Distributed Prediction Markets modeled by Weighted Bayesian Graphical Games

Janyl Jumadinova and Prithviraj Dasgupta

Computer Science Department
University of Nebraska at Omaha, Omaha, NE, USA,
{jjumadinova, pdasgupta}@unomaha.edu

Abstract. We consider a novel, yet practical setting of prediction markets called distributed prediction markets, where the aggregated price of a security of an event in one prediction market is affected dynamically by the prices of securities of similar events in other, simultaneously running prediction markets. We focus on the problem of decision making facing a market maker to determine the price of a security within such a setting. We propose a formal framework based on graphical games called a weighted Bayesian graphical game (WBGG) to model the distributed prediction market setting and to capture the local interactions between multiple market makers. We then describe a distributed message passing algorithm based on NashProp algorithm to calculate the Bayes-Nash equilibrium in a WBGG. We provide analytical results including convergence and incentivizing truthful revelation among market makers. Our experimental results show that market makers that consider the influence of other market makers in a distributed prediction market setting while using our proposed WBGG-based algorithm obtain higher utilities and set prices more accurately in comparison to market makers using a greedy strategy to set prices or those that do not consider the influence of other market makers. We also observe that extreme sizes of the neighborhood of a market maker have an adverse impact on its utilities.

Keywords: Prediction market, distributed, bayes-nash, graphical games

1 Introduction

Over the past decade prediction markets have shown ample success as a tool for aggregating information and predicting the outcome of events that are going to happen in the future. A prediction market operates by aggregating the opinions or beliefs on the outcome of a future, real-world event from the market's participants, called traders, and forecasts the event's possible outcome based on their aggregated opinion. A trader's belief about an event's possible outcome is usually represented as a price for a security related to the event and a trader can buy or sell securities according to its belief. Traders' beliefs or prices are aggregated by an entity called the market maker that implements rules to deter misreporting of prices by traders, limits trading quantity, etc., using an aggregation mechanism. In general, existing research on prediction markets mainly focusses on the operation of a single prediction market, and considers that the aggregated price of a security in a prediction market is determined by prices of traders only within that prediction market.

Contrary to single prediction markets with confined traders, there are several real-life instances where multiple prediction markets running simultaneously have similar events. For example, both Intrade and Iowa Electronic Market ran prediction markets on several events related to the 2012 U.S. Presidential elections. With such similar events across markets, it is very likely that the expected outcomes (prices) of an event in one market will influence the price of the same or similar event in a different market. Based on this insight, our main hypothesis in this paper is that the predicted outcome of events in one prediction market is dynamically affected not only by the market's traders but also by the expected outcome of similar events in other prediction markets. Such inter-market influence is frequently observed in financial markets, which operate very similarly to prediction markets. For example, after analyzing the data from the retail online brokerage, called eToro, Pan *et al.* reported that the prices of securities in one market affect the prices of similar securities in other markets and that social trading (when traders can see each others' trades) results in higher profits to the traders [11]. Therefore, it makes sense to investigate the effects of events across multiple prediction markets and analyze how prices evolve due to such inter-market effects. The main contributions made by our paper towards studying this problem are the following: we describe a model of a distributed prediction market that comprises multiple, parallel running prediction markets and uses a graphical structure between the market makers of the different markets to represent inter-market influence. We then propose a formal framework based on graphical games [7] called a Weighted Bayesian Graphical Game (WBGG) to capture the interaction between multiple market makers and describe an algorithm based on NashProp [9] to calculate an approximate Bayes-Nash equilibrium efficiently for an n-player WBGG. Finally, we conduct a number of experiments with two types of market makers and with two possible actions for each market makers to analyze the effect of different parameters in distributed prediction markets. We find that when the size of the neighborhood is too small or too large, market makers' utilities decrease, and, that market makers using our algorithm in a distributed prediction market setting can outperform the market makers using a greedy strategy or the market makers in the setting where prediction markets are disjoint. To the best of our knowledge, this work represents the first attempt at studying inter-market influences between similar events across multiple prediction markets through strategic decision making by market makers.

2 Related Work

Prediction Markets. A prediction market is a market-based aggregation mechanism that is predominantly used to combine the opinions on the outcome of a future, real-world event from different people, called the market's *traders* and forecast the event's possible outcome based on their aggregated opinion. The basic operation rules of a prediction market are similar to those of a continuous double auction, with the role of the auctioneer being taken up by an entity called the *market maker* that runs the prediction market. The seminal work on prediction market analysis [14] has shown that the mean belief values of individual traders about the outcome of a future event corresponds to the event's market price. Since then researchers have studied the traders' behavior in prediction markets [2, 3], rules that a market maker can use to combine the opinions (beliefs) from different traders [5], and the properties of prediction markets used for decision making [1, 10]. In contrast to these previous works, we consider a setting with multiple prediction markets where the inter-market influence on market prices is incorporated through decision-making (price selection) by the market-makers.

