History-Dependent Graphical Multiagent Models

Quang Duong* Michael P. Wellman* Satinder Singh* Yevgeniy Vorobeychik†

*Computer Science & Engineering
University of Michigan
Ann Arbor, MI 48109-2121 USA

†Computer & Information Science
University of Pennsylvania
Philadelphia, PA 19104 USA

ABSTRACT

A dynamic model of a multiagent system defines a probability distribution over possible system behaviors over time. Alternative representations for such models present tradeoffs in expressive power, and accuracy and cost for inferential tasks of interest. In a *history-dependent* representation, behavior at a given time is specified as a probabilistic function of some portion of system history. Models may be further distinguished based on whether they specify individual or joint behavior. Joint behavior models are more expressive, but in general grow exponentially in number of agents. *Graphical multiagent models* (GMMs) provide a more compact representation of joint behavior, when agent interactions exhibit some local structure. We extend GMMs to condition on history, thus supporting inference about system dynamics. To evaluate this *hGMM* representation we study a voting consensus scenario, where agents on a network attempt to reach a preferred unanimous vote through a process of smooth fictitious play. We induce hGMMs and individual behavior models from example traces, showing that the former provide better predictions, given limited history information. These hGMMs also provide advantages for answering general inference queries compared to sampling the true generative model.

Categories and Subject Descriptors

I.2.II [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Experimentation, Algorithms, Performance

Keywords

Multiagent reasoning, graphical multiagent models, agent-based modeling

1. INTRODUCTION

Multiagent systems research has produced many forms of models for dynamic multiagent behavior, taking into account a broad range of factors. Generative models (e.g., based on rules for individual agent behavior) are highly expressive, offering virtually unlimited capacity for specifying any dynamic behavior the modeler may conceive. Reasoning about the properties of such models (e.g., for prediction based on partial observation), however, is often computationally challenging, requiring in general a prohibitive amount of sampling of the possible system trajectories. Analytic models, in contrast, may facilitate more efficient reasoning, but often at the cost of strong limits on the complexity of the agent behavior and environments that can be represented.

One source of complexity in multiagent modeling is the interdependence of agent behaviors due to interactions among the agents. Even if the agents are autonomous (i.e., make decisions independently), conditioning their decisions on common observations (including each others' actions) induces correlation of their behaviors over time. Generative models typically specify agent behavior individually, exploiting the conditional independence of their actions given system history. From the modeler's perspective, however, we are often interested in properties of system behavior given only an incomplete view of this history. In this case, agent decisions are not independent, and we may wish to directly express joint behavior.

Direct representation of joint behavior may require space exponential in the number of agents, hence we seek structure that enables more compact specification. Approaches based on *graphical models* achieve this by exploiting conditional independence among model elements. In the context of agents, this observation has led to numerous models that exploit locality of agent interaction, including: multiagent influence diagrams (MAIDs) [Koller and Milch, 2001], graphical game models [Kearns et al., 2001], action-graph games [Jiang et al., 2008], and networks of influence diagrams [Gal and Pfeffer, 2008]. By mapping graphical games to Markov random fields, Daskalakis and Papadimitriou [2006] showed that statistical inference tools for graphical models can be applied for game-theoretic computation on graphically structured multiagent scenarios. The graphical multiagent model (GMM) framework [Duong et al., 2008] generalizes this approach beyond game-theoretic reasoning by allowing beliefs about agent behavior to be based on a variety of knowledge sources.

To enable inference about system dynamics, we extend the static GMM representation to condition on history, yielding *history-dependent graphical multiagent models* (hGMMs). By directly specifying joint behaviors, hGMM can capture correlations in agent actions without full specification of the

Cite as: History-Dependent Graphical Multiagent Models, Quang Duong, Michael P. Wellman, Satinder Singh, Yevgeniy Vorobeychik, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. XXX-XXX.

state history mediating agent interactions. Moreover, hGMMs provide a compact representation for domains with decomposable structure and consequent computational savings. Like GMMs, hGMMs accommodate a variety of sources of knowledge for multiagent modeling, and support general learning and inference facilities of graphical models.

We demonstrate the representational power of hGMMs and value of modeling joint behaviors, by comparison, to individual behavior models in a voting consensus scenario studied by Kearns et al. [2009]. To evaluate the models, we postulate that actual behavior is generated by a smooth fictitious play process. Since the generative fictitious play simulation (our ground truth) is an individual behavior model where agents update their beliefs and choose their responses independently given history, the study evaluates the ability of hGMMs to model behavior in scenarios with limited historical information. We find experimentally that hGMMs provide better predictions of agent actions than individual behavior multiagent models, given limited history. Similar results emerge when game play data are generated by an asynchronous model where agents may change their actions at any time, and when inference computations in hGMMs are approximate. Furthermore, hGMMs outperform generative models that sample from the true model in answering queries about the game's outcomes.

We motivate our study with an overview and discussion of the voting consensus game in Section 2. Section 3 describes the general problem of modeling dynamic multiagent behavior. In Section 4, we review the GMM framework and introduce our history-dependent extension. We develop specific hGMMs and IBMMs for the voting consensus game in Section 5, and employ these in an empirical study (Section 6) designed to evaluate their relative performance in various settings. We conclude with some observations on these results and future directions in Section 7.

