Learning and Predicting Dynamic Networked Behavior with Graphical Multiagent Models

Quang Duong[†] Michael P. Wellman[†] Satinder Singh[†] Michael Kearns[*]
{qduong,wellman,baveja}@umich.edu mkearns@cis.penn.edu
[†]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. The challenges of modeling real human behavior motivated us to extend the hGMM representation by distinguishing two types of agent interactions. This distinction opens the opportunity for learning dependence networks that are different from given graphical structures representing observed agent interactions. We propose a greedy algorithm for learning hGMMs from time-series data, inducing both graphical structure and parameters. Our empirical study employs human-subject experiment data for a dynamic consensus scenario, where agents on a network attempt to reach a unanimous vote. We show that the learned hGMMs directly expressing joint behavior outperform alternatives in predicting dynamic human voting behavior, and end-game vote results. Analysis of learned graphical structures reveals patterns of action dependence not directly reflected in the original experiment networks.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Multiagent Systems

General Terms

Experimentation, Algorithms, Human Factors

Keywords

graphical models, dynamic behavior, structure learning

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 [9, 11, 6, 5, 3]. *History-dependent graphical multiagent models* (hGMMs) [4] express multiagent behavior on an undirected graph, and capture dynamic relations by conditioning actions on history.

Prior work on hGMMs presumes a fixed graph structure defined by the modeler [4]. However, it is not always apparent how to choose the most salient inter-agent dependencies for accurate and tractable modeling. We seek methods for inducing hGMM structures from observational data about dynamic multiagent scenarios. In the process, we also extend the flexibility of hGMMs by allowing distinct dependence structures for within-time and across-time probabilistic relationships.

We empirically evaluate our techniques with data from laboratory experiments on dynamic consensus [8]. Human subjects were arranged on a network, specifying for each subject (also called *agent*) the set of others whose current choices are observable. The network associated with each experiment provides a basis for expecting that joint agent behavior may exhibit some locality that we can exploit in a graphical model for prediction.

We stress that the graph structure of the optimal predictive model 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. Therefore, we attempt to learn the graphical structure and parameters of an hGMM that can effectively and compactly capture joint dynamic behavior. Using human subject data, we evaluate the learned models' predictions of voting behavior and compare their performance with those of different baseline multiagent models. We generally find that models expressing joint behavior outperform the alternatives, including models originally proposed by authors of the dynamic consensus experiments, in predicting voting dynamics. The joint behavior model provides comparable predictions on the rate of reaching consensus, and superior predictions of which consensus is reached. We further examine the learned hGMM graphical structures in order to gain insights about the dependencies driving voting behavior, as well as the network structure's effect on collective action.

Sections 2 provides background information on hGMMs, and introduces our extension to the modeling framework. Section 3 describes the dynamic consensus experiments. We present a variety of candidate model forms in Section 4. Sec-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

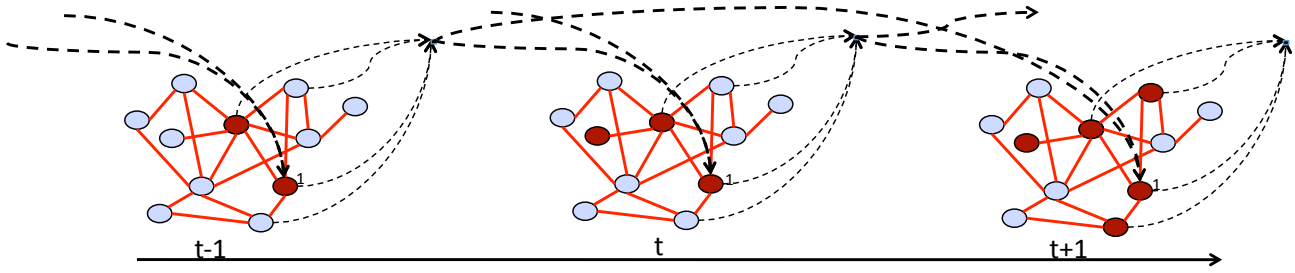

Figure 1: An example hGMM over three time periods. Undirected edges capture correlation among agents at a point in time. Directed edges (shown here only for agent 1) denote conditioning of an agent's action on others' past actions.

tion 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, and examines the learned graph structures against the original experiment networks.

2. HISTORY-DEPENDENT GMMS

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 up to time t, as captured in *history* $H^t = \{H_1^t, \ldots, H_n^t\}$, where H_i^t denotes the sequence of actions agent i has taken by 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) [4], $hG = (V, E, A, \pi)$, 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 $N_{-i} = N_i \backslash \{i\}$.

