Inference with Aggregate Data in Probabilistic Graphical Models: An Optimal Transport Approach

Rahul Singh, Isabel Haasler, Qinsheng Zhang, Johan Karlsson, and Yongxin Chen

***Abstract*—We consider inference (filtering) problems over probabilistic graphical models with aggregate data generated by a large population of individuals. We propose a new efficient belief propagation type algorithm over tree graphs with polynomial computational complexity as well as a global convergence guarantee. This is in contrast to previous methods that either exhibit prohibitive complexity as the population grows or do not guarantee convergence. Our method is based on optimal transport, or more specifically, multi-marginal optimal transport theory. In particular, we consider an inference problem with aggregate observations, that can be seen as a structured multi-marginal optimal transport problem where the cost function decomposes according to the underlying graph. Consequently, the celebrated Sinkhorn/iterative scaling algorithm for multi-marginal optimal transport can be leveraged together with the standard belief propagation algorithm to establish an efficient inference scheme which we call Sinkhorn belief propagation (SBP). We further specialize the SBP algorithm to cases associated with hidden Markov models due to their significance in control and estimation. We demonstrate the performance of our algorithm on applications such as inferring population flow from aggregate observations. We also show that in the special case where the observations are generated by a single individual, our algorithm naturally reduces to the standard belief propagation algorithm.**

I. Introduction

Filtering problems can be more generally posed as inference problems in probabilistic graphical models (PGMs) [1] in a large number of applications including robot localization and mapping, object tracking, and control [2], [3]. PGMs provide a powerful framework for modeling the dependence and relations between probabilistic quantities and probabilistic inference using PGMs have been widely used in signal processing, computer vision, computational biology, and many other real-world applications [1], [4], [5]. During the last decades, many inference algorithms have been proposed, among which the belief propagation (BP) algorithms [6], [7], [8] have been extremely effective and successful. The standard PGM framework and the associated inference algorithms are suitable for modeling of distinguishable/labeled individuals as in most standard applications. In fact, many standard filtering algorithms such as Kalman filter are instances of the BP algorithm operating in a special graphical model known as hidden Markov model (HMM) [9]. The inference problem in HMMs is a filtering problem that estimates dynamically changing unobservable (hidden) states from a sequence of noisy observed data.

Recently, there has been growing interest in applications involving a large population of individuals, e.g., in animal migration and crowd estimation, where data about individuals are not available. Instead, aggregate population-level observation in the form of counts or contingency tables are provided [10], [11], [12], [13], [14]. A distinct feature of this setting is that the individuals are no longer distinguishable to each other. This restriction in the observations could be due to privacy, security, or economic reasons. For example, in tourist flow analysis, individual trajectories may not be readily accessible, but the number of people in a given area can typically be counted using video surveillance or electronic gates. Similarly, it is much easier to obtain the population sizes of bird migration than tracking the trajectory of each bird. Furthermore, in applications related to epidemiology or other diseases, often only aggregate data is available due to privacy issue; normally patient data needs to be anonymized. In this setting with aggregate observations, the traditional inference algorithms such as BP in PGMs which heavily depend on individual observations, are thus not directly applicable. The development of reliable and efficient aggregate inference algorithms is of great importance and necessity.

The collective graphical model (CGM) introduced in [13] is a recent framework for filtering[1] (inference) and learning with aggregate data. A CGM is a graphical model that describes the histograms of individuals directly. Using this model, it was proved that the complexity of traditional inference algorithms for exact inference scales at least polynomially as the population grows [15]. To circumvent this difficulty, they proposed an approximate maximum a posteriori (MAP) formulation, which approximates the marginal inference solution well, especially when the size of the population is large [15]. Under some proper assumptions, the approximate MAP formulation is a convex optimization problem and the problem dimension is independent of the population size. To further accelerate the inference, they proposed the non-linear belief propagation (NLBP) algorithm [16]. The NLBP algorithm is a message-passing type algorithm relying on interactions between graph nodes. Despite of its similarity to the BP algorithm [6], NLBP suffers from instability and lack of convergence. Indeed, no convergence guarantee has been established so far [16]. Bethe-RDA is another existing algorithm for aggregate inference based on regularized dual averaging (RDA) [17]. It is a type of proximal gradient descent algorithm that exhibits convergence guarantee.

The goal of this paper is to establish an efficient and reliable algorithm for filtering with aggregate observations described by the precise distribution of the observation variables. This observation model is different from the one used for NLBP and Bethe-RDA and neither of these two algorithms works for our problems; see Figure 3b and the associated explanations in Section III for more discussions on the observation models. We build on the CGM framework and develop such an aggregate inference algorithm. Our algorithm is based on multi-marginal optimal transport (MOT) theory [18], [19], [20], [21], [22], which involves the transport among multiple marginal distributions. MOT is a generalization of the classical optimal transport (OT) problem [23] of Monge and Kantorovich to find a transport plan from a source distribution to a target one that minimizes the total transport cost. Within the MOT framework, the aggregate observations are viewed as fixed given marginal distributions. We show that the aggregate inference problem in CGM reduces to the special case

This work was supported by the Swedish Research Council (VR), grant 2014-5870, KTH Digital Futures, and the NSF under grant 1901599 and 1942523.

R. Singh, Q. Zhang and Y. Chen are with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA. {qzhang419,rasingh,yongchen}@gatech.edu

I. Haasler and J. Karlsson are with the Division of Optimization and Systems Theory, Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden. haasler@kth.se, johan.karlsson@math.kth.se

R. Singh, I. Haasler, and Q. Zhang contribute equally to this paper.

[1]We use the terms "filtering" and "inference" interchangeably in the paper.

of entropic regularized formulation of MOT with marginals specified by these aggregate observations. Thanks to this equivalence, the aggregate inference problem can be solved by the popular Sinkhorn a.k.a., iterative scaling algorithm [24], [25], [26], [20]. The Sinkhorn algorithm has the advantages of being extremely easy to implement and parallelize, and has global convergence guarantee [25], [20]. We show that the Sinkhorn algorithm for aggregate inference can be further accelerated by leveraging the underlying graphical structure of the inference problems with aggregate observations; a key projection step in the Sinkhorn algorithm, which could be potentially expensive, can be realized efficiently by standard BP for tree graphical models. This accelerated version of our algorithm is named Sinkhorn belief propagation (SBP).

SBP exhibits convergence guarantees when the underlying graph is a tree. The contributions of the paper are summarized as follows.

- We discover an equivalent relation between OT theory and inference/filtering problems with aggregate observations;
- Based on OT theory and belief propagation, we propose an efficient marginal inference/filtering algorithm with aggregate data that has a global convergence guarantee;
- We study the filtering problem in collective HMMs and establish connections between collective and standard filtering problems;
- We demonstrate the performance of our algorithm on applications such as inferring population flow from aggregate observations.

Related Work: Early works on aggregate data focused on learning of the parameters of the underlying models. For example, [10], [11], [12] studied the modeling of a single Markov chain by maximizing the aggregate posterior. More recent learning methods from aggregate data include [27], [14]. Since the formalism of CGMs by [13] there have been multiple works on inference for aggregate data. The complexity of exact inference in CGMs has been investigated in [15] and an approximate MAP formulation has been proposed in the same paper. The non-linear belief propagation algorithm [16] is a message passing type algorithm for approximate MAP inference in CGMs, but it does not have a convergence guarantee. The learning of a Markov chain within the CGM framework has been presented in [28]. On the other hand, the application of OT theory in filtering and estimation problems have been investigated in [29], [30], [31], [32]. Another closely related problem is the Schrödinger bridge problem [33], [34], [35], which is essentially equivalent to an entropic OT problem. Our work is also closely related to linear/nonlinear filtering [36], [37], [38], [3], [39], [40], which is widely studied in the control community. To some degree, our framework can be viewed as an extension of standard filtering theory to the setting with aggregate observations. More recently, built on this work, in [41] we developed a sliding window implementation of the collective forward-backward algorithm for aggregate filtering over HMMs to approximate the solution more efficiently. It is also important to note that our work is not aimed at improving the computational efficiency of solving MOT problems. There are multiple works towards this direction including [42], [43]. Instead, the focus of this paper is on solving the aggregate filtering problems via formulating it as an MOT problem.

The rest of the article is organized as follows. In Section II, we briefly discuss related background including PGMs, BP, CGMs, and MOT. We present our main results and the Sinkhorn belief propagation algorithm (Algorithm 4) in Section III. In Section IV, we specialize SBP to HMMs and obtain the collective forward-backward algorithm. It is followed by the experimental results in Section V and conclusion in Section VI.

II. Background

In this section, we briefly present relevant background on the components of our method which includes probabilistic graphical models, collective graphical models, and multi-marginal optimal transport.

A. Probabilistic Graphical Models

Probabilistic graphical models [1] are graph-based representations of a collection of random vectors that capture the conditional dependencies between them. A PGM is associated with an underlying graph $G = (V, E)$ where V and E denotes the set of vertices and edges respectively. Each node $i \in V$ corresponds to a random variable X_i which can be either discrete or continuous, though we consider the setting with discrete random variables throughout. The random variable at each node takes values from the same[2] finite set $\mathcal{X}$, with cardinality $|\mathcal{X}| = d$. Assuming that the underlying graph is undirected with $J = |V|$ nodes, the probability of the PGM is given by

$$p(\mathbf{x}) = p(x_1, x_2, \dots, x_J) = \frac{1}{Z} \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j), \tag{1}$$

where $\mathbf{x} = \{x_1, \dots, x_J\}$ is a particular assignment to the corresponding random variables, ψ_{ij} are known as edge potentials and Z is a normalization constant. The edge potential ψ_{ij} describes the correlation between the random variables X_i and X_j. For instance, the joint probability distribution of the four random variables in the PGM in Figure 1 factorizes as

$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z}\psi_{12}(x_1, x_2)\psi_{23}(x_2, x_3)\psi_{24}(x_2, x_4).$$

Occasionally, (local) node potentials $\phi_i(x_i)$ induced by, e.g., measurements, are also included in (1) to define the joint probability, but they can always be absorbed into the edge potentials [1]. Thus, for the sake of simplicity and without loss of generality, we adopt the probability structure (1).

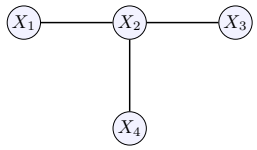


Fig. 1: An example PGM.

In this paper, we restrict our attention to PGMs with undirected underlying graphs. Indeed, a large class of PGMs are associated with directed graphs, including hidden Markov models (HMMs) which are widely used in the control and estimation community, however these directed models can be converted to undirected ones using a technique known as *moralization* [4]. The purpose of moralization is to ensure that all the dependencies implied by the local conditional distributions of the directed graph are captured in the corresponding moralized undirected graph. More specifically, the equivalent moralized graph of a directed graph is formed by converting all edges in the graph into undirected ones and adding additional edges between all pairs of non-adjacent nodes with a common child node. As an example, consider the directed graph in Figure 2a with three directed edges. Each link (edge) is associated with a conditional probability. This directed graph can be converted to an undirected graph via moralization as depicted in Figure 2b. Since no two variables have a common child, the moralization does not result in additional edges. The conditional probabilities can be viewed as the edge potentials, e.g., $\psi_{24}(x_2, x_4) = p(x_4|x_2)$.

