# ACTIVE ONLINE LEARNING OF TRUSTS IN SOCIAL NETWORKS

*Hoi-To Wai*[†], *Anna Scaglione*[†], *Amir Leshem*[‡]

[†]School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA.
[‡]Faculty of Engineering, Bar-Ilan University, Ramat-Gan, Israel.

Emails: htwai@asu.edu, Anna.Scaglione@asu.edu, leshema@eng.biu.ac.il

## ABSTRACT

This paper considers an online optimization algorithm for actively learning trusts on social networks. We first introduce a DeGroot model for opinion dynamics under the influence of stubborn agents and demonstrate how an observer with estimates of the individuals opinions can actively learn the relative trusts among different agents, by fitting the opinions to the steady state equations of the social system equations. The main contribution of this article is an online algorithm for extracting the trust parameters from streaming data of randomly sampled, noisy opinion estimates. The algorithm is based on the stochastic proximal gradient method and it is proven to converge almost surely. Finally, numerical results are presented to corroborate our findings.

***Index Terms***— social networks, online optimization algorithm, system identification, active learning

## 1. INTRODUCTION

Opinions in social networks are forged through the mutual *trust* that characterize the interactions of individuals with their own circle of friends. This, through opinion diffusion among connected circles of friends, enables individuals to leverage the *wisdom of the crowd*. Hence, learning the weight of trust in social network interactions, enables us to predict some global behavior. For example, studies have shown that several phenomena such as the spread of diseases/information [1], the polarization of opinions [2], etc., can be predicted from the friendship structure and the trusts.

Attempts to model the effects of trusts in social networks have been studied in [3–5]. Specifically, trusts are manifested in the process of *opinion dynamics*, of which social agent's opinion evolves according to the different amount of trusts he/she places on friends. Examples of such models include the DeGroot model [3], the Hegselmann-Krause model [4], etc. These models capture the diffusive nature of opinion dynamics and are similar to the average consensus algorithms for wireless sensor networks [6, 7].

Studies on the effects of having different types of agents in the network have been considered. An interesting one is when there exists stubborn agents whose opinions cannot be swayed away by their neighbors [8–12]. The stubborn agents are known to exist in social networks, e.g., a politician is stubborn due to his/her political views. Importantly, it was shown that the steady-state opinions can be shaped by the stubborn agents and the friendship structure [11].

This work considers the task of learning trusts (and the friendship structure) from observing opinion dynamics. Previous works [12–16] have considered the *passive* methods by merely tracking and recording the opinions. While their models are extensible to learning trusts when the opinion dynamics is non-linear, these methods require perfect knowledge on the frequency of *social interaction*, which may be hard to obtain. On the other hand, this work considers an *active sensing* method — by introducing a number of stubborn agents, we can learn the trusts from the steady-state opinions, which can be obtained *without* knowing the rate of social interaction.

The main contribution of this work is to develop an online algorithm for estimating trusts under the DeGroot model. The opinion data are acquired on-the-fly. We show that the online algorithm converges almost surely to an optimal solution of the active learning problem when the exact expected opinions are given. We also demonstrate that active learning outperforms passive learning.

### 1.1. Relation to Prior Work

This work is based on our previous papers on active sensing of social networks [17, 18], where we primarily focused on the trust identifiability issue. The learning approach we propose in this paper is also related to the study of inference in graphical models, e.g. the popular Ising's model studied in [19, 20]. While we share the same objectives, our algorithm differs from [19, 20] by considering an active learning approach. Lastly, our work utilizes results from the study of stochastic optimization algorithms; see [21, 22].

## 2. LEARNING TRUSTS IN SOCIAL NETWORKS

Consider a social network defined as a weighted, connected graph $G = (V, E, \overline{\boldsymbol{W}})$, where $V = [n] = \{1, ..., n\}$ is the set of agents, $E \subseteq [n] \times [n]$ is the friendship between agents and $\overline{\boldsymbol{W}} \in \mathbb{R}_+^{n\times n}$ is a stochastic matrix describing the amount of trust among the $n$ agents. Moreover, we have $E = \Omega_{\overline{\boldsymbol{W}}} = \{ij : \overline{W}_{ij} > 0\}$ such that the support of $\overline{\boldsymbol{W}}$ corresponds to the edge set/adjacency matrix of $G$.

The agents discuss online about several subjects, leaving a digital trace. In this work we postulate that, by analyzing their actions in a certain interval of time $t$ on a given subject $s$, it is possible to cluster the actions and determine a score (a probability measure) $\boldsymbol{x}(t; s)$ that represents their opinion (favorable or not, or indifferent) on the subject matter. We assume that the *opinion dynamics* in $G$ follows a randomized DeGroot model [3]. Given the initial *opinions*[1] $\boldsymbol{x}(0; s) \in \mathbb{R}^n$, the opinions on the $s$th discussion $\boldsymbol{x}(t; s)$ can be described by the random process:

$$\boldsymbol{x}(t + 1; s) = \boldsymbol{W}(t; s)\boldsymbol{x}(t; s), \tag{1}$$

where $x_i(t; s)$ denotes the opinion of the $i$th agent, $\boldsymbol{W}(t; s)$ is an independently and identically distributed (i.i.d.) random matrix drawn from a distribution such that $\mathbb{E}\{\boldsymbol{W}(t; s)\} = \overline{\boldsymbol{W}}$. Note that $\boldsymbol{W}(t; s)$ is stochastic, non-negative and its support set follows $\Omega_{\boldsymbol{W}(t;s)} \subseteq E$. Essentially, the dynamics (1) says that the opinions are mixed *randomly* between neighboring agents during each interaction $t$. The

This work is supported by NSF CCF-1011811 and ISF 903/13..

