Learning from Collective Behavior

Michael Kearns
Computer and Information Science
University of Pennsylvania
mkearns@cis.upenn.edu

Jennifer Wortman
Computer and Information Science
University of Pennsylvania
wortmanj@seas.upenn.edu

Abstract

Inspired by longstanding lines of research in sociology and related fields, and by more recent large-population human subject experiments on the Internet and the Web, we initiate a study of the computational issues in learning to model collective behavior from observed data. We define formal models for efficient learning in such settings, and provide both general theory and specific learning algorithms for these models.

1 Introduction

Collective behavior in large populations has been a subject of enduring interest in sociology and economics, and a more recent topic in fields such as physics and computer science. There is consequently now an impressive literature on mathematical models for collective behavior in settings as diverse as the diffusion of fads or innovation in social networks [10, 1, 2, 18], voting behavior [10], housing choices and segregation [22], herding behaviors in financial markets [27, 8], Hollywood trends [25, 24], critical mass phenomena in group activities [22], and many others. The advent of the Internet and the Web have greatly increased the number of both controlled experiments [7, 17, 20, 21, 8] and open-ended systems (such as Wikipedia and many other instances of "human peer-production") that permit the logging and analysis of detailed collective behavioral data. It is natural to ask if there are learning methods specifically tailored to such models and data.

The mathematical models of the collective behavior literature differ from one another in important details, such as the extent to which individual agents are assumed to act according to traditional notions of rationality, but they generally share the significant underlying assumption that each agent's current behavior is entirely or largely determined by the recent behavior of the other agents. Thus the collective behavior is a *social* phenomenon, and the population evolves over time according to its own internal dynamics — there is no exogenous "Nature" being reacted to, or injecting shocks to the collective.

In this paper, we introduce a computational theory of learning from collective behavior, in which the goal is to accurately model and predict the future behavior of a large population after observing their interactions during a training phase of polynomial length. We assume that each agent i in a population of size N acts according to a fixed but unknown strategy c_i drawn from a known class $\mathcal{C}$. A strategy probabilistically maps the current population state to the next state or action for that agent, and each agent's strategy may be different. As is common in much of the literature cited above, there may also be a network structure governing the population interaction, in which case strategies may map the local neighborhood state to next actions.

Learning algorithms in our model are given training data of the population behavior, either as repeated finite-length trajectories from multiple initial states (an *episodic* model), or in a single unbroken trajectory from a fixed start state (a *no-reset* model). In either case, they must efficiently (polynomially) learn to accurately predict or simulate (properties of) the future behavior of the same population. Our framework may be viewed as a computational model for learning the dynamics of an unknown Markov process — more precisely, a dynamic Bayes net — in which our primary interest is in Markov processes inspired by simple models for social behavior.

As a simple, concrete example of the kind of system we have in mind, consider a population in which each agent makes a series of choices from a fixed set over time (such as what restaurant to go to, or what political party to vote for). Like many previously studied models, we consider agents who have a desire to behave like the rest of the population (because they want to visit the popular restaurants, or want to vote for "electable" candidates). On the other hand, each agent may also have different and unknown intrinsic preferences over the choices as well (based on cuisine and decor, or the actual policies of the candidates). We consider models in which each agent balances or integrates these two forces in deciding how to behave at each step [12]. Our main question is: Can a learning algorithm watching the collective behavior of such a population for a short period produce an accurate model of their future choices?

The assumptions of our model fit nicely with the literature cited in the first paragraph, much of which indeed proposes simple stochastic models for how individual agents react to the current population state. We emphasize from the outset the difference between our interests and those common in multiagent systems and learning in games. In those fields, it is often the case that the agents themselves are acting according to complex and fairly general learning algorithms (such as Q-learning [26], no-regret learning [9], fictitious play [3], and so on), and the central question is whether and when the population converges to particular, "nice" states (such as Nash or correlated equilibria). In contrast, while the agent strategies we consider are certainly "adaptive" in a reactive sense, they are much simpler than general-purpose learning algorithms, and we are interested in learning algorithms that *model* the full collective behavior no matter what its properties; there is no special status given either to particular states nor to any notion of convergence. Thus our interest is not in learning by the agents themselves, but at the higher level of an observer of the population.

Our primary contributions are:

- The introduction of a computational model for learning from collective behavior.
- The development of some general theory for this model, including a polynomial-time reduction of learning from collective behavior to learning in more traditional, single-target I.I.D. settings, and a separation between efficient learnability in collective models in which the learner does and does not see all intermediate population states.
- The definition of specific classes of agent strategies, including variants of the "crowd affinity" strategies sketched above, and complementary "crowd aversion" classes.
- Provably efficient algorithms for learning from collective behavior for these same classes.

The outline of the paper is as follows. In Section 2, we introduce our main model for learning from collective behavior, and then discuss two natural variants. Section 3 introduces and motivates a number of specific agent strategy classes that are broadly inspired by earlier sociological models, and provides brief simulations of the collective behaviors they can generate. Section 4 provides a general reduction of learning from collective behavior to a generalized PAC-style model for learning from I.I.D. data, which is used subsequently in Section 5, where we give provably efficient algorithms for learning some of the strategy classes introduced in Section 3. Brief conclusions and topics for further research are given in Section 6.

2 The Model

In this section we describe a learning model in which the observed data is generated from observations of trajectories (defined shortly) of the collective behavior of N interacting agents. The key feature of the model is the fact that each agent's next state or action is always *determined by the recent actions of the other agents*, perhaps combined with some intrinsic "preferences" or behaviors of the particular agent. As we shall see, we can view our model as one for learning certain kinds of factored Markov processes that are inspired by models common in sociology and related fields.

Each agent may follow a different and possibly probabilistic strategy. We assume that the strategy followed by each agent is constrained to lie in a known (and possibly large) class, but is otherwise unknown. The learner's ultimate goal is not to discover each individual agent strategy per se, but rather to make accurate predictions of the *collective* behavior in novel situations.

2.1 Agent Strategies and Collective Trajectories

We now describe the main components of our framework:

- **State Space.** At each time step, each agent i is in some state s_i chosen from a known, finite set $\mathcal{S}$ of size K. We often think of K as being large, and thus want algorithms whose running time scales polynomially in K and other parameters. We view s_i as the *action* taken by agent i in response to the recent population behavior. The joint action vector $\vec{s} \in \mathcal{S}^N$ describes the current global state of the collective.
- **Initial State Distribution.** We assume that the initial population state $\vec{s}^{\,0}$ is drawn according to a fixed but unknown distribution P over $\mathcal{S}^N$. During training, the learner is able to see trajectories of the collective behavior in which the initial state is drawn from P, and as in many standard learning models, must generalize with respect to this same distribution. (We also consider a no-reset variant of our model in Section 2.3.)
- **Agent Strategy Class.** We assume that each agent's strategy is drawn from a known class $\mathcal{C}$ of (typically probabilistic) mappings from the recent collective behavior into the agent's next state or action in $\mathcal{S}$. We mainly consider the case in which $c_i \in \mathcal{C}$ probabilistically maps the current global state $\vec{s}$ into agent i's next state. However, much of the theory we develop applies equally well to more complex strategies that might incorporate a longer history of the collective behavior on the current trajectory, or might depend on summary statistics of that history.

Given these components, we can now define what is meant by a *collective trajectory*.

Definition 1 *Let $\vec{c} \in \mathcal{C}^N$ be the vector of strategies for the N agents, P be the initial state distribution, and $T \geq 1$ be an integer. A T***-trajectory of** $\vec{c}$ **with respect to** P *is a random variable $\langle \vec{s}^{\,0}, \cdots, \vec{s}^{\,T} \rangle$ in which the initial state $\vec{s}^{\,0} \in \mathcal{S}^N$ is drawn according to P, and for each $t \in \{1, \cdots, T\}$, the component s_i^t of the joint state $\vec{s}^{\,t}$ is obtained by applying the strategy c_i to $\vec{s}^{\,t-1}$. (Again, more generally we may also allow the strategies c_i to depend on the full sequence $\vec{s}^{\,0}, \ldots, \vec{s}^{\,t-1}$, or on summary statistics of that history.)*

Thus, a collective trajectory in our model is simply a Markovian sequence of states that *factors* according to the N agent strategies — that is, a dynamic Bayes net [19]. Our interest is in cases in which this Markov process is generated by particular models of social behavior, some of which are discussed in Section 3.

2.2 The Learning Model

We now formally define the learning model we study. In our model, learning algorithms are given access to an oracle $\mathcal{O}_{\mathrm{EXP}}(\vec{c}, P, T)$ that returns a T-trajectory $\langle \vec{s}^{\,0}, \cdots, \vec{s}^{\,T} \rangle$ of

$\vec{c}$ with respect to P. This is thus an *episodic* or *reset* model, in which the learner has the luxury of repeatedly observing the population behavior from random initial conditions. It is most applicable in (partially) controlled, experimental settings [7, 17, 20, 21, 8] where such "population resets" can be implemented or imposed. In Section 2.3 below we define a perhaps more broadly applicable variant of the model in which resets are not available; the algorithms we provide can be adapted for this model as well (Section 5.3).

The goal of the learner is to find a *generative model* that can efficiently produce trajectories from a distribution that is arbitrarily close to that generated by the true population. Thus, let $\hat{M}(\vec{s}^{\,0}, T)$ be a (randomized) model output by a learning algorithm that takes as input a start state $\vec{s}^{\,0}$ and time horizon T, and outputs a random T-trajectory, and let $Q_{\hat{M}}$ denote the distribution over trajectories generated by $\hat{M}$ when the start state is distributed according to P. Similarly, let $Q_{\vec{c}}$ denote the distribution over trajectories generated by $\mathcal{O}_{\mathrm{EXP}}(\vec{c}, P, T)$. Then the goal of the learning algorithm is to find a model $\hat{M}$ making the $\mathcal{L}_1$ distance $\varepsilon(Q_{\hat{M}}, Q_{\vec{c}})$ between $Q_{\hat{M}}$ and $Q_{\vec{c}}$ small, where

$$\varepsilon(Q_{\hat{M}}, Q_{\vec{c}}) \equiv \sum_{\langle \vec{s}^{\,0}, \cdots, \vec{s}^{\,T} \rangle} \left| Q_{\hat{M}}(\langle \vec{s}^{\,0}, \cdots, \vec{s}^{\,T} \rangle) - Q_{\vec{c}}(\langle \vec{s}^{\,0}, \cdots, \vec{s}^{\,T} \rangle) \right| .$$

A couple of remarks are in order here. First, note that we have defined the output of the learning algorithm to be a "black box" that simply produces trajectories from initial states. Of course, it would be natural to expect that this black box operates by having good approximations to every agent strategy in $\vec{c}$, and using collective simulations of these to produce trajectories, but we choose to define the output $\hat{M}$ in a more general way since there may be other approaches. Second, we note that our learning criteria is both strong (see below for a discussion of weaker alternatives) and useful, in the sense that if $\varepsilon(Q_{\hat{M}}, Q_{\vec{c}})$ is smaller than ϵ, then we can sample $\hat{M}$ to obtain $O(\epsilon)$-good approximations to the expectation of any (bounded) *function* of trajectories. Thus, for instance, we can use $\hat{M}$ to answer questions like "What is the expected number of agents playing the plurality action after T steps?" or "What is the probability the entire population is playing the same action after T steps?" (In Section 2.4 below we discuss a weaker model in which we care only about one *fixed* outcome function.)

