pi(1)] \cdots \times \mathbb{R}[\pi(T)|\pi(T-1), \dots, \pi(1)]. \quad (28)$$

For Step 1) of the sampling scheme, it is clear that, for  $\pi(1) = 1, \dots, T$ ,

$$\mathbb{R}[\pi(1)] \propto \theta_{0, x_{\pi(1)}}. \quad (29)$$

For the  $i$ th iteration, we have

$$\mathbb{R}[\pi(i)|\pi(i-1), \dots, \pi(1)] \propto \theta_{x_{\pi(i-1)}, x_{\pi(i)}} \mathbb{1}_{\{\pi(i) \notin \{\pi(i-1), \dots, \pi(1)\}\}}$$

where the indicator term simply expresses that  $\pi(i)$  cannot be equal to one of the previous samples  $\pi(i-1), \dots, \pi(1)$ . Observe that the normalization constant for this distribution can be expressed as

$$\sum_{j=1}^{|V|} \theta_{x_{\pi(i-1)}, j} f_j \quad (30)$$

where  $f_j = 1$  if  $x_t = j$  for some  $t \notin \{\pi(i-1), \dots, \pi(1)\}$ . Thus, for the  $i$ th iteration, we can write

$$\begin{aligned} \mathbb{R}[\pi(i)|\pi(i-1), \dots, \pi(1)] \\ = \theta_{x_{\pi(i-1)}, x_{\pi(i)}} \phi_i(\pi(i-1), \dots, \pi(1)) \mathbb{1}_{\{\pi(i) \notin \{\pi(i-1), \dots, \pi(1)\}\}} \end{aligned} \quad (31)$$

with

$$\phi_i(\pi(i-1), \dots, \pi(1)) = \left( \sum_{t \notin \{\pi(i-1), \dots, \pi(1)\}} \theta_{x_{\pi(i-1)}, x_t} \right)^{-1}.$$

Inserting (29) and (31) into (28), we finally have

$$\begin{aligned} \mathbb{R}[\pi] \\ \propto \left[ \theta_{0, x_{\pi(1)}} \prod_{t=2}^T \theta_{x_{\pi(t-1)}, x_{\pi(t)}} \right] \left[ \prod_{t=2}^T \phi_t(\pi(t-1), \dots, \pi(1)) \right] \\ \times \left[ \prod_{t=2}^T \mathbb{1}_{\{\pi(t) \notin \{\pi(t-1), \dots, \pi(1)\}\}} \right]. \end{aligned} \quad (32)$$

Note that the third factor on the right-hand side of (32) is simply the indicator that  $\pi$  is a permutation, i.e., it is equal to  $\mathbb{1}_{\{\pi \in \mathcal{S}_T\}}$ , for any  $\pi \in \{1, \dots, T\}^T$ .

Dividing (27) by (32), we obtain the correction factor  $z$  for a permutation sample  $\pi$  generated using this sequential scheme as

$$\begin{aligned} z &= \left( \prod_{i=2}^T \phi_i(\pi(i-1), \dots, \pi(1)) \right)^{-1} \\ &= \prod_{i=2}^T \sum_{t \notin \{\pi(i-1), \dots, \pi(1)\}} \theta_{x_{\pi(i-1)}, x_t}. \end{aligned}$$

With this quantity in hand, we have all the ingredients needed to produce IS estimates  $\hat{\alpha}_{0,t'}$  and  $\hat{\alpha}_{t',t''}$ . Notice that computing

the terms  $\phi_i$ , and thus computing  $z$ , is easy since these factors are the normalization terms for the distributions  $\theta_s|\mathbf{f}$ , which are already computed while performing each iteration of Step 2. Thus, we just need to store the product of these normalizing constants to finally obtain the weight  $z$ .

2) *Known Endpoints*: In the case where the endpoints are known, we fix  $\pi(1) = 1$ ,  $\pi(T) = T$ , and set  $f_{x_1} = 0$  and  $f_{x_T} = 0$  in Step 1); the remainder of the procedure is unchanged. Based on these constraints, the importance sampling weight takes a slightly different form

$$z = \theta_{0, x_1} \theta_{x_{\pi(T-1)}, x_T} \prod_{i=2}^{T-1} \sum_{t \notin \{\pi(i-1), \dots, \pi(1)\}} \theta_{x_{\pi(i-1)}, x_t}. \quad (33)$$

## B. Other Sampling Schemes

In addition to the causal sampling scheme that we have just described, we have also developed other sampling schemes that work in a hierarchical, rather than sequential, fashion. For the sake of space, we refrain from describing these other sampling schemes; detailed descriptions can be found in [23]. In particular, we have developed a two-stage hierarchical scheme and a fully hierarchical scheme. In the two-stage method, the first stage samples from the collection of all possible transitions occurring in a path; then, the second stage samples from the distribution on all arrangements of these transitions, to form a permutation. In the fully hierarchical method, the first stage samples a suitable set of transitions, say  $\mathcal{G}_1$ ; then, the following stage samples a suitable collection of pairs of elements of  $\mathcal{G}_1$ , yielding a collection of quadruples  $\mathcal{G}_2$  and the procedure is repeated until a permutation is obtained.

A detailed comparison of these sampling schemes with the causal sampler described above is presented in [23]. Empirically, we find that the causal sampler performs the best (lowest approximation error for a fixed number of importance samples), and so we use this sampling scheme for the remainder of this paper. Moreover, our original motivation for resorting to Monte Carlo methods was to improve the speed of computation of our algorithm. The causal sampling scheme has complexity that is linear in the length of the path to be sampled, and hence very clearly meets our needs.

## V. MONOTONICITY AND CONVERGENCE

Well-known convergence results due to Wu and Boyles [24], [25] guarantee convergence of our EM algorithm when the E-step calculation is performed exactly. By choosing  $\boldsymbol{\theta}^{k+1} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^k)$  in the M-step, our iterates satisfy the *monotonicity property*

$$Q(\boldsymbol{\theta}^{k+1}; \boldsymbol{\theta}^k) \geq Q(\boldsymbol{\theta}^k; \boldsymbol{\theta}^k). \quad (34)$$

The marginal log-likelihood (4) is continuous in its parameters  $\boldsymbol{\theta}$  and it is bounded above. In this setting, the monotonicity property (34) guarantees that each exact EM update monotonically increases the marginal log-likelihood, so the EM iterates converge to a local maximum.

When Monte Carlo methods are used in the E-step, monotonicity is no longer guaranteed since the M-step solves  $\hat{\boldsymbol{\theta}}^{k+1} = \arg \max_{\boldsymbol{\theta}} \hat{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^k)$ , where  $\hat{Q}$  is defined analogously to

$Q$  but with terms  $\bar{\alpha}_{t',t''}^{n,k}$  and  $\bar{\alpha}_{0,t'}^{n,k}$  replaced by  $\hat{\alpha}_{t',t''}^{n,k}$  and  $\hat{\alpha}_{0,t'}^{n,k}$ , their corresponding importance sampling approximations. Consequently, care must be taken to ensure that  $\hat{Q}$  approximates  $Q$  well enough so that the EM algorithm is not swamped with error from the Monte Carlo estimates.