2. EXAMPLE: THE VOTING CONSENSUS GAME

We illustrate the problem of representing multiagent behavior with a voting consensus scenario introduced and studied by Kearns et al. [2009]. The situation can be modeled as a game played on a network. Each agent (player) has two available actions (vote options), labeled 0 and 1. The scenario terminates when all agents' votes agree, or at the time limit T if no consensus is reached by then. In the asynchronous version of the scenario, agents may change their votes at any time, until the termination condition is reached. We also consider a discrete-time version, where opportunities to update votes occur at a finite number of points. Upon termination, each agent i receives an individual reward $r_i(a) > 0$ if everyone converges on action a, and nothing otherwise. The variation in reward functions by agent reflects differing relative preferences for the available options. Despite these preference differences, the agents have a common interest in achieving consensus, as without a unanimous vote nobody gets a reward.

Another pivotal feature of the voting consensus scenario is that agents have limited knowledge of the others' current votes. Specifically, the agents are connected in a graph structure, and can observe the votes of only their neighbors in the graph. In addition, agent i knows only its local graph structure, namely its neighbors N_i, the degree of each neighbor $k \in N_i$, and edges between its neighbors.

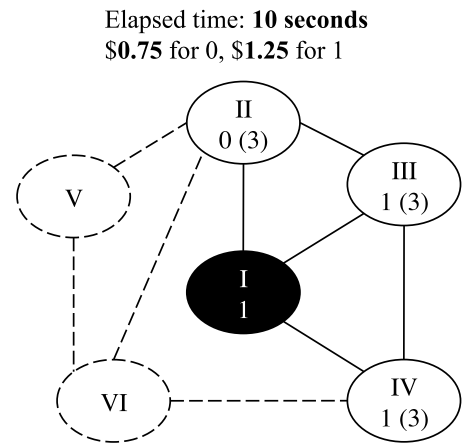

Figure 1: A typical game experiment from a player's perspective [Kearns et al., 2009].

Figure 1 shows what a player observes during a typical experiment. We display the game from the perspective of player I, whose payoffs are \$0.75 and \$1.25 for 0 and 1 consensus outcomes, respectively. Player I can observe only the agents to which I is connected in the graph—namely II, III, and IV, but not V and VI, shown in the dotted portion of the graph. In particular, the player knows each neighbor's degree (the number shown in parenthesis), the connections among its neighbors, and their current votes of either 0 or 1.

This scenario raises several interesting questions for agent behavior, including:

- How should agents balance their efforts to promote their own preferred outcome against the imperative to achieve some consensus?
- In pursuing their goals, how should agents take into account observed voting patterns of neighbors, and their partial knowledge of network structure?

Alternatively, rather than ask how agents *should* behave, we could pose questions about how agents *do* behave. We may be interested, for example, in the behavior of human agents, or of artificial agents constructed in various ways or induced by specified learning processes.

Kearns et al. [2009] conducted a series of human-subject experiments to collect data about how human agents behave in 81 different instances of the voting consensus game. By varying preference assignments and network structure, they gathered evidence about the effect of these factors on strategies employed, and the consequent voting results. 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].

3. MODELING DYNAMIC MULTIAGENT BEHAVIOR

To formalize the general problem, we consider a scenario with n agents, behaving over an interval of discrete time periods, $[0, \ldots, T]$. At each time period t, agent $i \in \{1, \ldots, n\}$ chooses an action a_i^t from its domain of available actions, A_i. These choices may in general depend on i's knowledge and observations, as specified by the agent's *strategy*, σ_i. In this work, we focus on an agent's observations of past actions by itself and others, as captured by *history* H^t up to time t. As the capacity of agents' memory is often limited, H^t includes information only for a fixed *horizon* h. Consequently, we write $H^t = \{(a_1^k, \ldots, a_n^k) \mid k \in [t-h, t)\}$. Furthermore, to allow that an agent may ignore some past actions of others, we denote by H_i^t the subset of history relevant to i's probabilistic choice of next action: $a_i^t \sim \sigma_i(H_i^t)$.

Let us examine the voting game example illustrated in Figure 1. Assume that the length of each discrete time period is one second, and players have a very short time horizon, $h = 1$. At time period $t = 10$, player I's history H_I^{10} consists of $\{(a_I^9 = 1, a_{II}^9 = 0, a_{III}^9 = 1, a_{IV}^9 = 1)\}$. One possible strategy is to vote according to the majority from the last time period, in which case $a_I^{10} = majority(H_I^{10}) = 1$.