Our algorithmic results consider cases in which the agent strategies may themselves already be rather rich, in which case the learning algorithm should be permitted resources commensurate with this complexity. For example, the crowd affinity models have a number of parameters that scales with the number of actions K. More generally, we use $dim(\mathcal{C})$ to denote the complexity or dimension of $\mathcal{C}$; in all of our imagined applications $dim(\cdot)$ is either the VC dimension for deterministic classes, or one of its generalizations to probabilistic classes (such as pseudo-dimension [11], fat-shattering dimension [15], combinatorial dimension [11], etc.).

We are now ready to define our learning model.

Definition 2 *Let $\mathcal{C}$ be an agent strategy class over actions $\mathcal{S}$. We say that $\mathcal{C}$ is* **polynomially learnable from collective behavior** *if there exists an algorithm A such that for any population size $N \geq 1$, any $\vec{c} \in \mathcal{C}^N$, any time horizon T, any distribution P over $\mathcal{S}^N$, and any $\epsilon > 0$ and $\delta > 0$, given access to the oracle $\mathcal{O}_{\mathrm{EXP}}(\vec{c}, P, T)$, algorithm A runs in time polynomial in N, T, $dim(\mathcal{C})$, $1/\epsilon$, and $1/\delta$, and outputs a polynomial-time model $\hat{M}$ such that with probability at least $1 - \delta$, $\varepsilon(Q_{\hat{M}}, Q_{\vec{c}}) \leq \epsilon$.*

We now discuss two reasonable variations on the model we have presented.

2.3 A No-Reset Variant

The model above assumes that learning algorithms are given access to repeated, independent trajectories via the oracle $\mathcal{O}_{\mathrm{EXP}}$, which is analogous to the *episodic* setting of reinforcement learning. As in that field, we may also wish to consider an alternative "no-reset" model in which the learner has access only to a *single*, unbroken trajectory of states generated by the Markov process. To do so we must formulate an alternative notion of generalization, since on the one hand, the (distribution of the) initial state may quickly become irrelevant as the collective behavior evolves, but on the other, the state space is exponentially large and thus it is unrealistic to expect to model the dynamics from an *arbitrary* state in polynomial time.

One natural formulation allows the learner to observe any polynomially long prefix of a trajectory of states for training, and then to announce its readiness for the test phase. If $\vec{s}$ is the final state of the training prefix, we can simply ask that the learner output a model $\hat{M}$ that generates accurate T-step trajectories *forward* from the current state $\vec{s}$. In other words, $\hat{M}$ should generate trajectories from a distribution close to the distribution over T-step trajectories that would be generated if each agent continued choosing actions according to his strategy. The length of the training prefix is allowed to be polynomial in T and the other parameters.

While aspects of the general theory described below are particular to our main (episodic) model, we note here that the algorithms we give for specific classes can in fact be adapted to work in the no-reset model as well. Such extensions are discussed briefly in Section 5.3.

2.4 Weaker Criteria for Learnability

We have chosen to formulate learnability in our model using a rather strong success criterion — namely, the ability to (approximately) simulate the full dynamics of the unknown Markov process induced by the population strategy $\vec{c}$. In order to meet this strong criterion, we have also allowed the learner access to a rather strong oracle, which returns all *intermediate* states of sampled trajectories.

There may be natural scenarios, however, in which we are interested only in specific *fixed* properties of collective behavior, and thus a weaker data source may suffice. For instance, suppose we have a fixed, real-valued *outcome function* $F(\vec{s}^{\,T})$ of final states (for instance, the fraction of agents playing the plurality action at time T), with our goal being to simply learn a function G that maps initial states $\vec{s}^{\,0}$ and a time horizon T to real values, and approximately minimizes

$$\mathrm{E}_{\vec{s}^{\,0} \sim P}\left[\left| G(\vec{s}^{\,0}, T) - \mathrm{E}_{\vec{s}^{\,T}}[F(\vec{s}^{\,T})] \right| \right]$$

where $\vec{s}^{\,T}$ is a random variable that is the final state of a T-trajectory of $\vec{c}$ from the initial state $\vec{s}^{\,0}$. Clearly in such a model, while it certainly would suffice, there may be no need to directly learn a full dynamical model. It may be feasible to satisfy this criterion without even observing intermediate states, but only seeing initial state and final outcome pairs $\langle \vec{s}^{\,0}, F(\vec{s}^{\,T})\rangle$, closer to a traditional regression problem.

It is not difficult to define simple agent strategy classes for which learning from only $\langle \vec{s}^{\,0}, F(\vec{s}^{\,T})\rangle$ pairs is provably intractable, yet efficient learning is possible in our model. This idea is formalized in Theorem 3 below. Here the population forms a rather powerful computational device mapping initial states to final states. In particular, it can be thought of as a circuit of depth T with "gates" chosen from $\mathcal{C}$, with the only real constraint being that each layer of the circuit is an identical sequence of N gates which are applied to the outputs of the previous layer. Intuitively, if only initial states and final outcomes are provided to the learner, learning should be as difficult as a corresponding PAC-style problem. On the other hand, by observing intermediate state vectors we can build arbitrarily accurate models for each agent, which in turn allows us to accurately simulate the full dynamical model.

Theorem 3 *Let $\mathcal{C}$ be the class of 2-input* AND *and* OR *gates, and one-input* NOT *gates. Then $\mathcal{C}$ is polynomially learnable from collective behavior, but there exists a binary outcome function F such that learning an accurate mapping from start states $\vec{s}^{\,0}$ to outcomes $F(\vec{s}^{\,T})$ without observing intermediate state data is intractable.*

Proof: (Sketch) We first sketch the hardness construction. Let $\mathcal{H}$ be any class of Boolean circuits (that is, with gates in $\mathcal{C}$) that is not polynomially learnable in the standard PAC model; under standard cryptographic assumptions, such a class exists. Let D be a hard distribution for PAC learning $\mathcal{H}$. Let $h \in \mathcal{H}$ be a Boolean circuit with R inputs, S gates, and depth D. To embed the computation by h in a collective problem, we let $N = R + S$ and $T = D$. We introduce an agent for each of the R inputs to h, whose value after the initial state is set according to an arbitrary AND, OR, or NOT gate. We additionally introduce one agent for every gate g in h. If a gate g in h takes as its inputs the outputs of gates g' and g'', then at each time step the agent corresponding to g computes the corresponding function of the states of the agents corresponding to g' and g'' at the previous time step. Finally, by convention we always have the Nth agent be the agent corresponding to the output gate of h, and define the output function as $F(\vec{s}) = s_N$. The distribution P over initial states of the N agents is identical to D on the R agents corresponding to the inputs of h, and arbitrary (e.g., independent and uniform) on the remaining S agents.

Despite the fact that this construction introduces a great deal of spurious computation (for instance, at the first time step, many or most gates may simply be computing Boolean functions of the random bits assigned to non-input agents), it is clear that if gate g is at depth d in h, then at time d in the collective simulation of the agents, the corresponding agent has exactly the value computed by g under the inputs to h (which are distributed according to D). Because the outcome function is the value of the agent corresponding to the output gate of h at time $T = D$, pairs of the form $\langle \vec{s}^{\,0}, F(\vec{s}^{\,T})\rangle$ provide exactly the same data as the PAC model for h under D, and thus must be equally hard.

For the polynomial learnability of $\mathcal{C}$ from collective behavior, we note that $\mathcal{C}$ is clearly PAC learnable, since it is just Boolean combinations of 1 or 2 inputs. In Section 4 we give a general reduction from collective learning of any agent strategy class to PAC learning the class, thus giving the claimed result. ■

Conversely, it is also not difficult to concoct cases in which learning the full dynamics in our sense is intractable, but we can learn to approximate a specific outcome function from only $\langle \vec{s}^{\,0}, F(\vec{s}^{\,T})\rangle$ pairs. Intuitively, if each agent strategy is very complex but the outcome function applied to final states is sufficiently simple (e.g., constant), we cannot but do not need to model the full dynamics in order to learn to approximate the outcome.

We note that there is an analogy here to the distinction between *direct* and *indirect* approaches to reinforcement learning [16]. In the former, one learns a policy that is specific to a fixed reward function without learning a model of next-state dynamics; in the latter, at possibly greater cost, one learns an accurate dynamical model, which can in turn be used to compute good policies for any reward function. For the remainder of this paper, we focus on the model as we formalized it in Definition 2, and leave for future work the investigation of such alternatives.

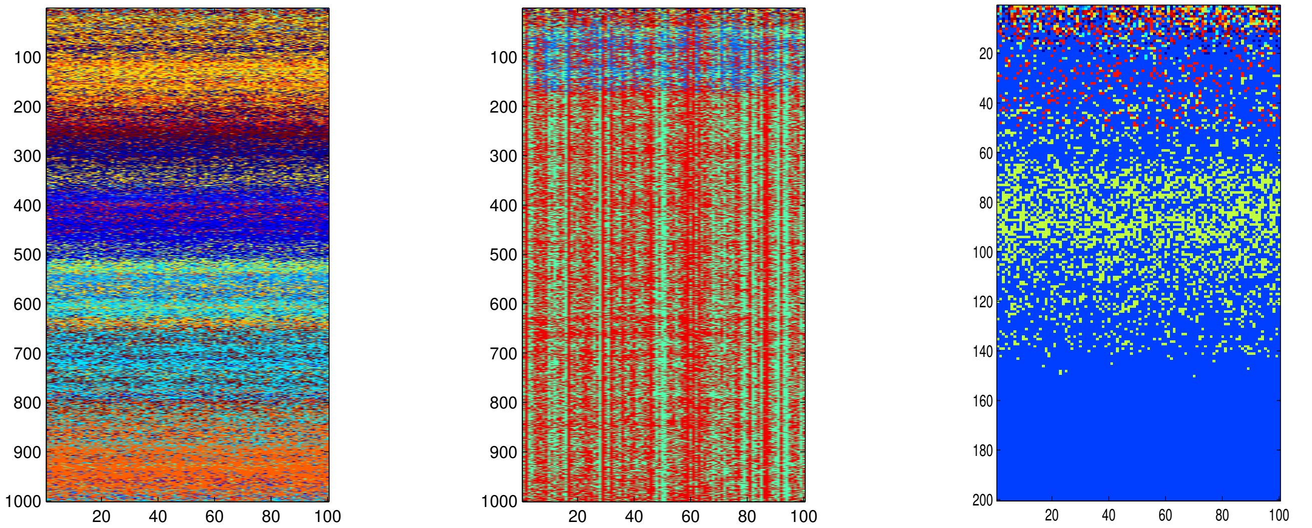
3 Social Strategy Classes

Before providing our general theory, including the reduction from collective learning to I.I.D. learning, we first illustrate and motivate the definitions so far with some concrete examples of social strategy classes, some of which we analyze in detail in Section 5.

3.1 Crowd Affinity: Mixture Strategies

The first class of agent strategies we discuss are meant to model settings in which each individual wishes to balance their intrinsic personal preferences with a desire to "follow the crowd." We broadly refer to strategies of this type as *crowd affinity* strategies (in contrast to the *crowd aversion* strategies discussed shortly), and examine a couple of natural variants.

