Learning Large Graph-based MDPs with Historical Data

Ravi N. Haksar, *Student Member, IEEE,* and Mac Schwager, *Member, IEEE*

***Abstract*—We consider learning the dynamics and measurement model parameters of a graph-based Markov decision process (GMDP) given a history of measurements. Graph-based models have been used in modeling many data-based applications, such as recognition tasks, disease epidemics, forest wildfires, freeway traffic, and social networks. We leverage the Expectation-Maximization framework and develop an algorithm that optimizes the measurement likelihood and has favorable complexity for large models. In contrast to prior work, we directly consider GMDPs with significantly large discrete state spaces, arbitrary coupling structure, and long measurement sequences. We also consider a special structural property called Anonymous Influence, which we use to test hypotheses and gain insights into the data. We demonstrate the effectiveness of our learning algorithm by considering two real-world data sets, on the 2020 Novel Coronavirus (COVID-19) pandemic in California and on user interactions on Twitter. Our results show that the learned GMDP models better explain the data compared to an uncoupled model assumption.**

***Index Terms*—Learning, Markov processes, optimization, social networks, stochastic/uncertain systems.**

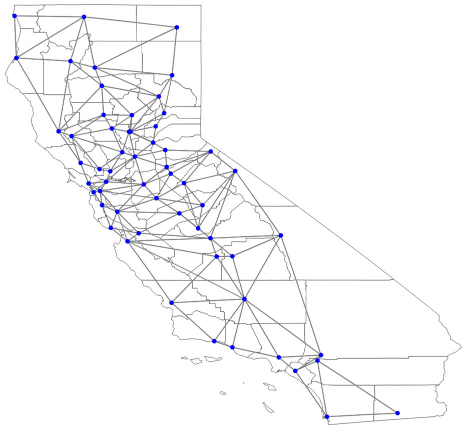


Fig. 1. A graph-based MDP is useful for describing the 2020 COVID-19 pandemic in California. Each vertex corresponds to a county (blue circles) and edges indicate major transportation routes (gray lines). Our experiments provide insight into the data, such as which counties are more susceptible to intra-county spread versus inter-county spread (and vice-versa).

I. Introduction

In this work, we address the model learning problem for a class of discrete time and discrete space structured models known as graph-based Markov decision processes (GMDPs). Many application domains have been modeled as a structured system of interacting Markov processes, such as recognition tasks [1], [2], biomedical data [3]–[5], disease epidemics [6], [7], forest wildfires [6], freeway traffic [8], music theory [9], [10], and social networks [11]–[14]. Structured Markov models have a number of benefits over standard Markov model formulations as well as other stochastic process modeling frameworks. First, structured representations can directly incorporate prior information about the process that generated the time series data. Second, with respect to other frameworks, there is significant research on using structured Markov models, e.g., for optimal control [15]–[17] or inference [7], which allow them to be deployed in real-world applications.

Our prior work has further developed algorithms for various applications based on GMDPs. Specifically, we previously developed methods for the optimal control problem with capacity constraints [6], [18], for online state estimation [19], and for constrained control under measurement uncertainty [20]. In addition, we considered applying a cooperative team of autonomous robots to a process modeled by a GMDP [21], [22]. In these works, we assumed prior knowledge of the parameters of the dynamics and measurement models, as well as the graph structure. In this work, we assume the graph is known, and we consider learning the dynamics and measurement model parameters from data. Therefore, by developing the model learning algorithm described in this paper, we complete a full algorithmic framework for controlling GMDPs in a practical setting, consisting of model learning, state inference, and optimal control.

For the graph-based models we consider in this work, each vertex in the graph corresponds to a standard MDP and edges between vertices indicate the coupling interactions which influence the dynamics of MDPs. A measurement model is also associated with each MDP, which describes the likelihood of observing different states of the MDP. We specifically consider GMDPs with a property commonly found in large-scale models, called “Anonymous Influence.” A GMDP has the Anonymous Influence property if the state dynamics of the constituent MDPs are based on the number of neighbor MDPs in particular states and not the identity of the neighbor MDPs. Furthermore, using Anonymous Influence allows us to develop structured experiments to investigate different influence structures that may exist in real-world data. For example, to what extent does confirmation bias play a role in user activity and interactions in social networks? For a disease epidemic,

This research was supported by NSF grant IIS-1646921, DARPA YFA award D18AP00064, and ONR grant N00014-18-1-2830. We are grateful for this support.

R. N. Haksar is with the Department of Mechanical Engineering, Stanford University, Stanford, CA, 94305 USA (e-mail: rhaksar@stanford.edu).

M. Schwager is with the Department of Aeronautics & Astronautics, Stanford University, Stanford, CA, 94305 USA (e-mail: schwager@stanford.edu).

does intra-community or inter-community transmission play a bigger role in daily case counts? We design experiments to evaluate hypotheses like these and gain insight into the data.

Learning algorithms for structured Markov models, and in particular graph-based Markov models, have been previously developed in literature and we review the relevant prior work in the next section. For the most part, relevant prior methods have focused mainly on relatively simple coupled models, e.g., few number of individual processes, or exploiting additional structure, e.g., conjugate or differentiable distributions. In contrast, we consider arbitrarily coupling between the discrete state dynamics of MDPs when there is potentially a large number of MDPs in the GMDP. Our approach is based on the Expectation-Maximization (EM) framework which leads to an approximately-optimal method with favorable complexity for large GMDPs.

The main contributions of this work are: (1) we develop a tractable learning algorithm for GMDPs with very large state spaces, based on the EM framework; (2) we apply Anonymous Influence to two real-world domains with publicly available data; (3) we show in our results that our algorithm learns meaningful models, which validates our modeling and framework assumptions; and (4) we show that the GMDP framework can describe different domains without significant hand-tailoring of models, as typically required in prior work.

The remainder of this paper is organized as follows. We review relevant related work in Section II. In Section III, we review the GMDP framework and the related learning problem, describe the Anonymous Influence property, and present models for the COVID-19 pandemic in California and user interactions on Twitter. In Section IV, we first provide background on the EM framework and the complexity of the standard approach. We then derive our approximate EM method and discuss the resulting complexity. In Section V, we introduce performance metrics for evaluating learned models, provide details on two real-world data sets, and demonstrate the benefits of our approach over an uncoupled model assumption. We provide concluding remarks in Section VI.

II. Related Work

Model learning algorithms have been presented in literature for a variety of application domains and modeling assumptions, and we are particularly interested in discrete structured Markov models. An equivalent standard hidden Markov model (HMM) formulation can be created from a GMDP model, by taking the product space of the individual MDP state and measurement spaces. From there, standard approaches in literature can be leveraged, e.g., Baum-Welch [23], Expectation-Maximization (EM) [24], and mean-field networks [25]. However, these methods quickly become intractable for moderately sized GMDPs due to the significantly large discrete state space of the overall GMDP, as there is an exponential dependence on the number of constituent MDPs. To address this challenge, the factorial HMM (FHMM) [9] and the coupled HMM (CHMM) [1] frameworks have been formulated, along with suitable methods to leverage model structure.

Several optimization frameworks have been applied to the learning problem for FHMMs and CHMMs, starting with approximate EM techniques [9] which were later expanded upon [26], [27]. Notably, the FHMM framework [9] consists of coupling only between the measurements of the individual HMMs, and the state dynamics are independent. Other work expanded on this formulation by considering structured state couplings, such as a hierarchical coupling structure [10] and a convex combination of simpler models [27]. While these works are less general cases of the models we consider, they still provide useful insight into parameter learning techniques.

The CHMM framework, in contrast, directly considers coupling interactions in the state dynamics, and initial learning methods were efficient techniques in the EM framework [1]. CHMMs quickly became popular for many applications and several techniques were developed, such as sampling methods [4], [8], neural networks [2], and variable clustering [8]. Many methods require a structured coupling model, such as a linear combination of marginal probabilities [3]. Notably, Raghavan et al. [12] proposed a simple CHMM consisting of two HMMs to model a social network, where one is a single user and the other is a sufficient statistic representing the influence exerted by other users. Although this sufficient statistic is a similar idea to using Anonymous Influence, this approach must be applied to each user sequentially which results in approximating the graph structure and the spread of influence. In contrast, we consider all users simultaneously to accurately describe the spread of influence, and our approach can be viewed as a generalization of the ideas proposed in [12]. Typically, prior work on CHMMs consider only a few interacting HMMs, whereas we consider GMDPs with a significantly large number of MDPs. Specifically, we consider models with tens to hundreds of MDPs. Even if each MDP has relatively few states, the resulting state space of the GMDP is intractable to enumerate, e.g., a GMDP consisting of 100 MDPs each with 3 states has 10^{47} total states.

Few works have directly considered the more general case of GMDPs, especially for the learning problem. Notably, Dong et al. [7] develop a Gibbs sampling strategy for a specific graph-based epidemiology model, which relies on conjugate distributions and sparse graph structure. The authors use a relatively smaller graph-based model than the ones we consider, and acknowledge the difficulty in generalizing this approach to other processes. A sampling-based method is computationally infeasible for our models without introducing more structure and assumptions, due to the large discrete state space of GDMPs. We specifically aim to address GMDPs with a large number of MDPs and arbitrary coupling structure, and therefore we leverage the EM framework. Finally, we note that other works that consider different models for coupled stochastic systems, such as other Markov model formulations [11], [13] and stochastic differential equations [14], are typically formulated for a specific application. In contrast, we wish to consider a more general class of models to draw connections between many different domains, and to provide additional insight and theoretical analysis.

