INCENTIVIZING INFORMATION SHARING IN NETWORKS

Jie Xu*, *Yangbo Song*†, *Mihaela van der Schaar**

*Department of Electrical Engineering, University of California, Los Angeles
†Department of Economics, University of California, Los Angeles

ABSTRACT

For many networks (e.g. opinion consensus, cooperative estimation, distributed learning and adaptation etc.) to proliferate and efficiently operate, the participating agents need to collaborate with each other by repeatedly sharing information which is often costly while brings no direct immediate benefit for the agents. In this paper, we develop a systematic framework for designing *distributed* rating protocols aimed at incentivizing the strategic agents to collaborate with each other by sharing information. The proposed incentive protocols exploit the ongoing nature of the agents' interactions to assign ratings and through them, determine future rewards and punishments through social reciprocation. Unlike existing rating protocols, the proposed protocol operates in a distributed manner, and takes into consideration the underlying interconnectivity of agents as well as their heterogeneity. We prove that in many deployment scenarios adopting the proposed rating protocols achieves full efficiency (i.e. price of anarchy is one) even with strategic agents.

Index Terms— Information sharing, repeated games, distributed rating protocol.

1. INTRODUCTION

In recent years, extensive research efforts have been devoted to studying cooperative networks where agents interact with each other over a topology repeatedly, by sharing information such as measurements, beliefs, or opinions, in order to solve important tasks in an efficient and distributed manner such as target tracking, resource allocation, learning, and estimation [1]-[3]. However, in many scenarios, participating in the cooperative process entails costs to the agents, such as the cost of transmitting and sharing information with their neighbors. For networks where agents are strategic, meaning that they aim to maximize their own utilities by strategically choosing their actions, the agents will choose to participate in the collaborative process only if they believe this action is beneficial to their current and long-term interests. A distinct feature of the network under consideration is that agents' incentives can be coupled in a possibly extremely complex way due to the underlying topology. Thus, a key challenge to ensure the survivability and efficient operation of networks in the presence of selfish agents is the design of incentive schemes that adapt to the network topology and encourage the agents' cooperation in accordance with the network objective.

We propose to resolve the above incentive problem by exploiting the repeated interactions among agents to enable social reciprocation, by deploying a *distributed* rating protocol. Such rating protocols are *designed* and *implemented* in a distributed manner and are tailored to the underlying topologies. The rating protocol, via the (non-strategic) software clients through which agents are interacting, recommends (in a distributed way) to each agent the levels of information it should share with it neighbors depending on the neighbor's current rating as well as the network topology. We refer to this recommendation as the *recommended strategy*. Agents have the freedom to follow or deviate from this strategy and hence, an important goal is to design the protocol to be *incentive-compatible*, meaning that agents have incentives to follow it. The agent's rating is then increased/decreased by the the software client based on its current rating, and whether it has followed/deviated from the recommended strategy. We refer to this as the *rating update* rule. High/Low-rated agents will be rewarded/punished – the protocol recommends more/less information sharing by their neighbors and hence they receive more/less benefit in the future.

We highlight two distinct features of the problem under consideration and the resulting key challenges for designing rating protocols for agents to cooperate. The first feature is that agents interact over an underlying topology and hence, agents' incentives are coupled in a complex manner. This is in stark contrast with existing works which assume that the agents are randomly matched [10][11][12]. Secondly, the considered networks are informationally decentralized, in the sense that communication can only occur between neighboring agents (and software clients) and there is not a central entity that knows everything. Decentralization prevents rating protocols proposed in prior works [11][12] from being applicable since they are designed and implemented in a centralized manner.