As a motivating example, imagine that there are K restaurants, and each week, every member of a population chooses one of the restaurants in which to dine. On the one hand, each agent has personal preferences over the restaurants based on the cuisine, service, ambiance, and so on. On the other, each agent has some desire to go to the currently "hot" restaurants — that is, where many or most other agents have been recently. To model this setting, let $\mathcal{S}$ be the set of K restaurants, and suppose $\vec{s} \in \mathcal{S}^N$ is the population state vector indicating where each agent dined last week. We can summarize the population behavior by the vector or distribution $\vec{f} \in [0,1]^K$, where f_a is the fraction of agents dining in restaurant a in $\vec{s}$. Similarly, we might represent the personal preferences of a specific agent by another distribution $\vec{w} \in [0,1]^K$ in which w_a represents the probability this agent would attend restaurant a in the absence of any information



(a) (b) (c)

Figure 1: Sample simulations of the (a) crowd affinity mixture model; (b) crowd affinity multiplicative model; (c) agent affinity model. Horizontal axis is population state; vertical axis is simulation time. See text for details.

about what the population is doing. One natural way for the agent to balance their preferences with the population behavior would be to choose a restaurant according to the mixture distribution $(1-\alpha)\vec{f} + \alpha\vec{w}$ for some agent-dependent mixture coefficient α. Such models have been studied in the sociology literature [12] in the context of belief formation.

We are interested in collective systems in which every agent i has some unknown preferences $\vec{w}_i$ and mixture coefficient α_i, and in each week t chooses its next restaurant according to $(1-\alpha_i)\vec{f}^t + \alpha_i\vec{w}_i$, which thus probabilistically yields the next population distribution $\vec{f}^{t+1}$. How do such systems behave? And how can we learn to model their macroscopic properties from only observed behavior, especially when the number of choices K is large?

An illustration of the rich collective behavior that can already be generated from such simple strategies is shown in Figure 1(a). Here we show a single but typical 1000-step simulation of collective behavior under this model, in which $N = 100$ and each agent's individual preference vector $\vec{w}$ puts all of its weight on just one of 10 possible actions (represented as colors); this action was selected independently at random for each agent. All agents have an α value of just 0.01, and thus are selecting from the population distribution 99% of the time. Each row shows the population state at a given step, with time increasing down the horizontal axis of the image. The initial state was chosen uniformly at random.

It is interesting to note the dramatic difference between $\alpha = 0$ (in which rapid convergence to a common color is certain) and this small value for α; despite the fact that almost all agents play the population distribution at every step, revolving horizontal waves of near-consensus to different choices are present, with no final convergence in sight. The slight "personalization" of population-only behavior is enough to dramatically change the collective behavior. Broadly speaking, it is such properties we would like a learning algorithm to model and predict from sufficient observations.

3.2 Crowd Affinity: Multiplicative Strategies

One possible objection to the crowd affinity mixture strategies described above is that each agent can be viewed as *randomly* choosing whether to *entirely* follow the population distribution (with probability $1-\alpha$) or to *entirely* follow their personal preferences (with probability α) at each time step. A more realistic model might have each agent truly *combine* the population behavior with their preferences at every step.

Consider, for instance, how an American citizen might alter their anticipated presidential voting decision over time in response to recent primary or polling news. If their first choice of candidate — say, an Independent or Libertarian candidate — appears over time to be "unelectable" in the general election due to their inability to sway large numbers of Democratic and Republican voters, a natural and typical response is for the citizen to shift their intended vote to whichever of the front-runners they most prefer or least dislike. In other words, the low popularity of their first choice causes that choice to be dampened or eradicated; unlike the mixture model above, where weight α is always given to personal preferences, here there may remain *no* weight on this candidate.

One natural way of defining a general such class of strategies is as follows. As above, let $\vec{f} \in [0,1]^K$, where f_a is the fraction of agents dining in restaurant a in the current state $\vec{s}$. Similar to the mixture strategies above, let $\vec{w}_i \in [0,1]^K$ be a vector of *weights* representing the intrinsic preferences of agent i over actions. Then define the probability that agent i plays action a to be $f_a \cdot w_{i,a}/Z(\vec{f}, \vec{w}_i)$, where the normalizing factor is $Z(\vec{f}, \vec{w}_i) = \sum_{b\in\mathcal{S}} f_b \cdot w_{i,b}$.

Thus, in such *multiplicative* crowd affinity models, the probability the agent takes an action is always proportional to the product of their preference for it and its current popularity.

Despite their similar motivation, the mixture and multiplicative crowd affinity strategies can lead to dramatically different collective behavior. Perhaps the most obvious difference is that in the mixture case, if agent i has a strong preference for action a there is *always* some minimum probability ($\alpha_i w_{i,a}$) they take this action, whereas in the multiplicative case even a strong preference can be eradicated from expression by small or zero values for the popularity f_a.

In Figure 1(b), we again show a single but typical 1000-step, $N = 100$ simulation for the multiplicative model in which agent's individual preference distributions $\vec{w}$ are chosen to be random normalized vectors over 10 actions. The dynamics are now quite different than for the additive crowd affinity model. In particular, now there is never near-consensus but a gradual dwindling of the colors represented in the population — from the initial full diversity down to 3 colors remaining at approximately $t = 100$, until by $t = 200$ there is a stand-off in the population between red and light green. Unlike the additive models, colors die out in the population permanently. There is also clear vertical structure corresponding to strong conditional preferences of the agents once the stand-off emerges.

3.3 Crowd Aversion and Other Variants

It is easy to transform the mixture or multiplicative crowd affinity strategies into *crowd aversion* strategies — that is, in which agents wish to balance or combine their personal preferences with a desire to act *differently* than the population at large. This can be accomplished in a variety of simple ways. For instance, if $\vec{f}$ is the current distributions over actions in the population, we can simply define a kind of "inverse" to the distribution by letting $g_a = (1 - f_a)/(K - 1)$, where $K - 1 = \sum_{b \in \mathcal{S}} (1 - f_b)$ is the normalizing factor, and applying the strategies above to $\vec{g}$ rather than $\vec{f}$. Now each agent exhibits a tendency to "avoid the crowd", moderated as before by their own preferences.

Of course, there is no reason to assume that the entire population is crowd-seeking, or crowd-avoiding; more generally we would allow there to be both types of individuals present. Furthermore, we might entertain other transforms of the population distribution than just g_a above. For instance, we might wish to still consider crowd affinity, but to first "sharpen" the distribution by replacing each f_a with f_a^2 and normalizing, then applying the models discussed above to the resulting vector. This has the effect of magnifying the attraction to the most popular actions. In general our algorithmic results are robust to a wide range of such variations.

3.4 Agent Affinity and Aversion Strategies

In the two versions of crowd affinity strategies discussed above, an agent has personal preferences over actions, and also reacts to the current population behavior, but only in an aggregate fashion. An alternative class of strategies that we call *agent affinity* strategies instead allows agents to prefer to agree (or disagree) in their choice with specific other agents.

For a fixed agent, such a strategy can be modeled by a weight vector $\vec{w} \in [0, 1]^N$, with one weight for each *agent* in the population rather than each action. We define the probability that this agent takes action a if the current global state is $\vec{s} \in \mathcal{S}^N$ to be proportional to $\sum_{i:s_i=a} w_i$. In this class of strategies, the strength of the agent's desire to take the same action as agent i is determined by how large the weight w_i is. The overall behavior of this agent is then probabilistically determined by summing over all agents in the fashion above.

In Figure 1(c), we show a single but typical simulation, again with $N = 100$ but now with a much shorter time horizon of 200 steps and a much larger set of 100 actions. All agents have random distributions as their preferences over other agents; this model is similar to traditional diffusion dynamics in a dense, random (weighted) network, and quickly converges to global consensus.

We leave the analysis of this strategy class to future work, but remark that in the simple case in which $K = 2$, learning this class is closely related to the problem of learning perceptrons under certain noise models in which the intensity of the noise increases with proximity to the separator [5, 4] and seems at least as difficult.

3.5 Incorporating Network Structure

Many of the social models inspiring this work involve a network structure that dictates or restricts the interactions between agents [18]. It is natural to ask if the strategy classes discussed here can be extended to the scenario in which each agent is influenced only by his neighbors in a given network. Indeed, it is straightforward to extend each of the strategy classes introduced in this section to a network setting. For example, to adapt the crowd affinity and aversion strategy classes, it suffices to redefine f_a for each agent i to be the fraction of agents in the local neighborhood of agent i choosing action a. To adapt the agent affinity and aversion classes, it is necessary only to require that $w_j = 0$ for every agent j outside the local neighborhood of agent i. By making these simple modifications, the learning algorithms discussed in Section 5 can immediately be applied to settings in which a network structure is given.

4 A Reduction to I.I.D. Learning

Since algorithms in our framework are attempting to learn to model the dynamics of a factored Markov process in which each component is known to lie in the class $\mathcal{C}$, it is natural to investigate the relationship between learning just a single strategy in $\mathcal{C}$ and the entire Markovian dynamics. One main concern might be effects of dynamic instability — that is, that even small errors in models for each of the N components could be amplified exponentially in the overall population model.

In this section we show that this can be avoided. More precisely, we prove that if the component errors are all small compared to $1/(NT)^2$, the population model also has small error. Thus fast rates of learning for individual components are polynomially preserved in the resulting population model.

To show this, we give a reduction showing that if a class $\mathcal{C}$ of (possibly probabilistic) strategies is polynomially learnable (in a sense that we describe shortly) from I.I.D. data, then $\mathcal{C}$ is also polynomially learnable from collective behavior. The key step in the reduction is the introduction of the experimental distribution, defined below. Intuitively, the experimental distribution is meant to capture the distribution over states that are encountered in the collective setting over repeated trials. Polynomial I.I.D. learning on this distribution leads to polynomial learning from the collective.

4.1 A Reduction for Deterministic Strategies

In order to illustrate some of the key ideas we use in the more general reduction, we begin by examining the simple case in which the number of actions $K = 2$ and and each strategy $c \in \mathcal{C}$ is deterministic. We show that if $\mathcal{C}$ is polynomially learnable in the (distribution-free) PAC model, then $\mathcal{C}$ is polynomially learnable from collective behavior.

In order to exploit the fact that $\mathcal{C}$ is PAC learnable, it is first necessary to define a single distribution over states on which we would like to learn.

Definition 4 *For any initial state distribution P, strategy vector $\vec{c}$, and sequence length T, the* **experimental distribution** *$D_{P,\vec{c},T}$ is the distribution over state vectors $\vec{s}$ obtained by querying $\mathcal{O}_{\mathrm{EXP}}(\vec{c}, P, T)$ to obtain $\langle \vec{s}^{\,0}, \cdots, \vec{s}^{\,T} \rangle$, choosing t uniformly at random from $\{0, \cdots, T-1\}$, and setting $\vec{s} = \vec{s}^{\,t}$.*

We denote this distribution simply as D when P, $\vec{c}$, and T are clear from context. Given access to the oracle $\mathcal{O}_{\mathrm{EXP}}$, we can sample pairs $\langle \vec{s}, c_i(\vec{s}) \rangle$ where $\vec{s}$ is distributed according to D using the following procedure:

1. Query $\mathcal{O}_{\mathrm{EXP}}(\vec{c}, P, T)$ to obtain $\langle \vec{s}^{\,0}, \cdots, \vec{s}^{\,T} \rangle$.
2. Choose $t \in \{0, \cdots, T-1\}$ uniformly at random.
3. Return $\langle \vec{s}^{\,t}, s_i^{t+1} \rangle$.