In the next section, we review the GMDP framework, discuss the Anonymous Influence property, and introduce models for the COVID-19 pandemic in California and user interactions on Twitter.

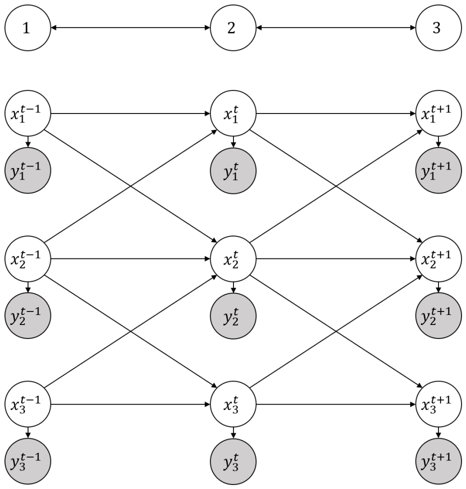


Fig. 2. (top) An example GMDP consisting of three vertices, each of which represents an MDP, where arrows indicate the mutual influence between MDPs. (bottom) The underlying graphical model of the example GMDP, where arrows indicate influence between time steps.

III. Modeling

A. Graph-based Markov Decision Processes

We first describe the graph-based Markov decision process (GMDP) framework [7], [16]. Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph with vertex set $\mathcal{V} = \{1, \ldots, n\}$ containing n vertices and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. In the standard GMDP model, each vertex $i \in \mathcal{V}$ corresponds to an MDP with latent state $x_i^t \in \mathcal{X}_i$, action $a_i^t \in \mathcal{A}_i$, and measurement $y_i^t \in \mathcal{Y}_i$ at time t. For the GMDP models we consider in this work, we drop the action and consider learning models without estimating the sequence of actions. In future work, we plan to introduce an action model which generalizes to many different domains, and develop algorithms and analysis for this more general case. See Fig. 2 for a visualization of the GMDP structure we use in this work. In a GMDP, the dynamics of MDP i are influenced by its neighbors, which are defined as follows.

Definition 1 (Neighbor Set)**.** For any set of vertices in the graph $\mathcal{S} \subseteq \mathcal{V}$, the neighbor set $\mathcal{N}(\mathcal{S}) : \mathcal{S} \to \mathcal{T} \subseteq \mathcal{V}$ is,

$$\mathcal{N}(\mathcal{S}) = \bigcup_{i \in \mathcal{S}} \{j \mid (j, i) \in \mathcal{E}\}.$$

In the case that $\mathcal{S} = \{i\}$, this is the typical neighbor set of vertex i, i.e., the other vertices that are connected by an edge to vertex i. However, our notion of a neighbor set also expresses the notion of neighbors of a set of vertices $\mathcal{S} \subseteq \mathcal{V}$. For the neighbor set of MDP i, we let $\mathcal{N}(\{i\}) = \mathcal{N}(i)$. Subscripts indicate the latent states or measurements of a subset of MDPs, e.g., x_i^t for MDP i and $x_{\mathcal{N}(i)}^t = \{x_j^t \mid j \in \mathcal{N}(i)\}$ for the neighbors of MDP i. We likewise use subscripts for the domain of variables, e.g., $x_{\mathcal{N}(i)}^t \in \mathcal{X}_{\mathcal{N}(i)} = \prod_{j \in \mathcal{N}(i)} \mathcal{X}_j$. We omit the subscript for the combination of all MDP states or measurements, $x^t = \{x_1^t, \ldots, x_n^t\} \in \mathcal{X}$ and $y^t = \{y_1^t, \ldots, y_n^t\} \in \mathcal{Y}$.

In this work, we assume the graph structure G to be known, and we plan to consider structure learning methods in future work. Our learning algorithm can directly be used in an iterative approach, where the first step determines the best graph given an estimate of the model parameters, and the second step uses our algorithm to update the parameters given the graph. We plan to investigate iterative methods, and other optimization-based approaches, to learn both the structure and the parameters from data.

The probability of transitioning from a state x_i^t to x_i^{t+1} for an MDP in the GMDP model is independent of all other variables, given the current state of the MDP x_i^t and the state of its neighbors $\{x_j^t \mid j \in \mathcal{N}(i)\}$ in the graph. Hence the dynamics can be written compactly as,

$$p_i(x_i^{t+1} \mid x_i^t, x_{\mathcal{N}(i)}^t). \tag{1}$$

One goal in the model learning problem is to determine the parameters of the dynamics model for every MDP in a GMDP. We represent the dynamics as a time-invariant matrix $\Lambda_i \in \mathbb{R}^{|\mathcal{X}_i| \prod_{j \in \mathcal{N}(i)} |\mathcal{X}_j| \times |\mathcal{X}_i|}$. The rows of this matrix must sum to one to represent the dynamics model of an MDP. We use brackets and subscripts to refer to specific elements (or locations) of matrices representing the dynamics, e.g.,

$$p_i(x_i^{t+1} \mid x_i^t, x_{\mathcal{N}(i)}^t) = [\Lambda_i]_{x_i^t, x_{\mathcal{N}(i)}^t, x_i^{t+1}}\,, \tag{2}$$

refers to the probability of MDP i transitioning from state x_i^t to x_i^{t+1}, given the state of its neighbors $x_{\mathcal{N}(i)}^t$. The dynamics without control for the aggregate state x^t describing the combination of all MDP states is then,

$$p(x^{t+1} \mid x^t) = \eta \prod_{i \in \mathcal{V}} p_i(x_i^{t+1} \mid x_i^t, x_{\mathcal{N}(i)}^t) \tag{3}$$

where η is a normalization constant. Measurements for each MDP are conditionally independent given the state of the underlying MDP,

$$p_i(y_i^t \mid x_i^t). \tag{4}$$

For this work, another goal of the model learning problem is determining the parameters of the measurement model for every MDP in a GMDP and we specifically consider the case of a continuous distribution and real-valued measurements y_i^t for (2). The measurement likelihood for the aggregate state is described by the distribution,

$$p(y^t \mid x^t) = \prod_{i \in \mathcal{V}} p_i(y_i^t \mid x_i^t). \tag{5}$$

We provide details of the specific measurement models we use in our case studies in Sections III-C and III-D.

For the remainder of this paper, we represent summing out (i.e., marginalizing) all MDP latent states from a distribution by $\sum_{x^t} = \sum_{x_1^t} \cdots \sum_{x_n^t}$. The marginalization of a subset of variables is specified in the summation, e.g., marginalizing out a neighbor set is $\sum_{x_{\mathcal{N}(i)}^t}$. We assume the data consists of a sequence of measurements for each vertex over the set of uniformly-spaced time steps starting at $t = 2$ and ending at time $t = T$, i.e., the data is $\{y_i^t \mid i \in \mathcal{V}, t \in [2, T]\}$. We use the notation $y_i^{2:T}$ to refer to the measurements for vertex i and the notation $y^{2:T}$ to refer to the measurements for all vertices. Likewise, we use $x_i^{1:T}$ and $x^{1:T}$ to refer to state trajectories for a single vertex and for all vertices, respectively. Note that an initial belief $p_i(x_i^1)$ is specified prior to the first measurement.

Our goal in the model learning problem is to maximize the log-likelihood of the measurements over the set of dynamics and measurement model parameters. Formally, we seek the parameters $\theta = \{\Lambda_i, p_i(y_i^t \mid x_i^t) \mid i \in \mathcal{V}\}$ which maximize,

$$\log p_\theta(y^{2:T}) = \log \sum_{x^{1:T}} p_\theta(y^{2:T}, x^{1:T}), \tag{6}$$

where the joint probability of the measurements and state trajectories is given by,

$$p_\theta(y^{2:T}, x^{1:T}) = \prod_{i \in \mathcal{V}} p_i(x_i^1) \prod_{t=2}^{T} p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}) p(y_i^t \mid x_i^t). \tag{7}$$

We use the notation p_θ to explicitly indicate the dependence of the joint probability on the unknown model parameters θ. We emphasize here that maximizing this objective is intractable as it requires enumerating all possible trajectories of the aggregate state, $x^{1:T}$. In particular, we consider GMDPs with significantly large state spaces and potentially long measurement sequences. Therefore, we consider approximations within the Expectation-Maximization framework to develop an approach with favorable complexity for large GMDPs, as we discuss in Section IV. Next, we provide details on a common structural property in GMDPs known as Anonymous Influence.

B. Anonymous Influence

We consider the case where (2) is based on the number of neighbors in particular states, rather than the identity of these neighbors. This property is called "Anonymous Influence" and we summarize the relevant ideas [28], [29].

We use $\mathbb{I}(x_i = j)$ to represent the indicator function which equals one when $x_i = j$ and zero otherwise. For a set of n discrete variables $x_i \in \{0, 1, \ldots, \mathcal{D} \in \mathbb{Z}_{\geq 0}\}$, the count aggregator (CA) is a vector $z \in \mathbb{Z}_{\geq 0}^{\mathcal{D}}$ where each element describes the number of variables taking on a particular value, $[z]_j = \sum_{i=1}^{n} \mathbb{I}(x_i = j)$ and $j \in \{0, 1, \ldots, \mathcal{D}\}$. A mixed-mode function (MMF) uses a CA, as well as other discrete arguments not part of a CA, and maps to the real numbers $\mathbb{R}$.