Graphical Games. The original work by Kearns et al. [7] considered acyclic graphical games of complete information in which the underlying graph is a tree and presented a message-passing algorithm, known as TreeProp or KLS algorithm, for computing approximate Nash equilibria (NE) efficiently. Consequently, in [9] the authors generalized the TreeProp algorithm to an arbitrary graph structure by proposing a message-passing NashProp algorithm for complete information games which involves an approach analogous to loopy belief propagation in graphical games. In other work, Vickrey and Koller [13] presented multi-agent algorithms for solving graphical games including hill-climbing, constraint satisfaction, and hybrid approaches. In [12] the authors studied the graphical games with incomplete information with discrete and continuous types and propose an extension to KLS algorithm to find Bayes-Nash equilibrium efficiently in a tree structured graphical game. In our work we build upon [9, 12] and propose a form of *weighted* graphical games *with incomplete information* and *an arbitrary graph structure*. We extend NashProp algorithm to find BNE in such games with an arbitrary graph structure and apply this algorithm to the distributed prediction market setting.

3 Distributed Prediction Markets

In this section we define and characterize distributed prediction markets. We consider n prediction markets with each prediction market having one market maker that is responsible for aggregating traders' beliefs and setting the market price for its market. Let $N = \{1, ..., n\}$ denote the set of market makers with i being the market maker for the i-th prediction market. Let $\Gamma = \{\Gamma_1, ..., \Gamma_n\}$ denote a set of trading agents representing human traders, with Γ_i being the set of trading agents in the i-th prediction market. Note that trading agents can participate in multiple prediction markets simultaneously.Also let $E = \{E_1, ..., E_n\}$ denote the set of events across all the prediction markets with E_i representing the set of events in the i-th prediction market[1]. Finally, let P denote the trader population.

Next, we present two axioms that outline the behavior of market makers in the distributed prediction market setting. Consider two market makers i and j running events e_{f_i} and e_{g_j} in their respective prediction markets. Let d_{f_i,g_j} be

[1] For legibility, we refer to the security corresponding to an event as the event itself.

a metric that measures the similarity between the definitions of the two events e_{f_i} and e_{g_j}[2].

Definition 1. Related Event. *An event e_{f_i} is related to event e_{g_j} if $d_{f_i,g_j} > \epsilon_{sim}$, where ϵ_{sim} is a constant.*

We denote the number of market makers that market maker i interacts with by η_i and the influence of the market maker i on market maker j by ϖ_{ij}.

Axiom 1 ***Local interaction, Influence, Competition:*** *If events e_{f_i} and e_{g_j} run by market makers i and j correspondingly are related, then (1) market maker i* interacts *with market maker j for determining the price of event e_{f_i}, (2) $0 < \varpi_{i,j} < 1$, and (3) market makers i and j are competitive.*

The first part of Axiom 1 determines the criterion for interaction between two market makers. Market maker i interacts with another market maker j for updating event e_{f_i}'s price, only if j's market has an event that is related to event e_{f_i}. The second part of Axiom 1 states that two market makers that have a pair of related events in their market have a non-zero, positive influence on each other. Influence values are normalized to a range of 0 and 1. In a prediction market a market maker needs to be able to calculate the market price (aggregate) and to stimulate the trading by always allowing traders to buy or sell securities. If there are prediction markets with similar events, the traders may choose one prediction market over the other, and market makers may end up competing over the traders just like in financial markets [6]. The third part of Axiom 1 summarizes this competitive behavior between market makers.

Axiom 2 ***Incentives:*** *Let θ_i be the private information of market maker i denoted as i's type. Let $u_i(a_i, \theta_i, t)$ be the utility that the market maker i gets for interacting with other market makers when its type is θ_i. If $d_{f_i,g_j} > \epsilon_{sim}$, for any i, j, e_{f_i}, e_{g_j}, then $\exists\ \theta_i'$ such that $u_i(a_i, \theta_i', t) > u_i(a_i, \theta_i, t)$.*

Finally, because market makers are competing with each other to attract traders on related events, a market maker may have incentives to misreport its inside information about aggregated prices when interacting with other market makers. In other words, market-makers have preferences over their types to improve their utility as mentioned in Axiom 2.

Definition 2. Distributed prediction market. *A distributed prediction market is specified by the tuple $\mathcal{M} = \langle N, P, E, \mathcal{W}\rangle$ where N, P, E are as defined before and $\mathcal{W} = \{\varpi_{ij} : i, j \in N\}$.*

For simplicity of analysis in the rest of the paper we assume the setting where multiple market makers interact over one related event; but the results are valid

[2] We assume that d_{f_i,g_j} is based on the similarity between the written description of the events e_{f_i} and e_{g_i} and is provided externally to the market makers either by a human expert or by an automated program.

for multiple events. Since the decision making for inter-market influence is done mainly by market makers, we abstract the operation of the trading agents and assume that the intra-market price of each prediction market is updated by the market maker's actions in the locality of the current market price. Nevertheless, our proposed technique can be combined easily with any other intra-market price update method such market scoring rules. Since we are not focussing on the interactions of trading agents, we refer to market maker agents as agents in the rest of the paper.

