Learning and Predicting Dynamic Behavior with Graphical Multiagent Models

Quang Duong[†] Michael P. Wellman[†] Satinder Singh[†] Michael Kearns[*]
[†]Computer Science and Engineering, University of Michigan
[*]Computer and Information Science, University of Pennsylvania

ABSTRACT

Factored models of multiagent systems address the complexity of joint behavior by exploiting locality in agent interactions. *History-dependent graphical multiagent models* (hGMMs) further capture dynamics by conditioning behavior on history. We propose a greedy algorithm for learning hGMMs from time-series data, inducing both graphical structure and parameters. To evaluate this learning method, we employ human-subject experiment data for a dynamic consensus scenario, where agents on a network attempt to reach a unanimous vote. We empirically show that the learned hGMMs directly expressing joint behavior outperform alternatives in predicting dynamic voting behavior. Analysis of learned graphs reveals patterns of interdependence relations not directly reflected in the original experiment networks.

1. INTRODUCTION

Modeling dynamic behavior of multiple agents presents inherent scaling problems due to the exponential size of any enumerated representation of joint activity. Even if agents make decisions independently, conditioning actions on each other's prior decisions or on commonly observed history induces interdependencies over time. To address this complexity problem, researchers have exploited the localized effects of agent decisions by employing *graphical models* of multiagent behavior. This approach has produced several (related) graphical representations capturing various facets of multiagent interaction [Koller and Milch, 2003, Kearns et al., 2001, Jiang et al., 2008, Gal and Pfeffer, 2008, Duong et al., 2008]. The *history-dependent graphical multiagent models* (hGMMs) of [Duong et al., 2010] express multiagent behavior on an undirected graph, and capture dynamic relations by conditioning action on history. The authors showed that for a fixed graph structure and history representation, the expressive power to specify local joint behavior provided advantages over models that assume conditional independence given history.

It is not always given or apparent how to organize agents on a sufficiently sparse graph for tractable modeling. Information the modeler may have about the agents' experiment network is not definitive, as the graphical structure of the optimal predictive multiagent model of behavior does not necessarily correspond to the experiment network. Moreover, these networks may be too complex for practical computation without imposing strong independence assumptions on behavior. We thus consider inducing the graphical structure a necessary part of the modeling effort.

We motivate and empirically evaluate our learning technique with the dynamic consensus experiments conducted by Kearns et al. [2009]. The human subjects in these experiments were arranged on a network, specifying for each subject (also called *player*, or *agent*) the set of other players whose voting decisions he or she can observe. The experiment network for this voting scenario provides a basis for expecting that joint agent behavior may exhibit some locality that we can exploit in a graphical model for prediction. However, the graph structure of the optimal predictive model, as noted above, need not mirror the experiment network of the voting scenario, and moreover, the complex experiment network instances we study render computation on the corresponding hGMMs intractable. Unlike previous models of this domain, we aim to capture dynamic voting behavior, with no particular focus on the final voting outcome.

In this study, we propose a greedy algorithm for learning the graphical structure and parameters of an hGMM that can effectively and compactly capture joint dynamic behavior. Moreover, we empirically investigate the learned models' predictions of voting behavior and compare their performance with those of different baseline multiagent models, and demonstrate that models expressing joint behavior outperform the alternatives in predicting voting behavior. We further examine the learned hGMM graphical structures in order to gain better insights and understanding of voting behavior dynamics on networks, as well as the network structure's effect on collective actions. We are particularly interested in examining connections between nodes of different characteristics in the hGMM induced graphical structures. For example, our analysis compares the average number of edges between more and less densely connected nodes in the learned hGMM graphs and their corresponding original experiment networks. We further start a discussion on the challenges and potential solutions of predicting final outcomes of the aforementioned dynamic consensus experiments.

The 5th SNA-KDD Workshop '11 (SNA-KDD'11), August 21, 2011, San Diego CA USA . Copyright 2011 ACM 978-1-4503-0225-8...$5.00.

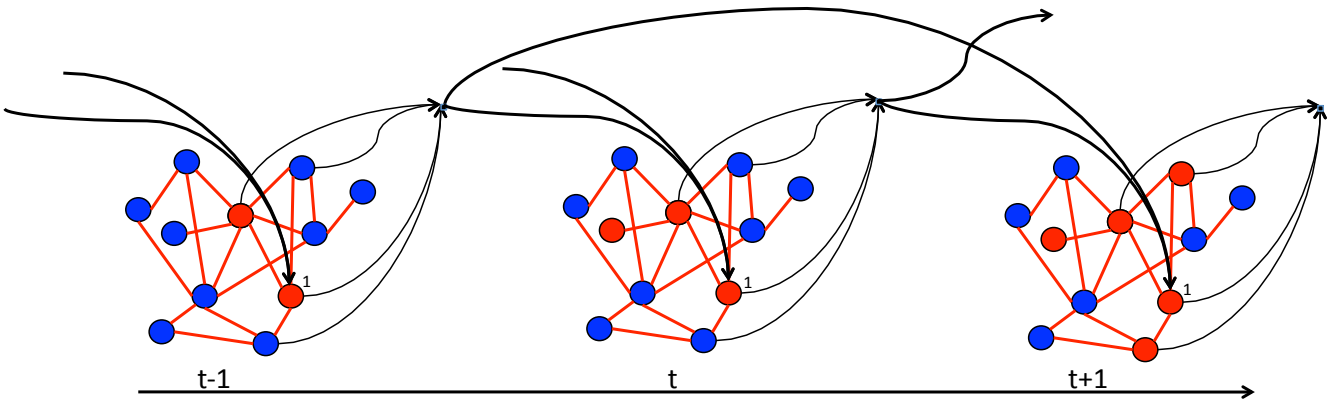

Figure 1: An example hGMM over three time periods. Undirected edges capture the correlation among agents at the present time. Directed edges model the conditioning of an agent's action on others' past actions. Only directed edges pertinent to agent 1 are shown in details: agent 1's current action is conditioned on its neighbors' actions in the previous two time periods.