3.1 Sources of Multiagent Models

Research in multiagent systems has encompassed many approaches to modeling agent behavior. Factors based on social norms, preferences, motives, conventions, authority structures, beliefs about others, and more have been incorporated in multiagent models. Different approaches can be characterized to some extent based on the factors they emphasize, and their underlying assumptions of agent capabilities and tendencies. For example, *game-theoretic models* fully specify agents' beliefs and preferences, and assume agents act rationally according to these attitudes. Within this framework, some modelers postulate adaptive learning processes that can lead agents to adopt game-theoretic solutions [Fudenberg and Levine, 1998]. Others may employ dynamic models based on reinforcement learning, irrespective of game-theoretic interests, or adopt behaviors generated by alternative dynamics, for example based on evolutionary models [Tuyls and Parsons, 2007] or combinations [Hennes et al., 2009]. Yet other models focus on particular decision contexts (e.g., social networks, organizations, markets, teams), and develop behavior rules that are justified by normative or descriptive theories for those contexts.

Our purpose here is not to survey the full range of multiagent modeling approaches, much less to comparatively evaluate them. Rather, we seek a common representational framework that can accommodate a broad class of behavioral models, and support inference and learning tasks based on such models. We start from the observation that ultimately any of these models define a system, where stochastic agent strategies induce a probability distribution over dynamic behaviors of the multiagent system. We draw a central distinction between representations based on specifying *individual* agent behavior, and those based on *joint* behavior.

3.2 Individual Behavior Multiagent Models

Models of agent behavior on social networks provide some of the clearest examples for representing multiagent scenarios as stochastic dynamic systems. For example, Granovetter [1978] specifies an agent's strategy as a threshold function over the choices of other agents. This model has been generalized by Kleinberg [2007], and extended to model the cumulative effect over time of an agent's neighbors' actions on its own behavior. For the voting consensus game of Section 2, a simple multiplicative model that combines vote preferences with trends in neighbor actions can provide an effective probabilistic model of subject behavior [Kearns et al., 2009, Kearns and Wortman, 2008]. In a study of related coordination games, Shoham and Tennenholtz [1997] propose a rule whereby agents choose the action that maximizes their cumulative reward over a recent interval, and characterize the convergence to conventional behavior. Information about other agents' historic actions has also been shown instrumental in predicting learned behavior in a simple repeated game [Mookherjee and Sopher, 1994].

Though varied in complexity and approach, the above models share one common feature: the behavior of the multiagent system is the product of separately specified individual behaviors. That is, for each agent we specify a probabilistic strategy $\sigma_i(H_i^t) = \Pr(a_i^t \mid H_i^t)$. In such a formulation, agent interactions are captured by the conditioning of individual behavior on commonly observed history. The agents' actions are probabilistically dependent, but conditionally independent given this common history, yielding the joint distribution

$$\Pr(a^t \mid H^t) = \prod_i \sigma_i(H_i^t). \tag{1}$$

We refer to a dynamic multiagent model expressible by (1) as an *individual behavior multiagent model* (IBMM). IBMMs provide an intuitive form for expressing collective behavior as a function of individuals. Their key premise is that the common observed history is sufficient to capture correlations in agent behavior. Although this may be a valid assumption in principle, it is usually infeasible to condition on the entire history in a practical specification of agent behavior.

3.3 Joint Behavior Multiagent Models

Without the assumption of conditional independence given history, we generally need to specify the joint behavior $\Pr(a^t \mid H^t)$ of multiple agents directly. This approach quickly becomes intractable, as the size of such a specification grows exponentially in the number of agents. However, if the interactions among agents can be localized (given history), we may be able to exploit this structure to achieve a more compact representation of the joint distribution. That is the role of graphical multiagent models [Duong et al., 2008], discussed in Section 4.1 below.

3.4 Inference in Multiagent Models

Whether the model is fundamentally specified in terms of individual or joint behaviors, its value lies in its amenability to analysis. Analysts may seek, for example, to establish general properties of a model (e.g., features common to all runs, or statistically prevalent in the distribution of runs), identify anomalous behavior, predict future behavior given a prefix of observations, or explain interesting behavioral phenomena. Many if not all analytical tasks can be decomposed into basic queries about the probability of some features of a behavior run, conditional on some given features. For example, in the voting game described in Section 2, we may want to capture a subject's "stubbornness" by $\Pr(a_i^t|H_i^t)$, the probability of that subject's voting 0 given that her neighbors have been voting 1 in the last 10 time periods. Observing that after the first half of each experiment run two subjects i and j appear to coordinate in their actions, for instance, may lead us to compute $\Pr(a_i^t, a_j^t | H^t)$. We may also be interested in predicting the future votes given past observations, which prompts us to calculate the joint probability distribution $\Pr(a^t | H^t)$.

4. HISTORY-DEPENDENT GMMS

We describe graphical multiagent models [Duong et al., 2008], and extend them to incorporate history for reasoning about agent behavior in dynamic multiagent scenarios.

4.1 Graphical Multiagent Models

Consider a special case of the aforementioned multiagent scenarios where the system's time horizon T is 1. The system's final outcome is a joint action specifying the action choice of all players. A *graphical multiagent model* (GMM) for this scenario is a graphical model, $G = (V, E, A, \pi)$, with vertices $V = \{v_1, \ldots, v_n\}$ corresponding to the agents (we refer to v_i and i interchangeably), and edges $(i, j) \in E$ indicating a local interaction between i and j [Duong et al., 2008]. The graph defines for each agent a neighborhood, $N_i = \{j \mid (i, j) \in E\} \cup \{i\}$, including i and its neighbors $N_{-i} = N_i \setminus \{i\}$. Each neighborhood i is associated with a potential function $\pi_i(a_{N_i}) : \Pi_{j \in N_i} A_j \to \mathbb{R}^+$. Intuitively a local configuration of actions with a higher potential is more likely to be part of the global outcome than one with lower potential.