3.1 Weighted Bayesian Graphical Games for Distributed Prediction Markets

We propose a form of graphical games [7, 9, 12] as a formal model for the interaction between market makers in a distributed prediction market. *Graphical games* are a compact representation of complete information, one-shot, normal-form games that use graphical models to represent the set of agents whose actions influence each others' payoffs. A graphical game representation is appropriate for modeling distributed prediction markets because they can capture the interactions and influences between agents that are within a certain local neighborhood of each other, unlike for example conventional Bayesian games. A graphical game is described by an undirected graph G in which agents are identified with nodes, and the edge between two nodes implies that the payoff of each of the two agents is dependent on the other agent's actions.

We present an augmented form of the conventional graphical game to represent distributed prediction markets, called a Weighted Bayesian Graphical Game (WBGG). Unlike previous works on graphical games, WBGG incorporates incomplete information, the influence of agents on each other as a pair of *directed* edges, and an arbitrary graphical structure in one representation. We assume that a set of similar events in a distributed prediction market have a duration of T periods with t denoting the current time period. The agents that are able to interact and therefore influence each other's utilities define a neighborhood $\mathcal{N}$, where $\mathcal{N}_{-i} = \{j | j \in N_{-i}, \xi_{ij} \in \Xi_i\}$ and $\mathcal{N}_i = \mathcal{N}_{-i} \cup \{i\}$. We also assume that each agent can be one of two possible types, i.e. if Θ_i is the type space of market maker agent i then $\Theta_i = \{\theta_{iopt}, \theta_{ipes}\}$ with θ_{iopt} implying that agent i is an *optimistic* market maker agent and θ_{ipes} implying that it is a *pessimistic* market maker agent. Specifically, we define a WBGG as follows:

Definition 3. *A WBGG is a tuple* $(N, \Theta, p, \Xi, \mathcal{W}, A, u)$*, where*
- $N = \{1, ..., n\}$ *- set of market maker agents.*
- $\Theta = \Theta_1 \times ... \times \Theta_n$*, where* Θ_i *is the type space of agent* i*.*
- $p : \Theta \to [0, 1]$ *is the common prior over types.*
- $\Xi = \{\Xi_1, ..., \Xi_n\}$ *- set of directed edges, where* $\Xi_i = \{\xi_{ij} | \xi_{ij} = (i, j), i, j \in N\}$ *with* ξ_{ij} *being an edge between agents i and j that are able to interact.*
- $\mathcal{W}$ *- set of edge weights, where* $\varpi_{ij} \in \mathcal{W}$ *is the weight of the edge* ξ_{ij} *between agents i and j.* ϖ_{ij} *is determined by agent i and indicates the influence of agent i on agent j. Given agent i's type is* θ_{iopt}*,* ϖ_{ij} *is calculated as:*

$$\varpi_{ij} = \alpha p(\theta_{jopt} | \theta_{iopt}) + (1 - \alpha) p(\theta_{jpes} | \theta_{iopt}), \tag{1}$$

where $\theta_{iopt} \in \Theta_i$, $\theta_{jopt}, \theta_{jpes} \in \Theta_j$ *and* α *is a confidence parameter representing* i*'s belief that* j *is of the same type as itself.*
- $A = A_1 \times ... \times A_n$*, where* A_i *is a finite set of actions available to agent* i.
- $u_i : \Theta_{i \in N} \times A_{i \in N} \times T \rightarrow R$ *is the utility of agent* i.

Following Bayesian games [8] we use $s_i(\theta_{iopt})$ to denote agent i's mixed strategy over A_i given its type is θ_{iopt}. S_i is the set of all i's mixed strategies. We use notation s_i for unconditional mixed strategy of agent i. We use $A_{\mathcal{N}_i}, \overline{s}_{\mathcal{N}_i}, \overline{\theta}_{\mathcal{N}_i}$ to denote the vector of actions, strategies and types of all agents in the neighborhood of agent i, $\overline{\varpi}_{\mathcal{N}_{-i}}$ denote the vector of weights between agent i and all agents in the neighborhood of agent i, a_i, s_i, θ_i to denote the action, strategy and type of agent i, and $A_{\mathcal{N}_{-i}}, \overline{s}_{\mathcal{N}_{-i}}, \overline{\theta}_{\mathcal{N}_{-i}}$ to denote actions, strategies and types of all agents in the neighborhood of agent i except agent i itself.