[2]This is only for the ease of notation. In general, these random variables can take values in different sets.

(a) Directed (b) Undirected (Moralized)

Fig. 2: Moralization of a PGM.

A fundamental problem in PGMs is to estimate the marginals of each variable from the given joint distribution $p(\mathbf{x})$; this is known as the Bayesian inference problem [1]. Belief propagation (BP) [6] is one of the most popular algorithms for accomplishing this task.

Belief Propagation: BP is an effective message-passing type algorithm for Bayesian inference in PGMs. It updates the marginal distribution of each node through communications of beliefs/messages between them. These messages are updated through

$$m_{i\to j}(x_j) \propto \sum_{x_i} \psi_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i), \tag{2}$$

where $m_{i\to j}(x_j)$ denotes the message from variable node i to variable node j, encapsulating the belief of node i on node j. Here, $N(i)$ is the set of neighboring nodes of i, and thus $N(i)\setminus j$ denotes the set of neighbors of i except for j. The messages in (2) are updated iteratively over the graph. When the algorithm converges, the node and edge marginals are given by

$$b_i(x_i) \quad \propto \prod_{k\in N(i)} m_{k\to i}(x_i) \tag{3a}$$

$$b_{ij}(x_i, x_j) \propto \psi_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i) \prod_{\ell\in N(j)\setminus i} m_{\ell\to j}(x_j). \tag{3b}$$

When the graph has no cycles (i.e., tree) it is well-known that the belief propagation algorithm converges globally [7] and the estimated marginal distributions in (3) recover the true marginals exactly. For general graphs with cycles, convergence is not guaranteed but it works well in practice [1].

Note that many time-sequence based filtering algorithms used in control and estimation, including Kalman filter [36], are instances of the BP algorithm operating on Gaussian HMM, a special kind of graphical model (see Section IV for more details).

B. Collective Graphical Models

CGMs were first introduced in [13] as a framework for inference and learning in graphical models with aggregate data. CGMs describe the distribution of the aggregate counts of a population sampled independently from a discrete PGM. Assume all the individuals share the same PGM (1).

Let $X_i^{(m)}$ be the random variable representing the state of the m^{th} individual at node i. To generate the aggregate data, first assume that M independent sample vectors $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(M)}$ are drawn from the individual probability model to represent the individuals in a population. Here, each entry of the vector $\mathbf{x}^{(m)}$ corresponding to node i is $x_i^{(m)}$ that takes one of the d possible states. Let $\mathbf{n}_i \in \mathbb{N}^d$ be the aggregate *node* distribution with entries $n_i(x_i) = \sum_{m=1}^{M} \mathbb{I}[X_i^{(m)} = x_i]$ that count the number of individuals in each state. Here, $\mathbb{I}[\cdot]$ denotes indicator function. Moreover, let $\mathbf{n}_{ij} \in \mathbb{N}^{d\times d}$ be the aggregate edge distributions with entries $n_{ij}(x_i, x_j) = \sum_{m=1}^{M} \mathbb{I}[X_i^{(m)} = x_i, X_j^{(m)} = x_j]$. The vectors $\mathbf{n}_1, \ldots, \mathbf{n}_J$ constitute the aggregate data and the aggregate edge distributions $\mathbf{n}_{ij}$ represent sufficient statistics of the individual model [13]. The collection of all the aggregate node distributions $\mathbf{n}_i$ together with the aggregate edge distributions $\mathbf{n}_{ij}$ is denoted as $\mathbf{n}$, i.e., $\mathbf{n} = \{\mathbf{n}_i, \mathbf{n}_{ij}\}$.

In CGMs [13], the observation noise is modeled explicitly as a conditional distribution $p(\mathbf{y}|\mathbf{n})$ with $\mathbf{y}$ being the aggregate noisy observations that probabilistically depend on the aggregate data $\mathbf{n}$. For instance, for a count n, the associated observation may follow a Poisson distribution $\text{Poisson}(\beta n)$ for some coefficient $\beta > 0$. The goal of inference in CGMs is to estimate $\mathbf{n}$ from the aggregate noisy observations through the posterior distribution $p(\mathbf{n}|\mathbf{y}) \propto p(\mathbf{n})p(\mathbf{y}|\mathbf{n})$, where $p(\mathbf{n})$ is known as the CGM distribution [13] which is derived from the individual model (1). When the underlying graph is a tree, the CGM distribution $p(\mathbf{n})$ equals

$$p(\mathbf{n}) = M! \frac{\prod_{i\in V} \prod_{x_i} ((n_i(x_i)!)^{(d_i-1)}}{\prod_{(i,j)\in E} \prod_{x_i,x_j} n_{ij}(x_i, x_j)!} p(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}), \tag{4}$$

where $d_i = |N(i)|$ is the number of neighbors of node i in G and

$$p(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}) = \frac{1}{Z^M} \prod_{(i,j)\in E} \prod_{x_i,x_j} \psi_{ij}(x_i, x_j)^{n_{ij}(x_i,x_j)}$$

is the joint probability of the entire population. The integer coefficient

$$M! \frac{\prod_{i\in V} \prod_{x_i} ((n_i(x_i)!)^{(d_i-1)}}{\prod_{(i,j)\in E} \prod_{x_i,x_j} n_{ij}(x_i, x_j)!}$$

accounts for the fact that the individuals in aggregate observation are indistinguishable.

The support of the CGM distribution $p(\mathbf{n})$ is such that each entry of $\mathbf{n}$ is an integer and $\mathbf{n}$ satisfies the following constraints

$$\begin{aligned} \sum_{x_i} n_i(x_i) &= M, \qquad\qquad \forall i \in V \\ n_i(x_i) &= \sum_{x_j} n_{ij}(x_i, x_j), \quad \forall i \in V,\ j \in N(i). \end{aligned} \tag{5}$$

Exact inference of $\mathbf{n}$, either maximum a posterior probability estimate or Bayesian inference, based on $p(\mathbf{n}|\mathbf{y})$ is unrealistic for large populations, since the computational complexity increases rapidly as the population size M grows [15]. It was pointed out in [15] that $-\ln p(\mathbf{n}|\mathbf{y})$ can be approximated by (up to a constant addition and multiplication) the CGM free energy

$$F_{\text{CGM}}(\mathbf{n}) = U_{\text{CGM}}(\mathbf{n}) - H_{\text{CGM}}(\mathbf{n}), \tag{6}$$

where $U_{\text{CGM}}(\mathbf{n})$ equals

$$-\sum_{(i,j)\in E} \sum_{x_i,x_j} n_{ij}(x_i, x_j) \ln \psi_{ij}(x_i, x_j) - \ln\ p(\mathbf{y}|\mathbf{n}),$$

and

$$\begin{aligned} H_{\text{CGM}}(\mathbf{n}) = &-\sum_{(i,j)\in E} \sum_{x_i,x_j} n_{ij}(x_i, x_j) \ln n_{ij}(x_i, x_j) \\ &+ \sum_{i\in V} (d_i - 1) \sum_{x_i} n_i(x_i)\ \ln\ n_i(x_i). \end{aligned}$$

After relaxing the constraints that $n_i(x_i), n_{ij}(x_i, x_j)$ are integers and under the assumption that the observation model $p(\mathbf{y} \mid \mathbf{n})$ is log-concave, the resulting problem of minimizing F_{CGM} is a convex optimization problem. This is the approximate MAP [15] framework for CGMs. Note that the problem size of minimizing F_{CGM} is independent of the population size M. Even though the approximate MAP framework is insensitive to population size, its complexity grows rapidly as the number of variables J increases. Two algorithms designed to solve the approximate MAP problems more efficiently

are non-linear belief propagation (NLBP) [16] and Bethe regularized dual averaging (Bethe-RDA) [17].

Non-linear Belief Propagation: The NLBP [16] algorithm addresses the aggregate inference problem by establishing a connection between the Bethe free energy [44] and the objective function (6) for approximate MAP inference in CGMs. It is a message passing type algorithm for aggregate MAP inference. Roughly, it is equivalent to running standard BP on a PGM with edge potentials $\hat{\psi}_{ij}$ which are being updated at each iteration. The steps of the NLBP algorithm are listed in Algorithm 1. Similar to BP, after convergence of the

Algorithm 1 Non-Linear Belief Propagation (NLBP)

Initialize $n_{ij}(x_i, x_j) \propto \psi_{ij}(x_i, x_j),\ \forall (i,j) \in E, \forall x_i, x_j$
repeat
Update the following in any order for all $(i,j) \in E$

$$\begin{aligned} \hat{\psi}_{ij}(x_i, x_j) &= \exp\left(-\frac{\partial U_{\mathrm{CGM}}(\mathbf{n})}{\partial n_{ij}(x_i, x_j)}\right) \\ m_{i\to j}(x_j) &\propto \sum_{x_i} \hat{\psi}_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i) \\ n_{ij}(x_i, x_j) &\propto \hat{\psi}_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i) \prod_{\ell\in N(j)\setminus i} m_{\ell\to j}(x_j) \end{aligned}$$

until convergence

messages in Algorithm 1, the aggregate marginals can be estimated as

$$n_i(x_i) \propto \prod_{k\in N(i)} m_{k\to i}(x_i). \tag{7}$$

One of the major drawbacks of the NLBP algorithm is that it does not exhibit any convergence guarantee [16]. The major factor affecting the convergence of NLBP is the update of the potentials $\hat{\psi}_{ij}$ which requires gradient computations of $p(\mathbf{y}|\mathbf{n})$. These gradients depend on the observation model at hand and might cause the explosion or saturation of potential updates based on the smoothness of the observation model $p(\mathbf{y}|\mathbf{n})$. To stabilize the NLBP to some degree, it was proposed [16] to dampen the estimates $\mathbf{n}$ as $\mathbf{n} = (1-\alpha)\mathbf{n} + \alpha\mathbf{n}^{new}$ in each iteration, where $0 < \alpha \leq 1$ and $\mathbf{n}^{new}$ is the estimate in the current iteration. However, the selection of the parameter α has to be done carefully to ensure appropriate potential updates [16].