Various incentive schemes were proposed in literature to encourage cooperation among agents (see e.g. [5] for a review of different game theoretic solutions). Pricing schemes [6][7] often require complex accounting and monitoring infrastructures, which introduce substantial communication and computation overhead and hence, they are not suitable for the considered networks. Differential service provided by the network operator is not suitable either since such a centralized network operator does not exist. Alternatively, differential services can also be provided by the other agents participating in the network since agents in the considered applications derive their utilities from their interactions with other agents [8]-[12]. Among them, direct (personal) reciprocation schemes (e.g. the widely adopted Tit-for-Tat strategy [8]) only work when two interacting agents have common interests. In social reciprocation schemes [9]-[12], individual agents obtain some (public) information about other individuals (e.g. their ratings) and decide their behavior toward other agents based on this information. These schemes are often studied using the familiar framework of repeated games. In [9], the information sharing game is studied in a narrower context of cooperative spectrum sensing where the agents' knowledge of the network is complete and symmetric but the investigated various simple strategies will fail to work for arbitrary topologies where agents have incomplete and asymmetric knowledge of the entire network. Rating/reputation mechanisms are proposed as another promising solution to implement social reciprocation. Many of the existing works

Financial support is provided by NSF under grant grant 0830556.

on reputation mechanism are concerned with practical implementation details [13] or determining the impact of reputation on a seller's prices and sales [14]. The few works providing theoretical results on rating protocol design consider either one (or a few) long-lived agent(s) interacting with many short-lived agents [15][16] or anonymous, homogeneous and unconnected agents selected to interact with each other using random matching [10][11][12].

The rest of this paper is organized as follows. Section 2 builds the system model and formulates the protocol design problem. Section 3 unravels the structure of the rating protocol under consideration, design the optimal rating protocol to maximize the social welfare. Section 4 provides numerical results. Finally, we conclude this paper in Section 5. All proofs in this paper can be found in [19].

2. SYSTEM MODEL

We consider a network of N agents, indexed by $\{1, 2, ..., N\} = \mathcal{N}$. Agents are connected subject to an underlying topology $G = \{g_{ij}\}_{i,j\in\mathcal{N}}$ with $g_{ij} = g_{ji} = 1$ representing agent i and j being connected (e.g. there is a communication channel between them) and $g_{ij} = g_{ji} = 0$ otherwise. We set $g_{ii} = 0$. We assume a fixed topology G in this paper but certain types of time-varying topologies can also be allowed in our framework. Time is discrete with infinite horizon. In each time period, each agent i chooses an information sharing action with respect to each of its neighbors j, denoted by $a_{ij} \in [0, 1]$. For example, action can represent the information sharing effort. We collect notations and write $\boldsymbol{a}_i = \{a_{ij}\}_{j:g_{ij}=1}$. Denote $\boldsymbol{a} = (\boldsymbol{a}_1, ..., \boldsymbol{a}_N)$ as the action profile of all agents and $\boldsymbol{a}_{-i} = (\boldsymbol{a}_1, ..., \boldsymbol{a}_{i-1}, \boldsymbol{a}_{i+1}, ..., \boldsymbol{a}_N)$ as the action profile of agents except i. Let $\mathcal{A}_i = [0, 1]^{d_i}$ be the action space of agent i where $d_i = \sum_j g_{ij}$ and $\mathcal{A} = \times_{i\in\mathcal{N}}\mathcal{A}_i$ be the action space of all agents. Agents obtain benefits from neighbors' sharing actions, i.e. $\hat{\boldsymbol{a}}_i$=$\{a_{ji}\}_{j:g_{ij}=1}$ for agent i. Let $b_i(\hat{\boldsymbol{a}}_i)$ denote the benefit [1]. Sharing information is costly and the cost $c_i(\boldsymbol{a}_i)$ depends on an agent i's own actions $\boldsymbol{a}_i$. Hence, given the action profile $\boldsymbol{a}$ of all agents, the utility of agent i is $u_i(\boldsymbol{a}) = b_i(\hat{\boldsymbol{a}}_i) - c_i(\boldsymbol{a}_i)$.

Assumption: For each i, (1) $b_i(\hat{\boldsymbol{a}}_i)$ is non-decreasing in each $a_{ij}, \forall j : g_{ij} = 1$ and is concave in $\hat{\boldsymbol{a}}_i$; (2) the cost is linear in the sum of actions, i.e. $c_i(\boldsymbol{a}_i) = \|\boldsymbol{a}_i\|_1 = \sum_{j:g_{ij}} a_{ij}$.

The above assumption states that (1) agents receive decreasing marginal benefits of information acquisition, which captures the fact that agents become more or less "satiated" when they possess sufficient information; (2) the cost incurred by an agent is equal to the sum of effort of collaboration with all its neighbors.

