Learning Cascaded Influence under Partial Monitoring

Jie Zhang*
Department of Physics
Tsinghua University
zhangjie12@mails.tsinghua.edu.cn

Jiaqi Ma*
Department of Automation
Tsinghua University
mjq12@mails.tsinghua.edu.cn

Jie Tang
Department of Computer Science
Tsinghua University
jietang@tsinghua.edu.cn

***Abstract*—Social influence has attracted tremendous attention from both academic and industrial communities due to the rapid development of online social networks. While most research has been focused on the direct influence between peers, learning cascaded indirect influence has not been previously studied. In this paper, we formulate the concept of cascade indirect influence based on the Independent Cascade model and then propose a novel online learning algorithm for learning the cascaded influence in the partial monitoring setting. We propose two bandit algorithms E-EXP3 and RE-EXP3 to address this problem. We theoretically prove that E-EXP3 has a cumulative regret bound of $O(\sqrt{T})$ over T, the number of time stamps. We will also show that RE-EXP3, a relaxed version of E-EXP3, achieves a better performance in practice. We compare the proposed algorithms with three baseline methods on both synthetic and real networks (Weibo and AMiner). Our experimental results show that RE-EXP3 converges $100\times$ faster than E-EXP3. Both of them significantly outperform the alternative methods in terms of normalized regret. Finally, we apply the learned cascaded influence to help behavior prediction and experiments show that our proposed algorithms can help achieve a significant improvement (10-15% by accuracy) for behavior prediction.**

I. Introduction

Social influence is the phenomenon that people's opinions, emotions or behaviors are affected by others. The recent success of viral marketing is a strong evidence of social influence [8], [12], [1], [22], [17]. Much research has been conducted about social influence including pairwise influence [14], [23], topic influence [25], external influence [7], community influence [21], [26] and local influence [28], [30]. However, most of the previous works focused on studying the influence between neighbors in the social network and very few research has been conducted for learning indirect influence between users who are not directly connected in the social network.

In this paper, we aim to conduct a systematical study for learning cascaded indirect influence. In particular, we study this problem in the partial monitoring setting—due to the fact that interactions between users who are not connected are very sparse.

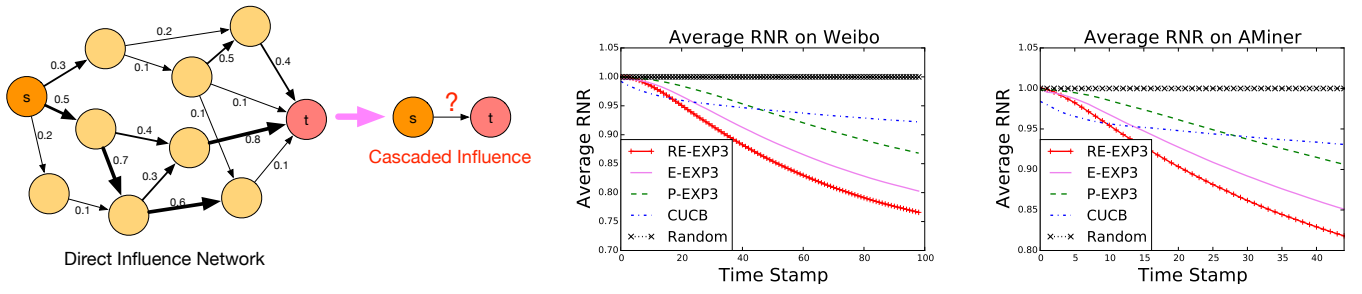
This problem is important and can benefit many real world applications in some ways. For example, indirect influence can help improve the efficiency of viral marketing (aka influence maximization). More information indicated by the indirect influence can be incorporated into the algorithms to avoid redundant influence to the same group of users. Indirect influence can also benefit friend recommendation or link prediction problems by knowing which pair of disconnected users may have potential high indirect influence.

Challenges and the Solution. The problem is very challenging as well. First, information about users who are not directly connected is rare as they may not have any interactions in the social network. It is natural for us to consider mining the indirect influence from their intermediate users and paths. However, the number of potential paths between two users is exponentially large. Investigating all the paths between two users is unrealistic when facing large scale social networks. It is nontrivial to determine which paths are more important for a user to influence another indirectly. Furthermore, most of the previous works on social influence infer the influence intensity from the propagation cascade data [14], [23] , which is often partial, sparse and could be very different over time. Thus, how to make full use of some local cascade data rather than that of the whole network is worth considering.

To address these difficulties, we first formally define the concept of *Cascaded indirect Influence* with *Influence Path*. Specifically, a user s may pass his influence to another user t by the paths between them. If t is heavily influenced by some intermediate users and these users are heavily influenced by the user s, t is probably influenced by s indirectly. Figure 1(a) shows an example of the indirect influence. The cascaded indirect influence will be the probability that user t is activated given the direct pairwise influence network bridging the two users and the fact user s is activated.

