Prediction Market-based Information Aggregation for Multi-sensor Information Processing

Janyl Jumadinova and Prithviraj Dasgupta

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

Abstract. Prediction markets have been shown to be a useful tool in forecasting the outcome of future events by aggregating public opinion about the event's outcome. We consider an analogous problem of information fusion from multiple sensors of different types with the objective of improving the confidence of inference tasks, such as object classification. We develop a multi-agent prediction market-based technique to solve this information fusion problem. To monitor the improvement in the confidence of the object classification as well as to dis-incentivize agents from misreporting information, we have introduced a market maker that rewards the agents based on the quality of the submitted reports. We have implemented the market maker's reward calculation in the form of a *scoring rule* and have shown analytically that it incentivizes truthful revelation by each agent. We have experimentally verified our technique for multi-sensor information fusion for an automated landmine detection scenario. Our experimental results show that, for identical data distributions and settings, using our information aggregation technique increases the accuracy of object classification favorably as compared to two other commonly used techniques for information fusion for landmine detection. [1]

Keywords: Prediction market, information fusion, multi-sensor aggregation, landmine detection

1 Introduction

A prediction market is a market-based aggregation mechanism that is used to combine the opinions on the outcome of a future, real-world event from different people and forecast the event's possible outcome based on their aggregated opinion [22]. Prediction markets have shown to be a very successful tool in forecasting the outcome of future events as is evidenced from the successful predictions of actual events done by prediction markets run by the Iowa Electronic Marketplace(IEM), Tradesports, Intrade, and by companies such as Hewlett Packard, Google and Yahoo's Yootles. The main idea behind the prediction market paradigm is that the collective, aggregated beliefs of humans on a

[1] This research has been sponsored as part of the COMRADES project funded by the Office of Naval Research, grant number N000140911174.

future event represents the probability of occurrence of the event more accurately than the corresponding surveys and opinion polls.

Multi-sensor fusion is concerned with the problem of fusing data from multiple sensors in order to make a more accurate estimation of the environment, and has been a central research topic in sensor-based systems [21]. Our work in this paper is based on the insight that the problem addressed by prediction markets of aggregating the beliefs of different humans to forecast the outcome of an initially unknown event is analogous to the problem in multi-sensor fusion of fusing information from multiple sources to predict the outcome of an initially unknown object. Recently several multi-agent techniques [20] have been proposed to address the multi-sensor fusion problem. Most of the solutions for multi-sensor information fusion and processing are based on Bayesian inference techniques [8, 14, 17]. While such techniques have been shown to be very effective, we investigate a complimentary problem where sensors can behave in a self-interested manner. Such self-interested behavior can be motivated by the sensors that deliberately misreport to save power, CPU cycles and memory (for example, to devote more resources to other tasks), and therefore always behave in a way that maximize their own benefits [10, 16]. Moreover, these sensors may be programmed by different people and the assumption of full cooperation may not always hold [19]. To address this problem, we develop a prediction market for multi-sensor information fusion that includes a utility driven mechanism to motivate each sensor, through its associated agent, to reveal accurate reports.

To motivate our problem we describe a distributed automated landmine detection scenario used for humanitarian demining. An environment contains different buried objects, some of which could potentially be landmines. A set of robots, each equipped with one of three types of landmine detection sensor such as a metal detector (MD), or a ground penetrating radar (GPR) or an infrared (IR) heat sensor, are deployed into this environment. Each robot is capable of perceiving certain features of a buried object through its sensor such as the object's metal content, area, burial depth, etc. However, the sensors give noisy readings for each perceived feature depending on the characteristics of the object as well as on the characteristics of the environment (e.g., moisture content, ambient temperature, sunlight, etc.). Consequently, a sensor that works well in one scenario, fails to detect landmines in a different scenario, and, instead of a single sensor, multiple sensors of different types, possibly with different detection accuracies can detect landmines with higher certainty [4]. Within this scenario, the central question that we intend to answer is: given an initial set of reports from *self-interested* sensors about the features of a buried object, what is a suitable set (number and type) of sensors to deploy over a certain time window to the object, so that, over this time window, the fused information from the different sensors successively reduces the uncertainty in determining the object's type.

Building on the previous models of prediction markets, in this paper, we describe a multi-agent prediction market for multi-sensor information fusion. Besides being an efficient aggregation mechanism, using prediction markets gives us several useful features - a mathematical formulation called a *scoring rule* that deters self-interested sensors from misreporting information, a regression-based belief update mechanism for the sensor agents for incorporating the aggregated beliefs (or information estimates) of other sensors into their own calculation, and the ability to incorporate an autonomous decision maker that uses expert-level domain knowledge to make utility maximizing decisions to deploy additional sensors appropriately to improve the detection of an object. Our experimental results illustrated with a landmine detection scenario while using identical data distributions and settings, show that the information fusion performed using our technique reduces the root mean squared error by $5-13\%$ as compared to a previously studied technique for landmine data fusion using the Dempster-Shafer theory [11] and by $3-8\%$ using distributed data fusion technique [9].

