Multi-Variate Distributed Data Fusion with Expensive Sensor Data

Blind

***Abstract*—Distributed fusion of complex information is critical to the success of large organizations. For such organizations, comprised of thousands of agents, improving and shaping the quality of conclusions reached is a challenging problem. The challenge is increased by the fact that acquisition of information could be costly. This leads to the crucial requirement that the organization should strive to reach correct conclusions while minimizing information acquisition cost. In this paper, we have developed a model of complex, interdependent information that is costly to acquire and where complex fusion can be optimized within an organization by optimizing the accuracy of the conclusions reached, while minimizing the cost of acquiring the sensor data. Our experimental results show a number of interesting effects. First, unselfish agents who spend resources (even when not strictly necessary) can lead to substantial improvement in the overall accuracy of conclusions. Second, an organization can substantially improve its performance by assigning resources to get sensor data appropriately. Third, over time, agents can learn the reliability of the members of the organization to whom they are directly connected to improve performance. Learning can also lead to better team decisions about whether to spend resources and how much resource to expend to get sensor data. Our conclusions and algorithms can help a range of organizations reach better conclusions while expending less resources procuring sensor data.**

I. INTRODUCTION

Members of organizations typically need to know various facts about the environment in which they are operating in order to take actions towards team goals. Domains of interest include commerce, disaster response, military and social networks. Humans or agents will draw conclusions about uncertain facts from both sensor data and information passed on from other members of the organization. Different facts will be important to different members of the organization and will have different importance to the organization. For example, in a disaster response scenario knowing whether a particular street is blocked is relevant to agents interested in using that street, while knowing the scale of the disaster is of interest to the commanders and is generally more important. In many interesting real-world cases, the collection of sensor data has some cost which should be avoided wherever possible or the organization may have some total budget for sensing. For example, collecting data might require sending a robot into the environment with costs of energy, wear and tear, or the cost may simply be the time taken to get the information. This paper examines human or agent organizations collecting expensive sensor data and using that data to reach a range of conclusions important to the team.

Information management in organizations has been extensively studied in the literature, in fields ranging from agents, to control, to economics. Agents researchers have looked closely at *simple* information sharing models, e.g., [3], [1], [16], [11], and developed a range of algorithms for moving around and fusing information efficiently. In the control literature, there has been extensive work looking at dynamics of *simple* models of consensus formation, e.g., [11], [9], [5], [14], [13]. Organization theory has examined very complex real-world organization and tried to work out how information moves around and how organizations can be changed to improve information flow, but data is typically not available to create detailed models of exactly what is happening. In this paper, we use agents with a complex model of interdependent information and expensive sensor data to examine in detail how complex fusion can be optimized within organizations and how the organization can minimize the price it pays for sensor data, while accurately reaching correct conclusions.

We use a model of distributed information fusion where each agent is interested in only one of a set of facts about the environment. The facts are conditionally dependent on each other and only some agents have direct access to sensors, so many must rely on communication from neighbors in the organizational structure to reach their conclusions. We assume that it is infeasible for the organization to share raw sensor data and instead agents must communicate only their conclusions. Despite very noisy data, the team would typically reach correct conclusions about all key facts. The more inter-dependent facts that the team needed to reason about, the better the team was at reaching correct conclusions because wrong information was more effectively ignored.

A rationale agent should not spend more on collecting sensor data than the value of reaching a correct conclusion. A local decision model for choosing to buy sensor data based on reliability of neighbors in the organization, time and the cost of data significantly performed other local models of deciding when sensors should be used. This can result in agents that have low value for their variable of interest also often being wrong about their conclusions, because they are unwilling to buy the data to reach a correct conclusion. In turn, this can lead to the whole team being less reliable at reaching conclusions because their neighbors are providing less reliable information. We investigated several methods for improving the team's overall reliability. By learning each others reliability the team can significantly improve performance. By agents being less *selfish* and paying for sensor even when neighbors are expected to provide sufficient information to lead to a conclusion can significantly improve team performance. By agents learning the value of a correct conclusion to the team, not just the local value and using the value to the team to decide whether there is value to buying sensor the team's performance can be improved. Finally, by allocating a total sensing budget intelligently across the team and allowing agents to spend up to their sensing budget to get data, the team's performance can be improved. Each of these optimizations is applicable to a range of real human or agent organizations.

The process of learning neighbor's reliability was expected to produce some interesting dynamics, since all the agents are simultaneously learning and much previous research has shown complex and often undesirable effects. However, in this case the learning was surprisingly effective, however oscillations occurred over time.