To further handle the challenges that the number of paths is too large as well as the cascade data is partial and sparse, we propose a novel problem of learning indirect influence from social networks under the partial monitoring setting. That is, the learning algorithm is asked to guess limited number of paths with highest influence at each time and then is allowed to observe the influence on these paths. At the next time stamp, the algorithm makes another guess based on the observed history data. The algorithm is required to minimize the gap between the cascaded indirect influence in the observed network and that in real network (aka regret). As the problem inherently asks to maintain an exploration-

* indicates equal contribution.



(a) Example of indirected influence (b) Average RNR on Weibo (c) Average RNR on AMiner

Figure 1. (a) Example of indirect influence, (b) Performance of comparison methods on Weibo (by Normalized Regret) and (c) Performance on AMiner.

exploitation balance to achieve good performance, we propose two algorithms, E-EXP3 and RE-EXP3 based on the well known online learning algorithms EXP3 [5]. We prove that the expected cumulative regret in E-EXP3 is $O(\sqrt{T})$.

Contribution. We summarize our main contributions as follows,

- We formulate the concept of cascade indirect influence and propose a novel online learning problem, learning cascaded indirect influence in large partial monitoring social networks, which make us able to use dynamic and local cascade data to infer indirect social influence.
- We propose two online learning algorithms, E-EXP3 and RE-EXP3, to solve the problem and theoretically prove that the E-EXP3 algorithm results in a cumulative regret bound of $O(\sqrt{T})$, where T is the number of time stamps.
- We evaluate the proposed algorithms on both synthetic and real network datasets (Weibo[1] and AMiner[2]). Our empirical study (as shown in Figure 1(b) and 1(c)) on two real networks shows that the proposed algorithms significantly outperform several alternative methods in terms of normalized regret.
- We also find that RE-EXP3 is more practical compared with E-EXP3, with a convergent ratio of $100\times$ faster than E-EXP3. We applied the learned indirect influence by RE-EXP3 to help behavior prediction. Experiments show that RE-EXP3 can help achieve a significant improvement (10-15% by accuracy) for behavior prediction.

Organization. The rest of the paper is organized as follows. Section 2 formulates the problem. Section 3 proposes the algorithms, E-EXP3 and RE-EXP3, solving the problem and provides theoretical analysis of the algorithms. Section 4 presents experimental results. Section 5 reviews related works and Section 6 concludes the paper.

II. Problem Formulation

Inspired by Independent Cascade (IC) Model[17], we first give the definition of *cascaded indirect influence* by formulating the concept of *influence path* and then give the formal description of the problem of *learning indirect influence under online partial monitoring setting*.

[1]http://weibo.com, the largest Chinese microblogging service.
[2]http://aminer.org, an author-centric search and mining system.

A. Cascaded Influence

Definition 1. ***Dynamic (Direct) Influence Network.*** *Given a time series $t, t = 1, 2, ..., T$, we define a social network as a weighted directed acyclic graph $G_t = (V, E, W_t)$, where V and E indicates the set of users and directed edges, respectively. And $w_{e,t} \in W_t, e \in E$ indicates the pairwise direct influence on edge e at time t.*

The pairwise direct influence on edges can be obtained in different ways, e.g., methods in [14], [23]. In this paper, we just define the pairwise influence $w_{e,t}, e = (u, v)$ as the intensity of the behavior following between u and v with an exponential decaying kernel:

$$w_{e,t} = \sum_i e^{-(t-\tau_i)/\delta}$$

where τ_i is the time of the i_{th} time when v follows u's behavior and δ is a constant.

Given a dynamic influence network, we define influence path as follows,

Definition 2. ***Influence Path.*** *Given two users $u, v \in V$, an influence path p_{uv} from u to v is a sequence of edges $e^{(1)}, e^{(2)}, ..., e^{(m)}$ such that $e^{(1)} = (u, v_1), e^{(j)} = (v_{j-1}, v_j)$ for all $j = 2, 3, ..., m-1$, and $e^{(m)} = (v_{m-1}, v)$. Let $\mathcal{P}_{uv} = \{p_{(uv)i}, i = 1, 2, ..., N\}$ denote the set of all such paths from u to v.*

For simplicity, we omit all the subscript (u, v) of the variables as we only talk about the influence from u to v in the whole paper.

Then we can derive the cascaded indirect influence from u to v based on IC Model. At time t, given u is activated, the influence probability $I_t(p_i)$ that v is activated through a path $p_i \in \mathcal{P}$ is

$$I_t(p_i) = \prod_{e \in p_i} w_{e,t} \quad (1)$$

We simplify the IC model by supposing each path has the chance to activate v independently, then the influence

probability I_t that v is activated by u indirectly is

$$I_t = 1 - \prod_{i=0}^{N} (1 - I_t(p_i)) = \sum_{i=0}^{N} I_t(p_i) + o(I_t(p_i)) \quad (2)$$

Omit the high-order terms of $I_t(p_i)$ and take the top-k terms of the first-order $I_t(p_i)$ as the cascaded indirect influence from u to v.

Definition 3. ***Cascaded Indirect Influence.*** *The cascaded indirect influence from u to v is defined as the sum of the top k influence score among all the paths in $\mathcal{P}$,*

$$I_t = \max_{Q \subset \mathcal{P}} \sum_{p_i \in Q} I_t(p_i) \quad s.t.\ |Q| = k \quad (3)$$

For simplicity, we use the term "indirect influence" instead of "cascaded indirect influence" in the following parts of this paper.