[1]Opinions may refer to a probability distribution of the attitude (e.g., like, hate or ignorant) towards a certain discussion. Here, $x(t; s)$ can be a vector.

described model generalizes the *randomized gossip model* considered in a few previous works, e.g., [6, 23].

Our aim is to learn the *trust matrix* $\overline{\boldsymbol{W}}$ by observing the opinion dynamics. Notice that $\overline{\boldsymbol{W}}$ is embedded in the expected opinion dynamics of (1). A *passive* way to sense the social network is by tracking the opinions $\boldsymbol{x}(t;s)$. In fact, one observes a noisy estimate:

$$\hat{\boldsymbol{x}}(t;s) = \boldsymbol{x}(t;s) + \boldsymbol{n}(t;s), \tag{2}$$

where $\boldsymbol{n}(t;s)$ is a zero mean additive noise with bounded variance. As proposed in [15], we can solve a least square regression problem:

$$\min_{\boldsymbol{W}\geq\boldsymbol{0}} \sum_{s=1}^{K} \sum_{t=t_s+1}^{T_s-1} \|\hat{\boldsymbol{x}}(t+1;s) - \boldsymbol{W}\hat{\boldsymbol{x}}(t;s)\|_2^2 \text{ s.t. } \boldsymbol{W}\boldsymbol{1} = \boldsymbol{1}, \tag{3}$$

where $\boldsymbol{W} \geq \boldsymbol{0}$ denotes that $\boldsymbol{W}$ is non-negative and $[t_s, T_s]$ is the sampling interval on the $s$th discussion where $\boldsymbol{y}(t;s)$ is recorded.

An important assumption made in the *passive sensing* of social networks is that one is able to record the evolution of opinions $\hat{\boldsymbol{x}}(t;s)$ for $t = t_s, t_s+1, ..., T_s$. Even knowing exactly the rate of social interaction, given that the opinions are not directly observable but need to be extracted through semantic analysis of the agents actions, the assumption made is clearly impractical. Our solution to this is to employ an *active sensing* method for social networks and to assume that the opinions on average approximately fit the steady state equations of (1).

### 2.1. Active Learning of Trusts — Problem Formulation

Our active network sensing method relies on the use of stubborn agents, whose opinions cannot be swayed away by its neighbors, while they are influencing the opinions of the social network. We assume such stubborn agents are known (e.g. they were actively recruited or detected). Let the set of stubborn agents be $V_s = [n_s]$ and the set of ordinary agents be $V_r = V \setminus V_s$. Denote $E_s \subseteq [n] \times [n_s+1, n]$ as the edge set between $V_r$ and $V_s$ and $E_r \subseteq [n_s+1, n] \times [n_s+1, n]$ as the edge set between agents in $V_r$. The trust matrix $\overline{\boldsymbol{W}}$ (and its realization $\boldsymbol{W}(t;s)$) has the block structure:

$$\overline{\boldsymbol{W}} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \overline{\boldsymbol{B}} & \overline{\boldsymbol{D}} \end{pmatrix}, \; \boldsymbol{W}(t;s) = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{B}(t;s) & \boldsymbol{D}(t;s) \end{pmatrix}, \tag{4}$$

where $\mathbb{E}\{\boldsymbol{D}(t;s)\} = \overline{\boldsymbol{D}}$, $\mathbb{E}\{\boldsymbol{B}(t;s)\} = \overline{\boldsymbol{B}}$, $\Omega_{\boldsymbol{D}(t;s)} \subseteq \Omega_{\overline{\boldsymbol{D}}} = E_r$ and $\Omega_{\boldsymbol{B}(t;s)} \subseteq \Omega_{\overline{\boldsymbol{B}}} = E_s$. Notice that the upper right block of $\overline{\boldsymbol{W}}_a$ is zero since the opinions of stubborn agents cannot be swayed away.

**Assumption 1** The support of $\overline{\boldsymbol{B}}$, $\Omega_{\overline{\boldsymbol{B}}} = \{ij : \overline{B}_{ij} > 0\}$, is known. Each agent in $V_r$ has a non-zero trust on at least one agent in $V_s$.

**Assumption 2** The induced subgraph $G[V_r]$ is connected.

Consequently, the submatrix $\overline{\boldsymbol{D}}$ satisfies $\|\overline{\boldsymbol{D}}\|_2 < 1$. We have that

**Observation 1** [11] Let $\boldsymbol{x}(t;s) \triangleq (\boldsymbol{z}^T(t;s)\ \boldsymbol{y}^T(t;s))^T \in \mathbb{R}^n$, where $\boldsymbol{z}(t;s)$ and $\boldsymbol{y}(t;s)$ are respectively the opinions of stubborn and ordinary agents. As $\boldsymbol{x}(t+1;s) = \boldsymbol{W}(t;s)\boldsymbol{x}(t;s)$, we have:

$$\lim_{t\to\infty} \mathbb{E}\{\boldsymbol{x}(t;s)|\boldsymbol{z}(0;s)\} = (\boldsymbol{I} - \overline{\boldsymbol{D}})^{-1}\overline{\boldsymbol{B}}\boldsymbol{z}(0;s). \tag{5}$$

