Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 56-2 (2023) 664–669

Joint Learning of Network Topology and Opinion Dynamics Based on Bandit Algorithms ⋆

Yu Xing * **Xudong Sun** * **Karl H. Johansson** *

* *Division of Decision and Control Systems, EECS, KTH Royal Institute of Technology; Digital Futures, SE-10044 Stockholm, Sweden (e-mail: yuxing2@kth.se, smilesun.east@gmail.com, kallej@kth.se).*

Abstract: We study joint learning of network topology and a mixed opinion dynamics, in which agents may have different update rules. Such a model captures the diversity of real individual interactions. We propose a learning algorithm based on multi-armed bandit algorithms to address the problem. The goal of the algorithm is to find each agent's update rule from several candidate rules and to learn the underlying network. At each iteration, the algorithm assumes that each agent has one of the updated rules and then modifies network estimates to reduce validation error. Numerical experiments show that the proposed algorithm improves initial estimates of the network and update rules, decreases prediction error, and performs better than other methods such as sparse linear regression and Gaussian process regression.

Keywords: Social networks, identification, network inference, bandit algorithms

1. INTRODUCTION

Opinion dynamics characterize how individuals interact with each other and change their opinions, which has attracted researchers in various disciplines from control to physics for decades (Proskurnikov and Tempo (2017)). There is a growing interest in learning such dynamics (Ravazzi et al. (2021a)). Most researches formulate the learning problem based on a single model. In real networks agents may have different update rules, so there is a need to study how to learn such networked dynamics.

1.1 Related Work

This paper considers opinion dynamics with continuous states (Proskurnikov and Tempo (2017)). A classic example is the DeGroot model (DeGroot (1974)), in which agents update to the opinion average of their neighbors. Friedkin and Johnsen (1999) generalize this model by assuming the agents are influenced by their initial positions. Bounded confidence models, such as the Hegselmann–Krause (HK) model (Hegselmann et al. (2002)), characterize the case when agents only interact with those who hold opinions similar to themselves. Opinion dynamics over signed networks are another important class of models (Shi et al. (2019)). Signed networks have not only positive but also negative edges. There has been research on testing opinion dynamics models against data. Clemm von Hohenberg et al. (2017) find the existence of linear averaging. The FJ model has been validated in small-group experiments (Friedkin and Johnsen (1999)). Recent empirical studies based on large-scale datasets (Kozitsin (2021, 2022)) also suggest the presence of bounded confidence and negative influence rules.

With the increasing availability of large-scale online datasets, the interest in learning influence networks has been growing (Ravazzi et al. (2021a)). De et al. (2014) and Wai et al. (2016) study sparse network learning of the DeGroot model. Ravazzi et al. (2017) investigate the learning of the FJ model, and Ravazzi et al. (2021b) consider the case where only partial observations are available. Wai et al. (2019) study joint learning of network topology and system parameters for bounded confidence models. Learning based on quantized observations are studied in Xing et al. (2022); Xie et al. (2023). Most of these papers assume that the type of the dynamics model to be learned is known and every agent follows the same update rule. This is often not the case in real networks, so it is necessary to study learning of networked dynamics mixing multiple update rules. Automated machine learning (AutoML) is an emerging domain studing how to design algorithms that automatically build machine learning (ML) models with optimized hyperparameter settings (He et al. (2021)). Thornton et al. (2012) introduce the sequential model-based algorithm configuration to select algorithms and optimize hyperparameters. Li et al. (2017); Sun et al. (2019) propose methods based on multi-armed bandit and reinforcement learning algorithms.

1.2 Contribution

We study joint learning of network topology and a mixed opinion dynamics, where agents may have different update rules. In the learning problem, the network and the types of agent update rules are unknown. We propose a learning algorithm based on multi-armed bandits to address the problem. The algorithm starts with initial

⋆ This work was supported by the Knut & Alice Wallenberg Foundation, the Swedish Research Council, and the Swedish Foundation for Strategic Research.

10.1016/j.ifacol.2023.10.1643

estimates of the network and the agent update rules. At each iteration, it either refines the network estimates by exploiting a given update rule by modifying the adjacency matrix, or examines the validation error of other update rules with small probability. Numerical experiments show that the proposed algorithm can improve the initial estimates, with better network recovery and smaller prediction error. The algorithm provides a search strategy for learning opinion dynamics models with multiple update rules, and performs better than linear sparse regression and Gaussian process regression. Compared with existing network learning methods, the proposed algorithm can get rid of the assumption of a single model, and can include multiple sets of constraints. Thus the proposed algorithm can provide better possibility for learning real dynamics. The algorithm is inspired by AutoML, but the problem cannot be solved directly by AutoML methods. A multi-agent system is considered here, and the best model needs be selected for each agent. In addition, the network and dynamics are coupled.

1.3 Outline

The rest of the paper is organized as follows. Section 2 introduces several opinion dynamics models and defines the mixed model. Section 3 formulates the learning problem. We propose a learning algorithm for the mixed model in Section 4. Section 5 presents numerical experiments.