We provide background information on hGMMs and the dynamic consensus experiments in Sections 2 and 3, respectively. We then present a variety of candidate model forms in Section 4. Section 5 provides motivations and details of our greedy model learning algorithm that simultaneously estimates a model's parameters and constructs its interaction graph. Our empirical study in Section 6 compares different models across three experiment settings, examines the learned graph structures against the original experiment networks, and discusses the problems and potential solutions of predicting these experiments' end state results. We offer concluding remarks and suggest potential extensions in Section 7.

2. HISTORY-DEPENDENT GRAPHICAL MULTIAGENT MODELS

We model behavior of n agents over a time interval divided into discrete periods, $[0, \ldots, T]$. At time period t, agent $i \in \{1, \ldots, n\}$ chooses an action a_i^t from its action domain, A_i, according to its *strategy*, σ_i. Agents can observe others' and their own past actions, as captured in *history* H^t, up to time t. Limited memory capacity or other computational constraints restrict an agent to focus attention on a subset of history H_i^t considered in its probabilistic choice of next action: $a_i^t \sim \sigma_i(H_i^t)$.

A *history-dependent graphical multiagent model* (hGMM) [Duong et al., 2010], $hG = (V, E, A, \pi)$, is a graphical model with graph elements V, a set of vertices representing the n agents, and E, edges capturing pairwise interactions between them. Component $A = (A_i, \ldots, A_n)$ represents the action domains, and $\pi = (\pi_1, \ldots, \pi_n)$ potential functions for each agent. The graph defines a neighborhood for each agent i: $N_i = \{j \mid (i,j) \in E\} \cup \{i\}$, including i and its neighbors $N_{-i} = N_i \setminus \{i\}$.

The hGMM captures agent interactions in dynamic scenarios by conditioning joint agent behavior on an abstracted history of actions H^t. The history available to agent i, $H_{N_i}^t$, is the subset of H^t pertaining to agents in N_i. Each agent i is associated with a potential function $\pi_i(a_{N_i}^t \mid H_{N_i}^t)$: $\prod_{j \in N_i} A_j \rightarrow R^+$. The potential of a local action configuration specifies its likelihood of being included in the global outcome, conditional on history. Specifically, the joint distribution of the system's actions taken at time t is the product of neighbor potentials [Daskalakis and Papadimitriou, 2006, Duong et al., 2010, Kakade et al., 2003]:

$$\Pr(a^t \mid H^t) = \frac{\prod_i \pi_i(a_{N_i}^t \mid H_{N_i}^t)}{Z}. \quad (1)$$

The complexity of computing the normalization factor Z in (1) is exponential in the number of agents, and thus precludes exact inference and learning in large models. Duong et al. [2010] address this problem by approximating Z using the *belief propagation* method [Broadway et al., 2000], which has shown good results with reasonable time in sparse cyclic graphical structures. In particular, we adopt the package libDAI [Mooij, 2010] for approximating Z in our hGMM implementation.

We stress that there are two different types of edges in an hGMM, as depicted in Figure 1. Each node i's undirected edges define its neighborhood N_i^u of the present time's configuration $a_{N_i^u}^t$, and thus model the correlations among their actions. The directed edges ending at i originate from nodes whose actions in the past influence how i chooses its action in the present, and consequently form i's other neighborhood N_i^d. These two different neighborhood definitions lead to a more generalized form of the potential function $\pi_i(s_{N_i^u}^t \mid s_{N_i^d}^{t-1})$. For simplicity, this paper assumes that N_i^u, N_i^d and N_i are the same for all i, and employ the potential function form $\pi_i(s_{N_i}^t \mid s_{N_i}^{t-1})$.

3. DYNAMIC CONSENSUS EXPERIMENTS

We evaluate our approach with human-subject data from the dynamic consensus game introduced and studied by Kearns et al. [2009]. Each agent in this game chooses to vote either blue (0) or red (1), and can change votes at any time. Agents are connected in a network, where each can observe its neigh-

bors' votes. The scenario terminates when: (i) agents converge on action $a \in \{0, 1\}$, in which case agent i receives reward $r_i(a) > 0$, or (ii) they cannot agree by the time limit T, in which case rewards are zero. Figure 2 illustrates the dynamic behavior of an example voting experiment network.

Agents may have different preferences for the available vote options, reflected in their reward functions. As nobody gets any reward without a unanimous vote, agents have to balance effort to promote their own preferred outcomes against the common goal to reach consensus. Another important feature of the dynamic consensus game is that agent i knows the votes of only its neighbors, and its own local graph structure, including its neighbors N_i, the degree of each neighbor $k \in N_i$, and edges between its neighbors. This raises the question of how agents take into account their neighbors' voting patterns and their partial knowledge of experiment network structure.