Bethe regularized dual averaging: Another algorithm for solving the aggregate inference problem is Bethe-RDA [17] which is inspired by one type of proximal algorithms called regularized dual averaging (RDA) [45]. The Bethe-RDA algorithm has been proven to be faster than NLBP by an order of magnitude in various experiments [17]. Similar to NLBP, in Bethe-RDA, standard BP is used to compute the marginals at each iteration based on modified edge potentials. However, the way to update these potentials in Bethe-RDA is very different. At iteration t, the edge potentials $\Psi_t = \{\psi^t_{ij}(x_i, x_j)\}$ are updated according to

$$\ln\ \Psi_t = \ln\ \Psi - \frac{t}{\beta_t + t}\ \overline{g}_t, \tag{8}$$

where $\Psi = \{\psi_{ij}(x_i, x_j)\}$ is the original potential, β_t is the learning rate, and

$$\overline{g}_t = \frac{t-1}{t}\ \overline{g}_{t-1} - \frac{1}{t}\ \frac{\partial \ln p(\mathbf{y}|\mathbf{n}_{t-1})}{\partial \mathbf{n}}. \tag{9}$$

After updating the potentials, the required marginals $\mathbf{n}_t$ are computed according to the standard BP algorithm. The steps of the Bethe-RDA algorithm are listed in Algorithm 2.

Algorithm 2 Bethe-RDA

Initialize all the marginals $\mathbf{n}_0 = \mathrm{BP}[\Psi]$, and $\bar{g}_0 = 0$
repeat
$\overline{g}_t = \frac{t-1}{t}\ \overline{g}_{t-1} - \frac{1}{t}\ \frac{\partial \ln p(\mathbf{y}|\mathbf{n}_{t-1})}{\partial \mathbf{n}}$
$\ln\ \Psi_t = \ln\ \Psi - \frac{t}{\beta_t + t}\ \overline{g}_t$
$\mathbf{n}_t = \mathrm{BP}[\Psi_t]$
until convergence

C. Multimarginal Optimal Transport

In an MOT [21], [22] problem one aims to find an optimal transport plan among a set of marginal distributions, minimizing an underlying given cost function. We consider the discrete settings where the marginal distributions are described by probability vectors $\mu_j \in \mathbb{R}^d,\ j \in \Gamma \subset \{1, 2, \ldots, J\}$ and denote the cost function and the transport plan by the J-mode tensors $\mathbf{C}, \mathbf{B} \in \mathbb{R}^{d\times d\cdots\times d}$. The Kantorovich formulation of MOT with constraints on a subset of marginals $\Gamma \subset \{1, 2, \ldots, J\}$ reads [46]

$$\begin{aligned} \min_{\mathbf{B}\in\mathbb{R}_+^{d\times\cdots\times d}}\quad & \langle \mathbf{C}, \mathbf{B}\rangle \\ \text{subject to}\quad & P_j(\mathbf{B}) = \mu_j,\ \text{for } j \in \Gamma, \end{aligned} \tag{10}$$

where $\langle \mathbf{C}, \mathbf{B}\rangle = \sum_{i_1,\ldots,i_J} \mathbf{C}_{i_1,\ldots,i_J} \mathbf{B}_{i_1,\ldots,i_J}$, and the projection on the j-th marginal of $\mathbf{B}$ is defined by

$$P_j(\mathbf{B}) = \sum_{i_1,\ldots,i_{j-1},i_{j+1},i_J} \mathbf{B}_{i_1,\ldots,i_{j-1},i_j,i_{j+1},\ldots,i_J}. \tag{11}$$

Note that the standard OT problem with two marginals is a special case of (10) with $J = 2$ and $\Gamma = \{1, 2\}$.

Though (10) is a standard linear programming, it can be computational expensive for large d and J. For faster computations, it was proposed by [26], [20], [47] to add a regularizing entropy term

$$H(\mathbf{B}) = -\sum_{i_1,\ldots,i_J} \mathbf{B}_{i_1,\ldots,i_J} \ln\ (\mathbf{B}_{i_1,\ldots,i_J}) \tag{12}$$

to the objective. This results in the strongly convex problem

$$\begin{aligned} \min_{\mathbf{B}\in\mathbb{R}_+^{d\times\cdots\times d}}\quad & \langle \mathbf{C}, \mathbf{B}\rangle - \epsilon H(\mathbf{B}) \\ \text{subject to}\quad & P_j(\mathbf{B}) = \mu_j,\ \text{for } j \in \Gamma, \end{aligned} \tag{13}$$

where $\epsilon > 0$ is a regularization parameter.

It can be shown that the unique optimal solution to (18) is of the form

$$\mathbf{B} = \mathbf{K} \odot \mathbf{U}, \tag{14}$$

where $\mathbf{K} = \exp(-\mathbf{C}/\epsilon)$ and $\mathbf{U} = u_1 \otimes u_2 \otimes \cdots \otimes u_J$ with $u_j = \mathbf{1}$ ($\mathbf{1}$ denotes the vector of all entries being 1) for $j \notin \Gamma$. Here $\otimes$ denotes tensor product and exp is entry-wise exponential map. The Sinkhorn scheme [48], [24], [26] for finding the vectors u_j, given that they are initialized to be $\mathbf{1}$, is to iteratively update them according to

$$u_j \leftarrow u_j \odot \mu_j ./ P_j(\mathbf{K} \odot \mathbf{U}), \tag{15}$$

for all $j \in \Gamma$. Here $\odot$ and $./$ denote entry-wise multiplication and division, respectively. It is worth noting that the update step (15) in Algorithm 3 is a *scaling* step that ensures that the j-th marginal of the updated tensor $\mathbf{K} \odot \mathbf{U}$ satisfies the constraint in (13), i.e., $P_j(\mathbf{K} \odot \mathbf{U}) = \mu_j$. The steps of Sinkhorn scheme are summarized in Algorithm 3 for future reference. The Sinkhorn algorithm may for instance be derived as Bregman iterations [20] or dual block

Algorithm 3 Sinkhorn Algorithm for MOT

Compute $\mathbf{K} = \exp(-\mathbf{C}/\epsilon)$
Initialize $u_1, u_2, \ldots, u_J$ to $\mathbf{1}$
repeat
  for $j \in \Gamma$ **do**
    $\mathbf{U} \leftarrow u_1 \otimes u_2 \otimes \cdots \otimes u_J$
    $u_j \leftarrow u_j \odot \mu_j ./ P_j(\mathbf{K} \odot \mathbf{U})$
  end for
until convergence

coordinate ascend [46]. It has global convergence guarantee with linear rate [49], [50]. We remark that even though the Sinkhorn algorithm has linear convergence rate, the cost for each update step could be high due to the projection P_j, whose complexity scales exponentially with J.

III. Inference with aggregate data

We consider Bayesian marginal inference problems with aggregate data as in CGMs with a different observation model. We reformulate them into MOT problems and then leverage Sinkhorn algorithm (Algorithm 3) to develop an efficient algorithm for our problems.

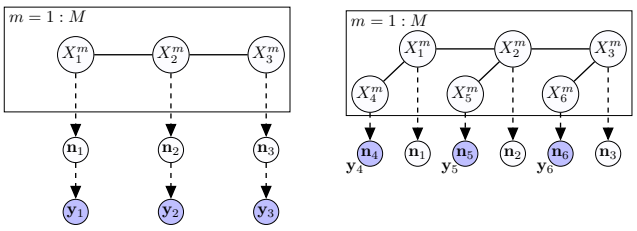
A. Problem formulation

Assume that the graph $G = (V, E)$ encodes the relationships among the node variables $X_1, X_2, \ldots, X_J$ of each individual, which consists of unobserved as well as observed variables, and let $\Gamma \subset V$ be the set of observation nodes. Let the unobserved individual variables take values in a finite set $\mathcal{X}_u$ and the observations come from another finite set $\mathcal{X}_o$, where in general, $\mathcal{X}_u \neq \mathcal{X}_o$. The joint distribution of individual variables is assumed to be factored as (1).

Then, similar to the generative model of aggregate data in CGMs, by drawing M independent samples from the individual model, the aggregate counts corresponding to each node is generated. Therefore, the aggregate data constitute $\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_J$ (here $J = |V|$). We assume that the system is closed [12], i.e., the population size M remains fixed. In such closed settings, the aggregate data can be thought of as probability distributions when **normalized** with the population size. With this setup, when the underlying graph structure is a *tree*, the aggregate variables have the same graph structure as of the individual probability model; this is due to the hyper-Markov property [15]. Now, we have the aggregate distributions constituting $\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_J$ with the underlying structure G. Suppose aggregate observations are made from a subset of nodes $\Gamma \subset V$ and denote these aggregate observation by $\mathbf{y}_i, \forall i \in \Gamma$, then our goal is to infer the aggregate marginals $\mathbf{n}_i,\ \forall i \notin \Gamma$.

Remark 1. *Without loss of generality, we assume that the observation nodes can only be leaves, otherwise, we can always split the underlying tree graph over the observation node and then each subgraph will have this observation node as a leaf.*

Note that in our setting, the observation model is a subset of the underlying graph as opposed to the original CGM setting, where the observation model is treated separately. Figure 3 depicts this difference between the noise models. In [15], the observation model we use here was regarded as exact observation since the aggregate measurements are exact marginal distributions of the associated node variable. We argue that this type of observation can also handle measurement noise. Taking Figure 3 as an example, X_4 is treated as a measurement node of X_1, therefore, the measurement noise is already encoded in the edge potential between them. Indeed, this



(a) (b)

Fig. 3: Different aggregate observation models (shaded nodes represent aggregate observations): (a) CGMs in its original form model the aggregate noisy observations explicitly and (b) in our model, the observation noise is incorporated in the underlying graphical model itself.

is the measurement noise model used in standard HMMs [51]; the measurement noise is captured by the emission probability, which is the edge potential between a hidden state node and an observation node (see Section IV for more details). As a side note, the algorithms developed in [15], [16], [17] do not apply to the cases with "exact observation". Taking the limit of those algorithms designed for "noisy observation" with vanishing noise will cause ill-conditioning issues in the updates of the algorithms.

In the following we derive an optimization problem to find the aggregate marginals $\{\mathbf{n}_i, i \notin \Gamma\}$. The arguments are adopted from the theory of large deviations [52]. In particular, the marginals can be found by maximizing the posterior distribution $p(\mathbf{y}|\mathbf{n}) \propto p(\mathbf{n}, \mathbf{y})$, which is equivalent to minimizing the negative logarithm $-\ln p(\mathbf{n}, \mathbf{y})$. Since we model the observation variables within the graphical model as in Figure 3b, the observation noise is implicitly incorporated in the CGM distribution given by Equation (4).