The hGMM representation 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 normalized product of neighbor potentials [2, 4, 7]:

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

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. We approximate Z using the *belief propagation* method [1], which has shown good results with reasonable time in sparse cyclic graphical structures.

We extend the original hGMM representation by distinguishing between within-time and across-time dependencies, as depicted in Figure 1. Formally, we introduce a *conditioning set* Γ_i for each i, denoting the set of agents whose histories condition this agent's potential function: $\pi_i(a_{N_i}^t \mid H_{\Gamma_i}^t)$. The neighborhood N_i in this extension governs only the within-time probabilistic dependencies of node i. With respect to this extended model, the original hGMM [4] corresponds to the special case where $\Gamma_i = N_i$. The joint distribution of actions at time t can be rewritten as:

$$\Pr(a^t \mid H^t) = \frac{\prod_i \pi_i(a_{N_i}^t \mid H_{\Gamma_i}^t)}{Z}. \quad (2)$$

3. DYNAMIC CONSENSUS

We evaluate our modeling framework with human-subject data from a dynamic consensus game [8]. 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, such that agent i can observe the votes of those in its *observation neighborhood* N_i^O. 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 have varying preferences for the possible consensus outcomes, 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 observes the votes of only those in its observation neighborhood N_i^O, and all it is shown of the graph is the degree of each observation neighbor, and observation edges among them. This raises the question of how agents take into account their neighbors' voting patterns and their partial knowledge of the experiment network structure.

A series of human-subject experiments were conducted to study how people behave in 81 different instances of the voting game [8]. The experimenters 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, and the 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 founda-

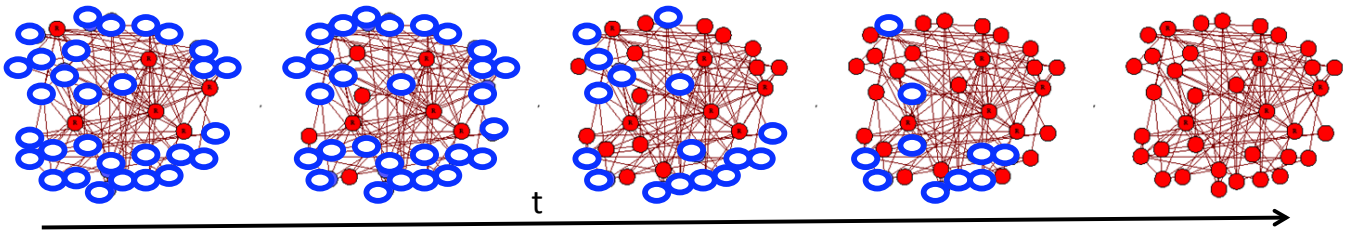

Figure 2: Time snapshots of a lab experiment run where the densely connected minority group that preferred red exerted strong influences on the blue-leaning majority. The minority group eventually succeeded in converting all the initial (unfilled) blue votes to (filled) red votes.

tion for analysis of adaptive strategies and theoretical constraints on convergence [10].

4. MODELING DYNAMIC VOTING

We present four model forms designed to capture voting behavior dynamics in the dynamic consensus experiments. All are expressible as hGMMs. Only the first exploits the flexibility of hGMMs to express dependence of actions within a neighborhood given history (2), hence we refer to this as the *joint behavior consensus model* (JCM).

The other three forms model agent behaviors individually: for each agent we specify a probabilistic strategy $\sigma_i(H^t) = \Pr(a_i^t \mid H^t_{\Gamma_i})$. 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:

$$\Pr(a^t \mid H^t) = \prod_i \sigma_i(H^t). \tag{3}$$

We refer to a dynamic multiagent model expressible by (3) 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 generally introduces correlations among agents actions, even if they are independently generated on full history [4]. Nevertheless, assuming conditional independence between agents' actions given history exponentially reduces the model's complexity, or more specifically, the representational complexity of the joint probability distribution over the system's actions.

The first of three IBMMs we present is designed as an independent behavior version of the JCM; thus, we call it simply the *individual behavior consensus model* (ICM). The remaining two models are based on proposals and observations from the original experimental analysis [8], and are labeled *proportional response model* (PRM) and *sticky proportional response model* (sPRM), respectively.

4.1 Joint Behavior Consensus Model

Based on observations from the original experiment analysis, we seek to formulate a potential function for JCM that captures the impact of past collective choices of i's neighborhood, i's own past voting patterns, and its relative preference for each action.