2.1. Obedient Agents - Benchmark

It is useful to first study how obedient agents (i.e. non-strategic agents who follow any prescribed strategy) interact in order to obtain a better understanding of the interactions and the achievable performance. The objective is to maximize the social welfare of the network, which is defined as the time-average sum utility of all agents, i.e. $V = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{\infty} \sum_i u_i(\boldsymbol{a}(t))$ where $\boldsymbol{a}(t)$ is the action profile in period t. If agents are obedient, then the system designer can assign socially optimal actions, denoted by $\boldsymbol{a}^{opt}(t), \forall t$, to agents and then agents will simply take these actions. Determining the socially optimal actions involves solving the following problem:

$$\begin{aligned} &\underset{a}{\text{maximize}} && V \\ &\text{subject to} && a_{ij}(t) \in [0, 1], \forall i, j : g_{ij} = 1, \forall t \end{aligned} \tag{1}$$

[1] In principle, an agent can obtain benefits from the information sharing over indirect links relayed by its neighbor. In this case, the action will also include the relaying action.

which can be easily solved: any action profile $\boldsymbol{a}^{opt}$ that satisfies $\hat{\boldsymbol{a}}_i^{opt}(t) \in \arg\max_{\hat{\boldsymbol{a}}} b_i(\hat{\boldsymbol{a}}_i(t)) - \|\hat{\boldsymbol{a}}_i(t)\|$ is its solution. We denote the optimal social welfare by V^{opt}. The structure of problem (1) lends itself to a fully decentralized implementation: each software client can compute the optimal actions $\hat{\boldsymbol{a}}_i^{opt}$ for its neighbors and send the solution to their neighboring software clients. Obedient agents take the actions solved by the software clients and hence, the social welfare is maximized.

2.2. Strategic Agents

When agents are strategic, we formally define the network information sharing game below.

Definition 1: A (one-shot) *network information sharing (NIS) game* is a tuple $\mathcal{G} = \langle \mathcal{N}, \mathcal{A}, \{u_i(\cdot)\}_{i\in\mathcal{N}}; G\rangle$ where $\mathcal{N}$ is the set of players, $\mathcal{A}$ is the action space of all players, $u_i(\cdot)$ is the utility function of player i and G is the underlying topology.

Theorem 1. *There exists a unique Nash equilibrium (NE) $\boldsymbol{a}^{NE} = \mathbf{0}$ in the (one-shot) NIS game.*

In the repeated game, the (one-shot) NIS game is played in every period $t = 0, 1, 2, ...$ Let $y_i^t \in Y$ be the public monitoring signal related to agent i's actions $\boldsymbol{a}_i(t)$ at time t. A public history of length t is a sequence of public signals $(y^0, y^1, ..., y^{t-1}) \in Y^t$. We note that in the considered network setting, public signals are "locally public" in the sense that agents only observe the public signals within their own neighborhood but not all public signals. We write $\mathcal{H}_\text{i}(t)$ for the set of local public histories for agent i of length t and $\mathcal{H}_\text{i} = \bigcup_{t=0}^{\infty} \mathcal{H}_\text{i}(t)$ for the set of all public histories of all finite lengths. A (local) public strategy of agent i is $\boldsymbol{\sigma}_i : \mathcal{H}_\text{i} \to \mathcal{A}_i$. We write $\boldsymbol{\sigma}$ as the collection of public strategies for all agents. Since interactions are on-going, each agent i cares about its long-term utility which is defined as $U_i(t) = u_i(\boldsymbol{a}(t)) + \delta u_i(\boldsymbol{a}(t+1)) + \delta^2 u_i(\boldsymbol{a}(t+2)) + ...$ where $\delta \in (0, 1)$ is the discount factor of agents. Perfect Public Equilibrium [17] is used as our solution concept for the repeated game.

Definition 1: (Perfect Public Equilibrium) A strategy profile $\boldsymbol{\sigma}$ is a perfect public equilibrium if $\forall h \in \mathcal{H}, \forall i$, $U_i(\sigma_i, \sigma_{-i}; h) \geq U_i(\sigma_i', \sigma_{-i}; h), \forall \sigma_i' \neq \sigma_i$.