For a GMDP where all MDPs have the same discrete domain $x_i^t \in \{0, 1, \ldots, \mathcal{D}\}$, the dynamics (2) for each MDP requires specifying (at most) $(\mathcal{D}+1)^{|\mathcal{N}(i)|+2}$ values. If a CA z_i^t is used to represent the influence of other MDPs, then (2) can be represented by a MMF,

$$p_i(x_i^{t+1} \mid x_i^t, x_{\mathcal{N}(i)}^t) = p_i(x_i^{t+1} \mid x_i^t, z_i^t). \tag{8}$$

This formulation requires specifying (at most) $(D+1)^2 \cdot \binom{|\mathcal{N}(i)|+D}{|\mathcal{N}(i)|}$ values, where $\binom{n}{k}$ is the binomial coefficient, and is typically a significant reduction. For the models in this work, each MDP has the same state space and uses the same CA definition for its dynamics. In general, each MDP can have a unique state space and CA definition, and use an MMF where the identity of some neighbors is important. Given the complexity of developing a model learning algorithm, we present relatively simple models to illustrate the key aspects of GMDPs and Anonymous Influence, and we are developing experiments to investigate different model formulations in future work. We illustrate the Anonymous Influence property in the discussion of our COVID-19 and Twitter models, which we present next.

C. Case Study: COVID-19 Pandemic in California

We use a GMDP to describe the spread of the Novel Coronavirus (COVID-19) in California, USA in 2020, which is used in our model fitting algorithm. We use an undirected graph to represent California, where each vertex $i \in \mathcal{V}$ corresponds to a county and there are 58 total counties. Edges between counties indicate major transportation connections; see Fig. 1. The state of each county x_i^t has one of four values, $\mathcal{X}_i = \{1, 2, 3, 4\}$, which is an Alert Level corresponding to the current level of risk and number of cases. The state dynamics of each county are influenced by the count aggregator (CA),

$$b_i^t = \sum_{j \in \mathcal{N}(i)} \mathbb{I}(x_j^t \in \{3, 4\}),$$

which corresponds to the number of neighbors above Alert Level 2. For a given county i, the quantity b_i^t can take on $|\mathcal{N}(i)| + 1$ values, $b_i^t \in \{0, 1, \ldots, |\mathcal{N}(i)|\}$. Therefore, the county dynamics can be represented by a matrix $\Lambda_i \in \mathbb{R}^{4(|\mathcal{N}(i)|+1) \times 4}$, where each row must sum to one.

We use publicly available health data in order to learn the state dynamics and measurement likelihood parameters. The data consists of a daily case count for all 58 counties in California, normalized by county population, from March 18th, 2020 to October 28th, 2020 [30]. The model has 10^{34} total states and $T = 226$ time intervals. The initial belief of each county is based on the first observed case count. If there are a non-zero number of initial cases, $y_i^2 > 0$, the belief is,

$$p_i(x_i^1) = \begin{cases} 0.4 & \text{if } x_i^1 = \text{Alert Level 2}, \\ 0.2 & \text{otherwise.} \end{cases} \tag{9}$$

If there are no initial cases, $y_2^1 = 0$, then the belief is,

$$p_i(x_i^1) = \begin{cases} 0.4 & \text{if } x_i^1 = \text{Alert Level 0}, \\ 0.2 & \text{otherwise.} \end{cases} \tag{10}$$

The measurement model is based on the Normal distribution,

$$\begin{aligned} p_i(y_i^t \mid x_i^t = s) &= \text{Normal}(y_i^t; s, \sigma_{is}) \\ &= \sigma_{is}^{-1} (2\pi)^{-1/2} \exp\Big(-\frac{1}{2}(y_i^t - s)^2 \sigma_{is}^{-2}\Big), \end{aligned} \tag{11}$$

where the mean of the Normal distribution is the state value, $x_i^t \in \mathcal{X}_i = \{0, 1, 2, 3\}$, and the standard deviation associated with each state value are the unknown parameters, $\{\sigma_{is} \in \mathbb{R} \mid s \in \mathcal{X}_i, i \in \mathcal{V}\}$. Furthermore, we also scale the measurements by a constant $c = 4000$, to correlate the Alert Level of a county with the expected number of daily cases scaled by population.

For our COVID-19 model, the unknown parameters are the dynamics model and the standard deviations in the measurement model for each county. Therefore, the model parameters for this GMDP are $\theta = \{\Lambda_i, \sigma_{is} \mid \forall i \in \mathcal{V}, s \in \mathcal{X}_i\}$.

D. Case Study: User Interactions on Twitter

We also consider a social network model for our model fitting algorithm, which is based on Twitter interactions. We use an undirected graph to represent different users and connections between users. Each vertex $i \in \mathcal{V}$ corresponds to a unique user, and an edge exists between users i and j if user i interacts with user j, e.g., through mentioning or retweeting. The state of each user x_i^t has one of five values,

$$\begin{aligned}\mathcal{X}_i = \{--, -, \circ, +, ++\} = \{&\text{negative, leaning negative,}\\ &\text{neutral,}\\ &\text{leaning positive, positive}\},\end{aligned}$$

which corresponds to the user's opinion on a given topic. Each user's dynamics is influenced by the count aggregator (CA),

$$b_i^t = \sum_{j \in \mathcal{N}(i)} \mathbb{I}(x_j^t \neq \circ),$$

which corresponds to the number of neighbors which are not neutral on the topic. The quantity b_i^t can take on $|\mathcal{N}(i)|+1$ values, $b_i^t \in \{0, 1, \ldots, |\mathcal{N}(i)|\}$. The dynamics for each user can then be represented by a transition matrix $\Lambda_i \in \mathbb{R}^{5(|\mathcal{N}(i)|+1)\times 5}$, where each row must sum to one. We collect publicly available tweets on a single topic in order to learn the state dynamics and measurement likelihood parameters.

We discretize time into eight hour intervals, and for each interval, the data consists of the number of tweets posted and the sentiment of the tweets. We use an off-the-shelf text classifier [31] to generate tweet sentiment, and the measurement model is based on the Geometric and Normal distributions,

$$\begin{aligned}&p_i(y_i^t \mid x_i^t = s)\\ &= \text{Geometric}(m_i^t; h_i)\text{Normal}(d_i^t; s\mathbf{1}_{m_i^t}, \sigma_{is}\mathbf{I}_{m_i^t})\\ &= (1-h_i)h_i^{m_i^t}(2\pi)^{-m_i^t/2}|\sigma_{is}^2\mathbf{I}_{m_i^t}|^{-1/2} \qquad (12)\\ &\qquad \exp\Big(-\frac{1}{2}(d_i^t - s\mathbf{1}_{m_i^t})^\intercal(\sigma_{is}^2\mathbf{I}_{m_i^t})^{-1}(d_i^t - s\mathbf{1}_{m_i^t})\Big),\end{aligned}$$

where $m_i^t \geq 0$ is the number of tweets posted, $h_i \in [0,1]$ is the user posting rate, and $d_i^t \in \mathbb{R}^{m_i^t}$ represents the tweet sentiments arranged as a vector. In addition, $\mathbf{I}_k$ refers to the $k \times k$ identity matrix, $\mathbf{1}_k$ refers to the k-dimensional vector of ones, and we use $\intercal$ to indicate the transpose. For intervals where a user does not post, $m_i^t = 0$, the observation model reduces to $p_i(y_i^t \mid x_i^t = s) = 1 - h_i$. We translate the user states into discrete values in order to use them as the means of the Normal distribution, $x_i^t \in \mathcal{X}_i = \{--, -, \circ, +, ++\} = \{-0.5, -0.25, 0, 0.25, 0.5\}$, as text sentiment is restricted to the interval $[-1, 1]$. In the measurement model, the unknown parameters are the posting rate and the standard deviation associated with each user state, $\{h_i \in \mathbb{R}, \sigma_{is} \in \mathbb{R} \mid s \in \mathcal{X}_i, i \in \mathcal{V}\}$.

Our data set consists of tweets on the topic of fake news from February 9th, 2017 to March 18th, 2017 [32]. We remove users with less than 50 total tweets to ensure each user has enough data to learn an accurate model of their posting behavior. We also remove tweets with a sentiment in the interval $[-0.05, 0.05]$ to eliminate tweets with an undetermined sentiment, which would incorrectly indicate that a user is neutral on a topic. Overall, the data consists of 470 users with 41,034 total tweets over $T = 112$ time intervals. The model has 10^{328} total states, and the initial belief for each user is uniform over the user states. Finally, we limit the number of neighbors for each user to at most five, and then add edges to the three users with the most tweets. We use this process to consider the effect of users seeing activity on the topic without explicitly responding or mentioning the top posters.

For our Twitter model, the unknown parameters are the dynamics model and the standard deviation and posting rate parameters in the measurement model for each user. Therefore, the model parameters for this GMDP are $\theta = \{\Lambda_i, h_i, \sigma_{is} \mid \forall i \in \mathcal{V}, s \in \mathcal{X}_i\}$. In the next section, we develop our novel model learning algorithm for GMDPs.