2 Related Work

Multi-agent Information Fusion. Multi-agent systems have been used to solve various sensor network related problems and an excellent overview is given in [20]. In the direction of multi-sensor information processing, significant works include the use of particle filters[17], distributed data fusion (DDF) architecture along with its extension, the Bayesian DDF [8, 9], Gaussian processes [14] and mobile agent-based information fusion [23]. For most of the application domains described in these works such as water-tide height measurement, wind speed measurement, robot tracking and localization, etc., self-interested behavior by the sensors is not considered a crucial problem. For our illustrated application domain of landmine detection, decision-level fusion techniques have been reported to be amenable for scenarios where the sensor types are different from each other, and, non-statistical decision-level fusion techniques, such as Dempster-Shafer theory [11], fuzzy logic [3], and rule-based fusion techniques [4] have been reported to generalize well. However, in contrast to our work, these techniques assume that sensors are fully cooperative and never behave self-interestedly by misreporting information. Most sensor-based information aggregation techniques either do not consider selfish behavior or use high-overhead, cryptographic techniques to combat it [13]. To deter false reports by sensor nodes in a data aggregation setting, researchers propose various lower overhead reputation and trust-based schemes [13, 18]. Our prediction market-based information aggregation technique is complimentary to such reputation-based aggregation techniques.

Decision-Making using Prediction Markets. A prediction market is a market-based aggregation mechanism that is 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. Recently, multi-agent systems have been used [7, 15] to analyze the operation of prediction markets, where the behaviors of the market's participants are implemented as automated software agents. The seminal work on prediction market analysis [22] has shown that the mean belief values of individual traders about the outcome of a future event corresponds to the event's market price. 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. Hanson [5] developed a mechanism, called a *scoring rule*, that can be used by market makers to reward traders for making and improving a prediction about the outcome of an event, and, showed that if a scoring rule is *proper* or incentive compatible, then it can serve as an automated market maker. Recently, authors in [2, 15] have theoretically analyzed the properties of prediction markets used for decision making. In [15], the authors analyzed the problem of a decision maker manipulating a prediction market and proposed a family of scoring rules to address the problem. In [2], the authors extended this work by allowing randomized decision rules, considering multiple possible outcomes and providing a simple test to determine whether the scoring rule is proper for an arbitrary decision rule-scoring rule pair. In this paper, we use a prediction market for decision making, but in contrast to previous works we consider that the decision maker can make multiple, possibly improved decisions over an event's duration, and, the outcome of an event is decided independently, outside the market, and not influenced by the decision maker's decisions. Another contribution our paper makes is a new, proper scoring rule, called the *payment function*, that incentivizes agents to submit truthful reports.

3 Problem Formulation

Let L be a set of objects. Each object has certain features that determine its type. We assume that there are f different features and m different object types. Let $\Phi = \{\phi_1, \phi_2, ..., \phi_f\}$ denote the set of object features and $\Theta = \{\theta_1, \theta_2, ..., \theta_m\}$ denote the set of object types. The features of an object $l \in L$ is denoted by $l_\Phi \subseteq \Phi$ and its type is denoted by $l_\theta \in \Theta$. As illustrated in the example given in Section 1, l_Φ can be perceived, albeit with measurement errors, through sensors, and, our objective is to determine l_θ as accurately as possible from the perceived but noisy values of l_Φ. Let $\Delta(\Theta) = \{(\delta(\theta_1), \delta(\theta_2), ..., \delta(\theta_m)) : \delta(\theta_i) \in [0, 1], \sum_{i=1}^{m} \delta(\theta_i) = 1\}$, denote the set of probability distributions over the different object types. For convenience of analysis, we assume that when the actual type of object l, $l_\theta = \theta_j$, its (scalar) type is expanded into a m-dimensional probability vector using the function $vec : \Theta \to [0, 1]^m : vec_j = 1, vec_{i \neq j} = 0$, which has 1 as its j-th component corresponding to l's type θ_j and 0 for all other components.

Let A denote a set of agents (sensors) and $A_{rep}^{t,l} \subseteq A$ denote the subset of agents that are able to perceive the object l's features on their sensors at time t. Based on the perceived object features, agent $a \in A_{rep}^{t,l}$ at time t reports a belief as a probability distribution over the set of object types, which is denoted as $\mathbf{b}^{a,t,l} \in \Delta(\Theta)$. The beliefs of all the agents are combined into a composite belief, $\mathbf{B}^{t,l} = Agg_{a \in A_{rep}^{t,l}}(\mathbf{b}^{a,t,l})$, and let $\hat{\Theta}^{t,l} : \mathbf{B}^{t,l} \to \Delta(\Theta)$ denote a function that computes a probability distribution over object types based on the aggregated agent beliefs. Within this setting we formulate the object classification problem as a decision making problem in the following manner: given an object l and an initial aggregated belief $\mathbf{B}^{t,l}$ calculated from one or more agent reports for that object, determine a set of additional agents (sensors) that need to be deployed at object l such that the following constraint is satisfied:

$$\min RMSE\left(\hat{\Theta}^{t,l}, vec(l_\theta)\right), \quad \text{for} \quad t = 1, 2, ...T, \tag{1}$$

where T is the time window for classifying an object l and RMSE is the root mean square error given by $RMSE(\mathbf{x}, \mathbf{y}) = \frac{||x-y||}{\sqrt{m}}$. In other words, at every time step t, the decision maker tries to select a subset of agents such that the root mean square error (RMSE) between the estimated type of object l and its actual type is successively minimized.