Using Equation (4), we arrive at the following

$$\begin{aligned} -\ln\ p(\mathbf{n}, \mathbf{y}) = &-\ln\ M! - \sum_{i \in V} (d_i - 1) \sum_{x_i} \ln(n_i(x_i)!) \\ &+ \sum_{(i,j) \in E} \sum_{x_i, x_j} \ln(n_{ij}(x_i, x_j)!) + M \ln\ Z \\ &- \sum_{(i,j) \in E} \sum_{x_i, x_j} n_{ij}(x_i, x_j) \ln\ \psi_{ij}(x_i, x_j). \end{aligned}$$

Again, as in CGM, it is computationally intractable to directly minimize the negative logarithm $-\ln\ p(\mathbf{n}, \mathbf{y})$ due to the presence of factorial terms and $\mathbf{n}$ being integers. By invoking the Stirling approximation $\ln(a!) = a\ \ln\ a - a + \mathcal{O}(\ln\ a)$ as in [15] (see also [30, Prop. 1]), we obtain

$$\begin{aligned} -\ln\ p(\mathbf{n}, \mathbf{y}) = &-\sum_{i \in V} (d_i - 1) \sum_{x_i} n_i(x_i)\ \ln\ n_i(x_i) \\ &+ \sum_{(i,j) \in E} \sum_{x_i, x_j} n_{ij}(x_i, x_j)\ \ln\ n_{ij}(x_i, x_j) \\ &- \sum_{(i,j) \in E} \sum_{x_i, x_j} n_{ij}(x_i, x_j) \ln(\psi_{ij}(x_i, x_j)) \\ &- M \ln(M/Z) + \mathcal{O}(\ln M). \end{aligned}$$

Denote $\hat{\mathbf{n}}_i = \mathbf{n}_i / M$ and $\hat{\mathbf{n}}_{ij} = \mathbf{n}_{ij} / M$ the normalizations of the

aggregate distributions, then it follows

$$-\frac{1}{M}\ln\ p(\mathbf{n},\mathbf{y}) = -\sum_{i\in V}(d_i-1)\sum_{x_i}\hat{n}_i(x_i)\ \ln\ \hat{n}_i(x_i) + \sum_{(i,j)\in E}\sum_{x_i,x_j}\hat{n}_{ij}(x_i,x_j)\ \ln\ \hat{n}_{ij}(x_i,x_j) + \ln Z - \sum_{(i,j)\in E}\sum_{x_i,x_j}\hat{n}_{ij}(x_i,x_j)\ln(\psi_{ij}(x_i,x_j)) + \mathcal{O}(\frac{1}{M}\ln M).$$

Thus, when M is large, the (scaled) log-likelihood converges to a quantity depending only on the normalized marginals $\hat{\mathbf{n}}_i, \hat{\mathbf{n}}_{ij}$. For the sake of simplicity, from now on, we use the $\mathbf{n}_i, \mathbf{n}_{ij}$ instead of $\hat{\mathbf{n}}_i, \hat{\mathbf{n}}_{ij}$ to denote the normalized marginal distributions. The aggregate observations $\mathbf{y}_i,\ \forall i \in \Gamma$ are also normalized so that $\sum_{x_i} y_i(x_i) = 1$. We further denote their joint distributions by $\mathbf{N}$. Thanks to the tree structure of the underlying graph, minimizing the above is equivalent to the following problem.

Problem 1.

$$\min_{\mathbf{N}} \quad \mathrm{KL}(\mathbf{N}\ ||\ \prod_{(i,j)\in E}\psi_{ij}(x_i,x_j)) \tag{16a}$$

$$\textit{subject to} \quad P_j(\mathbf{N}) = \mathbf{y}_j, \quad \forall j \in \Gamma, \tag{16b}$$

where P_j denotes projection operation, i.e., $\mathbf{n}_j = P_j(\mathbf{N})$.

Problem 1 is similar to the variational inference formulation in standard PGM [44] except for the existence of the extra constraints (16b). Such a constrained version of variational inference has indeed been studied in [53], however, from a very different perspective. Problem 1 provides a new viewpoint for our CGM inference problem: finding a joint aggregate distribution $\mathbf{N}$ that is closest to the prior model of the PGM while satisfying the marginal constraints (16b). A special case of Problem 1 that has been widely studied is the Schrödinger bridge problem [33], [34], which corresponds to the choice $J = 2$ and $\Gamma = \{1, 2\}$. Thus, Problem 1 can also be viewed as a multi-marginal generalization of the Schrödinger bridge problems.

B. Multimarginal Optimal Transport Approach

When treating the pairwise potentials of the graphical model as the local components of the cost function in MOT, i.e.,

$$\mathbf{C}(\mathbf{x}) = -\sum_{i,j}\ln\psi_{ij}(x_i,x_j), \tag{17}$$

Problem (16) can be viewed as a regularized MOT problem. Indeed, the regularized MOT problem (13) can be rewritten as

$$\min_{\mathbf{B}\in\mathbb{R}_+^{d\times\cdots\times d}} \quad \mathrm{KL}\left(\mathbf{B}\ ||\ \exp\left(-\frac{\mathbf{C}}{\epsilon}\right)\right) \qquad \text{subject to} \quad P_j(\mathbf{B}) = \mu_j, \quad \forall j \in \Gamma. \tag{18}$$

Plugging (17) into the above and taking $\epsilon = 1$ yields exactly the same expression as in (16a).

Consequently, we can adopt the Sinkhorn algorithm (Algorithm 3) to solve the MOT Problem 1, and this is guaranteed to converge [25]. Thanks to the graphical structure (17) of the cost $\mathbf{C}$, we can further accelerate the algorithm by utilizing standard BP to realize the key projection step $P_j(\mathbf{K}\odot\mathbf{U})$ in the Sinkhorn algorithm; we call this combination of Sinkhorn and BP the Sinkhorn belief propagation (SBP) algorithm.

Before presenting our algorithm, we first characterize the stationary points of the optimization (16) in terms of local messages; these will be used to compute the projections in a Sinkhorn scaling step. When the underlying graph is a tree, the objective function of (16) is the same as the Bethe free energy [44]

$$F_{\mathrm{Bethe}}(\mathbf{n}) = \sum_{(i,j)\in E}\sum_{x_i,x_j} n_{ij}(x_i,x_j)\ln\frac{n_{ij}(x_i,x_j)}{\psi_{ij}(x_i,x_j)} - \sum_i (d_i-1)\sum_{x_i} n_i(x_i)\ln n_i(x_i). \tag{19}$$

Departing from the CGM free energy given by (6) which contains an explicit term for the observation noise model, the Bethe free energy equation above does not have such a term since the observations are encoded as nodes in the model. Thus, Problem 1 takes the form

$$\min_{\mathbf{n}_{ij},\mathbf{n}_i} \quad F_{\mathrm{Bethe}}(\mathbf{n}) \tag{20a}$$

$$\text{subject to} \quad n_i(x_i) = y_i(x_i),\ \forall i \in \Gamma \tag{20b}$$

$$\sum_{x_j} n_{ij}(x_i,x_j) = n_i(x_i), \forall (i,j) \in E \tag{20c}$$

$$\sum_{x_i} n_i(x_i) = 1,\ \forall i \in V. \tag{20d}$$

Here (20b) corresponds to aggregate observation constraints and (20c)-(20d) represent consistency constraints. One can apply Lagrangian duality theory [54] to the constrained convex optimization (20) to obtain Theorem 1. The proof is deferred to the Appendix.

Theorem 1. *The solution to the aggregate inference problem (20) is characterized by*

$$n_i(x_i) \propto \prod_{k\in N(i)} m_{k\to i}(x_i),\ \forall i \notin \Gamma \tag{21}$$

where $m_{i\to j}(x_j)$ are fixed points of

$$m_{i\to j}(x_j) \propto \sum_{x_i}\psi_{ij}(x_i,x_j)\prod_{k\in N(i)\setminus j} m_{k\to i}(x_i); \quad \forall i \notin \Gamma,\ \forall j \in N(i), \tag{22a}$$

$$m_{i\to j}(x_j) \propto \sum_{x_i}\psi_{ij}(x_i,x_j)\frac{y_i(x_i)}{m_{j\to i}(x_i)}; \quad \forall i \in \Gamma,\ \forall j \in N(i). \tag{22b}$$

The expression $m_{i\to j}$ in (22) can be viewed as messages between nodes, as in BP. Among the two classes of messages in (22), (22a) resembles that in standard BP while (22b) corresponds to the scaling step (15).

Taking all these components into account, we arrive at the Sinkhorn belief propagation (SBP) algorithm (Algorithm 4). The SBP algorithm has convergence guarantees due to convergence of the Sinkhorn algorithm. Once converged, the marginals can be recovered by (21). The initialization runs a standard BP algorithm over the same PGM without accounting for the constraints (16b). Let $\bar{\Gamma}$ be a sequence containing elements in Γ such that all neighboring elements are different and each element in Γ appear in $\bar{\Gamma}$ infinitely often. For instance, $\bar{\Gamma}$ can be set to be $\{1, 2, 3, 1, 2, 3, \dots, \}$ for the example in Figure 4. For an element $i \in \bar{\Gamma}$, denote the element following i by i_{next}.

Remark 2. *The SBP algorithm inherits properties of the Sinkhorn algorithm and is a special case of a dual block ascent algorithm. It does not preserve primal feasibility in each update, but the algorithm converges to satisfies that $\sum_{j\in\Gamma}\|P_j(\mathbf{N})-\mathbf{y}_j\|_1 < \delta$ for an arbitrary chosen precision δ.*

In the language of the Sinkhorn algorithm (Algorithm 3), $m_{i\to j}$ is associated with u_i, thus step i) corresponds to modifying the PGM potential from $\mathbf{K}$ to $\mathbf{K}\odot\mathbf{U}$ with the most recent $\mathbf{U}$. The update ii)

Algorithm 4 Sinkhorn Belief Propagation (SBP)

Initialize all the messages $m_{i\to j}(x_j)$
while not converged **do**
  for $i \in \bar{\Gamma}$ **do**
    i) Update $m_{i\to j}(x_j)$ using (22b)
    ii) Update all the messages on the path from i to i_{next} according to (22a)
  end for
end while

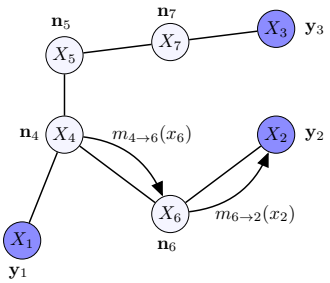


Fig. 4: Example graph demonstrating sequence of message updates.

then calculates the marginal distribution at node i_{next} of the PGM with this modified potential. Due to the tree structure of the PGM, it suffices to update only messages from node i to i_{next} as in step ii) of Algorithm 4. This can be easily explained via an example as depicted in Figure 4 with $i = 1, i_{\text{next}} = 2$. After updating the message $m_{1\to 4}(x_4)$, we only need to update $m_{4\to 6}(x_6)$ and $m_{6\to 2}(x_2)$ since these are the only two messages that contribute to the next scaling update of $m_{2\to 6}(x_6)$ as explained by (22b).