IV. Approximate Expectation-Maximization Learning Framework

As we discussed in Section III-A, we consider optimizing the log-likelihood of the measurements $y^{2:T}$ over the set of state dynamics and measurement model parameters θ,

$$\underset{\theta}{\text{maximize}} \;\; \log p_\theta(y^{2:T}) = \underset{\theta}{\text{maximize}} \;\; \log \sum_{x^{1:T}} p_\theta(y^{2:T}, x^{1:T}),$$

which requires marginalizing all possible state sequences for every vertex in the graph from the joint probability distribution (7). As a result, exactly computing this maximization is intractable. Instead, a lower bound on this objective can be generated by using Jensen's inequality with a distribution over state trajectories $Q(x^{1:T})$,

$$\begin{aligned}&\log \sum_{x^{1:T}} p_\theta(y^{2:T}, x^{1:T}) = \log \sum_{x^{1:T}} Q(x^{1:T}) \frac{p_\theta(y^{2:T}, x^{1:T})}{Q(x^{1:T})}\\ &\geq \sum_{x^{1:T}} Q(x^{1:T}) \log \frac{p_\theta(y^{2:T}, x^{1:T})}{Q(x^{1:T})}\\ &= \sum_{x^{1:T}} Q(x^{1:T}) \log p_\theta(y^{2:T}, x^{1:T}) - \sum_{x^{1:T}} Q(x^{1:T}) \log Q(x^{1:T})\\ &= f(Q, \theta).\end{aligned}$$

We emphasize the dependence of the joint probability on the unknown model parameters θ with the notation p_θ. The lower bound $f(Q, \theta)$ can alternatively be written,

$$f(Q, \theta) = \mathbb{E}_Q\left[\log p_\theta(y^{2:T}, x^{1:T})\right] - \mathbb{E}_Q\left[\log Q(x^{1:T})\right], \quad (13)$$

where $\mathbb{E}_Q$ indicates the expectation taken with respect to the distribution $Q(x^{1:T})$. The Expectation-Maximization algorithm (EM) [24] performs coordinate ascent in $f(Q, \theta)$,

$$\begin{aligned}&\text{Expectation (E-step): } Q^k(x^{1:T}) \leftarrow \arg\max_Q f(Q, \theta^k)\\ &\text{Maximization (M-step): } \theta^{k+1} \leftarrow \arg\max_\theta f(Q^k, \theta)\end{aligned}$$

We use superscripts to denote estimates of the distribution Q and the parameters θ computed during iterations of the EM algorithm, e.g., θ^k refers to the k-th estimate of the model parameters. In the standard EM algorithm, the E-step has a closed-form solution. The maximum for the E-step objective $f(Q, \theta^k)$ is achieved by choosing the conditional distribution of $x^{1:T}$, $Q(x^{1:T}) = p_{\theta^k}(x^{1:T} \mid y^{2:T})$, at which point the lower bound becomes an equality for a given θ^k. Then, for the M-step, the parameter update is the well-known result,

$$\theta^{k+1} \leftarrow \arg\max_{\theta} \sum_{x^{1:T}} p_{\theta^k}(x^{1:T} \mid y^{2:T}) \log p_\theta(y^{2:T}, x^{1:T}).$$

For the models we consider in this work, the standard approach is computationally intractable as it requires computing the conditional distribution $p_{\theta^k}(x^{1:T} \mid y^{2:T})$ for the E-step, as well as marginalizations of this distribution for the M-step. For example, in our Twitter model, this distribution requires specifying on the order of 10^{36000} values. Our insight is that a mean-field approximation in the E-step results in an efficient algorithm that still incorporates the key properties and structure of GMDPs. Using this approximation, the resulting coordinate-ascent algorithm indirectly optimizes the log-likelihood of the data, and the inequality is no longer tight in the E-step. We show in our results in Section V that this approach scales to significantly large GMDPs and learns meaningful models.

A. Approximate Expectation Step (E-Step)

We begin by developing a tractable E-step which is scalable to GMDP models with arbitrary coupling interactions. Given an estimate of the model parameters θ^k, we approximate the true distribution over state trajectories $p(x^{1:T})$ by a mean-field approximation $Q(x^{1:T})$ which considers each MDP $i \in \mathcal{V}$ as an independent process given the observations,

$$Q(x^{1:T}) = \frac{1}{Z_Q} \prod_{i \in \mathcal{V}} p_i(x_i^1) \prod_{t=2}^{T} Q_i(x_i^t \mid x_i^{t-1}) p_i(y_i^t \mid x_i^t), \quad (14)$$

where Z_Q is a normalization constant,

$$Z_Q = \sum_{x^{1:T}} \prod_{i \in \mathcal{V}} p_i(x_i^1) \prod_{t=2}^{T} Q_i(x_i^t \mid x_i^{t-1}) p_i(y_i^t \mid x_i^t).$$

We specify Q_i as a time-varying matrix $\phi_i^t \in \mathbb{R}^{|\mathcal{X}_i| \times |\mathcal{X}_i|}$,

$$Q_i(x_i^t \mid x_i^{t-1}) = \left[\phi_i^t\right]_{x_i^{t-1}, x_i^t},$$

and we refer to ϕ_i^t as a "time-varying bias." By inspection of the joint probability (7), the bias term for each MDP $i \in \mathcal{V}$ will need to incorporate information from the MDP's neighbors $j \in \mathcal{N}(i)$ (i.e, the MDP's Markov blanket), to represent the MDP as an independent process given the observations. We emphasize here that the mean-field form (14) of $Q(x^{1:T})$ is the key approximation in the EM framework, which allows us to derive an efficient E-step and M-step that scale to large GMDPs. We show at the end of this section that exploiting Anonymous Influence further improves the computational complexity, although our algorithm can still be used without this property.

The parameters which define the approximate distribution $Q(x^{1:T})$ are $\psi = \{\phi_i^t \mid i \in \mathcal{V}, t \in [2,T]\}$. Substituting this form of $Q(x^{1:T})$ into the lower bound (13) yields,

$$\begin{aligned} f(Q,\theta^k) &= \log Z_Q - \mathbb{E}_Q\left[\sum_{i\in\mathcal{V}} \log p_i(x_i^1)\right] \\ &+ \mathbb{E}_Q\left[\sum_{i\in\mathcal{V}} \log p_i(x_i^1) + \sum_{i\in\mathcal{V}}\sum_{t=2}^{T} \log p_i(y_i^t \mid x_i^t)\right] \\ &+ \mathbb{E}_Q\left[\sum_{i\in\mathcal{V}}\sum_{t=2}^{T} \log p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1})\right] \\ &- \mathbb{E}_Q\left[\sum_{i\in\mathcal{V}}\sum_{t=2}^{T} \log Q_i(x_i^t \mid x_i^{t-1}) + \sum_{i\in\mathcal{V}}\sum_{t=2}^{T} \log p_i(y_i^t \mid x_i^t)\right] \end{aligned}$$

where $\mathbb{E}_Q$ refers to expectation taken with respect to the approximate distribution $Q(x^{1:T})$. Notably, several terms cancel in the above expression, due to our choice of approximation. As a result, the E-step objective can be further simplified to,

$$\begin{aligned} f(Q,\theta^k) &= \log Z_Q \\ &+ \sum_{i\in\mathcal{V}}\sum_{t=2}^{T} \mathbb{E}_Q\left[\log p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}) - \log Q_i(x_i^t \mid x_i^{t-1})\right], \end{aligned} \quad (15)$$

where we have also moved the summations outside the expectation. Given this E-step objective, the goal is to derive a tractable update expression of the distribution parameters ψ given an estimate of the model parameters θ^k. We begin by working out the expectations in order to understand the dependence of the objective on the distribution parameters. The first expectation is,

$$\begin{aligned} \mathbb{E}_Q\left[\log p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1})\right] &= \sum_{x_i^{t-1}} \sum_{x_i^t} p_\psi(x_i^{t-1}, x_i^t \mid y_i^{2:T}) \\ \sum_{x_{\mathcal{N}(i)}^{t-1}} \Big(\prod_{j\in\mathcal{N}(i)} & p_\psi(x_j^{t-1} \mid y_j^{2:T})\Big) \log\left[\Lambda_i\right]_{x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, x_i^t}. \end{aligned}$$

For each MDP $i \in \mathcal{V}$, computing this expectation requires the state occupation probabilities given the measurements, $p_\psi(x_i^{t-1}, x_i^t \mid y_i^{2:T})$ and $p_\psi(x_j^{t-1} \mid y_j^{2:T})$, and the MDP dynamics model Λ_i. Note that an estimate of the model parameters θ^k, including Λ_i, is given for the E-step. We use the notation p_ψ to emphasize that these distributions are based on the time-varying biases $\psi = \{\phi_i^t \mid i \in \mathcal{V}, t \in [2,T]\}$ which define the approximate distribution, $Q(x^{1:T})$.