II. Related Work

[17] empirically studies the global cascades under random networks with different interpersonal influences. [8] studies the information sharing and propagation in social networks. He argues that individual's decision most likely depends on social information instead of his own private information and thus causes information cascades. [10] builds a stochastic model for spreading of rumors. They use mean-field analysis equation to describe the dynamics of the model for complex social network and show that large complex networks has critical threshold in the rumor spreading rate. [2] proposes an distributed gossip algorithm for agents in peer to peer network to share information. There are other works on the similar topic, such as [15] which studies the epidemic spreading, and [4] which studies the spreading of forest fire model. Glinton *et al.* [7] and [6] have studied the emergent behavior of the dynamics of large scale networks. Their model is based on a network of small number of sensors and a team of agents who share information about a single fact. They have shown that system behaves optimally in a very small range of system parameters. When the parameters deviate from that range, the system performance degrades dramatically. They have shown that this behavior is caused by cascades of belief changes due to a single sensor reading. All these works show the emergent behavior of information cascading, but they have not studied the multidimensional facts or the relation of the dynamic behavior and the correlation of different facts.

Reece *et al.* [12] have provided a multi-dimensional trust model to allow agents to share correlated multi-dimensional contracts. They have developed an approach based on Kalman filter to fuse relevant information from other agents. They have shown that their approach improves significantly over the simple approach based on single dimension of trust.

III. Decision Models

A. Model

We will describe the underlying model in this section. Agents $A = \{a_1, \dots, a_{|A|}\}$ are connected by an undirected network graph $G = (A, E)$ where E is the set of links which connect agents in A. A link connecting agent A and B is represented by $e(A, B)$. Agent can only communicate directly with its neighbors. Agent A's neighbor $N(A)$ is defined by $N(A) = \{B|e(A, B) \in E\}$. Some agents are connected to sensors, $S = \{s_1, \dots, s_{|S|}\}$, which provide noisy observations to the team. Each observation from a sensor is treated independently. Sensors charge for each observation they offer. The charges may differ from sensor to sensor. Only one agent can see directly the output of each sensor. $F = \{f_1, f_2, \dots, f_{|F|}\}$ is the set of facts. Each sensor returns a binary observation $b \in \{true, false\}$ about the fact it is monitoring. In this paper, we assume that the probability of the fact f_i being true is p_i, $1 \leq i \leq |F|$, and the covariance matrix of the distribution functions of those facts is C. We refer to the probability that a sensor s returns a correct observation as its reliability r_s. Each agent is interested in one of the facts. A reward is given if the agent reaches the correct conclusion about the fact it is interested in. The amount of reward may differ from agent to agent. The probability distribution of the facts, the covariance matrix and the reliability of a sensor is known to the agent who receives the observation. In the remainder of this paper, unless otherwise specified, $|A| = 1000$, $|S| = |A|/20$ and $r_s = 0.55 \forall s$. So all agents must deal with noisy data and most do not have direct access to the sensors.

Each sensor monitors the status of only one fact and each agent is interested in only one fact . Each agent forms its belief in the fact of interest by updating its belief based on the information it receives from other agents or sensors. If the information it receives is about a fact it is not interested in, it can still use the information by taking advantage of the correlation between those facts.

We assume that agents only communicate their conclusions to other agents instead of degree of belief. A conclusion about a fact is one of *true*, *false*, and *unknown*. We define a threshold T where $0.5 < T < 1$. When agent's belief is greater than T, the agent's conclusion about the fact is *true*, if the belief is below $1 - T$, the conclusion is *false*, if the belief is between $1 - T$ and T, the conclusion is *unknown*. Agent communicates only when its conclusion changes. Only the last communication from any neighbor is used, which means that the newest conclusion overrides all previous conclusions from the same agent about the fact. Conclusions from different neighbors are treated as independent information. Treating conclusions from neighbors as independent information causes *double counting* which is impossible to remove *double counting* completely without communicating actual sensor data to everybody, In the remainder of this section, we describe how agents make decisions.

B. Belief Update

The first part of the agent's decision is how to form conclusions based on incoming data. We assume a bayesian model and a known covariance matrix of facts. Agents update belief based on information it receives. There are two cases:

- Case 1: The information received is about the fact the agent is interested in
- Case 2: The information received is about a different fact

Sensor data and information from neighbors are treated the same way. Agent uses the same formula to update its belief based on sensor's reliability or neighbor's reliability.

a) Information about the same fact: Suppose the reliability of the information source is c and the prior belief about the fact being true is b. The new belief is b'. By Bayesian rule, when the information says the fact is true, then

$$b' = \frac{bc}{bc + (1-b)(1-c)} \tag{1}$$

When the information says the fact is false, then

$$b' = \frac{b(1-c)}{b(1-c) + (1-b)c} \tag{2}$$

b) Information about a different fact:

Theorem 1: Suppose the reliability of the information source is c, the information is about fact j and the agent is interested in fact i. Let p_i and p_j be the probability of fact i and j being true respectively. The covariance between the two facts is a. The prior belief of the agent about the fact i being true is b. Suppose the new belief is b'. By Bayesian rule, when the information says that fact j is true

$$b' = \frac{p_i(1-p_i)(2cp_j - c - p_j + 1)b + (2c-1)(1-p_i)ab}{p_i(1-p_i)(2cp_j - c - p_j + 1) + (2cb + p_i - b - 2cp_i)a} \tag{3}$$

When the information says that fact j is false, then

$$b' = \frac{p_i(1-p_i)(p_j + c - 2cp_j)b + (1-p_i)(1-2c)ab}{p_i(1-p_i)(p_j + c - 2cp_j) + (2cp_i - p_i - 2cb + b)a} \tag{4}$$

C. Belief Update Operators and Theory

The key for an organization to minimize sensing costs is to know the potential value of performing sensing. In this section, we lay the foundation for this computation by defining the effectiveness of a sensor reading at helping an agent reach a conclusion.