The major components of the object classification problem described above consists of two parts: integrating the reports from the different sensors and making sensor deployment decisions based on those reports so that the objective function given in Equation 1 is satisfied. To address the first part, we have used distributed information aggregation with a multi-agent prediction market, while for the latter we have used an expected utility maximizing decision-making framework. A schematic showing the different components of our system and their interactions is shown in Figure 1 and explained in the following sections.

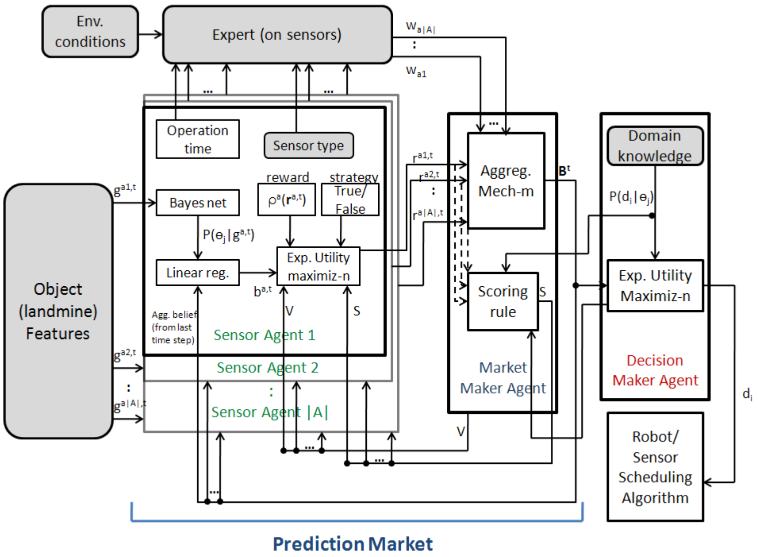

Fig. 1. The different components of the prediction market for decision making and the interactions between them.

3.1 Sensor Agents

As mentioned in Section 1, there is a set of robots in the scenario and each robot has an on-board sensor for analyzing the objects in the scenario. Different robots can have different types of sensors and sensors of the same type can have different degrees of accuracy determined by their cost. Every sensor is associated with a software agent that runs on-board the robot and performs calculations related to the data sensed by the robot's sensor. In the rest of the paper, we have used the terms sensor and agent interchangeably. For the ease of notation, we drop the subscript l corresponding to an object for the rest of this section.

When an object is within the sensing range of a sensor (agent) a at time t, the sensor observes the object's features and its agent receives this observation in the form of an information signal $g^{a,t} =< g_1, ..., g_f >$ that is drawn from the space of information signals $G \subseteq \Delta(\Theta)$. The conditional probability distribution of object type θ_j given an information signal $g \in G$, $P(\theta_j|g) : G \to [0, 1]$, is constructed using domain knowledge $[3, 11, 12]$ within a Bayesian network and is made available to each agent. Agent a then updates its belief distribution $\mathbf{b}^{a,t}$ using the following equation:

$$\mathbf{b}^{a,t} = w_{bel} \cdot \mathbf{P}(\Theta|g^{a,t}) + (1 - w_{bel}) \cdot \mathbf{B}^t, \tag{2}$$

where $\mathbf{B}^t$ is the belief value vector aggregated from all sensor reports.

Agent Rewards. Agents behave in a self-interested manner to ensure that they give their 'best' report using their available resources including sensor, battery power, etc. An agent a that submits a report at time t, uses its belief distribution $\mathbf{b}^{a,t}$ to calculate the report $\mathbf{r}^{a,t} =< r_1^{a,t}, ..., r_m^{a,t} >\in \Delta(\Theta)$. An agent can have two strategies to make this report - truthful or untruthful. If the agent is truthful, its report corresponds to its belief, i.e., $\mathbf{r}^{a,t} = \mathbf{b}^{a,t}$. But if it is untruthful, it deliberately reports an inaccurate belief to save its belief computation costs. Each agent a can update its report $\mathbf{r}^{a,t}$ within the time window T by obtaining new measurements from the object and using Equation 2 to update its belief. The report from an agent a at time t is analyzed by a human or agent expert [11] to assign a weight $w^{a,t}$ depending on the current environment conditions and agent a's sensor type's accuracy under those environment conditions (e.g., rainy weather reduces the weight assigned to the measurement from an IR heat sensor, or, soil that is high in metal content reduces the weight assigned to the measurement from an metal detector).

To motivate an agent to submit reports, an agent a gets an instantaneous reward, $\rho^{a,t}$, from the market maker for the report $\mathbf{r}^{a,t}$ it submits at time t, corresponding to its instantaneous utility, which is given by the following equation:

$$\rho^{a,t} = V(n^{t'=1..t}) - C^a(\mathbf{r}^{a,t}), \tag{3}$$

where $V(n^{t'=1..t})$ is the value for making a report with $n^{t'=1..t}$ being the number of times the agent a submitted a report up to time t, and, $C^a(\mathbf{r}^{a,t})$ is the cost of making report $\mathbf{r}^{a,t}$ for agent a. $C^a(\mathbf{r}^{a,t})$ is equal to the actual cost of resources used by the robot, such as expended time, battery power, etc. if the agent uses truthful strategy, and it is equal to some value $c_\epsilon <<$ actual cost of resources if the agent uses untruthful strategy. We denote the agent's value for each report $V(n^{t'=1..t})$ as a constant-valued function up to a certain threshold and a linearly decreasing function thereafter, to de-incentivize agents from making a large number of reports. Agent a's value function is given by the following equation:

$$V(n^{t'=1..t}) = \begin{cases} \nu & , n^{t'=1..t} \leq n^{threshold} \\ \frac{\nu(n^{t'=1..t} - n^{max})}{(n^{threshold} - n^{max})} & , otherwise \end{cases}$$

where $\nu \in \mathrm{Z}^+$, is a constant value that a gets by submitting reports up to a threshold, $n^{threshold}$ is the threshold corresponding to the number of reports a can submit before its report's value starts decreasing, and, n^{max} is the maximum number of reports agent a can submit before V becomes negative. Finally, to determine its strategy while submitting its report, an agent selects the strategy that maximizes its expected utility obtained from its cumulative reward given by Equation 3 plus an expected value of its final reward payment if it continues making similar reports up to the object's time window T.

3.2 Decision Maker Agent

The decision maker agent's task is to use the composite belief about an object's type, $\mathbf{B}^t$, given by the prediction market, and take actions to deploy additional robots(sensors) based on the value of the objective function given in Equation 1. Let AC denote a set of possible actions corresponding to deploying a certain number of robots, and $D = \{d_1, ...d_h\} : d_i \in Ac \subseteq AC$ denote the decision set of the decision maker, where h is the number of decisions the decision maker has. The decision function of the decision maker is given by $dec : \Delta(\Theta) \rightarrow D$. Let $\mathbf{u}_j^{dec} \in \mathrm{R}^m$ be the utility that the decision maker receives by determining an object to be of type θ_j and let $P(d_i|\theta_j)$ be the probability that the decision maker makes decision $d_i \in D$ given object type θ_j. $P(d_i|\theta_j)$ and $\mathbf{u}_j^{dec}$ are constructed using domain knowledge [3, 11, 12]. Given the aggregated belief distribution $\mathbf{B}^t$ at time t, the expected utility to the decision maker for taking decision d_i at time t is then $EU^{dec}(d_i, \mathbf{B}^t) = \sum_{j=1}^{m} P(d_i|\theta_j) \cdot \mathbf{u}_j^{dec} \cdot \mathbf{B}^t$. The decision that the decision maker takes at time t, also called its *decision rule*, is the one that maximizes its expected utility and is given by: $d^t = \arg\max_{d_i} EU^{dec}(d_i, \mathbf{B}^t)$.

3.3 Prediction Market

A conventional prediction market uses the aggregated beliefs of the market's participants or traders about the outcome of a future event, to predict the event's outcome. The outcome of an event is represented as a binary variable (event happens/does not happen). The traders observe information related to the event and report their beliefs, as probabilities about the event's outcome. The market maker aggregates the traders' beliefs and uses a scoring rule to determine a payment or payoff that will be received by each reporting trader. In our multi-agent prediction market, traders correspond to sensor agents, the market maker agent automates the calculations on behalf of the conventional market maker, and, an event in the conventional market corresponds to identifying the type of a detected object. The time window T over which an object is sensed is called the duration of the object in the market. This time window is divided into discrete time steps, $t = 1, 2, ..., T$. During each time step, each sensor agent observing the object submits a report about the object's type to the market maker agent. The market maker agent performs two functions with these reports. First, at each time step t, it aggregates the agent reports into an aggregated belief about the object, $\mathbf{B}^t \in \Delta(\Theta)$. Secondly, it calculates and distributes payments for the sensor agents. It pays an immediate but nominal reward to each agent for its report at time step t using Equation 3. Finally, at the end of the object's time window T, the market maker also gives a larger *payoff* to each agent that contributed towards classifying the object's type. The calculations and analysis related to these two functions of the market maker agent are described in the following sections.

Final Payoff Calculation. The payoff calculation for a sensor agent is performed by the market maker using a *decision scoring rule* at the end of the object's time window. A decision scoring rule [2] is defined as any real valued function that takes the agents' reported beliefs, the realized outcome and the decisions made by the decision maker as input, and produces a payoff for the agent for its reported beliefs, i.e. $S : \Delta(\Theta) \times \Theta \times D \longrightarrow$ R. We design a scoring rule for decision making that is based on how much agent a's final report helped the decision maker to make the right decisions throughout the duration of the prediction market and by how close the agent a's final report is to actual object type. Our proposed scoring rule for decision making given that object's true type is θ_j is given in Equation 4:

$$S(r_j^{a,t}, d^{[1:t]}, \theta_j) = \varpi(d^{[1:t]}, \theta_j) log \left(r_j^{a,t} \right), \tag{4}$$

where, $r_j^{a,t}$ is the reported belief that agent a submitted at time t for object type θ_j, $d^{[1:t]}$ is the set consisting of all the decisions that the decision maker took related to the object up to the current time t, θ_j is the object's true type that was revealed at the end the object's time window, $log \left(r_j^{a,t} \right)$ measures the goodness of the report at time t relative to the true object type θ_j, and, $\varpi(d^{[1:t]}, \theta_j)$ is the weight, representing how good all the decisions the decision maker took up to time t were compared to the true object type θ_j. $\varpi(d^{[1:t]}, \theta_j)$ is determined by the decision maker and made available to the agents through the market maker. We assume that $\varpi(d^{[1:t]}, \theta_j) = \sum_{i=1}^{t} P(d_i \mid \theta_j) \cdot \mathbf{u}_j^{dec}$, which gives the expected utility of the decision maker agent for making decision i when the true type of the object is θ_j.