Notice that the set of all possible steady states without stubborn agents (cf. (1)) spans a subspace of dimension one. In contrast, the steady states' space under the influence of stubborn agents spans a subspace of dimension $n_s$. The expanded dimension of the steady states' space allow us to identify the trust matrix $\overline{\boldsymbol{D}}$ from observing the steady state opinions. In particular, we stack the opinion data across $K$ discussions as:

$$\boldsymbol{Y} \triangleq \Big(\mathbb{E}\{\hat{\boldsymbol{y}}(\infty;1)|\boldsymbol{z}(0;1)\} \cdots \mathbb{E}\{\hat{\boldsymbol{y}}(\infty;K)|\boldsymbol{z}(0;K)\}\Big), \tag{6a}$$

$$\boldsymbol{Z} \triangleq \Big(\mathbb{E}\{\hat{\boldsymbol{z}}(\infty;1)|\boldsymbol{z}(0;1)\} \cdots \mathbb{E}\{\hat{\boldsymbol{z}}(\infty;K)|\boldsymbol{z}(0;K)\}\Big), \tag{6b}$$

where $\hat{\boldsymbol{y}}(t;s) = \boldsymbol{y}(t;s) + \boldsymbol{n}_y(t;s)$ and $\hat{\boldsymbol{z}}(t;s) = \boldsymbol{z}(t;s) + \boldsymbol{n}_z(t;s)$ similar to (2). Leveraging on the equality $\boldsymbol{Y} = (\boldsymbol{I} - \overline{\boldsymbol{D}})^{-1}\overline{\boldsymbol{B}}\boldsymbol{Z}$, we see that the stubborn agents act similarly as a *RADAR* applied on social networks, where one can treat $\boldsymbol{Y}, \boldsymbol{Z}$ as the output/input to the social network, respectively. As proposed in [18], by exploiting the sparseness of $\overline{\boldsymbol{D}}$, i.e., agents typically do not trust from a large number of other agents, the regression problem for active network sensing is given as a LASSO problem: let $\boldsymbol{y}_s$, $\boldsymbol{z}_s$ be the $s$th column of $\boldsymbol{Y}, \boldsymbol{Z}$,

$$\begin{aligned} \min_{\substack{\boldsymbol{B}\geq\boldsymbol{0},\\ \boldsymbol{D}\geq\boldsymbol{0}}} \quad & \sum_{s=1}^{K} \|\boldsymbol{y}_s - \boldsymbol{D}\boldsymbol{y}_s - \boldsymbol{B}\boldsymbol{z}_s\|_2^2 + \lambda\|\boldsymbol{D}\|_1 + \\ & \qquad\qquad \gamma\|\boldsymbol{D}\boldsymbol{1} + \boldsymbol{B}\boldsymbol{1} - \boldsymbol{1}\|_2^2 \\ \text{s.t.} \quad & \mathcal{P}_{\mathcal{S}^c}(\boldsymbol{D}) = \boldsymbol{0},\ \mathcal{P}_{\Omega^c_{\overline{\boldsymbol{B}}}}(\boldsymbol{B}) = \boldsymbol{0},\ \mathrm{diag}(\boldsymbol{D}) = \boldsymbol{c}, \end{aligned} \tag{7}$$

where $\gamma, \lambda \geq 0$, $\mathcal{S}$ is a known superset of $\Omega_{\overline{\boldsymbol{D}}}$, $\mathcal{S}^c$ is its complement and $\boldsymbol{0} \leq \boldsymbol{c} < \boldsymbol{1}$ is a preset vector for fixing the ambiguity issue inherent in (5) (see [17]). Moreover, the operator $\mathcal{P}_\Omega(\cdot)$ is:

$$[\mathcal{P}_\Omega(\boldsymbol{A})]_{ij} = \begin{cases} A_{ij}, \ ij \in \Omega, \\ 0, \ \text{otherwise}, \end{cases} \tag{8}$$

where $\Omega$ is an index set for the elements in $\boldsymbol{A}$. Notice that (7) is a convex optimization problem.

An identifiability condition of $(\overline{\boldsymbol{B}}, \overline{\boldsymbol{D}})$ using a non-convex variant of (7) has been studied in our previous work [18]. In particular, it was shown that the number of stubborn agents $n_s$ required for perfect recovery is proportional to the number of non-zeros in $\overline{\boldsymbol{D}}$. In this paper, our focus is to study an online learning algorithm to solve (7) using opinion samples that are acquired on-the-fly.

## 3. ONLINE LEARNING ALGORITHM OF TRUSTS

As noted by [17], it is possible to estimate $\mathbb{E}\{\hat{\boldsymbol{x}}(\infty;s)|\boldsymbol{z}(0;s)\}$ by evaluating its temporal average, i.e., let us define

$$\hat{\boldsymbol{x}}_s(\mathcal{T}) \triangleq \tfrac{1}{|\mathcal{T}|}\textstyle\sum_{t\in\mathcal{T}} \rho^{t_{max}-t}\hat{\boldsymbol{x}}(t;s), \tag{9}$$

which admits the following partition

$$\hat{\boldsymbol{x}}_s(\mathcal{T}) \triangleq (\hat{\boldsymbol{y}}_s^T(\mathcal{T})\ \ \hat{\boldsymbol{z}}_s^T(\mathcal{T}))^T$$

where $0 < \rho \leq 1$ is a forgetting factor, $t_{max} = \max\{t \in \mathcal{T}\}$ and $\mathcal{T} \subseteq \{t_o, t_o+1, \ldots, \infty)$ is the sampling instances. The following holds [17, Theorem 4.2]:

**Fact 1** Let $\rho = 1$, $t_o = \min_{t\in\mathcal{T}} t$ and $t_o \to \infty$, we have

$$\mathbb{E}\{\|\hat{\boldsymbol{x}}_s(\mathcal{T}) - \mathbb{E}\{\boldsymbol{x}(\infty;s)|\boldsymbol{z}(0;s)\}\|_2^2\} = \mathcal{O}(1/|\mathcal{T}|). \tag{10}$$

In light of Fact 1, to apply the active sensing method for social networks, a naïve approach is to estimate the expected value $\hat{\boldsymbol{Y}}$ and $\hat{\boldsymbol{Z}}$ using (9) by acquiring a sufficient amount of samples and, then, using these estimates to solve the convex optimization in (7).