3. DISTRIBUTED OPTIMAL PROTOCOL DESIGN

3.1. Proposed Rating Protocol

We describe the proposed distributed rating protocol and its operation in a distributed network. Keep in mind that software clients are non-strategic and design and implement the rating protocol but the agents *are strategic* in choosing the information sharing actions (i.e. they will selfishly decide whether or not to follow the strategy recommended by the software) such that their own utility is maximized. A rating protocol consists of three components – a set of ratings, a set of recommended strategies to agents, and a rating update rule.

(1) We consider a finite set of ratings $\Theta = \{1, 2, ..., K\}$. Agent i's rating in period t is $\theta_i(t) \in \Theta$ and its neighbors' ratings are $\hat{\boldsymbol{\theta}}_i = \{\theta_j\}_{j:g_{ij}=1}$. K serves as an upper bound of the rating set size.

(2) The software clients determine the recommended (public) strategy profile in a distributed manner and recommend actions to their own agent depending on neighbors' ratings $\boldsymbol{\sigma} : \mathcal{N} \times \mathcal{N} \times \Theta \to [0, 1]$, where $\sigma_{ij}(\theta_j)$ represents the recommended action of agent i with respect to j if j's rating is θ_j. Since high/low-rated agents should be rewarded/punished, $\sigma_{ij}(\theta) \leq \sigma_{ij}(\theta')$ if $\theta < \theta'$. We collect notations $\boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i) = \{\sigma_{ij}(\theta_j)\}_{j:g_{ij}=1}$ and $\hat{\boldsymbol{\sigma}}_i(\theta_i) = \{\sigma_{ji}(\theta_i)\}_{j:g_{ij}=1}$.

(3) The software of agent i updates agent i's rating at the end of each period. Let $y_i \in Y = [0,1]$ be the monitoring signal with respect to agent i's action. Specifically, $y_i = 1\{\boldsymbol{a}_i(t) = \boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i(t))\}$. The rating update rule is therefore a mapping $\tau : \mathcal{N} \times \Theta \times Y \to \Delta(\Theta)$, where $\tau_i(\theta_i^+; \theta_i, y_i)$ is the probability that the updated rating is θ_i^+ if agent i's current rating is θ_i and the public signal is y_i. In particular, we consider the following parameterized rating update rule, for agent i, if $\theta_i = k$,

$$\tau_i(\theta_i^+; \theta_i, y) = \begin{cases} \alpha_{i,k}, & \text{if } \theta_i^+ = \max\{1, k-1\}, y_i = 0 \\ 1 - \alpha_{i,k}, & \text{if } \theta_i^+ = k, y_i = 0 \\ \beta_{i,k}, & \text{if } \theta_i^+ = \min\{K, k+1\}, y_i = 1 \\ 1 - \beta_{i,k}, & \text{if } \theta_i^+ = k, y_i = 1 \end{cases} \tag{2}$$

In words, compliant agents are rewarded by receiving a higher rating with some probability while deviating agents are punished by receiving a lower rating with some (other) probability.

To sum up, the rating protocol is uniquely determined by the recommended strategies $\boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i), \forall i, \forall \hat{\boldsymbol{\theta}}_i$ and the rating update probabilities $\alpha_{i,k}, \beta_{i,k}, \forall i, \forall k$. We denote the rating protocol by $\pi = (\Theta, \boldsymbol{\sigma}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ and the achievable social welfare by adopting the rating protocol by $V(\pi)$. The rating protocol design problem thus is

$$\begin{array}{ll} \underset{\pi=(\Theta,\boldsymbol{\sigma},\boldsymbol{\alpha},\boldsymbol{\beta})}{\text{maximize}} & V(\pi) \\ \text{subject to} & \boldsymbol{\sigma} \text{ constitutes a PPE} \end{array} \tag{3}$$

The operation of the rating protocol comprises two phases: the design phase and the implementation phase. In the design phase, the software clients determine in a distributed way the recommended strategy and rating update rules according to the network topology, and the agents do nothing. In the implementation phase (run-time), the agents (freely and selfishly) choose their actions in each period in order to maximize their own utility. Depending on whether the agents are following or deviating from the recommended strategy, each software executes the rating update of its agent and sends the new ratings of its agent to the neighboring software clients.