Agent action set and utility function. For specifying the actions in agent i's action set A_i, we assume that agent i can have two possible actions in A_i - to raise the current market price or to lower it by a certain amount that is specified by a *jump* parameter λ_i^t, i.e., $A_i = \{\lambda_i^t, -\lambda_i^t\}$. To prevent arbitrary values of λ_i^t, we make λ_i^t inversely proportional to the market price π_i^t, i.e. $\lambda_i^t = \frac{\delta_i^t}{\pi_i^t}$, where $\pi_i^t \in (\zeta, 1]$ is the market price at time step t, with ζ being a small positive constant corresponding to a very nominal price change, and δ_i^t is a constant that determines the direction (up or down) of the price change [3]. The value of δ_i^t is determined by agent i depending on its type and by observing the direction of the market prices changes among its neighbors, as given by the following equations:

$$\delta^t_{i|\theta_i=opt} = \begin{cases} +\zeta & \text{if } \frac{\sum_{j \in \mathcal{N}_{-i}} \pi_j^t - \pi_j^{t-1}}{|\mathcal{N}_{-i}|} \geq 0, \\ -\zeta & \text{otherwise} \end{cases}$$

$$\delta^t_{i|\theta_i=pes} = \begin{cases} +\zeta & \text{if } \forall j \in \mathcal{N}_{-i},\, \pi_j^t - \pi_j^{t-1} \geq 0, \\ -\zeta & \text{if } \exists j \text{ s.t. } \pi_j^t - \pi_j^{t-1} < 0 \end{cases}$$

If agent i is optimistic, it sets $\delta_i^t = +\zeta$ if the average change in the market price of its neighbors in the last time step has been non-negative, otherwise it sets $\delta_i^t = -\zeta_i$. Similarly, if agent i is pessimistic, it sets $\delta_i^t = +\zeta_i$ only if all of its neighbors increased their prices in the last time step but sets $\delta_i^t = -\zeta_i$ if at least one of its neighbors decreased its prices in the last time step. [4]

The utility of agent i is calculated as:

$$u_i(a_i, \theta_i, t) = (T - t)e^{-a_i(\theta_i)(T-t)}, \tag{2}$$

where t is the current time period ($t = 0$ at the start of the market). The above utility equation guarantees that the utility of changing the market price (by

[3] If π_i crosses either its lower or upper bound due to action A_i, we set it back to its lower or upper bound correspondingly.

[4] We drop superscript t from λ_i^t, δ_i^t, and π_i^t henceforth, assuming it to be understood from the context.

taking an action in the WBGG) is proportional to the remaining duration of the event in the market, and, more exploration (large price changes) gives higher utility towards the beginning of the event, but as the event nears its end and its price converges, large explorations are punished with lower utility.

Next, we define the agent's expected utility in a weighted Bayesian graphical game as

$$EU_i(s_i, \overline{s}_{\mathcal{N}_{-i}}, \theta_i) = \sum_{a_{\mathcal{N}_i} \in A_{\mathcal{N}_i}} \overline{\varpi}_{\mathcal{N}_{-i}} \times \left(s_i \overline{s}_{\mathcal{N}_{-i}} u_i(A_{\mathcal{N}_i}, \overline{\theta}_{\mathcal{N}_{-i}}, \theta_i) \right),$$

where $\overline{\varpi}_{\mathcal{N}_{-i}} = \prod_{j \in \mathcal{N}_{-i}} \varpi_j$. Note that agent i has to consider every assignment of types to the other agents in its neighborhood $\theta_{\mathcal{N}_{-i}}$ and every action profile $\overline{a}_{\mathcal{N}_i}$ in order to calculate the utility $u_i(\overline{a}_{\mathcal{N}_i}, \overline{\theta}_{\mathcal{N}_{-i}}, \theta_i)$.

Definition 4. *For agent i a strategy s_i is said to be* ***best response (BR)*** *in a WBGG for type θ_i to $\overline{\theta}_{\mathcal{N}_i}$, if*

$$\forall s', EU(s_i, \overline{s}_{\mathcal{N}_{-i}}, \theta_i) \geq EU(s', \overline{s}_{\mathcal{N}_{-i}}, \theta_i) \tag{3}$$

Definition 5. *A strategy vector $\overline{s}$ is a* ***Bayes-Nash Equilibrium (BNE)*** *in a WBGG if and only if every agent i is playing a best response to the others.*

3.2 Computing Bayes-Nash Equilibrium

In this section we first present an abstract algorithm for computing BNE in a weighted Bayesian graphical game of an arbitrary graphical structure. This algorithm is similar to NashProp algorithm [9] that has been extended to incomplete games with an arbitrary graphical structure, thus the definition of the expected utility and the best response have been modified.

For now, we will not purposefully specify a certain representation and a certain implementation. After proving the correctness of the abstract algorithm, we will fill in the unspecified gaps. The abstract algorithm is basically a two-stage message passing algorithm. In the first step, local optimal response is found for each agent, where each agent calculates the optimal strategy given its neighbor's strategies and sends it to its neighbors. In the second step, global solution is constructed by eliminating inconsistent local optimal response.