Remark 3. *If the cost tensor* $\mathbf{C}$ *is finite, then the Sinkhorn algorithm (Algorithm 3) has a linear convergence rate [49], [55]. Therefore, if all edge potentials* ψ_{ij} *are strictly positive, the SBP algorithm on a tree (Algorithm 4) also converges linearly.*

Each update in Algorithm 3 requires a projection step P_j, which is realized by belief propagation between neighboring observation nodes i and i_{next}. For a graph with J nodes, where each node takes d possible values, the complexity of operation (22a) is $O(d^2)$. The update ii) in SBP takes at most J numbers of operation (22a), thus, the worst case complexity of SBP for each update is $O(Jd^2)$.

IV. Filtering over collective hidden Markov models

In this section, we study filtering problems in a hidden Markov model with aggregate observations. Towards this, we present the collective forward-backward algorithm (Algorithm 5) which is a special case of SBP in HMMs. We also discuss the connections between collective filtering and filtering in standard (individual) HMMs.

An HMM consists of a hidden (unobservable) Markov chain that evolves over time and corresponding noisy variables that are observed. For the sake of simplicity in notation, denote the unobserved variables as $X_1, X_2, \ldots$ and observed variables as $O_1, O_2, \ldots$. Therefore, the underlying graph consists of the variables $V = \{X_1, X_2, \ldots, O_1, O_2, \ldots\}$ with $\Gamma = \{O_1, O_2, \ldots\}$. An HMM is parameterized by the initial distribution $\pi(X_1)$, the state transition probabilities $p(X_{t+1} \mid X_t)$, and the observation probabilities $p(O_t \mid X_t)$ for each time step $t = 1, 2, \ldots$. An HMM of length T is represented graphically as shown in Figure 5

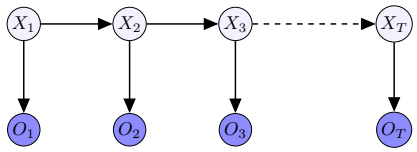


Fig. 5: Graphical representation of a length T HMM.

The joint distribution of an HMM with length T can be factorized as

$$p(\mathbf{x}, \mathbf{o}) = \pi(x_1) \prod_{t=1}^{T-1} p(x_{t+1} \mid x_t) \prod_{t=1}^{T} p(o_t \mid x_t), \tag{23}$$

where $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ and $\mathbf{o} = \{o_1, o_2, \ldots, o_T\}$ represent a particular assignment of hidden and observation variables respectively. Although the graphical model of HMM depicted in Figure 5 is directed, it can be equivalently represented by an undirected model (Markov random field) via the moralization technique as discussed in Section II-A. It turns out that the corresponding undirected model for an HMM is found by just replacing all the directed edges with undirected ones as in Figure 6. More specifically, the factorization of the joint distribution described by (23) in terms of transition and emission probabilities is equivalent to the factorization in (1) with edge potentials being the corresponding conditional probabilities, e.g.,

$$\begin{aligned}
\psi_{1,1}(x_1, o_1) &= p(o_1 \mid x_1) p(x_1) \\
\psi_{t,t}(x_t, o_t) &= p(o_t \mid x_t) \qquad \text{for } t = 2, \ldots, T, \\
\psi_{t,t+1}(x_t, x_{t+1}) &= p(x_{t+1} \mid x_t) \quad \text{for } t = 1, \ldots, T-1.
\end{aligned}$$

A. Collective forward-backward algorithm

To generate aggregate data in HMM settings, we follow the generative model discussed in Section II-B using trajectories of a large number (M) of individuals. Let us denote the messages in collective HMM as shown in Figure 6, where $\alpha_t(x_t)$ are the messages in the forward direction and $\beta_t(x_t)$ are the messages in the backward direction. Moreover, $\gamma_t(x_t)$ denote the messages from observation node to hidden node and $\xi_t(o_t)$ are the messages from hidden node to observation node. Note that the forward messages are $\alpha_t(x_t) = m_{t-1\to t}(x_t)$ and the backward messages are $\beta_t(x_t) = m_{t+1\to t}(x_t)$. Moreover, the upward and downward messages correspond to $\gamma_t(x_t) = m_{t\to t}(x_t)$ and $\xi_t(o_t) = m_{t\to t}(o_t)$, respectively. By abuse of notation, here both the upward and downward messages are denoted as $m_{t\to t}$, and distinguished only by the argument. Based on Theorem 1, these messages are characterized via the following.

Corollary 1. *The solution to aggregate filtering problem (Problem 1) in a collective HMM is*

$$n_t(x_t) \propto \alpha_t(x_t)\beta_t(x_t)\gamma_t(x_t), \quad \forall t = 1, 2 \ldots, T \tag{24}$$

where $\alpha_t(x_t), \beta_t(x_t)$, *and* $\gamma_t(x_t)$ *are the fixed points of the following updates*

$$\alpha_t(x_t) \propto \sum_{x_{t-1}} p(x_t|x_{t-1})\alpha_{t-1}(x_{t-1})\gamma_{t-1}(x_{t-1}), \tag{25a}$$

$$\beta_t(x_t) \propto \sum_{x_{t+1}} p(x_{t+1}|x_t)\beta_{t+1}(x_{t+1})\gamma_{t+1}(x_{t+1}), \tag{25b}$$

$$\gamma_t(x_t) \propto \sum_{o_t} p(o_t|x_t)\frac{y_t(o_t)}{\xi_t(o_t)}, \tag{25c}$$

$$\xi_t(o_t) \propto \sum_{x_t} p(o_t|x_t)\alpha_t(x_t)\beta_t(x_t), \tag{25d}$$

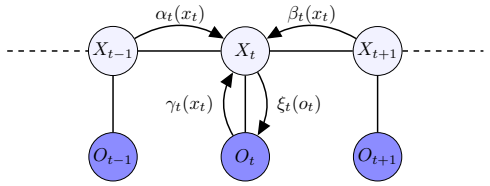


Fig. 6: Messages for inference in a collective HMM.

with boundary conditions

$$\alpha_1(x_1) = \pi(x_1) \quad \textit{and} \quad \beta_T(x_T) = 1. \tag{26}$$

Algorithm 5 Collective forward-backward algorithm

Initialize all the messages $\alpha_t(x_t), \beta_t(x_t), \gamma_t(x_t), \xi_t(o_t)$
while not converged **do**
  Forward pass:
  for $t = 2, 3, \ldots, T$ **do**
    i) Update $\gamma_{t-1}(x_{t-1})$ using (25c)
    ii) Update $\alpha_t(x_t), \xi_t(o_t)$ using (25a) and (25d)
  end for
  Backward pass:
  for $t = T-1, \ldots, 1$ **do**
    i) Update $\gamma_{t+1}(x_{t+1})$ using (25c)
    ii) Update $\beta_t(x_t), \xi_t(o_t)$ using (25b) and (25d)
  end for
end while

Combining Corollary 1 and Algorithm 4 we arrive at the collective forward-backward algorithm (Algorithm 5). Since the underlying graph in HMM is a tree, Algorithm 5 is guaranteed to converge and the estimated marginals are exact. Upon the convergence of the algorithm, the marginals can be estimated as

$$n_t(x_t) \propto \alpha_t(x_t)\beta_t(x_t)\gamma_t(x_t), \quad \forall t = 1, 2 \ldots, T.$$

Note that Algorithm 5 uses the scheduling sequence $\bar{\Gamma} = \{o_1, o_2, \ldots, o_T, o_{T-1}, \ldots, o_1, o_2, \ldots\}$. Other choice of $\bar{\Gamma}$ would also work. The sequence message updates involving all the messages in the forward pass and backward pass is termed as a single iteration. Next, we discuss the scenario where the measurements $y_1, y_2, \ldots, y_T$ are Diracs and show that, in such a case, Algorithm 5 reduces to the standard forward-backward algorithm [37] which is widely used in filtering problems for HMMs.

B. Connections to standard HMM filtering

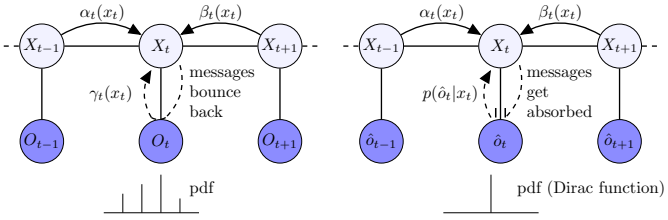
In standard HMMs, a fixed observation sample is recorded at each time step t forming a T length sequence constituting an individual's observations. This filtering/smoothing task in a standard HMM is achieved using the standard BP algorithm discussed in Section II-A. In this special case of HMM graph as depicted in Figure 5, the standard BP takes a simple form, known as forward-backward algorithm [37].

The required marginals, given the observation sequence $\hat{o}_{1:T} = \{\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_T\}$, are

$$\begin{aligned} p(x_t|\hat{o}_{1:T}) &\propto p(\hat{o}_{1:T}|x_t)p(x_t) \\ &= p(\hat{o}_{1:t-1}|x_t)p(\hat{o}_t|x_t)p(\hat{o}_{t+1:T}|x_t)p(x_t) \\ &= p(\hat{o}_t|x_t)p(\hat{o}_{1:t-1}, x_t)p(\hat{o}_{t+1:T}|x_t) \\ &= p(\hat{o}_t|x_t)\alpha_t(x_t)\beta_t(x_t), \end{aligned} \tag{27}$$

Algorithm 6 Forward-backward algorithm

Initialize the messages $\alpha_t(x_t), \beta_t(x_t)$ with $\alpha_1(x_1) = \pi(x_1)$ and $\beta_T(x_T) = 1$
Forward pass:
for $t = 2, 3, \ldots, T$ **do**
  Update $\alpha_t(x_t)$ using (28)
end for
Backward pass:
for $t = T-1, \ldots, 1$ **do**
  Update $\beta_t(x_t)$ using (29)
end for



(a) Aggregate observations (b) Delta observations (standard)

Fig. 7: Relationship between standard and collective HMMs.

where $\alpha_t(x_t) = p(x_t, \hat{o}_{1:t-1})$ denote the forward messages and $\beta_t(x_t) = p(\hat{o}_{t+1:T}|x_t)$ represent the backward messages. These messages in the forward-backward algorithm take the form

$$\begin{aligned} \alpha_t(x_t) &= p(x_t, \hat{o}_{1:t-1}) \\ &= \sum_{x_{t-1}} p(x_t, x_{t-1}, \hat{o}_{1:t-1}) \\ &= \sum_{x_{t-1}} p(\hat{o}_{t-1}|x_{t-1})p(x_t|x_{t-1}, \hat{o}_{1:t-2})p(x_{t-1}, \hat{o}_{1:t-2}) \\ &= \sum_{x_{t-1}} p(x_t|x_{t-1})\alpha_{t-1}(x_{t-1})p(\hat{o}_{t-1}|x_{t-1}), \end{aligned} \tag{28}$$

$$\begin{aligned} \beta_t(x_t) &= p(\hat{o}_{t+1:T}|x_t) \\ &= \sum_{x_{t+1}} p(\hat{o}_{t+1}, \hat{o}_{t+2:T}, x_{t+1}|x_t) \\ &= \sum_{x_{t+1}} p(x_{t+1}|x_t)p(\hat{o}_{t+2:T}|x_{t+1})p(\hat{o}_{t+1}|x_{t+1}) \\ &= \sum_{x_{t+1}} p(x_{t+1}|x_t)\beta_{t+1}(x_{t+1})p(\hat{o}_{t+1}|x_{t+1}). \end{aligned} \tag{29}$$

All the steps of the standard forward-backward algorithm are listed in Algorithm 6. Now we establish the relationship between filtering in standard and collective HMMs.