In practice, collecting the samples $\{\boldsymbol{y}_a(t;s)\}_{t\in\mathcal{T}}$ may take a long time. Using an online learning method one can take advantage

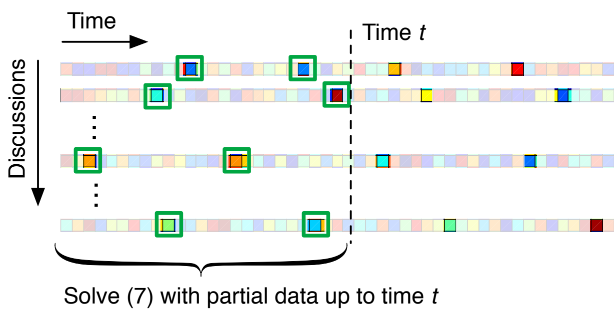


**Fig. 1**: Streaming opinion data collected from different topics. Each colored dot represents the opinions of $n$ agents on a discussion, i.e., $\hat{\boldsymbol{x}}(t;s)$, and the highlighted dots are the non-uniformly sampled opinions. At time $t$, only a finite number of samples are available to the estimator in (9) and the online learning algorithm.

of the partial samples collected up to time $t$ and provide a crude estimate of the trust matrix that continues to improve as more opinion samples are gathered (see Fig. 1).

This motivated us to adapt the *stochastic* proximal gradient (SPG) algorithm studied in [21]. Compared to the conventional proximal gradient algorithm, the difference is that SPG uses a noisy estimate of the gradient at each iteration. More specifically, consider the following optimization problem:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + h(\boldsymbol{x}), \tag{11}$$

where $f$ is convex, $L$-continuously differentiable and $h$ is convex (possibly non-smooth). The SPG algorithm is described by the following recursion. At the $k$th iteration,

$$\boldsymbol{x}^{k+1} = \text{prox}_{\alpha h}(\boldsymbol{x}^k - \alpha \boldsymbol{g}^k), \tag{12}$$

where $\alpha \in (0, 1/L]$ with $L$ being the Lipschitz constant for the gradient of $f$. The proximal operator $\text{prox}_{\alpha h}(\cdot)$ [24] is defined as

$$\text{prox}_{\alpha h}(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}} \; (\alpha h(\boldsymbol{y}) + \|\boldsymbol{x} - \boldsymbol{y}\|_2^2). \tag{13}$$

In the above, $\boldsymbol{g}^k$ is a noisy estimate of the gradient:

$$\boldsymbol{g}^k = \nabla f(\boldsymbol{x}^k) + \boldsymbol{\eta}^k. \tag{14}$$

Using the fact that $f$ has a Lipschitz continuous gradient and (11) is convex, the following was proven in [21, Theorem 6]:

**Theorem 1.** *Consider the SPG algorithm* (12). *If* $\limsup_{k\to\infty} \|\boldsymbol{\eta}^k\| < \infty$ *and* $\lim_{k\to\infty} \boldsymbol{\eta}^k = 0$ *almost surely (a.s.), then* $\lim_{k\to\infty} \boldsymbol{x}^k = \boldsymbol{x}^\star$ *a.s., where* $\boldsymbol{x}^\star$ *is an optimal solution to* (11).

To study the SPG algorithm on (7), we first define:

$$\begin{aligned} f(\boldsymbol{B}, \boldsymbol{D}) &= \sum_{s=1}^{K} \|\boldsymbol{y}_s - \boldsymbol{D}\boldsymbol{y}_s - \boldsymbol{B}\boldsymbol{z}_s\|_2^2 + \gamma \|(\boldsymbol{D}\ \boldsymbol{B})\mathbf{1} - \mathbf{1}\|_2^2, \\ h(\boldsymbol{B}, \boldsymbol{D}) &= \lambda \|\boldsymbol{D}\|_1 + \mathcal{I}_{\mathcal{F}}(\boldsymbol{B}, \boldsymbol{D}), \end{aligned} \tag{15}$$

where $\mathcal{I}_{\mathcal{F}}(\cdot,\cdot)$ is the indicator function for the constraints in (7) such that $\mathcal{I}_{\mathcal{F}}(\boldsymbol{B}, \boldsymbol{D}) = \infty$ if $(\boldsymbol{B}, \boldsymbol{D}) \notin \mathcal{F}$, and is zero otherwise. Both $f$ and $h$ are convex. In addition, $f$ is a continuously differentiable function. Notice that this shows that (7) can fit into the general form of (11). Now, the gradient matrices of $f$ are given by:

$$\begin{aligned} \nabla_{\boldsymbol{D}} f(\boldsymbol{B}, \boldsymbol{D}) &= 2 \textstyle\sum_{s=1}^{K} \left( \boldsymbol{B}\boldsymbol{z}_s \boldsymbol{y}_s^T + (\boldsymbol{D} - \boldsymbol{I})\boldsymbol{y}_s \boldsymbol{y}_s^T \right) \\ &\quad + 2\gamma(\boldsymbol{D}\mathbf{1}\mathbf{1}^T + \boldsymbol{B}\mathbf{1}\mathbf{1}^T - \mathbf{1}\mathbf{1}^T), \\ \nabla_{\boldsymbol{B}} f(\boldsymbol{B}, \boldsymbol{D}) &= 2 \textstyle\sum_{s=1}^{K} \left( \boldsymbol{B}\boldsymbol{z}_s \boldsymbol{z}_s^T + (\boldsymbol{D} - \boldsymbol{I})\boldsymbol{y}_s \boldsymbol{z}_s^T \right) \\ &\quad + 2\gamma(\boldsymbol{D}\mathbf{1}\mathbf{1}^T + \boldsymbol{B}\mathbf{1}\mathbf{1}^T - \mathbf{1}\mathbf{1}^T), \end{aligned} \tag{16}$$

**Algorithm 1** Active online learning of trusts using SPG.

1: **Initialize:** $(\boldsymbol{B}^0, \boldsymbol{D}^0) \in \mathcal{F}$, $k = 1$;
2: **while** *convergence is not reached* **do**
3: Observe new opinion samples $\{\hat{\boldsymbol{x}}(t;s)\}_{t \in \mathcal{T}_{k,s} \setminus \mathcal{T}_{k-1,s}}$ and update the estimators $\hat{\boldsymbol{y}}_s^k, \hat{\boldsymbol{z}}_s^k$ accordingly.
4: Compute the gradient $\boldsymbol{g}\boldsymbol{D}^k, \boldsymbol{g}\boldsymbol{B}^k$ using (17).
5: Perform the proximal gradient updates (cf. (18)):

$$(\boldsymbol{B}^{k+1}, \boldsymbol{D}^{k+1}) \leftarrow \text{prox}_{\alpha h}\big((\boldsymbol{B}^{k+1} - \alpha \boldsymbol{g}\boldsymbol{B}^k, \boldsymbol{D}^{k+1} - \alpha \boldsymbol{g}\boldsymbol{D}^k)\big)$$

6: $k \leftarrow k + 1$.
7: **end while**
8: **Return:** $(\boldsymbol{B}^k, \boldsymbol{D}^k)$.

where the all ones matrices $\mathbf{1}\mathbf{1}^T$ are with compatible dimensions. As mentioned before, obtaining the *exact* expectation $(\boldsymbol{y}_s, \boldsymbol{z}_s)$ requires a large number of samples on $\hat{\boldsymbol{x}}(t;s)$ which may take a long time to collect. Therefore, we replace these gradient matrices by estimates of them evaluated using the data available up to iteration $k$.

Define $\mathcal{T}_{k,s}$ as the set of sampling instances that the samples on $\hat{\boldsymbol{x}}(t;s)$ are collected up to the $k$th iteration, i.e., at iteration $k$, we have observed the samples $\{\hat{\boldsymbol{x}}(t;s)\}_{t \in \mathcal{T}_{k,s}}$ for the $s$th discussion. Naturally, we have $\mathcal{T}_{k-1,s} \subseteq \mathcal{T}_{k,s}$, $\min(\mathcal{T}_{k-1,s}) = \min(\mathcal{T}_{k,s})$ and $\max(\mathcal{T}_{k-1,s}) \leq \max(\mathcal{T}_{k,s})$. Using the shorthand notations $\hat{\boldsymbol{z}}_s^k = \hat{\boldsymbol{z}}_s(\mathcal{T}_{k,s})$ and $\hat{\boldsymbol{y}}_s^k = \hat{\boldsymbol{y}}_s(\mathcal{T}_{k,s})$ (cf. (9)), the following estimate of $\nabla_{\boldsymbol{D}} f(\boldsymbol{B}^k, \boldsymbol{D}^k), \nabla_{\boldsymbol{B}} f(\boldsymbol{B}^k, \boldsymbol{D}^k)$ can be obtained:

$$\begin{aligned} \boldsymbol{g}\boldsymbol{W}^k &= 2 \textstyle\sum_{s=1}^{K} \left( \boldsymbol{B}^k \hat{\boldsymbol{z}}_s^k (\hat{\boldsymbol{y}}_s^k)^T + (\boldsymbol{D}^k - \boldsymbol{I}) \hat{\boldsymbol{y}}_s^k (\hat{\boldsymbol{y}}_s^k)^T \right) \\ &\quad + 2\gamma(\boldsymbol{D}^k\mathbf{1}\mathbf{1}^T + \boldsymbol{B}^k\mathbf{1}\mathbf{1}^T - \mathbf{1}\mathbf{1}^T), \\ \boldsymbol{g}\boldsymbol{B}^k &= 2 \textstyle\sum_{s=1}^{K} \left( \boldsymbol{B}^k \hat{\boldsymbol{z}}_s^k (\hat{\boldsymbol{z}}_s^k)^T + (\boldsymbol{D}^k - \boldsymbol{I}) \hat{\boldsymbol{y}}_s^k (\hat{\boldsymbol{z}}_s^k)^T \right) \\ &\quad + 2\gamma(\boldsymbol{D}^k\mathbf{1}\mathbf{1}^T + \boldsymbol{B}^k\mathbf{1}\mathbf{1}^T - \mathbf{1}\mathbf{1}^T). \end{aligned} \tag{17}$$

We can prove the following:

**Proposition 1** *If* $\rho = 1$, *the estimator* (9) $\hat{\boldsymbol{x}}_s(\mathcal{T}_{k,s})$ *converges to the expected value* $\overline{\boldsymbol{x}}_s \triangleq \mathbb{E}\{\boldsymbol{x}_a(\infty;s)|\boldsymbol{x}_z(0;s)\}$ *a.s. if* $k \to \infty$ *and* $t_o \to \infty$, *where* $t_o = \min_{t \in \mathcal{T}_{k,s}} t$.