Definition 1: Suppose an agent receives information f, its prior belief about the fact being true is b, its new belief is b'. We define the operator $\oplus$, such that

$$b' = b \oplus f \tag{5}$$

Theorem 2: Suppose an agent is interested in fact i and it receives information f_1 about fact i, the reliability of the its source is c. Suppose information f_2 is about fact j. Let p_i and p_j be the probability of fact i and j being true respectively. The covariance between the two facts is a. Let the reliability of the source of f_2 is $r_{ji}(c)$, then

$$r_{ji}(c) = \frac{p_i(1-p_i)(1 + 2cp_j - c - p_j) + a(1-p_i)(2c-1)}{2p_i(1-p_i)(1 + 2cp_j - c - p_j) + a(2p_i + 2c - 4cp_i - 1)} \tag{6}$$

we have

$$b \oplus f_1 = b \oplus f_2$$

Theorem 2 builds an equivalence relation between information about different facts and information about the same fact. So in order to update belief based on information from a different fact, we can always transfer the reliability of the information source based on formula 6 to a new reliability and update the belief based on the information as if it is about the same fact.

Theorem 3: Suppose information f_1 and f_2 are from the same source, and about same fact. If f_1 and f_2's statements about the state of fact is different, i.e, one says the fact is true and the other says the fact is false, then both information cancels out each other, which is

$$(b \oplus f_1) \oplus f_2 = b \tag{7}$$

Theorem 4: Suppose f_1 and f_2 are two pieces of information, then

$$(b \oplus f_1) \oplus f_2 = (b \oplus f_2) \oplus f_1 \tag{8}$$

Theorem 5: Suppose f_1 and f_2 are two pieces of information about the same fact, the reliabilities of their sources are c_1 and c_2, then if we define a new information f about the same fact, source reliability $c = \frac{c_1 c_2}{2c_1c_2 - c_1 - c_2 + 1}$, then we have

$$(b \oplus f_1) \oplus f_2 = b \oplus f \tag{9}$$

Definition 2: f_1 , f_2 and f are defined in Theorem 5, we define

$$f_1 \oplus f_2 = f \tag{10}$$

From theorem 4, 5 and definition 2, we have

$$(b \oplus f_1) \oplus f_2 = b \oplus (f_1 \oplus f_2) = b \oplus f_1 \oplus f_2 \tag{11}$$

Definition 3: Suppose the agent cares about fact i and information f is about the same fact i, the reliability of the information source is c, the threshold is T, we define the effectiveness of the information to be

$$e(f) = e(c) = \frac{\ln(\frac{1}{c} - 1)}{\ln(\frac{1}{T} - 1)} \tag{12}$$

If the information is about a different fact j, we define

$$e(f) = e(r_{ji}(c)) = \frac{\ln(\frac{1}{r_{ji}(c)} - 1)}{\ln(\frac{1}{T} - 1)} \tag{13}$$

where $e_{ji}(c)$ is defined in equation 6

Theorem 6: Let f_1 and f_2 be two pieces of information about the same information source, we have

$$e(f_1 \oplus f_2) = e(f_1) + e(f_2) \tag{14}$$

Information with effectiveness e is called e piece of effective information. Suppose a piece of information is about the same fact as the agent, and the reliability of the source is T. If it says that the fact is true, then from Equation 1, we have $0.5 \oplus f = 0.8$, if it says that the fact is false, then from 2, we have $0.5 \oplus f = 0.2$. Since $e(f) = 1.0$, so an agent needs a total 1.0 information effectiveness in order to reach from unknown state to "true" or total -1.0 information effectiveness in order to reach a "false" state.

Definition 4: Suppose the agent's belief about the fact is b, the threshold is T, we define the effectiveness of the belief by

$$e(b) = \frac{\ln(\frac{1}{b} - 1)}{\ln(\frac{1}{T} - 1)} \tag{15}$$

So an agent with belief b needs total information effectiveness of $1 - e(b)$ in order to reach a true state and effectiveness of $-e(b) - 1$ to reach a false state. In other words, the agent needs $1 - e(b)$ piece of effective information in order to reach a conclusion about the fact.

D. Resource Allocation

Agents need to decide locally if they should pay for sensor readings. Agents decisions should based on available budget, price of sensors, their belief states and their value to other agents. When the total budget or resource is limited, the resources should be carefully assigned to those agents with the sensors, since only agents with sensors can use resources. The naive assignment is to allocate resource evenly to all agents near sensors. But in most cases, it is not an efficient method. Instead we need to allocate resources to more important agents. So we need to define weight for each agent, which represents how important an agent is. We first define the value of the conclusion from one agent to the other agent.