To illustrate this issue, consider the following synthetic example. We generate 40 co-occurrence observations by taking a random walk on a graph with 140 vertices. Each co-occurrence has between four and eight vertices. Fig. 1(a) plots  $Q(\theta^k; \theta^{k-1})$  for the exact E-step, along with  $\hat{Q}(\hat{\theta}^{k+1}; \hat{\theta}^k)$  and  $Q(\hat{\theta}^{k+1}; \hat{\theta}^k)$  for the MCEM algorithm using only ten importance samples per co-occurrence. Although  $\hat{Q}(\hat{\theta}^{k+1}; \hat{\theta}^k)$  increases at each iteration,  $Q(\hat{\theta}^{k+1}; \hat{\theta}^k)$  clearly does not and the monotonicity property does not hold. This is apparent in Fig. 1(b), where the dashed-dotted line shows the progress of the marginal log-likelihood (our optimization criterion) for the ten-sample MCEM algorithm. When enough importance samples are used, the MCEM algorithm performs comparably to the exact EM algorithm; see the dashed line in Fig. 1(b) corresponding to an MCEM algorithm using 1000 importance samples per co-occurrence. All three instances of the EM algorithm used in this example start from the same initialization.

Recently, researchers have considered the question of how many importance samples should be used in a Monte Carlo E-step [18]–[20], [26]. The goal is to balance monotonicity and computational efficiency by using enough samples to have a good chance at monotonicity while not using excessively many samples. Booth *et al.* [20] 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. 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.

Fort and Moulines consider asymptotic convergence of MCEM in [19]. In particular, they prove consistency of the MCEM for curved exponential families using various forms of the ergodic theorem for Markov chains under the assumption that the number of Monte Carlo samples grows at a suitable rate with respect to the number of EM iterations. Caffo *et al.* [26] propose a method for automatically adapting the number of Monte Carlo samples used at each EM iteration. Let  $\Delta(\theta^{k+1}) = Q(\theta^{k+1}; \theta^k) - Q(\theta^k; \theta^k)$  and  $\hat{\Delta}(\hat{\theta}^{k+1}) = \hat{Q}(\hat{\theta}^{k+1}; \hat{\theta}^k) - \hat{Q}(\hat{\theta}^k; \hat{\theta}^k)$ . Recall that  $L$  importance samples are used to calculate the terms in  $\hat{Q}$ . The algorithm of Caffo *et al.* is based on a central limit theorem-like approximation in which they show that  $\sqrt{L}(\hat{\Delta}(\hat{\theta}^{k+1}) - \Delta(\hat{\theta}^{k+1}))$  converges in distribution to the standard normal. Observe that the monotonicity property (34) is equivalent to the condition  $\Delta(\hat{\theta}^{k+1}) \geq 0$ . Although  $\Delta(\hat{\theta}^{k+1})$  cannot be computed without computing the exact sufficient statistics  $\{\bar{\alpha}_{t',t''}^{n,k}\}$  and  $\{\bar{\alpha}_{0,t'}^{n,k}\}$ , we can compute  $\hat{\Delta}(\hat{\theta}^{k+1})$ . Their scheme then amounts to increasing the number of Monte Carlo samples until  $\hat{\Delta}(\hat{\theta}^{k+1}) > \epsilon$  for a user-specified  $\epsilon > 0$ . Then, applying an asymptotic standard normal tail approximation, they obtain a statement of the

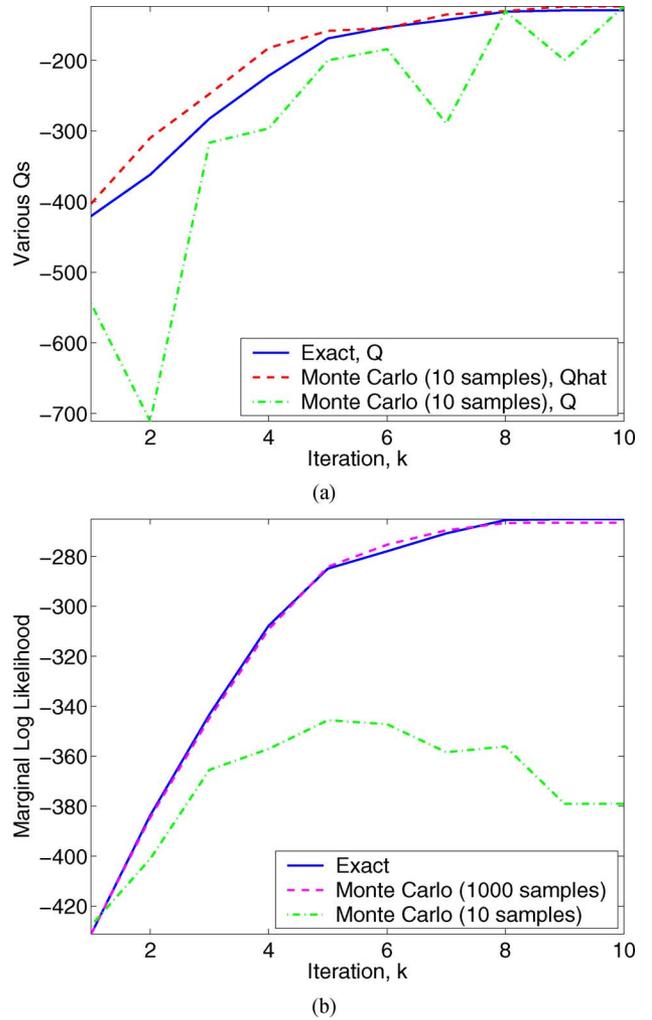


Fig. 1. Example with simulated observations illustrating that the MCEM algorithm may not result in monotonic increase of the marginal log-likelihood if too few Monte Carlo samples are used. The solid line in (a) is  $Q(\theta^{k+1}; \theta^k)$  for exact EM iterations, the dashed line is  $\hat{Q}(\hat{\theta}^{k+1}; \hat{\theta}^k)$ , and the dashed-dotted line is  $Q(\hat{\theta}^{k+1}; \hat{\theta}^k)$  for MCEM iterations using only ten samples. Even though  $\hat{Q}$  increases monotonically,  $Q$  may not be monotonic for the MCEM algorithm. (b) Marginal log-likelihood for exact EM iterates and for two versions of the MCEM. MCEM performance closely resembles that of the exact EM algorithm when sufficiently many importance samples are used.

form  $\Pr(\hat{\Delta}(\hat{\theta}^{k+1}) - \Delta(\hat{\theta}^{k+1}) \geq \epsilon) \leq \delta(\epsilon)$ . Based on this statement they claim that monotonicity holds with probability at least  $1 - \delta(\epsilon)$ . They further remark that if a different  $\epsilon_k$  is chosen at each iteration, so that  $\sum_{k=1}^{\infty} \delta(\epsilon_k) < \infty$ , then, by the Borel–Cantelli lemma,  $\Pr(\hat{\Delta}(\hat{\theta}^k) - \Delta(\hat{\theta}^k) \geq \epsilon_k \text{ i.o.}) = 0$ , so there exists a  $K > 0$  such that  $\hat{\Delta}(\hat{\theta}^k) - \Delta(\hat{\theta}^k) < \epsilon_k$  for all  $k \geq K$  with probability 1; i.e., eventually every EM update is monotonic. Of course, in practice, the algorithm is terminated after a finite number of iterations, so we may never reach the stage where all iterates are monotonic.

