Automatica 137 (2022) 110100

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Learning hidden Markov models from aggregate observations ☆

Rahul Singh [a], Qinsheng Zhang [a], Yongxin Chen [b,*]

[a] Machine Learning Center, Georgia Institute of Technology, Atlanta, GA, USA
[b] School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

ARTICLE INFO

Article history:
Received 2 November 2020
Received in revised form 21 June 2021
Accepted 3 November 2021
Available online xxxx

Keywords:
Hidden Markov models
Aggregate observations
Parameter learning
Expectation–maximization algorithm

ABSTRACT

In this paper, we propose an algorithm for estimating the parameters of a time-homogeneous hidden Markov model (HMM) from aggregate observations. This problem arises when only the population level counts of the number of individuals at each time step are available, and one seeks to learn the individual HMM from these observations. Our algorithm is built upon the classical expectation–maximization algorithm and the recently proposed aggregate inference algorithm (Sinkhorn belief propagation). We present the parameter learning algorithm for two different settings of HMMs: one with discrete observations and one with continuous observations, and the algorithm exhibits convergence guarantees in both cases. Moreover, our learning framework naturally reduces to the standard Baum–Welch learning algorithm for HMMs when the population size is 1. The efficacy of our algorithm is demonstrated through several numerical experiments.

1. Introduction

There has been a growing interest in applications where data about individuals are not accessible, instead aggregate population-level observations in the form of counts of the individuals are available (Luo, Xu, Zhen, Dilkina, Zha, Yang, & Zhang, 2016; Sheldon & Dietterich, 2011). For various reasons including measurement fidelity, privacy preservation, cost of data collection, and scalability, data is often collected as aggregates. For example, in human ensemble flow analysis, individual trajectories may not be readily accessible due to privacy concerns, but the number of individuals in a certain geographical area can typically be counted by cell phone carriers. More examples include voter turnout based on demography from census data (King, 2013) and bird migration analysis (Sun, Sheldon, & Kumar, 2015). One fundamental part in modeling such aggregate data is estimating the individual model parameters. Learning the underlying individual model from aggregate observations is a challenging task since the full trajectory of each individual is not accessible.

We are interested in learning hidden Markov models (HMMs) using aggregate data. HMMs are popular graphical models used in various scenarios involving unobservable (hidden) data sequences arising in ecology, social dynamics, and emergence of an epidemic (Cappé, Moulines, & Rydén, 2006; Dong, Pentland, & Heller, 2012; Rabiner & Juang, 1986; Singh, Haasler, Zhang, Karlsson, & Chen, 2020). Due to their ability to address the non-stationarity in observed data sequences, HMMs are capable of modeling a rich class of problems. In aggregate HMM settings, a large set of homogeneous individuals transit from one state to another according to the underlying HMM and at each time-step, corresponding aggregated observations are recorded. For example, in epidemiology, one can model spread of an infectious disease such as COVID-19 over time in a geographical area using the population level aggregate data generated by an HMM. In this work, we consider the problem of estimating the parameters of a time-homogeneous hidden Markov model, i.e., transition and observation probabilities, from noisy aggregate data.

A traditional method for learning HMM is the Baum–Welch algorithm (Baum & Eagon, 1967; Baum, Petrie, Soules, & Weiss, 1970), which is a special case of the expectation–maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977; Neal & Hinton, 1998). For the given observations sampled from a model consisting of latent variables (variables that are not observable) with unknown parameters, the EM algorithm aims to find the maximum likelihood estimates of the model parameters. In its first step (E-step), the EM algorithm estimates a function of the expected values of the latent variables and subsequently in the second step (M-step), it finds the maximum likelihood parameter estimates. For the case of learning HMM parameters, inference algorithm such as belief propagation (BP) algorithm (Pearl, 1988) is utilized in the E-step of the EM algorithm. The Baum–Welch algorithm for estimating an HMM uses the forward–backward inference algorithm, one type of BP algorithms, in the E-step

☆ This work was supported by the NSF under grant 1901599, 1942523 and 2008513. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Adrian Wills under the direction of Editor Torsten Söderström.

* Corresponding author.
E-mail addresses: rasingh@gatech.edu (R. Singh), qzhang419@gatech.edu (Q. Zhang), yongchen@gatech.edu (Y. Chen).

https://doi.org/10.1016/j.automatica.2021.110100
0005-1098/

to complete the data. Unfortunately, traditional HMM learning methods such as Baum–Welch algorithm (Baum et al., 1970) cannot be applied to aggregate setting. Learning the individual model from such population-level observations becomes challenging since great amount of information about individuals is lost due to data aggregation and observation noise.

Recently, the learning and inference problems in aggregate settings have been formalized under the collective graphical model (CGM) framework (Sheldon & Dietterich, 2011). Within the CGM framework, for learning the parameters of the individual model, several aggregate inference methods such as non-linear belief propagation (NLBP) (Sun et al., 2015) and Bethe-RDA (Vilnis, Belanger, Sheldon, & McCallum, 2015) algorithms have been utilized in the E-step of the EM algorithm aiming to maximize the complete data likelihood. Both of the inference algorithms work on an explicit observation model. In addition, since NLBP does not exhibit convergence guarantee, it does not lead to stable learning methods.

The primary contribution of our work is a novel algorithm for estimating the HMM parameters with theoretical guarantees from noisy aggregate observations. We utilize a modified EM algorithm for the learning task, where the E-step of the algorithm is solved using recently proposed aggregate inference method, the Sinkhorn belief propagation (SBP) algorithm (Singh et al., 2020). We show that our algorithm exhibits convergence guarantee. Instead of explicitly considering the noise model, we incorporate observation noise in the underlying graph and as a result, our algorithm reduces to the standard Baum–Welch algorithm when only one individual is considered. We further extend our algorithm to learn the model parameters with *continuous* observation noise model. We evaluate the performance of our algorithm on a variety of scenarios including human ensemble flow on real-world data.

Related Work: Estimating Markov chains from aggregate data, also referred to as *macro* data in earlier works, has a long history. It was first studied in Lee, Judge, and Zellner (1970) where the transition matrices were estimated based on maximum likelihood method. In Kalbfleisch, Lawless, and Vollmer (1983), MacRae (1977) and Sundberg (1975), the modeling of a single Markov chain was studied by maximizing the aggregate posterior. More recent learning methods from aggregate data include Luo et al. (2016) and Pasanisi, Fu, and Bousquet (2012). After the introduction of the CGM framework in Sheldon and Dietterich (2011), there have been a few works on learning the underlying individual model from aggregate data. The NLBP algorithm (Sun et al., 2015), a message passing type algorithm for approximate inference in CGMs, has been utilized in EM for the task of learning a Markov chain. Another existing aggregate inference algorithm utilized in the E-step of the EM algorithm is Bethe-RDA (Vilnis et al., 2015) which exhibits convergence guarantees. Finally, Bernstein and Sheldon (2016) proposed a method of moments estimator for learning a Markov chain within the CGM framework. Other works along this line include estimating spatiotemporal population flow (Iwata & Shimizu, 2019) and recurrent estimation of HMM (Lyubchyk, Grinberg, Dunaievska, & Lubchick, 2019) from aggregate data, learning stochastic behavior of aggregate data (Ma, Liu, Zha, & Zhou, 2020), learning hidden nonlinear dynamics from aggregate data (Wang, Dai, Kong, Erfani, Bailey, & Zha, 2018), and estimating group behavior from ensemble observations (Zeng, 2019).