Let $D_{i,j}$ be the binary table indexed by all possible strategies of agent i and agent j that is sent from agent j to agent i. $D_{i,j}$ indicates player j's belief that there is a global Nash equilibrium when players i and j choose s_i, s_j respectively. Let $\mathcal{N}^j_{-i} = (1, ..., m-1)$ denote the neighbors of agent j besides agent i. And let $\overline{s}_{\mathcal{N}^j_{-i}} = (s_1, ..., s_{m-1})$ be the vector of mixed strategies of agents in $\mathcal{N}^j_{-i}$, called the *witness* to $D_{i,j}$. Also let P_j be the projection set of agent j that is used just to consolidate the information sent to agent j by all of its neighbors.

Theorem 1. *Algorithm 1 computes BNE for an arbitrary graphical game and the tables and witnesses calculated by it contain all possible BNE of the game.*

findBNE( )
Input: Game specification, duration T
Output: BNE of the game
Set $t = 0$; // initialize the prediction market's time period
Set $r = 0$; // initialize the round for the first stage
Set $D^0_{j,i}(s_i, s_j) = 1\ \forall s_i, s_j,\ \forall i, j$; // initialize the table, where agent i and agent j are the neighbors
foreach *time period* $t \leq T$ **do**
  Stage 1 : Local Optima
  while *not converged* **do**
    foreach *agent pair* i, j **do**
      foreach s_i, s_j **do**
        if $D^r(s_j, s_h) = 1\ \forall h$ *AND* s_j *is Best Response to* $\overline{s}_{\mathcal{N}^j_{-i}}$ *and* s_i
        then
          $D^{r+1}(s_i, s_j) = 1$;
          save $\overline{s}_{\mathcal{N}^j_{-i}}$ as a witness to $D^{r+1}(s_i, s_j) = 1$;
        end
        else
          $D^r(s_j, s_h) = 0\ \forall h$;
        end
      end
      send D^{r+1} to all of its neighbors;
    end
    $r = r + 1$;
  end
  Stage 2 : Global Optima
  foreach *agent* j **do**
    if $\exists\ \overline{s}_{\mathcal{N}^j_{-i}}$ *s.t.* $D(s_j, s_h) = 1\ \forall h \in \mathcal{N}^j_{-i}$ **then**
      $P_j(s_j) = 1$;
    end
    else
      $P_j(s_j) = 0$;
    end
    // construct BNE - local search
    1. Pick any agent j and any s_j s.t. $P(s_j) = 1$ with witness $\overline{s}_{\mathcal{N}^j_{-i}}$;
    2. Agent j assigns itself s_j and each of its neighbors h it assigns s_h;
    3. Pick next node and assign of all its unassigned neighbors with its witness;
    4. Backtrack if s_j is not the Best Response when all of the agent j's neighbors are assigned;
  end
  t++;
end

Algorithm 1: Algorithm to find BNE in a WBGG.

Proof. The proof is a constructive argument of the workings of the algorithm. The stage 1 starts with an arbitrary node. Each node(agent) i sends each of its neighbors j a binary-valued table $D_{j,i}$ indexed by all possible strategies of agents j and i. For any pair of strategies (s_j, s_i) a table $D_{j,i}$ is 1 if and only if there exists a BNE in which agent i plays s_i when its neighboring agent j plays s_j.

Consider a node i with neighbors j and $\mathcal{N}^j_{-i} = \{1, ..., m-1\}$. For induction, assume that each h sends node i table $D_{i,h}$. For any pair of strategies (s_j, s_i) a table $D_{j,i}$ is 1 if and only if there exists a vector of strategies $\overline{s}_{\mathcal{N}^j_{-i}} = \{s_1, ..., s_{m-1}\}$ (witness) for $\mathcal{N}^j_{-i}$ such that:
1. $D_{i,h}(s_i, s_h) = 1 \ \forall 1 \leq h \leq m$, and
2. s_i is the best-response to $\overline{s}_h$ and s_j.

There maybe more than one witness for $D_{j,i}(s_j, s_i) = 1$. In addition to computing the binary-valued tables (i.e. $D_{j,i}$), stage 1 of the algorithm also saves a list of witnesses for each pair of strategies (s_j, s_i) for which the table $(D_{j,i})$ is 1.

Now assume that $D_{j,i} = 1$ for some node i with neighbors j and $\mathcal{N}^j_{-i}$ for some witness $\overline{s}_{\mathcal{N}^j_{-i}}$. By construction, $D_{i,h}(s_i, s_h) = 1 \ \forall h$, and therefore by induction it must be that there exists BNE in which h plays s_h given that node i plays s_i and by construction of $D_{j,i}$ s_i is a best response of agent i and must be a part of BNE given that agent j plays s_j.

Stage 1 converges because all tables begin filled with 1 entries and entries can only change from 0 to 1, [9].