First, we consider how to summarize a history $H^t_{\Gamma_i}$ of length h relevant to agent i. Let indicator $I(a_i, a_k) = 1$ if $a_i = a_k$ and 0 otherwise. We define $f(a_i, H^t_{\Gamma_i})$ as the frequency of action a_i being chosen by other agents in i's conditioning set, which by definition contains nodes whose past influence how i chooses its action in the present,

$$f(a_i, H^t_{\Gamma_i}) = \frac{\sum_{k\in\Gamma_i\setminus\{i\}} \sum_{\tau=t-h}^{t-1} I(a_i, a_k^\tau) + \epsilon}{h|\Gamma_i \setminus \{i\}|}. \tag{4}$$

We add $\epsilon = 0.01$ to the numerator to ensure that the frequency term does not vanish when a_i does not appear in $H^t_{\Gamma_i}$.

Second, we capture agent i's own update history in an *inertia* term,

$$\mathcal{I}(a_i, H_i^t) = \begin{cases} t - \max_{\tau<t} \tau(1 - I(a_i^\tau, a_i)) & \text{if } a_i = a_i^{t-1} \\ [t - \max_{\tau<t} \tau(1 - I(a_i^\tau, a_i))]^{-1} & \text{otherwise} \end{cases}$$

In other words, we model agent i's voting inertia as proportional to how long it has maintained its most recent action a_i^{t-1}. We treat inertia as a factor tending to support the candidate of retaining the same action. If the candidate action a_i is different from a_i^{t-1}, the term expresses the inverse of this inertia factor.

Third, we define $r_i(a_{N_i})$ as the product of $r_i(a_i)$ and a heuristic attenuation based on how many nodes in the within-time neighborhood currently vote differently:

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

where $\alpha \in [0, 1]$. In our study, we set $\alpha = 0.9$. Observe that $r_i(a_{N_i})$ as defined is increasing in the number of i's neighbors voting a_i, reflecting the positive influence of neighbor choices on i.

The potential function for agent i combines these terms,

$$\pi_i(a_{N_i} \mid H^t_{\Gamma_i}) = r_i(a_{N_i}) f(a_i, H^t_{\Gamma_i})^\gamma \mathcal{I}(a_i, H_i^t)^\beta, \tag{5}$$

where $\gamma, \beta \geq 0$ denote the weight or importance of the historical frequency $f(a_i, H^t_{\Gamma_i})$ and the inertia $\mathcal{I}(a_i, H_i^t)$ relative to the estimated reward $r_i(a_{N_i})$. The normalized product of these potentials specifies joint behavior as described in (2). The model maintains two free parameters β and γ.

4.2 Individual Behavior Consensus Model

The ICM for dynamic consensus retains the main elements of JCM (5), while imposing conditional independence among agents' actions given the common history. The result is a within-time neighborhood N_i that contains only i itself for

each i. The probabilistic ICM behavior is then given by:

$$\Pr(a_i \mid H^t_{\Gamma_i}) = \frac{1}{Z_i} r_i(a_i) f(a_i, H^t_{\Gamma_i})^{\gamma} \mathcal{I}(a_i, H^t_i)^{\beta}.$$

The normalization ranges only over single-agent actions $a_i \in A_i$, thus Z_i is easy to compute for this model.

4.3 Proportional Response Model

We also consider for comparison the proportional response model, PRM, suggested in the original dynamic consensus study [8] 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 $r_i(a_i)$ and $g(a_i, a^{t-1}_{\Gamma_i})$, the number of i's neighbors who chose a_i in the last time period,

$$\Pr(a_i \mid H^t_{\Gamma_i}) \propto r_i(a_i) g(a_i, a^{t-1}_{\Gamma_i}).$$

4.4 Sticky Proportional Response Model

PRM does not capture the subjects' tendency to start with their preferred option, reconsidering their votes only after collecting additional information about their neighbors over several time periods [8]. Therefore, we introduce the *sticky proportional response model*, sPRM, which contains a parameter $\rho \in [-1, 1]$ reflecting an agent's stubbornness in maintaining its preferred option, regardless of observed neighbors' past choices. Intuitively, an agent's inherent bias toward its preferred option decays proportionally until there is no bias:

$$\Pr(a_i \mid H^t_{\Gamma_i}) \propto r_i(a_i) g(a_i, a^{t-1}_{\Gamma_i})(1 + \frac{I^{\max}_{a_i} \rho}{t}),$$

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

5. LEARNING

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 to find 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 all model forms, except for the PRM which contains no parameters, in our study.