3.2. Distributed Computation of the Recommended Strategy

If a rating protocol constitutes a PPE, then all agents will find it in their self-interests to follow the recommended strategie and eventually all agents will have the highest ratings forever (assuming no update errors). Therefore, the social welfare is asymptotically the same as the sum of utility of all agents when they have the highest ratings and follow the recommended strategy, i.e. $V = \sum_i (b_i(\hat{\boldsymbol{\sigma}}_i(K)) - \|\boldsymbol{\sigma}_i(\boldsymbol{K})\|)$. This means that the recommended strategies for the highest ratings determine the social welfare that can be achieved by the rating protocol. However, in the presence of strategic agents, these strategies, together with the other components of a rating protocol, need to satisfy the equilibrium constraint such that self-interested agents have incentives to follow the recommended strategies. In Theorem 2, we identify a sufficient and necessary condition on $\boldsymbol{\sigma}(\boldsymbol{K})$ (i.e. the recommended strategies when agents have the highest ratings) such that an equilibrium rating protocol can be constructed. With this, the software clients are able to determine the optimal rating protocol in a distributed way in order to maximize the social welfare.

Theorem 2. *Given the rating protocol structure and the network structure (topology and individual utility functions), there exists at least one PPE (of the rating protocol) if and only if* $\delta b_i(\hat{\boldsymbol{\sigma}}_i(K)) \geq c_i(\boldsymbol{\sigma}_i(\boldsymbol{K})), \forall i$.

The optimal values of $\boldsymbol{\sigma}(\boldsymbol{K})$ can be determined by solving the following *optimal recommended strategy design (ORSD)* problem:

$$\begin{array}{ll} \underset{\boldsymbol{\sigma}}{\text{maximize}} & \sum_i (b_i(\hat{\boldsymbol{\sigma}}_i(K)) - c_i(\boldsymbol{\sigma}_i(\boldsymbol{K}))) \\ \text{subject to} & c_i(\boldsymbol{\sigma}_i(\boldsymbol{K})) \leq \delta b_i(\hat{\boldsymbol{\sigma}}_i(K)), \forall i \end{array} \tag{4}$$

Note that this problem implicitly depends on the network topology since both $\hat{\boldsymbol{\sigma}}_i(K)$ and $\boldsymbol{\sigma}_i(\boldsymbol{K}), \forall i$ are topology-dependent (since for each agent i, the strategy is only with respect to its neighbors). In this subsection, we will write $\boldsymbol{\sigma}_i(\boldsymbol{K})$ as $\boldsymbol{\sigma}_i$ and $\hat{\boldsymbol{\sigma}}_i(K)$ as $\hat{\boldsymbol{\sigma}}_i$ to keep the notations simple.

Algorithm: Distributed Computation of the Recommended Strategy (DCRS) (Run by each software client of agent i)

Input: Connectivity and utility function of agent i.
Output: $\boldsymbol{\sigma}_i(\boldsymbol{K}) = \{\sigma_{ij}(K)\}_{j:g_{ij}=1}$

Initialization: $q = 0$; $\lambda_i(q) = 0$
Repeat:
Send $\lambda_i(q)$ to neighbor $j, \forall j : g_{ij} = 1$. (Obtain $\lambda_j(q)$ from j)
Solve (6) using $\lambda_i(q)$, $\{\lambda_j(q)\}_{j:g_{ij}=1}$ to obtain $\hat{\boldsymbol{\sigma}}_i(\boldsymbol{\lambda}(q))$.
Send $\sigma_{ji}(\boldsymbol{\lambda}(q))$ to neighbor $j, \forall j : g_{ij} = 1$.
(Obtain $\sigma_{ij}(\boldsymbol{\lambda}(q))$ from j)
Update $\lambda_i(q+1)$ according to (8).
Stop until $\|\lambda_{ji}(q+1) - \lambda_{ji}(q)\|_2 < \varepsilon_\lambda$