Notation. Let $\mathbf{1}_n$ be the n-dimensional all-one vector, and $\mathbf{0}_{n,m}$ be the $n \times m$-dimensional all-zero matrix. The Euclidean norm and the l_1 norm are $\|\cdot\|$ and $\|\cdot\|_1$. For a vector $x \in \mathbb{R}^n$, denote its subvector from i-th to j-the component by $x_{i:j}$, $i \leq j$. For a matrix $A = [a_{ij}] \in \mathbb{R}^{n\times n}$, denote its (i,j)-th entry by a_{ij} or $[A]_{ij}$ and denote its i-th column (row) by $[A]_{:,i}$ ($[A]_{i,:}$). The function $\mathbb{I}_{[\text{property}]}$ is the indicator function. If the property is stated for a vector, then the function is entry-wise (e.g., $\mathbb{I}_{[y>0]} = [\mathbb{I}_{[y_1>0]} \ \cdots \ \mathbb{I}_{[y_n>0]}]^{\mathrm{T}}$ for $y \in \mathbb{R}^n$). Denote the entry-wise sign of a vector y by $\operatorname{sgn}(y)$. The structure of a network is defined by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V} = \{1, \ldots, n\}$ is the agent set, $\mathcal{E}$ is the edge set, and A is the adjacency matrix. Each edge $\{i,j\} \in \mathcal{E}$ has a sign, either positive or negative. The disjoint sets $\mathcal{E}^+$ and $\mathcal{E}^-$ collect all positive and negative edges, respectively. The positive (negative) neighbors of an agent i is denoted by $\mathcal{N}_i^+ = \{j : \{i,j\} \in \mathcal{E}^+\}$ ($\mathcal{N}_i^- = \{j : \{i,j\} \in \mathcal{E}^-\}$), and $\mathcal{N}_i := \mathcal{N}_i^+ \cup \mathcal{N}_i^-$. An agent i has positive and negative degrees $d_i^+ = |\mathcal{N}_i^+|$ and $d_i^- = |\mathcal{N}_i^-|$, respectively, and degree $d_i := d_i^+ + d_i^-$. The adjacency matrix A satisfies that $a_{ij} = 1$ if $\{i,j\} \in \mathcal{E}^+$, $a_{ij} = -1$ if $\{i,j\} \in \mathcal{E}^-$, and $a_{ij} = 0$ if $\{i,j\} \notin \mathcal{E}$.

2. PRELIMINARIES

This section introduces considered models. We assume that each agent i has a self loop $\{i,i\} \in \mathcal{E}^+$ and has a state $x_i(t)$ at time $t \in \mathbb{N}$. Denote the state vector by $x(t) \in \mathbb{R}^n$.

The DeGroot Model (DeGroot (1974)) is one of the most classic opinion dynamics. The underlying graph does not have negative edges (i.e., $\mathcal{E}^- = \emptyset$). Each agent i updates its opinion to the weighed average of its neighbors' opinions:

$$x_i(t+1) = \sum_{j=1}^{n} w_{ij} x_j(t) =: f^{\mathcal{M}_{\mathrm{DG}}}(x(t), \theta^{(\mathcal{M}_{\mathrm{DG}},i)}), \quad (1)$$

where $t \in \mathbb{N}$ and w_{ij} is the influence weight of the agent j on the agent i and we denote

$$\theta^{(\mathcal{M}_{\mathrm{DG}},i)} = [w_{i1} \ \cdots \ w_{in}]^{\mathrm{T}}, \ i \in \mathcal{V}. \quad (2)$$

The weight w_{ij} satisfies that $w_{ij} > 0$ if $a_{ij} = 1$, and $w_{ij} = 0$ if $a_{ij} = 0$. In addition, $\sum_{i=1}^{n} w_{ij} = 1$.

The FJ model generalizes the DeGroot model. Here $\mathcal{E}^- = \emptyset$, and every agent i has susceptibility $\lambda_i \in [0,1]$ to others. An agent i updates according to the following rule

$$x_i(t+1) = \lambda_i \left(\sum_{i=1}^{n} w_{ij} x_j(t) \right) + (1-\lambda_i) x_i(0)$$
$$=: f^{\mathcal{M}_{\mathrm{FJ}}}(x(t), \theta^{(\mathcal{M}_{\mathrm{FJ}},i)}), \quad (3)$$

where w_{ij} are nonnegative weights such that $\sum_j w_{ij} = 1$, $w_{ij} > 0$ if $a_{ij} = 1$, and $w_{ij} = 0$ if $a_{ij} = 0$. Here we denote

$$\theta^{(\mathcal{M}_{\mathrm{FJ}},i)} := [w_{i1} \ \cdots \ w_{in} \ \lambda_i]^{\mathrm{T}}, i \in \mathcal{V}. \quad (4)$$

Note that $f^{\mathcal{M}_{\mathrm{FJ}}}$ also depends on $x_i(0)$, which is omitted for notation simplicity. When $\lambda_i \neq 1$, the opinion of the agent i is constantly influenced by its initial position $x_i(0)$. An agent i with $\lambda_i = 0$ is called stubborn and never changes its opinion. When $\lambda_i = 1$ for all $i \in \mathcal{V}$, the FJ model (3) degenerates to the DeGroot model (1).