5.2 Structure Learning

Each of the consensus voting experiments involves 36 human subjects. 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 [1] on these models, where $N = \Gamma = N^{\mathsf{O}}$, indicated that this computational saving would still be insufficient for effective learning. Thus, we need to employ models with simpler graphs in order to take advantage of hGMMs' 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, there is no inherent reason that the observation graph should constitute the ideal structure for a predictive graphical model for agent behavior. In other words, the most effective N is not necessarily the same as N^{O}. Since actual agent behavior is naturally conditioned on its observable history, we do assume that the *conditioning set* coincides with the observation neighborhood, $\Gamma = N^{\mathsf{O}}$. Nevertheless, 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 that defines each i's within-time neighborhood may provide interesting insights on the agents' networked behavior.

Our structure learning algorithm addresses the problem of learning N_i for every i, taking $\Gamma_i = N^{\mathsf{O}}_i$ as fixed. Note that the IBMMs described in Section 4 impose $N_i = \{i\}$ for each i, and thus do not need to learn the within-time graphs. Starting from an empty graph, we greedily add edges to improve the log-likelihood of the training data, subject to a constraint that the maximum node degree not exceed a specified bound $d_{\max}$. Since the set of edges E is the only structural model feature that changes during our search, we use $L_E(X; \theta)$ to abbreviate $L_{hG}(X; \theta)$ as induced by the hGMM $hG = (V, E, A, \Gamma, \pi)$. We have found that the optimal settings of our parameters $\theta = (\beta, \gamma)$ is insensitive to within-time dependencies, hence we apply the parameter learning operation (Section 5.1) only once, at the beginning of our search. The algorithm is defined formally below.

$E \leftarrow \emptyset$
Use gradient descent to identify $\theta \approx \arg\max L_E(X; \theta)$.
$\tilde{E} \leftarrow \{(i, j) \mid i \in V, j \in V\}$
repeat
    newedge $\leftarrow$ **false**
    $(i_*, j_*) \leftarrow \arg\max_{(i,j) \in \tilde{E}} L_{E \cup (i,j)}(X; \theta)$
    if $L_{E \cup (i_*, j_*)}(X; \theta) \geq L_E(X; \theta)$ **then**
        $E \leftarrow E \cup \{(i_*, j_*)\}$
        newedge $\leftarrow$ **true**
    end if
    $\tilde{E} \leftarrow \tilde{E} \setminus \{(i_*, j_*)\} \setminus \{(i, j) \mid \max(|N_i|, |N_j| = d_{\max}\}$
until $\tilde{E} = \emptyset \vee$ *newedge* = **false**

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 negative 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 JCMs in comparison with ICMs, PRMs, and sPRMs, using the dy-