Stage 2 is a backtracking local assignment passing stage. It starts at an arbitrary node j which can chose any s_j for which $P(s_j) = 1$ and any witness $\overline{s}_{\mathcal{N}^j_{-i}}$ from the associated witness list. The node j then passes (s_j, s_h) to each its neighbors h telling h to play s_h. From the semantics of this message passing and backtracking step if s_j turns out to not be the best response when all of j's neighbors are assigned strategies, it must be true that s_j is the best response to its neighbors for any node j.

3.3 Computing approximate Bayes-Nash Equilibrium

Algorithm 1 is incompletely specified because the representation and computation of the step of passing tables in stage 1 is not completely specified. Since the strategy for an agent is a mapping from types to the simplex of probability distributions over actions, it may not be possible to represent tables D compactly or finitely for an arbitrary graphical game. We now present an algorithm for computing approximate BNE in incomplete information general structured graphical games with discrete types.

We adopt our abstract algorithm to compute ϵ-BNE in graphical games of an arbitrary structure with discrete types. Our updated algorithm takes parameter ϵ as input, that specifies how close of an approximation to BNE we want to get. The strategy space is discretized analogous to [7], such that any agent can only choose actions with probabilities that are multiples of τ, for some τ, instead of playing an arbitrary mixed strategy in $[0, 1]$. For a graphical games that contains l actions the probability that each action will be selected is a multiple of τ with the sum of all probabilities being 1. Then any agent i will have $O(\frac{1}{\tau^{2(l-1)}})$ different strategies.

Approximate-findBNE( )

Input: Game specification and ϵ approximation parameter Output: ϵ-BNE of the game

Run Algorithm 1 with two changes:

1. Only consider type-conditional discretized strategies
2. Change the requirement of best response to ϵ-best response.

Algorithm 2: Approximation algorithm to find BNE in a weighted graphical game.

Theorem 2. *For any $\epsilon > 0$, $k = max_i \eta_i \ll n$, and the discretization parameter $\tau \leq \frac{\epsilon}{l^k(4klog(k))}$ Algorithm 2 computes ϵ-BNE for an arbitrarily structured graphical game with incomplete information.*

Proof. In [12] it was shown that if the mixed strategy space for every type is restricted to multiples of τ, then for any ϵ and $\tau \leq \frac{\epsilon}{l^k(4klog(k))}$ there exists ϵ-BNE in tree structured graphical games. Their result however does not depend on the underlying graph being a tree, and therefore holds for arbitrary graphs also. The witness lists and tables of Algorithm 2 represent all ϵ-BNE. Therefore, Algorithm 2 is guaranteed to converge to an ϵ-BNE.

Theorem 3. *For arbitrary structured graphical games with discrete types stage 1 in Algorithm 2 converges in at most $\frac{nk}{\tau^{4(l-1)}}$ rounds.*

Proof. The total number of entries in each table D is $O(\frac{1}{\tau^{2*2(l-1)}})$ since the number of entries is determined by the number of joint strategies of two agents with two possible types. Every round r before the algorithm converges has to change at least one entry in one table. Therefore, stage 1 of the Algorithm 2 has to converge in at most $\frac{nk}{\tau^{4(l-1)}}$ rounds, where k is the maximum degree of any node in the graph.

Since our work in this chapter extends the existing NashProp algorithm, we don't expect scalability and complexity to be significantly different from [9]. Instead we report the dynamics in market-maker prices and utilities which demonstrate the behavior, important features and successful operation of our model in a distributed prediction markets setting.

Proposition 1. *Algorithm 2 applied to a distributed prediction market problem that uses utility function given in Equation 2 encourages truthful revelation.*

Proof. For the simplicity of notation we show the proof for two agents i and j, with two possible types θ_{iopt} and θ_{ipes}, but the proof is extendable to multiple agents with several possible types. We want to show that the expected utility that the agent i gets when choosing action A_i^{true} truthfully is greater or equal to the expected utility it gets when it chooses action A_i^{false}, i.e. $EU_i^{true} \geq EU_i^{false}$. Since there are only two possible actions $A_i = \{\lambda_i, -\lambda_i\}$, misreporting would mean that when Algorithm 1 recommends agent i to take action $A_i = \lambda_i$, it takes action $-\lambda_i$ instead; i.e. $A_i^{false} = -A_i^{true}$. From definition of utility given in Equation 2 we can express the utility as $u_i(\lambda_i) = (T-t)e^{-|A_i|(T-t)}$. Now,

$$EU_i^{true} = \alpha p(\theta_{jopt}|\theta_{iopt})(T-t)\left(e^{-|A_i^{true}|(T-t)}\right)$$
$$+ (1-\alpha)p(\theta_{jpes}|\theta_{iopt})(T-t)\left(e^{-|A_i^{true}|(T-t)}\right)$$
$$\text{and } EU_i^{false} = \alpha p(\theta_{jopt}|\theta_{iopt})(T-t)\left(e^{-|-A_i^{true}|(T-t)}\right)$$
$$+ (1-\alpha)p(\theta_{jpes}|\theta_{iopt})(T-t)\left(e^{-|-A_i^{true}|(T-t)}\right).$$

Since $|A_i^{true}| = |-A_i^{true}|$, we get $EU_i^{true} = EU_i^{false}$. Therefore, agent i does not have any incentive to reveal its action untruthfully.