B. Partial Monitoring Setting

As the number of the intermediate paths bridging two users are exponentially large, learning indirect influence from all the paths is intractable. Thus we formulate the problem in a more realistic setting where supposing we could only be able to access limited number of influence paths. This problem can be viewed as a partial monitoring game.

We first define the estimated indirect influence given a decision strategy.

$$\hat{I}_t(\mathcal{D}_t) = \sum_{p_i \in \mathcal{D}_t} I_t(p_i) \quad (4)$$

where $\mathcal{D}_t \subset \mathcal{P}, |\mathcal{D}_t| = k$ is a set of k paths chosen by a decision strategy at time t.

In the partial monitoring games, a strategy is usually measured in terms of *regret*, which is here defined as the difference between the estimated indirect influence and the real indirect influence. In general, the regret grows with the game rounds T. If the regret is sublinear of T, the strategy is said to be Hanna consistent, which means that the strategy's average per-round regret will approach to the best strategy in hindsight[6]. Here we define the normalized regret,

$$\frac{1}{T}(\max_{Q \subset \mathcal{P}} \sum_{p_i \in Q} \sum_{t=1}^{T} I_t(p_i) - \sum_{t=1}^{T} \hat{I}_t(\mathcal{D}_t)) \quad \text{s.t. } |Q| = k \quad (5)$$

Then we could define the task of learning indirect influence as a problem minimizing the normalized regret.

Problem 1. *Given the weighted DAG $G = (V, E, W_t)$, at each time step t, the decision maker is asked to choose k paths from u to v to calculate the estimated indirect influence $\hat{I}_t$ where the the pairwise direct influence $w_{e,t}$ is visible to the decision maker only if e is included in the paths. The goal is to minimize the average regret between the estimated indirect influence and the best decision over T rounds*

$$\min_{decision} \frac{1}{T}(\max_{Q \subset \mathcal{P}} \sum_{p_i \in Q} \sum_{t=1}^{T} I_t(p_i) - \sum_{t=1}^{T} \hat{I}_t(\mathcal{D}_t)) \quad s.t.\ |Q| = k \quad (6)$$

III. The Proposed Algorithms

A. Exploration-Exploitation Strategy

In Section 2, we formulated the problem of learning indirect influence in large partial monitoring social networks, where the goal is to choose top k influence paths between two user nodes in each time stamp in order to minimize the regret of the estimated indirect influence. In this section, we encode this problem specifically into the multi-armed bandit setting, which is a special case of partial monitoring games.

In traditional bandit models, the player is presented with a set of N actions. In each round, the player chooses an action out of them. The environment assigns a gain to each action and then the player suffers a regret between the chosen action and the best action but gains of the not chosen actions not chosen are invisible to the player.

Although sharing the need of exploration-exploitation balance with bandit models, the problem in this paper, which has networked data and requires more than one action at each time, could not fit the traditional bandit model directly.

Therefore, we propose an algorithm called E-EXP3 based on the EXP3 (Exponential-weight algorithm for Exploration and Exploitation) algorithm[5] and CMAB (Combinatorial Multi-Arm Bandit) problem model [10], which can achieve the bound $O(\sqrt{T})$ of cumulative regret over T rounds.

Influence Normalization. Due to technical reasons, we need to first transform the formulation of influence path from production form to summation form. Also, we need to transform the influence $w_{e,t}$ into a unit gain form $g_{e,t}$ in order to facilitate the following theoretical analysis.

Let $l_{e,t} = -log(\frac{w_{e,t}}{\max_{e,t} w_{e,t}})$ and define

$$g_{e,t} = 1 - \frac{l_{e,t}}{\max_{e,t} l_{e,t}}$$

It's easy to check that $0 \leq g_{e,t} \leq 1$.

Subsequently we could write the influence on path p_i in the gain form

$$g_{i,t} = \sum_{e \in p_i} g_{e,t}$$

We further introduce the notations denoting the cumulative gains ($G_{e,T}$ and $G_{i,T}$) of the edges and paths,