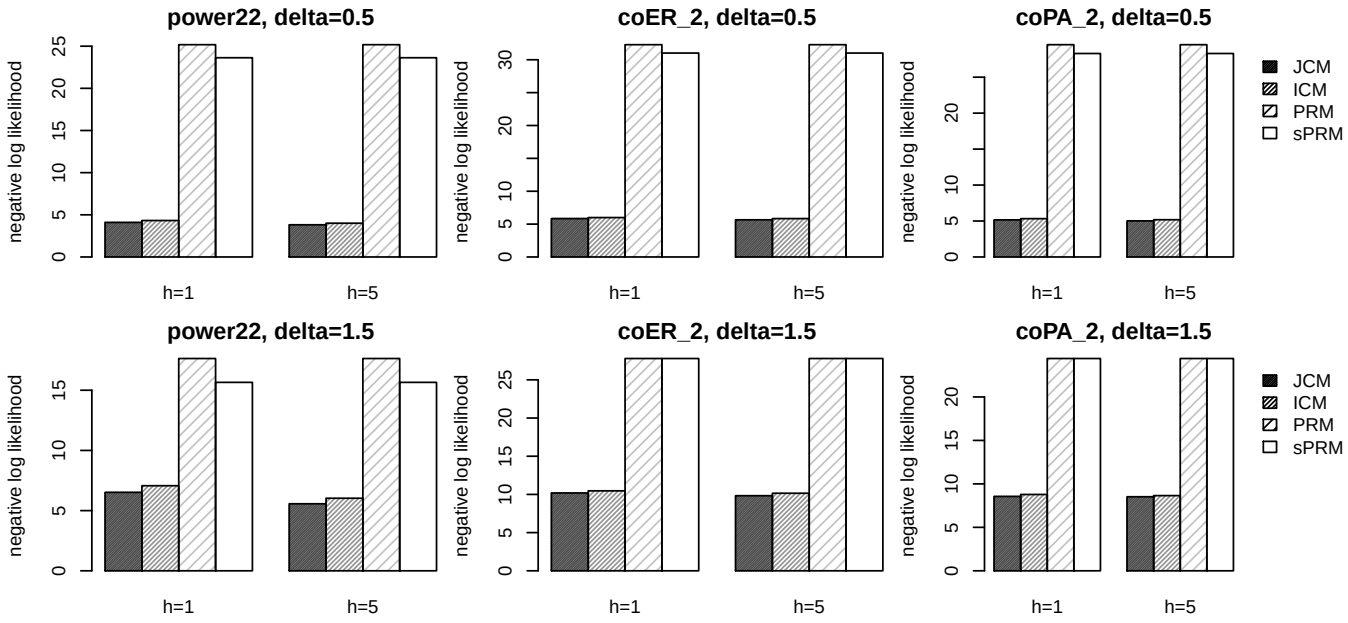

Figure 3: JCMs provide better predictions of the system's dynamics than ICMs, PRMs, and sPRMs in twelve settings: the three experiment networks power22 (left), coER_2 (middle), and coPA_2 (right), each for two history lengths, using time discretization intervals $\delta = 0.5$ (top) and $\delta = 1.5$ (bottom). The prediction quality differences between JCM and ICM are significant ($p < 0.025$) in all scenarios.

namic consensus experiment data [8]. We also examine the graphs induced by structure learning, and relate them to the corresponding observation networks by various statistical measures.

6.1 Experiment Settings

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 [8]. In particular, networks whose edges are generated by a random Erdos-Renyi (ER) process have a notably more heavy-tailed degree distribution than those generated by a preferential attachment (PA) process. 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 subjects 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 subjects' votes at the end of each time interval of length δ seconds. Our experiments examine interval lengths $\delta \in \{0.5, 1.5\}$.

In our study, we learn predictive models for each experiment 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 original study findings [8]. In each experiment set, we use eight of the nine trials for training the predictive models for each form. The within-time graphs are learned with node degree constraint $d_{\max} = 10$. We then evaluate the models based on their predictions over a test set comprising the left-out experimental trial. This process is repeated five times, with a different randomly chosen trial reserved for testing. Each data point in our reported empirical results averages over these five repetitions.

Using the original experiment labels, we distinguish three experiment networks according to their graph generator processes and the existence of a minority group of well-connected nodes that share the same vote preference (see 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

6.2 Predictions

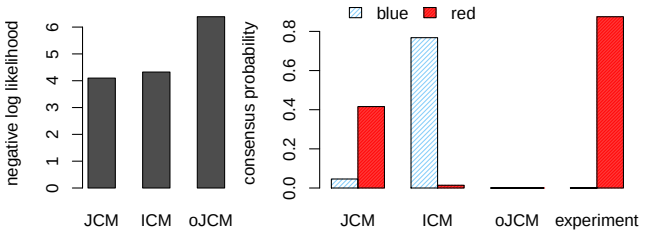

Figure 5: oJCMs provide worse predictions than JCMs and ICMs for both the system's dynamics and end-game results (power22, $h = 1$ and $\delta = 0.5$).