The rest of the paper is organized as follows. In Section 2, we briefly discuss related background. We present our main results and algorithms in Section 3 for discrete observations. The counterpart with continuous observations is developed in Section 4 followed by experimental results in Section 5 and a concluding remark in Section 6.

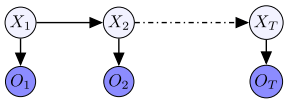


Fig. 1. A length T HMM.

2. Background

In this section, we present related background on HMMs, their extension to aggregate settings, and the CFB inference algorithm.

2.1. Hidden Markov models

An HMM is a Markov chain where the variables are not directly observable, but corresponding noisy variables are observed. Denote the unobserved hidden variables as $X_1, X_2, \ldots$ and observed variables as $O_1, O_2, \ldots$. Here X_t and O_t are random variables taking values from sets $\mathcal{X}$ and $\mathcal{O}$ respectively. In general, both $\mathcal{X}$ and $\mathcal{O}$ can be either finite sets or infinite sets. For discrete HMMs, $\mathcal{X}$ and $\mathcal{O}$ are finite sets with cardinalities $|\mathcal{X}| = d$ and $|\mathcal{O}| = s$, respectively.

A time-homogeneous HMM is parameterized by the initial distribution $\pi(X_1)$, the state transition probabilities $p(X_{t+1}|X_t)$, and the observation probabilities $p(O_t \mid X_t)$ independent of time steps $t = 1, 2, \ldots$. An HMM is a special type of probabilistic graphical model (PGM) (Wainwright & Jordan, 2008). The graphical representation of a length T HMM is shown in Fig. 1. The joint distribution of an HMM with length T factorizes as

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

where $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ and $\mathbf{o} = \{o_1, o_2, \ldots, o_T\}$ denote particular assignments to the hidden and observation variables, respectively.

One of the most important problems in HMMs is Bayesian inference where the goal is to calculate the posterior distributions of the hidden states X_t given a sequence of observations $\mathbf{o} = \{o_1, o_2, \ldots, o_T\}$. This is also known as filtering/smoothing (Murphy, 2012) in systems and control community. A well-known algorithm for this task is the standard forward–backward algorithm (Rabiner, 1989), which itself is a special case of belief propagation (Pearl, 1988) for Bayesian inference of general graphical models.

Another important problem in HMMs is the parameter learning, which is also known as system identification. Denote the set of parameters to be learned as

$$\theta = \{\pi(x_1), p(x_{t+1}|x_t), p(o_t|x_t)\}. \tag{2}$$

Let $\{\mathbf{o}^{(m)}\}_{m=1}^{M}$ with $\mathbf{o}^{(m)} = \{o_1^{(m)}, o_2^{(m)}, \ldots, o_T^{(m)}\}$ be a set of observed trajectories. The objective of parameter learning of HMMs is to estimate the parameter θ using the available data $\{\mathbf{o}^{(m)}\}_{m=1}^{M}$. Since the HMM is a latent variable model where the latent variable X_t is not observable, the maximum likelihood estimation cannot be applied directly. A popular approach for learning latent variable models is the expectation–maximization (EM) algorithm (Bishop, 2006; Neal & Hinton, 1998). The EM algorithm is an iterative method that involves two steps in each iteration: E-step and M-step. In the E-step, the values associated with the hidden variables are estimated to make the data complete and then, in M-step, the parameters of the underlying model are optimized based on the complete data likelihood. When specialized to HMMs, the EM algorithm reduces to the Baum–Welch algorithm (Koller & Friedman, 2009).

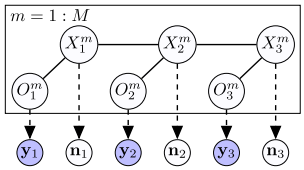


Fig. 2. Observation model of aggregate HMMs (shaded nodes represent aggregate observations).

2.2. Aggregate hidden Markov models

Aggregate HMM is a framework for learning and inference from noisy aggregate data generated from an HMM describing the behavior of individuals. It is a special case of the collective graphical model (Sheldon & Dietterich, 2011), which is a framework for general probabilistic graphical models. The aggregate data is generated from M independent individuals following an HMM. The HMMs are aggregate in the sense that they are indistinguishable to each other. Let $X_t^{(m)}$ be the (unobservable) state of the mth individual at time t and $O_t^{(m)}$ be the observable state. The observations are made in the form of $y_t(o_t) = \sum_{m=1}^{M} \mathbb{I}[O_t^{(m)} = o_t] = n_t^o(o_t)$, where $\mathbb{I}$ denotes the indicator function. It is the histogram of M observations over $\mathcal{O}$. The aggregate observation model for length of $T = 3$ is depicted in Fig. 2. Given these aggregate observations, the goal of inference in aggregate HMMs is to estimate the latent distributions $n_{t,t+1}(x_t, x_{t+1}) = \sum_{m=1}^{M} \mathbb{I}[X_t^{(m)} = x_t, X_{t+1}^{(m)} = x_{t+1}]$, $n_{t,t}(x_t, o_t) = \sum_{m=1}^{M} \mathbb{I}[X_t^{(m)} = x_t, O_t^{(m)} = o_t]$, and $n_t(x_t) = \sum_{m=1}^{M} \mathbb{I}[X_t^{(m)} = x_t]$.

The exact inference is proved to be computationally infeasible (Sheldon & Dietterich, 2011) for problems with large T and M. It is proposed in Singh et al. (2020) that this aggregate inference can be approximately achieved by solving a free energy minimization problem. Moreover, the approximation error vanishes as the size M of the population goes to infinity. The integral constraints on $\mathbf{n}$ can be relaxed (Singh et al., 2020) without affecting the precision much. With this relaxation, the latent distributions $\mathbf{n} = \{\mathbf{n}_t, \mathbf{n}_t^o, \mathbf{n}_{t,t}, \mathbf{n}_{t,t+1}\}$ satisfy the local polytope constraints

$$\sum_{x\in\mathcal{X}} n_t(x) = M, \forall t \in \{1, \dots, T\} \tag{3a}$$

$$\sum_{x\in\mathcal{X}} n_{t,t+1}(x, x_{t+1}) = n_{t+1}(x_{t+1}),$$

$$\sum_{x\in\mathcal{X}} n_{t,t+1}(x_t, x) = n_t(x_t), \quad \forall t \in \{1, \dots, T-1\} \tag{3b}$$

$$\sum_{o\in\mathcal{O}} n_{t,t}(x, o) = n_t(x), \quad \forall t \in \{1, \dots, T\} \tag{3c}$$

$$\sum_{x\in\mathcal{X}} n_{t,t}(x, o) = n_t^o(o), \quad \forall t \in \{1, \dots, T\}. \tag{3d}$$

Denote the local polytope described in (3) by $\mathbb{M}$ (Wainwright & Jordan, 2008). For aggregate HMMs, the free energy equals (Singh et al., 2020; Wainwright & Jordan, 2008)

$$\begin{aligned}\mathcal{F}(\mathbf{n}, \theta) = &- \sum_{t=1}^{T} \sum_{x_t, o_t} n_{t,t}(x_t, o_t) \log p(o_t|x_t) \\ &- \sum_{t=1}^{T-1} \sum_{x_t, x_{t+1}} n_{t,t}(x_t, x_{t+1}) \log p(x_{t+1}|x_t) \\ &- \sum_{x_1} n_1(x_1) \log \pi(x_1) - \sum_{x_1} n_1(x_1) \log n_1(x_1) \\ &- 2\sum_{t=2}^{T-1} \sum_{x_t} n_t(x_t) \log n_t(x_t) - \sum_{x_T} n_T(x_T) \log n_T(x_T) \\ &+ \sum_{t=1}^{T} \sum_{x_t, o_t} n_{t,t}(x_t, o_t) \log n_{t,t}(x_t, o_t) \\ &+ \sum_{t=1}^{T-1} \sum_{x_t, x_{t+1}} n_{t,t+1}(x_t, x_{t+1}) \log n_{t,t+1}(x_t, x_{t+1}).\end{aligned} \tag{4}$$

It is in fact equal to the Kullback–Leibler divergence between the inferred distribution and the prior distribution over the space of trajectories (Singh et al., 2020). The aggregate inference problem is equivalent (Singh et al., 2020) to the following convex optimization problem.