The proof can be found in Appendix A. As a consequence, we have $\lim_{k\to\infty} \hat{\boldsymbol{y}}_s^k = \boldsymbol{y}_s$ and $\lim_{k\to\infty} \hat{\boldsymbol{z}}_s^k = \boldsymbol{z}_s$ a.s. and the gradient estimates converges to the true gradient matrices a.s.. As the gradient estimation error is bounded, applying Theorem 1 shows that the SPG algorithm applied to (7) converges a.s..

Lastly, the proximal operator can be computed in closed form. Define $(\tilde{\boldsymbol{B}}, \tilde{\boldsymbol{D}}) = \text{prox}_{\gamma h}(\boldsymbol{B}, \boldsymbol{D})$. We have

$$\tilde{\boldsymbol{B}} = \mathcal{P}_{\Omega_{\frac{c}{\boldsymbol{B}}}}(\boldsymbol{B}), \; \text{off}(\tilde{\boldsymbol{D}}) = \mathcal{P}_{\mathcal{S}^c}(\texttt{soft\_th}_{\gamma\lambda}(\text{off}(\boldsymbol{D}))), \tag{18}$$

and $\text{diag}(\tilde{\boldsymbol{D}}) = \boldsymbol{c}$, where $\text{off}(\cdot)$ denotes the off-diagonal elements in the square matrix, $\texttt{soft\_th}_\lambda(\cdot)$ is a *one-sided* soft thresholding operator [25] that applies element-wisely and $\texttt{soft\_th}_\lambda(x) = u(x)\max\{0, x - \lambda\}$. We summarize the stochastic proximal gradient algorithm for (7) in Algorithm 1.

## 4. NUMERICAL RESULTS & CONCLUSIONS

We consider applying Algorithm 1 to learn the trust matrix in a synthetically generated social network. The graph topology of $G$ is generated as Erdos-Renyi (ER) graph with connectivity $p = 0.05$ with $n = 100$ ordinary agents. Meanwhile, we apply $n_s = 36$ stubborn agents and the subnetwork corresponding to the edge set $E_s$,

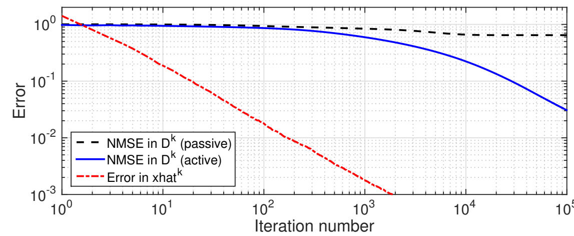


**Fig. 2**: The NMSE of the learning algorithms against the iteration number $k$. The figure also shows the squared error in estimating $\boldsymbol{z}_s, \boldsymbol{y}_s$ as new samples are collected by evaluating $\sum_{s=1}^{K} \|(\hat{\boldsymbol{z}}_s^k, \hat{\boldsymbol{y}}_s^k) - (\boldsymbol{z}_s, \boldsymbol{y}_s)\|_2^2/K$.

i.e., between stubborn agents and ordinary agents, is generated as a $d$-random regular bipartite graph. Each ordinary agent is randomly connected to $d = 5$ stubborn agents such that the stubborn-ordinary agent network forms a random regular graph. The random opinion exchange follows that of a randomized broadcast gossip model in [23] with parameter $\gamma$ set to 0.5. The self-trust vector $\boldsymbol{c}$ in (7) is set to $\boldsymbol{0}$. Moreover, we set $\mathcal{S} = [n] \times [n]$, i.e., the online learning method has no prior knowledge on the support of $\boldsymbol{D}$. The opinion samples are simulated according to model (1) and (2), and we generate $\boldsymbol{n}(t; s)$ as an i.i.d. vector with independent $\mathcal{N}(0, 0.1)$ elements.

We consider the opinion data from $K = 2n_s$ discussions. For each discussion $s$, the sampling instances set $\mathcal{T}_s$ are uniformly sampled from $\{10^3, 10^7\}$ such that $|\mathcal{T}_s| = 5 \times 10^5$. The first numerical example compares the normalized mean square error against the iteration number, i.e., $\text{NMSE} = \|\boldsymbol{D}^k - \overline{\boldsymbol{D}}'\|_F^2 / \|\overline{\boldsymbol{D}}'\|_F^2$ where $\overline{\boldsymbol{D}}'$ is the relative trust matrix defined by normalizing the rows of $\overline{\boldsymbol{D}}$ such that $\text{diag}(\overline{\boldsymbol{D}}') = \boldsymbol{0}$. In particular, the online learning algorithm performs an update to the variable $(\boldsymbol{B}^k, \boldsymbol{D}^k)$ whenever $5K$ new opinion samples are collected for the $K$ discussions. The algorithm parameters are $\rho = 1, \gamma = 0.1, \lambda = 10^{-10}$ and $\alpha = 0.01$.

The numerical results are presented in Fig. 2. As observed, the NMSE of $\boldsymbol{D}^k$ decays as the number of iteration grows. After $10^5$ iterations, the NMSE is decreased to $5 \times 10^{-2}$. Moreover, the squared error in the estimate of $(\boldsymbol{z}_s, \boldsymbol{y}_s)$ decays in the order of $\mathcal{O}(1/k)$, corroborating with Fact 1. The results demonstrates that the trusts can be learnt when opinion samples are collected in an online setting.