Kearns et al. [2009] conducted a series of human-subject experiments studying how human agents behave in 81 different instances of the voting game. They varied reward preference assignments and experiment network structure in these experiment instances, and thus were able to collect data about these factors' effects on the consensus voting results, as well as the agent strategies employed. Figure 2 exhibits a run for the experimental network labeled power22, discussed below. Study goals included developing models to predict a given scenario's voting outcome, and if a consensus is reached, its convergence time. This problem also served as the foundation for analysis of adaptive strategies and theoretical constraints on convergence [Kearns and Tan, 2008]. In particular, they were interested in developing models that would predict whether a given scenario would be likely to converge to consensus, and if so, how fast and on which outcome. Exploring the problem also led this group to analyze a family of adaptive strategies, and establish the impossibility of converging to the preferred outcome in the worst case [Kearns and Tan, 2008].

4. MODELING DYNAMIC VOTING BEHAVIOR

We present four multiagent behavior model forms designed to capture voting behavior dynamics in the dynamic consensus experiments. All are expressible as hGMMs. Only the first, however exploits the flexibility of hGMMs to express dependence of actions within a neighborhood given history (1), hence we refer to this as the *joint behavior model* (JBM).

The other three forms model agent behaviors individually: for each agent we specify a probabilistic strategy $\sigma_i(H_i^t) = \Pr(a_i^t \mid H_i^t)$. Such a formulation captures agent interactions by the conditioning of individual behavior on observed history. The agents' actions are probabilistically dependent, but conditionally independent given this common history, yielding the joint distribution

$$\Pr(a^t \mid H^t) = \prod_i \sigma_i(H_i^t). \tag{2}$$

We refer to a dynamic multiagent model expressible by (2) as an *individual behavior hGMM* (IBMM). Conditional independence given history is a compelling assumption for autonomous agents. Indeed, independent choice may even be considered definitional for autonomy. In practice, however, it is often infeasible to specify the entire history for conditioning due to finite memory and computational power, and the assumption may not hold with respect to partial history. History abstraction will generally introduce correlations between agents actions, even if they are independently generated on full history [Duong et al., 2010]. Nevertheless, assuming conditional independence between agents' actions given history exponentially reduces the model's complexity, or more specifically, the computational complexity of the joint probability distribution of the system's actions.

The first of three IBMMs we present is designed as an independent behavior version of JBM; thus, we call it simply the *individual behavior model* (IBM). The remaining two models are based on proposals and observations from the original experimental analysis [Kearns et al., 2009], and are labeled *proportional response model* (PRM) and *sticky proportional response model* (sPRM), respectively.

4.1 JOINT BEHAVIOR MODEL

First, we consider how to summarize a history $H_{N_i}^t$ of length h relevant to agent i. Let indicator $I(a_i, a_k) = 1$ if $a_i = a_k$ and 0 otherwise, and $I(a_{N_i}^{t_1}, a_{N_i}^{t_2}) = 1$ iff $I(a_k^{t_1}, a_k^{t_2}) = 1$ for all $k \in N_k$. We encode the historical frequency of a local configuration a_{N_i} as

$$e(a_{N_i}, H_{N_i}^t) = \frac{\sum_{\tau=t-h}^{t-1} I(a_{N_i}, a_{N_i}^\tau)}{h}.$$

However, this encoding only counts exact matches of a_{N_i} in history H_{N_i}, and thus heavily biases against local configurations that have not happened in the past, effectively hindering a neighborhood of agents with different vote preferences from reach consensus. We introduce an alternative frequency function that captures how close a local configuration to past configurations:

$$f(a_{N_i}, H_{N_i}^t) = \frac{\sum_{\tau=t-h}^{t-1} 0.5^{\sum_{k \in N_i} 1 - I(a_k, a_k^\tau)}}{h},$$

where the exponent $\sum_{k \in N_i} 1 - I(a_k, a_k^\tau)$ basically counts the number of pair-wise mismatches between a_{N_i} and $a_{N_i}^\tau$ and the discount factor is fixed as 0.5. To simplify exposition, we henceforth drop the time superscript t and the neighborhood subscript N_i from $H_{N_i}^t$.

In formulating JBM's potential function, we attempt to capture the impact of past collective choices of i's neighborhood, and i's relative preference for each action. As subjects in the lab experiments were able to keep track of how much time they have left before having to reach a consensus, we also would like to capture the time effect on agents' reasoning in the potential function. We define $r_i(a_{N_i})$ as the product of time-discounted $r(a_i)$ and a heuristic attenuation based on how many neighbors currently vote differently:

$$r_i(a_{N_i}) = \alpha^{\sum_{k \in N_i} (1 - I(a_i, a_k))} r(a_i)^{\frac{T-t}{T}\beta},$$

where $\alpha \in [0, 1]$ and $\beta \geq 0$. Observe that $r_i(a_{N_i})$ as we define it is increasing in the number of iÕs neighbors playing a_i, reflecting the positive influence of neighbor choices on i. The potential function for agent i is given by