$$G_{e,T} = \sum_{t=1}^{T} g_{e,t},\ G_{i,T} = \sum_{t=1}^{T} g_{i,t}$$

Consequently, the normalized regret in terms of gain becomes,

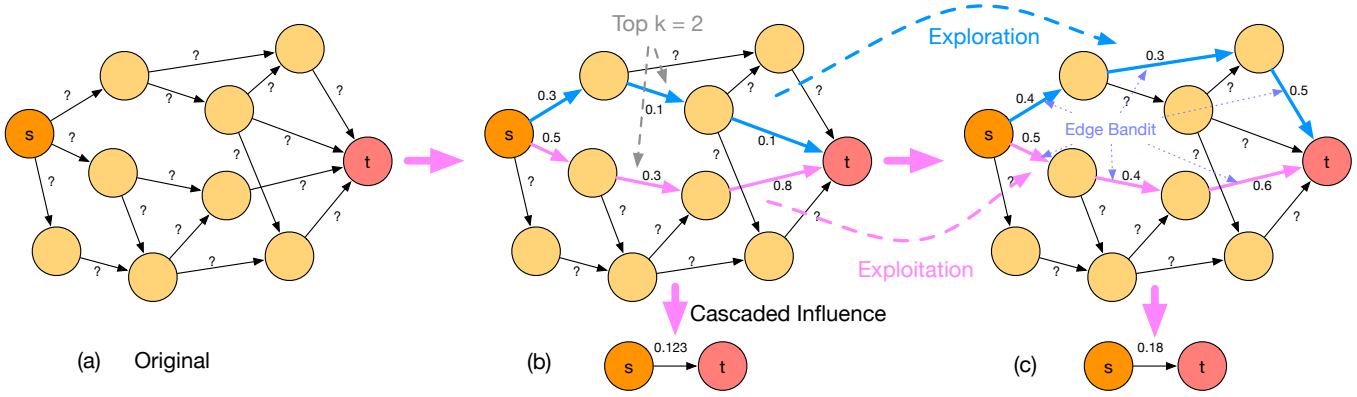


Figure 2. An example explaining how E-EXP3 works. (a) At the original time the algorithm knows nothing about the influence on the network. (b) The algorithm chooses k paths to get an estimated cascaded influence at each time stamp. (c) At next time stamp, the algorithm exploits the historical data as well as randomly explores some unknown paths to get better estimates on the cascaded influence (top k paths).

$$\frac{1}{T}(\max_{Q\subset\mathcal{P}} \sum_{p_i\in Q} G_{i,T} - \sum_{t=1}^{T} \sum_{p_i\in D_t} g_{i,t}) \quad \text{s.t. } |Q| = k \tag{7}$$

Combinatorial Path Bandit. Combinatorial multi-armed bandit (CMAB) problems are proposed in recent years, where simple arms in the traditional bandit model compose super arms. CMAB setting matches our problem of choosing top-k influence paths. Suppose there are d paths from s to t, we can consider each path as a simple arm while any path set consisting of k paths treated as a super arm. In each round, a super arm is chosen to play while only the gain of each related simple arm is revealed.

A simple way to choose a super arm at each round is to sample it from the distribution over all of these super arms, and the bound of regret of this basic CMAB strategy can be proved to be $\sqrt{T}$ over T rounds [3]. Nonetheless, This method is impractical. The number of the super arms may increase exponentially due to combinatorial explosion. Thus the algorithm may behave no better than a uniform random method unless running exponential numbers of time steps. To address this problem, we adopt a heuristic greedy strategy, where we sample k paths (simple arms) from the path distribution to get the super arm at each round. This strategy performs well in the the experiments.

There are many bandit algorithms could be instantiated under the CMAB framework. A UCB (Upper Confidence Bound) Type instance is given by [10]. However, UCB type algorithms acquire invariant statistic assumptions on data during the game. In this paper, we adopt the EXP3 type algorithm, which is designed for non-stochastic problem settings without any statistic assumptions on data.

Edge Bandit. Once we consider each path as a simple bandit, the mutual independence among these paths is set as a hypothesis. This assumption will lose the network information. Different paths may share common edges thus provide some information about paths even that are not chosen. Therefore, using edges as bandits will reduce the cost of exploration, just as [15] did to address the path planning problem.

Specifically, [15] introduces a concept of path cover set $\mathcal{C}$, which is a set of that for each edge $e \in E$, there is a path $p_i \in \rfloor$ such that $e \in p_i$. As one can always find the set $\mathcal{C}$ such that $|\mathcal{C}| \leq |\mathcal{P}|$, the exploration cost could be reduced by merely exploring the paths in $\mathcal{C}$ instead of $\mathcal{P}$.

This property could also be applied to our problem. We estimate the gains for each edge instead of each path. The estimated gains of paths are calculated from the estimated gains of the edges they consist of.

Summarizing the analysis above, we propose the algorithm E-EXP3 using a greedy CMAB model, based on the classical Exponential Weight algorithm for Exploration and Exploitation (aka EXP3) in [4]. EXP3 ensures exploration using a mixing term in the sampling distribution, which is usually derived from a uniform distribution over global sampling set. Here, we use a uniform distribution over the path cover set $\mathcal{C}$ instead of the full path $\mathcal{P}$.

Figure 2 shows an example of how E-EXP3 runs. Suppose there are d paths from the user s to the user t, at each round we draw k paths from the distribution over these paths and observe the real-time gain of each edge included in these chosen paths. This partial observed information refresh our knowledge about these paths' distribution in exponential way. At the next round, we are more likely to choose the paths whose cumulative gains were large in the history, which is called the exploitation. However, we will also choose some poorly recorded paths at certain probabilities as an exploration. In general, E-EXP3 achieves a promising trade-off between exploitation and exploration.

The details of E-EXP3 is shown in Algorithm 1.