The size of the GMM description is exponential only in the size of local neighborhoods rather than in the total number of players. Each neighborhood corresponds to a clique in the triangulated graphical model, where additional links are introduced to form sets of maximally connected subgraphs. As a result, we can factor the joint distribution into neighborhood potentials [Daskalakis and Papadimitriou, 2006]:

$$\Pr(a) = \frac{\Pi_i \pi_i(a_{N_i})}{Z}, \tag{2}$$

where Z is the normalization term.

GMMs provide a flexible representation framework for *static* graphically structured multiagent scenarios that support the specification of probability distributions over joint actions based on game-theoretic models as well as heuristic or other qualitatively different characterizations of agent behavior [Duong et al., 2008]. For a dynamic multiagent system, GMMs may represent steady state distributions or a prediction of final joint outcomes, but since the potential functions reference a single time step, GMMs as originally formulated cannot express probabilistic behavior trajectories over time.

4.2 History-Dependent GMMs

A *history-dependent graphical multiagent model* is also a graphical model, $hG = (V, E, A, \pi)$, with V, E, and A defined just as in the original GMM framework. The essential extension is in the potential functions, π, which for hGMMs are a function of history. Specifically, hGMMs capture agent interactions by conditioning joint agent behavior on an abstracted history of actions H^t. The abstracted history available to agent i, denoted $H^t_{N_i}$, is the subset of H^t pertaining to only agents in N_i. Thus, $H^t_{N_i}$ is basically an explicit form of H_i^t defined in Section 3. Each agent i is associated with a potential function $\pi_i(a^t_{N_i} \mid H^t_{N_i})$, which represents i's behavior conditioned on history. These potentials may be based on information from various sources, including agent payoffs in corresponding states, historical observations, models of social network interactions, and so on. The product of these potentials defines the joint probability distribution of the system's actions taken at time t,

$$\Pr(a^t \mid H^t) = \frac{\prod_i \pi_i(a^t_{N_i} \mid H^t_{N_i})}{Z}. \tag{3}$$

We interpret $H^t_{N_i}$ generally as a summary or abstraction of local history, since finite memory and computational power often preclude complete retention of historic observations. Moreover, from the perspective of the system modeler, only a partial view of the full history may be available. Yet another motivation for abstraction is provided by the need to limit complexity in order to effectively learn the model from a limited amount of data. Given an abstracted history representation, agent decisions will generally appear correlated, even if they are independently generated conditional on full history. In the voting consensus game for instance, at time period t we could choose to summarize agent i's full history by a count of how many times agents in i's neighborhood have voted 0 in $H^t_{N_i}$.

The complexity of computing the normalization factor Z in Equation (3) is exponential in the number of agents, and thus renders exact inference and learning in large undirected graphical models intractable. To handle large network scenarios, we have adopted the *belief propagation* method for approximately computing Z [Broadway et al., 2000]. Belief propagation is exact in tree-graph models, and has shown acceptably good results with reasonable runtime in sparse cyclic graphical models. This property makes belief propagation a viable approximation inference method for hGMMs, at least for scenarios involving sparsely connected networks of many small-size neighborhoods. Our implementation of hGMMs employs the package libDAI [Mooij, 2008] for computing Z using the belief propagation algorithm.

4.3 Model Learning and Evaluation

Here we 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 make θ explicit for the voting consensus game in the next section). We are interested in selecting a θ that maximizes the log likelihood of X,

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

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 IBMMs (B) as well.

We evaluate the learned dynamic multiagent models by their ability to predict future outcomes, as represented by a testing set Y in the same format as the training set X.

Given two models M_1 and M_2, we compute the ratio of their corresponding log-likelihood measures for the testing data set Y: $R_{M_1/M_2}(Y) = \frac{L_{M_1}(Y)}{L_{M_2}(Y)}$. In this study, we are particularly interested in the ratio $R_{hG/B}$. Note that since log likelihood is negative (we exclude certain predictions), $R_{hG/B} < 1$ indicates that the hGMM is better than the IBMM at predicting Y, and vice versa if the ratio exceeds one.

5. MODELING BEHAVIOR IN THE VOTING CONSENSUS GAME

We present parameterized hGMMs and IBMMs designed specifically to capture agent behavior in a version of the voting consensus game [Kearns et al., 2009]. We start by assuming that agents play synchronously, and introduce a second, asynchronous scenario in Section 6.