Since there is no central entity that knows the entire topology, we propose a distributed algorithm to compute these recommended strategies. The ORSD problem is decomposed into N sub-problems, each of which is solved locally by the software clients. Note that unlike the case with obedient agents, these sub-problems have coupled constraints. Therefore, software clients will need to go through an iterative process to exchange messages (the Lagrangians) with their neighboring software clients such that their local solutions converge to the global optimal solution. We perform dual decomposition on (4) and relax the constraints as follows:

$$\underset{\boldsymbol{\sigma}}{\text{maximize}} \quad \sum_i (b_i(\hat{\boldsymbol{\sigma}}_i) - \|\boldsymbol{\sigma}_i\|) - \sum_i \lambda_i(\|\boldsymbol{\sigma}_i\| - \delta b_i(\hat{\boldsymbol{\sigma}}_i)) \tag{5}$$

where $\lambda_i \geq 0, \forall i$ are the Lagrangian multipliers. The optimization thus separates into two levels of optimization. At the lower level, we have the sub-problems (one for each software client), $\forall i$

$$\underset{\hat{\boldsymbol{\sigma}}_i}{\text{maximize}} \quad (1 + \lambda_i \delta) b_i(\hat{\boldsymbol{\sigma}}_i) - \sum_{j:g_{ij}=1} (1 + \lambda_j) \sigma_{ji} \tag{6}$$

It is easy to see that the optimal solution to these subproblems is also the optimal solution to the relaxed problem (5). At the higher level, the master dual problem is in charge of updating the dual variables,

$$\begin{array}{ll} \underset{\boldsymbol{\lambda}}{\text{minimize}} & g(\boldsymbol{\lambda}) = \sum_i g_i(\boldsymbol{\lambda}) \\ \text{subject to} & \lambda_i \geq 0, \forall i \end{array} \tag{7}$$

where $g_i(\boldsymbol{\lambda})$ is the maximum value of the Lagrangian (6) given $\boldsymbol{\lambda}$ and $g(\boldsymbol{\lambda})$ is the maximum value of the Lagrangian (5) of the primal problem. The following subgradient method is used to update $\boldsymbol{\lambda}$,

$$\lambda_i(q+1) = [\lambda_i(q) + w(\|\boldsymbol{\sigma}_i\| - \delta b_i(\hat{\boldsymbol{\sigma}}_i))]^+, \forall i \tag{8}$$

where q is the iteration index, $w > 0$ is a sufficiently small positive step-size. Because (4) is a convex optimization problem, such an iterative algorithm will converge [18] to the dual optimal $\boldsymbol{\lambda}^*$ as $q \to \infty$ and the primal variable $\boldsymbol{\sigma}^*(\boldsymbol{\lambda}(q))$ will also converge to the primal optimal $\boldsymbol{\sigma}^*$. This iterative process can be made fully distributed which requires only limited message exchange between neighboring software clients. We present the Distributed Computation of the Recommended Strategy (DCRS) Algorithm below which is run locally by each software of the agents.

3.3. Computing the Remaining Components

There are many possible rating protocols that can constitute PPE given the obtained recommended strategies. In fact, the constructional proof of Theorem 2 has already provided one way to compute these remaining components. In this subsection, we provide the optimal design given a binary rating set $\Theta = \{1, 2\}$ when the rating update is subject to small monitoring errors. We set $\sigma_{ij}(\theta = 1) = 0, \forall i, j : g_{ij} = 1$.

Proposition 1. *Given a binary rating protocol* $\Theta = \{1, 2\}$, $\sigma_{ij}(2), \forall i, j : g_{ij} = 1$ *determined by the DCRS Algorithm and* $\sigma_{ij}(1) = 0, \forall i, j : g_{ij} = 1$, *when the monitoring error* $\epsilon > 0$, *the optimal rating update probability that maximize the social welfare is,* $\forall i$, $\beta^*_{i,1} = 1, \alpha^*_{i,2} = \frac{\|\boldsymbol{\sigma}_i(\mathbf{2})\|}{\delta b_i(\hat{\boldsymbol{\sigma}}_i(2))}$

It is worth noting that these probabilities can be computed locally by the software of the agents which do not require any information from other agents.