The second expectation works out to,

$$\begin{aligned} &\mathbb{E}_Q\left[\log Q_i(x_i^t \mid x_i^{t-1})\right] \\ &\quad = \sum_{x_i^{t-1}} \sum_{x_i^t} p_\psi(x_i^{t-1}, x_i^t \mid y_i^{2:T}) \log\left[\phi_i^t\right]_{x_i^{t-1}, x_i^t}, \end{aligned}$$

which directly depends on the time-varying bias as well as the state occupation probabilities. Substituting these expressions into (15) results in,

$$\begin{aligned} f(Q,\theta^k) &= \log Z_Q \\ &+ \sum_{i\in\mathcal{V}}\sum_{t=2}^{T}\sum_{x_i^{t-1}}\sum_{x_i^t} p_\psi(x_i^{t-1}, x_i^t \mid y_i^{2:T})\Big(-\log\big[\phi_i^t\big]_{x_i^{t-1},x_i^t} \\ &+ \sum_{x_{\mathcal{N}(i)}^{t-1}} \Big(\prod_{j\in\mathcal{N}(i)} p_\psi(x_j^{t-1}\mid y_j^{2:T})\Big)\log\,[\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t}\Big). \end{aligned}$$

We now optimize $f(Q,\theta^k)$ over the approximate model for MDP i and for a single time step and state transition, i.e., the parameter $[\phi_i^t]_{x_i^{t-1},x_i^t}$. Once we develop an update rule for this specific parameter, we can apply this result to other time steps and transitions for MDP i, and to other MDPs in the GMDP. Therefore, we take the derivative of $f(Q,\theta^k)$ with respect to $\log\,[\phi_i^t]_{x_i^{t-1},x_i^t}$ to determine the maximum likelihood estimate,

$$\begin{aligned} \frac{\partial f(Q,\theta^k)}{\partial \log\,[\phi_i^t]_{x_i^{t-1},x_i^t}} &= \frac{\partial \log Z_Q}{\partial \log\,[\phi_i^t]_{x_i^{t-1},x_i^t}} - p_\psi(x_i^{t-1}, x_i^t \mid y_i^{2:T}) \\ &+ \frac{\partial p_\psi(x_i^{t-1}, x_i^t \mid y_i^{2:T})}{\partial \log\,[\phi_i^t]_{x_i^{t-1},x_i^t}}\Big(-\log\big[\phi_i^t\big]_{x_i^{t-1},x_i^t} \\ &+ \sum_{x_{\mathcal{N}(i)}^{t-1}} \Big(\prod_{j\in\mathcal{N}(i)} p_\psi(x_j^{t-1}\mid y_j^{2:T})\Big)\log\,[\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t}\Big). \end{aligned} \tag{16}$$

By using the fact that,

$$\frac{\partial \log Z_Q}{\partial \log\,[\phi_i^t]_{x_i^{t-1},x_i^t}} = p_\psi(x_i^{t-1}, x_i^t \mid y_i^{2:T}),$$

we can further simplify (16),

$$\begin{aligned} \frac{\partial f(Q,\theta^k)}{\partial \log\,[\phi_i^t]_{x_i^{t-1},x_i^t}} &= \frac{\partial p_\psi(x_i^{t-1}, x_i^t \mid y_i^{2:T})}{\partial \log\,[\phi_i^t]_{x_i^{t-1},x_i^t}}\Bigg[-\log\big[\phi_i^t\big]_{x_i^{t-1},x_i^t} \\ &+ \sum_{x_{\mathcal{N}(i)}^{t-1}} \Big(\prod_{j\in\mathcal{N}(i)} p_\psi(x_j^{t-1}\mid y_j^{2:T})\Big)\log\,[\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t}\Bigg]. \end{aligned}$$

The derivative $\partial f(Q,\theta^k)/\partial \log\,[\phi_i^t]_{x_i^{t-1},x_i^t}$ is zero when the expression in brackets is zero, which leads to the following fixed-point equation,

$$\begin{aligned} &\log\big[\phi_i^t\big]_{x_i^{t-1},x_i^t} \\ &= \sum_{x_{\mathcal{N}(i)}^{t-1}} \Big(\prod_{j\in\mathcal{N}(i)} p_\psi(x_j^{t-1}\mid y_j^{2:T})\Big)\log\,[\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t}. \end{aligned} \tag{17}$$

The result is a fixed-point equation because updating the bias ϕ_i^t for MDP i requires other unknown distribution parameters, $\{\phi_j^t \mid j\in\mathcal{N}(i)\}$, to compute the state occupation distributions $p_\psi(x_j^{t-1} \mid y_j^{2:T})$. Furthermore, this relationship shows the dependence of MDP i's time-varying bias on information from its neighbors, i.e., its Markov blanket.

The relationship (17) can be applied for any MDP $i\in\mathcal{V}$, for any time $t\in[2,T]$, and for any transition $x_i^{t-1},x_i^t\in\mathcal{X}_i$. Therefore, solving the fixed-point equation requires iterating between the following two steps until convergence,

(1) Compute the distributions $p_\psi(x_i^{t-1},x_i^t \mid y_i^{2:T})$ $\forall i\in\mathcal{V}, t\in[2,T]$ given $\{\phi_i^t \mid i\in\mathcal{V}, t\in[2,T]\}$;

(2) Update the time-varying bias ϕ_i^t $\forall i\in\mathcal{V}, t\in[2,T]$ given $\{p_\psi(x_i^{t-1},x_i^t \mid y_i^{2:T}) \mid i\in\mathcal{V}, t\in[2,T]\}$.

Remark. Our approximate E-step is similar in style to a message-passing scheme, where information spreads around the entire graph by repeatedly passing local information between vertices. Since every MDP needs the distributions $p_\psi(x_j^{t-1} \mid y_j^{2:T})$ for its neighbors, the messages shared between MDPs is the time-varying bias ϕ_i^t. As a result, the global graph topology affects the rate at which the distribution parameters $\psi=\{\phi_i^t \mid i\in\mathcal{V}, t\in[2,T]\}$ will converge. For our COVID-19 and Twitter models, our experiments show that the approximate E-step converges relatively quickly, in about 10 iterations. We plan to learn the graph structure, and provide theoretical insight and analysis regarding the effect of the graph on convergence, in future work.

Exploiting Anonymous Influence. Using (17) to compute ϕ_i^t for a single MDP i requires enumerating $T|\mathcal{X}_i|^2\prod_{j\in\mathcal{N}(i)}|\mathcal{X}_j|$ values, which may be intractable due to the number of observations T, the size of the neighbor set $|\mathcal{N}(i)|$, or the state space of the MDPs $|\mathcal{X}_i|$. Therefore, we now exploit Anonymous Influence to address this potential issue. By instead considering whether or not the neighboring MDPs $j\in\mathcal{N}(i)$ are in an influencing state, the fixed-point equation becomes,

$$\log\big[\phi_i^t\big]_{x_i^{t-1},x_i^t} = \sum_{b_i^{t-1}} p_\psi\big(b_i^{t-1} \mid y_{\mathcal{N}(i)}^{2:T}\big)\log\,[\Lambda_i]_{x_i^{t-1},b_i^{t-1},x_i^t}. \tag{18}$$

In the above expression, we make use of a count aggregator (CA) $b_i^{t-1}\in[0,1,\dots,|\mathcal{N}(i)|]$ to represent the influence of the MDPs $j\in\mathcal{N}(i)$ which has significantly lower computational cost. In Sections III-C and III-D, we presented two GMDP models where the MDP dynamics were influenced by a neighbor-based quantity b_i^t and exploiting this structure allows us to improve the computational complexity of our approach. We also note that the distribution $p_\psi\big(b_i^{t-1} \mid y_{\mathcal{N}(i)}^{2:T}\big)$ is computed from the distributions $\{p_\psi(x_j^t \mid y_j^{2:T}) \mid j\in\mathcal{N}(i)\}$.

Our approximate E-step requires iterating between computing distributions and updating the time-varying biases, as we previously discussed. The first step can be efficiently computed by using the Forward-Backward algorithm [23] with the model assumption in (14) which has time complexity $\mathcal{O}(T|\mathcal{X}_i|^2)$ for each vertex. The second step has time complexity $\mathcal{O}(T|\mathcal{X}_i|^2\prod_{j\in\mathcal{N}(i)}|\mathcal{X}_j|)$, and with Anonymous Influence, this can be reduced to $\mathcal{O}(T|\mathcal{X}_i|^2|\mathcal{N}(i)|)$ for each vertex. We note that both steps can easily be parallelized and we take advantage of this in our experiments in Section V. We alternate between these two steps until successive estimates of the time-varying biases converge, and the result of the E-step are the parameters ψ^k representing the distribution $Q(x^{1:T})$. With this distribution in hand, we can turn to the M-step to compute an updated approximation of the model parameters θ, as we discuss next.

B. Maximization Step (M-Step)

The result of the E-step is an an estimate of the approximate distribution $Q(x^{1:T})$, which we denote as Q^k and ψ^k in the following discussion. Given this estimate, the objective for the Maximization step (M-step) is,

$$f(Q^k,\theta) = \mathbb{E}_{Q^k}\left[\log p(y^{2:T},x^{1:T})\right] = \underbrace{\sum_{i\in\mathcal{V}} \mathbb{E}_{Q^k}\left[\log p_i(x_i^1)\right]}_{\text{first term: prior}}$$
$$+ \underbrace{\sum_{i\in\mathcal{V}}\sum_{t=2}^{T} \mathbb{E}_{Q^k}\left[\log p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1})\right]}_{\text{second term: dynamics}}$$
$$+ \underbrace{\sum_{i\in\mathcal{V}}\sum_{t=2}^{T} \mathbb{E}_{Q^k}\left[\log p_i(y_i^t \mid x_i^t)\right]}_{\text{third term: measurement model}}. \tag{19}$$

We dropped the entropy of Q^k in the above expression as it is constant with respect to θ once the distribution is specified. Similar to our approximate E-step, we work through the expectations in order to understand the dependence of the objective on the model parameters θ, and to derive update expressions. The first term of (19) is constant as the prior is assumed to be known and therefore can be dropped from the objective as well. Expanding the second term of (19),

$$\sum_{i\in\mathcal{V}}\sum_{t=2}^{T} \mathbb{E}_{Q^k}\left[\log p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1})\right]$$
$$= \sum_{i\in\mathcal{V}}\sum_{t=2}^{T}\sum_{x_i^{t-1}}\sum_{x_{\mathcal{N}(i)}^{t-1}}\sum_{x_i^t}\Big(p_{\psi^k}(x_i^{t-1}, x_i^t \mid y_i^{2:T}) \tag{20}$$
$$\prod_{j\in\mathcal{N}(i)} p_{\psi^k}(x_j^{t-1} \mid y_j^{2:T})\Big) \log\,[\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t}\,.$$