Notice that for the monotonicity condition  $\Delta(\hat{\theta}^{k+1}) \geq 0$  to truly hold in the above framework, the events

$$\left\{ \hat{\Delta}(\hat{\theta}^{k+1}) - \Delta(\hat{\theta}^{k+1}) \leq \epsilon \right\} \quad \text{and} \quad \left\{ \hat{\Delta}(\hat{\theta}^{k+1}) \geq \epsilon \right\}$$

must occur simultaneously. Because the probabilistic bound above only addresses one of these events, we refer to this type of result as guaranteeing an  $(\epsilon, \delta)$ -probably approximately monotonic update (PAM). More generally, an  $(\epsilon, \delta)$ -PAM result states that with probability at least  $1 - \delta$ , the update will be  $\epsilon$ -approximately monotonic; i.e.,  $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}^{k+1}) - \Delta(\widehat{\boldsymbol{\theta}}^{k+1}) \leq \epsilon$  implies  $\Delta(\widehat{\boldsymbol{\theta}}^{k+1}) \geq -\epsilon$ , because, by definition,  $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}^{k+1}) \geq 0$ .

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 finite-sample PAM result presented next. Recall that independent importance samples are drawn for each co-occurrence observation in the Monte Carlo E-step. Denote by  $L_n$  the number of importance samples used to compute sufficient statistics for observation  $\mathbf{x}^{(n)}$ . The computational complexity of the exact E-step computation for this observation requires  $T_n!$  operations [enumerating all permutations of  $\mathbf{x}^{(n)}$ ], and thus increases with the size of the co-occurrence. 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 ( $T_n^2$  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  $z_i = \mathbb{P}[\pi_i | \mathbf{x}, \boldsymbol{\theta}] / \mathbb{R}[\pi_i | \mathbf{x}, \boldsymbol{\theta}]$ , where  $\mathbb{P}$  is the target distribution and  $\mathbb{R}$  is the importance sampling distribution. A key assumption was made that  $\mathbb{P}$  is absolutely continuous with respect to  $\mathbb{R}$ ; that is,  $\mathbb{P}[\pi | \mathbf{x}, \boldsymbol{\theta}] = 0$  for every permutation  $\pi$  with  $\mathbb{R}[\pi | \mathbf{x}, \boldsymbol{\theta}] = 0$ . We adopt the convention  $0/0 = 0$  so that  $z_i = 0$  for such samples. This guarantees that  $z_i < \infty$ . The bounds below depend on the quality of our importance sample estimators as gauged by

$$b_n = \max_{\pi \in \mathcal{S}_{T_n}} \frac{\mathbb{P}[\pi | \mathbf{x}^{(n)}, \boldsymbol{\theta}]}{\mathbb{R}[\pi | \mathbf{x}^{(n)}, \boldsymbol{\theta}]} \quad (35)$$

Because the set  $\mathcal{S}_{T_n}$  is finite,  $\mathbb{P}[\pi | \mathbf{x}^{(n)}, \boldsymbol{\theta}]$  and  $\mathbb{R}[\pi | \mathbf{x}^{(n)}, \boldsymbol{\theta}]$  have finite support, and the maximum is well defined (finite). If  $\mathbb{R}$  matches the target distribution  $\mathbb{P}$  well, then  $b_n$  should not be very large.

There is one other subtlety that we must introduce for our bounds. Because the terms of  $\widehat{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^k)$  have factors  $\log \theta_{i,j}$  and  $\log \theta_{0,i}$ , in practice, we typically bound  $\theta_{i,j}$  and  $\theta_{0,i}$  away from zero to ensure that  $\widehat{Q}$  does not go to  $-\infty$ . This is easily accomplished with a Dirichlet prior, as discussed after the theorem below. Thus, for the theorem, we will assume that  $\theta_{i,j}^k \geq \theta_{\min}$  and  $\theta_{0,i}^k \geq \theta_{\min}$  for some  $0 < \theta_{\min} < |V|^{-1}$ . The upper bound on  $\theta_{\min}$  ensures it is still possible to satisfy the constraints (1).

We have the following finite-sample PAM result for our MCEM algorithm.

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

$$L_n \geq \frac{2N^2 T_n^4 b_n^2 |\log \theta_{\min}|^2}{\epsilon^2} \log \left( \frac{2T_n^2}{1 - (1 - \delta)^{1/T}} \right) \quad (36)$$

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

The proof of Theorem 1 appears in the Appendix. Because  $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}^{k+1}) \geq 0$  by definition, the theorem guarantees that  $\Delta(\widehat{\boldsymbol{\theta}}^{k+1}) > -\epsilon$  with probability greater than  $1 - \delta$ .

*Remark 1:* If the EM algorithm is initialized with  $\theta_{i,j}^0 > 0$  (i.e., all entries initialized with positive values), then all finite iterates will also be bounded away from zero. However, the iterates may tend arbitrarily close to zero, violating the assumption of the theorem. This problem can be resolved by using a Dirichlet prior with  $u_{i,j} = c > 1$ , for all  $i, j$ , effectively adding a bit of mass to all possible transitions (see Section III-F for discussion of priors). For example, taking  $c = 2$  has the effect of assuming one observation of each and every transition. The prior places a small amount of mass on every transition, and results in EM iterates that satisfy the lower bound  $\theta_{\min} := \frac{1}{|V|(1+N)}$ . Recalling the M-step formula using the Dirichlet prior (19) and taking  $u_{0,i} = 2$  for all  $i$  produces

$$\theta_{0,i}^{k+1} = \frac{1 + \sum_{n=1}^N \sum_{t'=1}^{T_n} \bar{\alpha}_{0,t'}^{n,k} \chi_{t',i}^{(n)}}{\sum_{i=1}^{|V|} \left( 1 + \sum_{n=1}^N \sum_{t'=1}^{T_n} \bar{\alpha}_{0,t'}^{n,k} \chi_{t',i}^{(n)} \right)} \geq \frac{1}{|V|(1+N)}$$

where the inequality follows by noting that the minimum of the numerator is 1 and the denominator is bounded above by  $|V|(1+N)$ . The latter can be seen since all the summands of  $\sum_{n=1}^N \sum_{t'=1}^{T_n} \bar{\alpha}_{0,t'}^{n,k} \chi_{t',i}^{(n)}$  lie in the set  $[0, 1]$ , and for each  $n$ , at most one  $\chi_{t',i}^{(n)}$ ,  $t' = 1, \dots, T_n$  is nonzero. A similar bounding argument shows that  $\theta_{i,j}^{k+1} \geq \frac{1}{|V|(1+N)}$  when  $u_{i,j} = 2$  for all  $i, j$ . Observe that the incorporation of the prior does not alter the proof of Theorem 1 since the prior terms (i.e., log of the Dirichlet prior) in  $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}^{k+1})$  and  $\Delta(\widehat{\boldsymbol{\theta}}^{k+1})$  are independent of the sufficient statistics and thus cancel each other out. Note that this choice of prior results in the following requirement on the number of importance samples

$$L_n \geq \frac{2N^2 T_n^4 b_n^2 [\log(|V|(1+N))]^2}{\epsilon^2} \log \left( \frac{2T_n^2}{1 - (1 - \delta)^{1/N}} \right). \quad (37)$$

Finally, we also point out that if two vertices  $i$  and  $j$  do not co-occur in any of the observations, then one can set  $\theta_{i,j}^0 = 0$ , effectively eliminating it from further consideration. This will not affect the EM algorithm or the bounds above. However, if one suspects that the observations do not necessarily reflect all possible paths, then it may be sensible to use the Dirichlet prior in such situations.