5.1 Parameters of hGMMs and IBMMs

First, we consider how to summarize a history $H^t_{N_i}$ 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 $f(a_i, H^t_{N_i})$ as the frequency of action a_i being chosen by other agents in $H^t_{N_i}$,

$$f(a_i, H^t_{N_i}) = \frac{\sum_{k \in N_i - \{i\}} \sum_{\rho=t-h}^{t-1} I(a_i, a_k^\rho) + 1}{h|N_i - \{i\}|}.$$

The frequency function f captures the degree to which a_i is similar to past choices by i's neighbors in $H^t_{N_i}$. Its joint analog reflects the historical frequency of a *local configuration* a_{N_i},

$$f(a_{N_i}, H^t_{N_i}) = \frac{\sum_{\rho=t-h}^{t-1} I(a_{N_i}, a^\rho_{N_i}) + 1}{h}.$$

We add 1 in both definitions above to ensure that the corresponding term does not vanish when the action a_i or the configuration a_{N_i}, respectively, do not appear in $H^t_{N_i}$. To simplify exposition, we henceforth drop the time superscript t and the neighborhood subscript N_i from $H^t_{N_i}$, taking these modifiers of history H as understood.

In formulating the hGMM 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 reflected in the reward $r_i(a_i)$. We encode other factors bearing on choice of action a_i by a parameter α_{i,a_i}.

The potential function for agent i is given by

$$\pi_i(a_{N_i}|H) = \exp\Big(\beta_i r_i(a_{N_i}) f(a_{N_i}, H) + \alpha_{i,a_i}\Big), \quad (4)$$

where for all i and $c \in A_i$, $0 \leq \alpha_{i,c}, \beta_i \leq 1$, and $\beta_i + \sum_c \alpha_{i,c} = 1$. The term $r_i(a_{N_i})$ is the expected reward for agent i given its neighborhood's play a_{N_i}. If we adopted the strict definition of reward from the game description, $r_i(a_{N_i})$ would be non-zero only when all actions in a_{N_i} are the same. To model the network contagion phenomenon, we use a modification $r_i(a_{N_i}) = \gamma^{\sum_{k \in N_i}(1-I(a_i,a_k))} r(a_i)$, where $r_i(a_i)$ is the reward i receives if everyone in the network plays a_i, and $\gamma \in (0, 1]$. Observe that $r_i(a_{N_i})$ as we define it is increasing in the number of i's neighbors playing a_i, reflecting the positive influence of neighbor choices on i.

In order to conduct a fair comparison between hGMMs and IBMMs for this problem domain, we seek to preserve as many features from the hGMM as possible in constructing an IBMM for the voting consensus game. We thus define the probabilistic IBMM strategy as follows:

$$\Pr(a_i|H) = \frac{1}{Z_i} \exp\Big(\beta_i r_i(a_i) f(a_i, H) + \alpha_{i,a_i}\Big). \quad (5)$$

As above, for all i and c, $0 \leq \alpha_{c,i}, \beta_i \leq 1$, and $\beta_i + \sum_c \alpha_{c,i} = 1$. 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).

5.2 Data from Fictitious Play Simulation

Given a voting consensus game with an underlying graphical structure, we generate data using a *smooth* fictitious play process [Camerer and Ho, 1999]. For purposes of our model evaluation, we treat this generated data as the actual multiagent system behavior. In fictitious play, each agent i maintains probabilistic beliefs about its neighbors' actions, based on the observed frequency of historic behavior. Let $b_{ij}(a_j)$ be i's belief that j chooses action a_j. At the beginning of a simulated game, each agent starts with uniform beliefs: $b_{ij}(a_j) = \frac{1}{|A_j|}$. Agents update their beliefs as follows. If at time t, j executed action a'_j, then for all a_j, and every neighbor i of j,

$$b_{ij}(a_j) \leftarrow \frac{b_{ij}(a_j)t + I(a_j, a'_j)}{t+1}.$$

Whereas in classic fictitious play agents respond optimally to their beliefs, in the smooth version agents select actions probabilistically in proportion to the expected reward computed with respect to these beliefs. We adopt an existing multiplicative model for this game [Kearns et al., 2009], which weighs the probability that an agent takes action a_i in proportion to the product of its reward $r_i(a_i)$ and the probability that all neighbors play the same action, $\prod_{j \in N_i} b_{ij}(a_i)$. Thus, in each round the simulation samples each agent i's action according to the probability distribution

$$\Pr(a_i) \propto r_i(a_i) \prod_{j \in N_i} b_{ij}(a_i). \quad (6)$$

The simulation terminates when all players reach a consensus or the allotted time runs out. This fictitious play simulation is itself a generative individual behavior multiagent model, as each agent chooses its action independently from others conditional on the commonly seen history. However, unlike the IBMM introduced in Section 5.1, which only employs a limited history of length h, this simulation model incorporates the full-length history in modeling and consequently generating agent actions.

6. EMPIRICAL STUDY

We empirically evaluate the predictive power of hGMMs and IBMMs in our version of the voting consensus game given limited history, as described in Section 5.1, on simulated play data generated by a smooth fictitious play process, introduced in Section 5.2.