Theorem 2. *In case of Dirac observations, the collective forward-backward algorithm reduces to the standard forward-backward algorithm.*

Proof. In case of Dirac observations, a fixed sequence of observations $\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_T$ is made and the (aggregate) observations take the form

$$y_t(o_t) = \delta(o_t - \hat{o}_t), \tag{30}$$

where $\delta(\cdot)$ denotes Dirac function. With these fixed delta observations, the messages in Equation (25c) become

$$\gamma_t(x_t) \propto \sum_{o_t} p(o_t|x_t)\frac{\delta(o_t - \hat{o}_t)}{\xi_t(o_t)} = \frac{p(\hat{o}_t|x_t)}{\xi_t(\hat{o}_t)}.$$

Note that the denominator in the last term of the above equation can be omitted since it only serves as a scaling coefficient and therefore,

$$\gamma_t(x_t) = p(\hat{o}_t|x_t). \tag{31}$$

Now the forward and backward messages take the form

$$\alpha_t(x_t) \propto \sum_{x_{t-1}} p(x_t|x_{t-1})\alpha_{t-1}(x_{t-1})p(\hat{o}_{t-1}|x_{t-1}), \tag{32a}$$

$$\beta_t(x_t) \propto \sum_{x_{t+1}} p(x_{t+1}|x_t)\beta_{t+1}(x_{t+1})p(\hat{o}_{t+1}|x_{t+1}). \tag{32b}$$

Using above, the required marginals are estimated as

$$n_t(x_t) \propto p(\hat{o}_t|x_t)\alpha_t(x_t)\beta_t(x_t). \tag{33}$$

The messages given by Equation (32) are nothing but the messages used in the forward-backward algorithm for standard HMMs as in Algorithm 6. □

The relationship between the filtering problems in standard and collective HMMs can be intuitively explained as depicted in Figure 7. In case of general aggregate observations, the downward messages $\xi_t(o_t)$ bounce back (Figure 7a) and contribute to the corresponding upward messages $\gamma_t(x_t)$ as in Equation (25c). In case of Dirac measurements, the aggregate observations result in Dirac distributions and as a consequence, the downward messages get absorbed (Figure 7b) and do not contribute to upward messages as explained by Equation (31).

Remark 4. *Indeed, the relationship between standard HMM and collective HMM can be extended to more general graphical models. It turns out that for any arbitrary tree graphical model, in the case of fixed Dirac aggregate observations, the collective inference algorithm SBP coincides with the standard BP.*

V. Evaluation

We conduct four sets of experiments to evaluate the performance of our algorithms. The first one aims to evaluate the efficiency and convergence rate of SBP, compared with NLBP and Bethe-RDA. Note that since SBP uses a different observation model to NLBP and Bethe-RDA, this experiment is carried out only to compare the convergent behaviors of the algorithms. In the second experiment, we present an application of SBP in estimating ensembles with sparse information. The third experiment is concerned with the sensor fusion problem where the underlying graph is a star graph. Finally, we empirically show that SBP algorithms can have good performance even in a PGM with loops. All the experiment were run on Intel i7-9700 CPU.

A. Bird Migration

First, we study a synthetic bird migration problem with underlying generative model following an HMM. Similar to the environment in [16], we simulate M birds flying over a $L \times L$ grid, aiming from bottom-left to top-right. The position transition probability between previous time and current time step follows a log-linear distribution that accounts for four factors: the distance between two positions, the angle between the movements direction and the wind, the angle between the direction of movement and the direction to the goal, and the preference to stay in the original cell. Each bird is simulated independently, following a T-length Markov chain. The parameter for the log-linear model is denoted by $\mathbf{w}$. In the NLBP setting, the sensors count the number of birds flying through each cell. Independent Poisson noise is added to each sensor measurement, which follows $y \propto \text{Poisson}(\beta n)$. In the SBP settings, we incorporate the noise in the observations via a discrete Gaussian kernel [56] with bandwidth of 4 such that the probability for an individual bird to be observed by a sensor follows a discrete Gaussian distribution centered at the sensor. In all experiments, we employ model parameters $\mathbf{w} = (3, 5, 5, 10)$; the same parameters are used in all the inference algorithms. We set $M = 5000$, and $\beta = 1$. We compare the convergence performance between NLBP, Bethe-RDA and SBP with different L and T values, as depicted in Figure 8. In fact, we also implemented a simplified version, PROX, of Bethe-RDA which is based on standard proximal gradient algorithm with Kullback-Leibler divergence. When T is fixed and L varies, SBP is faster than NLBP, Bethe-RDA and PROX. When L is fixed and T varies, SBP is also faster than NLBP, Bethe-RDA and PROX. Moreover, we find that the run time does not grow much as T increases. This makes SBP suitable for large HMMs. The convergence behaviors of NLBP, Bethe-RDA, PROX, and SBP are displayed in Figures 8c. Note that we show the total number of iterations instead of total running time for the comparison as all these algorithms have similar per iteration complexity. The 1-norm distance between true marginal distributions and the estimated distributions decrease monotonically for SBP, Bethe-RDA and PROX, whereas some instability occurs for NLBP. To stabilize NLBP, a damping ratio α is needed. However, higher damping rate implies smaller step size which in turn slows down the algorithm. We also find that the convergence property of NLBP is sensitive to the prior distribution and damping ratio when T is large.

B. Ensemble Estimation with Sparse Information

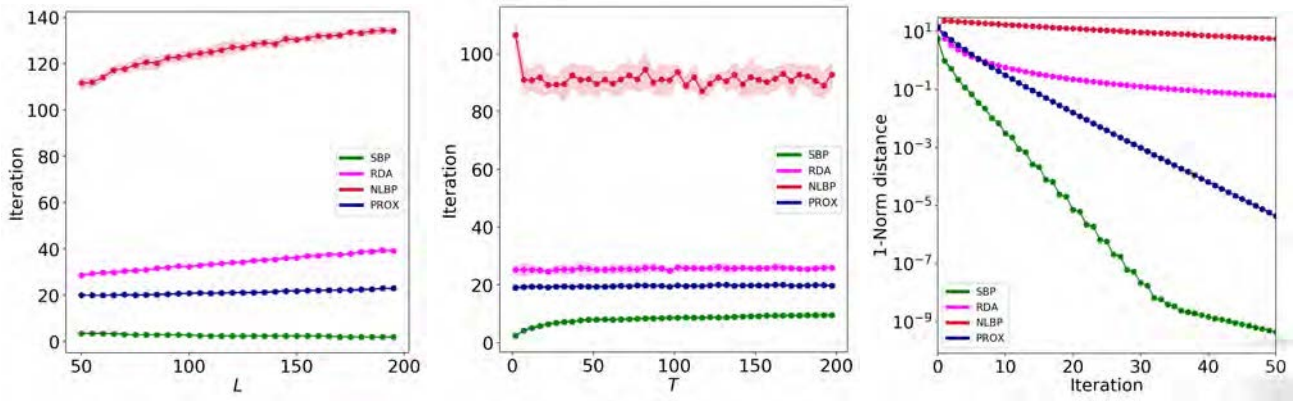
Next, we conduct a more challenging experiment wherein the aggregate observations are sparse. The task is to track ensembles over a network with limited number of sensors. The sensors can not tell the exact locations of the agents, such as in the case of Wi-Fi hotspots, or cell phone based stations, which can only tell the number of connected devices. Ensemble estimation is needed in tracking the human group activity without loss of individual privacy. We evaluate the model over a 20×20 grid network with 16 sensors placed randomly as in Figure 9. The problem is modeled as a collective HMM, where the hidden state space has cardinality $|\mathcal{X}_u| = 400$, and the observation state space has cardinality $|\mathcal{X}_o| = 16$.

At each timestamp, each sensor observes a count, which records the number of agents connected to the sensor. Each agent can only connect with one sensor at a time and the probability of the connection decreases exponentially as the distance between agent and the sensor increases. To demonstrate the performance of SBP in estimating multi-modal distributions, we simulate a population with two clusters: one from left-bottom and one from center-bottom; both aim to the right-top corner of the grid in a $T = 15$ time interval. We model the transition probability as in the bird migration setup discussed in Section V-A.

We run simulations with 100 agents (Figure 10a) and 10000 agents (Figure 10b). As can be seen from the figures, even with such a sparse observation model, SBP can still infer the population movements to a satisfying accuracy.

C. SBP on sensor fusion problems

In this experiment, we consider the sensor fusion problem [46] where the goal is to combine the measurements from multiple sensors into one by some sort of averaging. In the aggregate setting, each sensor is associated with a distribution and the problem can be formulated



(a) (b) (c)

Fig. 8: Comparison of performance between NLBP, Bethe-RDA, PROX and SBP: (a) illustrates the performance for different grid sizes L with a constant $T = 20$, (b) compares the oracle complexity for different values of T with fixed $L = 50$, (c) shows the convergent behavior in terms of 1-norm distance with respect to the optimal solutions. Here each iteration comprises of updating all the messages involved in the underlying HMM. For both (a) and (b), each algorithm is run over 10 different trials. The solid curves represent the mean time and the shaded regions represent the corresponding $\pm 1\times$ standard deviation. For (c), the parameters are fixed as $L = 50$ and $T = 50$.

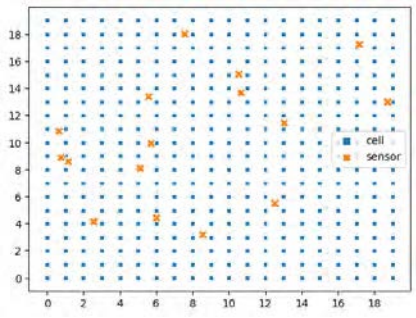


Fig. 9: Sensor Location

as a aggregate inference problem with the graph structure shown in Figure 11. The edge potentials are induced by the quadratic Euclidean distance which penalizes the differences between the average and the sensor measurements. In the simulation, we generate 50-dimensional discrete distributions from a symmetric Dirichlet distribution with concentration parameter 1.0 for the leaf nodes. We compare the convergence behavior in terms of the 1-norm distance with respect to the true average in Figure 12. Note that in cases of NLBP, Bethe-RDA, or PROX algorithms, Poisson noise is added to the measurements. It can be observed from Figure 12 that our SBP algorithm converges faster than the other algorithms for this example as well.