Aggregation. Since a sensor agent gets paid both through its immediate rewards for making reports during the object's time window and through the scoring rule function for decision making at the end of the object's time window, we define the total payment that the agent has received by the end of the object's time window as a *payment function.*

Definition 1. *A function $\Psi(\mathbf{r}^{a,t}, d^{[1:t]}, \theta_j, n^{t'=1..t})$ is called a* ***payment function*** *if each agent a's total received payment at the end of the object's time window (when $t = T$) is*

$$\Psi(\mathbf{r}^{a,t}, d^{[1:t]}, \theta_j, n^{t'=1..t}) = \sum_{k=1}^{t} \rho^{a,k} + S(r_j^{a,t}, d^{[1:t]}, \theta_j) \tag{5}$$

where $\rho^{a,k}$, $S(r_j^{a,t}, d^{[1:t]}, \theta_j)$ and their components are defined as in Equations 3 and 4.

Let Ψ^{ave} denote a weighted average of the payment function in Equation 5 over all the reporting agents, using the report-weights assigned by the expert in Section 3.1, as given below:

$$\Psi^{ave}(\mathbf{r}^{A_{rep}^t,t}, d^{[1:t]}, \theta_j, n^{A_{rep}^t,t}) = \sum_{k=1}^{t} \sum_{a \in A_{rep}^t} w^{a,k} \rho^{a,k} + \varpi(d^{[1:t]}, \theta_j) \sum_{a \in A_{rep}^t} w^{a,t} log \left(r_j^{a,t} \right), \tag{6}$$

where A_{rep}^t is the subset of agents that are able to perceive object feature at time t and $w^{a,k}$ is the weight assigned to agent a at time k by the expert. To calculate an aggregated belief value in a prediction market, Hanson [5] used the generalized inverse function of the scoring rule. Likewise, we calculate the aggregated belief for our market maker agent by taking the generalized inverse of the average payment function given in Equation 6:

$$B_j^t = Agg_{a \in A_{rep}^t}(\mathbf{b}^{a,t}) = \frac{\frac{exp\left(\Psi^{ave} - \sum_{k=1}^{t} \sum_{a \in A_{rep}^t} w^{a,k} \rho^{a,k}\right)}{\varpi(d^{[1:t]}, \theta_j)}}{\sum_{\theta_j=\theta_1}^{\theta_m} \frac{exp\left(\Psi^{ave} - \sum_{k=1}^{t} \sum_{a \in A_{rep}^t} w^{a,k} \rho^{a,k}\right)}{\varpi(d^{[1:t]}, \theta_j)}} \tag{7}$$

where $B_j^t \in \mathbf{B}^t$ is the j-th component of the aggregated belief for object type θ_j. The aggregated belief vector, $\mathbf{B}^t$, calculated by the market maker agent is sent to the decision maker agent so that it can calculate its expected utility given in Section 3.2, as well as, sent back to each sensor agent that reported the object's type till time step t, so that the agent can refine its future reports, if any, using this aggregate of the reports from other agents. We also note that the aggregation mechanism used by our market maker is similar to an LMSR technique which was shown to lead to Weak Perfect Bayesian Equilibrium when, like in our setting, the information signals of the traders are independent conditional on the state of the world [1].

4 Payment function: Properties and Characteristics

In this section we first show that the payment function is proper, or incentive compatible. Then we show that when the market maker uses this payment function to reward each agent for its reported beliefs, reporting beliefs truthfully is the optimal strategy for each agent.

We can characterize a proper payment function similar to a proper scoring rule.

Definition 2. *A payment function Ψ is proper, or incentive compatible, if*

$$\Psi(\mathbf{b^{a,t}}, d^{[1:t]}, \theta_j, n^{t'=1..t}) \geq \Psi(\mathbf{r^{a,t}}, d^{[1:t]}, \theta_j, n^{t'=1..t}), \tag{8}$$

$\forall \mathbf{b^{a,t}}, \mathbf{r^{a,t}} \in \Delta(O)$. *$\Psi$ is strictly proper if the inequality in Equation 8 strict.*

Proposition 1. *The payment function given in Equation 5 is proper.*
The proof is given in [6].

Agent Reporting Strategy. Assume that agent a's report at time t is its final report, then its utility function can be written as $u_j^{a,t} = \sum_{k=1}^{t} \rho^{a,k} + S(r_j^{a,t}, d^{[1:t]}, \theta_j)$. Then, agent a's expected utility for object type θ_j given its reported belief for object type θ_j, $r_j^{a,t}$, and its true belief about object type θ_j, $b_j^{a,t}$ at time t is

$$EU_j^a(r_j^{a,t}, b_j^{a,t}) = \sum_{i=1}^{h} P(d_i|\theta_j) b_j^{a,t} u_j^{a,t} = \sum_{i=1}^{h} P(d_i|\theta_j) b_j^{a,t} \left(\sum_{k=1}^{t} \rho^{a,k} + S(r_j^{a,t}, d^{[1:t]}, \theta_j) \right), \tag{9}$$

where $P(d_i|\theta_j)$ is the probability that the decision maker takes decision d_i when the object's type is θ_j.