The repelling negative dynamics (Shi et al. (2019)) can capture the opinion evolution where the network contains negative edges (i.e., $\mathcal{E}^- \neq \emptyset$):

$$\begin{aligned} &x_i(t+1) \\ &= x_i(t) + \alpha_i \sum_{j\in\mathcal{N}_i^+} (x_j(t) - x_i(t)) - \beta_i \sum_{j\in\mathcal{N}_i^-} (x_j(t) - x_i(t)) \\ &= (1 - \alpha_i d_i^+ + \beta_i d_i^-) x_i(t) + \alpha_i \sum_{j\in\mathcal{N}_i^+} x_j(t) - \beta_i \sum_{j\in\mathcal{N}_i^-} x_j(t) \\ &=: f^{\mathcal{M}_{\mathrm{RP}}}(x(t), \theta^{(\mathcal{M}_{\mathrm{RP}},i)}) \end{aligned} \quad (5)$$

where $0 < \alpha_i \leq 1/d_i^+$ and $\beta_i > 0$ are the influence strength of positive and negative neighbors on the agent i, respectively. Here we denote the parameters

$$\theta^{(\mathcal{M}_{\mathrm{RP}},i)} := [w_{i1} \ \cdots \ w_{in}]^{\mathrm{T}}, \ i \in \mathcal{V}, \quad (6)$$

where $w_{ii} = 1 - \alpha_i d_i^+ + \beta_i d_i^-$, $w_{ij} = \alpha_i$ if $j \in \mathcal{N}_i^+ \setminus \{i\}$, $w_{ij} = -\beta_i$ if $j \in \mathcal{N}_i^-$, and $w_{ij} = 0$ if $j \notin \mathcal{N}_i$. When $\mathcal{N}_i^- = \emptyset$ or $\beta_i = 0$ for all $i \in \mathcal{V}$, the model (5) becomes the DeGroot model (1).

Now we assume $\mathcal{E}^- = \emptyset$ again, and introduce the social HK model (Parasnis et al. (2018)). In the original HK model, the network is assumed to be complete, which is not realistic because individuals in a large network cannot know everyone. The social HK model addresses this issue by introducing an underlying network: At each time t, an agent i selects a set of trusted individuals from its neighbors (i.e., agents that have opinions similar to i),

$$\mathcal{I}_i(t) = \{j \in \mathcal{N}_i : |x_j(t) - x_i(t)| \leq c_i\},$$

where c_i is the confidence bound of i. Then the agent updates its opinions as the average of $x_j(t)$:

$$x_i(t+1) = \frac{1}{|\mathcal{I}_i(t)|} \sum_{j\in\mathcal{I}_i(t)} x_j(t) =: f^{\mathcal{M}_{\mathrm{HK}}}(x(t), \theta^{(\mathcal{M}_{\mathrm{HK}},i)}), \quad (7)$$

where

$$\theta^{(\mathcal{M}_{\rm HK},i)} := c_i,\ i \in \mathcal{V}. \quad (8)$$

Note that $f^{\mathcal{M}_{\rm HK}}$ depends on i, which is omitted for notation simplicity. When all c_i are larger than the range of $x(0)$, the model (7) reduces to the DeGroot model (1).

3. PROBLEM FORMULATION

Agents in real networks may have completely different update rules, as suggested by empirical evidence (Clemm von Hohenberg et al. (2017); Friedkin and Johnsen (1999); Kozitsin (2021, 2022)). Thus, to learn the real network and the dynamics, it is natural to consider a mixture of opinion update rules Dong et al. (2017); Wu et al. (2022). A mixed opinion dynamics is defined as follows.

Definition 1. A mixed opinion dynamics over an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ with $|\mathcal{V}| = n$ is a discrete-time system with states $x(t) \in \mathbb{R}^n$, $t \in \mathbb{N}$, satisfying that

$$x_i(t+1) = f^{\mathcal{M}_i}(x(t), \theta^{(\mathcal{M}_i,i)}),$$

where $\mathcal{M}_i \in \{\mathcal{M}_{\rm DG}, \mathcal{M}_{\rm FJ}, \mathcal{M}_{\rm RP}, \mathcal{M}_{\rm HK}\}$ is the type of update rule of the agent i, $f^{\mathcal{M}_i}$ are given in (1), (3), (5), and (7), and $\theta^{(\mathcal{M}_i,i)}$ in (2), (4), (6), and (8).

To make the definition well-posed, we introduce the following assumptions.

Assumption 2.
(i) The graph $\mathcal{G}$ is undirected and connected.
(ii) There exists a positive constant $\varepsilon_\lambda \in (0,1)$ such that, if $\mathcal{M}_i = \mathcal{M}_{\rm FJ}$ for some $i \in \mathcal{V}$, then $\varepsilon_\lambda \le \theta_{n+1}^{(\mathcal{M}_{\rm FJ},i)} \le 1 - \varepsilon_\lambda$.
(iii) For $i \in \mathcal{V}$, $\mathcal{N}_i^- \neq \emptyset$ holds if and only if $\mathcal{M}_i = \mathcal{M}_{\rm RP}$.
(iv) If $\mathcal{M}_i = \mathcal{M}_{\rm HK}$ for $i \in \mathcal{V}$, then $\theta^{(\mathcal{M}_{\rm HK},i)} < \max_{j\in\mathcal{V}}\{|x_j(0)|\}$.

Remark 3. The second assumption ensures that the susceptibility of an agent i should be neither too small nor too large. The third assumption guarantees that agents who have other update rules do not have negative edges. The last assumption ensures that the agent with a HK update rule cannot be considered as a DeGroot-type agent.

We consider the joint learning of the network topology and the dynamics for the mixed model:

Problem. Given a trajectory $\{x(0), x(1), \ldots, x(T)\}$ of the mixed model, propose an algorithm jointly learning the network $\mathcal{G}$, the types of agent update rules $\{\mathcal{M}_i\}$, and the parameters of each agent $\{\theta^{(\mathcal{M}_i,i)}\}$, where $T \ge 1$ is the final time step of the trajectory.