If $\mathcal{C}$ is polynomially learnable in the PAC model, then by definition, with access to the oracle $\mathcal{O}_{\mathrm{EXP}}$, for any $\delta, \epsilon > 0$, it is possible to learn a model $\hat{c}_i$ such that with probability $1 - (\delta/N)$,

$$\mathrm{Pr}_{\vec{s} \sim D}[\hat{c}_i(\vec{s}) \neq c_i(\vec{s})] \leq \frac{\epsilon}{NT}$$

in time polynomial in N, T, $1/\epsilon$, $1/\delta$, and the VC dimension of $\mathcal{C}$ using the sampling procedure above; the dependence on N and T come from the fact that we are requesting a confidence of $1 - (\delta/N)$ and an accuracy of $\epsilon/(TN)$. We can learn a set of such strategies $\hat{c}_i$ for all agents i at the cost of an additional factor of N.

Consider a new sequence $\langle \vec{s}^{\,0}, \cdots, \vec{s}^{\,T} \rangle$ returned by the oracle $\mathcal{O}_{\mathrm{EXP}}$. By the union bound, with probability $1 - \delta$, the probability that there exists any agent i and any $t \in \{0, \cdots, T-1\}$, such that $\hat{c}_i(\vec{s}^{\,t}) \neq c_i(\vec{s}^{\,t})$ is less than ϵ. If this is not the case (i.e., if $\hat{c}_i(\vec{s}^{\,t}) = c_i(\vec{s}^{\,t})$ for all i and t) then the same sequence of states would have been reached if we had instead started at state $\vec{s}^{\,0}$ and generated each additional state $\vec{s}^{\,t}$ by letting $s_i^t = c_i(\vec{s}^{\,t-1})$. This implies that with probability $1 - \delta$, $\varepsilon(Q_{\hat{M}}, Q_{\vec{c}}) \leq \epsilon$, and $\mathcal{C}$ is polynomially learnable from collective behavior.

4.2 A General Reduction

Multiple analogs of the definition of learnability in the PAC model have been proposed for distribution learning settings. The probabilistic concept model [15] presents a definition for learning conditional distributions over binary outcomes, while later work [13] proposes a definition for learning unconditional distributions over larger outcome spaces. We combine the two into a single PAC-style model for learning conditional distributions over large outcome spaces from I.I.D. data as follows.

Definition 5 *Let $\mathcal{C}$ be a class of probabilistic mappings from an input $\vec{x} \in \mathcal{X}$ to an output $y \in \mathcal{Y}$ where $\mathcal{Y}$ is a finite set. We say that $\mathcal{C}$ is* **polynomially learnable** *if there exists an algorithm A such that for any $c \in \mathcal{C}$ and any distribution D over $\mathcal{X}$, if A is given access to an oracle producing pairs $\langle \vec{x}, c(\vec{x}) \rangle$ with x distributed according to D, then for any $\epsilon, \delta > 0$, algorithm A runs in time polynomial in $1/\epsilon$, $1/\delta$, and $dim(\mathcal{C})$ and outputs a function $\hat{c}$ such that with probability $1 - \delta$,*

$$\mathrm{E}_{\vec{x} \sim D}\left[\sum_{y \in \mathcal{Y}} |\mathrm{Pr}(c(\vec{x}) = y) - \mathrm{Pr}(\hat{c}(\vec{x}) = y)|\right] \leq \epsilon\ .$$

We could have chosen instead to require that the expected KL divergence between c and $\hat{c}$ be bounded. Using Jensen's inequality and Lemma 12.6.1 of Cover and Thomas [6], it is simple to show that if the expected KL divergence between two distributions is bounded by ϵ, then the expected $\mathcal{L}_1$ distance is bounded by $\sqrt{2\ln(2)\epsilon}$. Thus any class that is polynomially learnable under this alternate definition is also polynomially learnable under ours.

Theorem 6 *For any class $\mathcal{C}$, if $\mathcal{C}$ is polynomially learnable according to Definition 5, then $\mathcal{C}$ is polynomially learnable from collective behavior.*

Proof: This proof is very similar in spirit to the proof of the reduction for deterministic case. However, several tricks are needed to deal with the fact that trajectories are now random variables, even given a fixed start state. In particular, it is no longer the case that we can argue that starting at a given start state and executing a set of strategies that are "close to" the true strategy vector usually yields *the same* full trajectory we would have obtained by executing the true strategies of each agent. Instead, due to the inherent randomness in the strategies, we must argue that the *distribution* over trajectories is similar when the estimated strategies are sufficiently close to the true strategies.

To make this argument, we begin by introducing the idea of sampling from a distribution P_1 using a "filtered" version of a second distribution P_2 as follows. First, draw an outcome $\omega \in \Omega$ according to P_2. If $P_1(\omega) \leq P_2(\omega)$, output ω. Otherwise, output ω with probability $P_1(\omega)/P_2(\omega)$, and with probability $1 - P_1(\omega)/P_2(\omega)$, output an alternate action drawn according to a third distribution P_3, where

$$P_3(\omega) = \frac{P_1(\omega) - P_2(\omega)}{\sum_{\omega' : P_2(\omega') < P_1(\omega')} P_1(\omega') - P_2(\omega')}$$

if $P_1(\omega) > P_2(\omega)$, and $P_3(\omega) = 0$ otherwise.

It is easy to verify that the output of this filtering algorithm is indeed distributed according to P_1. Additionally, notice that the probability that the output is "filtered" is

$$\sum_{\omega:P_2(\omega)>P_1(\omega)} P_2(\omega)\left(1-\frac{P_1(\omega)}{P_2(\omega)}\right)=\frac{1}{2}||P_2-P_1||_1\ . \quad (1)$$

As in the deterministic case, we make use of the experimental distribution D as defined in Definition 4. If $\mathcal{C}$ is polynomially learnable as in Definition 5, then with access to the oracle $\mathcal{O}_{\text{EXP}}$, for any $\delta, \epsilon > 0$, it is possible to learn a model $\hat{c}_i$ such that with probability $1-(\delta/N)$,

$$\mathrm{E}_{\vec{s}\sim D}\left[\sum_{s\in\mathcal{S}}|\Pr(c_i(\vec{s})=s)-\Pr(\hat{c}_i(\vec{s})=s)|\right]\leq\left(\frac{\epsilon}{NT}\right)^2 \quad (2)$$

in time polynomial in N, T, $1/\epsilon$, $1/\delta$, and $dim(\mathcal{C})$ using the three-step sampling procedure described in the deterministic case; as before, the dependence on N and T stem from the fact that we are requesting a confidence of $1-(\delta/N)$ and an accuracy that is polynomial in both N and T. It is possible learn a set of such strategies $\hat{c}_i$ for all agents i at the cost of an additional factor of N.

If Equation 2 is satisfied for agent i, then for any $\tau \geq 1$, the probability of drawing a state $\vec{s}$ from D such that

$$\sum_{s\in\mathcal{S}}|\Pr(c_i(\vec{s})=s)-\Pr(\hat{c}_i(\vec{s})=s)|\geq\tau\left(\frac{\epsilon}{NT}\right)^2 \quad (3)$$

is no more than $1/\tau$.

Consider a new sequence $\langle\vec{s}^{\,0},\cdots,\vec{s}^{\,T}\rangle$ returned by the oracle $\mathcal{O}_{\text{EXP}}$. For each $\vec{s}^{\,t}$, consider the action s_i^{t+1} chosen by agent i. This action was chosen according to the distribution c_i. Suppose instead we would like to choose this action according to the distribution $\hat{c}_i$ using a filtered version of c_i as described above. By Equation 1, the probability that the action choice of c_i is "filtered" (and thus not equal to s_i^{t+1}) is half the $\mathcal{L}_1$ distance between $c_i(\vec{s}^{\,t})$ and $\hat{c}_i(\vec{s}^{\,t})$. From Equation 3, we know that for any $\tau \geq 1$, with probability at least $1-1/\tau$, this probability is less than $\tau(\epsilon/(NT))^2$, so the probability of the new action being different from s_i^{t+1} is less than $\tau(\epsilon/(NT))^2+1/\tau$. This is minimized when $\tau = 2NT/\epsilon$, giving us a bound of $\epsilon/(NT)$.

By the union bound, with probability $1-\delta$, the probability that there exists any agent i and any $t \in \{1,\cdots,T\}$, such that s_i^{t+1} is not equal to the action we get by sampling $\hat{c}_i(\vec{s}^{\,t})$ using the filtered version of c_i must then be less than ϵ. As in the deterministic version, if this is *not* the case, then the same sequence of states would have been reached if we had instead started at state $\vec{s}^{\,0}$ and generated each additional state $\vec{s}^{\,t}$ by letting $s_i^t = \hat{c}_i(\vec{s}^{\,t-1})$ filtered using c_i. This implies that with probability $1-\delta$, $\varepsilon(Q_{\hat{M}}, Q_{\vec{c}}) \leq \epsilon$, and $\mathcal{C}$ is polynomially learnable from collective behavior. ■

5 Learning Social Strategy Classes

We now turn our attention to efficient algorithms for learning some of the specific social strategy classes introduced in Section 3. We focus on the two crowd affinity model classes. Recall that these classes are designed to model the scenario in which each agent has an intrinsic set of preferences over actions, but simultaneously would prefer to choose the same actions chosen by other agents. Similar techniques can be applied to learn the crowd aversion strategies.

Formally, let $\vec{f}$ be a vector representing the distribution over current states of the agents; if $\vec{s}$ is the current state, then for each action a, $f_a = |\{i : s_i = a\}|/N$ is the fraction of the population currently choosing action a. (Alternately, if there is a network structure governing interaction among agents, f_a can be defined as the fraction of nodes in an agent's local neighborhood choosing action a.) We denote by D^f the distribution over vectors $\vec{f}$ induced by the experimental distribution D over state vectors $\vec{s}$. In other words, the probability of a vector $\vec{f}$ under D^f is the sum over all state vectors $\vec{s}$ mapping to $\vec{f}$ of the probability of $\vec{s}$ under D.

We focus on the problem of learning the parameters of the strategy of a single agent i in each of the models. We assume that we are presented with a set of samples $\mathcal{M}$, where each instance $\mathcal{I}_m \in \mathcal{M}$ consists of a pair $\langle\vec{f}_m, a_m\rangle$. Here $\vec{f}_m$ is the distribution over states of the agents and a_m is the next action chosen by agent i. We assume that the state distributions $\vec{f}_m$ of these samples are distributed according to D^f. Given access to the oracle $\mathcal{O}_{\text{EXP}}$, such samples could be collected, for example, using a three-step procedure like the one in Section 4.1. We show that each class is polynomially learnable with respect to the distribution D^f induced by *any* distribution D over states, and so by Theorem 6, also polynomially learnable from collective behavior.

While it may seem wasteful to gather only one data instance for each agent i from each T-trajectory, we remark that only small, isolated pieces of the analysis presented in this section rely on the assumption that the state distributions of the samples are distributed according to D^f. In practice, the entire trajectories could be used for learning with no impact on the structure of the algorithms. Additionally, while the analysis here is geared towards learning under the experimental distribution, the algorithms we present can be applied without modification in the no-reset variant of the model introduced in Section 2.3. We briefly discuss how to extend the analysis to the no-reset variant in Section 5.3.

5.1 Learning Crowd Affinity Mixture Models

In Section 3.1, we introduced the class of crowd affinity mixture model strategies. Such strategies are parameterized by a (normalized) weight vector $\vec{w}$ and parameter $\alpha \in [0, 1]$. The probability that agent i chooses action a given that the current state distribution is $\vec{f}$ is then $\alpha f_a + (1-\alpha)w_a$. In this section, we show that this class of strategies is polynomially learnable from collective behavior and sketch an algorithm for learning estimates of the parameters α and $\vec{w}$.