Proposition 2. *If agent a is paid according to Ψ, then it reports its beliefs about the object types truthfully.*
The proof is a straight forward solution to expected utility maximization problem. The complete proof is given in [6].

Name	**Value**
Object types	mine, metallic object(non-mine), non-metallic object(non-mine)
Features	metallic content, object's area, object's depth, sensor's position
Sensor types	MD, IR, GPR
Max no. of sensors	10 (5MD,3IR,2GPR)
Max no. of decisions	14
T (object ident. window)	10
ν (value if $n^{t'=1..t} \leq$) $n^{threshold}$)	5
n_{max} (max no. of reports before value is negative)	20
$n_{threshold}$ (no. of reports before value $< \nu$)	5

Table 1. Parameters used for our simulation experiments.

Object type	**Time steps**	**PM**	**DDF**	**D-S**
Mine	1	1(1MD)	1(1MD)	1(1MD)
	2	3(1MD, 1IR)	3(1MD, 1GPR)	3(1IR, 1GPR)
	3	4(1GPR)	5(1MD,1IR)	4(1MD)
	4	5(1MD)	6(1IR)	5(1MD)
	5	6(1MD)	7(11MD)	6(1IR)
	6	7(1IR)	8(1MD)	7(1IR)
	7	-	9(1IR)	8(1MD)
Met.	1	1(1MD)	1(1MD)	1(1MD)
or	2	3(1MD, 1IR)	4(1MD,1IR, 1GPR)	3(1IR, 1GPR)
Fr.	3	4(1GPR)	5(1MD)	4(1MD)
D-S	4	5(1MD)	6(1IR)	5(1IR)
	5	6(1IR)	7(11MD)	6(11IR)
	6	7(1MD)	8(1IR)	7(1MD)
	7	8(1IR)	9(1GPR)	8(1MD)
	8	-	9(1MD)	8(1MD)
Non-	1	1(1MD)	1(1MD)	
met.	2	2(1MD)	2(1IR)	
	3	3(1IR)	3(1MD)	
	4	4(1MD)	4(1GPR)	
	5	5(1IR)	5(1MD)	
	6	6(1MD)	6(1IR)	
	7	-	7(1MD)	

Table 2. Different number of sensors and the sensor types deployed over time by a decision maker to classify different types of objects.

5 Experimental Results

We have conducted several experiments using our aggregation technique for decision-making within a multi-sensor landmine detection scenario described in Section 1. Our environment contains different buried objects, some of which are landmines. The true types of the objects are randomly determined at the beginning of the simulation. Due to the scarcity of real data related to landmine detection, we have used the domain knowledge that was reported in [3, 11, 12] to determine object types, object features, sensor agents' reporting costs, decision maker agent's decision set, decision maker agent's utility of determining objects of different types, and, to construct the probability distributions for $P(\theta_j|g)$ and $P(d_i|\theta_j)$. We report simulation results for root mean squared error (RMSE) defined in Section 3 and also for number of sensors over time, cost over object types, and average utility of the sensors over time.

Compared Techniques. For comparing the performance of our prediction market based object classification techniques, we have used two other well-known techniques for information fusion: (a) Dempster-Shafer (D-S) theory for landmine classification [11], where a two-level approach based on belief functions is used. At the first level, the detected object is classified according to its metal content. At the second level the chosen level of metal content is further analyzed to classify the object as a landmine or a friendly object. The belief update of the sensors that we used for D-S method is the same one we have described in Section 3.1. (b) Distributed Data Fusion (DDF) [9], where sensor measurements are refined over successive observations using a temporal, Bayesian inference-based, information filter. To compare DDF with our prediction market-based technique, we replaced our belief aggregation mechanism given in Equation 7 with a DDF-based information filter. We compare our techniques using some standard evaluation metrics from multi-sensor information fusion [14]: root mean squared error (RMSE) defined as in Section 3, normed mean squared errors (NMSE) calculated as: $NMSE^t(\hat{\Theta}^t - vec(\theta_j)) = 10\, log_{10} \frac{\frac{1}{m}\sum_{j=1}^{m}\left(\hat{\Theta}_j^t, vec(\theta_j)\right)^2}{\frac{1}{m}\left(\sum_{j=1}^{m} vec(\theta_j)^2\right) - \left(\frac{1}{m}\sum_{j=1}^{m} vec(\theta_j)\right)^2}$, and, the information gain, also known as Kullback-Leibler divergence and relative entropy, calculated as: $D_{KL}^t(\hat{\Theta}^t || vec(\theta_j) = \sum_{j=1}^{m} \hat{\Theta}_j^t log\left(\frac{\hat{\Theta}_j^t}{vec(\theta_j)}\right)$.