$$\pi_i(a_{N_i} \mid H) = r_i(a_{N_i}) f(a_{N_i}, H)^\gamma, \tag{3}$$

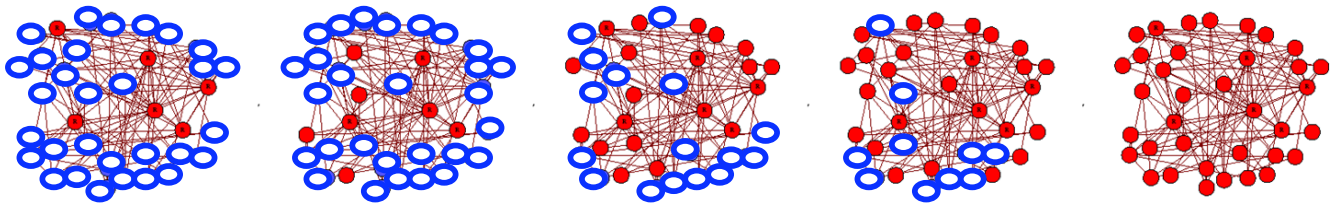

Figure 2: Time snapshots of an experiment run where the densely connected minority group (red) exerts strong influences on others' votes [Kearns et al., 2009].

where $\gamma \geq 0$ denotes the weight or importance of the historical frequency $f(a_{N_i}, H)$ relative to the estimated reward $r_i(a_{N_i})$. The normalized product of these potentials specifies joint behavior as described in (1). The model maintains three parameters α, β and γ.

4.2 INDIVIDUAL BEHAVIOR MODEL

The IBM of the dynamic consensus experiments retains the main elements of JBM (3) with a similar parameter set $\{\alpha, \beta, \gamma\}$, while imposing conditional independence among agents' actions given the common history. Let us define $f(a_i, H^t_{N_i})$ as the frequency of action a_i being chosen by other agents in $H^t_{N_i}$, capturing the degree to which a_i is similar to past choices by i's neighbors in $H^t_{N_i}$.

$$f(a_i, H^t_{N_i}) = \frac{\sum_{k \in N_i - \{i\}} \sum_{\tau=t-h}^{t-1} I(a_i, a^\tau_k) + 1}{h|N_i \setminus \{i\}|} \quad (4)$$

We add one to the numerator to ensure that the corresponding term does not vanish when the configuration a_i does not appear in $H^t_{N_i}$. The probabilistic IBM strategy is then given by:

$$\Pr(a_i \mid H) = \frac{1}{Z_i} r_i(a_i)^{\alpha + \frac{T-t}{T}\beta} f(a_i, H)^\gamma. \quad (5)$$

Z_i is the normalization factor over all $a_i \in A_i$ (this normalization sums only over the actions of a single agent and is, therefore, easy to compute).

4.3 PROPORTIONAL RESPONSE MODEL

We also include in our study the proportional response model, PRM, suggested by Kearns et al. [2009] as a reasonably accurate predictor of their experiments' final outcomes. PRM specifies that voter i chooses action a_i at time t with probability proportional to $r_i(a_i)g(a_i, a^{t-1}_{N_i})$, where $g(a_i, a^{t-1}_{N_i})$ denotes the number of i's neighbors who chose a_i in the last time period,

$$\Pr(a^t_i \mid H^t) \sim r_i(a^t_i) g(a_i, a^{t-1}_{N_i}). \quad (6)$$

4.4 STICKY PROPORTIONAL RESPONSE MODEL

PRM does not capture players' tendency to start with their preferred option, switching their votes only after collecting additional information about their neighbors over several time periods [Kearns et al., 2009]. Therefore, we introduce the *sticky proportional response model*, sPRM, which contains an additional parameter ρ reflecting an agent's stubbornness in voting its preferred option, regardless of its neighbors' past choices. Intuitively, a player's inherent bias toward its preferred option decays exponentially until there is no bias:

$$\Pr(a^t_i \mid H^t) \sim r_i(a^t_i) g(a_i, a^{t-1}_{N_i})(1 + I^{\max}_{a_i} e^{1-t\rho}), \quad (7)$$

where $I^{\max}_{a_i} = 1$ if $a_i = \arg\max r_i(a)$ and 0 otherwise.

5. LEARNING PARAMETERS AND GRAPHICAL STRUCTURES

5.1 PARAMETER LEARNING

We first address the problem of learning the parameters of an hGMM hG given the underlying graphical structure and data in the form of a set of joint actions for m time steps, $X = (a^0, \ldots, a^m)$. For ease of exposition, let θ denote the set of all the parameters that define the hGMM's potential functions. We seek a θ maximizing the log likelihood of X,

$$L_{hG}(X; \theta) = \sum_{k=0}^{m-h} \ln(\Pr{}_{hG}(a^{k+h} \mid (a^k, \ldots, a^{k+h-1})); \theta)).$$

We use gradient ascent to update the parameters: $\theta \leftarrow \theta + \lambda \nabla\theta$, where the gradient is

$$\nabla\theta = \frac{\partial L_{hG}(X; \theta)}{\partial \theta},$$

and λ is the learning rate, stopping when the gradient is below some threshold. We employ this same technique to learn the parameters of the baseline models.

5.2 MODEL LEARNING

Each of the consensus voting experiments involves 36 human players. The largest neighborhood size in these games ranges from 16 to 20, rendering computing exact data likelihood for a joint behavior model of this complexity (required for parameter learning described above) infeasible. Preliminary trials with the belief propagation approximation algorithm [Broadway et al., 2000] on these models indicated that its computational saving would still be insufficient for effective learning. Thus, we need to employ models with simpler graphs in order to take advantage of hGMM's expressiveness in representing joint behavior. Toward this end, we developed a structure learning algorithm that produces graphs for hGMMs within specified complexity constraints.