4. LEARNING ALGORITHMS

This section presents separate learning algorithms for models $\{\mathcal{M}_{\rm DG}, \mathcal{M}_{\rm FJ}, \mathcal{M}_{\rm RP}, \mathcal{M}_{\rm HK}\}$, and proposes a learning algorithm for the mixed model. Denote the data matrices by $\boldsymbol{X} := [x(0)\ x(1)\ \ldots\ x(T-1)]^{\rm T}$ and $b^{(i)} := [x_i(1)\ \ldots\ x_i(T)]^{\rm T}$, $i \in \mathcal{V}$, and the estimates of the adjacency matrix A, agent update rules $\{\mathcal{M}_i\}$, and the parameter $\{\theta^{(\mathcal{M}_i,i)}\}$ by $\hat{A}$, $\{\hat{\mathcal{M}}_i\}$, and $\{\hat{\theta}^{(\mathcal{M}_i,i)}\}$, respectively.

4.1 Learning of Single Models

Note that $\theta^{(\mathcal{M}_{\rm DG},i)}$ satisfies $\boldsymbol{X}\theta^{(\mathcal{M}_{\rm DG},i)} = b^{(i)}$, for $i \in \mathcal{V}$ such that $\mathcal{M}_i = \mathcal{M}_{\rm DG}$. To learn $\theta^{(\mathcal{M}_{\rm DG},i)}$ we look for

Algorithm 1 Learn$\mathcal{M}_{\rm DG}$($\boldsymbol{X}$,$b^{(i)}$,$\mathcal{N}_i^{(\rm neigh)}$,$\mathcal{N}_i^{(\rm non)}$)

1: For $y \in \mathbb{R}^n$, solve

$$\begin{aligned} \min\ & \|y\|_1 \\ \text{s.t. } & \boldsymbol{X}y = b^{(i)}, \\ & \mathbf{1}^T y = 1, \\ & y_j \ge \varepsilon_w,\ j \in \mathcal{N}_i^{(\rm neigh)}, \\ & y_j = 0,\ j \in \mathcal{N}_i^{(\rm non)}, \\ & y_j \ge 0,\ j \in \mathcal{V} \setminus (\mathcal{N}_i^{(\rm neigh)} \cup \mathcal{N}_i^{(\rm non)}). \end{aligned}$$

2: Return $[\hat{A}]_{i,:} = \mathbb{I}_{[y^{\rm T}>0]}$, $\hat{\theta}^{(\mathcal{M}_{\rm DG},i)} = y$.

Algorithm 2 Learn$\mathcal{M}_{\rm FJ}$($\boldsymbol{X}$,$b^{(i)}$,$\mathcal{N}_i^{(\rm neigh)}$,$\mathcal{N}_i^{(\rm non)}$)

1: For $y \in \mathbb{R}^{n+1}$, solve

$$\begin{aligned} \min\ & \|y\|_1 \\ \text{s.t. } & [\boldsymbol{X}\ [\boldsymbol{X}]_{1,i}\mathbf{1}_T]y = b^{(i)}, \\ & \mathbf{1}^T y = 1, \\ & y_j \ge \varepsilon_w\varepsilon_\lambda,\ j \in \mathcal{N}_i^{(\rm neigh)}, \\ & y_j = 0,\ j \in \mathcal{N}_i^{(\rm non)}, \\ & y_j \ge 0,\ j \in \mathcal{V} \setminus (\mathcal{N}_i^{(\rm neigh)} \cup \mathcal{N}_i^{(\rm non)}), \\ & \varepsilon_\lambda \le y_{n+1} \le 1 - \varepsilon_\lambda. \end{aligned}$$

2: Return $[\hat{A}]_{i,:} = \mathbb{I}_{[y_{1:n}^{\rm T}>0]}$, $\hat{\theta}_{1:n}^{(\mathcal{M}_{\rm FJ},i)} = y_{1:n}/(1-y_{n+1})$, $\hat{\theta}_{n+1}^{(\mathcal{M}_{\rm FJ},i)} = 1 - y_{n+1}$.

Algorithm 3 Learn$\mathcal{M}_{\rm RP}$($\boldsymbol{X}$,$b^{(i)}$,$\mathcal{N}_i^{(\rm neigh)}$,$\mathcal{N}_i^{(\rm non)}$)

1: For $y \in \mathbb{R}^n$, solve

$$\begin{aligned} \min\ & \|y\|_1 \\ \text{s.t. } & \boldsymbol{X}y = b^{(i)}, \\ & y_j \ge \varepsilon_w, j \in \mathcal{N}_i^{(\rm neigh)}, \\ & y_j = 0, j \in \mathcal{N}_i^{(\rm non)}. \end{aligned}$$

2: Return $[\hat{A}]_{i,:} = \text{sgn}(y^{\rm T})$, $\hat{\theta}^{(\mathcal{M}_{\rm RP},i)} = y$.

a sparse solution to the least l_1-norm problem given in Algorithm 1. The motivation is that in practice only a few samples of the process are available ($T < n$) but real networks are often sparse (Ravazzi et al. (2017)). The parameter constraints $\mathbf{1}^T y = 1$ and $y_j \ge 0$, $1 \le j \le n$, are added. The algorithm has two additional inputs $\mathcal{N}_i^{(\rm neigh)}$ and $\mathcal{N}_i^{(\rm non)}$, which are subsets of $\mathcal{V}$. We introduce these two sets to search for other solutions. If an agent j is considered as a neighbor of the agent i ($j \in \mathcal{N}_i^{(\rm neigh)}$), we add the constraint $y_j \ge \varepsilon_w$, where ε_w is a small positive constant. If j is assumed to not be a neighbor ($j \in \mathcal{N}_i^{(\rm non)}$), the constraint $y_j = 0$ is imposed.