Note that only the second term of (19) depends on the dynamics Λ_i of the MDPs. We now optimize (20) over a single state transition of MDP i, $[\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t}$, to develop an update rule. We can then extend this rule to other parameters of MDP i, as well as other MDPs in the GMDP. For the dynamics, it is necessary to include the constraint,

$$\sum_{x_i^t} [\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t} = 1 \;\forall x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1},$$

to ensure the dynamics represent a valid transition distribution. Therefore, to determine the maximum likelihood estimate, we take the derivative of (20 and the distribution constraint with respect to $[\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t}$,

$$\frac{\partial}{\partial\,[\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t}}\left[(20) + \lambda_i\Big(\sum_{x_i^t}[\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t} - 1\Big)\right] = 0,$$

where λ_i is the Lagrange multiplier associated with the constraint. Solving the above expression for the parameter results in the update,

$$[\Lambda_i]_{x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},x_i^t} =$$
$$\frac{\sum_{t=2}^{T} p_{\psi^k}(x_i^{t-1}, x_i^t \mid y_i^{2:T}) \prod_{j\in\mathcal{N}(i)} p_{\psi^k}(x_j^{t-1} \mid y_j^{2:T})}{\sum_{t=2}^{T} p_{\psi^k}(x_i^{t-1} \mid y_i^{2:T}) \prod_{j\in\mathcal{N}(i)} p_{\psi^k}(x_j^{t-1} \mid y_j^{2:T})}. \tag{21}$$

Using this update rule, we can update the dynamics models $\{\Lambda_i \mid i \in \mathcal{V}\}$ given an estimate of the parameters ψ^k from the approximate E-step.

Exploiting Anonymous Influence. The computational cost of updating the dynamics models is reduced by taking advantage of count aggregators (CAs),

$$[\Lambda_i]_{x_i^{t-1},b_i^{t-1},x_i^t}$$
$$= \frac{\sum_{t=2}^{T} p_{\psi^k}(x_i^{t-1}, x_i^t \mid y_i^{2:T}) p_{\psi^k}\big(b_i^{t-1} \mid y_{\mathcal{N}(i)}^{2:T})\big)}{\sum_{t=2}^{T} p_{\psi^k}(x_i^{t-1} \mid y_i^{2:T}) p_{\psi^k}\big(b_i^{t-1} \mid y_{\mathcal{N}(i)}^{2:T}\big)}. \tag{22}$$

The CA $b_i^{t-1} \in [0,1,\dots,|\mathcal{N}(i)|]$ summarizes the influence from the neighbors of MDP i. See Sections III-C and III-D for our discussion on count aggregators for the two applications we consider in this work.

Lastly, working out the third term of (19) leads to,

$$\sum_{i\in\mathcal{V}}\sum_{t=2}^{T} \mathbb{E}_{Q^k}\left[\log p_i(y_i^t \mid x_i^t)\right]$$
$$= \sum_{i\in\mathcal{V}}\sum_{t=2}^{T}\sum_{x_i^t} p_{\psi^k}(x_i^t \mid y_i^{2:T}) \log p_i(y_i^t \mid x_i^t),$$

and the resulting update for the parameters of $p_i(y_i^t \mid x_i^t)$ depends on the choice of observation model. We derive the update expressions for the measurement models for our case studies in the following sections.

C. Case Study: COVID-19 Pandemic in California

Given the model we introduced in Section III-C for the 2020 COVID-19 pandemic in California, the third term in (19) is,

$$\sum_{i\in\mathcal{V}}\sum_{t=2}^{T} \mathbb{E}_{Q^k}\left[\log p(y_i^t \mid x_i^t)\right]$$
$$= \sum_{i\in\mathcal{V}}\sum_{t=2}^{T}\sum_{s\in\mathcal{X}_i} p_{\psi^k}(x_i^t = s \mid y_i^{2:T})\Big(\log\sigma_{is}^{-1}$$
$$+ \log(2\pi)^{-1/2} - \frac{1}{2}(y_i^t - s)^2\sigma_{is}^{-2}\Big).$$

To compute the maximum likelihood estimate of the covariance σ_{is}^2, we take the derivative of this expression with respect to σ_{is}^{-1} and set it equal to zero,

$$\sum_{t=2}^{T} p_{\psi^k}(x_i^t = s \mid y_i^{2:T})\Big(\sigma_{is} - (y_i^t - s)^2\sigma_{is}^{-1}\Big) = 0.$$

The resulting update for the covariance is,

$$\sigma_{is}^2 = \frac{\sum_{t=2}^{T} p_{\psi^k}(x_i^t = s \mid y_i^{2:T})(y_i^t - s)^2}{\sum_{t=2}^{T} p_{\psi^k}(x_i^t = s \mid y_i^{2:T})}. \tag{23}$$

Therefore, for each county i in our COVID-19 model, the dynamics parameters are updated using (22) and the covariance parameters $\{\sigma_{is} \mid s \in \mathcal{X}_i\}$ are updated using (23).

D. Case Study: User Interactions on Twitter

For the Twitter user interaction model we introduced in Section III-D, the third term in (19) works out to,

$$\begin{aligned}\sum_{i\in\mathcal{V}}\sum_{t=2}^{T}&\mathbb{E}_{Q^k}\left[\log p_i(y_i^t\mid x_i^t)\right]\\ =\sum_{i\in\mathcal{V}}\sum_{t=2}^{T}\sum_{s\in\mathcal{X}_i}& p_{\psi^k}(x_i^t=s\mid y_i^{2:T})\Big(\log(1-h_i)\\ &+m_i^t\log h_i+\log(2\pi)^{-m_i^t/2}+\log|\sigma_{is}^2\mathbf{I}_{m_i^t}|^{-1/2}\\ &-\frac{1}{2}(d_i^t-s\mathbf{1}_{m_i^t})^\intercal(\sigma_{is}^2\mathbf{I}_{m_i^t})^{-1}(d_i^t-s\mathbf{1}_{m_i^t})\Big),\end{aligned}\tag{24}$$

We take derivatives of (24) with respect to the different measurement model parameters in order to derive the maximum likelihood estimates. Taking the derivative with respect to the posting rate h_i and setting it equal to zero gives,

$$-\frac{T-1}{1-h_i}+\frac{1}{h_i}\sum_{t=2}^{T}m_i^t=0,$$

which results in the closed-form solution,

$$h_i=\frac{\frac{1}{T-1}\sum_{t=2}^{T}m_i^t}{1+\frac{1}{T-1}\sum_{t=2}^{T}m_i^t}.\tag{25}$$

Taking the derivative of (24) with respect to σ_{is}^{-1} and setting it equal to zero yields,

$$\begin{aligned}\sum_{t=2}^{T}p_{\psi^k}(x_i^t=s\,|y_k^{2:T})\Big(&m_i^t\sigma_{is}\\ &-(d_i^t-s\mathbf{1}_{m_i^t})^\intercal(d_i^t-s\mathbf{1}_{m_i^t})\sigma_{is}^{-1}\Big)=0.\end{aligned}$$

and thus the update expression is,

$$\sigma_{is}^2=\frac{\sum_{t=2}^{T}p_{\psi^k}(x_i^t=s\mid y_i^{2:T})(d_i^t-s\mathbf{1}_{m_i^t})^\intercal(d_i^t-s\mathbf{1}_{m_i^t})}{\sum_{t=2}^{T}p_{\psi^k}(x_i^t=s\mid y_i^{2:T})m_i^t}.\tag{26}$$

For each user i in our Twitter model, the dynamics parameters are updated using (22). For the measurement model, the posting rate h_i has a closed-form solution (25) and is computed before running our model learning algorithm. The covariance parameters $\{\sigma_{is}\mid s\in\mathcal{X}_i\}$ are updated using (26).

E. Algorithm Summary

Algorithm 1 summarizes our approximate EM approach. The E-step iterates on the bias terms until convergence (lines 3 to 5), and we monitor convergence using the absolute difference between successive iterations. Our experiments show that this step typically converges in 10 iterations or less. Given an estimate of the biases, the M-step (line 6) updates the dynamics and measurement model parameters of all the MDPs in the GMDP. We monitor convergence of these parameters using successive values of the log-likelihood of the data, which we discuss further in our experiments in Section V. Our experiments show that overall convergence is typically achieved in 30 to 50 parameter update iterations. The computational complexity of our algorithm is dominated by the E-step, with cost $\mathcal{O}(T|\mathcal{X}_i|^2\prod_{j\in\mathcal{N}(i)}|\mathcal{X}_j|)$ per vertex. Using Anonymous Influence, this cost can be reduced to $\mathcal{O}(T|\mathcal{X}_i|^2|\mathcal{N}(i)|)$ per vertex. In the next section, we present performance metrics and results for our two case studies.

Algorithm 1 Model Learning Algorithm

```
1: Initialize model parameters
2: while parameters have not converged do
3:     while time-varying bias has not converged do
4:         Compute distributions $p_\psi(x_i^{t-1}, x_i^t \mid y_i^{2:T})$
5:         Update time-varying bias, (17) or (18)
6:     Update dynamics and measurement model parameters
```

V. Experiments and Results

We now demonstrate the use of our approximate EM algorithm to learn meaningful GMDP models. We use publicly available datasets for our COVID-19 [30] and Twitter [32] models, and apply Algorithm 1. For both models, we initialize the dynamics parameters as the uniform distribution, and the initial covariance values are one. In addition, for the E-step of both models, the Forward-Backward algorithm is used to compute the distributions $p_\psi(x_i^{t-1}, x_i^t\mid y_i^{2:T})$, and the time-varying biases are updating using (18). For the M-step of the COVID-19 model, the dynamics and measurement model parameters are updated using (22) and (23). For the M-step of the Twitter model, the dynamics and measurement model parameters are updated using (22), (25), and (26).