Recall that exact E-step computation requires  $T_n!$  operations for the  $n$ th observation. The bound above stipulates that the number of importance samples required is proportional to  $T_n^4 \log T_n^2$ . Generating one importance sample using the causal sampling scheme requires  $T_n$  operations, and thus, the computational complexity of a PAM Monte Carlo update only depends on  $T_n^5 \log T_n^2$ , which clearly demonstrates that the computational complexity of the Monte Carlo E-step depends polynomially on  $T_n$  in comparison to exponential dependence for the exact E-step.

To put this result in perspective, observe that the value of  $L_n$  given by (36) is roughly a factor of  $N$  away from the value

we would expect based on an asymptotic variance calculation. Ignoring constants and log terms, for fixed  $\theta$ , we have

$$\begin{aligned} \text{Var}(\widehat{\Delta}(\theta)) &\approx \text{Var}\left(\sum_{n=1}^N \sum_{t',t''=1}^{T_n} \widehat{\alpha}_{t',t''}^{n,k} + \sum_{n=1}^N \sum_{t'=1}^{T_n} \widehat{\alpha}_{0,t'}^{n,k}\right) \\ &= \sum_{n=1}^N \text{Var}\left(\sum_{t',t''=1}^{T_n} \widehat{\alpha}_{t',t''}^{n,k} + \sum_{t'=1}^{T_n} \widehat{\alpha}_{0,t'}^{n,k}\right) \end{aligned}$$

since independent sets of importance samples are used to calculate sufficient statistics for different observations. It is easily shown that the variance of an individual approximate statistic  $\widehat{\alpha}_{t',t''}^{n,k}$  or  $\widehat{\alpha}_{0,t'}^{n,k}$  decays according to the parametric rate; i.e.,  $\text{Var}(\widehat{\alpha}_{t',t''}^{n,k}) = O(1/L_n)$ . In total, there are  $T_n^2$  sufficient statistics for the  $n$ th observation, and they are all potentially correlated since they are functions of the same set of importance samples. Then, we have

$$\text{Var}(\widehat{\Delta}(\theta)) = O\left(\sum_{n=1}^N \frac{(T_n^2)^2}{L_n}\right).$$

To drive  $\text{Var}(\widehat{\Delta}(\theta))$  down to a constant level, independent of  $N$  and  $T_n$ , we need  $L_n \propto NT_n^4$ . The additional factor of  $N$  in our bound is essentially an artifact from the union bound.

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

In addition to demonstrating that the MCEM algorithm has polynomial computational complexity, this bound gives a useful guideline for determining how many importance samples should be used. However, because they involve worst case analysis, the number of samples dictated by this bound tends to be on the conservative side. For example, in the Internet experiments described in Section VI,  $N = 249$  and the average path length is 17 hops. Theorem 1 suggests that roughly ten billion 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 2000 samples per observation. Of course, in practice, the only way to know the value of  $b_n$  is to enumerate all permutations, so this bound cannot be used as an explicit guideline.

## VI. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our NICO algorithm on simulated data and on data gathered from the public Internet. In the results reported below, network reconstructions are obtained by first estimating an initial state distribution and probability transition matrix via the EM algorithm. Then, we compute the most likely order of each observation according to the inferred model and use this ordering to reconstruct a feasible network. The EM algorithm cannot be guaranteed to converge to a global maximum (the marginal log-likelihood is not concave) and there may even be multiple global maxima. To address this

issue, we rerun the EM algorithm from multiple random initializations and report the collective results.

We compare the performance of our algorithm with that of the frequency method (FM), defined in [1] and mentioned in Section I. The FM also reconstructs a network topology by estimating an order of the vertices in each observation. This method individually determines each path ordering independently by sorting the elements in the path according to how correlated each vertex is with the endpoints of the path. It is possible that multiple vertices may receive identical FM scores, in which case their sorting would be arbitrary (one could exchange elements with identical scores without violating the FM criteria). In fact, we observe this phenomenon in many of our experiments. Ties are resolved by choosing a random order for elements with identical scores. Multiple restarts are also performed using the FM, yielding a collection of feasible 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 vertices 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 that do not exist in the true network) and the number of false negatives (edges in the true network not appearing in the reconstructed network).

### A. Simulated Networks

Our synthetic data are obtained as described next. 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 [27]. Groups of nodes are randomly chosen as sources and destinations, transmission paths are generated between each source-destination pair according to either a shortest path or random routing model, and then, co-occurrence observations are formed from each path. We keep the number of sources fixed at five and vary the number of destinations between five and 40, to see how the number of observations affects performance. Each experiment is repeated on 100 different topologies, rerunning both NICO and the FM from ten different random seeds per topology. Exact E-step calculation is used for observations with  $T_n \leq 12$ , and causal importance sampling (2000 samples) is used for longer paths. The longest observation in our data was obtained by random routing and has  $T_n = 19$  (notice that  $19! \approx 10^{17}$ ). No prior is used in any of the results reported here. In our experience, we found little practical difference between the MLE and the MAP estimate based on a Dirichlet prior with  $u_{i,j} = 2$ , as discussed in Remark 1.

Fig. 2 plots edge symmetric difference performance for synthetic data generated using: 1) shortest path routing and 2) random routing. The edge symmetric difference error is computed between the inferred network and the graph obtained from correctly ordered observations. Of the ten solutions corresponding to different NICO initializations, we use the one based on parameter estimates yielding the highest likelihood

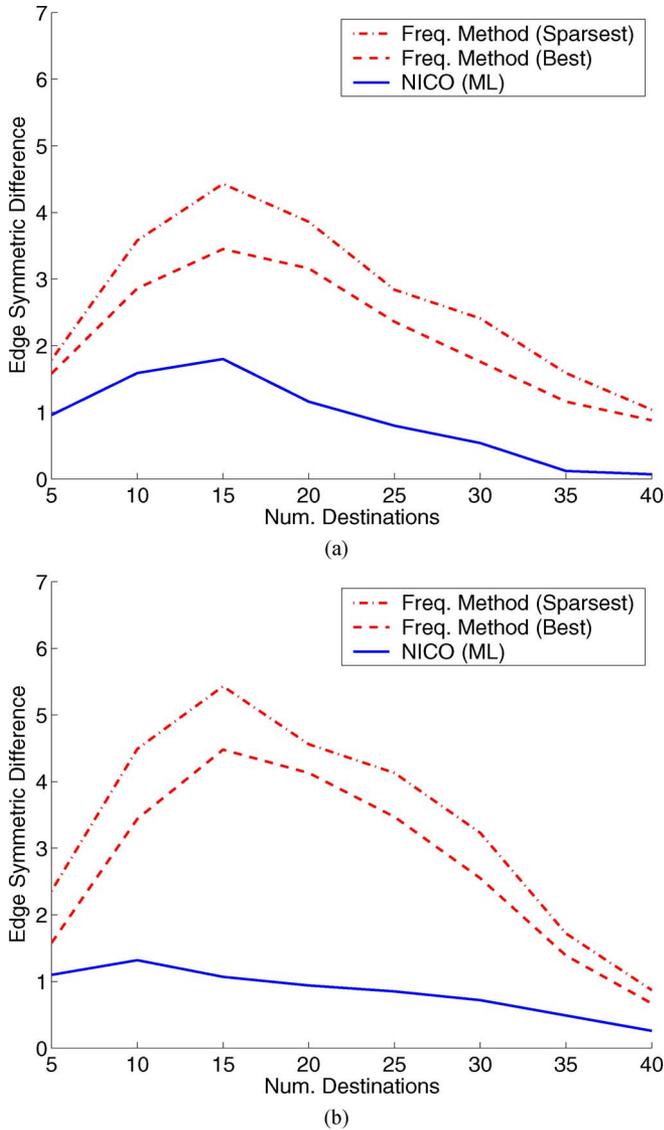


Fig. 2. Edge symmetric differences between inferred networks and the network one would obtain using co-occurrence measurements arranged in the correct order. Performance is averaged over 100 different network configurations. For each configuration, ten NICO and FM solutions are obtained via different initializations. We then choose the NICO solution yielding the largest likelihood, and compare with both the sparsest and clairvoyant best FM solution. (a) Shortest path routes. (b) Random routes.

score. For this simulation, the most likely NICO solution also always resulted in the best edge symmetric difference error.