Definition 5: Let agent a be interested in fact i, its reliability is c, the probability that a will reach a conclusion is p. Its neighbor b is interested in fact j. The probability that b will reach a conclusion is q. Let the value of agent b to the whole team be w, Then we define v_{ab}, the value of agent a to agent b by

$$v_{ab} = e(r_{ij}(c))w$$

Since agent a's conclusion is $e(r_{ij}(c))$ piece of effective information to agent b, and agent b needs 1 piece of effective information to reach a conclusion from an unknown state, so the value of agent a's correct conclusion worth $e(r_{ij}(c))w$ to agent b.

If we treat weight the same as value, then since agent's value is its own reward plus its value to its neighbors.

So weight should satisfy the following condition. Suppose agent a is interested in fact i, the weight is w, reward is r, the reliability is c and its neighbors are $b_1, \cdots, b_k$, which are interested in facts $i_1, \cdots, i_k$, respectively. Suppose their weights are $w_1, \cdots, w_k$, then the effectiveness of a correct conclusion of agent a to b_j is $e(r_{ii_m}(c))$, so the value of the correct conclusion is $e(r_{ii_m}(c))w_m$. But its neighbor can't always take advantage of the conclusion, since sometimes, they are not able to reach a conclusion, and the information is wasted. So we need to introduce a damping factor $0 < d < 1$.

$$w = r + \sum_{m=1}^{k} de(r_{ii_m}(c))w_m \tag{16}$$

We can calculate weights of all agents iteratively until they converge to a fixed value.

Algorithm 1 Calculating weights

$accurate = false$
while $accurate = false$ **do**
  $accurate = true$
  for each location agent A **do**
    old = weight of A
    $weightofA = A'sreward$
    for each neighbors N of A **do**
      calculate A's value V, to N
      $incrementA'sweightbyV$
    end for
    if $|A'sweight - old| > 0.001$ **then**
      $accurate = false$
    end if
  end for
end while

If we assigned total resource purely according to weights and sensor costs, it might be the case that none of the agents gets enough resources to buy sensor readings to reach a conclusion. Agent needs certain amount of resources in order to buy enough sensor readings. So we need to satisfy the need of the most important agent first, then second, etc. If there are still resources left, they can be allocated to them according to their importance. Importance is defined by the weight divided by cost of sensor reading of the agent. The need for agents can be calculated by 17

$$\frac{1}{2c-1}\frac{\ln(\frac{1}{T} - 1)}{\ln(\frac{1}{c} - 1)}x \tag{17}$$

where c is the sensor reliability and x is the cost. Since each reading provide $\frac{\ln(\frac{1}{c}-1)}{\ln(\frac{1}{T}-1)}$ piece of effective information, so the agent needs $\frac{\ln(\frac{1}{T}-1)}{\ln(\frac{1}{c}-1)}$ many correct readings in order to reach a conclusion. Since the sensor provides correct readings with probability c, so on average, the agent needs to buy $\frac{1}{2c-1}\frac{\ln(\frac{1}{T}-1)}{\ln(\frac{1}{c}-1)}$ many readings.

Algorithm 2 Allocate resources

INPUT: totoal resource T
calculate weights of agents
order agents with sensors by $\frac{weight}{sensorcost}$ in decreasing order
for each agent A in the ordered set **do**
  calculate A's need , N_A
  if $N_A <= T$ **then**
    callocate N_A to A
  else
    allocate T to A
  end if
end for
if $T > 0$ **then**
  allocate T proportionally to $\frac{weight}{sensorcost}$
end if

E. Agent Decision

With the definitions in hand, we move onto showing the correct action for the agent to take. An agent near the sensor needs to decide if they should buy the sensor readings or not. The decision depends on the value of the agent (or weight), its belief, sensor cost, its budget, and time.

c) Selfish and unselfish: Agent needs to decide if the sensor reading is worth the cost. It can calculate the value based on its own reward or on its value to the whole team. Selfish agents only care about their own reward, unselfish agents consider their value to the whole team.

d) Dependency on other agents: Agent can estimate how much information it can receive from its neighbors. If it has not reached a conclusion, amd if there will be information from its neighbors enough for it to reach a conclusion, it does not need to buy sensor readings.

e) Time sensitive: Agents may also take time into consideration. When the time is approaching to the end, agent might depend less on the neighbors. Suppose h is the average help it can expect from its neighbors in the whole game, then at time t, the help it should expect from its neighbors should be $h(t)$ where

$$h(t) = dh(1 - \frac{t}{t_0}) \tag{18}$$

where t_0 is the total time.

f) Expected help from neighbors: Agent can calculate the expected help from its neighbors. If it uses part of the expected help, we call the agent partially dependent on others. When all agents depend on their neighbors, it can happen that all agents wait for their neighbors and nobody wants to buy the sensor readings. So we define the dependency factor $0 \leq d \leq 1$. When $d = 0$, agents do not depend on their neighbors when making a decision, when $d = 1$, agents depend fully on their neighbors when making a decision.