Problem 1.

$$\min_{\mathbf{n}\in\mathbb{M}} \mathcal{F}(\mathbf{n}, \theta) \tag{5a}$$

$$\text{subject to } \mathbf{n}_t^o = \mathbf{y}_t, \quad \forall t \in \{1, \dots, T\}. \tag{5b}$$

Thanks to the large deviation theory (Singh et al., 2020; Varadhan, 1984), the conditional distribution $p(\mathbf{n}|\mathbf{y}, \theta)$ of $\mathbf{n}$ given the observation $\mathbf{y}$ approximately concentrates on the solution to Problem 1. The very same approximation is the foundation of the Schrödinger bridge problem (Chen, Georgiou, & Pavon, 2016, 2021) which has been explored extensively in stochastic control.

In Singh et al. (2020), we proposed the SBP algorithm for solving aggregate inference problems over more general CGMs with tree-structure. The SBP algorithm has convergence guarantees with linear rate (Singh et al., 2020). There exist some other algorithms for aggregate inference problems in CGMs including approximate MAP (Sheldon, Sun, Kumar, & Dietterich, 2013), NLBP (Sun et al., 2015) and Bethe-RDA (Vilnis et al., 2015). One major difference between SBP and these methods is the observation model. In Sheldon et al. (2013), Sun et al. (2015) and Vilnis et al. (2015), the noise is added to the aggregate observation $\mathbf{y}_t$ directly, meaning the real observed histogram is a perturbed version of $\mathbf{y}_t$ by some random noise. In contrast, in Problem 1, we assume that the observation noise enters the system in the individual level and the measurement of the histogram is precise. It has the nice property that when $M = 1$, it reduces to a standard inference problem for PGMs or HMMs. We refer the reader to Singh et al. (2020) for more details on the comparison of the observation models.

2.3. Collective forward–backward algorithm

The collective forward–backward algorithm (CFB) is a special case of the general SBP algorithm when the underlying graphical model is an HMM (Singh et al., 2020). It is a message passing type algorithm, similar to BP, consisting of four types of messages. Fig. 3 depicts the messages employed by the CFB algorithm with $\alpha_t(x_t)$ being the messages in the forward direction and $\beta_t(x_t)$ being 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 nodes to observation nodes. These messages are characterized by

$$\alpha_t(x_t) \propto \sum_{x_{t-1}} p(x_t|x_{t-1}) \alpha_{t-1}(x_{t-1}) \gamma_{t-1}(x_{t-1}) \tag{6a}$$

$$\beta_t(x_t) \propto \sum_{x_{t+1}} p(x_{t+1}|x_t) \beta_{t+1}(x_{t+1}) \gamma_{t+1}(x_{t+1}) \tag{6b}$$

$$\gamma_t(x_t) \propto \sum_{o_t} p(o_t|x_t) \frac{y_t(o_t)}{\xi_t(o_t)} \tag{6c}$$

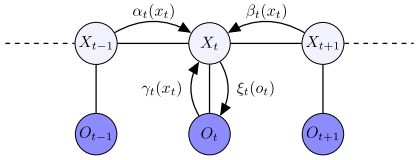


Fig. 3. Messages for inference in aggregate HMMs.

Algorithm 1 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 $\gamma_{t-1}(x_{t-1})$
    (ii) Update $\alpha_t(x_t)$, $\xi_t(o_t)$
  end for
  Backward pass:
  for $t = T-1, \ldots, 1$ **do**
    (i) Update $\gamma_{t+1}(x_{t+1})$
    (ii) Update $\beta_t(x_t)$, $\xi_t(o_t)$
  end for
end while

$$\xi_t(o_t) \propto \sum_{x_t} p(o_t|x_t)\alpha_t(x_t)\beta_t(x_t), \tag{6d}$$

with boundary conditions $\alpha_1(x_1) = \pi(x_1)$, $\beta_T(x_T) = 1$.

The sequence of update steps are listed in Algorithm 1.

Once the algorithm converges, which is guaranteed, the latent marginals can be estimated as

$$n_t(x_t) \propto \alpha_t(x_t)\beta_t(x_t)\gamma_t(x_t),$$
$$n_{t,t+1}(x_t, x_{t+1}) \propto p(x_{t+1}|x_t)\alpha_t(x_t)\gamma_t(x_t)\beta_t(x_{t+1})\gamma_t(x_{t+1})$$
$$n_{t,t}(x_t, o_t) \propto \frac{p(o_t|x_t)\alpha_t(x_t)\beta_t(x_t)}{\xi_t(o_t)}.$$

Clearly, when the population size is 1, i.e., $M = 1$, the CFB algorithm reduces to the standard Forward–backward algorithm for the inference of HMMs (Haasler, Singh, Zhang, Karlsson, & Chen, 2020; Singh et al., 2020).

3. Learning discrete aggregate HMMs

The learning problem in CGMs is concerned with estimating the individual model parameters of the underlying graphical model from aggregate observations. For learning the parameters of a latent variable model, the EM algorithm (Dempster et al., 1977) is a standard approach. The EM algorithm consists of two operations: the E-step to compute the log-likelihood of the observations given the current estimation of parameters, and the M-step to maximize the log-likelihood. The challenge to apply the EM algorithm for learning CGMs lies in the fact that the E-step requires inferring the conditional distribution of $\mathbf{n}$ on the observation $\mathbf{y}$, which is untractable (Sheldon et al., 2013; Singh et al., 2020).

In this section, we propose the approximate EM algorithm (Algorithm 2) for learning HMMs with observations in aggregate form. The key idea is to use the tractable CFB algorithm to approximately infer the aggregate distributions $\mathbf{n}$. Note that the SBP algorithm can be used for learning more general CGMs.