To evaluate the performance of our algorithm, we consider a comparison model which uses a complete independence assumption, where each MDP in the GMDP has a transition matrix $\Lambda_i^{\text{nMDP}}\in\mathbb{R}^{|\mathcal{X}_i|\times|\mathcal{X}_i|}$ with no coupling interactions with other MDPs. The same parameterization for the measurement model $p_i(y_i^t\mid x_i^t)$ is used. We refer to this model as "nMDP," to emphasize the uncoupled assumption. The standard Baum-Welch algorithm is used to learn the parameters for each MDP in the nMDP model. Next, we present performance metrics to evaluate our learned GMDP models.

A. Performance Metrics

Data Likelihood. We start by comparing the objective value of the learning algorithms for each model assumption after reaching convergence, which is a common metric when learning discrete Markov models. In particular, this corresponds to the log-likelihood of the data under the different model assumptions. We emphasize here that computing the log-likelihood of the data, $\log p_\theta(y^{2:T})$, is intractable for GMDPs as it requires enumerating all possible state trajectories $x^{1:T}\in\prod_{i\in\mathcal{V}}|\mathcal{X}_i|^T$. Therefore, we use the log-likelihood computed from the approximate distribution $Q(x^{1:T})$ instead, which provides an under-approximation of the log-likelihood.

Coupling Strength. We develop additional metrics, based on the Kullback-Leibler (KL) divergence, to evaluate the coupling interactions between MDPs in the learned GMDPs. First, we consider the "coupling strength," which is defined as,

$$f_i^{\text{coupling}} = \sum_{x_i^{t-1}} \sum_{x_{\mathcal{N}(i)}^{t-1}} \sum_{x_i^t} [\Lambda_i]_{x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, x_i^t} \Big(\log [\Lambda_i]_{x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, x_i^t} - \log \big[\Lambda_i^{\text{nMDP}}\big]_{x_i^{t-1}, x_i^t} \Big),$$

and which uses the learned dynamics from the baseline nMDP model. This metric is computed for each MDP in the GMDP, and we present the values $\{f_i^{\text{coupling}} \mid i \in \mathcal{V}\}$ as a box plot for the COVID-19 and Twitter models. If this metric is zero for many MDPs, then the learned GMDP does not represent significant coupling interactions between MDPs, and the independence assumption is more appropriate for the data.

Influence Strength. Second, we consider the "influence strength," which is defined as,

$$f_i^{\text{influence}} = \sum_{x_i^{t-1}} \sum_{x_{\mathcal{N}(i)}^{t-1}} \sum_{x_i^t} [\Lambda_i]_{x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, x_i^t} \bigg(\log [\Lambda_i]_{x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, x_i^t} - \log \frac{\sum_{x_{\mathcal{N}(i)}^{t-1}} [\Lambda_i]_{x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, x_i^t}}{\prod_{j \in \mathcal{N}(i)} |\mathcal{X}_j|} \bigg),$$

and which averages the learned GMDP dynamics model over configurations of neighbor states for a given MDP. We use this average distribution as a comparison to determine if the influence of neighboring MDPs is not significant, i.e., the dynamics of an MDP do not change based on the neighbor MDP states. We again use a box plot to visualize the range of values $\{f_i^{\text{influence}} \mid i \in \mathcal{V}\}$. If this metric is zero for many MDPs, then the Anonymous Influence property is not appropriate for the process.

B. COVID-19 Pandemic in California

We present the objective value of the learning algorithm for the GMDP and nMDP models in Fig. 3, the coupling metric in Fig. 4, and the influence metric in Fig. 5. First, the negative log-likelihood plots show that the GMDP model assumption has a significantly lower value at convergence, which corresponds to a higher likelihood of observing the data. Furthermore, the coupling strength f_i^{coupling} and the influence strength $f_i^{\text{influence}}$ are non-zero for a significant fraction of counties in California. The spread of values in both metric shows that our GMDP model is able to learn a combination of coupled and uncoupled models for different counties, in order to best fit the observed data. An uncoupled model may be more descriptive based on the measures taken by a county, such as mask mandates, limiting gathering sizes, and requiring quarantines for incoming travelers, which limits the influence of neighboring counties on the daily case count. At this point, we can investigate the learned models further to understand what measures are particularly effective in reducing the infection rate.

C. User Interactions on Twitter

We present the objective value of the learning algorithm for the GMDP and nMDP models in Fig. 6, the coupling metric in Fig. 7, and the influence metric in Fig. 8. For the log-likelihood, both the GMDP and the nMDP models approximately achieve the same value as the trajectories are overlaid in the figure. However, the objective value can be misleading, as we are using an approximation to the true log-likelihood of the data under the GMDP model assumption, which is intractable to directly compute. Therefore, we have introduced additional metrics to provide further insight and analysis for the learned models. In particular, the coupling and influence metrics show that the coupled interaction model provided by the GMDP better explains the observed data than the independent model assumption for a number of users. On the other hand, some users are simply posting due to the aggregate activity on the topic, and not because of particular interactions with other users.

At this point, it is possible to investigate which users drive posting activity and user interactions on a given topic, compared to other users who are expressing their opinion without explicit interactions. For this data set, we note that it is difficult

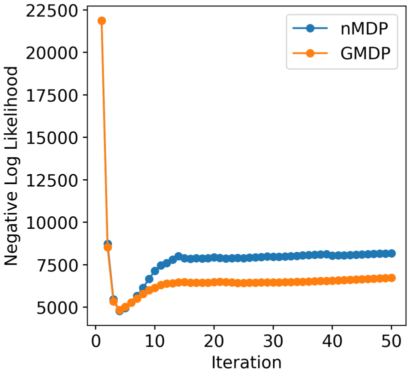


Fig. 3. Comparison of the objective values achieved by the GMDP and NMDP models, on the COVID-19 data set. At convergence, the GMDP model negative log-likelihood is 6732.25 whereas the nMDP model negative log-likelihood is 8176.01. Therefore, the GMDP model is a significant improvement over the uncoupled assumption in the nMDP model. Additional insight is provided by our other metrics, which we present in Figs. 4 and 5.

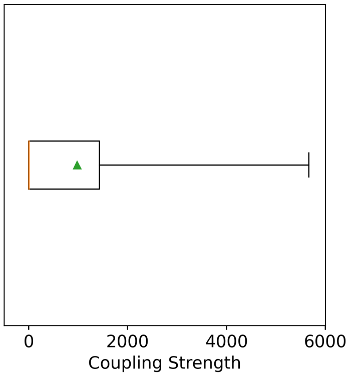


Fig. 4. Box plot of the coupling strength values $\{f_i^{\text{coupling}} \mid i \in \mathcal{V}\}$ for the COVID-19 data set. The minimum is 0.00, the mean (green triangle) is 980.72, and the maximum is 5666.00. The orange line is the median and the caps refer to the minimum and maximum. From this metric, virus transmission between counties is a more significant effect for nearly all counties in California, compared to intra-county spread. By explicitly considering this aspect in our GMDP model, we are able to better explain the observed data compared to a completely uncoupled model assumption.

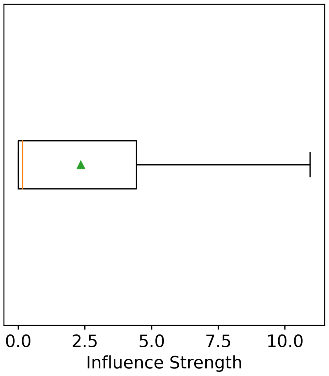


Fig. 5. Box plot of the influence strength values $\{f_i^{\text{influence}} \mid i \in \mathcal{V}\}$ for the COVID-19 data set. The minimum is 0.00, the mean (green triangle) is 2.35, and the maximum is 10.94. The orange line is the median and the caps refer to the minimum and maximum. This metric shows that it is important to model the influence as state-dependent to describe how the virus spreads in California. At this point, we can investigate the actions taken by different counties to understand how to best limit the spread.

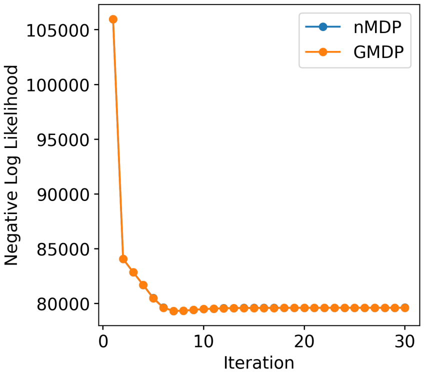


Fig. 6. Comparison of the objective values achieved by the GMDP and nMDP models, on the Twitter data set. Both models achieve approximately the same value at convergence, 79591.79 for the GMDP model and 79607.84 for the nMDP model, and the trajectories are overlaid in the plot. We emphasize that the log-likelihood can be a misleading metric, as it is intractable to directly compute for the large models we consider in this work, and instead we must approximate it. Therefore, we developed additional metrics to gain more insight into our models, which we present in Figs. 7 and 8.

to determine sentiment from tweets [33], due to the 140 character limit and the frequent use of abbreviations, slang, and emojis. However, there are no easily available off-the-shelf text sentiment classifiers specifically for tweets, which limits the quality of the data given to the learning algorithms, and accurately determining tweet sentiment remains an open research question. We believe that improving the sentiment analysis will directly benefit our framework, and we intend to investigate this aspect further in future work.