We can similarly solve a system of linear equations to learn the parameters in the FJ model (3). We search for a sparse solution in Algorithm 2. The algorithm has two hyperparameters: the lower bound of influence weights $\varepsilon_w > 0$, and the bound for the susceptibility $\varepsilon_\lambda \in (0,1)$ given in Assumption 2 (ii).

The expression of the repelling negative dynamics (5) is the same as (1), but (5) has less constraints. Hence so does Algorithm 3. Testing whether a negative edge should

Algorithm 4 Learn$\mathcal{M}_{\mathrm{HK}}(\boldsymbol{X}, b^{(i)}, \mathcal{N}_i^{(\mathrm{neigh})}, \mathcal{N}_i^{(\mathrm{non})})$

Set $[\hat{A}]_{i,:}$ such that $[\hat{A}]_{i,j} = 1$ if $j \in \mathcal{N}_i^{(\mathrm{neigh})}$ and $[\hat{A}]_{i,j} = 0$ otherwise.
for t from 0 to $T-1$ **do**
  Sort the neighbors $j \in \mathcal{N}_i^{(\mathrm{neigh})}$ by the distance of $x_j(t)$ from $x_i(t)$ to be $x_{i_1}, \ldots, x_{i_{n_i}}$, where $n_i := [\hat{A}]_{i,:}\mathbf{1}$.
  Compute

$$\hat{x}_i^{(m)}(t+1) = \frac{1}{m}\sum_{j=1}^{m} x_{i_j}(t),\ 1 \le m \le n_i,$$

$$m(t) = \max\left\{\arg\min_{1\le m\le n_i} |\hat{x}_i^{(m)}(t+1) - x_i(t+1)|\right\},$$

$$c(t) = |x_{i_{m(t)+1}}(t) - x_i(t)| - \varepsilon_c.$$

end for
Compute

$$t^* = \arg\min_{0\le t\le T-1} c(t),$$

$$c^* = c(t^*).$$

Return $[\hat{A}]_{i,:}$, $\hat{\theta}^{(\mathcal{M}_{\mathrm{HK}},i)} = c^*$.

be kept complicates the learning algorithm. We leave this complement to future work.

For the Social HK model, it is only possible to obtain a bound for the confidence bounds based on finite samples. We provide a heuristic to estimate the confidence bound. Numerical experiments in Section 5 show that the proposed algorithm can have relatively small prediction error. Assume first that the network topology is known. For an agent i with $\mathcal{M}_i = \mathcal{M}_{\mathrm{HK}}$, sort i's neighbors by their states' distance from $x_i(t)$ and denote the sorted neighbors by $i_1, i_2, \ldots, i_{|\mathcal{N}_i|}$, where $i_1 = i$. Then compute the averages $(\sum_{j=1}^{m} x_{i_j}(t))/m$, $1 \le m \le |\mathcal{N}_i|$, and find the index such that the average has the smallest distance from $x_i(t+1)$, i.e., $\arg\min_{1\le m\le|\mathcal{N}_i|} |(\sum_{j=1}^{m} x_{i_j}(t))/m - x_i(t+1)|$. If no noise exists, there exist $m_1 < \cdots < m_q$ such that $|(\sum_{j=1}^{m_r} x_{i_j}(t))/m_r - x_i(t+1)| = 0$, where $1 \le r \le q$ and $q \ge 1$. We then can conclude that $|x_{i_{m_q+1}}(t) - x_i(t)|$ is a strict upper bound for the confidence bound $\theta^{(\mathcal{M}_{\mathrm{HK}},i)} = c_i$. If the topology is unknown, to obtain a bound we need test all 2^{n-1} combinations. We narrow down the search by introducing testing sets $\mathcal{N}_i^{(\mathrm{neigh})}$ and $\mathcal{N}_i^{(\mathrm{non})}$, as in the preceding algorithms. In Algorithm 4, for each sample $(x(t), x_i(t+1))$, we obtain an estimate of the confidence bound $c(t)$ by conducting the process discussed previously. In the algorithm, ε_c is a small positive number to make sure that the upper bound is strict. For all $0 \le t \le T-1$, if the neighbor set is correct, the confidence bound should be smaller than all $c(t)$. The algorithm returns an approximation of i's dynamics.

4.2 Learning of Mixed Model

After presenting the learning algorithms for each update rule, now we are ready to introduce the learning algorithm for the mixed model. We propose a multi-armed bandit algorithm (Algorithm 5) to address the problem, and consider the four types of update rules as four arms. We introduce a Q-table $Q(l) \in \mathbb{R}^{n\times 4}$ for each iteration l. The entry $[Q(l)]_{i,m}$ indicates the payoff of refining model parameters of the arm $M[m]$ for the agent i at the iteration l, where $1 \le m \le 4$ and we define the look-up table M with respect to $\{\mathcal{M}_{\mathrm{DG}}, \mathcal{M}_{\mathrm{FJ}}, \mathcal{M}_{\mathrm{RP}}, \mathcal{M}_{\mathrm{HK}}\}$ such that $M[1] = \mathcal{M}_{\mathrm{DG}}$, $M[2] = \mathcal{M}_{\mathrm{FJ}}$, $M[3] = \mathcal{M}_{\mathrm{RP}}$, $M[4] = \mathcal{M}_{\mathrm{HK}}$. We run Algorithms 1–4 for each i with $\mathcal{N}_i^{(\mathrm{neigh})} = \{i\}$ and $\mathcal{N}_i^{(\mathrm{non})} = \emptyset$, to initialize the estimates of update rule types $\hat{k}(0) \in \{1,2,3,4\}^n$, the estimates of the adjacency matrix for each model $\hat{A}^{(M[m])}(0)$, and the parameter estimates $\hat{\theta}^{(M[m],i)}(0)$. $[Q(0)]_{i,m}$ is set to be the negative logarithm of validation error of the model $M[m]$ for i (Line 2).