B. Theoretical Analysis on Regret Bound

In this section we provide theoretical analysis that the expected cumulative regret of E-EXP3 is sublinear of round

Algorithm 1: E-EXP3

Input : The edge set E, The path set $\mathcal{P}$, initialize $w_{e,0} = 1$ for each $e \in E$, $\overline{w}_{i,0} = 1$ for each $i \in \mathcal{P}$, normalization factor $\overline{W}_0 = |\mathcal{P}|$, mixing coefficient $\gamma > 0$, learning rate $\eta > 0$

Output: The set of k paths $\mathcal{D}_T$ chosen at the time T

```
t ← 1
while t ≤ T do
    foreach i ∈ P do
        if i ∈ C then
            p_{i,t} ← (1 − γ) w̄_{i,t−1} / W̄_{t−1} + γ/|C|
        else
            p_{i,t} ← (1 − γ) w̄_{i,t−1} / W̄_{t−1}
    foreach e ∈ E do
        q_{e,t} ← Σ_{i:e∈i} p_{i,t}
    D_t ← Sample(p_t, k)
    Observe g_{e,t} of the edges included in D_t
    foreach e ∈ i ∈ D_t do
        w_{e,t} ← w_{e,t−1} e^{η g_{e,t}/q_{e,t}}
    foreach i ∈ P do
        w̄_{i,t} ← Π_{e∈i} e^{η g_{e,t}/q_{e,t}}
    W̄_t ← Σ_{i∈P} w̄_{i,t}
    t ← t + 1
```

T and has a bound of $O(\sqrt{T})$.

We use the variable with a prime to denote the corresponding estimate variable, e.g., $g^{'}_{e,t}$ denotes the estimate of $g_{e,t}$. We use i instead of p_i to represent a path p_i to avoid notation confusions between the probability and the path.

Here we prove that, in the case when $k = 1$ ($|\mathcal{D}_t| = 1$), $|\mathcal{P}| = N$ and the maximum length of all paths is K, we have the following theorem,

Theorem 1. *For* $\gamma = \sqrt{\frac{|\mathcal{C}| \ln N}{(e-1)T}}$ *and* $\eta = \frac{\gamma}{K|\mathcal{C}|}$

$$\max_{i \in \mathcal{P}} G_{i,T} - \mathbb{E}\left[\hat{G}_T\right] \leq 2K\sqrt{(e-1)T|\mathcal{C}| \ln N} \tag{8}$$

where

$$\hat{G}_T = \sum_{t=1}^{T} g_{\mathcal{D}_t,t}$$

Proof. In general, the bound of the expected cumulative regret is rooted in the relationship between the gain estimate and the real gain as well as the *boundedness* of the gain estimate, which can be represented in the facts showing in Equation (9) and Inequality (10), (11),

$$\sum_{i \in \mathcal{P}} p_{i,t} g^{'}_{i,t} = g_{\mathcal{D}_t,t} \tag{9}$$

$$g^{'}_{e,t} \leq 1/q_{e,t} \leq |C|/\gamma \tag{10}$$

$$\sum_{i \in \mathcal{P}} p_{i,t} g^{'2}_{i,t} \leq K \sum_{e \in E} g^{'}_{e,t} \tag{11}$$

These facts can be easily derived from the definitions and the details can be found in the Appendix of a longer version of our paper[3].

Based on these facts, a pair of lower bound and upper bound of the exponential weights relate the gain of the best bandit in hindsight and the expected gain of the strategy. As a result, the regret between these two gains is bounded.

We first investigate the bounds of the quantity $\ln \frac{\overline{W}_T}{\overline{W}_0}$. We can easily obtain that, for any $i \in \mathcal{P}$, a lower bound is,

$$\ln \frac{\overline{W}_T}{\overline{W}_0} = \ln \sum_{i \in \mathcal{P}} e^{\eta G^{'}_{i,T}} - \ln N \geq \eta \max_{i \in \mathcal{P}} G^{'}_{i,T} - \ln N \tag{12}$$

On the other hand, we can use the fact that $e^x \leq 1 + x + (e-2)x^2$ for all $x \leq 1$ to obtain the upper bound. Due to the page limits we give the upper bound of $\ln \frac{\overline{W}_T}{\overline{W}_0}$ as a lemma and leave the proof to the Appendix.

Lemma 2. *As long as* $\eta g^{'}_{i,t} \leq 1$,

$$\ln \frac{\overline{W}_T}{\overline{W}_0} \leq \frac{\eta}{1-\gamma} \hat{G}_T + \frac{(e-2)\eta^2 K|\mathcal{C}|}{1-\gamma} \max_{i \in \mathcal{P}} G^{'}_{i,t} \tag{13}$$

Since we set $\eta = \gamma/K|\mathcal{C}|$ in E-EXP3 algorithm, we can get the inequality $\eta g^{'}_{i,t} \leq 1$. However, the magnitude of η calculated for the large scale data is much smaller than 1, so inequality $\eta g^{'}_{i,t} \leq 1$ can be satisfied easily in reality. Moreover, since the meaning of η in our algorithm is learning rate, we expect that the value of η is as large as possible in the limitation of $\eta g^{'}_{i,t} \leq 1$, so as to accelerate the speed of convergence of the algorithm. Therefore, we can adjust the value of η if necessary.