Theorem 2. *The Approximate EM algorithm converges.*

Proof. The E-step and M-step in Algorithm 2 are coordinate descent updates of the free energy $\mathcal{F}(\mathbf{n}, \theta)$ with respect to $\mathbf{n}$ and θ, and thus the objective function is monotonically decreasing. Moreover, since the free energy $\mathcal{F}$ in (4) is equal to Kullback–Leibler divergence between the inferred distribution and the prior distribution over the space of trajectories (Singh et al., 2020), it is bounded below by 0. Thus, in view of the fact that $\mathcal{F}$ is continuously differentiable, the approximate EM algorithm converges to a local minimum. ■

Algorithm 2 Approximate EM algorithm

Initialize model parameters θ^0
for $\ell = 1, 2, \ldots$ **do**
  E-step: Obtain the solution $\mathbf{n}^*$ to Problem 1 using CFB with parameters $\theta^{\ell-1}$
  M-step: $\theta^\ell = \operatorname{argmin}_\theta \mathcal{F}(\mathbf{n}^*, \theta)$
end for

We next argue that the log-likelihood $L(\theta^\ell) := \log p(\mathbf{y}|\theta^\ell)$ approximately monotonically increases. The improvement of L at the ℓth iteration is

$$\begin{aligned}
L(\theta^\ell) - L(\theta^{\ell-1}) &= \log \sum_{\mathbf{n}} p(\mathbf{y}, \mathbf{n} \mid \theta^\ell) - \log\ p(\mathbf{y} \mid \theta^{\ell-1}) \\
&= \log \sum_{\mathbf{n}} p(\mathbf{n}|\mathbf{y}, \theta^{\ell-1}) \frac{p(\mathbf{y}, \mathbf{n} \mid \theta^\ell)}{p(\mathbf{n}|\mathbf{y}, \theta^{\ell-1})} - \log\ p(\mathbf{y} \mid \theta^{\ell-1}) \\
&\geq \sum_{\mathbf{n}} p(\mathbf{n}|\mathbf{y}, \theta^{\ell-1}) \log \frac{p(\mathbf{y}, \mathbf{n} \mid \theta^\ell)}{p(\mathbf{n}|\mathbf{y}, \theta^{\ell-1})} - \log\ p(\mathbf{y} \mid \theta^{\ell-1}) \\
&= \sum_{\mathbf{n}} p(\mathbf{n}|\mathbf{y}, \theta^{\ell-1}) \log \frac{p(\mathbf{y}, \mathbf{n} \mid \theta^\ell)}{p(\mathbf{y}, \mathbf{n} \mid \theta^{\ell-1})}.
\end{aligned}$$

Since $p(\mathbf{n}|\mathbf{y}, \theta^{\ell-1})$ approximately concentrates on $\mathbf{n}^*$,

$$L(\theta^\ell) - L(\theta^{\ell-1}) \approx \log \frac{p(\mathbf{y}, \mathbf{n}^* \mid \theta^\ell)}{p(\mathbf{y}, \mathbf{n}^* \mid \theta^{\ell-1})}.$$

Again, due to the large deviation theory (Singh et al., 2020), $p(\mathbf{y}, \mathbf{n} \mid \theta) \approx \exp[-M\mathcal{F}(\mathbf{n}, \theta)]$. Thus,

$$\frac{1}{M}(L(\theta^\ell) - L(\theta^{\ell-1})) \approx -\mathcal{F}(\mathbf{n}^*, \theta^\ell) + \mathcal{F}(\mathbf{n}^*, \theta^{\ell-1}).$$

The approximate monotonicity of likelihood then follows from the definition of the M-step in Algorithm 2.

Thanks to the special structure of HMMs, the M-step can be implemented efficiently in closed-form.

Proposition 1. *The M-step in learning aggregate HMMs is given by*

$$\pi(x_1) = n_1(x_1), \tag{7a}$$
$$p(x_{t+1} \mid x_t) = \frac{\sum_{t=1}^{T-1} n_{t,t+1}(x_t, x_{t+1})}{\sum_{t=1}^{T-1} n_t(x_t)}, \tag{7b}$$
$$p(o_t \mid x_t) = \frac{\sum_{t=1}^{T} n_{t,t}(x_t, o_t)}{\sum_{t=1}^{T} n_t(x_t)}. \tag{7c}$$

Proof. See Appendix A. ■

Remark 1. If parts of the parameters are known, then we only need to update the other parameters in the M-step. For instance, if the emission probability $p(o_t|x_t)$ is known, then only (7a)–(7b) are needed for the M-step.

Algorithm 2 is for learning from a single sequence of aggregate data generated from a certain number of samples. Learning from multiple sequences of observations was initially explored in the

Baum–Welch algorithm (Rabiner, 1989). Building on the same idea, we extend it to the setting (Algorithm 3) with an ensemble of K number of aggregate observation sequences generated from the same HMM model. Note that here each aggregate observation is based on the collective information of M individuals, therefore K such aggregate observations in fact corresponds to $N = MK$ individuals.

Denote the ensemble of aggregate observations by $\{\mathbf{y}^k\}_{k=1}^K$, then in the E-step, we need to find the solution $\mathbf{n}^k$ to Problem 1 for each of these observations. In the M-step, one solves

$$\min_{\theta} \sum_{k=1}^{K} \mathcal{F}(\mathbf{n}^k, \theta).$$

This can again be expressed in closed form for HMMs as

$$\pi(x_1) = \frac{1}{K}\sum_{k=1}^{K} n_1^k(x_1) \tag{8a}$$

$$p(x_{t+1} \mid x_t) = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T-1} n_{t,t+1}^k(x_t, x_{t+1})}{\sum_{k=1}^{K}\sum_{t=1}^{T-1} n_t^k(x_t)} \tag{8b}$$

$$p(o_t \mid x_t) = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T} n_{t,t}^k(x_t, o_t)}{\sum_{k=1}^{K}\sum_{t=1}^{T} n_t^k(x_t)}. \tag{8c}$$

Algorithm 3 Learning HMMs with an ensemble of aggregate observations

Initialize $\pi(x_1)$, $p(x_{t+1} \mid x_t)$, $p(o_t \mid x_t)$
repeat
  Compute $\mathbf{n}^k$ by solving Problem 1 with measurement $\mathbf{y}^k$ using CFB for all $k = 1, \ldots, K$
  Update the parameters using (8)
until convergence

Proposition 2. *Algorithms 2 and 3 reduce to the Baum–Welch algorithm when observations are from populations of size $M = 1$.*

Proof. See Appendix B. ■

4. Learning aggregate HMMs with continuous observations

Next we turn our attention to the parameter learning problems of HMMs with continuous observation space $\mathcal{O} = \mathbb{R}^s$ (the state space $\mathcal{X}$ is still discrete). Such a HMM with continuous observation is similar to the discrete HMM except that it has a continuous emission density. The continuous observation model in standard HMMs has been studied in Juang (1985) and Juang, Levinson, and Sondhi (1986). In this section, we extend our learning algorithm to aggregate HMMs with continuous emission densities.

Suppose we have a total of M trajectories of continuous observations $\{o_1^{(m)}, o_2^{(m)}, \ldots, o_T^{(m)}\}$, $\forall m = 1, 2, \ldots, M$, $o_t^{(m)} \in \mathbb{R}^s$ over an HMM of length T. Note that the individuals are indistinguishable, which implies the order $\{o_t^{(1)}, o_t^{(2)}, \ldots, o_t^{(M)}\}$ at each time point t is arbitrary and meaningless. In discrete aggregate HMMs, the observation at time t can be summarized in a histogram $\mathbf{y}_t$. This is not an efficient representation of observation in the setting with continuous observations as it requires discretizing the observation space $\mathcal{O}$ which would be potentially expensive. Instead, we keep the observation at time t in its raw format $\{o_t^{(1)}, o_t^{(2)}, \ldots, o_t^{(M)}\}$, as a bunch of samples. Similarly, since the observation space is continuous, the joint distribution $n_{t,t}(x_t, o_t)$ is no longer an efficient representation. We instead use $n_t^{(m)}(x_t)$ to capture the association between the states and the observations.