Though dictated by computational necessity, automated structure learning has additional advantages. First, we observe that there is no inherent reason that the interaction graph should constitute the ideal structure for a predictive graphical model for agent behavior. Even though actual agent behavior is naturally conditioned on its observable history (as captured by the interaction graph), once we abstract the history representation it may well turn out that non-local

historical activity provides more useful predictive information. If so, the structure of the learned graph itself may provide interesting insights on the agents' networked behavior.

Note that the complexity problem does not apply for IBMMs, which are tractable based on their conditional independence assumption. Nevertheless, since as GMMs their graphical structures capture dependence unconditional on history among agents' actions, it may also be interesting to employ our algorithm for learning IBMM structure.

Our learning algorithm combines greedy addition of edges with gradient-descent parameter optimization, as described by the following steps:

1: Start with an hGMM whose graphical structure $(V, E = \emptyset)$ is completely disconnected.
2: Learn θ that maximizes the training data's likelihood $L(X;\theta)$
3: **repeat**
4: **repeat**
5: $\tilde{E} \leftarrow \{(i,j) \mid (i,j) \notin E; |N_i|, |N_j| < d_{\max}; \tilde{L}_{(i,j)} = L_{E\cup(i,j)}(X;\theta) \geq L(X;\theta)\}$
6: $E \leftarrow E \cup (i_{\max}, j_{\max})$ such that $(i_{\max}, j_{\max}) = \arg\max_{(i,j)\in\tilde{E}} \tilde{L}_{(i,j)}$
7: **until** $\tilde{E}$ is empty or the number of added edges is greater than $e_{\max}$
8: Learn θ that maximizes the training data's likelihood $L(X;\theta)$
9: **until** no edges can be added

5.3 EVALUATION

We evaluate the learned multiagent models by their ability to predict future outcomes, as represented by a test set Y. Given two models M_1 and M_2, we compute their corresponding log-likelihood measures for the test data set Y: $L_{M_1}(Y)$ and $L_{M_2}(Y)$. Note that since log-likelihood is negative, we instead examine the negated log-likelihood measures, which means that M_1 is better than M_2 predicting Y if $-L_{M_1}(Y) < -L_{M_2}(Y)$, and vice versa.

6. EMPIRICAL STUDY

We empirically evaluate the predictive power of JBM in comparison with IBM, PRM, and sPRM, using the dynamic consensus experiment data from Kearns et al. [2009]. We also compare JBM against a naive guessing model, nM, which initially assigns each a_i a probability proportional to $r_i(a_i)$ and linearly converges to a uniform distribution of agent actions as the game progresses. We are further interested in examining the graphs induced by structure learning, and relating them to their corresponding original game networks using different statistical measures.

The human-subject experiments are divided into nine different sets, each associated with a network structure. These structures differ qualitatively in various ways, characterized by node degree distribution, ratio of inter-group and intra-group edges, and the existence of a well-connected minority [Kearns et al., 2009]. In particular, networks whose edges are generated by a random Erdos-Renyi (ER) process [Erdos and Renyi, 1959] has a notably more heavy-tailed degree distribution than those generates by a preferential attachment (PA) process [Barabási and Albert, 1999]. For each experimental trial, human subjects were randomly assigned to nodes in the designated network structure, and preferences based on one of three possible incentive schemes. Since players in these experiments can change their votes at any time, the resulting data is a stream of asynchronous vote actions. We discretize these streams for data analysis, recording the players' votes at the end of each time interval of length δ seconds. We experiment with different interval lengths $\delta \in \{0.5, 1.5\}$ in this study.

In our study, we learn predictive models for each network structure, pooling data across subject assignments and incentive schemes. This approach is based on the premise that network structure is the main factor governing the system's collective behavior, in line with the findings of Kearns et al. [2009]. In each experiment set, we use four of the nine trials for training the predictive models for each form. The hGMM graphical structures are learned with node degree constraint $d_{\max} = 10$, while the maximum degree of the individual behavior models is restricted to the greatest node degree of the original network. We also set $e_{\max} = 5$. We then evaluate these models based on their predictions over a test set comprising the other five experimental trials. This process is repeated five times, each of which uses a different training trial set randomly chosen from the original trials. Each data point in our reported experimental results averages over these five repetitions.

We reuse Kearns et al.'s labels for the three experiment networks studied in this analysis, listed in Table 1.

Table 1: Voting Experiment Settings

Label	Strong Minority	Graph Generator Process
coER_2	No	Erdos-Renyi
coPA_2	No	Preferential attachment
power22	Yes	Preferential attachment

6.1 PREDICTIONS

We first examine predictions of players' votes in each time period conditional on available history. Figure 3 shows the negated log-likelihood of the testing data set induced by models JBM,IBM, PRM, sPRM, and nM. Note that the learned graphical structures in JBM are different from the original game experiments' networks, upon which other models are constructed. We observe that JBM performs significantly better than IBM, PRM, sPRM, and nM in predicting dynamic agent behavior in the dynamic consensus experiments for all three experiment sets, given data discretized with two different interval lengths of 0.5 and 1.5 (differences significant at $p < 0.02$. Contrary to the expectation that the less historical information a model uses, the lower its prediction performance, JBM and IBM that employ only the last $h = 3$ periods of historical data generate better predictions than those with $h = 8$. This phenomenon is likely a consequence of the heuristic nature of the frequency functions in Section 4, and moreover may indicate that human subjects take into account only a short history of their neighbors' actions when choosing their own present actions.