At each iteration and for each agent, the algorithm tries to fit the data into the model with the best payoff with probability $1-\varepsilon_{\mathrm{M}}$. With exploration probability ε_{M}, the algorithm attempts to refine estimates of other models (Line 4–7). At each iteration, the algorithm modifies the adjacency matrix corresponding to the selected models with probability $1-\varepsilon_{\mathrm{G}}$, and modifies a randomly generated adjacency matrix with probability ε_{G} (Line 8). The algorithm then examines whether removing or adding an edge can improve validation error (Line 11). Next, the algorithm updates the payoff Q with a step-size α and the parameter estimates (Line 13). Finally the algorithm returns parameter estimates, according to the update rule with the best payoff (Line 17).

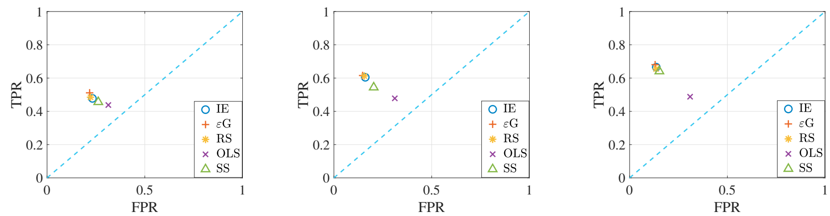
5. NUMERICAL EXPERIMENTS

This section presents a numerical experiment for illustration of algorithm performance. We use the CVX toolbox [1] to find sparse solutions.

To generate the mixed model, we set the network size to be $n = 20$, with four types of agents and each type with 5 agents sharing the same model as ground truth. We generate 10 graphs and run the model over each graph to get a trajectory with a final time step $t = 20$. The positive graphs are generated from an Erdös-Rényi graph with link probability $(1.1\log n)/n$ and are checked to be connected. Generate negative edges between the 5 agents with the repelling rule, with the same link probability, and ensure that each agent has at least one negative edge. The influence weights α_i and β_i are set to be $0.2/d_i$. The susceptibility λ_i of the agents with the FJ rule is set to be 0.5, and the confidence bound c_i of the social HK model is set to be 0.25. The initial opinions of the agents are generated independently and uniformly from $(-1, 1)$.

We compare the proposed algorithm (Algorithm 5, εG) with an initial estimate (IE) of the mixed model, random search (RS, Algorithm 5 with model and topology exploration probabilities $\varepsilon_M = 1$ and $\varepsilon_G = 1$), the ordinary least-squares algorithm (OLS), sparse solutions (SS) to $\boldsymbol{X}y = b$, and Gaussian process regression (GPR). For IE, we run Algorithm 1–4 (with $\varepsilon_w = 0.001$, $\varepsilon_\lambda = 0.1$, and $\varepsilon_c = 10^{-6}$) for each agent, and obtain initial estimates of update rules, model parameters, and the adjacency matrix. εG starts with the estimates of IE, the number of iterations is set to be 20, both ε_M and ε_G are set to be 0.2, respectively, and the step size is 0.1. For OLS, SS, and GPR, the data matrix for an agent i is $\begin{bmatrix} x(0) & \cdots & x(T-1) \\ x_i(0) & \cdots & x_i(0) \end{bmatrix}^{\mathrm{T}}$, thus including the effect of initial states in the FJ model. Here T is the number of samples. For OLS and SS, we search

[1] http://cvxr.com/cvx



(a) $T = 10$ (b) $T = 15$ (c) $T = 20$

Figure 1. Performance of algorithms IE, εG, RS, OLS, and SS for network topology learning.

Algorithm 5 ε-Greedy($\{x(0), \ldots, x(T)\}$)