The FM does not provide a similar mechanism for ranking different solutions. A possible heuristic would be to choose the sparsest solution (with fewest edges). The figures show performance for both this heuristic, and clairvoyantly choosing the best (lowest error) solution of the ten. In fact, using the sparsest solution does better than just choosing an FM solution at random but not as well as using the clairvoyant best. In these simulations, NICO consistently outperforms the FM.

Notice that both algorithms exhibit their worst performance at an intermediate number of destinations. When very few destinations are used, the measured topology closely resembles a tree, regardless of the underlying routing mechanism. Relative

frequencies of co-occurrence accurately reflect the network distance of each internal vertex from the path endpoints. At the other extreme, when many destinations are used, there is significant overlap among the co-occurrence observations, which aids in localizing vertices. In general, the FM seems to be much more sensitive to the amount of data available.

As expected, the FM generally performs better on shortest path data than it does on random routes. When routes are generated randomly, the corresponding topology is less tree-like and pairwise co-occurrence frequencies do not reflect network distances. Because NICO is not based on a particular routing paradigm, it performs similarly in both scenarios, possibly even a little better when routing is random.

### B. 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 from sources at the University of Wisconsin-Madison, the Instituto Superior Técnico, Lisbon, Portugal, and Rice University, Houston, TX, to 83 servers affiliated with corporations, universities, and governments around the world. Our motivation for using this type of data is twofold. 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 ground truth network contains a total of 1105 nodes and 1317 edges, and the longest observed path has length 27.

For this data set, we rerun FM and NICO each from 50 random initializations and look at performance across all solutions rather than focusing on the maximum likelihood or clairvoyant best. The exact E-step is used to compute sufficient statistics for paths of up to nine hops. For paths longer than nine hops, we use the causal importance sampling described in Section IV-A, with 2000 samples per observation.

Minimum, median, and maximum edge symmetric difference errors are shown in Fig. 3. Both algorithms have seemingly high error rates, as there are roughly 1300 links in the true network. However, keep in mind that both algorithms are attempting to fill in the entries of a roughly  $1100 \times 1100$  matrix. For 50 networks constructed by choosing a random order for the elements of each observation, the average edge symmetric difference error was 4300, so both algorithms are indeed doing considerably better than random guessing. NICO performance is again noticeably better than that of the FM; 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).

Fig. 4 shows statistics for the number of edges in the reconstructed networks. There is an interesting correlation between the number of edges and reconstruction accuracy in this example. As seen above, the typical NICO reconstruction is more

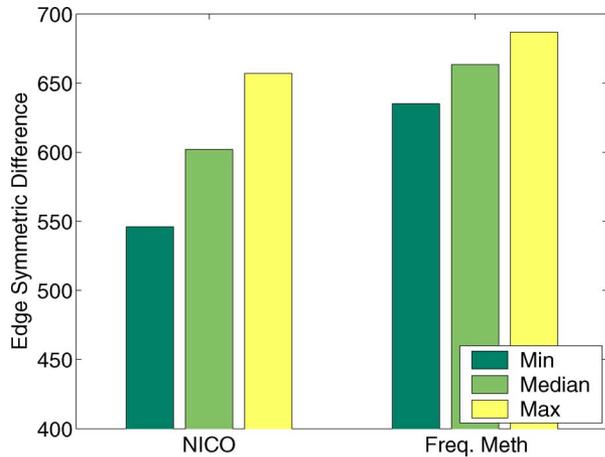


Fig. 3. Edge symmetric difference error comparison of NICO and FM on Internet data. The reported values come from 50 random initializations of each algorithm.

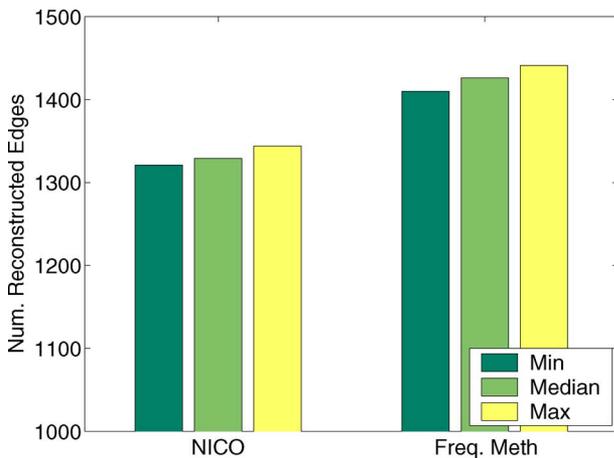


Fig. 4. Number of edges in networks reconstructed using each method. The median number of edges per reconstruction is 1329 for NICO and 1426 for FM. The true network has 1317 edges, and so it appears that NICO does a better job of capturing the complexity of the true network.

accurate, in terms of edge errors, than an FM reconstruction. NICO also consistently returns a sparser estimate. The median number of links in a NICO reconstruction is 1329, whereas the median number of links in an FM reconstruction is 1426. There are 1317 edges in the true network, so in this sense the NICO reconstructions more accurately reflect the inherent level of complexity in the true network.

Marginal log-likelihood values for each of the 50 NICO estimates are depicted in Fig. 5. The marginal log-likelihood, given by (4), is the cost function being optimized by the EM algorithm. In contrast to the experiments with simulated data reported above, there is no exact correlation between higher marginal likelihood values and lower edge symmetric difference error for this example. The topology with the highest likelihood value results in an edge symmetric difference error of 627. This is better than the clairvoyant best FM error, but only average for NICO. The three repetitions that 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.

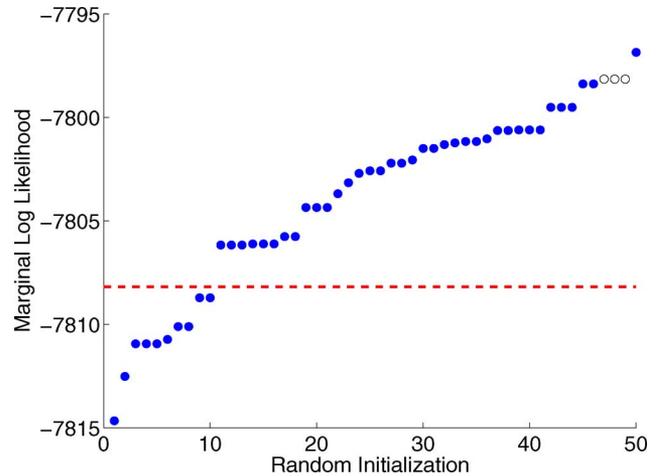


Fig. 5. Marginal log-likelihood values for different random initializations of NICO, sorted in ascending order. The three hollow circles correspond to the solutions that achieve the lowest edge symmetric difference error of all NICO trials. The dashed line shows the marginal log-likelihood value computed using the true path orders to estimate a Markov transition matrix. Most NICO solutions have higher marginal log-likelihood than the true topology, suggesting that our generative model does not accurately describe Internet topology data.