Algorithm 4 Learning aggregate Gaussian-HMMs

Initialize $\pi(x_1)$, $p(x_{t+1} \mid x_t)$, $\mu(x_t)$, $\Sigma(x_t)$
repeat
  Compute $n_{t,t+1}(x_t, x_{t+1})$, $n_t(x_t)$, $n_t^{(m)}(x_t)$ using CFB
  Update the parameters using (12)
until convergence

Recently, the inference problem in aggregate HMMs with continuous emission densities has been studied in Zhang, Singh, and Chen (2020). It was shown that the latent marginals can be estimated as (Corollary 2, Zhang et al., 2020)

$$n_t(x_t) \propto \alpha_t(x_t)\beta_t(x_t)\gamma_t(x_t), \tag{9a}$$

$$n_{t,t+1}(x_t, x_{t+1}) \propto p(x_{t+1}|x_t)\alpha_t(x_t)\gamma_t(x_t)\beta_t(x_{t+1})\gamma_t(x_{t+1}) \tag{9b}$$

$$n_t^{(m)}(x_t) \propto \frac{p(o_t^{(m)}|x_t)\alpha_t(x_t)\beta_t(x_t)}{\xi_t(m)}, \tag{9c}$$

where $\alpha_t(x_t)$, $\beta_t(x_t)$, and $\gamma_t(x_t)$ are the messages in aggregate HMMs as depicted in Fig. 3. They correspond to the fixed point of the updates

$$\alpha_t(x_t) = \sum_{x_{t-1}} p(x_t|x_{t-1})\alpha_{t-1}(x_{t-1})\gamma_{t-1}(x_{t-1}), \tag{10a}$$

$$\beta_t(x_t) = \sum_{x_{t+1}} p(x_{t+1}|x_t)\beta_{t+1}(x_{t+1})\gamma_{t+1}(x_{t+1}), \tag{10b}$$

$$\gamma_t(x_t) = \frac{1}{M}\sum_{m=1}^{M} \frac{p(o_t^{(m)}|x_t)}{\xi_t(m)}, \tag{10c}$$

$$\xi_t(m) = \sum_{x_t} p(o_t^{(m)}|x_t)\alpha_t(x_t)\beta_t(x_t) \tag{10d}$$

with $\alpha_1(x_1) = \pi(x_1)$, $\beta_T(x_T) = 1$.

The inference estimates given by (9) are applicable to aggregate HMMs with any general continuous emission density. Next, we derive the formulas for parameter estimation of the underlying continuous observation HMM with Gaussian emission density.

Assuming the Gaussian noise model for emission density, it takes the form

$$p(o_t|x_t) = \mathcal{N}(o_t; \mu(x_t), \Sigma(x_t)), \tag{11}$$

i.e., each (discrete) hidden state corresponds to a single Gaussian density parameterized by mean $\mu(x_t)$ and variance $\Sigma(x_t)$. In such a model, an observation $o_t^{(m)}$ corresponding to the mth individual at time t is nothing but a sample from one of the Gaussian components.

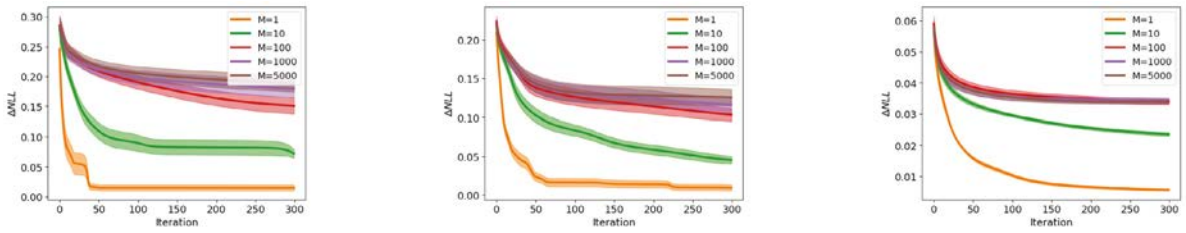
The learning of aggregate HMMs with Gaussian emission density can be achieved using the approximate EM algorithm with slightly modifications in the two steps. The E-step is an inference step using (9). The M-step has a closed-form expression given by the following Proposition. We omit its proof due to space constraints (it is similar to the proof of Proposition 1).

Proposition 3. *The M-step for aggregate HMMs with Gaussian emission density takes the form*

$$\pi(x_1) = n_1(x_1), \tag{12a}$$

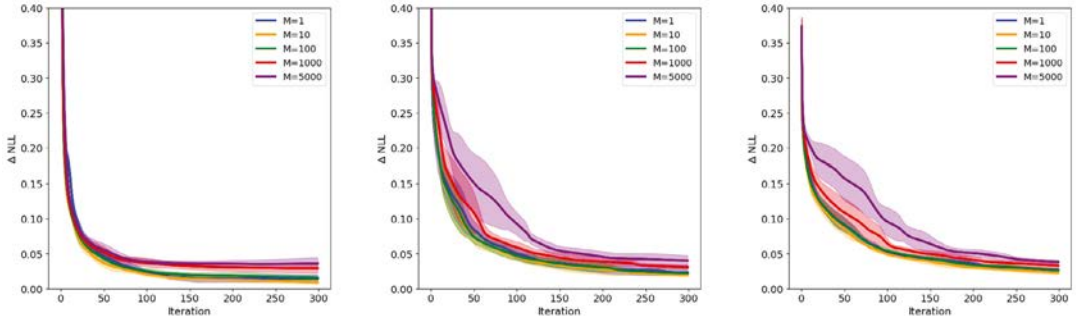
$$p(x_{t+1} \mid x_t) = \frac{\sum_{t=1}^{T-1} n_{t,t+1}(x_t, x_{t+1})}{\sum_{t=1}^{T-1} n_t(x_t)}, \tag{12b}$$

$$\mu(x_t) = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M} n_t^{(m)}(x_t)\, o_t^{(m)}}{\sum_{t=1}^{T} n_t(x_t)}, \tag{12c}$$



(a) $d = 3$ (b) $d = 5$ (c) $d = 10$

Fig. 4. Learning curves of HMMs with discrete observations. Curves in different color depict the results with different M. All three experiments share the same values of $T = 5, N = 5000$. The figures show how ΔNLL evolves with the number of iterations, for $d = 3$, $d = 5$ and $d = 10$ respectively. The shaded region represents standard deviation over 10 random seeds. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) $d = 10$ (b) $d = 20$ (c) $d = 25$

Fig. 5. Learning curves of various HMMs with Gaussian observation models. Curves in different color depict the results with different M. All three experiments are HMMs with $T = 5$ and $N = 5000$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\Sigma(x_t) = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M} n_t^{(m)}(x_t)(o_t^{(m)} - \mu_t)(o_t^{(m)} - \mu_t)'}{\sum_{t=1}^{T} n_t(x_t)} \tag{12d}$$

where prime denotes matrix transpose.

Based on Proposition 3, the parameters of a Gaussian-HMM are estimated using Algorithm 4. Note that in this aggregate Gaussian-HMM setting, the estimation updates for the initial distribution $\pi(x_1)$ and the transition probabilities $p(x_{t+1}|x_t)$ are the same as in Algorithm 2.

Remark 2. The convergence of Algorithm 4 follows from the same arguments as in the proof of Theorem 2.

Remark 3. Similar to discrete HMMs, one can extend Algorithm 4 to the setting with an ensemble of continuous aggregate observations.

5. Experiments

To illustrate the efficacy of the proposed aggregate learning algorithms, we perform multiple sets of experiments on synthetic as well as real-world dataset of population flow over a geographical area.