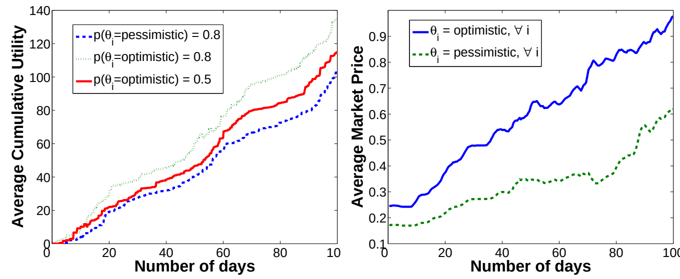
4 Experimental Results

We have conducted several simulations using our algorithm for a distributed prediction market setting to observe and verify the effect of different parameters on the evolution of market maker utilities and prices in the markets. To make it easier to analyze the effect of different parameters, in all of our simulations we assume that the number of neighbors is fixed for each market maker. For each experiment we vary one set of parameters as specified in each set of experiments and we hold the other parameters fixed at their default values given in Table 1.

Name	**Symbol**	**Value**
Number of market makers	N	20
Market price's jump scaling parm.	ζ	0.01
Confidence that other market maker is of the same type	α	0.5
Prob. distr. over types	p	$U[0,1]$
Discretization parameter	τ	0.3
Approximation value	ϵ	0.05
Number of actions	$l_i \forall i$	2
Number of neighbors	$\eta_i \forall i$	8

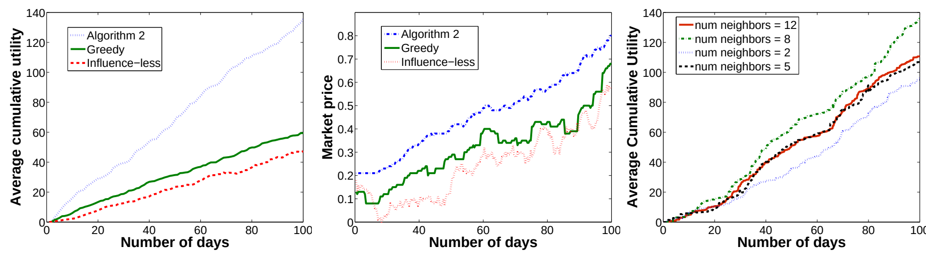
Fig. 1. Parameters used for our simulation experiments.

For our first set of experiments, we vary p, the probability distribution over types. We allow for three types of market maker populations: mostly pessimistic, equal number of optimistic and pessimistic, and optimistic. Figure 2(a) shows the cumulative utility averaged over all 20 market makers, where the market maker population is either 80% pessimistic, 50% pessimistic and 50% optimistic, and 80% optimistic. The type of each market maker is determined at the beginning of the prediction market and it does not change over time. We observe that when the majority of market makers is pessimistic the average utility is 23% less than when the majority of market makers is optimistic. This is because optimistic market makers' strategy selection is affected by the average strategies of their neighbors, whereas pessimistic market makers' strategy selection is affected by just one other market maker choosing a pessimistic strategy. In Figure 2(b) we continue analyzing the effect of different market makers'



a b

Fig. 2. The average cumulative utility(a) and the average market price(b) for different types of market makers.



a b c

Fig. 3. The average cumulative utility(a) and the market price(b) of one market maker using Algorithm 2, Greedy, or Influence-less markets strategies, the cumulative utility averaged over all neighbors for different number of neighbors, i.e. $2, 5, 8$, and 12(c).

types by looking at the average market price produced by the optimistic and the pessimistic market makers for the outcome of the event that happens (market price $= 1$). We note that optimistic market makers are able to predict a more accurate market price as opposed to the pessimistic ones. Again, this is due to the optimistic market makers taking into account the average strategies of all the market makers in their neighborhood. However, in real prediction market, there may be a mix of different types of market makers. Therefore, for the default setting for the type distribution we assume that there is about the same number of optimistic market makers as pessimistic ones.

Next, we compare our Algorithm 2 to two other strategies:

- **Greedy strategy:** In this setting, each agent i chooses λ_i that maximizes immediate utility given in Equation 2. This strategy does not consider the types of the market makers.

- **Influence-less markets**: In this setting, we consider conventional single, isolated markets where the market price is determined by the market maker based on that market's traders' decisions only. This setting is the completely opposite scenario of a distributed prediction market and it captures the effect of inter-market influences on the market makers' utilities and prices.