Let $\mathrm{I}(x)$ be the indicator function that is 1 if x is true and 0 otherwise. From the definition of the model it is easy to see that for any m such that $\mathcal{I}_m \in \mathcal{M}$, for any action $a \in \mathcal{S}$, $\mathrm{E}[\mathrm{I}(a_m = a)] = \alpha f_a + (1-\alpha)w_a$, where the expectation is over the randomness in the agent's strategy. By linearity of expectation,

$$\mathrm{E}\left[\sum_{m:\mathcal{I}_m\in\mathcal{M}}\mathrm{I}(a_m=a)\right]=\alpha\sum_{m:\mathcal{I}_m\in\mathcal{M}}f_{m,a}+(1-\alpha)w_a|\mathcal{M}|\ . \quad (4)$$

Standard results from uniform convergence theory say that we can approximate the left-hand side of this equation arbitrarily well given a sufficiently large data set $\mathcal{M}$. Replacing the expectation with this approximation in Equation 4 yields a single equation with two unknown variables, α and w_a. To solve for these variables, we must construct a *pair* of equations with two unknown variables. We do so by splitting the data into instances where $f_{m,a}$ is "high" and instances where it is "low."

Specifically, let $M = |\mathcal{M}|$. For convenience of notation, assume without loss of generality that M is even; if M is odd, simply discard an instance at random. Define $\mathcal{M}_a^{low}$ to be the set containing the $M/2$ instances in $\mathcal{M}$ with the lowest values of $f_{m,a}$. Similarly, define $\mathcal{M}_a^{high}$ to be the set containing the $M/2$ instances with the highest values of $f_{m,a}$. Replacing $\mathcal{M}$ with $\mathcal{M}_a^{low}$ and $\mathcal{M}_a^{high}$ respectively in Equation 4 gives us two linear equations with two unknowns. As long as these two equations are linearly independent, we can solve the system of equations for α, giving us

$$\alpha = \frac{\mathrm{E}\left[\sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{high}} \mathrm{I}(a_m = a) - \sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{low}} \mathrm{I}(a_m = a)\right]}{\sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{high}} f_{m,a} - \sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{low}} f_{m,a}} .$$

We can approximate α from data in the natural way, using

$$\hat{\alpha} = \frac{\sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{high}} \mathrm{I}(a_m = a) - \sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{low}} \mathrm{I}(a_m = a)}{\sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{high}} f_{m,a} - \sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{low}} f_{m,a}} . \quad (5)$$

By Hoeffding's inequality and the union bound, for any $\delta > 0$, with probability $1 - \delta$,

$$\begin{aligned} |\alpha - \hat{\alpha}| &\leq \frac{\sqrt{\ln(4/\delta) M}}{\sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{high}} f_{m,a} - \sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{low}} f_{m,a}} \\ &= (1/Z_a)\sqrt{\ln(4/\delta)/M} , \end{aligned} \quad (6)$$

where

$$Z_a = \frac{1}{M/2} \sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{high}} f_{m,a} - \frac{1}{M/2} \sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{low}} f_{m,a} .$$

The quantity Z_a measures the difference between the mean value of $f_{m,a}$ among instances with "high" values of $f_{m,a}$ and the mean value of $f_{m,a}$ among instances with "low" values. While this quantity is data-dependent, standard uniform convergence theory tells us that it is stable once the data set is large. From Equation 6, we know that if there is an action a for which this difference is sufficiently high, then it is possible to obtain an accurate estimate of α given enough data. If, on the other hand, no such a exists, it follows that there is very little variance in the population distribution over the sample. We argue below that it is not necessary to learn α in order to mimic the behavior of an agent i if this is the case.

For now, assume that Z_a is sufficiently large for at least one value of a, and call this value a^*. We can use the estimate of α to obtain estimates of the weights for each action. From Equation 4, it is clear that for any a,

$$w_a = \frac{\mathrm{E}\left[\sum_{m:\mathcal{I}_m \in \mathcal{M}} \mathrm{I}(a_m = a)\right] - \alpha \sum_{m:\mathcal{I}_m \in \mathcal{M}} f_{m,a}}{(1-\alpha)M} .$$

We estimate this weight using

$$\hat{w}_a = \frac{\sum_{m:\mathcal{I}_m \in \mathcal{M}} \mathrm{I}(a_m = a) - \hat{\alpha} \sum_{m:\mathcal{I}_m \in \mathcal{M}} f_{m,a}}{(1-\hat{\alpha})M} . \quad (7)$$

The following lemma shows that given sufficient data, the error in these estimates is small when Z_{a^*} is large.

Lemma 7 *Let $a^* = \mathrm{argmax}_{a \in \mathcal{S}} Z_a$, and let $\hat{\alpha}$ be calculated as in Equation 5 with $a = a^*$. For each $a \in \mathcal{S}$, let $\hat{w}_a$ be calculated as in Equation 7. For sufficiently large M, for any $\delta > 0$, with probability $1 - \delta$,*

$$|\alpha - \hat{\alpha}| \leq (1/Z_{a^*})\sqrt{\ln((4+2K)/\delta)/M} ,$$

and for all actions a,

$$\begin{aligned} &|w_a - \hat{w}_a| \\ &\leq \frac{((1-\hat{\alpha})Z_{a^*}/\sqrt{2} + 2)\sqrt{\ln((4+2K)/\delta)}}{Z_{a^*}(1-\hat{\alpha})^2\sqrt{M} - (1-\hat{\alpha})\sqrt{\ln((4+2K)/\delta)}} . \end{aligned}$$

The proof of this lemma, which is in the appendix, relies heavily on the following technical lemma for bounding the error of estimated ratios, which is used frequently throughout the remainder of the paper.

Lemma 8 *For any positive u, $\hat{u}$, v, $\hat{v}$, k, and ϵ such that $\epsilon k < v$, if $|u - \hat{u}| \leq \epsilon$ and $|v - \hat{v}| \leq k\epsilon$, then*

$$\left|\frac{u}{v} - \frac{\hat{u}}{\hat{v}}\right| \leq \frac{\epsilon(v + uk)}{v(v - \epsilon k)} .$$

Now that we have bounds on the error of the estimated parameters, we can bound the expected $\mathcal{L}_1$ distance between the estimated model and the real model.

Lemma 9 *For sufficiently large M,*

$$\begin{aligned} &\mathrm{E}_{\vec{f} \sim D^f} \sum_{a \in \mathcal{S}} |(\alpha f_a + (1-\alpha)w_a) - (\hat{\alpha} f_a + (1-\hat{\alpha})\hat{w}_a)| \\ &\leq \frac{2\sqrt{\ln((4+2K)/\delta)}}{Z_{a^*}\sqrt{M}} \\ &\quad + \min\left\{ \frac{K(Z_{a^*}/\sqrt{2} + 2)\sqrt{\ln((4+2K)/\delta)}}{Z_{a^*}(1-\hat{\alpha})\sqrt{M} - \sqrt{\ln((4+2K)/\delta)}}, \right. \\ &\qquad \left. 2(1-\hat{\alpha}) \right\} . \end{aligned}$$

In this proof of this lemma, which appears in the appendix, the quantity

$$\sum_{a \in \mathcal{S}} |(\alpha f_a + (1-\alpha)w_a) - (\hat{\alpha} f_a + (1-\hat{\alpha})\hat{w}_a)|$$

is bounded *uniformly* for all $\vec{f}$ using the error bounds. The bound on the expectation follows immediately.

It remains to show that we can still bound the error when Z_{a^*} is zero or very close to zero. We present a light sketch of the argument here; more details appear in the appendix.

Let η_a and μ_a be the true median and mean of the distribution from which the random variables $f_{m,a}$ are drawn. Let f_a^{high} be the mean value of the distribution over $f_{m,a}$

conditioned on $f_{m,a} > \eta_a$. Let $\bar{f}_a^{high}$ be the empirical average of $f_{m,a}$ conditioned on $f_{m,a} > \eta_a$. Finally, let $\hat{f}_a^{high} = (2/M)\sum_{m:\mathcal{I}_m\in\mathcal{M}_a^{high}} f_{m,a}$ be the empirical average of $f_{m,a}$ conditioned on $f_{m,a}$ being greater than the *empirical* median. We can calculate $\hat{f}_a^{high}$ from data.

We can apply standard arguments from uniform convergence theory to show that f_a^{high} is close to $\bar{f}_a^{high}$, and in turn that $\bar{f}_a^{high}$ is close to $\hat{f}_a^{high}$. Similar statements can be made for the analogous quantities f_a^{low}, $\bar{f}_a^{low}$, and $\hat{f}_a^{low}$. By noting that $Z_a = \hat{f}_a^{high} - \hat{f}_a^{low}$ this implies that if Z_a is small, then the probability that a random value of $f_{m,a}$ is far from the mean μ_a is small. When this is the case, it is not necessary to estimate α directly. Instead, we set $\hat{\alpha} = 0$ and

$$\hat{w_a} = \frac{1}{M}\sum_{m:\mathcal{I}_m\in\mathcal{M}} I(a_m = a)\ .$$

Applying Hoeffding's inequality again, it is easy to show that for each a, $\hat{w_a}$ is very close to $\alpha\mu_a + (1-\alpha)w_a$, and from here it can be argued that the $\mathcal{L}_1$ distance between the estimated model and the real model is small.

Thus for any distribution D over state vectors, regardless of the corresponding value of Z_{a^*}, it is possible to build an accurate model for the strategy of agent i in polynomial time. By Theorem 6, this implies that the class is polynomially learnable from collective behavior.

Theorem 10 *The class of crowd affinity mixture model strategies is polynomially learnable from collective behavior.*

5.2 Learning Crowd Affinity Multiplicative Models

In Section 3.2, we introduced the crowd affinity multiplicative model. In this model, strategies are parameterized only by a weight vector $\vec{w}$. The probability that agent i chooses action a is simply $f_a w_a / \sum_{b\in\mathcal{S}} f_b w_b$.

Although the motivation for this model is similar to that for the mixture model, the dynamics of the system are quite different (see the simulations and discussion in Section 3), and a very different algorithm is necessary to learn individual strategies. In this section, we show that this class is polynomially learnable from collective behavior, and sketch the corresponding learning algorithm. The algorithm we present is based on a simple but powerful observation. In particular, consider the following random variable:

$$\chi_a^m = \begin{cases} 1/f_{m,a} & \text{if } f_{m,a} > 0 \text{ and } a_m = a\ , \\ 0 & \text{otherwise}\ . \end{cases}$$

Suppose that for all m such that $\mathcal{I}_m \in \mathcal{M}$, it is the case that $f_{m,a} > 0$. Then by the definition of the strategy class and linearity of expectation,

$$\begin{aligned} \mathrm{E}\left[\sum_{m:\mathcal{I}_m\in\mathcal{M}} \chi_a^m\right] &= \sum_{m:\mathcal{I}_m\in\mathcal{M}} \frac{1}{f_{m,a}}\left(\frac{f_{m,a}w_a}{\sum_{s\in\mathcal{S}} f_{m,s}w_s}\right) \\ &= w_a \sum_{m:\mathcal{I}_m\in\mathcal{M}} \frac{1}{\sum_{s\in\mathcal{S}} f_{m,s}w_s}\ , \end{aligned}$$

where the expectation is over the randomness in the agent's strategy. Notice that this expression is the product of two terms. The first, w_a, is precisely the value we would like to calculate. The second term is something that depends on the set of instances $\mathcal{M}$, but *does not* depend on action a. This leads to the key observation at the core of our algorithm. Specifically, if we have a second action b such that $f_{m,b} > 0$ for all m such that $\mathcal{I}_m \in \mathcal{M}$, then

$$\frac{w_a}{w_b} = \frac{\mathrm{E}\left[\sum_{m:\mathcal{I}_m\in\mathcal{M}} \chi_a^m\right]}{\mathrm{E}\left[\sum_{m:\mathcal{I}_m\in\mathcal{M}} \chi_b^m\right]}\ .$$

Although we do not know the values of these expectations, we can approximate them arbitrarily well given enough data. Since we have assumed (so far) that $f_{m,a} > 0$ for all $m \in \mathcal{M}$, and we know that $f_{m,a}$ represents a fraction of the population, it must be the case that $f_{m,a} \geq 1/N$ and $\chi_a^m \in [0, N]$ for all m. By a standard application of Hoeffding's inequality and the union bound, we see that for any $\delta > 0$, with probability $1-\delta$,

$$\left|\sum_{m:\mathcal{I}_m\in\mathcal{M}} \chi_a^m - \mathrm{E}\left[\sum_{m:\mathcal{I}_m\in\mathcal{M}} \chi_a^m\right]\right| \leq \sqrt{\frac{N\ln(2/\delta)}{2|\mathcal{M}|}}\ . \quad (8)$$

This leads to the following lemma. We note that the role of β in this lemma may appear somewhat mysterious. It comes the fact that we are bounding the error of a ratio of two terms; an application of Lemma 8 using the bound in Equation 8 gives us a factor of $\chi_{a,b} + \chi_{b,a}$ in the numerator and a factor of $\chi_{b,a}$ in the denominator. This is problematic only when $\chi_{a,b}$ is significantly larger than $\chi_{b,a}$. The full proof appears in the appendix.