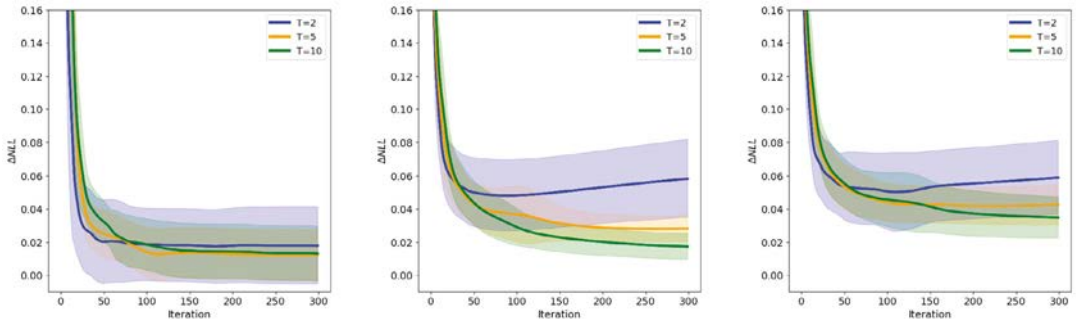
5.1. Learning HMMs with synthetic data

In this section, we consider synthetic data for evaluating our learning algorithms. We perform multiple sets of experiments for performance comparison of fitted time-invariant HMM models with discrete as well as continuous observations. The initial state probability $\pi(x_1)$ is sampled from the uniform distribution over the probability simplex. To produce the transition matrix, we first randomly permute rows of noised identity matrix $\mathcal{I}+0.05\times\sqrt{d}\times$ exp($Uniform[-1, 1]$). We scale rows of the permuted matrix so that the resulting matrix is a valid conditional distribution. For discrete observation setting, the emission matrix is generated in a similar way as transition matrix, but with a different random seed. In case of HMMs with continuous observations, we consider the Gaussian emission model. For each state, the corresponding Gaussian distribution is parameterized by a random mean and variance. The mean is sampled from $Uniform[-5d, 5d]$ and variance is from $Uniform[1, 5]$. In continuous observation setting, the algorithm is required to estimate the initial distribution, the transition matrix and the means of Gaussian emission densities. We generate N individual trajectories from the HMM parameterized with θ^* and aggregate them. Each aggregate sequence consists of collective observations of M independent trajectories of length T. The HMM parameters are learned based on $K = \frac{N}{M}$ number of aggregate sequences. For testing purpose, we generate another set of N individual trajectories.

We use the negative log likelihood (NLL) as a metric for evaluating performance of our learning algorithm. The difference of NLLs between the learned model θ and ground truth θ^* is

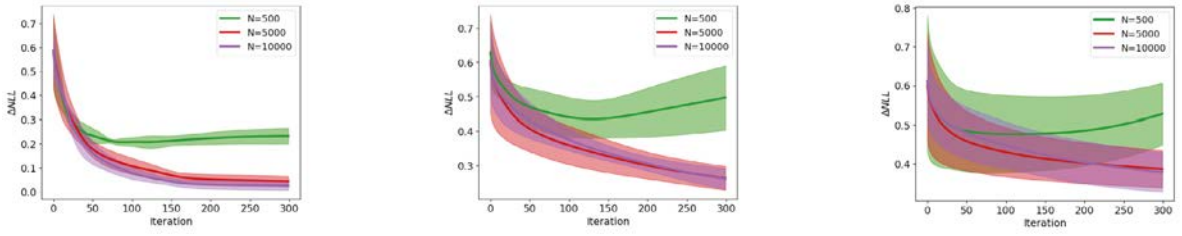
$$\Delta NLL(\theta) = \frac{1}{N}NLL(\theta) - \frac{1}{N}NLL(\theta^*).$$

The HMM model with learned parameters is evaluated on test dataset with the same number of total trajectories N as in training data. Fig. 4 shows the performance of our algorithm for different values of state dimension d and population size M on HMMs with discrete observations. Curves in the same figure show learning performance with different values of M but with fixed values for d, T, and N. It can be observed that one achieves best performance for the case of no aggregation ($M = 1$) and as the aggregate size M increases, ΔNLL also increases. Similar observations can be made for the case of Gaussian observation model as depicted in Fig. 5. It shows that our algorithm can effectively



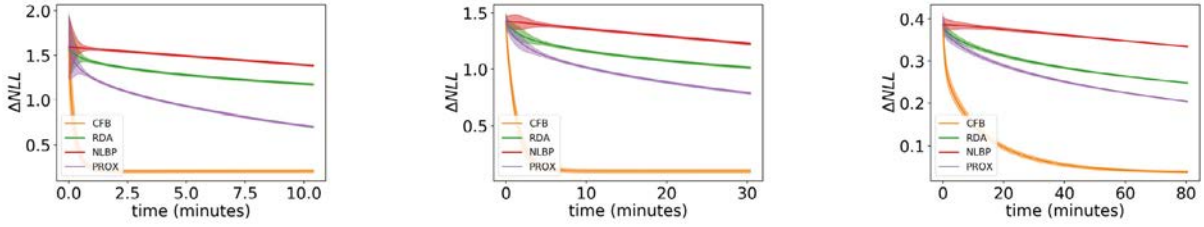
(a) $M = 1$ (b) $M = 100$ (c) $M = 250$

Fig. 6. Effect of the HMM length on the learning performance. Curves with different color correspond to different T values. All three experiments are Gaussian observation HMMs with $d = 5$ and $N = 500$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) $M = 1$ (b) $M = 10$ (c) $M = 100$

Fig. 7. Performance of aggregate learning with various data sizes. Curves with different color depict the learning curves with different data sizes N. The insufficient data causes overfitting to the training data. Our algorithm shows better performance with more samples available. All three experiments are discrete observation HMMs with $d = 10$, and $T = 10$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) $d = 3$ (b) $d = 5$ (c) $d = 10$

Fig. 8. Comparison of different learning algorithms under discrete HMM with $T = 5, N = 5000, M = 10$. We report evolution of ΔNLL with respect to wall time. It clearly shows that learning with CFB outperforms other existing approaches.

learn the generative models. Larger aggregate size corresponds to lower convergence rate as expected; with larger aggregate size, more information is lost about the individuals.

To further demonstrate the scalability of our algorithm, we conduct experiments with various HMM lengths and sample sizes as depicted in Fig. 6 and Fig. 7, respectively. In Fig. 6, the curves in different colors depict the learning performance with different HMM lengths T. We observe that larger T leads to better performance. This is because larger T is associated with more training data. Moreover, one can also observe that as the aggregate size M increases, the performance degrades as expected. Fig. 7 demonstrates the effect on HMM learning varying data sizes N. It can be observed that with more data available, the performance of our algorithm improves. With data size smaller than $N = 500$, the overfitting problem occurs; even though the algorithm converges on training data, the ΔNLL evaluated on test data tends to increase.