D. Empirical Loopy Graph Validation

Finally, we run a simple experiment where the underlying graph has loops as shown in Figure 13. The purpose of this experiment is to show that our proposed SBP algorithm may also be applicable to graphs with loops similar to the standard BP algorithm in loopy graphs [57]. We set $|\mathcal{X}_u| = |\mathcal{X}_o| = 5$ and generate the aggregate node distributions randomly. Moreover, the edge potential were generated using $\exp(I + Q)$, where I is the identity matrix and Q represents a random matrix generated by a Gaussian distribution. We estimate the marginal distribution by solving (16) using the SBP algorithm and then compare them with exact marginals by solving (16) using generic convex optimization algorithms. The convergence of the estimates for five different random seeds in terms or 1-norm distance is shown in Figure 14. We observe that the SBP algorithm has a good performance on the loopy graph in this example.

VI. Conclusion

In this paper, we presented a reliable algorithm for inference/filtering from aggregate data based on multi-marginal optimal transport theory. We established that the aggregate inference/filtering problem is a special case of the entropic regularized MOT problem when the cost of MOT is structured according to the graphical model. We then combined the Sinkhorn algorithm for the MOT problems and the standard belief propagation algorithm to establish our method. Our algorithm enjoys fast convergence and has a convergence guarantee when the underlying graph structure is a tree. For the cases of HMMs which are widely used in control and estimation, we specialize our SBP algorithm to establish the collective forward-backward algorithm. The latter naturally generalizes the forward-backward algorithm in standard filtering problems for HMMs. We evaluated the performance of our algorithm on multiple applications involving inference from aggregate data such as bird migration and human mobility based on hidden Markov models. In the future, we plan to extend our current algorithm to cover graphical models with continuous state space. We also plan to investigate the parameter learning of CGMs using the MOT framework.

References

[1] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

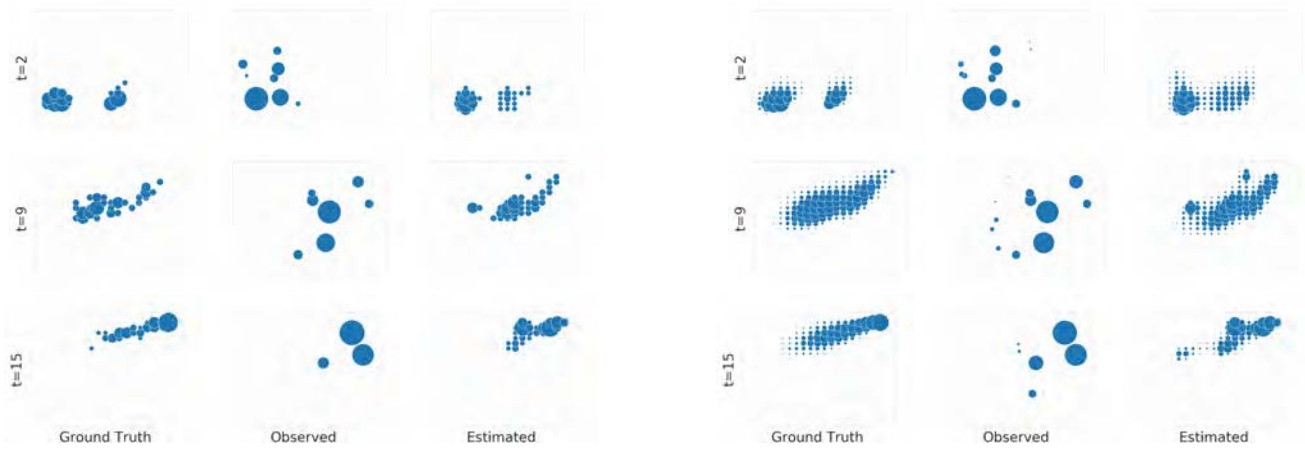
[2] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[3] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.

[4] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[5] G. E. Box and G. C. Tiao, *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011, vol. 40.

[6] J. Pearl, "Probabilistic reasoning in intelligent systems: Networks of plausible inference," *Morgan Kaufmann Publishers Inc*, 1988.



(a) $M = 100$ (b) $M = 10000$

Fig. 10: Simulation of the movement of (a) 100 agents and (b) 10000 agents over 20×20 grid for $T = 15$. In each of the figures, the first column depicts the real movement of agents at different time steps, the second column represents the aggregate sensor observations, and the third column depicts estimated aggregated positions. Here, the size of the circles is proportional to the number of agents.

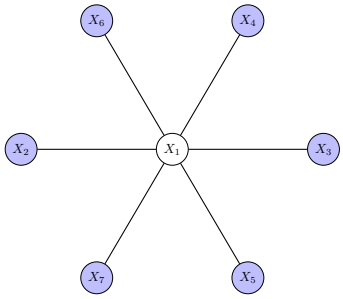


Fig. 11: Graph structure for the sensor fusion problem with six fixed observation distributions.

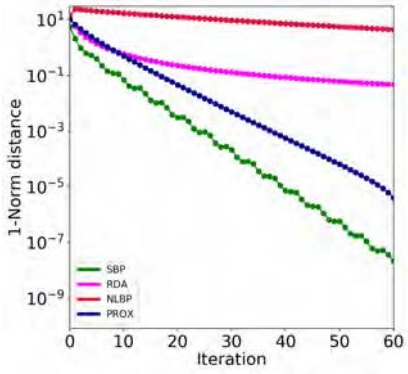


Fig. 12: Convergence behavior on the sensor fusion problem with 50 leaf nodes.

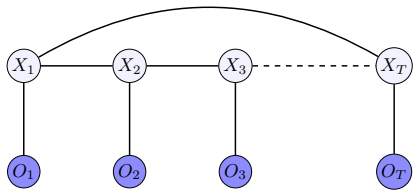


Fig. 13: A loopy graph.

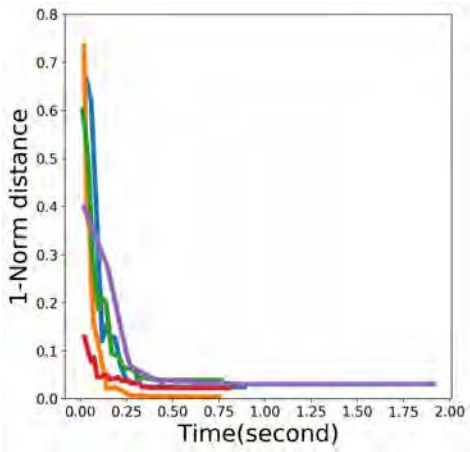


Fig. 14: Convergence of SBP on the loopy graph shown in Figure 13. Different colors represent different realizations.

[7] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Advances in neural information processing systems*, 2001, pp. 689–695.

[8] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, vol. 8, pp. 236–239, 2003.

[9] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[10] R. Sundberg, "Some results about decomposable (or Markov-type) models for multidimensional contingency tables: distribution of marginals and partitioning of tests," *Scandinavian Journal of Statistics*, pp. 71–79, 1975.

[11] E. C. MacRae, "Estimation of time-varying Markov processes with aggregate data," *Econometrica: journal of the Econometric Society*, pp. 183–198, 1977.

[12] J. D. Kalbfleisch, J. F. Lawless, and W. M. Vollmer, "Estimation in Markov models from aggregate data," *Biometrics*, pp. 907–919, 1983.

[13] D. R. Sheldon and T. G. Dietterich, "Collective graphical models," in *Advances in Neural Information Processing Systems*, 2011, pp. 1161–1169.

[14] D. Luo, H. Xu, Y. Zhen, B. Dilkina, H. Zha, X. Yang, and W. Zhang, "Learning mixtures of Markov chains from aggregate data with structural constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1518–1531, 2016.

[15] D. Sheldon, T. Sun, A. Kumar, and T. Dietterich, "Approximate inference in collective graphical models," in *International Conference on Machine Learning*, 2013, pp. 1004–1012.

[16] T. Sun, D. Sheldon, and A. Kumar, "Message passing for collective graphical models," in *International Conference on Machine Learning*, 2015, pp. 853–861.

[17] L. Vilnis, D. Belanger, D. Sheldon, and A. McCallum, "Bethe projections for non-local inference," in *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, 2015, pp. 892–901.

[18] W. Gangbo and A. Święch, "Optimal maps for the multidimensional Monge-Kantorovich problem," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 51, no. 1, pp. 23–45, 1998.

[19] B. Pass, "Multi-marginal optimal transport: theory and applications," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 49, no. 6, pp. 1771–1790, 2015.

[20] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, "Iterative Bregman projections for regularized transportation problems," *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. A1111–A1138, 2015.

[21] L. Nenna, "Numerical methods for multi-marginal optimal transportation," Ph.D. dissertation, 2016.

[22] B. Pass, "On the local structure of optimal measures in the multi-marginal optimal transportation problem," *Calculus of Variations and Partial Differential Equations*, vol. 43, no. 3-4, pp. 529–536, 2012.

[23] C. Villani, *Topics in optimal transportation*. American Mathematical Soc., 2003, no. 58.

[24] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The annals of mathematical statistics*, vol. 35, no. 2, pp. 876–879, 1964.

[25] J. Franklin and J. Lorenz, "On the scaling of multidimensional matrices," *Linear Algebra and its applications*, vol. 114, pp. 717–735, 1989.

[26] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in neural information processing systems*, 2013, pp. 2292–2300.

[27] A. Pasanisi, S. Fu, and N. Bousquet, "Estimating discrete Markov models from various incomplete data schemes," *Computational Statistics & Data Analysis*, vol. 56, no. 9, pp. 2609–2625, 2012.

[28] G. Bernstein and D. Sheldon, "Consistently estimating Markov chains with noisy aggregate data," in *Artificial Intelligence and Statistics*, 2016, pp. 1142–1150.

[29] Y. Chen and J. Karlsson, "State tracking of linear ensembles via optimal mass transport," *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 260–265, 2018.

[30] I. Haasler, A. Ringh, Y. Chen, and J. Karlsson, "Estimating ensemble flows on a hidden Markov chain," in *58th IEEE Conference on Decision and Control*, 2019.

[31] Y. Chen, T. T. Georgiou, and M. Pavon, "Optimal steering of a linear stochastic system to a final probability distribution, part I," *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1158–1169, 2015.

[32] A. Taghvaei and P. G. Mehta, "An optimal transport formulation of the ensemble Kalman filter," *IEEE Transactions on Automatic Control*, 2020.

[33] C. Léonard, "A survey of the Schrödinger problem and some of its connections with optimal transport," *DYNAMICAL SYSTEMS*, vol. 34, no. 4, pp. 1533–1574, 2014.

[34] Y. Chen, T. T. Georgiou, and M. Pavon, "On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint," *Journal of Optimization Theory and Applications*, vol. 169, no. 2, pp. 671–691, 2016.