These two strategies were used for comparison as there are no other existing strategies for distributed prediction market setting. To abstract the details of the traders' decisions, we have assumed that each agent i uses a derivative follower (DF) [4] strategy where it keeps on increasing its market price π_i by δ_i until its immediate utility u_i starts decreasing, at that time agent i starts decreasing π_i by δ_i. This strategy does not consider the types or the interaction among market makers.

Figures 3 (a) and (b) show the utilities and market prices for the market makers using our proposed algorithm, greedy strategy, or influence-less markets strategy correspondingly. We observe that market makers using our Algorithm 2 obtain 56% more utility than the market makers following the next best greedy strategy. We also note that market prices fluctuate more when market makers use a greedy strategy or are in influence-less markets than when they use WBGG because greedy and influence-less market strategies are myopic and do not consider market makers' types. Finally, this result provides an important justification of our work - that, as compared to influence-less markets with isolated market makers, interacting market makers in a distributed prediction market are able to improve their utilities and predict prices with less fluctuations.

Finally, we report the results for different number of market makers and different number of neighbors of each market maker. Figure 3(c) shows the cumulative utility averaged over neighboring market makers for a setting with 20 market makers. We can see that when market makers have a small number of neighbors then they get less utility than when the number of neighbors is larger, up to a certain point. For example, market makers with 2 neighbors (10% of the total number of market makers) get 29% less utility than when market makers have 8 neighbors (40% of the total number of market makers). However, this relationship is not linear, i.e. when market makers have 12 neighbors (60% of the total number of market makers) they get 19% less utility than when market makers have 8 neighbors. We posit that increasing the number of neighbors up to a point translates to an increased utility because the market maker can improve its decision based on the information of its neighbors, but having too many neighbors may end up creating more noisy information for the market maker.

We also conduct experiments showing the scalability of our Algorithm 2 with respect to the number of market makers, and our results show that our algorithm scales linearly with the number of market makers and that the running time increases with the increased number of neighbors.

5 Conclusion

In this paper we proposed a novel distributed prediction market setting where the aggregated (market) price of a security of an event in one prediction market is affected dynamically by the prices of securities of similar events in other, simultaneously running prediction markets. Our proposed formal framework, called a weighted Bayesian graphical game (WBGG), is able to capture the local interactions between multiple market makers and uses the Bayes-Nash equilibrium concept to find a suitable action for each market maker in a WBGG. Our experimental results showed that our algorithm results in higher utilities and more accurate prices in comparison to a greedy strategy or a disjoint prediction markets.

In the future, we plan to extend our current distributed prediction market model to include direct influences between traders across multiple prediction markets. We also want to extend our algorithm to study stochastic graphical games to model uncertainty in repeated games. We also plan to extend our algorithm to make it strategy-proof for graphical games in general. In summary, the novel framework for distributed prediction markets that we introduced in this paper can lead to several challenging and important directions that can help to gain a better understanding of the distributed information aggregation problem.

References

1. Y. Chen and I. Kash. Information elicitation for decision making. In *Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems(AAMAS 2011)*, pages 175–182, 2011.
2. Y. Chen, T. Mullen, and C. Chu. An in-depth analysis of information markets with aggregate uncertainty. *Electronic Commerce Research*, 6(2):201–222, 2006.
3. J. Feigenbaum, L. Fortnow, D. Pennock, and R. Sami. Computation in a distributed information market. *Theor. Computer Science*, 343(1-2):114–132, 2005.
4. A. Greenwald and J. Kephart. Shopbots and pricebots. In *Sixteenth International Joint Conference on Artificial Intelligence, Stockholm*, pages 506–511, 1999.
5. R. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1(1):3–15, 2007.
6. D. Jrgen. Price competition between market makers. *The Review of Economic Studies*, 60(3):pp. 735–751, 1993.
7. M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 253–260, 2001.
8. B. Myerson. Bayesian equilibrium and incentive-compatibility: An introduction. In *Social Goals and Social Organization: Essays in Memory of Elisha Pazner*, pages 229–259, 1985.
9. L. Ortiz and M. Kearns. Nash propagation for loopy graphical games. In *In Neural Information Processing Systems*, pages 793–800, 2003.
10. Othman, A. and Sandholm, T. Automated Market Making in the Large: The Gates Hillman Prediction Market. In *Proceedings of the 11th ACM Conference on Electronic Commerce(EC 2010)*, pages 367–376, 2010.
11. W. Pan, Y. Altshuler, and A. Pentland. Decoding Social Influence and the Wisdom of the Crowd in Financial Trading Network. In *IEEE Social Computing*, 2012.
12. S. Singh, V. Soni, and M. Wellman. Computing approximate bayes-nash equilibria in tree-games of incomplete information. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC 2005)*, pages 81–90, 2004.
13. D. Vickrey. Multi-agent algorithms for solving graphical games. In *Proc. of the National Conference on Artificial Intelligence*, pages 345–351, 2002.
14. J. Wolfers and E. Zitzewitz. Prediction Markets. *Journal of Econ. Perspectives*, 18(2):107–126, 2004.