We also note that the results remain largely the same with

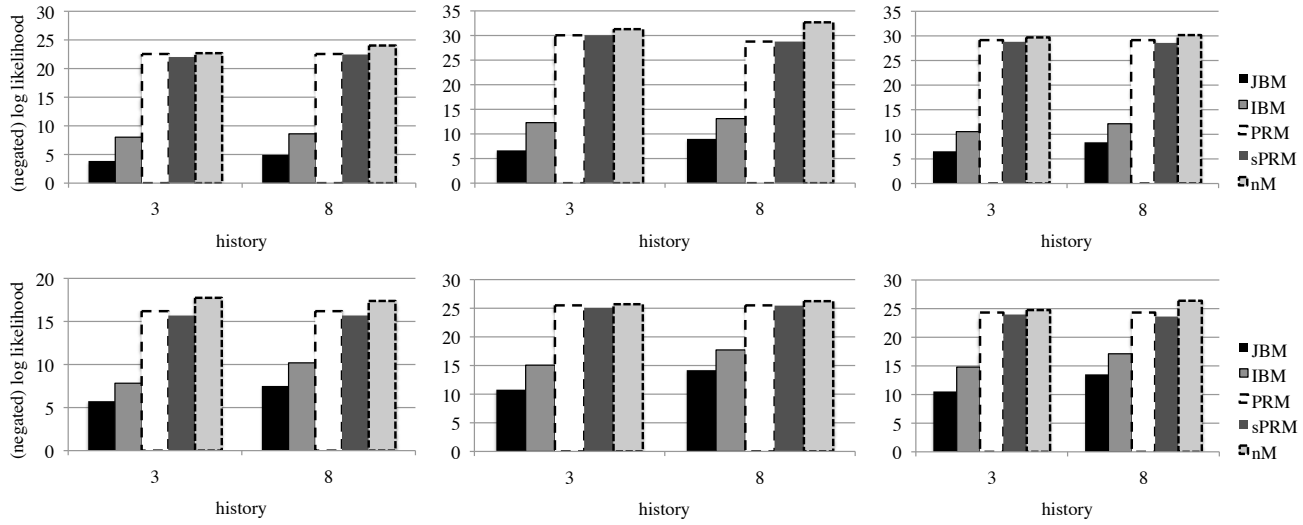

Figure 3: JBM provides better predictions than IBM, PRM, sPRM, and nM in three experiment sets with two different interval lengths $\delta = 0.5$ (top row) and $\delta = 1.5$ (bottom row): power22 (left), coER_2 (middle), and coPA_2 (right).

different discretization intervals δ. The large ratio between the two examined intervals implies a significant difference in the quantity and quality of discretized data, further highlighting the robustness of our learning method, as well as our empirical prediction results. These outcomes in general demonstrate JBM's ability to capture joint dynamic behavior, especially behavior correlations induced by limited historical information, as opposed to different IBMMs and a naive guessing model.

6.2 GRAPH ANALYSIS

Our first graph analysis focuses on the connections within and between nodes of different degrees in the learned graphs. We mainly focus on experiments with history length $h = 3$ and discretization interval $\delta = 0.5$. We partition the nodes into four classes, based on their degrees in the original experiment networks. Class 4 contains the most connected nodes, and class 1 the least. We narrow our focus on classes 1 and 4: in particular, we examine the average number of *inter edges*, which connect nodes within the same class, and *intra* edges, which link nodes in different classes. Note that intra and inter edges concern their end nodes' degree, while the characterization of intra-group and inter-group edges of experiment networks is based on the end nodes' vote preferences. Let us define E^-_M as the set of edges present in the original G, but excluded in the graph M, and E^+_M as the set of edges not present in G, but included in M.

We observe in Figure 4 that the distribution of edges within and between node classes of different connectivity in JBM diverges significantly from that in the original network for coER_2 and power22.[1] The degree constraint in the learning algorithm forces JBM to shed edges from class 4 nodes. In fact, the learned models end up with more intra edges in class 1 than class 4, whereas G has more intra edges in class 4 than in class 1.

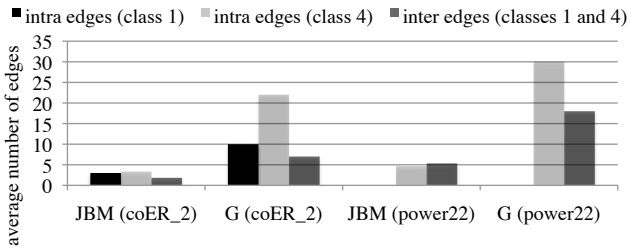

Figure 4: Distributions of edges within and between classes 1 and 4, for JBM and G in scenarios coER_2 and power22 ($h = 3$ and $\delta = 0.5$).