Combining the upper bound with the lower bound (12),we obtain the inequality below

$$\begin{aligned} \hat{G}_T \geq & (1 - \gamma - (e-2)\eta K|\mathcal{C}|) \max_{i \in \mathcal{P}} G^{'}_{i,T} \\ & - \frac{1-\gamma}{\eta} \ln N \\ \geq & (1 - (e-1)\gamma) \max_{i \in \mathcal{P}} G^{'}_{i,T} \\ & - \frac{K|\mathcal{C}| \ln N}{\gamma} \end{aligned} \tag{14}$$

Hence,

$$\max_{i \in \mathcal{P}} G^{'}_{i,T} - \hat{G}_T \leq (e-1)\gamma \max_{i \in \mathcal{P}} G^{'}_{i,T} + \frac{K|\mathcal{C}| \ln N}{\gamma} \tag{15}$$

Take the expectation on both sides and we can get the normalized regret,

$$\max_{i \in \mathcal{P}} G_{i,T} - \mathbb{E}\left[\hat{G}_T\right] \leq (e-1)\gamma \max_{i \in \mathcal{P}} G_{i,T} + \frac{K|\mathcal{C}| \ln N}{\gamma} \tag{16}$$

[3]http://www.jiaqima.me/papers/learning-cascaded-influence.pdf

Algorithm 2: Preprocessing Schedule of RE-EXP3

Input : Preprocessing Round T_p γ, K, $|\mathcal{C}|$
Output: η

$\eta \leftarrow \gamma / K|\mathcal{C}|$
$\mathcal{G} \leftarrow \emptyset$
foreach t *in* $range(T_p)$ **do**
  Choose $\mathcal{D}_t$ with E-EXP3
  $\mathcal{G} \leftarrow \mathcal{G} \cup \{g_{i,t} : i \in \mathcal{D}_t\}$
$\eta \leftarrow \eta \times \min\{\frac{1}{mean(\mathcal{G})+3var(\mathcal{G})}, 1\}$

As $\max_{i\in\mathcal{P}} G_{i,T} \leq KT$, we have

$$\max_{i\in\mathcal{P}} G_{i,T} - \mathbb{E}\left[\hat{G}_T\right] \leq 2K\sqrt{(e-1)T|\mathcal{C}|\ln N} \tag{17}$$

holds for

$$\gamma = \sqrt{\frac{|\mathcal{C}|\ln N}{(e-1)T}}$$

$$\eta = \frac{\gamma}{K|\mathcal{C}|} = \frac{1}{K}\sqrt{\frac{\ln N}{(e-1)|\mathcal{C}|T}}$$

Therefore, the regret bound obtained by our algorithm is sublinear and the learner can approach the performance of the optimal action. □

C. Relaxation of E-EXP3

Notice that we set $\eta = \gamma/K|\mathcal{C}|$ in E-EXP3 algorithm so that we can get the inequality $\eta g'_{i,t} \leq 1$.

However, the magnitude of η for the large scale data is much smaller than 1, so inequality $\eta g'_{i,t} \leq 1$ can be satisfied easily in practice. Moreover, since the meaning of η in our algorithm is learning rate, we expect that the value of η is as large as possible in the limitation of $\eta g'_{i,t} \leq 1$, so as to accelerate the speed of convergence of the algorithm.

Therefore, we propose RE-EXP3 as a relaxation version of E-EXP3 algorithm to improve the performance on the real data.

Specifically, we add a preprocessing schedule automatically estimating an appropriate η for the dataset. The preprocessing schedule is described in Algorithm 2

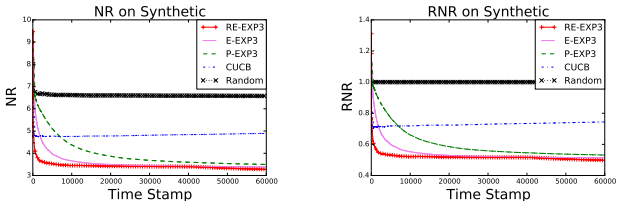
IV. Experiments

In this section, we conduct various experiments to evaluate the proposed methods for learning indirect influence.

A. Experimental Setup

Datasets. We evaluate the proposed method on three different networks: Synthetic, Weibo and AMiner.

1) Synthetic: Synthetic data includes 2000 vertexes, 5000 edges and 3000 unique paths. The graph is randomly generated and the edges are randomly split into two classes. At each time stamp, the weight of the edge from the first class is uniformly drawn from $[0, 0.3]$ while that from the second class is drawn from $[0.6, 1]$. This setting is based on the assumption that the weights of the edges are not uniformly distributed, which distinguishes the bandit algorithms from a random pick strategy. This assumption is realistic as the real world data usually has a skewed distribution. 60,000 time stamps are generated in this way.



(a) NR (b) RNR

Figure 3. Normalized Regret on Synthetic Data

2) Weibo[30][4]: This dataset comes from Sina Weibo[5], the most popular Twitter-like microblogging service in China, and consists of over 1,776,950 users, 308,739,489 "following" relationships and 23,755,810 retweets. The dataset is split into 100 time stamps. The directed edges on Weibo are defined as the following relationships. The weight associated with each edge is the intensity that a user retweeting another user in each time stamp.