The dashed line shows the likelihood value based on a transition matrix estimated using the true path orders as measured by traceroute. Notice that the majority of the NICO solutions 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.

## VII. DISCUSSION AND ONGOING RESEARCH

This paper presents a novel approach to network inference 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 random walks on the underlying graph structure. Co-occurrence observations are modeled as i.i.d. samples of the random walk subjected to a random permutation that accounts for the lack of observed path order. Treating the random permutations as latent variables we derive an EM algorithm for efficiently computing maximum-likelihood or maximum *a posteriori* estimates of the random walk parameters (initial state distribution and transition matrix).

The complexity of the EM algorithm is dominated by the E-step calculation, which is exponential in the length of the longest transmission path. To handle large networks, we describe fast approximation methods based on importance sampling and Monte Carlo techniques. We derive concentration-style bounds on the accuracy of the Monte Carlo approximation. These bounds prescribe how many importance samples must be used to ensure a monotonic increase in the log-likelihood, thereby guaranteeing convergence of the algorithm with high probability. The resulting MCEM computational complexity only depends polynomially on the length of the longest path.

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. The parameter estimates produced by the EM algorithm may be useful for other tasks such as guiding an expert to alternative reconstructions by assigning likelihoods to different permutations, or predicting unobserved paths through the network as in [12]. One could also 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  $\theta_{i,j}$  can be interpreted as estimates of the probability that a transmission will be passed from vertex  $i$  to  $j$ , conditioned on the path reaching  $i$ ; that is,  $\theta_{i,j} = \mathbb{P}[w_{k+1} = j | w_k = i]$ . In particular, they are *not* estimates of the probability of a link existing from  $i$  to  $j$ . Since  $\theta$  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  $\mathbb{P}[w_k = i, w_{k+1} = j]$  via Bayes theorem

$$\mathbb{P}[w_{k+1} = j | w_k = i] = \frac{\mathbb{P}[w_k = i, w_{k+1} = j]}{\mathbb{P}[w_k = i]}$$

where  $\mathbb{P}[w_k = i]$  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. In our experiments, we found that our current model leads to reasonable Internet reconstructions, but we feel there is room for improvement. For example, the structure of Internet paths may depend strongly on the destination of the traffic. In fact, one could partition the co-occurrence data into source-dependent (or destination-dependent) subsets and learn different Markov models for each subset (see, e.g., [28]). However, if two or more sources (respectively, destinations) have similar routes, then one could potentially obtain a better overall estimate by pooling observations from the sources. We are currently investigating models based on “mixtures of random walks” to account for this added level of dependency. Nevertheless, although the source-dependent model more accurately reflects how routing is performed in actual communication systems, there are scenarios where a single transition matrix estimate is preferable. For example, a more holistic characterization of network routing is valuable if one is interested in predicting the route between a source and destination that was not previously observed, or if one is interested in predicting the endpoints of a route given only the activated internal routers [12].

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 may be significantly increased when more general structures are considered.

In this paper, we have also restricted our attention to noise-free observations. We are also interested in 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.

## APPENDIX PROOF OF THEOREM 1

There are two main steps in the proof of Theorem 1. First, we derive a concentration inequality for the importance sample approximations  $\hat{\alpha}_{t',t''}^{n,k}$  and  $\hat{\alpha}_{0,t'}^{n,k}$ . Then, we use the inequality to construct a bound for  $\hat{\Delta}(\hat{\theta}^{k+1}) - \Delta(\theta^{k+1})$ .

Recall the expressions (21) and (22) for importance sample approximations calculated in the Monte Carlo E-step. Both are of the general form  $\hat{\mu}_L = \frac{\sum_{i=1}^L Z(\pi_i)X(\pi_i)}{\sum_{i=1}^L Z(\pi_i)}$ , where  $Z : \mathcal{S}_T \rightarrow [0, b]$  and  $X : \mathcal{S}_T \rightarrow \{0, 1\}$ , and they are approximating  $\mu = \frac{\sum_{\pi \in \mathcal{S}_T} X(\pi) \mathbb{P}[\pi | \mathbf{x}, \theta]}{\sum_{\pi \in \mathcal{S}_T} \mathbb{P}[\pi | \mathbf{x}, \theta]}$ . The permutations  $\pi_1, \dots, \pi_L$  are i.i.d. samples from the distribution  $\mathbb{R}[\pi | \mathbf{x}, \theta]$ . Note that  $\mathbb{E}[\hat{\mu}_L] \neq \mu$ , so standard concentration results, such as Hoeffding’s inequality or McDiarmid’s bounded-differences inequality, do not directly apply; e.g., consider the case  $L = 1$

$$\mathbb{E} \left[ \frac{Z(\pi_1)X(\pi_1)}{Z(\pi_1)} \right] = \sum_{\pi \in \mathcal{S}_T} X(\pi) \mathbb{R}[\pi | \mathbf{x}, \theta] \quad (38)$$

$$\neq \sum_{\pi \in \mathcal{S}_T} X(\pi) \mathbb{P}[\pi | \mathbf{x}, \theta]. \quad (39)$$

We can, however, show that  $\hat{\mu}_L$  yields an asymptotically consistent estimate of  $\mu$ . Observe that

$$\mathbb{E}[Z(\pi_i)] = \sum_{\pi \in \mathcal{S}_T} \frac{\mathbb{P}[\pi | \mathbf{x}, \theta]}{\mathbb{R}[\pi | \mathbf{x}, \theta]} \mathbb{R}[\pi | \mathbf{x}, \theta] \quad (40)$$

$$= 1 \quad (41)$$

since  $\mathbb{P}$  is a probability distribution on  $\mathcal{S}_T$ , and

$$\mathbb{E}[Z(\pi_i)X(\pi_i)] = \sum_{\pi \in \mathcal{S}_T} \frac{\mathbb{P}[\pi | \mathbf{x}, \theta]}{\mathbb{R}[\pi | \mathbf{x}, \theta]} X(\pi) \mathbb{R}[\pi | \mathbf{x}, \theta] \quad (42)$$

$$= \sum_{\pi \in \mathcal{S}_T} X(\pi) \mathbb{P}[\pi | \mathbf{x}, \theta] \quad (43)$$

$$= \mu. \quad (44)$$

It follows from the strong law of large numbers that  $\hat{\mu}_L \rightarrow \mu$  as  $L \rightarrow \infty$ .

The following finite-sample concentration inequality demonstrates that the approximation error  $\hat{\mu}_L - \mu$  decays exponentially in the number of importance samples  $L$ .