3.4. Price of Anarchy

We denote the social welfare that can be achieved by the optimal rating protocol as V^* and use *the price of anarchy* (PoA), defined as $PoA = V^{opt}/V^*$, as the performance measure of the rating protocol. Observe the social welfare maximization problems (1) and (4) for obedient agents and strategic agents (by using rating protocols), respectively. It is clear that $V^* \leq V^{opt}$ due to the equilibrium constraint; hence, i.e. $PoA \geq 1$. The exact value of PoA will, in general, depend on the specific network structure. In this subsection, we identify a sufficient condition for the connectivity degree of the topology such that PoA is one. To simplify the analysis, we assume $b_i(\hat{\boldsymbol{a}}_i) = b(\sum_{j:g_{ij}=1} a_{ji})$. Let $d_i = \sum_j g_{ij}$ be the number of neighbors of agent i. The degree of network G is defined as $d = \max_i d_i$.

Proposition 2. *Suppose* $b_i(\hat{\boldsymbol{a}}_i) = b(\sum_{j:g_{ij}=1} a_{ji}), \forall i$ *and* a_{ij} *is upper-bounded, then there exists a* $\bar{d}$ *such that if the connectivity degree* d *is no larger than* $\bar{d}$, *then* $V^* = V^{opt}$, *i.e. PoA is one.*

4. ILLUSTRATIVE RESULTS

In this section, we provide simulation results to illustrate the performance of the rating protocol. In all simulations, agents' utility function takes the form of $u_i(\boldsymbol{a}(t)) = [r^2 - MSE_i(\hat{\boldsymbol{a}}_i(t))] - \boldsymbol{a}_i(t)$ which is an approximate utility for the network cooperative estimation problem [3].

First we investigate how agents' connectivity shapes agents' incentives and influences the resulting social welfare. In Figure 1, we investigate star topologies with different sizes (hence, different connectivity degrees). As predicted by Proposition 2, when the connectivity degree is small enough, the PoA equals one and hence, the performance gap is zero. As the network size increases, the socially optimal action requires the center agent to share more information with the periphery agents but it becomes more and more difficult for the center agent to have incentives to do so. Hence, the rating protocol recommends less information sharing from the center agent to each periphery agent. However, incentives are provided at a cost of reduced social welfare. In the next simulation, we study scale-free networks in the imperfect monitoring scenarios. In scale-free networks, the number of neighboring agents is distributed as a power law (denote the power law parameter by d^{SF}). Table 1 shows the PoA achieved by the rating protocol for various values of d^{SF} and different monitoring error probabilities ϵ. As we can see, the proposed rating protocol achieves close-to-optimal social welfare in all the simulated environments. Finally, Figure 2 illustrates the PoA achieved by the proposed rating protocol and the Tit-for-Tat strategy. The rating protocol yields at least as much social welfare as the Tit-for-Tat strategy. As the discount factor becomes smaller, agents' incentives to cooperate become less and hence, the PoA is larger.

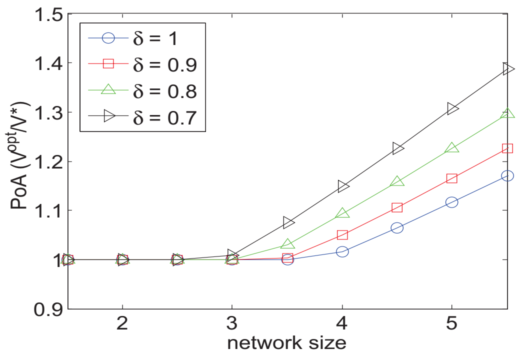

Fig. 1. Performance of the rating protocol for various connectivity degrees in star topologies.($r^2 = 8$)

	d^{SF}	2	3	4	5	6	7	8
PoA	$\epsilon = 0$	1.03	1.03	1.00	1.01	1.01	1.01	1.00
	$\epsilon = 0.05$	1.04	1.04	1.02	1.02	1.03	1.02	1.01
	$\epsilon = 0.1$	1.05	1.05	1.03	1.03	1.04	1.03	1.02

Table 1. Performance of the rating protocol for various in scale-free topologies.

5. CONCLUSION

In this paper, we studied how to design distributed incentives protocols (based on ratings) aimed at maximizing the social welfare of repeated information sharing among strategic agents in social networks. The proposed design framework enables an efficient way to implement social reciprocity in distributed information sharing networks with arbitrary topologies and achieve much higher social welfare than existing incentive mechanisms. The proposed rating protocols can be adopted in a wide range of applications where selfish behavior arises due to cost-benefit considerations including problems involving interactions over social networks, communication networks, power networks, transportation networks, and computer networks.