To demonstrate the efficacies of the active learning method, the last numerical example compares the estimated trust matrix to that resulted from the *passive* learning method (cf. (3)) for sensing $\overline{\boldsymbol{W}}$ using the same set of data collected. As the rate of social interaction is unknown, the passive method takes the non-uniformly sampled opinion data as input. Moreover, the a-priori knowledge that $\overline{\boldsymbol{W}}$ follows the block structure in (4) and $\mathcal{P}_{\Omega_{\overline{\boldsymbol{B}}}}(\boldsymbol{B}) = \boldsymbol{0}$ are enforced. To handle the large amount of opinion data collected, the constrained least square problem is tackled using a projected gradient method.

We plot the trust matrices learnt at different epochs of the algorithms using the active and passive methods in Fig. 3. Notice that the NMSE of the matrix learnt using the passive method after $10^5$ iterations of projected gradient is 0.6463. As observed, the trust matrix estimated by the passive method contains a large number of links with high squared error, resulting in a dense matrix. It is due to the fact that the opinion data are collected non-uniformly. The active method is able to estimate accurately the trust matrix.

To conclude, we have described an online algorithm for actively learning trusts in social networks. Our approach consists in exploiting the presence of stubborn agents and their impact on the network-wide opinions to identify the relative trusts of the social network, without prior knowledge on the network structure. The online algorithm is developed based on the stochastic proximal gradient method and its convergence is proven. Numerical results show

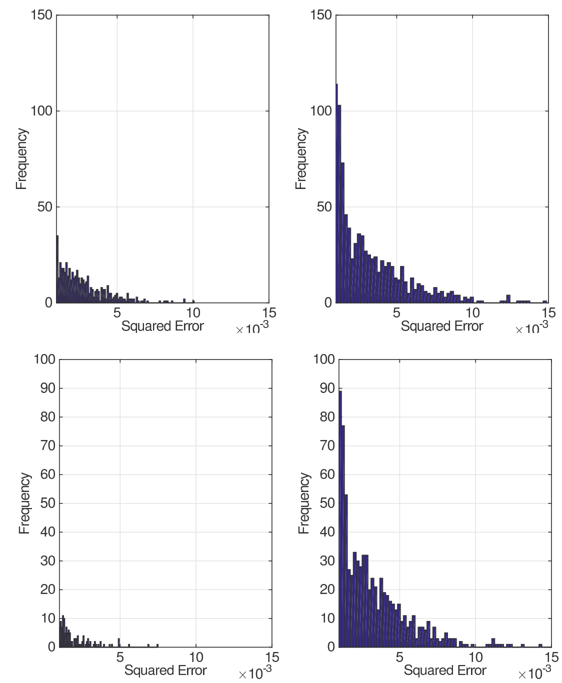


**Fig. 3**: Histogram of the squared error in the trust matrices learnt $(\hat{D}_{ij} - \overline{D}_{ij})^2$ at different epochs. (top) the trust matrix learnt after $5 \times 10^3$ iterations — (left) using online active learning (7), (right) using passive learning (3); (bottom) the trust matrices learnt after $50 \times 10^3$ iterations.

that our method is effective and works better than *passive* methods.

Future works include extending the online learning framework to cases when the social network is dynamical/quasi-static and using real opinion data collected from the public domain.

## A. PROOF OF PROPOSITION 1

The proof is based on a strong law of large number for correlated random variables from [26, p. 28] [27]:

**Theorem 2.** *Consider* $S_k = (1/k) \sum_{i=1}^k X_i$*, where* $X_i$ *are correlated random variables such that* $\sup_{m \geq 1} \frac{\text{Cov}(X_m, X_{m+i})}{\text{Var}(X_m)\text{Var}(X_{m+i})} \leq \rho_i$*, where* $\text{Cov}(\cdot,\cdot)$ *and* $\text{Var}(\cdot)$ *denote the covariance and variance of the random variable(s), respectively. If* $\sum_{i=1}^\infty \rho_i < \infty$*, then* $S_k \rightarrow (1/k) \sum_{i=1}^k \mathbb{E}\{X_i\}$ *a.s..*

We see that $\mathbb{E}\{\hat{\boldsymbol{x}}(t; s) | \boldsymbol{z}(0; s)\} = \overline{\boldsymbol{x}}_s$ for all $t \in \mathcal{T}_{k,s}$ as $t_o \rightarrow \infty$. Thus, it suffices to prove Proposition 1 by checking the covariance between terms in the summation (9). To this end, we assume $|\mathcal{T}_{k,s}| = k$ without loss of generality and $\mathcal{T}_{k,s} = \{t_{k,s}^1, ..., t_{k,s}^k\}$. For $i \geq 1$, the covariance can be bounded as:

$$\begin{aligned} &\text{Tr}(\text{Cov}(\hat{\boldsymbol{x}}(t_{k,s}^m; s), \hat{\boldsymbol{x}}(t_{k,s}^{m+i}; s))) \\ &\leq \mathbb{E}\{(\hat{\boldsymbol{x}}(t_{k,s}^m; s) - \overline{\boldsymbol{x}}_s)^T (\hat{\boldsymbol{x}}(t_{k,s}^{m+i}; s) - \overline{\boldsymbol{x}}_s) | \boldsymbol{z}(0; s)\} \\ &= \mathbb{E}\{(\boldsymbol{x}(t_{k,s}^m; s) - \overline{\boldsymbol{x}}_{a,s})^T (\boldsymbol{x}(t_{k,s}^{m+i}; s) - \overline{\boldsymbol{x}}_s) | \boldsymbol{z}(0; s)\}, \end{aligned} \tag{19}$$