Next, we compare our algorithm with the learning framework involving NLBP (Sun et al., 2015), Bethe-RDA (Vilnis et al., 2015), and Prox (Singh et al., 2020) in Fig. 8. Since those algorithms assume different observation models, we only learn the initial distribution and transition matrix with given observations models for a fair comparison. For the cases of NLBP, Bethe-RDA, and Prox, we choose the explicit aggregate noise model following independent Poisson distributions for each aggregate state (see Sun et al. (2015) for more details on this aggregate noise model). We conduct experiments on discrete HMM with $T = 5, N = 5000$, and $M = 10$. The comparison of learning performances for different values of d is depicted in Fig. 8. One can clearly observe from Fig. 8 that our learning framework based on the CFB

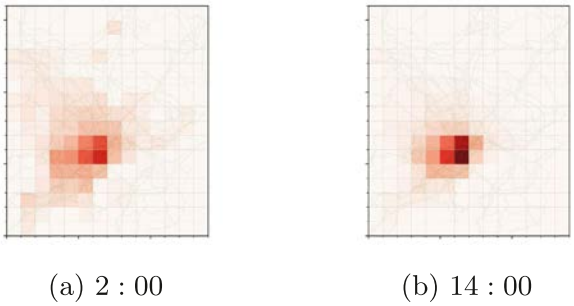


Fig. 9. Heatmap observation of population around the city of Tokyo. The whole area of the city is divided into 14 × 16 blocks. With more people stay in a block, color inside the block becomes deeper. The underlying green curves represent main roads around the city. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

algorithm converges faster and performs better than the existing aggregate learning approaches.

5.2. Estimating spatio-temporal population flow

We now turn to real-world aggregated data of population flow within the Japanese city of Tokyo. The dataset[1] consists of anonymous individual trajectories containing latitude and longitude of each person over time. The individual locations were recorded over time via geo-tagged tweets. We discretize the whole city area into 14 × 16 blocks with each block representing a 15 km × 15 km area, resulting in (hidden) state space dimension of $d = 224$. The observations are collected by aggregating the individual trajectories every 30 min. A total of 6,432, 9,166, 6,822, 10,134, 6,646, 10,338 trajectories were collected respectively on July 1, July 7, October 7, October 13, December 16 and December 29 in the year 2013. We assume that the observations are corrupted by Gaussian Noise. Additionally, with a small chance, a point in the center block can be categorized to eight neighboring blocks incorrectly, which account for sensor inaccuracy. In Fig. 9, we show the aggregate observation at timestamps 2:00 and 14:00 generated from the noisy observation model. The observations are based on all the individual trajectories recorded on the previously mentioned six number of days at corresponding times ($N = 49538$).

Our task is to estimate the transition probabilities characterizing the population flow at different times 2:00, 8:00, 14:00, and 20:00. The estimations at each time point are based on one and half hour window such that the underlying HMM has a length of $T = 3$. The observations are aggregated based on a time-homogeneous HMM with length $T = 3$ and aggregate size $M = 20$. We estimate the HMM parameters directly from the aggregated data. We consider the estimated parameters with $M = 1$ as the ground truth while assuming that the observation noise model is known. Fig. 10 depicts the comparison between our estimation and ground truth movement at the four timestamps. The red arrows in the figure implicitly represent the underlying transition probabilities multiplied by the total population $N = 49538$. One can observe that our algorithm successfully recovers the underlying movement of population with noisy aggregate observations.

[1] Data Source: SNS-based People Flow Data, Nightley, Inc., Shibasaki & Sekimoto Laboratory, the University of Tokyo, Micro Geo Data Forum, People Flow project, and Center for Spatial Information Science at the University of Tokyo, http://nightley.jp/archives/1954.

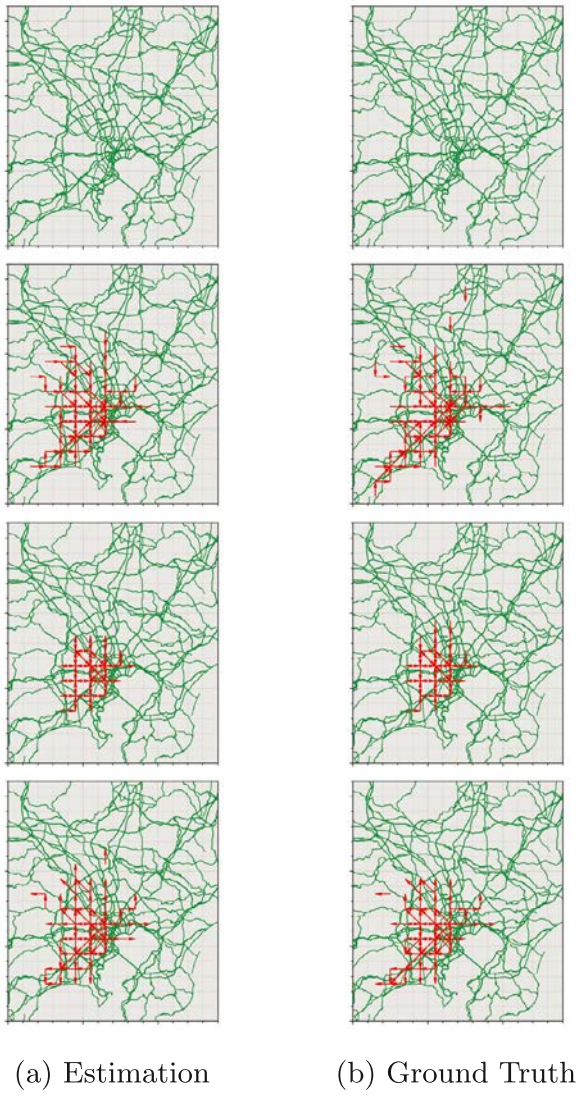


Fig. 10. Comparison between estimation based on our algorithm and ground truth movement. The four rows show the comparison at times 2:00, 8:00, 14:00 and 20:00, respectively. The red arrow depicts that flow between two block exceeds a threshold, 35. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6. Conclusion

In this paper, we proposed an algorithm for learning the parameters of a time-homogeneous HMM from aggregate data. Our algorithm is based on a modified version of the EM algorithm, wherein we utilized the Sinkhorn belief propagation algorithm to infer the unobservable states. In contrast to the existing state-of-the-art algorithms that explicitly consider the aggregate observation noise, our algorithm employs the aggregate observation noise within the graphical model and due to which it is consistent with the standard Baum–Welch algorithm when aggregate data consists of only a single individual. Moreover, our algorithm enjoys convergence guarantees. We further extended our algorithm to incorporate continuous observations and presented estimates for Gaussian observation model. In this work, we have assumed that the HMMs are time-homogeneous, which restricts the modeling capability of the data. We plan to explore learning of time-varying HMMs in our future research.

Appendix A. Proof of Proposition 1

Proof. The M-step in Algorithm 2 for aggregate HMMs solves

$$\min_{\theta} \quad \mathcal{F}(\mathbf{n}, \theta) \tag{A.1a}$$

$$\text{subject to} \quad \sum_{x_1} \pi(x_1) = 1, \tag{A.1b}$$

$$\sum_{x_{t+1}} p(x_{t+1} \mid x_t) = 1, \tag{A.1c}$$

$$\sum_{o_t} p(o_t \mid x_t) = 1, \tag{A.1d}$$

where $\theta = \{\pi(x_1), p(x_{t+1}|x_t), p(o_t|x_t)\}$ and $\mathcal{F}(\mathbf{n}, \theta)$ as in (4).