Lemma 11 *Suppose that $f_{m,a} > 0$ and $f_{m,b} > 0$ for all m such that $\mathcal{I}_m \in \mathcal{M}$. Then for any $\delta > 0$, with probability $1-\delta$, for any $\beta > 0$, if $\chi_{a,b} \leq \beta\chi_{b,a}$ and $\chi_{b,a} \geq 1$, then if $|\mathcal{M}| \geq N\ln(2/\delta)/2$, then*

$$\left|\frac{w_a}{w_b} - \frac{\sum_{m:\mathcal{I}_m\in\mathcal{M}} \chi_a^m}{\sum_{m:\mathcal{I}_m\in\mathcal{M}} \chi_b^m}\right| \leq \frac{(1+\beta)\sqrt{N\ln(2/\delta)}}{\sqrt{2|\mathcal{M}|} - \sqrt{N\ln(2/\delta)}}\ .$$

If we are fortunate enough to have a sufficient number of data instances for which $f_{m,a} > 0$ for all $a \in \mathcal{S}$, then this lemma supplies us with a way of approximating the ratios between all pairs of weights and subsequently approximating the weights themselves. In general, however, this may not be the case. Luckily, it is possible to estimate the ratio of the weights of each pair of actions a and b that are used together frequently by the population using only those data instances in which at least one agent is choosing each. Formally, define

$$\mathcal{M}_{a,b} = \{\mathcal{I}_m \in \mathcal{M} : f_{m,a} > 0, f_{m,b} > 0\}\ .$$

Lemma 11 tells us that if $\mathcal{M}_{a,b}$ is sufficiently large, and there is at least one instance $\mathcal{I}_m \in \mathcal{M}_{a,b}$ for which $a_m = b$, then we can approximate the ratio between w_a and w_b well.

What if one of these assumptions does not hold? If we are not able to collect sufficiently many instances in which $f_{m,a} > 0$ and $f_{m,b} > 0$, then standard uniform convergence results can be used to show that it is very unlikely that we see a new instance for which $f_a > 0$ and $f_b > 0$. This idea is formalized in the following lemma, the proof of which is in the appendix.

Lemma 12 *For any $M < |\mathcal{M}|$, for any $\delta \in (0,1)$, with probability $1-\delta$,*

$$\Pr_{\vec{f}\sim D^f}\left[\exists a,b \in \mathcal{S} : f_a > 0, f_b > 0, |\mathcal{M}_{a,b}| < M\right] \leq \frac{K^2}{2}\left(\frac{M}{|\mathcal{M}|} + \sqrt{\frac{\ln(K^2/(2\delta))}{2|\mathcal{M}|}}\right) .$$

Similarly, if $\chi_{a,b} = \chi_{b,a} = 0$, then a standard uniform convergence argument can be used to show that it is unlikely that agent i would ever select action a or b when $f_{m,a} > 0$ and $f_{m,b} > 0$. We will see that in this case, it is not important to learn the ratio between these two weights.

Using these observations, we can accurately model the behavior of agent i. The model consists of two phases. First, as a preprocessing step, we calculate a quantity

$$\chi_{a,b} = \sum_{m:\mathcal{I}_m \in \mathcal{M}_{a,b}} \chi_a^m$$

for each pair $a, b \in \mathcal{S}$. Then, each time we are presented with a state $\vec{f}$, we calculate a set of weights for all actions a with $f_a > 0$ on the fly.

For a fixed $\vec{f}$, let $\mathcal{S}'$ be the set of actions $a \in \mathcal{S}$ such that $f_a > 0$. By Lemma 12, if the data set is sufficiently large, then we know that with high probability, it is the case that for all $a, b \in \mathcal{S}'$, $|\mathcal{M}_{a,b}| \geq M$ for some threshold M.

Now, let $a^* = \operatorname{argmax}_{a\in\mathcal{S}'} |\{b : b \in \mathcal{S}', \chi_{a,b} \geq \chi_{b,a}\}|$. Intuitively, if there is sufficient data, a^* should be the action in $\mathcal{S}'$ with the highest weight, or have a weight arbitrarily close to the highest. Thus for any $a \in \mathcal{S}'$, Lemma 11 can be used to bound our estimate of w_a/w_{a^*} with a value of β arbitrarily close to 1. Noting that

$$\frac{w_a}{\sum_{s\in\mathcal{S}'} w_s} = \frac{w_a/w_{a^*}}{\sum_{s\in\mathcal{S}'} w_s/w_{a^*}} ,$$

we approximate the *relative* weight of action $a \in \mathcal{S}'$ with respect to the other actions in $\mathcal{S}'$ using

$$\hat{w}_a = \frac{\chi_{a,a^*}/\chi_{a^*,a}}{\sum_{s\in\mathcal{S}'} \chi_{s,a*}/\chi_{a^*,s}} ,$$

and simply let $\hat{w}_a = 0$ for any $a \notin \mathcal{S}'$. Applying Lemma 8, we find that for all $a \in \mathcal{S}'$, with high probability,

$$\left|\frac{w_a}{\sum_{s\in\mathcal{S}'} w_s} - \hat{w}_a\right| \leq \frac{(1+\beta)K\sqrt{N\ln(2K^2/\delta)}}{\sqrt{2M} - (1+\beta)K\sqrt{N\ln(2K^2/\delta)}} , \quad (9)$$

where M is the lower bound on $|\mathcal{M}_{a,b}|$ for all $a, b \in \mathcal{S}'$, and β is close to 1. With this bound in place, it is straightforward to show that we can apply Lemma 8 once more to bound the expected $\mathcal{L}_1$,

$$\mathrm{E}_{\vec{f}\sim D^f}\left[\sum_{a\in\mathcal{S}}\left|\frac{w_a f_a}{\sum_{s\in\mathcal{S}} w_s f_s} - \frac{\hat{w}_a f_a}{\sum_{s\in\mathcal{S}} \hat{w}_s f_s}\right|\right] ,$$

and that the bound goes to 0 at a rate of $O(1/\sqrt{M})$ as the threshold M grows. More details are given in the appendix.

Since it is possible to build an accurate model of the strategy of agent i in polynomial time under any distribution D over state vectors, we can again apply Theorem 6 to see that this class is polynomially learnable from collective behavior.

Theorem 13 *The class of crowd affinity multiplicative model strategies is polynomially learnable from collective behavior.*

5.3 Learning Without Resets

Although the analyses in the previous subsections are tailored to learnability in the sense of Definition 2, they can easily be adapted to hold in the alternate setting in which the learner has access only to a single, unbroken trajectory of states. In this alternate model, the learning algorithm observes a polynomially long prefix of a trajectory of states for training, and then must produce a generative model which results in a distribution over the values of the subsequent T states close to the true distribution.

When learning individual crowd affinity models for each agent in this setting, we again assume that we are presented with a set of samples $\mathcal{M}$, where each instance $\mathcal{I}_m \in \mathcal{M}$ consists of a pair $\langle \vec{f}_m, a_m\rangle$. However, instead of assuming that the state distributions $\vec{f}_m$ are distributed according to D^f, we now assume that the state and action pairs represent a single trajectory. As previously noted, the majority of the analysis for both the mixture and multiplicative variants of the crowd affinity model does not depend on the particular way in which state distribution vectors are distributed, and thus carries over to this setting as is. Here we briefly discuss the few modifications that are necessary.

The only change required in the analysis of the crowd affinity mixture model relates to handling the case in which Z_a is small for all a. Previously we argued that when this is the case, the distribution D^f must be concentrated so that for all a, f_a falls within a very small range with high probability. Thus it is not necessary to estimate the parameter α directly, and we can instead learn a single probability for each action that is used regardless of $\vec{f}$. A similar argument holds in the no-reset variant. If it is the case that Z_a is small for all a, then it must be the case that for each a, the value of f_a has fallen into the same small range for the entire observed trajectory. A standard uniform convergence argument says that the probability that f_a suddenly changes dramatically is very small, and thus again it is sufficient to learn a single probability for each action that is used regardless of $\vec{f}$.

To adapt the analysis of the crowd affinity multiplicative model, it is first necessary to replace Lemma 12. Recall that the purpose of this lemma was to show that when the data set does not contain sufficient samples in which $f_a > 0$ and $f_b > 0$ for a pair of actions a and b, the chance of observing a new state distribution $\vec{f}$ with $f_a > 0$ and $f_b > 0$ is small. This argument is actually much more straightforward in the no-reset case. By the definition of the model, it is easy to see that if $f_a > 0$ for some action a at time t in a trajectory, then it must be the case that $f_a > 0$ at all previous points in the trajectory. Thus if $f_a > 0$ on any test instance, then f_a must have been non-negative on *every* training instance, and we do not have to worry about the case in which there is insufficient data to compare the weights of a particular pair of actions.

One additional, possibly more subtle, modification is necessary in the analysis of the multiplicative model to handle the case in which $\chi_{a,b} = \chi_{b,a} = 0$ for all "active" pairs of actions $a, b \in \mathcal{S}'$. This can happen only if agent i has extremely small weights for every action in $\mathcal{S}'$, and had previously been choosing an alternate action that is no longer available, i.e., an action s for which f_s had previously been non-negative but suddenly is not. However, in order for f_s to become 0, it must be the case that agent i himself chooses an alternate action (say, action a) instead of s, which cannot happen since the estimated weight of action a used by the model is 0. Thus this situation can never occur in the no-reset variant.

6 Conclusions and Future Work

We have introduced a computational model for learning from collective behavior, and populated it with some initial general theory and algorithmic results for crowd affinity models. In addition to positive or negative results for further agent strategy classes, there are a number of other general directions of interest for future research. These include extension of our model to agnostic [14] settings, in which we relax the assumption that every agent strategy falls in a known class, and to reinforcement learning [23] settings, in which the learning algorithm may itself be a member of the population being modeled, and wishes to learn an optimal policy with respect to some reward function.

Acknowledgments

We thank Nina Balcan and Eyal Even-Dar for early discussions on models of social learning, and Duncan Watts for helpful conversations and pointers to relevant literature.