Choose $\tilde{T}$ and divide the trajectory into a training dataset $\boldsymbol{X}_{\text{tr}} = [x(0) \ \cdots \ x(\tilde{T})]^{\mathrm{T}}$ and a validation dataset $\boldsymbol{X}_{\text{val}} = [x(\tilde{T}+1) \ \cdots \ x(T)]^{\mathrm{T}}$.
Initialize the Q-table $Q(0)$, the estimates of update rule types $\hat{k}(0)$, the estimates of the adjacency matrix for each model $\hat{A}^{(M[m])}(0)$, and the parameter estimates $\hat{\theta}^{(M[m],i)}(0)$, $1 \le m \le 4$, where the look-up table M is s.t. $M[1] = \mathcal{M}_{\text{DG}}$, $M[2] = \mathcal{M}_{\text{FJ}}$, $M[3] = \mathcal{M}_{\text{RP}}$, $M[4] = \mathcal{M}_{\text{HK}}$.
for l from 1 to n_{iter} **do**
  for i from 1 to n **do**
    With probability (w.p.) $1 - \varepsilon_{\text{M}}$, set
$$k_i(l) = \arg\max_{1 \le m \le 4} [Q(l)]_{i,m}.$$
    W.p. ε_{M}, generate $k_i(l) \in \{1, 2, 3, 4\}$ uniformly.
  end for
  W.p. $1 - \varepsilon_{\text{G}}$, let $[\tilde{A}]_{i,:} := [\hat{A}^{(k_i(l))}(l-1)]_{i,:}$, for $i \in \mathcal{V}$. W.p. ε_{G}, randomly generate an adjacency matrix $\tilde{A}$.
  for i from 1 to n **do**
    for j from 1 to n and $j \neq i$ **do**
      If $\tilde{A}_{ij} = 0$ set $\mathcal{N}_i^{(\text{neigh})} := \{q \in \mathcal{V} : \tilde{A}_{iq} = 1\} \cup \{j\}$ and $\mathcal{N}_i^{(\text{non})} := \emptyset$. If $\tilde{A}_{ij} = 1$ set $\mathcal{N}_i^{(\text{neigh})} := \{q \in \mathcal{V} : \tilde{A}_{iq} = 1\} \setminus \{j\}$ $\mathcal{N}_i^{(\text{non})} := \{j\}$.
$$[\hat{A}_{\text{temp}_j}]_{i,:}, \hat{\theta}_{\text{temp}_j}^{(M[k_i(l)],i)}$$
$$= \text{Learn}M[k_i(l)]([\boldsymbol{X}_{\text{tr}}]_{1:\tilde{T}-1,:}, [\boldsymbol{X}_{\text{tr}}]_{2:\tilde{T},i}, \mathcal{N}_i^{(\text{neigh})}, \mathcal{N}_i^{(\text{non})})$$
    end for
    Compute the validation error of $f^{M[k_i(l)]}$ based on $([\hat{A}_{\text{temp}_j}]_{i,:}, \hat{\theta}_{\text{temp}_j}^{(M[k_i(l)],i)})$, $j \in \mathcal{V} \setminus \{i\}$. Choose the smallest error $p_i(l)$ with respect to j, update $[\hat{A}^{(M[k_i(l)])}(l)]_{i,:}$, $\hat{\theta}^{(M[k_i(l)],i)}(l)$ accordingly, and
$$[Q(l)]_{i,k_i(l)} = (1-\alpha)[Q(l-1)]_{i,k_i(l)} - \alpha \log(p_i(l)).$$
  end for
end for
Let $m_i := \arg\max_{1 \le m \le 4} [Q(n_{\text{iter}})]_{i,m}$, $\hat{\mathcal{M}}_i := M[m_i]$, $[\hat{A}]_{i,:} = [\hat{A}^{(\hat{\mathcal{M}}_i)}(n_{\text{iter}})]_{i,:}$, and $\hat{\theta}^{(\hat{\mathcal{M}}_i,i)} = \hat{\theta}^{(\hat{\mathcal{M}}_i,i)}(n_{\text{iter}})$, $i \in \mathcal{V}$.
Return $\hat{A}$, $\{\hat{\mathcal{M}}_i\}$, and $\{\hat{\theta}^{(\hat{\mathcal{M}}_i,i)}\}$.

solutions in $[-2, 2]^{n+1}$ to avoid large weight estimates. To investigate the effect of sample number, for each $T = 10, 15, 20$, we apply the algorithms to the 10 trajectories $\{x(0), \ldots, x(T)\}$ (for εG, set $\tilde{T} = [4T/5]$), and study the averaged performance.

First, we examine the recovery of adjacency matrices by EI, εG, RS, OLS, and SS. The performance of the recovery is characterized by the the true positive rate (TPR) and the false positive rate (FPR) (see Bishop (2006)). Fig. 1 shows the pair (FPR, TPR) for the algorithms. As the number of samples increase, all algorithms perform better. εG is slightly better then IE when T is small, indicating refinement of the initial estimates. SS has similar FPR and TPR to IE, which may be because SS also searches for sparse networks. But SS does not learn the model well and has large prediction error, as discussed later.

Next, we compute the prediction error for all algorithms. For each graph, we generate 50 time-adjecent state pairs $\{x^{(g)}(0)), (x^{(g)}(1))\}$, $1 \le g \le 50$. The prediction error is defined by the root mean square error (RMSE) $[(\sum_{g=1}^{50} \|\hat{x}^{(g)}(1) - x^{(g)}(1)\|^2)/50]^{\frac{1}{2}}$, where $\hat{x}^{(g)}(1)$ is the prediction of the states $x^{(g)}(1)$. The results are shown in Fig. 2 with boxplots. IE, εG, and RS have smaller prediction error than the rest, because they are equipped with more model information. When the number of samples is small ($T = 10$), εG and RS have better performance for some trajectories but they do not improve IE too much. Improvement can be observed in the case $T = 15$, where εG has smaller error than IE and RS. When $T = 20$, the performance of εG is similar to IE and RS, since there is sufficient information for all algorithms. OLS cannot capture the nonlinear dynamics so it performs the worst. SS and GPR has less prediction error than OLS, because SS finds sparse solutions, resulting in less error, and GPR can capture nonlinear dynamics.

We illustrate how the estimates of update rule types and parameters are influenced by the number of samples. Fig. 3 plots the accuracy (the proportion of correct estimates) of IE, εG, and RS for each update rule type. The accuracy increases with the number of samples. εG has similar accuracy to RS, so the smaller prediction error may result from the exploitation for correctly learned update rules.

In summary, the proposed algorithm can improve the initial estimates of the network and the update rules, and reduce prediction error. Future work will study improvement and generalization of the proposed algorithm.