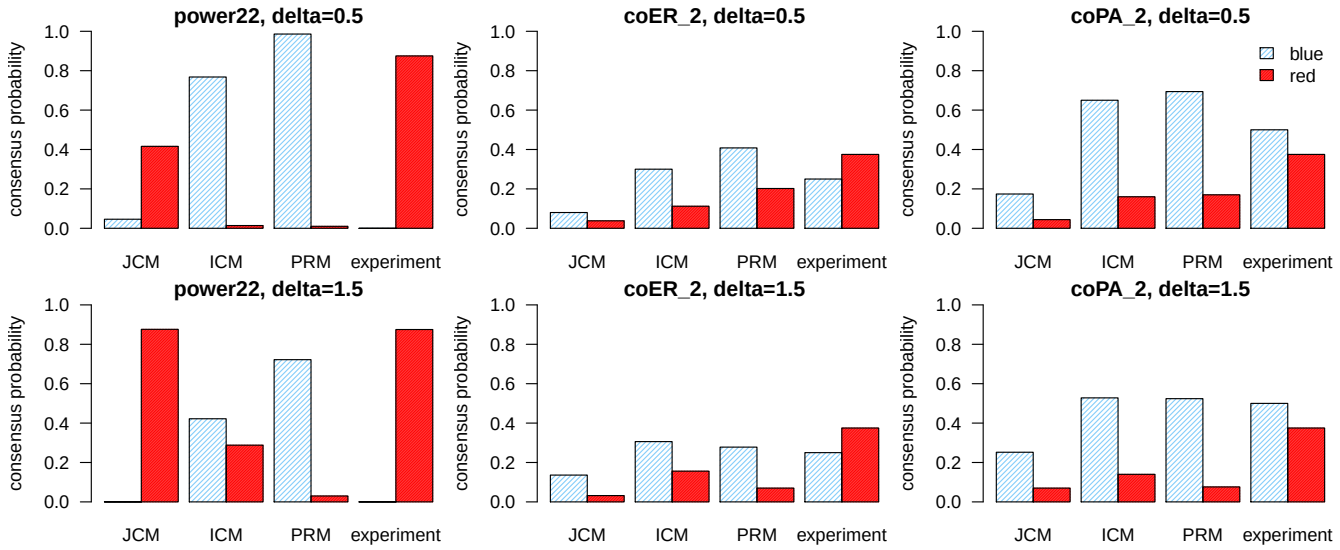

Figure 4: JCM predictions on the probability of reaching consensus are lower than predictions from ICMs and PRMs, as well as experiment outcomes. However, the JCM is significantly more accurate than ICMs or PRMs on predicting the ultimate consensus colors.

We first examine predictions of subjects' votes in each time period conditional on available history. A comparison of four models on twelve scenarios is presented in Figure 3. We measure predictive performance by negative log-likelihood of the test data, according to the respective models. JCMs perform better than ICMs, PRMs, sPRMs, in predicting dynamic behavior in the dynamic consensus experiments for all three experiment settings, given data discretized at interval lengths of 0.5 and 1.5 (differences significant at $p < 0.025$). Both the JCM and ICM representations, which share similar fundamental elements, handily outperform PRM and its sticky version sPRM.

Contrary to the expectation that the less historical information a model uses, the lower its prediction performance, JCMs and ICMs that employ only the last $h = 1$ period of historical data generate similar predictions to those with $h = 5$. This phenomenon is likely a consequence of the heuristic nature of the frequency function (4), and moreover may indicate that some human subjects take into account only a short history of their neighbors' actions when choosing their own actions. All models perform worse with the larger time interval $\delta = 1.5$, which is unsurprising in that the coarser discretization entails aggregating data. More salient is that the results are qualitatively identical for the two δ settings, further illustrating the robustness of our findings. These results in general demonstrate JCMs' ability to capture joint dynamic behavior, especially behavior interdependencies induced by limited historical information, as opposed to the IBMM alternatives.

We next evaluate the models' ability 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 a simulation 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 100 run instances for each environment setting and model. As we do not observe any considerably qualitative differences in the models' end-game predictions for different history lengths h, we choose to display only results for $h = 1$ henceforth.

The proportion of simulation instances reaching consensus induced by ICMs and PRMs correlates with observed experiment results, as shown in Figure 4.[1] Simulated runs drawn from JCMs converge to consensus at lower rates than in ICMs, PRMs, and human-subject experiments in general. However, their end-game predictions improve with greater $\delta = 1.5$, especially in the power22 setting where JCMs predict the experiment outcomes almost exactly. A closer look at the end-game prediction results reveals a different picture about the relative performances of the three models. In particular, the individual behavior models' predictions on the final consensus color are considerably out of line with the actual experiments for both coER_2 and power22, rendering them ineffective in predicting end-game color results. JCMs, on the other hand, provide significantly more accurate predictions on the consensus color in the power22 setting. The ratio between blue and red consensus instances by JCMs in coPA_2 resembles that of the actual experiments more than ICMs and PRMs. In the coER_2 setting all models' predictions on the favored consensus color (blue) miss the actual experiments' favored consensus color (red), though the ratio of red-to-blue consensus predicted by JCM is less skewed than that of ICMs and PRMs.

Last, we demonstrate the benefits of our extension to the original hGMM representation by comparing the JCM representation against oJCM, which retains the original hGMM definition, assuming that the conditioning set is identical to the learned within-time neighborhood: $\Gamma = N$. Figure 5 shows that oJCMs perform worse than both JCMs and ICMs in predicting the system's votes for each time period and end-game results, for the power22 setting with $h = 1$ and

[1]End-game results from sPRMs are similar to those from PRMs, and not shown here.

$\delta = 0.5$.[2] Moreover, we note that the resulting graphs by oJCMs contain disconnected node subsets, which potentially prevent vote decisions to propagate throughout the network, causing failures in producing any consensus instances.