3) AMiner[27][6]: This dataset comes from AMiner.org[7], and contains 231,728 papers, 269,508 authors and 347,735 citation relationships. From the original citation data, we extracted a weighted citation graph from 1988 to 2013. The directed edge from a user u to v exists if and only if v has ever cited u. The weight associated with each edge is the intensity that an author citing another author in each year.

Comparison Methods. The following methods are compared in the experiments:

1) P-EXP3: P-EXP3 is the degenerated version of E-EXP3 without making use of the network structure information.

2) E-EXP3: E-EXP3 is the proposed edge-bandit EXP3-type algorithm, which make use of the structure information to utilize heavier exploitation.

3) RE-EXP3: RE-EXP3 is the proposed relaxed E-EXP3 algorithm, which better suits the skewed data distribution and converge faster in practice.

4) CUCB[10]: CUCB (Combinatorial Upper Confidence Bound) is a UCB-type bandit algorithm dealing with combinatorial bandits.

5) Random: This method randomly selects paths to probe at each time stamp.

Evaluation Metrics. To quantitatively evaluate the proposed method, we consider the following performance metrics:

[4]Weibo data source: https://aminer.org/billboard/id:55af4227dabfae1ce3ed1235
[5]Weibo website: http://weibo.com/
[6]AMiner data source: https://aminer.org/billboard/id:56d7ef72c35f4f94cd5238a7
[7]AMiner website: https://aminer.org/

1) Normalized Regret (NR): Normalized regret is the difference of the algorithm's average gain and that of the best expert, which is defined in problem 1.

2) Relative Normalized Regret (RNR): Relative Normalized Regret is the fraction of normalized regret of an algorithm over that of the trivial baseline algorithm Random. This metric is designed to show average performance of sampled data in real networks.

3) Application Improvement: We apply the obtained indirect influence score to help the application of predicting the behavior following. Specifically, we treat the influence scores obtained by our algorithms as features, then use SVM and Logistic Regression to predict whether a pair of users have behavior following action. In this paper, we use retweeting on Weibo as the behavior following action. We compare the prediction result of influence scores with that of some user profile features, e.g., number of tweets, available in the Weibo dataset.

B. Experiments on Normalized Regrets

We evaluate the five algorithms by both Normalized Regret and Relative Normalized Regret on synthetic and by only Relative Normalized Regret on Weibo and AMiner. On Weibo and AMiner, we extracted 1500 pairs of users as well as all the paths from the source user to the target user with length no larger than 4. We run the five algorithms respectively on all the sampled networks and calculated the average Relative Normalized Regret. In all the experiments here, k is set to 10. The algorithm is allowed to run ten times at each time stamp on the two real networks to compensate the lack of data to establish more time stamps. Although the algorithms terminate soon after the $100th$ (for Synthetic and Weibo) or $44th$ (for AMiner) time stamps, T is set to $600,000$ to guarantee that $\gamma < 1$.

Figure 3 shows the NR and RNR on synthetic data and provides a visual comparison between the two metrics. The synthetic data has a longer time range and thus better characterize the convergence property of the algorithms. Figure 1 shows the average RNR on Weibo and AMiner.

Despite different types of datasets, the results show very similar trends. At the start time, all the algorithms, except CUCB, have no information and appear to be the same as Random. As the bandit algorithms gathering more information, they appear to be better and better comparing to Random. CUCB has a fast convergence due to the preprocessing schedule. However, the performance of RE-EXP3 comes to be the best and CUCB falls behind. The EXP3-type algorithms can better handle the non-stochastic situation and thus outperform CUCB after enough rounds of time. From P-EXP3, E-EXP3 to RE-EXP3, the algorithms grow to have reasonable heavier exploitation than exploration and thus make the normalized regret converge faster. E-EXP3 and RE-EXP3 should have similar performance in the end but RE-EXP3 performs better in real data thanks to its fast convergence.

Table I
APPLICATION IMPROVEMENT - LOGISTIC REGRESSION

Methods	Accuracy	Precision	Recall	F1 score
PF	0.55	0.58	0.45	0.51
P-EXP3	0.57	0.58	0.55	0.57
E-EXP3	0.59	0.61	0.55	0.58
RE-EXP3	**0.64**	**0.65**	**0.63**	**0.64**
FO	*0.70*	*0.77*	*0.60*	*0.68*

Table II
APPLICATION IMPROVEMENT - SVM

Methods	Accuracy	Precision	Recall	F1 score
PF	0.58	0.57	**0.72**	**0.63**
P-EXP3	0.56	0.58	0.53	0.55
E-EXP3	0.58	0.60	0.55	0.57
RE-EXP3	**0.63**	**0.65**	0.61	**0.63**
FO	*0.70*	*0.77*	*0.57*	*0.66*

C. Experiments on Application Improvement

We use the learned indirect influence score to help improve the application of predicting behavior following. Specifically, we use the task of predicting retweeting on Weibo.

Given a pair of users, we treat the influence scores obtained by our algorithms as well as by fully observed network as features to predict the retweeting behaviors. We compare the results with that using the user profile features available in the Weibo dataset. We use SVM and Logistic Regression respectively as the classifiers in these experiments.