References

[1] S. Bikhchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy*, 100:992–1026, 1992.

[2] S. Bikhchandani, D. Hirshleifer, and I. Welch. Learning from the behavior of others: Conformity, fads, and informational cascades. *The Journal of Economic Perspectives*, 12:151–170, 1998.

[3] G. W. Brown. Iterative solutions of games by fictitious play. In T.C. Koopmans, editor, *Activity Analysis of Production and Allocation*. Wiley, 1951.

[4] T. Bylander. Learning noisy linear threshold functions. Technical Report, 1998.

[5] E. Cohen. Learning noisy perceptrons by a perceptron in polynomial time. In *38th IEEE Annual Symposium on Foundations of Computer Science*, 1997.

[6] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, NY, 1991.

[7] P. Dodds, R. Muhamad, and D. J. Watts. An experimental study of search in global social networks. *Science*, 301:828–829, August 2003.

[8] M. Drehmann, J. Oechssler, and A. Roider. Herding and contrarian behavior in financial markets: An Internet experiment. *American Economic Review*, 95(5):1403–1426, 2005.

[9] D. Foster and R. Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29:7–35, 1999.

[10] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 61:1420–1443, 1978.

[11] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.

[12] P. Hedstrom. Rational imitation. In P. Hedstrom and R. Swedberg, editors, *Social Mechanisms: An Analytical Approach to Social Theory*. Cambridge University Press, 1998.

[13] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R.E. Schapire, and L. Sellie. On the learnability of discrete distributions. In *26th Annual ACM Symposium on Theory of Computing*, pages 273–282, 1994.

[14] M. Kearns, R. Schapire, and L. Sellie. Towards efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.

[15] M. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.

[16] M. Kearns and S. Singh. Finite-sample rates of convergence for Q-learning and indirect methods. In *Advances in Neural Information Processing Systems 11*, 1999.

[17] M. Kearns, S. Suri, and N. Montfort. A behavioral study of the coloring problem on human subject networks. *Science*, 313(5788):824–827, 2006.

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

[19] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[20] M. Salganik, P. Dodds, and D. J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 331(5762):854–856, 2006.

[21] M. Salganik and D. J. Watts. Social influence, manipulation, and self-fulfilling prophecies in cultural markets. Preprint, 2007.

[22] T. Schelling. *Micromotives and Macrobehavior*. Norton, New York, NY, 1978.

[23] R. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, 1998.

[24] A. De Vany. *Hollywood Economics: How Extreme Uncertainty Shapes the Film Industry*. Routledge, London, 2004.

[25] A. De Vany and C. Lee. Quality signals in information cascades and the dynamics of the distribution of motion picture box office revenues. *Journal of Economic Dynamics and Control*, 25:593–614, 2001.

[26] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.

[27] I. Welch. Herding among security analysts. *Journal of Financial Economics*, 58:369–396, 2000.

A Proofs from Section 5.1

A.1 Proof of Lemma 8

For the first direction,

$$\begin{aligned}\frac{u}{v}-\frac{\hat{u}}{\hat{v}} &\leq \frac{u}{v}-\frac{u-\epsilon}{v+k\epsilon}=\frac{u}{v}-\frac{uv-\epsilon v}{v(v+k\epsilon)}\\ &= \frac{u}{v}-\frac{uv+uk\epsilon-\epsilon v-uk\epsilon}{v(v+k\epsilon)}\\ &= \frac{u}{v}-\frac{u(v+k\epsilon)-\epsilon(v+uk)}{v(v+k\epsilon)}\\ &= \frac{\epsilon(v+uk)}{v(v+\epsilon k)}\leq\frac{\epsilon(v+uk)}{v(v-\epsilon k)}\ .\end{aligned}$$

Similarly,

$$\begin{aligned}\frac{\hat{u}}{\hat{v}}-\frac{u}{v} &\leq \frac{u+\epsilon}{v-k\epsilon}-\frac{u}{v}=\frac{uv+\epsilon v}{v(v-k\epsilon)}-\frac{u}{v}\\ &= \frac{uv-uk\epsilon+\epsilon v+uk\epsilon}{v(v-k\epsilon)}-\frac{u}{v}\\ &= \frac{u(v-k\epsilon)+\epsilon(v+uk)}{v(v-k\epsilon)}-\frac{u}{v}\\ &= \frac{\epsilon(v+uk)}{v(v-\epsilon k)}\ .\end{aligned}$$

A.2 Proof of Lemma 7

We bound the error of these estimations in two parts. First, since from Equation 6 we know that for any $\delta_1 > 0$, with probability $1-\delta_1$,

$$\begin{aligned}&\left|\hat{\alpha}\sum_{m:\mathcal{I}_m\in\mathcal{M}} f_{m,a}-\alpha\sum_{m:\mathcal{I}_m\in\mathcal{M}} f_{m,a}\right|\\ &\leq \frac{1}{Z_{a^*}}\sqrt{\frac{\ln(4/\delta_1)}{M}}\sum_{m:\mathcal{I}_m\in\mathcal{M}} f_{m,a}\leq\frac{1}{Z_{a^*}}\sqrt{M\ln(4/\delta_1)}\ ,\end{aligned}$$

and

$$|(1-\hat{\alpha})M-(1-\alpha)M|\leq\frac{1}{Z_{a^*}}\sqrt{M\ln(4/\delta_1)}\ ,$$

we have by Lemma 8 that for sufficiently large M

$$\begin{aligned}&\left|\frac{\hat{\alpha}\sum_{m:\mathcal{I}_m\in\mathcal{M}} f_{m,a}}{(1-\hat{\alpha})M}-\frac{\alpha\sum_{m:\mathcal{I}_m\in\mathcal{M}} f_{m,a}}{(1-\alpha)M}\right|\\ &\leq \frac{(1/Z_{a^*})\sqrt{M\ln(4/\delta_1)}\left((1-\hat{\alpha})M+\hat{\alpha}\sum_{m:\mathcal{I}_m\in\mathcal{M}} f_{m,a}\right)}{(1-\hat{\alpha})^2M^2-(1-\hat{\alpha})M\sqrt{M\ln(4/\delta_1)}/Z_{a^*}}\\ &\leq \frac{\sqrt{M\ln(4/\delta_1)}((1-\hat{\alpha})M+\hat{\alpha}M)}{Z_{a^*}(1-\hat{\alpha})^2M^2-(1-\hat{\alpha})M\sqrt{M\ln(4/\delta_1)}}\\ &= \frac{\sqrt{\ln(4/\delta_1)}}{Z_{a^*}(1-\hat{\alpha})^2\sqrt{M}-(1-\hat{\alpha})\sqrt{\ln(4/\delta_1)}}\ .\end{aligned}$$

Now, by Hoeffding's inequality and the union bound, for any $\delta_2 > 0$, with probability $1-\delta_2$, for all a,

$$\begin{aligned}&\left|\sum_{m:\mathcal{I}_m\in\mathcal{M}}\mathrm{I}(a_m=a)-\mathrm{E}\left[\sum_{m:\mathcal{I}_m\in\mathcal{M}}\mathrm{I}(a_m=a)\right]\right|\\ &\leq \sqrt{M\ln(2K/\delta_2)/2}\ .\end{aligned}$$

Setting $\delta_2 = K\delta_1/2$, we can again apply Lemma 8 and see that for sufficiently large M

$$\begin{aligned}&\left|\frac{\sum_{m:\mathcal{I}_m\in\mathcal{M}}\mathrm{I}(a_m=a)}{(1-\hat{\alpha})M}-\frac{\mathrm{E}\left[\sum_{m:\mathcal{I}_m\in\mathcal{M}}\mathrm{I}(a_m=a)\right]}{(1-\alpha)M}\right|\\ &\leq \frac{\sqrt{M\ln(4/\delta_1)/2}\left((1-\hat{\alpha})Z_{a^*}M+\sqrt{2}M\right)}{Z_{a^*}(1-\hat{\alpha})^2M^2-(1-\hat{\alpha})M\sqrt{M\ln(4/\delta_1)}}\\ &\leq \frac{\sqrt{\ln(4/\delta_1)/2}\left((1-\hat{\alpha})Z_{a^*}+\sqrt{2}\right)}{Z_{a^*}(1-\hat{\alpha})^2\sqrt{M}-(1-\hat{\alpha})\sqrt{\ln(4/\delta_1)}}\\ &= \frac{((1-\hat{\alpha})Z_{a^*}/\sqrt{2}+1)\sqrt{\ln(4/\delta_1)}}{Z_{a^*}(1-\hat{\alpha})^2\sqrt{M}-(1-\hat{\alpha})\sqrt{\ln(4/\delta_1)}}\ .\end{aligned}$$

Setting $\delta_1 = \delta/(1+K/2)$ and applying the union bound yields the lemma.

A.3 Proof of Lemma 9

As long as M is sufficiently large, for any fixed $\vec{f}$,

$$\begin{aligned}&\sum_{a\in\mathcal{S}}|(\alpha f_a+(1-\alpha)w_a)-(\hat{\alpha}f_a+(1-\hat{\alpha})\hat{w}_a)|\\ &\leq \sum_{a\in\mathcal{S}}|\alpha-\hat{\alpha}|f_a+\sum_{a\in\mathcal{S}}|(1-\alpha)w_a-(1-\hat{\alpha})\hat{w}_a|\\ &\leq |\alpha-\hat{\alpha}|+\sum_{a\in\mathcal{S}}(|\alpha-\hat{\alpha}|\hat{w}_a+|w_a-\hat{w}_a|(1-\hat{\alpha})\\ &\qquad -|\alpha-\hat{\alpha}|\cdot|w_a-\hat{w}_a|)\\ &\leq 2|\alpha-\hat{\alpha}|+(1-\hat{\alpha})\sum_{a\in\mathcal{S}}|w_a-\hat{w}_a|\\ &\qquad -|\alpha-\hat{\alpha}|\sum_{a\in\mathcal{S}}|w_a-\hat{w}_a|\\ &\leq \frac{2\sqrt{\ln((4+2K)/\delta)}}{Z_{a^*}\sqrt{M}}\\ &\qquad +\min\left\{\frac{K(Z_{a^*}/\sqrt{2}+2)\sqrt{\ln((4+2K)/\delta)}}{Z_{a^*}(1-\hat{\alpha})\sqrt{M}-\sqrt{\ln((4+2K)/\delta)}},\right.\\ &\qquad\qquad \left.2(1-\hat{\alpha})\right\}\ .\end{aligned}$$

Notice that this holds uniformly for all $\vec{f}$, so the same bound holds when we take an expectation over $\vec{f}$.

A.4 Handling the case where Z_{a^*} is small

Suppose that for an action a, $Z_a < \epsilon$. Let η_a and μ_a be the true median and mean respectively of the distribution from which the random variables $f_{m,a}$ are drawn. Let f_a^{high} be the mean value of the distribution over $f_{m,a}$ conditioned on $f_{m,a} > \eta_a$, and similarly let f_a^{low} be the mean value conditioned on $f_{m,a} < \eta_a$, so $\mu_a = \left(f_a^{low}+f_a^{high}\right)/2$.[1] Let $\bar{f}_a^{high}$ be the empirical average of $f_{m,a}$ conditioned on $f_{m,a} > \eta_a$, and $\bar{f}_a^{low}$ be the empirical average of $f_{m,a}$ conditioned on $f_{m,a} < \eta_a$. (Note that we cannot actually compute $\bar{f}_a^{high}$ and $\bar{f}_a^{low}$ since η_a is unknown.) Fi-

[1] Assume for now that it is never the case that $f_{m,a} = \eta_a$. This simplifies the explanation, although everything still holds if $f_{m,a}$ can be η_a.

nally, let $\hat{f}_a^{high} = (2/M)\sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{high}} f_{m,a}$ and $\hat{f}_a^{low} = (2/M)\sum_{m:\mathcal{I}_m \in \mathcal{M}_a^{low}} f_{m,a}$. Notice that $Z_a = \hat{f}_a^{high} - \hat{f}_a^{low}$.