6.1 Experiment Settings

We consider voting consensus games with 10 players. A game instance specifies player payoffs sampled uniformly randomly from $[0, 1]$, as well as a graphical structure of agent interactions where the maximum degree of the resulting graph is controlled to be d. Each experiment will provide

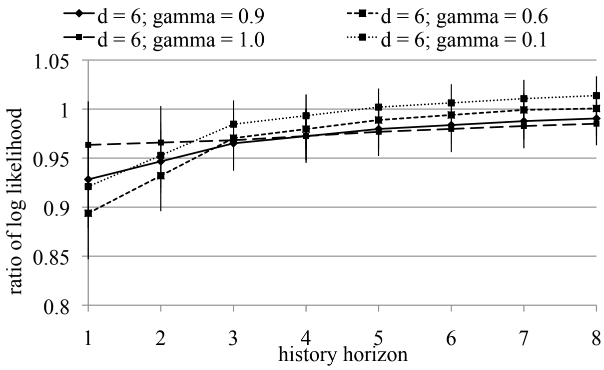

Figure 2: hGMMs outperform IBMMs in predicting game plays across scenarios of different values of γ

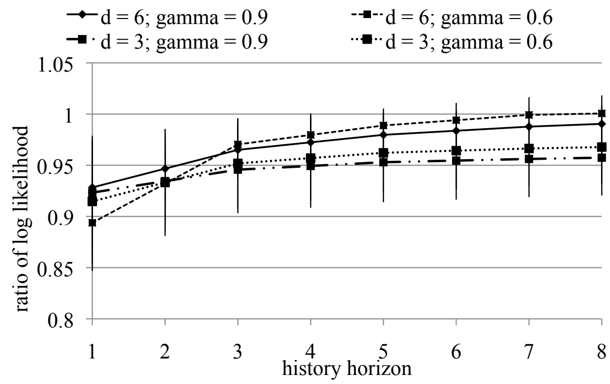

Figure 3: hGMMs provide better predictions than IBMMs across scenarios of different maximum node degrees d

results averaged over 20 game instances. We generate a data set of 20 game runs for each game instance by simulating independent runs of the smooth fictitious play process defined in Section 5.2. Each simulation run lasts until either everyone agrees on one of the two voting options or the time limit T (which we set at 50) is reached. We use $d = 6$ and $\gamma = 0.9$, introduced in Section 5.1, in our simulations unless otherwise specified.

Our study includes both synchronous and asynchronous game scenarios. In the synchronous scenario, which is the one we explore most extensively, agents update their beliefs and choose a response simultaneously in every round. The asynchronous version of the game, on the hand, allows agents to change their votes at any time. Our generative model described in Section 5.2 assumes a synchronous update model. In order to mimic asynchrony in voting decisions, we allow agents to have different update rates, with agent i updating his vote every t_i time periods. In the simulations we let $t_i = i$ for each agent i (that is, agent 1 updates his vote every period, agent 2 every other period, and so on). While we actually use a discrete model of asynchrony, we would like to test the effect of discretization that would be forced upon us (given that our models only work with synchronous scenarios) if agents in fact made their decisions continuously. To do this, we create a coarser discretization of time, combining a number of rounds into one by the use of the *summarization interval*, that is, a window of length v that combines all decisions v rounds at a time. Under this transformation, all actions that occur during the same epoch (that is, v "actual" time steps) are grouped to be effectively simultaneous.

For each game instance, we train the models using 10 game runs, and compare their performances in predicting the remaining 10 game runs by the ratio $R_{hG/B}$ defined in Section 4.3. Recall that all results present averages over the 20 game instances and that $R < 1$ indicates that hGMMs performed better than IBMMs.

6.2 Results

In our first set of experiments we evaluate the performance of hGMMs (as compared to IBMMs) as functions of the history horizon length for values of $\gamma \in [0, 1]$. We display results for a representative selection, $\gamma \in \{0.1, 0.6, 0.9, 1.0\}$. The results in Figure 2 show that in general hGMMs outperform IBMMs in predicting agent behavior for shorter history horizons. Their performances appear to converge as the history horizon increases. Since agents condition their actions on the complete history in the generative fictitious play simulation, a history of limited horizon renders agent actions correlated from the modeler's perspective. As a result, hGMMs, which directly model joint behavior, are more effective due to capturing action correlations induced by truncated history, even though the generative model is in fact an IBMM.

Figure 2 also demonstrates that for shorter history lengths, the performance of hGMMs peaks when γ is around 0.6, while longer history yields a higher "optimal" value of γ. The reason for this phenomenon is that action correlations are less prominent when history horizon increases, and, consequently, models closer to the generative IBMM—in this case, those with higher γ (equating the different reward terms $r_i(a_{N_i})$ and $r_i(a_i)$ in Equations 4 and 6)—perform better.

As the density of the underlying graphical structure and the correlations of agent behavior are intuitively related, we consider the effects of varying the maximum degree of generated graphs d. The results in Figure 3 for $d = 3$ and $d = 6$ suggest that hGMMs are better on sparser graphical structures. The intuition behind this is that action configurations in smaller neighborhoods tend to repeat more often and, consequently, increase the contribution of the $f(a_{N_i}, H_{N_i})$ term in the hGMM potentials, enhancing their power in capturing joint behavior.[1]

A natural alternative to either hGMM or IBMM is *fictitious play sampling*, which effectively mimics how one would use a generative fictitious play model to predict future plays. Specifically, fictitious play sampling computes the probability of an action profile a given history H as the empirical distribution of a in training data conditional on H. Our experimental results in Figure 4 show that hGMMs consistently outperforms fictitious play sampling, and this advantage is smallest at $h = 1$ and greatest at $h = 4$. Intuitively, shorter history horizons imply more that more data from the training data set can be used in the fictitious play sampling model to compute $\Pr\{a|H\}$ as there are more instances of the specific history H for shorter H, leading to more accurate predictions, even relative to hGMMs. The main illus-

[1]Of course, there is no correlation to be found when $d = 0$ and, thus, the difference between hGMM and IBMM is likely to be small for *very* sparse graphs.