Suppose agent a is interested in fact i, the weight is w, reward is r, the reliability is c and its neighbors are $b_1, \cdots, b_k$, which are interested in facts $i_1, \cdots, i_k$, respectively. Suppose their weights are $w_1, \cdots, w_k$,and reliability $c_{i_1}, \cdots, c_{i_k}$, the probabilities that they reach conclusions are $p_{i_1}, \cdots, p_{i_k}$ then the expected help from the neighbors is

$$h = \sum_{m=1}^{k} e(r_{i_m,i}(c_{i_m}))p_{i_m}(2c_{i_m} - 1) \tag{19}$$

g) Value of sensor readings: The value of the sensor reading based on agent own reward is

$$v = \frac{\ln(\frac{1}{c} - 1)}{\ln(\frac{1}{T} - 1)} \frac{y}{1 - e(b) - hd(1 - \delta\frac{t}{t_0})} \tag{20}$$

Where $e(b)$ is defined in 15, h is defined in 19, and δ represents if the agent considers time when it makes a decision. $\delta = 1$ if the agent is considering time, and $\delta = 0$ if not. If agent is using reward when making a decision, then replace y by agent reward, if the agent uses its value to the whole team, replace it by its weight.

Algorithm 3 Agent decision

INPUT: sensor cost c, reliability r_s, agent's belief in the fact b
$e(b) = \frac{\ln(\frac{1}{b}-1)}{\ln(\frac{1}{T}-1)}$
agent's help $h = 0$
for each neighbor N **do**
  $c = N'sreliability$
  $p = N'sprobabilityinreachingaconclusion$
  calculate neighbor's equivalent reliability c' as if the neighbor is interested in the same fact
  $e(c') = \frac{\ln(\frac{1}{c'}-1)}{\ln(\frac{1}{T}-1)}$
  $h = h + e(c')p(2c - 1)$
end for
if the agent is selfish **then**
  $y = agent'sreward$
else
  $y = agent'sweight$
end if
if the agent is time sensitive **then**
  $\delta = 1$
else
  $\delta = 0$
end if
$e(r) = \frac{\ln(\frac{1}{r}-1)}{\ln(\frac{1}{T}-1)}$
$d = agent'sdependency$
$t_0 = total\ time, t = current\ time$
$v = e(r)\frac{y}{1-e(b)-dh(1-\delta\frac{t}{t_0})}$
if $v <= cost$ **then**
  return true
end if
return false

IV. Results

A. Experiment Setup

By default, there are 1000 agents, 100 different facts and 50 sensors in the network. We study three different networks, random, grid, and scale free. Average number of degree of each node is 4. The cost per observation is between 0 and 1, which is randomly assigned to each sensor. The reliability of sensor is set at 0.55. Agent rewards is normally distributed with mean 1.0 and variation 0.2. We run the experiment for 500 cycles, each cycle has 1000 steps. In the beginning of each cycle, resources are allocated to those agents near sensors. In the end of each cycle, each agent get a reward if it has the correct conclusion about the fact it is interested in. Each agent also update its trust in all its neighbors at the end of the cycle. At each step, each sensor provides a reading with probability 0.2, which means it provides a reading every 5 steps. Agent near the sensor will decide if it needs to buy the sensor based on its decision policy. Then it will freely pass on the information to all its neighbors. Each agent updates its belief whenever new information is received. If it changes its conclusion about the fact it is interested in, it broadcast that conclusion to all its neighbors. We study the impact of different techniques and parameters on the system performance. Which include resource allocation, agent decision, selfishness of agents, dependency of agents on other agents, correlation between facts, network types, number of agents, number of sensors, and number of facts. In this paper, we use fund and resource interchangeably.

B. Resource Allocation and Total Resource

When the total budget is tight, it is critical to assign the resource to the right agents. We vary the total fund from 0 to 1000. We compare the simple even allocation and our weighted allocation. As in Figure 1, weighted allocation

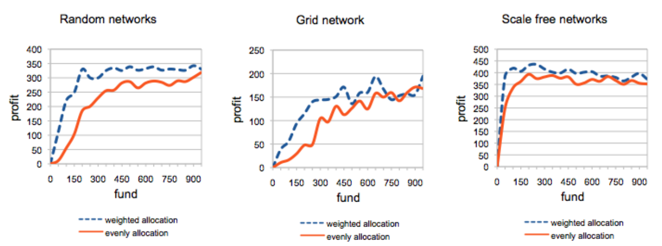

Fig. 1. Resource allocation, agents use unselfish and time sensitive decision, dependency 0.5.

method outperforms the even allocation method dramatically when total fund is relatively low, when the total fund increases to a certain level, they perform about the same, that's because, when total fund is high, even allocation will assign enough funds to important agents too.