6.3 Graph Analysis

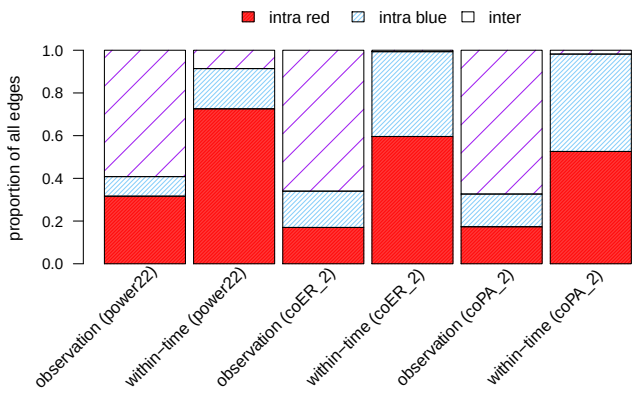

Figure 6: Distributions of edges from three different categories, intra red, intra blue, and inter, in the given observation and learned within-time graphs for JCM ($\delta = 0.5$).

In this section, we seek to characterize the learned edges that define N in the JCM representation, and discover connections between the learned graphs and the aforementioned prediction results. First, we categorize edges by their endpoint nodes' vote preferences: we refer to edges that connect two red (blue) nodes as *intra* red (blue), and those between red and blue nodes as *inter* edges. Figure 6 presents the proportion of each edge type in both the given observation graphs and the learned within-time graphs. While a majority of edges in the observation graphs are inter edges, the within-time graphs that define N consist mostly of intra edges. That is, there are more interdependencies in JCMs among agents of the same preference than among conflicting agents. The ability to discover these inter edges and incorporate the information they carry in its joint action distribution may help the JCM representation to better capture dynamic behavior and end-game results, as illustrated and discussed in Section 6.2. For the power22 setting in particular, JCMs often assign a majority of edges as intra red, and thus effectively identify the presence of a strongly connected red minority who dictated end-game colors in the actual experiments. This construction allows JCMs to predict end-game consensus colors much more accurately than ICMs and PRMs, which rely entirely on the observation graphs.

We further investigate whether these proportion measures provide any predictions on the number of consensus instances induced by JCMs. We pool data from the three experiment settings—power22, coPA_2, and coER_2—and compute a simple linear regression of the number of red (blue) consensus instances with respect to the proportion of intra red (blue) edges. The resulting regression coefficients are statistically significant for both blue and red ($p < 0.05$).

[2]We also obtain similar results for oJCMs in other experiment settings and environment parameters, which are not shown here.

Figure 7 suggests that a weak positive correlation between the within-time graphs' intra edges and the number of consensus instances. Intuitively, more interdependence between same-preference nodes allows them to have more influence on one another, helping to diffuse vote choices more rapidly throughout the system.

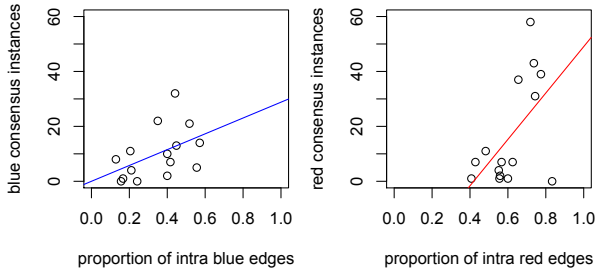

Figure 7: The number of consensus instances in blue (left) and red (right), and proportion of JCM intra edges of the corresponding colors.

We next examine JCM edges in terms of how far apart are the nodes they connect in the observation graph. Let $\phi_{i,j} \geq 1$ denote the length of shortest path from i to j in the observation graph. Given a graph G on the same set of nodes, we can calculate the proportion of edges in G that connect nodes separated by a certain distance in the original observation graph. Figure 8 presents the profile of such distances for pairs of nodes in the learned JCMs. For comparison, the profiles labeled "fully connected" simply reflect the distribution of node distances in the original observation graph: most of the nodes are one hop or less apart from each other ($\phi \leq 2$), and the modal distance is $\phi = 2$. A large majority of edges in the learned within-time graphs have $\phi = 2$, that is, are close but not connected in the observation graphs.

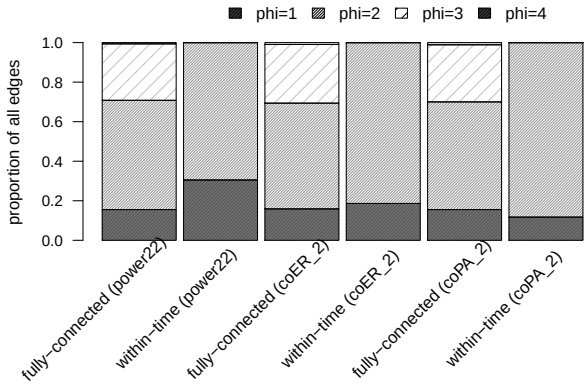

Figure 8: Distributions of edges in the within-time graphs based on the distance between their end-nodes ϕ in the observation graph ($\delta = 0.5$).