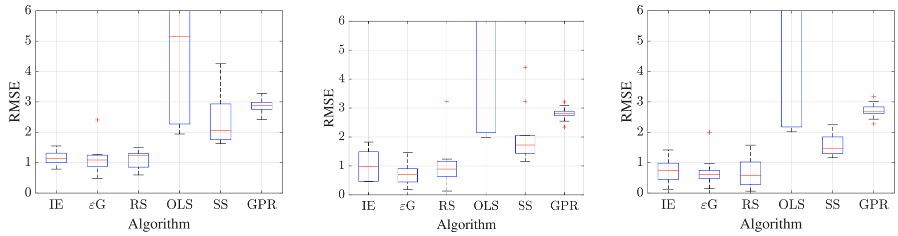
REFERENCES

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Clemm von Hohenberg, B., Maes, M., and Pradelski, B. (2017). Micro influence and macro dynamics of opinion formation. *SSRN 2974413*.

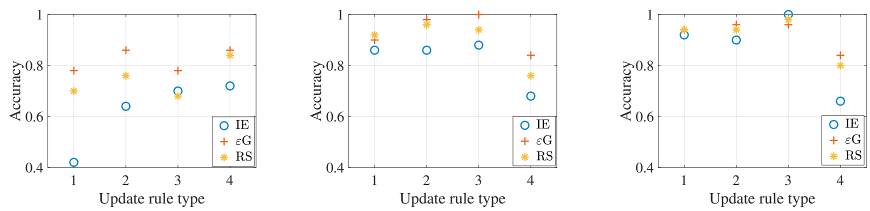
De, A., Bhattacharya, S., Bhattacharya, P., Ganguly, N., and Chakrabarti, S. (2014). Learning a linear influence model from transient opinion dynamics. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 401–410.

DeGroot, M.H. (1974). Reaching a consensus. *Journal of the American Statistical association*, 69(345), 118–121.



(a) $T = 10$ (b) $T = 15$ (c) $T = 20$

Figure 2. Prediction error of algorithms IE, εG, RS, OLS, SS, and GPR.



(a) $T = 10$ (b) $T = 15$ (c) $T = 20$

Figure 3. Accuracy of algorithms IE, εG, and RS for learning of update rule types.

Dong, Y., Ding, Z., Chiclana, F., and Herrera-Viedma, E. (2017). Dynamics of public opinions in an online and offline social network. *IEEE Transactions on Big Data*, 7(4), 610–618.

Friedkin, N.E. and Johnsen, E.C. (1999). Social influence networks and opinion change. *Advances in Group Processes*, 16, 1–29.

He, X., Zhao, K., and Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212, 106622.

Hegselmann, R., Krause, U., et al. (2002). Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3).

Kozitsin, I.V. (2021). Opinion dynamics of online social network users: A micro-level analysis. *The Journal of Mathematical Sociology*, 1–41.

Kozitsin, I.V. (2022). Formal models of opinion formation and their application to real data: Evidence from online social networks. *The Journal of Mathematical Sociology*, 46(2), 120–147.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1), 6765–6816.

Parasnis, R., Franceschetti, M., and Touri, B. (2018). Hegselmann-Krause dynamics with limited connectivity. In *IEEE Conference on Decision and Control*, 5364–5369.

Proskurnikov, A.V. and Tempo, R. (2017). A tutorial on modeling and analysis of dynamic social networks. Part I. *Annual Reviews in Control*, 43, 65–79.

Ravazzi, C., Dabbene, F., Lagoa, C., and Proskurnikov, A.V. (2021a). Learning hidden influences in large-scale dynamical social networks: A data-driven sparsity-based approach, in memory of Roberto Tempo. *IEEE Control Systems Magazine*, 41(5), 61–103.

Ravazzi, C., Hojjatinia, S., Lagoa, C.M., and Dabbene, F. (2021b). Ergodic opinion dynamics over networks: Learning influences from partial observations. *IEEE Transactions on Automatic Control*, 66(6), 2709–2723.

Ravazzi, C., Tempo, R., and Dabbene, F. (2017). Learning influence structure in sparse social networks. *IEEE Transactions on Control of Network Systems*, 5(4), 1976–1986.

Shi, G., Altafini, C., and Baras, J.S. (2019). Dynamics over signed networks. *SIAM Review*, 61(2), 229–257.

Sun, X., Lin, J., and Bischl, B. (2019). Reinbo: Machine learning pipeline conditional hierarchy search and configuration with Bayesian optimization embedded reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 68–84.

Thornton, C., Hutter, F., Hoos, H.H., and Leyton-Brown, K. (2012). Auto-WEKA: Automated selection and hyper-parameter optimization of classification algorithms. *CoRR, abs/1208.3719*.

Wai, H.T., Scaglione, A., Barzel, B., and Leshem, A. (2019). Joint network topology and dynamics recovery from perturbed stationary points. *IEEE Transactions on Signal Processing*, 67(17), 4582–4596.

Wai, H.T., Scaglione, A., and Leshem, A. (2016). Active sensing of social networks. *IEEE Transactions on Signal and Information Processing over Networks*, 2(3), 406–419.

Wu, Z., Zhou, Q., Dong, Y., Xu, J., Altalhi, A.H., and Herrera, F. (2022). Mixed opinion dynamics based on DeGroot model and Hegselmann-Krause model in social networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(1), 296–308.

Xie, X., Katselis, D., Beck, C.L., and Srikant, R. (2023). Finite sample analysis for structured discrete system identification. *IEEE Transactions on Automatic Control*, Early Access.

Xing, Y., He, X., Fang, H., and Johansson, K.H. (2022). Recursive network estimation for a model with binary-valued states. *IEEE Transactions on Automatic Control*, Early Access.