Increase total fund will increase system performance when it is low. When the total fund reaches a certain level, it does not increase the system performance. It is because agents will not spend the fund when it reaches a conclusion or it thinks the sensor reading is not worth the money. So when it has what it needs, it does not help to have more. For random networks, the system performance remains about the same after total fund reaches 200, for grid network, it is 650 and 100 for scale free network. For scale free networks, there is a sharp increase in system performance, when total fund increase from 0 to 50. (Explain)

C. Agent Decision

We run the experiment under different kinds of decision policies, selfish and time insensitive, selfish and time sensitive, unselfish and time insensitive , unselfish and time insensitive. We set the dependency to be 0.5. Weighted allocation is used in all cases. Total fund varies from 0 to 900.

h) Selfish and Unselfish: As in Figure 2, As for random and grid networks, unselfish, time sensitive agents performs much better than selfish, time sensitive agents, which in turn outperforms time insensitive agents. For scale free networks, when agents are time sensitive, selfish and unselfish agents perform about the same. But time sensitive agents performs much better than time insensitive agents.

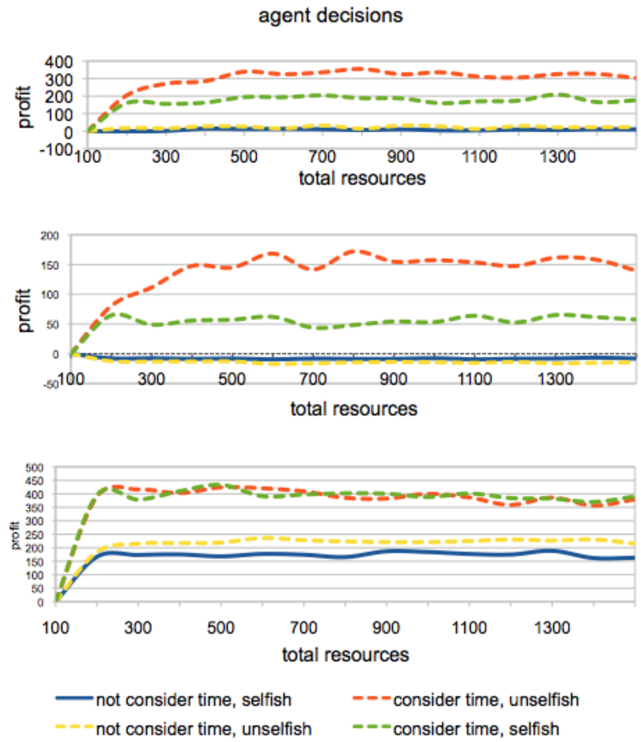

Fig. 2. Comparison of agent decisions.

i) Time sensitive decision: as in Figure 2. For all three kinds of networks, time sensitivity is more important than selfishness.

j) Dependency on other agents: We fix the agent decision to be unselfish and time sensitive. We vary dependency from 0 to 1.0. as in Figure 3, we found that when dependency is low, system performs poor, but once the dependency reaches a certain level, the system perform remains about the same with some fluctuations.

D. Variable Correlation

In this experiment, we compare different level of correlations between facts. Strong correlation, low correlation and no correlation. Weighted allocation is used in all cases, dependency is set at 0.5, unselfish and time sensitive decision policy is used. We vary the total fund from 0 to 1000. As in Figure 4, we see that system performance with strong correlation among facts is significantly higher than that with low correlation, which is significantly higher than that with no correlation.

E. System Parameters

In order to study the impact of system parameters on the performance, we run experiments under different number of agents, sensors and facts. Weighted allocation is used in all cases. Dependency is set at 0.5. Unselfish and time sensitive decision is used. Total fund is 300.