*Proposition 1:* Let  $\{(X_i, Z_i)\}$  be a sequence of i.i.d. 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 set  $\hat{\mu}_L = \frac{\sum_{i=1}^L Z_i X_i}{\sum_{i=1}^L Z_i}$ . Then, with probability greater than  $1 - \delta$ ,

$$\hat{\mu}_L - \mu < \sqrt{\frac{2b^2 \log \frac{2}{\delta}}{L}}. \quad (45)$$

*Proof:* From the definitions of  $Z_i$  and  $X_i$ ,  $Z_i X_i \in [0, b]$ . Applying Hoeffding's inequality [29] yields that for any  $t > 0$

$$\Pr\left(\sum_{i=1}^L Z_i X_i - L\mu \geq Lt\right) \leq e^{-2Lt^2/b^2} \quad (46)$$

and for any  $t > 0$

$$\Pr\left(\sum_{i=1}^L Z_i - L \leq -Lt\right) \leq e^{-2Lt^2/b^2}. \quad (47)$$

Define the event  $E_t = \{\sum_{i=1}^L Z_i X_i - L\mu \geq Lt\} \cup \{\sum_{i=1}^L Z_i - L \leq -Lt\}$ . By the union bound,  $\Pr(E_t) \leq 2e^{-2Lt^2/b^2}$  for any  $t > 0$ . The complement of  $E_t$  implies that for  $t \in (0, 1)$

$$\hat{\mu}_L - \mu = \frac{\sum_{i=1}^L Z_i X_i - L\mu}{\sum_{i=1}^L Z_i} + \frac{L\mu}{\sum_{i=1}^L Z_i} - \mu \quad (48)$$

$$< \frac{Lt}{L(1-t)} + \frac{L\mu}{L(1-t)} - \mu \quad (49)$$

$$= \frac{t(1+\mu)}{1-t}. \quad (50)$$

It follows that  $\{\hat{\mu}_L - \mu \geq \frac{t(1+\mu)}{1-t}\} \subseteq E_t$ , and so  $\Pr\left(\hat{\mu}_L - \mu \geq \frac{t(1+\mu)}{1-t}\right) \leq \Pr(E_t) \leq 2e^{-2Lt^2/b^2}$ . Since  $\hat{\mu}_L \leq 1$ , if  $\frac{t(1+\mu)}{1-t} + \mu > 1$ , then  $\Pr\left(\hat{\mu}_L - \mu \geq \frac{t(1+\mu)}{1-t}\right) = 0$ , and the proposition holds trivially. Thus, without loss of generality, we consider the case  $\frac{t(1+\mu)}{1-t} + \mu \leq 1$ , or equivalently,  $t \leq (1 - \mu)/2$ . This restriction on  $t$  implies  $\frac{t(1+\mu)}{1-t} \leq 2t$ , and so we have  $\Pr(\hat{\mu}_L - \mu < 2t) > 1 - 2e^{-2Lt^2/b^2}$ . Set  $\delta = 2e^{-2Lt^2/b^2}$  to obtain the desired result.  $\square$

We apply Proposition 1 to the Monte Carlo approximations  $\{\hat{\alpha}_{t',t''}^{n,k}\}$  and  $\{\hat{\alpha}_{0,t'}^{n,k}\}$  as follows. Recall that the Monte Carlo weights are bounded according to  $z_i \in [0, b_n]$ , with  $b_n$  as defined in (35). Define the event

$$B_{\delta',L_n}^n = \left( \bigcup_{\substack{t',t''=1 \\ t' \neq t''}}^{T_n} \left\{ \hat{\alpha}_{t',t''}^{n,k} - \bar{\alpha}_{t',t''}^{n,k} \geq \sqrt{\frac{2b_n^2 \log \frac{2}{\delta'}}{L_n}} \right\} \right) \cup \left( \bigcup_{t'=1}^{T_n} \left\{ \hat{\alpha}_{0,t'}^{n,k} - \bar{\alpha}_{0,t'}^{n,k} \geq \sqrt{\frac{2b_n^2 \log \frac{2}{\delta'}}{L_n}} \right\} \right).$$

This is a union over  $2\binom{T_n}{2} + T_n = T_n^2$  events, each of which holds with probability at most  $\delta'$  according to Proposition 1. By

the union bound, it follows that  $\Pr(B_{\delta',L_n}^n) \leq T_n^2 \delta'$ . Next, let  $C_{\delta',L_n}^n$  denote the event that

$$\sum_{t',t''=1}^{T_n} \left( \hat{\alpha}_{t',t''}^{n,k} - \bar{\alpha}_{t',t''}^{n,k} \right) + \sum_{t'=1}^{T_n} \left( \hat{\alpha}_{0,t'}^{n,k} - \bar{\alpha}_{0,t'}^{n,k} \right) \geq T_n^2 \sqrt{\frac{2b_n^2 \log \frac{2}{\delta'}}{L_n}}$$

and observe that  $C_{\delta',L_n}^n \subseteq B_{\delta',L_n}^n$ , therefore  $\Pr(C_{\delta',L_n}^n) \leq \Pr(B_{\delta',L_n}^n) \leq T_n^2 \delta'$ . Let  $\delta'' = T_n^2 \delta'$  and let  $L > 0$  be a value to be determined later. For each  $n = 1, \dots, N$ , set

$$L_n = \frac{2LT_n^4 b_n^2 \log \frac{2T_n^2}{\delta''}}{\log \frac{1}{\delta''}} \quad (51)$$

so that

$$T_n^2 \sqrt{\frac{2b_n^2 \log \frac{2}{\delta'}}{L_n}} = T_n^2 \sqrt{\frac{2b_n^2 \log \frac{2T_n^2}{\delta''}}{L_n}} = \sqrt{\frac{\log \frac{1}{\delta''}}{L}}. \quad (52)$$

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

$$\sum_{t',t''=1}^{T_n} \left( \hat{\alpha}_{t',t''}^{n,k} - \bar{\alpha}_{t',t''}^{n,k} \right) + \sum_{t'=1}^{T_n} \left( \hat{\alpha}_{0,t'}^{n,k} - \bar{\alpha}_{0,t'}^{n,k} \right) < \sqrt{\frac{\log \frac{1}{\delta''}}{L}}. \quad (53)$$

Recall that  $\chi_{t',i}^{(n)}$  are indicator variables satisfying  $\sum_{i,j=1}^{|V|} \chi_{t',i}^{(n)} \chi_{t',j}^{(n)} = 1$  and  $\sum_{i=1}^{|V|} \chi_{t',i}^{(n)} = 1$ . Multiplying each term in (53) by the appropriate sum of indicators, rearranging terms, and recalling that importance sample estimates for different observations are statistically independent, we have that with probability greater than  $(1 - \delta'')^T$ ,

$$\bigcap_{n=1}^N \left\{ \sum_{t',t''=1}^{T_n} \sum_{i,j=1}^{|V|} \left( \hat{\alpha}_{t',t''}^{n,k} - \bar{\alpha}_{t',t''}^{n,k} \right) \chi_{t',i}^{(n)} \chi_{t',j}^{(n)} + \sum_{t'=1}^{T_n} \sum_{i=1}^{|V|} \left( \hat{\alpha}_{0,t'}^{n,k} - \bar{\alpha}_{0,t'}^{n,k} \right) \chi_{t',i}^{(n)} < \sqrt{\frac{\log \frac{1}{\delta''}}{L}} \right\}$$

which implies that with probability greater than  $(1 - \delta'')^N$ ,

$$\sum_{n=1}^N \sum_{t',t''=1}^{T_n} \sum_{i,j=1}^{|V|} \left( \hat{\alpha}_{t',t''}^{n,k} - \bar{\alpha}_{t',t''}^{n,k} \right) \chi_{t',i}^{(n)} \chi_{t',j}^{(n)} + \sum_{n=1}^N \sum_{t'=1}^{T_n} \sum_{i=1}^{|V|} \left( \hat{\alpha}_{0,t'}^{n,k} - \bar{\alpha}_{0,t'}^{n,k} \right) \chi_{t',i}^{(n)} < N \sqrt{\frac{\log \frac{1}{\delta''}}{L}}.$$