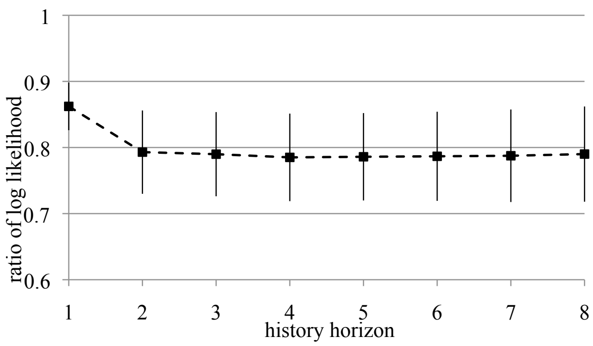

Figure 4: hGMMs provide better predictions than fictitious play sampling

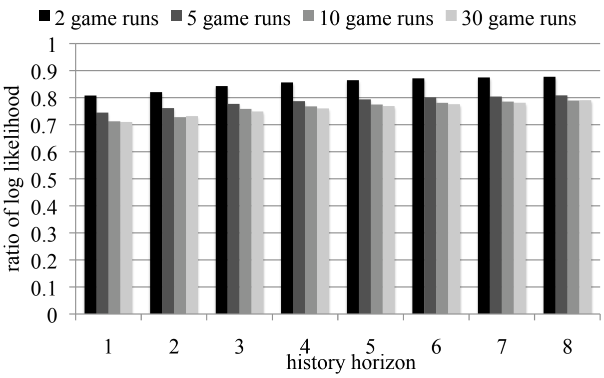

Figure 5: The effects of training data availability on the performance of hGMMs in comparison with untrained hGMMs.

trative point of this experiment set is that direct sampling from a fictitious play model is not only more computationally expensive but also less powerful in predicting outcomes than an hGMM capable of effectively extracting information about agent behavior from the same data set.

In order to assess the added value of training hGMM model parameters we compare trained and untrained hGMMs, where parameter values in the untrained models were chosen uniformly randomly. Specifically, we used untrained hGMMs (training data set size of 0) as a baseline, and varied the size of the training data in the set $\{2, 5, 10, 30\}$. The corresponding likelihood ratio of trained to untrained hGMMs evaluated on test data are displayed in Figure 5. In essence, the outcomes confirm that our default size for the training data set (10 game runs) is sufficient for learning hGMMs. Analogous results were also obtained for IBMMs.

Our next set of experiments evaluates the impact of asynchrony in agent behavior on the relative efficacy of hGMMs and IBMMs. Specifically, we consider two summarization intervals v, with $v = 2$ and $v = 4$, with the results shown in Figure 6. We observe that hGMMs are more powerful in modeling correlations with longer summarization intervals v. This is further reinforced by observing that the advantage of hGMM over IBMM for both summarization window sizes here is greater than we observed in the synchronous environment (which would roughly correspond to $v = 1$, although all agents also make their decisions simultaneously

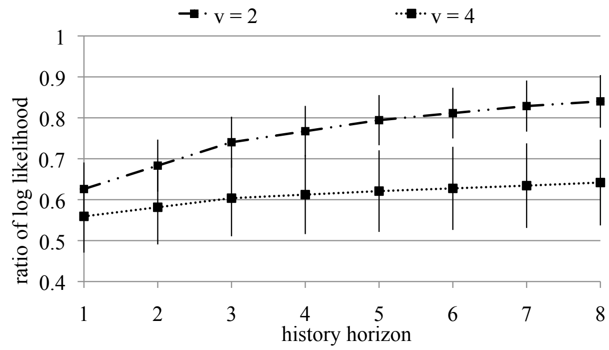

Figure 6: hGMMs provides better predictions than IBMMs in asynchronous scenarios with different summarization intervals v

in that model). To understand this phenomenon note that actions that occur in different periods that are collapsed to fall in the same summarization window v in an asynchronous scenario are treated as simultaneous actions in both models, even though earlier actions may well cause the later ones. As v increases, more cause-effect action pairs are grouped in the same rounds in the transformed (synchronized) data, yielding greater action correlations that are unmodeled in IBMMs. Since asynchrony in multiagent systems is more a rule than an exception, our results suggest that hGMMs may be especially efficacious in practice.