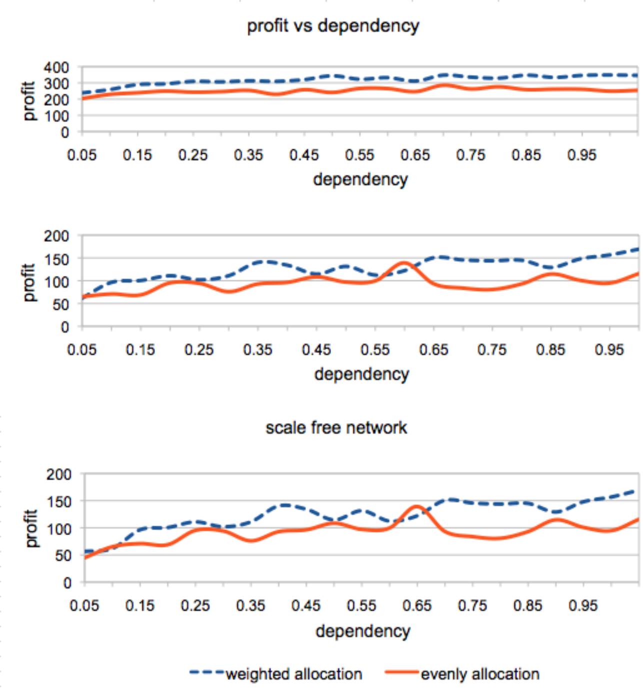

Fig. 3. system performance vs dependency

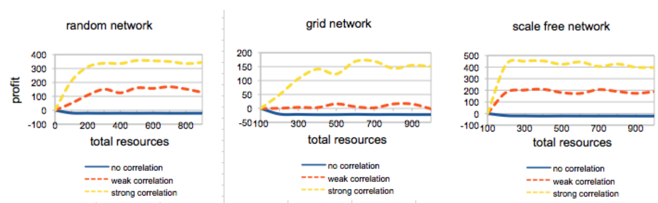

Fig. 4. system performance vs correlation in facts

k) Number of facts: As in Figure 5,, the system performance fluctuate when the number of variable varies. The performance in scale free network is the most stable, random network less stable , and the performance fluctuates the most in grid network.

l) Number of sensors: As in Figure 6, system performs better with more sensors, when the number reaches a level, increasing the number of sensors does not increase system performance. Random and Gird networks perform poorly when the number of sensors is low, but scale free network performs well with low number of sensors, this is because scale free network has few agents with very high degrees and avalanche happens with small amount of readings, which leads to better system performance, [6].

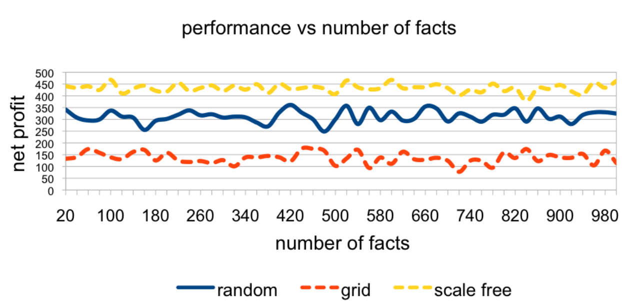

Fig. 5. system performance vs number of facts

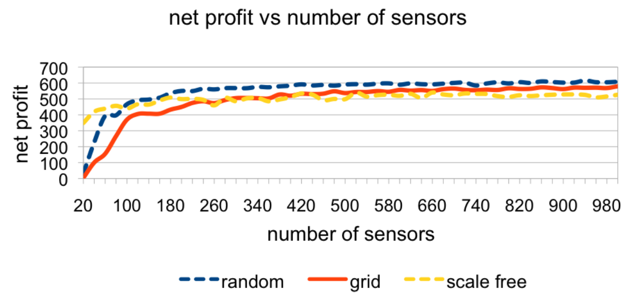

Fig. 6. system performance vs number of sensors

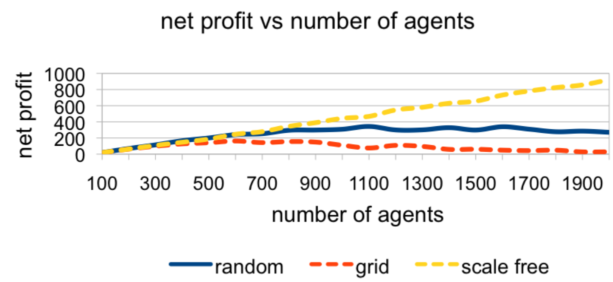

Fig. 7. system performance vs number of agemts

m) Number of agents: As in As in Figure 7, for random networks, the system performance increases with more agents, after it reaches a certain level, it begin to decrease slowly. For grid networks, it increases first, then decreases to zero. This is because the total fund is limited, with more agents, fewer agents get what they need. It is getting worse in grid network. As in scale free network, the performance increases almost linearly when the number of agents increases. This is because there are few agents with high degrees, small amount information can lead to system wide avalanches, increasing the size of network, can lead to overall profit.

n) Agent learning: Glinton *et al.* [7] and [6] have studied the emergent behavior of the dynamics of large scale networks. Their model is based on a network of small number of sensors and a team of agents who share information about a single fact. They have shown that system behaves optimally when c_p is in a very small interval. When c_p deviates from that range, the system performance degrades dramatically. c_p is a fixed parameter representing agents reliability.It needs to be tuned to increase system performance. We think c_p should be based on past experience and should differ from agents to agents. So we conduct an experiment in which agents learn c_p of their neighbors based on their past performance. There are 500 cycles in our experiment, each cycle has 1000 steps. At the end of each cycle, agents learn neighbors reliability. We divide 500 cycles into 50 steps, 10 cycles each step. We measure the average performance in each step. We set agent reliability at 0.5, As in Figre 8, agents learn quickly and perform nearly optimal after 10 steps. Random and scale free networks learn much faster than grid network due to their scale invariant dynamics, [6].