Our second analysis study examines different statistics and properties of E^+_{JBM}, E^-_{JBM}. In particular, we find that 40% to 60% of the edges in E^+_{JBM} connect nodes that share some neighbors in G, suggesting that JBM manages to identify nodes whose behavior correlates due to their observing common neighbors. Our investigation further focuses on *bridge* edges, which connect nodes that share no common neighbors in the original G. Let B denote the set of all bridges in G. Further, define $b^-_{\mathsf{M}} = \frac{|E^-_{\mathsf{M}} \cap B|}{|E^-_{\mathsf{M}}|}$ as the percentage of bridges among edges eliminated during the construction of model M, and $b_{\mathsf{G}} = \frac{|B|}{|G|}$ as the percentage of bridges in the original G. We observe that the percentage of eliminated bridges E^-_{JBM} is consistently below the percentage of bridges over all edges in G, as shown in Table 2, though only by small margins. This result may suggest the learning algorithm's slight preference in retaining bridges rather than edges of other types.

[1]We only include results for coER_2 and power22 in our graph analysis session, as coPA_2's outcomes appear similar to those of coER_2.

Table 2: Percentage of bridges in the set of eliminated edges by JBM and bridges in the set of all G's edges

	coER_2	power22
b^-_{JBM}	42%	30%
b_{G}	44%	31%

Table 3: JBM greatly diverges from G in terms of assortativity.

	coER_2	power22
JBM	-0.081	0.09
G	0.09	0.09

The last analysis concerns the learned graphs and original experiment networks' *assortativity* [Newman, 2003]. A graph G's assortativity coefficient $v_G \in [-1, 1]$ captures the tendency for nodes to attach to others that are similar or different in connectivity. Positive values of v_G indicate a correlation between nodes of similar degree, whereas negative values indicate relationships between nodes of different degrees.

The table shows that in scenario coER_2 JBM graphs exhibit negative assortativity, whereas the original network's v_{G} is positive. In other words, JBM graphs appear to contain sparsely connected hubs of highly connected nodes, which implies the learning method's ability to identify nodes of greater influence in the original networks. This observation is further supported by the significant difference in the proportion of intra edges among highly connected nodes between G and JBM in coER_2, as presented in Figure 4. Figure 4 indicates the same phenomenon for scenario power22, which, however, does not correspond to any divergence between G's and JBM's assortativity in the same scenario. We attribute this discrepancy to the existence of a strongly connected minority group in power22. Overall, the retention rate of G's edges within the minority group in the learned JBM (19.5%) is higher than that within class 4 (15.6%), which contains both minority and majority nodes. The structure learning algorithm appears to be able to identify the importance and influence of the minority group, a phenomenon also observed and reported by Kearns et al. [2009].

6.3 DISCUSSIONS OF CONSENSUS OUTCOMES

We also evaluate the models' capacity to predict the end state of a dynamic consensus experiment. As noted above, the original aim of modeling in these domains was to predict this final outcome. For a particular model M, we start an experiment run with agents choosing their preferred colors, and then proceed to draw samples from M for each time period until a consensus is reached or the number of time periods exceeds the time limit. We average over 500 runs for each experiment setting and model.

The convergence of PRM to consensus strongly correlates with observed experimental results in terms of the percentage of experiments reaching consensus, as shown in Figure 5. However, PRM's predictions on the color of consensus are out of line with the actual experiments for some experiment settings, rendering PRM ineffective in predicting end state results.[2]

We find that simulated runs drawn from JBM converge to a consensus at a relatively lower rates than PRM in the power22 setting where all lab experiments reach consensus, and at a significantly lower rates in other settings. As a result, JBM is not directly useful for predicting end state specifically. That the model's success in capturing transient dynamics fails to translate to outcome prediction is an interesting anomaly. It will be worth further investigating whether incorporating time dependence or other additional factors can remedy the discrepant effectiveness.

It is important to note that the input of our study has been transformed from ordinal-time to snapshot data [Cosley et al., 2010], which inherently results in a loss of information about the exact vote-update times. Moreover, the difference in the discretization interval δ does make a noticeable difference in the end-game results for both JBM and PRM, while the effect of δ in predicting transient dynamics is negligible. These observations suggest the need to experiment with constructing and evaluating asynchronous versions of this study's models.

We further observe that in settings coER_2 and coPA_2 where consensus appears less frequently, the vote majority tended to switch from one color to another before assuming its original color. As we suspect that the presence of "stubborn" players fueled this pattern, JBM could benefit from having an additional history summary function other than f that captures the level of "stubbornness" of an agent, gains greater importance as time passes, and thus effectively informs the potential function about future vote updates. We will also further incorporate in our models elements from different statistical models specifically developed for capturing agents' abilities to reach consensus on distributed networks [Masuda et al., 2010, Mossel and Schoenebeck, 2010]

7. CONCLUSIONS

We have introduced and empirically evaluated an algorithm for learning history-dependent graphical multiagent models of dynamic behavior given time series of agent actions. The empirical study demonstrates an ability to learn compact graphical representations capturing the dynamics of human-subject voting behavior on a network. In particular, we have shown that the learned joint behavior model JBM provides significantly better predictions of dynamic behavior than different individual behavior models, including the multiplicative model PRM and its variation sPRM, suggested by the original experimental analysis. This provides evidence that expressing joint behavior is important for dynamic modeling, even given partial history information for conditioning individual behavior. Our graph analysis further reveals characteristics of the learned JBM graphical structures, particu-

[2]Kearns et al. [2009] only examined the correlations between the number of seconds for human subjects to reach consensus during lab experiments and the number of simulation updates by a PRM-type model before reaching consensus, but not the color of consensus.