$\hat{\Theta}^t$ was calculated using D-S, DDF, and our prediction market technique ($\hat{\Theta}^t = \mathbf{B}^t$). Since the focus of our work is on the quality of information fusion, we will concentrate on describing the results for one object. We assume that there are three types of sensors, MD (least operation cost, most noisy), IR (intermediate operation cost, moderately noisy), and GPR (expensive operation cost, most accurate). We also assume that there are a total of 5 MD sensors, 3 IR sensors, and 2 GPR sensors available to the decision maker for classifying this object. Initially, the object is detected using one MD sensor. Once the object is detected, the time window in the prediction market for identifying the object's type starts. The MD sensor sends its report to the market maker in the prediction market and the decision maker makes its first decision based on this one report. We assume that decision maker's decision (sent to the robot/sensor scheduling algorithm in Figure 1) is how many $(0-3)$ and what type (MD,IR,GPR) of sensors to send to the site of the detected object subsequently. We have considered a set of 14 out of all the possible decisions under this setting. From [12], we derive four object features, which are metallic content, area of the object, depth of the object, and the position of the sensor. Combinations of the values of these four features constitute the signal set G and at each time step, a sensor perceiving the object receives a signal $g \in G$. The value of the signal also varies based on the robot/sensor's current position relative to the object. We assume that the identification of an object stops and the object type is revealed when either $\mathbf{B}_j^t \geq 0.95$, for any j, or after 10 time steps. The default values for all domain related parameters are shown in Table 1. All of our results were averaged over 10 runs and the error bars indicate the standard deviation over the number of runs. For our first group of experiments we analyze the performance of our technique

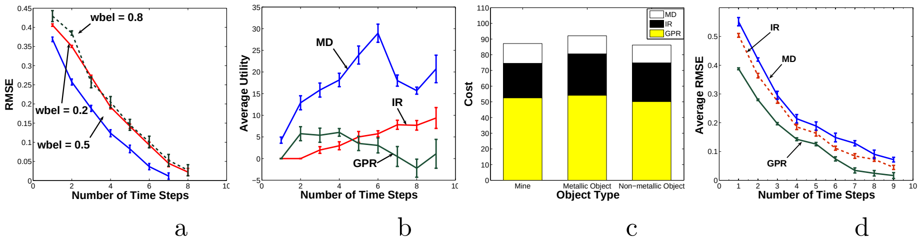

Fig. 2. RMSE for different values of w_{bel}(a), Average sensors' utilities for different sensor types(b), Cost for different object types(c), RMSE for sensors' reports averaged over sensor types(d).

w.r.t. the variables in our model, such as w_{bel} and time, and, w.r.t. to sensor and object types. We observe that as more information gets sensed for the object, the RMSE value, shown in Figure 2(a), decreases over time. It takes on average $6-8$ time steps to predict the object type with 95% or greater accuracy depending on the object type and the value of w_{bel}. We also observe that our model performs the best with $w_{bel} = 0.5$ (in Equation 2), when the agent equally incorporates its private signal and also the market's aggregated belief at each time step into its own belief update. Figure 2(b) shows the average utility of the agents based on their type. We can see that MD sensors get more utility because their costs of calculating and submitting reports are generally less, whereas GPR sensors get the least utility because they encounter the highest cost. This result is further verified in Figure 2(c) where we can see the costs based on sensor types and also based on object types. We observe that detecting a metallic object that is not a mine has the highest cost. We posit that it is because both MD and IR sensors can detect metallic content in the object and extra cost is due to the time and effort spent differentiating metallic object from a mine. Although most of the mines are metallic [11, 12], we can see that the cost of detecting a mine and a non-metallic object are similar because we require a prediction of at least 95%. Due to the sensitive nature of the landmine detection problem, it is important to ensure that even a non-metallic object is not a mine even if we encounter higher costs. However, despite MD's high utility (Figure 2(b)) and low cost (Figure 2(c)), its error of classifying the object type is the largest, as can be seen from Figure 2(d). In Table 2, we show how the decision maker's decisions using our prediction market technique results in the deployment of different numbers and types of sensors over the time window of the object. We report the results for the value of belief update weight $w_{bel} = 0.5$(used in Equation 2) while using our prediction market model, as well as using D-S and DDF. We see that non-metallic object classification requires less number of sensors as both MD and IR sensors can distinguish between metallic vs. non-metallic objects, and so, deploying just these two types of sensors can help to infer that the object is not a mine. In contrast, metallic objects require more time to get classified as not

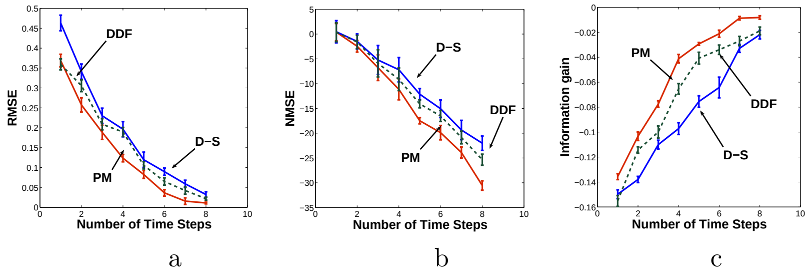

Fig. 3. Comparison of our Prediction Market-based information aggregation with Dempster-Shafer and Distributed Data Fusion Techniques using different metrics: RMSE(a), NMSE(b), Information gain(c).

being a mine because more object features using all three sensor types need to be observed. We also observe that on average our aggregation technique using prediction market deploys a total of $6-8$ sensors and detects the object type with at least 95% accuracy in $6-7$ time steps, while the next best compared DDF technique deploys a total of $7-9$ sensors and detects the object type with at least 95% accuracy in $7-8$ time steps.