VI. Conclusions

In this work, we developed a model learning algorithm appropriate for GMDPs with significantly large state spaces and arbitrary coupling structure. We leverage the Expectation-Maximization framework to derive our approach, and use real-world data to learn insightful models which validate our assumptions. We have considered a relatively simple formulation of GMDPs in this work, given the complexity of deriving a model learning algorithm. There are a number avenues for future work, starting with structure learning of the graph. As part of this, we plan to investigate directed graphs, sparsity patterns, and analysis techniques for the graph topology, such as convergence rate and solution quality. With regards to Anonymous Influence framework, we are investigating variations of the models in this work, such as unique state spaces and count aggregators for each MDP. Furthermore, some domains may benefit from a hybrid model, where the identity of some neighbors is a significant effect. Finally, we aim to develop a generalized action model, suitable for a variety of domains, along with algorithms to estimate the sequence of actions from data.

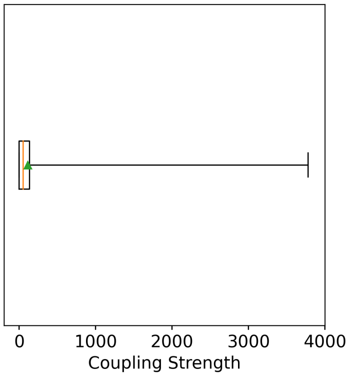


Fig. 7. Box plot of the coupling strength values $\{f_i^{\text{coupling}} \mid i \in \mathcal{V}\}$ for the Twitter data set. The minimum is 0.00, the mean (green triangle) is 111.79, and the maximum is 3779.15. The orange line is the median and the caps refer to the minimum and maximum. This metric shows the learned GMDP consists of a mix of coupled and uncoupled models to best describe the data. Therefore, a segment of users are posting to express their opinion without any explicit interaction, and another segment of users are explicitly interacting on the topic. Our GMDP can accommodate both of these effects compared to other modeling frameworks. We note that it is difficult to extract sentiment from tweets which limits the insights into the learned models.

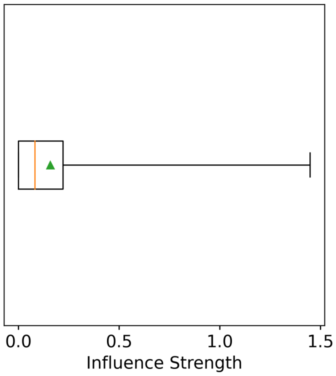


Fig. 8. Box plot of the influence strength values $\{f_i^{\text{influence}} \mid i \in \mathcal{V}\}$ for the Twitter data set. The minimum is 0.00, the mean (green triangle) is 0.16, and the maximum is 1.45. The orange line is the median, and the caps refer to the minimum and maximum. For the majority of users, posting behavior is driven by bulk activity on the topic, with a skew towards users agreeing with each other when posting. We can then investigate particular users and understand how they drive the overall activity and engagement on the topic.

References

[1] A. Pentland, N. Oliver, and M. Brand, “Coupled hidden Markov models for complex action recognition,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 994–999.

[2] S. M. Chu and T. S. Huang, “Bimodal speech recognition using coupled hidden Markov models,” in *ICSLP-2000*, vol. 2, 2000, pp. 747–750.

[3] S. Zhong and J. Ghosh, “A new formulation of coupled hidden Markov models,” Tech. Rep., 2001.

[4] I. Rezek and S. J. Roberts, “Estimation of coupled hidden Markov models with application to biosignal interaction modelling,” in *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No.00TH8501)*, vol. 2, 2000, pp. 804–813.

[5] I. Rezek, P. Sykacek, and S. J. Roberts, “Learning interaction dynamics with coupled hidden Markov models,” *IEE Proceedings - Science, Measurement and Technology*, vol. 147, no. 6, pp. 345–350, 2000.

[6] R. N. Haksar and M. Schwager, “Controlling large, graph-based MDPs with global control capacity constraints: An approximate LP solution,” in *57th IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 35–42.

[7] W. Dong, A. S. Pentland, and K. A. Heller, “Graph-coupled HMMs for modeling the spread of infection,” in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012, pp. 227–236.

[8] J. Kwon and K. Murphy, “Modeling freeway traffic with coupled HMMs,” Tech. Rep., 2000.

[9] Z. Ghahramani and M. I. Jordan, “Factorial hidden Markov models,” in *Advances in Neural Information Processing Systems 8*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds. MIT Press, 1996, pp. 472–478.

[10] M. I. Jordan, Z. Ghahramani, and L. K. Saul, “Hidden Markov decision trees,” in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. MIT Press, 1997, pp. 501–507.

[11] R. D. Malmgren, J. M. Hofman, L. A. Amaral, and D. J. Watts, “Characterizing individual communication patterns,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 607–616.

[12] V. Raghavan, G. Ver Steeg, A. Galstyan, and A. G. Tartakovsky, “Modeling temporal activity patterns in dynamic social networks,” *IEEE Transactions on Computational Social Systems*, vol. 1, no. 1, pp. 89–107, 2014.

[13] T. Chis and P. G. Harrison, “Modeling multi-user behaviour in social networks,” in *2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*, 2014, pp. 168–173.

[14] A. De, I. Valera, N. Ganguly, S. Bhattacharya, and M. Gomez-Rodriguez, “Learning and forecasting opinion dynamics in social networks,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS’16, 2016, p. 397–405.

[15] D. Koller and R. Parr, “Computing factored value functions for policies in structured MDPs,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1999, pp. 1332–1339.

[16] N. Forsell and R. Sabbadin, “Approximate linear-programming algorithms for graph-based Markov decision processes,” in *Proceedings of the European Conference on Artificial Intelligence*, 2006, pp. 590–594.

[17] F. Chen, Q. Cheng, J. Dong, Z. Yu, G. Wang, and W. Xu, “Efficient approximate linear programming for factored MDPs,” *International Journal of Approximate Reasoning*, vol. 63, pp. 101–121, 2015.

[18] R. N. Haksar, F. Solowjow, S. Trimpe, and M. Schwager, “Controlling heterogeneous stochastic growth processes on lattices with limited resources,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, Dec 2019, pp. 1315–1322.

[19] R. N. Haksar, J. Lorenzetti, and M. Schwager, “Scalable filtering of large graph-coupled hidden Markov models,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 1307–1314.

[20] R. N. Haksar and M. Schwager, “Constrained control of large graph-based MDPs under measurement uncertainty,” *IEEE Transactions on Automatic Control (TAC)*, 2020, under review.

[21] ——, “Distributed deep reinforcement learning for fighting forest fires with a network of aerial robots,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 1067–1074.

[22] R. N. Haksar, S. Trimpe, and M. Schwager, “Spatial scheduling of informative meetings for multi-agent persistent coverage,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 2, pp. 3027–3034, April 2020.

[23] 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.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[25] C. Williams and G. Hinton, “Mean field networks that learn to discriminate temporally distorted strings,” in *Connectionist Models*, 1991, pp. 18–22.

[26] A. P. Dunmur and D. M. Titterington, “On a modification to the mean field EM algorithm in factorial learning,” in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. MIT Press, 1997, pp. 431–437.

[27] L. K. Saul and M. I. Jordan, “Mixed memory Markov models: Decomposing complex stochastic processes as mixtures of simpler ones,” *Mach. Learn.*, vol. 37, no. 1, p. 75–87, Oct 1999.

[28] V. Raghavan, G. ver Steeg, A. Galstyan, and A. G. Tartakovsky, “Coupled hidden Markov models for user activity in social networks,” in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2013, pp. 1–6.

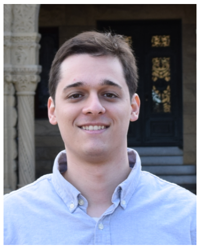
[29] P. Robbel, F. Oliehoek, and M. Kochenderfer, “Exploiting anonymity in approximate linear programming: Scaling to large multiagent MDPs,” in *AAAI Conference on Artificial Intelligence*, 2016, pp. 2537–2543.

[30] California Department of Public Health, “Covid-19 cases,” https://data.ca.gov/dataset/covid-19-cases, accessed 2020-10-28.

[31] Textblob, “Textblob: Simplified text processing,” https://textblob.readthedocs.io/en/dev/.

[32] E. Summers, “Fake news tweets,” https://archive.org/details/fakenews-tweets, accessed 2020-11-04.

[33] A. Hannak, E. Anderson, L. F. Barrett, S. Lehmann, A. Mislove, and M. Riedewald, “Tweetin’ in the rain: Exploring societal-scale effects of weather on mood,” in *International AAAI Conference on Web and Social Media*, 2012.



Ravi N. Haksar Ravi N. Haksar is a Ph.D. candidate with the Department of Mechanical Engineering at Stanford University in the Multi-Robot Systems Lab. He obtained his BS degree from the Georgia Institute of Technology in 2014 and his MS degree from Stanford University in 2017. His research interests include control theory, probabilistic frameworks, decentralized optimization, and cooperative multi-robot teams.

Mac Schwager Mac Schwager is an assistant professor with the Aeronautics and Astronautics Department at Stanford University. He obtained his BS degree in 2000 from Stanford University, his MS degree from MIT in 2005, and his PhD degree from MIT in 2009. He was a postdoctoral researcher working jointly in the GRASP lab at the University of Pennsylvania and CSAIL at MIT from 2010 to 2012, and was an assistant professor at Boston University from 2012 to 2015. He received the NSF CAREER award in 2014, the DARPA YFA in 2018, and a Google faculty research award in 2018. His research interests are in distributed algorithms for control, perception, and learning in groups of robots and animals.