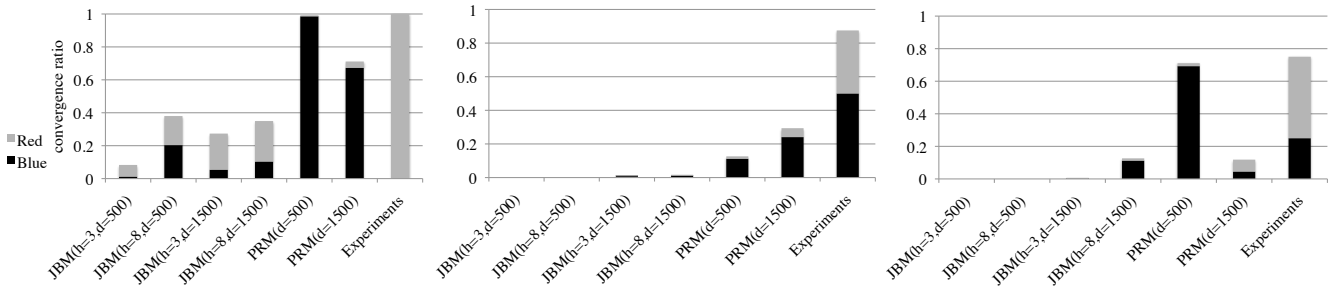

Figure 5: End-game results for three different experiment groups: power22 (left), coER_2 (middle), and coPA_2 (right).

larly in how they diverge from the original experiment networks.

In future work we plan to improve the learning algorithm for individual behavior models, which are currently constructed with a predetermined maximum degree constraint, by replacing this degree constraint with a cross-validation condition that can better help avoid overfitting. Another potential research study would examine and compare different approaches to learning hGMM structure in terms of efficacy and complexity. We would also like to build on this study's graph analysis a more systematic and comprehensive toolbox for dissecting graphical models of dynamic behavior and assessing model learning algorithms' contributions. Finally, we are interested in applying our modeling technique in studying similar problem domains where agents must coordinate their actions or make collective decisions while only communicating with their neighbors [Suri and Watts, 2011, Judd et al., 2010], as well as large network scenarios, such as social networks and internet protocols.

References

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

Jonathan Yedidia Broadway, Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized belief propagation. In *Thirteenth Annual Conference on Advances in Neural Information Processing Systems*, pages 689–695, Denver, 2000.

Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, Xiangyang Lan, and Siddharth Suri. Sequential influence models in social networks. In *Fourth International AAAI Conference on Weblogs and Social Media*, pages 26–33, Washington, DC, 2010.

Constantinos Daskalakis and Christos H. Papadimitriou. Computing pure Nash equilibria in graphical games via Markov random fields. In *Seventh ACM conference on Electronic Commerce*, pages 91–99, Ann Arbor, MI, 2006.

Quang Duong, Michael P. Wellman, and Satinder Singh. Knowledge combination in graphical multiagent models. In *Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 153–160, Helsinki, 2008.

Quang Duong, Michael P. Wellman, Satinder Singh, and Yevgeniy Vorobeychik. History-dependent graphical multiagent models. In *Ninth International Conference on Autonomous Agents and Multiagent Systems*, Toronto, 2010.

P. Erdos and A. Renyi. On random graphs. i. *Publicationes Mathematicae*, 6(290-297):156, 1959.

Ya'akov Gal and Avi Pfeffer. Networks of influence diagrams: A formalism for representing agents' beliefs and decision-making processes. *Journal of Artificial Intelligence Research*, 33:109–147, 2008.

Albert Xin Jiang, Kevin Leyton-Brown, and Navin A. R. Bhat. Action-graph games. Technical Report UBC CS TR-2008-13, University of British Columbia, 2008.

Stephen Judd, Michael Kearns, and Yevegeiny Vorobeychik. Behavioral dynamics and influence in networked coloring and consensus. *Proceedings of the National Academy of Sciences*, 107(34):14978, 2010.

Sham Kakade, Michael Kearns, John Langford, and Luis Ortiz. Correlated equilibria in graphical games. In *Fourth ACM Conference on Electronic Commerce*, pages 42–47, San Jose, CA, 2003.

Michael Kearns and Jinsong Tan. Biased voting and the Democratic primary problem. In *Fourth International Workshop on Internet and Network Economics*, pages 639–652, Shanghai, 2008.

Michael Kearns, Michael L. Littman, and Satinder Singh. Graphical models for game theory. In *Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 253–260, Seattle, 2001.

Michael Kearns, Stephen Judd, Jinsong Tan, and Jennifer Wortman. Behavioral experiments on biased voting in networks. *Proceedings of the National Academy of Sciences*, 106(5):1347–52, 2009.

Daphne Koller and Brian Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45:181–221, 2003.

N. Masuda, N. Gibert, and S. Redner. Heterogeneous voter models. *Physical Review E*, 82(1):100–103, 2010.

Joris M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, 2010.

Elchanan Mossel and Grant Schoenebeck. Reaching consensus on social networks. In *First Symposium on Innovations in Computer Science*, Beijing, China, 2010.

M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, 67(2), 2003.

Siddharth Suri and Duncan J. Watts. Cooperation and contagion in web-based, networked public goods experiments. *PloS One*, 6(3):e16836, 2011.