Our results shown in Figure 3(a) illustrate that the RMSE using our PM-based technique is below the RMSEs using D-S and DDF by an average of 8% and 5% respectively. Figure 3(b) shows that the NMSE values using our PM-based technique is 18% and 23% less on average than D-S and DDF techniques respectively. Finally, in Figure 3(c) we observe that the information gain for our PM-based technique is 12% and 17% more than D-S and DDF methods respectively.

6 Conclusions

In this paper, we have described a sensor information aggregation technique for object classification with a multi-agent prediction market and developed a payment function used by the market maker to incentivize truthful revelation by each agent. Currently, the rewards given by the market maker agent to the sensor agents are additional side payments incurred by the decision maker. In the future we plan to investigate a payment function that can achieve budget balance. We are also interested in integrating our decision making problem with the problem of scheduling robots(sensors), and, incorporating the costs to the overall system into the decision-making costs. Another direction we plan to investigate in the future is a problem of minimizing the time to detect an object in addition to the accuracy of detection. Lastly, we plan to incorporate our aggregation technique into experiments with more number of sensors and also with real robots.

References

1. Chen Y., Dimitrov S., Sami R., Reeves D.M., Pennock D.M., Hanson R.D., Fortnow L., Gonen R. Gaming Prediction Markets: Equilibrium Strategies with a Market Maker. *Algorithmica J.*, 58(4):930–969, 2009.

2. Chen Y., Kash I. Information Elicitation for Decision Making. In *AAMAS 2011*, pages 175–182, 2011.
3. Cremer F., Schutte K., Schavemaker J.G.M, den Breejen E. A comparison of decision-level sensor-fusion methods for anti-personnel landmine detection. *Information Fusion*, 2(1):187–208, 2001.
4. Gros B., Bruschini C. Sensor technologies for the detection of antipersonnel mines: a survey of current research and system developments. In *Proc. of the International Symposium on Measurement and Control in Robotics*, pages 509–518, 1996.
5. Hanson R. Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation. *Journal of Prediction Markets*, 1(1):3–15, 2007.
6. Jumadinova J., Dasgupta P. Multi-sensor Information Processing using Prediction Market-based Belief Aggregation. arXiv:1201.2207 [cs.MA].
7. Jumadinova J., Dasgupta P. A Multi-Agent System for Analyzing the Effect of Information on Prediction Markets. *International Journal of Intelligent Systems*, 26(1):383–409, 2011.
8. Makarenko A., Durrant-Whyte H. Decentralized Bayesian algorithms for active sensor networks. In *Information Fusion 7*, pages 418–433, 2006.
9. Manyika J., Durrant-Whyte H. *Data fusion and sensor management.* Prentice Hall, 1995.
10. Michiardi P., Molva R. Simulation-based Analysis of Security Exposures in Mobile Ad Hoc Networks. In *Mobile Ad Hoc Networks, Eur. Wireless Conf.*, 2002.
11. Milisavljeviȅ N., Bloch I. Sensor Fusion in Anti-Personnel Mine Detection Using a Two-Level Belief Function Model, Part C: Applications and Reviews. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(2):269–283, 2003.
12. Milisavljeviȅ N., Bloch I., Acheroy M. Characterization of Mine Detection Sensors in Terms of Belief Functions and their Fusion, First Results. In *Proc. of the Third International Conference on Information Fusion*, pages 15–22, 2000.
13. Mukherjee P., Sen S. Comparing Reputation Schemes for Detecting Malicious Nodes in Sensor Networks. *The Computer Journal*, 54(3):482–489, 2011.
14. Osborne M., Roberts S., Rogers A., Ramchurn S., Jennings N. Towards realt-time information processing of sensor network data using computationally efficient multi-output gaussian processes. In *IPSN*, pages 109–120, 2008.
15. Othman A., Sandholm T. Decision Rules and Decision Markets. In *AAMAS 2010*, pages 625–632, 2010.
16. Rogers A. and David E. and Jennings N.R. Selfish Sensors in Wireless Micro-Sensor Networks. In *Proceedings of AAMAS 2004*.
17. Rosencrantz M., Gordon G., Thrun S. Decentralized sensor fusion with distributed particle filters. In *IPSN*, pages 55–62, 2003.
18. Srinivasan A., Teitelbaum J., Wu J., Cardei M., and Liang H. *Reputation-and-Trust-Based Systems for Ad Hoc Networks, in Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks.* Wiley, 2008.
19. Stone P., Kaminka G., and Rosenschein J. Leading a Best-Response Teammate in an Ad Hoc Teams. In *Proceedings of AMEC 2009*, pages 132–146, 2009.
20. Vinyals M., Rodriguez-Aguilar J., Cerquides J. A Survey of Sensor Networks from a Multiagent Perspective. *Computer Journal*, 54(3):455–470, 2011.
21. Waltz E., Llinas J. *Multisensor Datafusion.* Artech House, 1990.
22. Wolfers J. and Zitzewitz E. Prediction Markets. *Journal of Econ. Perspectives*, 18(2):107–126, 2004.
23. Wu Q., Rao N., Barhen J., Iyengar S., Vaishnavi V., Qi H., Chakrabarty K. On Computing Mobile Agent Routes for Data Fusion in Distributed Sensor Networks. *IEEE Trans. on Knowledge and Data Engineering*, 16(6):740–753, 2004.