A real concern present in using hGMMs in realistic settings is the scalability of our techniques when the number of players is large. While all the problems that we consider here are small enough to enable exact learning, large games would require us to use approximation. We now proceed to verify that approximation is feasible in our setting and the advantage of hGMMs over IBMMs does not dissipate when learning is approximate rather than exact. Figure 7 illustrates the performance of hGMMs that use both exact and approximate inference: in fact, approximate hGMMs actually *outperform* hGMMs that use exact inference. Our intuition for this surprising result is that action profiles with sufficiently small values of potentials are likely to be dropped from the approximate computation of Z. As a result, the approximate hGMMs become more "focused" in explaining and predicting more frequent outcomes, while getting "punished" more when a less frequent outcome occurs. More significantly, approximate hGMMs outperform IBMMs in predicting the game's sequential outcomes, allowing our techniques to scale to realistic scenarios.

7. CONCLUSIONS

History-dependent graphical multiagent models support efficient and effective inference about system dynamics given abstract representations of history that may induce action correlations from the modeler's perspective. In particular, hGMMs provide a compact and flexible representation framework for scenarios of decomposable structure, while enabling direct reasoning about joint behaviors. We illustrated the representational and inferential capabilities of hGMMs by empirically showing that they outperform individual behavior multiagent models and fictitious play sampling models in predicting data and answering inference queries. In addition, approximate inference did not degrade the predictive

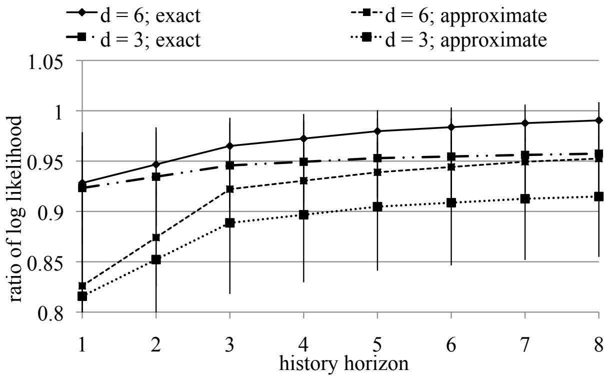

Figure 7: hGMMs show greater predictive power than IBMMs when employing generalized belief propagation approximation.

power of hGMMs in our studies, which is a promising indicator for application of hGMMs to large networks.

We seek to apply our hGMM framework in modeling agent behavior in various dynamic multiagent scenarios. A follow-up study will focus on analyzing voting behavior from data of the voting consensus experiments conducted by Kearns et al. [2009]. To evaluate the capacity of hGMMs to capture a qualitatively different scenario, we could attempt to model the competition among retailers specializing in different products, using behavior observed in the Trading Agent Competition Ad Auction game [Jordan and Wellman, 2010]. Another research direction is to extend hGMMs to a more complete dynamic representation, supporting not only forward inference but also backward inference, to reason about unobserved past states. Finally, we might employ techniques for learning underlying graphical structures of multiagent scenarios [Duong et al., 2009] in developing learning methods for not only hGMM parameters, but also the graph topology itself.

References

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.

C. Camerer and T. Ho. Experienced-weighted attraction learning in normal form games. *Econometrica*, 67(4):827–874, 1999.

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.

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.

Q. Duong, Y. Vorobeychik, S. Singh, and M. P. Wellman. Learning graphical game models. In *Twenty-First International Joint Conference on Artificial Intelligence*, pages 116–121, Pasadena, CA, 2009.

D. Fudenberg and D. K. Levine. *The Theory of Learning in Games.* MIT Press, 1998.

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.

M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–43, 1978.

D. Hennes, K. Tuyls, and M. Rauterberg. State-coupled replicator dynamics. In *Eighth International Conference on Autonomous Agents and Multi-Agent Systems*, pages 789–796, Budapest, 2009.

A. X. Jiang, K. Leyton-Brown, and N. A. R. Bhat. Action-graph games. Technical Report UBC CS TR-2008-13, University of British Columbia, 2008.

P. R. Jordan and M. P. Wellman. Designing an ad auctions game for the trading agent competition. In *Agent-Mediated Electronic Commerce: Designing Trading Strategies and Mechanisms for Electronic Markets*, Lecture Notes in Business Information Processing. Springer-Verlag, 2010.

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.

M. Kearns and J. Wortman. Learning from collective behavior. In *Twenty-First Annual Conference on Learning Theory*, pages 99–110, Helsinki, 2008.

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.

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):1347–52, 2009.

J. Kleinberg. Cascading behavior in networks: Algorithmic and economic issues. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, pages 613–632. Cambridge University Press, 2007.

D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. In *Seventeenth International Joint Conference on Artificial Intelligence*, pages 1027–1036, Seattle, WA, 2001.

J. M. Mooij. libdai 0.2.2: A free/open source c++ library for discrete approximate inference methods, 2008. URL `http://www.libdai.org`.

D. Mookherjee and B. Sopher. Learning behavior in an experimental matching pennies game. *Games and Economic Behavior*, 7(1):62 – 91, 1994.

Y. Shoham and M. Tennenholtz. On the emergence of social conventions: Modeling, analysis, and simulations. *Artificial Intelligence*, 94:139–166, 1997.

K. Tuyls and S. Parsons. What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, 171:406–416, 2007.