We show first that f_a^{high} and f_a^{low} are close to $\bar{f}_a^{high}$ and $\bar{f}_a^{low}$ respectively. Next we show that $\bar{f}_a^{high}$ and $\bar{f}_a^{low}$ are close to $\hat{f}_a^{high}$ and $\hat{f}_a^{low}$ respectively. Finally, we show that this implies that if Z_a is small, then the probability that a random value of $f_{m,a}$ is far from η_a is small. This in turn implies a small $\mathcal{L}_1$ distance between our estimated model and the real model for each agent.

To bound the difference between f_a^{high} and $\bar{f}_a^{high}$, it is first necessary to lower bound the number of points in the empirical samples with $f_{m,a} > \eta_a$. Let z_m be a random variable that is 1 if $f_{m,a} > \eta_a$ and 0 otherwise. Clearly $\Pr(z_m = 1) = \Pr(z_m = 0) = 1/2$. By a straightforward application of Hoeffding's inequality, for any δ_3, with probability $1 - \delta_3$,

$$\left| \sum_{m:\mathcal{I}_m \in \mathcal{M}} z_m - \frac{M}{2} \right| \le \sqrt{\frac{\ln(2/\delta_3)M}{2}} ,$$

and so the number of samples averaged to get $\bar{f}_a^{high}$ is at least $(M/2) - \sqrt{\ln(2/\delta_3)M/2}$. Applying Hoeffding's inequality again, with probability $1 - \delta_4$,

$$\left| f_a^{high} - \bar{f}_a^{high} \right| \le \sqrt{\frac{\ln(2/\delta_4)}{M - \sqrt{2\ln(2/\delta_3)M}}} .$$

Now, $\hat{f}_a^{high}$ is an empirical average of $M/2$ values in $[0,1]$, while $\bar{f}_a^{high}$ is an empirical average of the same points, plus or minus up to $\sqrt{\ln(2/\delta_3)M/2}$ points. In the worst case, $\bar{f}_a^{high}$ either includes an additional $\sqrt{\ln(2/\delta_3)M/2}$ points with values lower than any point in $\mathcal{M}_a^{high}$, or excludes the $\sqrt{\ln(2/\delta_3)M/2}$ lowest values points in $\mathcal{M}_a^{high}$. This implies that in the worst case,

$$\left| \bar{f}_a^{high} - \hat{f}_a^{high} \right| \le \frac{\sqrt{\ln(2/\delta_3)}}{\sqrt{M/2} - \sqrt{\ln(2/\delta_3)}} .$$

By the triangle inequality,

$$\begin{aligned} \left| f_a^{high} - \hat{f}_a^{high} \right| &\le \sqrt{\frac{\ln(2/\delta_4)}{M - \sqrt{2\ln(2/\delta_3)M}}} \\ &\quad + \frac{\sqrt{\ln(2/\delta_3)}}{\sqrt{M/2} - \sqrt{\ln(2/\delta_3)}} . \end{aligned}$$

The same can be show for f_a^{low} and $\hat{f}_a^{low}$. Hence if $Z_a \le \epsilon$, then

$$\begin{aligned} f_a^{high} - f_a^{low} &\le \epsilon + 2\sqrt{\frac{\ln(2/\delta_4)}{M - \sqrt{2\ln(2/\delta_3)M}}} \\ &\quad + \frac{2\sqrt{\ln(2/\delta_3)}}{\sqrt{M/2} - \sqrt{\ln(2/\delta_3)}} . \end{aligned}$$

Call this quantity ϵ'. Clearly we have

$$\mu_a - \epsilon' \le f_a^{low} \le \mu_a \le f_a^{high} \le \mu_a + \epsilon'.$$

Since f_a^{high} is an average of points which are all higher than f_a^{low}, and similarly f_a^{low} is an average of points all lower than f_a^{high}, this implies that for any $\tau \ge 0$,

$$\begin{aligned} &\Pr(|f_{m,a} - \mu_a| \ge \epsilon' + \tau) \\ &\quad \le \Pr(f_{m,a} \ge f_a^{high} + \tau) + \Pr(f_{m,a} \le f_a^{low} - \tau) \\ &\quad \le \epsilon'/(\epsilon' + \tau) . \end{aligned}$$

Recall that when Z_a is small for all a, we set $\hat{\alpha} = 0$ and $\hat{w}_a = \sum_{m:\mathcal{I}_m \in \mathcal{M}} \mathrm{I}(a_m = a)$. Let $\bar{w}_a = \alpha\mu_a + (1-\alpha)w_a$. Notice that $\mathrm{E}[\hat{w}_a] = \bar{w}_a$. Applying Hoeffding's inequality yet again, with probability $1 - \delta_5$, $|\hat{w}_a - \bar{w}_a| \le \sqrt{\ln(2/\delta_5)/2M}$. For any $\tau > 0$,

$$\begin{aligned} &\mathrm{E}_{\vec{f} \sim D^f}\left[\sum_{a \in \mathcal{S}} |(\alpha f_a + (1-\alpha)w_a) - (\hat{\alpha} f_a + (1-\hat{\alpha})\hat{w}_a)| \right] \\ &= \sum_{a \in \mathcal{S}} \mathrm{E}_{\vec{f} \sim D^f}\left[|\alpha f_a + (1-\alpha)w_a - \hat{w}_a| \right] \\ &\le \sum_{a \in \mathcal{S}} \mathrm{E}_{\vec{f} \sim D^f}\left[|\alpha f_a + (1-\alpha)w_a - \bar{w}_a| + |\bar{w}_a - \hat{w}_a| \right] \\ &\le \alpha \sum_{a \in \mathcal{S}} \mathrm{E}_{\vec{f} \sim D^f}\left[|f_a - \mu_a| \right] + K\sqrt{\ln(2/\delta_5)/2M} \\ &\le K\left(\left(1 - \frac{\epsilon'}{\epsilon' + \tau}\right)(\epsilon' + \tau) + \frac{\epsilon'}{\epsilon' + \tau} \right) \\ &\qquad + K\sqrt{\ln(2/\delta_5)/2M} \\ &= K\left(\tau + \frac{\epsilon'}{\epsilon' + \tau} + \sqrt{\ln(2/\delta_5)/2M} \right) . \end{aligned}$$

B Proofs from Section 5.2

B.1 Proof of Lemma 12

By the union bound,

$$\begin{aligned} &\Pr_{\vec{f} \sim D^f}\left[\exists a, b \in \mathcal{S} : f_a > 0, f_b > 0, |\mathcal{M}_{a,b}| < M \right] \\ &\quad \le \sum_{a,b \in \mathcal{S}} \Pr_{\vec{f} \sim D^f}\left[f_a > 0, f_b > 0, |\mathcal{M}_{a,b}| < M \right] \\ &\quad \le \sum_{a,b \in \mathcal{S} : |\mathcal{M}_{a,b}| < M} \Pr_{\vec{f} \sim D^f}\left[f_a > 0, f_b > 0 \right] . \end{aligned}$$

For any fixed pair of actions a, b such that $|\mathcal{M}_{a,b}| < M$, it follows from Hoeffding's inequality that for any $\delta \in (0,1)$, with probability $1 - \delta$,

$$\Pr_{\vec{f} \sim D^f}\left[f_a > 0, f_b > 0 \right] \le \frac{M}{|\mathcal{M}|} + \sqrt{\frac{\ln(1/\delta)}{2|\mathcal{M}|}} .$$

Noting that the number of pairs is less than $K^2/2$ and setting $\delta = 2\delta/K^2$ yields the lemma.

B.2 Bounding the $\mathcal{L}_1$

Here we show that with high probability (over the choice of $\mathcal{M}$ and the draw of $\vec{f}$), if $\mathcal{M}$ is sufficiently large,

$$\begin{aligned} &\sum_{a \in \mathcal{S}} \left| \frac{w_a f_a}{\sum_{s \in \mathcal{S}} w_s f_s} - \frac{\hat{w}_a f_a}{\sum_{s \in \mathcal{S}} \hat{w}_s f_s} \right| \\ &\quad \le \frac{2(1+\beta)KN\sqrt{N\ln(2K/\delta)}}{\sqrt{2M} - (1+\beta)K(N+1)\sqrt{N\ln(2K/\delta)}} . \end{aligned}$$

First note that we can rewrite the expression on the left-hand side of the inequality above as

$$\sum_{a \in \mathcal{S}'} \left| \frac{w'_a f_a}{\sum_{s \in \mathcal{S}'} w'_s f_s} - \frac{\hat{w}_a f_a}{\sum_{s \in \mathcal{S}'} \hat{w}_s f_s} \right| ,$$

where for all a, $w'_a = w_a / (\sum_{s \in \mathcal{S}'} w_s)$. We know from Equation 9 that with high probability (over the data set and the choice of $\vec{f}$), for all $a \in \mathcal{S}'$,

$$|w'_a f_a - \hat{w}_a f_a| \leq \frac{f_a (1+\beta) |\mathcal{S}'| \sqrt{N \ln(2K^2/\delta)}}{\sqrt{2M} - (1+\beta)|\mathcal{S}'| \sqrt{N \ln(2K^2/\delta)}} ,$$

and

$$\begin{aligned} & \left| \sum_{s \in \mathcal{S}'} w'_s f_s - \sum_{s \in \mathcal{S}'} w_s f_s \right| \\ \leq & \sum_{s \in \mathcal{S}'} f_s |w'_s - \hat{w}_s| \\ \leq & \frac{(1+\beta)|\mathcal{S}'| \sqrt{N \ln(2K^2/\delta)}}{\sqrt{2M} - (1+\beta)|\mathcal{S}'| \sqrt{N \ln(2K^2/\delta)}} . \end{aligned}$$

Thus, using the fact that $\sum_{s \in \mathcal{S}'} \hat{w}_s f_s \geq 1/N$, we can apply Lemma 8 once again to get that

$$\begin{aligned} & \left| \frac{w'_a f_a}{\sum_{s \in \mathcal{S}'} w'_s f_s} - \frac{\hat{w}_a f_a}{\sum_{s \in \mathcal{S}'} \hat{w}_s f_s} \right| \\ \leq & \frac{(1+\beta) K N \sqrt{N \ln(2K/\delta)} \left(f_a + \frac{\hat{w}_a f_a}{\sum_{s \in \mathcal{S}'} \hat{w}_s f_s} \right)}{\sqrt{2M} - (1+\beta) K (N+1) \sqrt{N \ln(2K/\delta)}} , \end{aligned}$$

and

$$\begin{aligned} & \sum_{a \in \mathcal{S}'} \left| \frac{w'_a f_a}{\sum_{s \in \mathcal{S}'} w'_s f_s} - \frac{\hat{w}_a f_a}{\sum_{s \in \mathcal{S}'} \hat{w}_s f_s} \right| \\ \leq & \frac{2(1+\beta) K N \sqrt{N \ln(2K/\delta)}}{\sqrt{2M} - (1+\beta) K (N+1) \sqrt{N \ln(2K/\delta)}} . \end{aligned}$$