Let the Lagrange multipliers corresponding to the constraints (A.1b), (A.1c), and (A.1d) be λ, ν, and μ respectively. Then, the Lagrangian is

$$\mathcal{L}(\theta, \lambda, \nu, \mu) = \mathcal{F}(\mathbf{n}, \theta) - \sum_{x_t} \nu_{x_t} \Big(\sum_{x_{t+1}} p(x_{t+1} \mid x_t) - 1\Big)$$
$$- \lambda \Big(\sum_{x_1} \pi(x_1) - 1\Big) - \sum_{x_t} \mu_{x_t} \Big(\sum_{o_t} p(o_t \mid x_t) - 1\Big).$$

Setting the derivatives of the Lagrangian with respect to the variables to zero, we get

$$\frac{\partial \mathcal{L}}{\partial \pi(x_1)} = -\frac{n_1(x_1)}{\pi(x_1)} - \lambda = 0,$$

$$\frac{\partial \mathcal{L}}{\partial p(x_{t+1} \mid x_t)} = -\sum_{t=1}^{T-1} \frac{n_{t,t+1}(x_t, x_{t+1})}{p(x_{t+1} \mid x_t)} - \nu_{x_t} = 0,$$

$$\frac{\partial \mathcal{L}}{\partial p(o_t \mid x_t)} = -\sum_{t=1}^{T} \frac{n_{t,t}(x_t, o_t)}{p(o_t \mid x_t)} - \mu_{x_t} = 0.$$

Solving above equations, in view of the constraints (A.1b)–(A.1c)–(A.1d), we obtain

$$\pi(x_1) = n_1(x_1), \tag{A.2a}$$

$$p(x_{t+1} \mid x_t) = \frac{\sum_{t=1}^{T-1} n_{t,t+1}(x_t, x_{t+1})}{\sum_{t=1}^{T-1} n_t(x_t)}, \tag{A.2b}$$

$$p(o_t \mid x_t) = \frac{\sum_{t=1}^{T} n_{t,t}(x_t, o_t)}{\sum_{t=1}^{T} n_t(x_t)}. \quad \blacksquare \tag{A.2c}$$

Appendix B. Proof of Proposition 2

Proof. Here we only present proof for the statement related to Algorithms 2. It can be easily extended to the other part of the theorem. In case $M = 1$, the aggregate observation $\mathbf{y}$ corresponds to a sequence of observations $\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_T$. In particular, the aggregate observations take the form

$$y_t(o_t) = \delta(o_t - \hat{o}_t), \tag{B.1}$$

where $\delta(\cdot)$ denotes the Dirac function. Then the messages in collective forward–backward algorithm coincide with the messages in standard forward–backward algorithm (Singh et al., 2020) and take the following form

$$\alpha_t(x_t) \propto \sum_{x_{t-1}} p(x_t|x_{t-1}) \alpha_{t-1}(x_{t-1}) p(\hat{o}_{t-1}|x_{t-1}), \tag{B.2a}$$

$$\beta_t(x_t) \propto \sum_{x_{t+1}} p(x_{t+1}|x_t) \beta_{t+1}(x_{t+1}) p(\hat{o}_{t+1}|x_{t+1}), \tag{B.2b}$$

$$\gamma_t(x_t) = p(\hat{o}_t|x_t). \tag{B.2c}$$

Using above messages, the required marginals can be estimated as

$$n_t(x_t) \propto p(\hat{o}_t|x_t) \alpha_t(x_t) \beta_t(x_t), \tag{B.3a}$$

$$n_{t,t+1}(x_t, x_{t+1}) \propto \alpha_t(x_t) p(x_{t+1}|x_t) \beta_t(x_{t+1})$$
$$p(\hat{o}_t|x_t) p(\hat{o}_{t+1}|x_{t+1}), \tag{B.3b}$$

$$n_{t,t}(x_t, \hat{o}_t) = n_t(x_t). \tag{B.3c}$$

Finally, the parameter update equations given in Algorithm 2 reduce to the standard Baum–Welch algorithm. $\blacksquare$

References

Baum, Leonard E., & Eagon, John Alonzo (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *American Mathematical Society. Bulletin*, *73*(3), 360–363.

Baum, Leonard E., Petrie, Ted, Soules, George, & Weiss, Norman (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, *41*(1), 164–171.

Bernstein, Garrett, & Sheldon, Daniel (2016). Consistently estimating Markov chains with noisy aggregate data. In *Artificial intelligence and statistics* (pp. 1142–1150).

Bishop, Christopher M. (2006). *Pattern recognition and machine learning*. springer.

Cappé, Olivier, Moulines, Eric, & Rydén, Tobias (2006). *Inference in hidden Markov models*. Springer Science & Business Media.

Chen, Yongxin, Georgiou, Tryphon T., & Pavon, Michele (2016). On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications*, *169*(2), 671–691.

Chen, Yongxin, Georgiou, Tryphon T., & Pavon, Michele (2021). Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a Schrödinger bridge. *SIAM Review*, *63*(2), 249–313.

Dempster, Arthur P., Laird, Nan M., & Rubin, Donald B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *39*(1), 1–22.

Dong, Wen, Pentland, Alex Sandy, & Heller, Katherine A. (2012). Graph-coupled HMMs for modeling the spread of infection. In *Proceedings of the twenty-eighth conference on uncertainty in artificial intelligence* (pp. 227–236).

Haasler, Isabel, Singh, Rahul, Zhang, Qinsheng, Karlsson, Johan, & Chen, Yongxin (2020). Multi-marginal optimal transport and probabilistic graphical models. arXiv preprint arXiv:2006.14113.

Iwata, Tomoharu, & Shimizu, Hitoshi (2019). Neural collective graphical models for estimating spatio-temporal population flow from aggregated data. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33* (pp. 3935–3942).

Juang, B.-H. (1985). Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Technical Journal*, *64*(6), 1235–1249.

Juang, Bing-Hwang, Levinson, Stephene, & Sondhi, M. (1986). Maximum likelihood estimation for multivariate mixture observations of markov chains (corresp.). *IEEE Transactions on Information Theory*, *32*(2), 307–309.

Kalbfleisch, John David, Lawless, Jerald Franklin, & Vollmer, William M (1983). Estimation in Markov models from aggregate data. *Biometrics*, 907–919.

King, Gary (2013). *A solution to the ecological inference problem: reconstructing individual behavior from aggregate data*. Princeton University Press.

Koller, Daphne, & Friedman, Nir (2009). *Probabilistic graphical models: principles and techniques*. MIT Press.

Lee, Tsoung-Chao, Judge, George G., & Zellner, Arnold (1970). *Estimating the parameters of the Markov probability model from aggregate time series data*. North-Holland.

Luo, Dixin, Xu, Hongteng, Zhen, Yi, Dilkina, Bistra, Zha, Hongyuan, Yang, Xiaokang, et al. (2016). Learning mixtures of Markov chains from aggregate data with structural constraints. *IEEE Transactions on Knowledge and Data Engineering*, *28*(6), 1518–1531.

Lyubchyk, Leonid, Grinberg, Galyna, Dunaievska, Olha, & Lubchick, Maria (2019). Recurrent estimation of hidden Markov model transition probabilities from aggregate data. In *2019 9th international conference on advanced computer information technologies (ACIT)* (pp. 64–67). IEEE.

Ma, Shaojun, Liu, Shu, Zha, Hongyuan, & Zhou, Haomin (2020). Learning stochastic behaviour of aggregate data. arXiv preprint arXiv:2002.03513.

MacRae, Elizabeth Chase (1977). Estimation of time-varying Markov processes with aggregate data. *Econometrica*, 183–198.

Murphy, Kevin P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.

Neal, Radford M., & Hinton, Geoffrey E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models* (pp. 355–368). Springer.