[35] Y. Chen, G. Conforti, T. T. Georgiou, and L. Ripani, "Multi-marginal Schrödinger bridges," in *International Conference on Geometric Science of Information*. Springer, 2019, pp. 725–732.

[36] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[37] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[38] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE transactions on automatic control*, vol. 45, no. 5, pp. 910–927, 2000.

[39] T. Yang, P. G. Mehta, and S. P. Meyn, "Feedback particle filter," *IEEE transactions on Automatic control*, vol. 58, no. 10, pp. 2465–2480, 2013.

[40] R. J. Lorentzen and G. Nævdal, "An iterative ensemble Kalman filter," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1990–1995, 2011.

[41] R. Singh, I. Haasler, Q. Zhang, J. Karlsson, and Y. Chen, "Incremental inference of collective graphical models," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 421–426, 2020.

[42] P. Dvurechensky, A. Gasnikov, and A. Kroshnin, "Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm," in *International conference on machine learning*. PMLR, 2018, pp. 1367–1376.

[43] T. Lin, N. Ho, and M. I. Jordan, "On the efficiency of Sinkhorn and greenkhorn and their acceleration for optimal transport," 2021.

[44] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on information theory*, vol. 51, no. 7, pp. 2282–2312, 2005.

[45] L. Xiao, "Dual averaging method for regularized stochastic learning and online optimization," in *Advances in Neural Information Processing Systems*, 2009, pp. 2116–2124.

[46] F. Elvander, I. Haasler, A. Jakobsson, and J. Karlsson, "Multi-marginal optimal transport using partial information with applications in robust localization and sensor fusion," *Signal Processing*, 2020.

[47] Y. Chen, T. Georgiou, and M. Pavon, "Entropic and displacement interpolation: a computational approach using the Hilbert metric," *SIAM Journal on Applied Mathematics*, vol. 76, no. 6, pp. 2375–2396, 2016.

[48] W. E. Deming and F. F. Stephan, "On a least squares adjustment of a sampled frequency table when the expected marginal totals are known," *The Annals of Mathematical Statistics*, vol. 11, no. 4, pp. 427–444, 1940.

[49] Z.-Q. Luo and P. Tseng, "On the convergence of the coordinate descent method for convex differentiable minimization," *Journal of Optimization Theory and Applications*, vol. 72, no. 1, pp. 7–35, 1992.

[50] I. Haasler, A. Ringh, Y. Chen, and J. Karlsson, "Multimarginal optimal transport with a tree-structured cost and the Schrödinger bridge problem," *SIAM Journal on Control and Optimization*, vol. 59, no. 4, pp. 2428–2453, 2021.

[51] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov model: Analysis and applications," *Machine learning*, vol. 32, no. 1, pp. 41–62, 1998.

[52] S. S. Varadhan, *Large deviations and applications*. Society for Industrial and Applied Mathematics(SIAM), 1984.

[53] Y. W. Teh and M. Welling, "The unified propagation and scaling algorithm," in *Advances in neural information processing systems*, 2002, pp. 953–960.

[54] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[55] Z.-Q. Luo and P. Tseng, "On the convergence rate of dual ascent methods for linearly constrained convex minimization," *Mathematics of Operations Research*, vol. 18, no. 4, pp. 846–867, 1993.

[56] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. USA: Prentice-Hall, Inc., 2006.

[57] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 467–475.

APPENDIX

A. Proof of Theorem 1

We first construct a Lagrangian for Problem (20) with Lagrange multipliers $\nu_{ji}(x_i)$ corresponding to the marginalization constraints (20c), and ζ_i corresponding to the normalization constraints (20d). This yields the Lagrangian

$$\begin{aligned}\mathcal{L} = & F_{\text{Bethe}}(\mathbf{n}) \\ & + \sum_{i\in V}\sum_{j\in N(i)}\sum_{x_i}\nu_{ji}(x_i)\left(\sum_{x_j} n_{ij}(x_i,x_j) - n_i(x_i)\right) \\ & + \sum_{i\in V}\zeta_i\left(\sum_{x_i} n_i(x_i) - 1\right). \end{aligned} \tag{34}$$

Since the marginal constraints (20b) are trivial, we choose not to introduce Lagrangian multipliers for them and instead use these constraints explicitly when optimizing the Lagrangian. Differentiating the Lagrangian with respect to the aggregate marginals $\mathbf{n}_{ij}$ and equating this derivative to zero, we obtain

$$\begin{aligned}&\frac{\partial\mathcal{L}}{\partial n_{ij}(x_i,x_j)} = 1 + \ln\frac{n_{ij}(x_i,x_j)}{\psi_{ij}(x_i,x_j)} + \nu_{ji}(x_i) + \nu_{ij}(x_j) = 0 \\ &\Rightarrow n_{ij}(x_i,x_j) \propto \psi_{ij}(x_i,x_j)\exp\left(-\nu_{ji}(x_i) - \nu_{ij}(x_j)\right). \end{aligned} \tag{35}$$

Now differentiating the Lagrangian with respect to $\mathbf{n}_i$, for $i \notin \Gamma$ and $d_i > 1$, we have

$$\begin{aligned}&\frac{\partial\mathcal{L}}{\partial n_i(x_i)} = \zeta_i - (d_i-1)[1+\ln n_i(x_i)] - \sum_{j\in N(i)}\nu_{ji}(x_i) = 0 \\ &\Rightarrow\ n_i(x_i) \propto \exp\left(-\frac{1}{d_i-1}\left\{\sum_{j\in N(i)}\nu_{ji}(x_i)\right\}\right). \end{aligned} \tag{36}$$

Similarly, when $i \notin \Gamma$ and $d_i = 1$, we have

$$\frac{\partial\mathcal{L}}{\partial n_i(x_i)} = 0 \ \Rightarrow\ \nu_{ji}(x_i) - \zeta_i = 0, \tag{37}$$

where $j \in N(i)$ is the only neighbor of node i.

Denote

$$m_{i\to j}(x_j) := \sum_{x_i}\psi_{ij}(x_i,x_j)\ \exp(-\nu_{ji}(x_i)). \tag{38}$$

Using (35), (36) and marginalization constraint (20c), we obtain, for i such that $d_i > 1$,

$$\begin{aligned}&\sum_{x_j}\psi_{ij}(x_i,x_j)\ \exp\left(-\nu_{ji}(x_i) - \nu_{ij}(x_j)\right) \propto \\ &\qquad \exp\left(-\frac{1}{d_i-1}\left\{\sum_{j\in N(i)}\nu_{ji}(x_i)\right\}\right) \\ &\Rightarrow m_{j\to i}(x_i)\ \exp(-\nu_{ji}(x_i)) \propto \prod_{j\in N(i)}\exp\ (-\nu_{ji}(x_i))^{\frac{1}{d_i-1}} \\ &\Rightarrow \prod_{j\in N(i)} m_{j\to i}(x_i)\ \exp(-\nu_{ji}(x_i)) \propto \\ &\qquad \prod_{j\in N(i)}\left(\exp\ (-\nu_{ji}(x_i))^{\frac{1}{d_i-1}}\prod_{k\in N(i)\setminus j}\exp\ (-\nu_{ki}(x_i))^{\frac{1}{d_i-1}}\right). \end{aligned}$$

It follows that, by leveraging the fact that $|N(i)| = d_i$,

$$\begin{aligned}&\prod_{j\in N(i)} m_{j\to i}(x_i)\ \exp(-\nu_{ji}(x_i)) \propto \\ &\qquad \left(\prod_{j\in N(i)}\exp\ (-\nu_{ji}(x_i))^{\frac{1}{d_i-1}}\right)\left(\prod_{k\in N(i)}\exp\ (-\nu_{ki}(x_i))\right) \\ &\Rightarrow \prod_{j\in N(i)} m_{j\to i}(x_i) \propto \prod_{j\in N(i)}\exp\ (-\nu_{ji}(x_i))^{\frac{1}{d_i-1}}. \end{aligned} \tag{39}$$

Combining (36) and (39), we arrive at

$$n_i(x_i) \propto \prod_{j\in N(i)} m_{j\to i}(x_i), \tag{40}$$

which is (21) for $d_i > 1$ and $i \notin \Gamma$.

For $i \notin \Gamma$ and $d_i = 1$, i.e., unconstrained variable at a leaf node, using (35) and (37) we deduce that

$$n_{ij}(x_i,x_j) \propto \psi_{ij}(x_i,x_j)\ \exp\left(-\nu_{ij}(x_j)\right). \tag{41}$$

In view of (38) and marginalization constraint (20c), we get

$$\begin{aligned} n_i(x_i) &= \sum_{x_j} n_{ij}(x_i,x_j) \\ &\propto \sum_{x_j}\psi_{ij}(x_i,x_j)\ \exp\left(-\nu_{ij}(x_j)\right) \\ &= m_{j\to i}(x_i), \end{aligned} \tag{42}$$

which is (21) for $d_i = 1$ and $i \notin \Gamma$.

Combining (35) and (20c), we obtain

$$\begin{aligned}&\sum_{x_j}\psi_{ij}(x_i,x_j)\ \exp\left(-\nu_{ji}(x_i) - \nu_{ij}(x_j)\right) \propto n_i(x_i) \\ &\Rightarrow m_{j\to i}(x_i)\ \exp(-\nu_{ji}(x_i)) \propto n_i(x_i) \\ &\Rightarrow \exp(-\nu_{ji}(x_i)) \propto \frac{n_i(x_i)}{m_{j\to i}(x_i)} \\ &\Rightarrow \sum_{x_i}\psi_{ij}(x_i,x_j)\exp(-\nu_{ji}(x_i)) \propto \sum_{x_i}\psi_{ij}(x_i,x_j)\frac{n_i(x_i)}{m_{j\to i}(x_i)} \\ &\Rightarrow m_{i\to j}(x_j) \propto \sum_{x_i}\psi_{ij}(x_i,x_j)\frac{n_i(x_i)}{m_{j\to i}(x_i)}. \end{aligned} \tag{43}$$

Therefore, for $i \notin \Gamma$, using (40) and (43), we get

$$\begin{aligned} m_{i\to j}(x_j) &\propto \sum_{x_i}\psi_{ij}(x_i,x_j)\frac{\prod_{k\in N(i)} m_{k\to i}(x_i)}{m_{j\to i}(x_i)} \\ &\propto \sum_{x_i}\psi_{ij}(x_i,x_j)\prod_{k\in N(i)\setminus j} m_{k\to i}(x_i), \end{aligned} \tag{44}$$

which is (22a).

Furthermore, for $i \in \Gamma$, combining (43) and observation (fixed marginal) constraints (20b), we arrive at

$$m_{i\to j}(x_j) \propto \sum_{x_i}\psi_{ij}(x_i,x_j)\frac{y_i(x_i)}{m_{j\to i}(x_i)}, \tag{45}$$

which is (22b). This completes the proof.