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

$$\sum_{n=1}^N \sum_{t',t''=1}^{T_n} \sum_{i,j=1}^{|V|} \left( \hat{\alpha}_{t',t''}^{n,k} - \bar{\alpha}_{t',t''}^{n,k} \right) \chi_{t',i}^{(n)} \chi_{t',j}^{(n)} |\log \theta_{\min}| + \sum_{n=1}^N \sum_{t'=1}^{T_n} \sum_{i=1}^{|V|} \left( \hat{\alpha}_{0,t'}^{n,k} - \bar{\alpha}_{0,t'}^{n,k} \right) \chi_{t',i}^{(n)} |\log \theta_{\min}| < N |\log \theta_{\min}| \sqrt{\frac{-\log(1 - (1 - \delta)^{1/N})}{L}}. \quad (54)$$

To complete the proof, observe that

$$\begin{aligned} & \widehat{\Delta}(\widehat{\theta}^{k+1}) - \Delta(\widehat{\theta}^{k+1}) \\ &= \sum_{n=1}^N \sum_{i,j=1}^{|V|} \sum_{t',t''}^{T_n} \left( \widehat{\alpha}_{t',t''}^{n,k} - \bar{\alpha}_{t',t''}^{n,k} \right) \\ & \quad \times \chi_{t',i}^{(n)} \chi_{t'',j}^{(n)} \left( \log \widehat{\theta}_{i,j}^{k+1} - \log \theta_{i,j}^k \right) \\ &+ \sum_{n=1}^N \sum_{i=1}^{|V|} \sum_{t'=1}^{T_n} \left( \widehat{\alpha}_{0,t'}^{n,k} - \bar{\alpha}_{0,t'}^{n,k} \right) \chi_{t',i}^{(n)} \left( \log \widehat{\theta}_{0,i}^{k+1} - \log \theta_{0,i}^k \right). \end{aligned} \quad (55)$$

By assumption,  $\theta_{\min} \leq \theta_{i,j}^k \leq 1$  for each  $(i,j) \in V^2$ . It follows that

$$\log \widehat{\theta}_{i,j}^{k+1} - \log \theta_{i,j}^k \leq -\log \theta_{\min} = |\log \theta_{\min}|. \quad (56)$$

Similarly,  $\log \widehat{\theta}_{0,i}^{k+1} - \log \theta_{0,i}^k \leq |\log \theta_{\min}|$  for each  $i \in V$ . Apply these bounds in (55) to find that the right-hand side of (55) is no greater than the left-hand side of (54). Set

$$\epsilon = N |\log \theta_{\min}| \sqrt{\frac{1}{L \frac{1-(1-\delta)^{1/N}}{1-\delta}}}. \quad (57)$$

Then,  $\widehat{\Delta}(\widehat{\theta}^{k+1}) - \Delta(\widehat{\theta}^{k+1}) < \epsilon$  with probability greater than  $1 - \delta$ . Solve for  $L$  in (57) and plug the resulting value back into (51) with  $\delta'' = 1 - (1 - \delta)^{1/N}$  to obtain the desired result.

#### ACKNOWLEDGMENT

The authors would like to thank Laura Balzano for her careful reading and comments on this manuscript.

#### REFERENCES

- [1] M. G. Rabbat, J. R. Treichler, S. L. Wood, and M. G. Larimore, "Understanding the topology of a telephone network via internally-sensed network tomography," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Philadelphia, PA, Mar. 2005, vol. 3, pp. 977–980.
- [2] M. Coates, A. O. Hero, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Process. Mag.*, vol. 19, no. 3, pp. 47–65, May 2002.
- [3] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach, *Systems Biology in Practice: Concepts, Implementation and Application*. New York: Wiley, 2005.
- [4] B. O. Palsson, *Systems Biology: Properties of Reconstructed Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [5] S. Wasserman, K. Faust, D. Iacobucci, and M. Granovetter, *Social Network Analysis: Methods and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 1994.
- [6] M. Newman, A. L. Barabasi, and D. J. Watts, *The Structure and Dynamics of Networks*. Princeton, NJ: Princeton Univ. Press, 2006.
- [7] O. Sporns and G. Tononi, "Classes of network connectivity and dynamics," *Complexity*, vol. 7, no. 1, pp. 28–38, 2002.
- [8] O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag, "Organization, development and function of complex brain networks," *Trends Cogn. Sci.*, vol. 8, no. 9, 2004.
- [9] D. Zhu, A. O. Hero, H. Cheng, R. Khanna, and A. Swaroop, "Network constrained clustering for gene microarray data," *Bioinformatics*, vol. 21, no. 21, pp. 4014–4020, 2005.
- [10] Y. Liu and H. Zhao, "A computational approach for ordering signal transduction pathway components from genomics and proteomics data," *BMC Bioinformatics*, vol. 5, no. 158, Oct. 2004.
- [11] 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 Link-Anal. Workshop*, Acapulco, Mexico, Aug. 2003, pp. 22–31.
- [12] D. Justice and A. O. Hero, "Estimation of message source and destination from link intercepts," *IEEE Trans. Inf. Forens. Secur.*, vol. 1, no. 3, pp. 374–385, Sep. 2006.
- [13] D. Heckerman, D. Geiger, and D. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Mach. Learn.*, vol. 20, pp. 197–243, 1995.
- [14] N. Friedman and D. Koller, "Being Bayesian about Bayesian network structure: A Bayesian approach to structure discovery in Bayesian networks," *Mach. Learn.*, vol. 50, no. 1–2, pp. 95–125, 2003.
- [15] M. Teyssier and D. Koller, "Ordering-based search: A simple and effective algorithm for learning Bayesian networks," in *Proc. Conf. Uncertainty Artif. Intell.*, Edinburgh, Scotland, Jul. 2005, pp. 584–590.
- [16] J. Bernardo and A. Smith, *Bayesian Theory*. New York: Wiley, 1994.
- [17] G. C. G. Wei and M. A. Tanner, "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *J. Amer. Statist. Assoc.*, vol. 85, pp. 699–704, 1990.
- [18] W. Jank, "Stochastic variants of the EM algorithm: Monte Carlo, quasi-Monte Carlo and more," in *Proc. Joint Statistical Meetings*, Minneapolis, MN, Aug. 2005.
- [19] G. Fort and E. Moulines, "Convergence of the Monte Carlo expectation maximization for curved exponential families," *Ann. Statist.*, vol. 31, no. 4, pp. 1220–1259, 2003.
- [20] J. G. Booth, J. P. Hobert, and W. S. Jank, "A survey of Monte Carlo algorithms for maximizing the likelihood of a two-stage hierarchical model," *Statist. Model.*, vol. 1, pp. 333–349, 2001.
- [21] C. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.
- [22] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. New York: Springer-Verlag, 2001.
- [23] M. G. Rabbat, M. A. T. Figueiredo, and R. D. Nowak, "Network inference from co-occurrences" Dept. Electr. Comput. Eng., Univ. Wisconsin-Madison, Tech. Rep. ECE-06-02, 2006.
- [24] C. F. J. Wu, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol. 11, no. 1, pp. 95–103, 1983.
- [25] R. A. Boyles, "On the convergence of the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 45, no. 1, pp. 47–50, 1983.
- [26] B. S. Caffo, W. Jank, and G. L. Jones, "Ascent-based Monte Carlo EM," *J. Roy. Statist. Soc. B*, vol. 67, no. 2, pp. 235–252, 2005.
- [27] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
- [28] D. Justice and A. Hero, "Online methods for network endpoint localization," *IEEE Trans. Inf. Theory*, Apr. 2007, submitted for publication.
- [29] W. J. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Statist. Assoc.*, vol. 58, pp. 713–721, 1963.