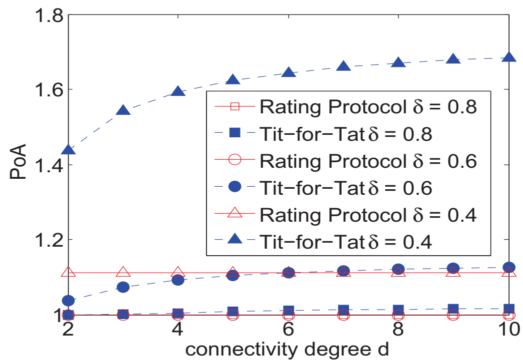

Fig. 2. Performance comparison with Tit-for-Tat. ($r^2 = 4$)

6. REFERENCES

[1] V. Krishnamurthy and H. V. Poor, “Social learning and Bayesian games in multiagent signal processing: How do local and global decision makers interact?” IEEE Signal Process. Mag., vol. 30, no. 3, pp. 43-57, 2013.

[2] D. Acemouglu and A. Ozdaglar, “Opinion dynamics and learning in social networks,” LIDS Report 2851, inaugural issue of Dynamic Games and Applications, vol. 1, no. 1, pp. 3- 49, 2010.

[3] A. H. Sayed, S. –Y. Tu, J. Chen, X. Zhao, and Z. Towfic, “Diffusion strategies for adaptation and learning over networks,” IEEE Sig. Process. Mag., vol. 30, no. 3, pp. 155-171, May 2013.

[4] R. Lucky, “Tragedy of the Commons,” IEEE Spectrum, Jan 2006.

[5] J. Park and M. van der Schaar, “A game theoretic analysis of incentives in content production and sharing over peer-to-peer networks,” IEEE J. Sel. Topics Signal Process., vol. 4, no. 4, pp. 704-717, Aug. 2010.

[6] D. Bergemann, D. Ozman, “Optimal pricing with recommender systems,” in 7^{th} ACM Conf. on Electric Commerce, pp. 43-51, 2006.

[7] J. K. MacKie-Mason, H. R. Varian, “Pricing congestible network resources,” IEEE J. Sel. Areas Commun., vol. 13, no. 7, pp. 1141-1149, 1995.

[8] R. Axelrod, “The emergence of cooperation among egoists,” The Amer. Pol. Sci. Rev., vol. 75, no.2, 1981.

[9] C. Song and Q. Zhang, “Achieving cooperative spectrum sensing in wireless cognitive radio networks,” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 13, no. 2, pp. 14-25, 2009.

[10] M. Kandori, “Social norms and community enforcement,” Rev. of Econ. Stud., vol. 59, pp. 63-80, 1992.

[11] Y. Zhang, J. Park and M. van der Schaar, “Rating protocols for online communities,” ACM Transactions on Economics and Computation, 2013.

[12] J. Xu and M. van der Schaar, “Social norm design for information exchange systems with limited observations,” IEEE J. Sel. Areas Commun., vol. 30, no. 11, pp. 2126-2135, 2012.

[13] S. D. Kamvar, M. T. Schlosser and H. Garcia-Molina, “The eigentrust algorithm for reputation management in P2P networks,” in Proc. 12^{th} international conference on World Wide Web, pp. 640-651, 2003.

[14] P. Resnick and R. Zeckhaagent, “Trust among strangers in internet transactions: empirical analysis of eBay’s reputation system,” Advances in applied microeconomics, pp. 127-157, 2002.

[15] C. Dellarocas, “Reputation mechanism design in online trading environments with pure moral hazard,” Information Systems Research, vol. 16, no. 2, pp. 209-230, 2005.

[16] G. Zacharia, A. Moukas and P. Maes, “Collaborative reputation mechanism in electronic marketplaces,” Decision Support Systems, vol. 29, no. 4, pp. 371-388, 2000.

[17] G. J. Mailath, L. Samuelson, Repeated games and reputations: long-run relationships. Oxford Univ. Press. 2006.

[18] S. P. Boyd, L. Vandenberghe. Convex Optimization. Cambridge university press, 2004.

[19] Online Appendix available at http://www.seas.ucla.edu/~jiex/documents/icassp_infoshare.pdf