where we have used $\mathbb{E}\{\boldsymbol{n}^T(t_{k,s}^m; s)\boldsymbol{n}^T(t_{k,s}^{m+i}; s)\} = 0$ for all $i \geq 1$ as the noise are independent in (2). Using [17, Eq. (32)-(33)], the latter term can be bounded by $\rho_i = \mathcal{O}(\lambda^{t_{k,s}^{n+i} - t_{k,s}^n})$ where $\lambda = \|\overline{\boldsymbol{D}}\|_2 < 1$. Moreover, it is easy to check that $\mathbb{E}\{\|\hat{\boldsymbol{x}}(t_{k,s}^{n+i}) - \mathbb{E}\{\hat{\boldsymbol{x}}(t_{k,s}^{n+i})\}\|_2^2\} = \Theta(1)$. Since $t_{k,s}^{n+i} - t_{k,s}^n \geq 1$ for all $i \geq 1$, we get $\sum_{i=1}^\infty \rho_i < \infty$ and the conditions in Theorem 2 are satisfied.

### B. REFERENCES


[1] M. E. J. Newman, "Spread of epidemic disease on networks," *Physical Review E*, vol. 66, no. 1, Jul. 2002.

[2] L. Li, A. Scaglione, A. Swami, and Q. Zhao, "Consensus, polarization and clustering of opinions in social networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 1072–1083, 2013.

[3] M. DeGroot, "Reaching a consensus," in *Journal of American Statistcal Association*, vol. 69, 1974, pp. 118–121.

[4] R. Hegselmann and U. Krause, "Opinion dynamics and bounded confidence models, analysis and simulations," in *Journal of Artificial Societies and Social Simulation*, vol. 5, 2002.

[5] R. A. Holley and T. M. Liggett, "Ergodic theorems for weakly interacting infinite systems and the voter model," *The annals of probability*, pp. 643–663, 1975.

[6] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[7] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847 –1864, nov. 2010.

[8] D. Acemoglu, G. Como, F. Fagnani, and A. Ozdaglar, "Opinion Fluctuations and Disagreement in Social Networks," *Mathematics of Operations Research*, vol. 38, no. 1, pp. 1–27, Feb. 2013.

[9] P. Jia, A. Mirtabatabaei, N. E. Friedkin, and F. Bullo, "Opinion Dynamics and the Evolution of Social Power in Influence Networks," *SIAM Review*, pp. 1–27, 2013.

[10] W. Ben-Ameur, P. Bianchi, and J. Jakubowicz, "Robust Average Consensus using Total Variation Gossip Algorithm," in *VALUETOOLS*, 2012, pp. 99–106.

[11] M. E. Yildiz and A. Scaglione, "Computing along routes via gossiping," *IEEE Trans. on Signal Process.*, vol. 58, no. 6, pp. 3313–3327, 2010.

[12] U. A. Khan, S. Kar, and J. M. F. Moura, "Higher dimensional consensus: Learning in large-scale networks," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2836–2849, 2010.

[13] M. Timme, "Revealing network connectivity from response dynamics," *Physical Review Letters*, vol. 98, no. 22, pp. 1–4, 2007.

[14] W.-X. Wang, Y.-C. Lai, C. Grebogi, and J. Ye, "Network Reconstruction Based on Evolutionary-Game Data via Compressive Sensing," *Physical Review X*, vol. 1, no. 2, pp. 1–7, 2011.

[15] A. De, S. Bhattacharya, P. Bhattacharya, N. Ganguly, and S. Chakrabarti, "Learning a Linear Influence Model from Transient Opinion Dynamics," *CIKM '14*, pp. 401–410, 2014.

[16] J. Mei and M. F. Moura, "Signal Processing on Graphs: Modeling Causal Relations in Big Data," no. 412, pp. 1–22, 2015. [Online]. Available: http://arxiv.org/abs/1503.0017

[17] H.-T. Wai, A. Scaglione, and A. Leshem, "The Social System Identification Problem," accepted by *IEEE CDC 2015*. [Online]. Available: http://arxiv.org/abs/1503.07288

[18] ——, "Active Sensing of Social Networks," submitted to *IEEE Trans. Sig. and Inf. Proc. over Networks*, 2015.

[19] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 12, pp. 1–305, 2008.

[20] A. Anandkumar, V. Y. F. Tan, F. Huang, and A. S. Willsky, "High-dimensional structure estimation in ising models: Local separation criterion," *Annals of Statistics*, vol. 40, no. 3, pp. 1346–1375, 2012.

[21] Y. F. Atchade, G. Fort, and E. Moulines, "On stochastic proximal gradient algorithms," *ArXiv e-prints*, Feb. 2014.

[22] L. Rosasco, S. Villa, and B. C. Vu, "Convergence of Stochastic Proximal Gradient Algorithm," *ArXiv e-prints*, 2014.

[23] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. on Signal Process.*, vol. 57, no. 7, pp. 2748–2761, 2009.

[24] N. Parikh and S. Boyd, "Proximal Algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2014.

[25] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[26] W. F. Stout, *Almost sure convergence*. Academic Press New York, 1974.

[27] T.-C. Hu, A. Rosalsky, and A. Volodin, "On convergence properties of sums of dependent random variables under second moment and covariance restrictions," *Statistics & Probability Letters*, vol. 78, no. 14, pp. 1999 – 2005, 2008.