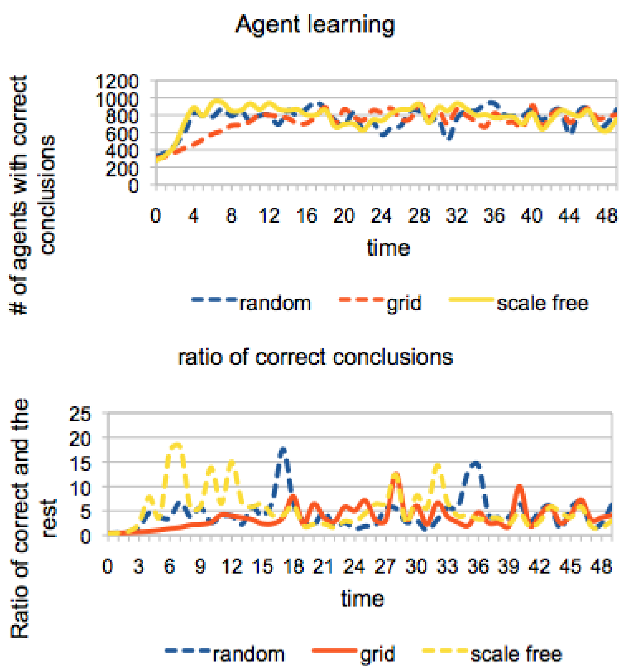

Fig. 8. Performance vs time

V. Conclusions

In this paper we considered the question of how large organizations can reach complex conclusions, while minimizing the price they pay for sensor data. We showed that if agents learned their importance to the overall organization and took this into account when deciding whether to pay for sensor data, the performance of the whole organization improved, although some oscillations in performance could occur. Moreover, we showed an intelligent way to allocate an organization's sensing budget across the organization so as to get the most accurate conclusions, despite agents only considering their local sensing budget.

While we believe that this work represents an important first step for improving the information processing of organizations that rely on expensive sensing to reach conclusions, there is much more work to be done. Most critically, the work presented here assumes the world, the organization and the reward for reaching correct conclusions about specific facts is constant over time. This is not true for interesting organizations. Key challenges to extending this work to more realistic organizations include being able to reuse learned information as the organizational structure changes and working out when to take additional sensor readings when the world may have changed.

References

[1] K. Suzanne Barber and Joonoo Kim. Belief revision process based on trust: Agents evaluating reputation of information sources. In *Proceedings of the workshop on Deception, Fraud, and Trust in Agent Societies held during the Autonomous Agents Conference*, pages 73–82, London, UK, 2001. Springer-Verlag.

[2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *Information Theory, IEEE Transactions on*, 52(6):2508–2530, 2006.

[3] Marco Carbone, Mogens Nielsen, and Vladimiro Sassone:. Formal model for trust in dynamic networks. In *Proceedings of 1st International Conference on Software Engineering and Formal Methods (SEFM 2003),*, pages 54–63, Brisbane, Australia, 2003. IEEE Computer Society.

[4] S. Clar, B. Drossel, and F. Schwabl. Scaling laws and simulation results for the self-organized critical forest-fire model. *Physical Review E*, 50(2):1009–1018, 1994.

[5] Karen Fullam and K. Suzanne Barber. Dynamically learning sources of trust information: Experience vs. reputation. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1062–1069, Columbia, SC, 2007. IFAAMAS.

[6] R. Glinton, P. Scerri, and K. Sycara. Exploiting scale invariant dynamics for efficient information propagation in large teams. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 21–30. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[7] R.T. Glinton, P. Scerri, and K. Sycara. Towards the understanding of information dynamics in large scale networked systems. In *Information Fusion, 2009. FUSION'09. 12th International Conference on*, pages 794–801. IEEE, 2009.

[8] D. Hirshleifer. The Blind Leading the Blind: Social Influence, Fads, and Informational Cascades. *University of California at Los Angeles, Anderson Graduate School of Management*, 1993.

[9] Bernardo A. Huberman and Fang Wu. The dynamics of reputations. *Journal of Statistical Mechanics*, 2004.

[10] M. Nekovee, Y. Moreno, G. Bianconi, and M. Marsili. Theory of rumour spreading in complex social networks. *Physica A: Statistical Mechanics and its Applications*, 374(1):457–470, 2007.

[11] Sharon Paradesi, Prashant Doshi, and Sonu Swaika. Integrating behavioral trust in web service compositions. In *Proceedings of the 7th International Conference on Web Services (ICWS)*, Los Angles, 2009.

[12] S. Reece, S. Roberts, A. Rogers, and N.R. Jennings. A multi-dimensional trust model for heterogeneous contract observations. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 22, page 128. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

[13] Wei Ren and Randal W. Beard. Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 49, May 2004.

[14] Reza Olfati Saber and Richard M. Murray. Consensus protocols for networks of dynamic agents. In *Proceedings of the American Control Conference*, 2003.

[15] R. Pastor Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev. Lett*, 86(14):3200–3203, 2001.

[16] Philippe Smets. The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:447–458, 1990.

[17] D.J. Watts. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(9):5766, 2002.