Next we compare the *assortativity* [12] of the learned and original graphs. A graph G's assortativity coefficient in $[-1, 1]$ captures the tendency for nodes to attach to others that are similar (positive values) or different (negative values) in connectivity. As illustrated in Figure 9, the large difference in assortativity for the power22 setting stresses the

JCM's ability to discover interdependencies among agents' actions that are not captured in the observation graph. In particular, the resulting JCMs are able to capture action correlations among nodes of similar degrees in the power22 setting, where the minority nodes are more densely connected than the majority, confirming the findings by aforementioned graph analyses on intra and inter edges. We also investigate the sparsity of the learned graphs for different values of δ. Our sparsity measure is the number of edges in the learned within-time graph divided by the number of edges in the corresponding observation graph. Figure 9 illustrates that the within-time graphs become sparser as the discretization interval shrinks from 1.5 to 0.5 in all experiment settings. Intuitively, the finer grain the discretization is, the fewer simultaneous vote changes there are in one time period. As a result, there may be fewer interdependencies among agents' actions, which explains the aforementioned relations between discretization interval and graph sparsity across all experiment settings.

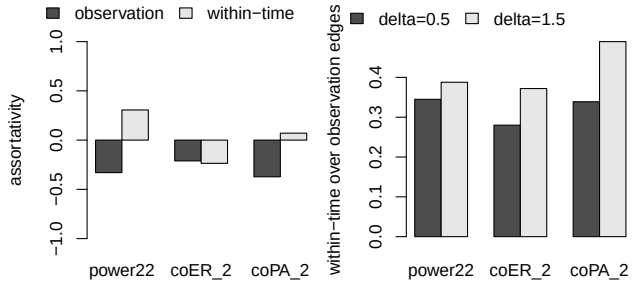

Figure 9: (left) Assortativity of the observation graphs and the learned within-time graphs ($\delta = 0.5$). (right) Sparsity of the within-time graphs.

7. CONCLUSIONS

Our main result is a demonstration of the feasibility of learning probabilistic models of dynamic multiagent behavior from real traces of agent activity on a network. To accomplish this we extend the original hGMM framework [4], by distinguishing within-time dependencies from conditioning sets, and introducing a structure-learning algorithm to induce these dependencies from time-series data. We evaluated our techniques by learning compact graphs capturing the dynamics of human-subject voting behavior on a network. Our investigation finds that the learned joint behavior model provides better predictions of dynamic behavior than several individual behavior models, including the proportional-response models 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 within-time graphs that provide insights about patterns of agent interdependence, and their relation to structure of the agent interaction network.

We plan to improve the learning algorithm for individual behavior models, by replacing the maximum-degree constraint with a cross-validation condition that can better help avoid over-fitting. Given the formalism's generality, we consider it promising to apply our modeling technique to similar problem domains, such as graph coloring, where agents must coordinate their actions or make collective decisions while only communicating with their neighbors, as well as large network scenarios, such as social networks and Internet protocols.

8. REFERENCES

[1] J. Y. Broadway, J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Thirteenth Annual Conference on Advances in Neural Information Processing Systems*, pages 689–695, Denver, 2000.

[2] C. Daskalakis and C. 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.

[3] Q. Duong, M. P. Wellman, and S. Singh. Knowledge combination in graphical multiagent models. In *Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 153–160, Helsinki, 2008.

[4] Q. Duong, M. P. Wellman, S. Singh, and Y. Vorobeychik. History-dependent graphical multiagent models. In *Ninth International Conference on Autonomous Agents and Multiagent Systems*, pages 1215–1222, Toronto, 2010.

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

[6] A. X. Jiang, K. Leyton-Brown, and N. A. R. Bhat. Action-graph games. *Games and Economic Behavior*, 71:141–173, 2010.

[7] S. Kakade, M. Kearns, J. Langford, and L. Ortiz. Correlated equilibria in graphical games. In *Fourth ACM Conference on Electronic Commerce*, pages 42–47, San Jose, CA, 2003.

[8] M. Kearns, S. Judd, J. Tan, and J. Wortman. Behavioral experiments on biased voting in networks. *Proceedings of the National Academy of Sciences*, 106(5):1347–1352, 2009.

[9] M. Kearns, M. L. Littman, and S. Singh. Graphical models for game theory. In *Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 253–260, Seattle, 2001.

[10] M. Kearns and J. Tan. Biased voting and the Democratic primary problem. In *Fourth International Workshop on Internet and Network Economics*, pages 639–652, Shanghai, 2008.

[11] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45:181–221, 2003.

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