As the retweeting behavior is relatively rare and sparse, we view the retweeting relationship of a pair of users exists if there is a retweeting in any time stamp. Meanwhile, we use the indirect influence scores in the last ten time stamps as the features. We use $2/3$ of the sampled data in Weibo as training data and $1/3$ as test data. We evaluate the performance of behavior prediction in terms of Accuracy, Precision, Recall and F-1 score. The results are shown in Table I and Table II, where PF represents user profile features and FO represent influence scores obtained from the fully observed network, which serves as a ground-truth.

The results show that the fully observed (FO) influence scores are the best features to predict the retweeting behaviors. The result of FO outperforms other features in terms of almost all the metrics using either SVM or Logistic Regression. While this feature is hard to obtain in real world online social networks, a partial monitoring model based algorithm RE-EXP3 is also able to achieve near performance.

V. RELATED WORK

Social Influence. Considerable work has been conducted to quantify the effect of social influence in terms of different forms. [29] tackled the data sparsity of influence propagation. [13] investigated the effect of novelty decay on influence propagation. [21] proposed a probabilistic model to quantify the external influence out-of-network sources. [7] investigated and measure the influence between two communities. [20] used the exploration-exploitation framework to handle the

online influence maximization problem while our work is more general purposed. Similar to this work, [23] and [14] derived influence measure from IC model [17] but they focused on pairwise influence instead of indirect influence. To our best knowledge, [24] is the only work studied indirect influence but we study the dynamic indirect influence in the partial monitoring setting, which is more realistic in the real applications.

Multi-armed Bandit. Multi-armed bandit was first in the nonstochastic setting by [5], who provided an exponential weight algorithm with a $O(\sqrt{T})$ bound of the cumulative regret, and [2] improved the result. Many variants of the basic multi-armed bandit problem have been developed , including combinatorial multi-armed bandit problem (CMAB) in [9], [10], [11]. Our problem belongs to the semi-bandit version of CMAB [18], [19] with nonstochastic assumption. This version is required to choose a set of arms as a super-arm at each time and can observe the feedback of each arm within this super-arm. This setting has more information than the basic bandit problem and accords with our cascaded influence learning problem. more effective algorithms are developed in [15], [16] by incorporating the bandit structure into the algorithm. We take advantages from [14] and [16] developing two effective algorithms for our problem.

VI. Conclusions

In this paper, we study a novel problem of cascade indirect influence based on the Independent Cascade model and propose two online learning algorithms (E-EXP3 and RE-EXP3) for learning the cascaded influence in the partial monitoring setting. We theoretically prove that E-EXP3 has a cumulative regret bound of $O(\sqrt{T})$. We compare the proposed algorithms with three baseline methods on both synthetic and real networks (Weibo and AMiner). Our empirical study on both real and synthetic networks shows that the proposed algorithms significantly outperform several alternative methods in terms of normalized regret. We also apply the learned cascaded influence to help behavior prediction and experiments show that our proposed algorithms can significantly help improve the accuracy of behavior prediction.

References

[1] S. Aral and D. Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, 2012.

[2] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT*, pages 217–226, 2009.

[3] J.-Y. Audibert, S. Bubeck, and G. Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2013.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[5] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[6] G. Bartók, D. P. Foster, D. Pál, A. Rakhlin, and C. Szepesvári. Partial monitoring-classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997, 2014.

[7] V. Belák, S. Lam, and C. Hayes. Cross-community influence in discussion fora. In *ICWSM*, 2012.

[8] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. I. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489:295–298, 2012.

[9] N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[10] W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pages 151–159, 2013.

[11] R. Combes, M. S. T. M. Shahi, A. Proutiere, et al. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems*, pages 2107–2115, 2015.

[12] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD'01*, pages 57–66, 2001.

[13] S. Feng, X. Chen, G. Cong, Y. Zeng, Y. M. Chee, and Y. Xiang. Influence maximization with novelty decay in social networks. In *AAAI*, pages 37–43, 2014.

[14] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM, 2010.

[15] A. Gyorgy, T. Linder, G. Lugosi, and G. Ottucsak. The on-line shortest path problem under partial monitoring. *arXiv preprint arXiv:0704.1020*, 2007.

[16] S. Kale, L. Reyzin, and R. E. Schapire. Non-stochastic bandit slate problems. In *Advances in Neural Information Processing Systems*, pages 1054–1062, 2010.

[17] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[18] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. *arXiv preprint arXiv:1410.0949*, 2014.

[19] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari. Combinatorial cascading bandits. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1450–1458. Curran Associates, Inc., 2015.

[20] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart. Online influence maximization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 645–654. ACM, 2015.

[21] S. A. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–41. ACM, 2012.

[22] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD'02*, pages 61–70, 2002.

[23] K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *Knowledge-based intelligent information and engineering systems*, pages 67–75. Springer, 2008.

[24] X. Shuai, Y. Ding, J. Busemeyer, S. Chen, Y. Sun, and J. Tang. Modeling indirect influence on twitter. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(4):20–36, 2012.

[25] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816. ACM, 2009.

[26] J. Tang, S. Wu, and J. Sun. Confluence: Conformity influence in large social networks. In *KDD'13*, pages 347–355, 2013.

[27] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998. ACM, 2008.

[28] J. Ugander, L. Backstrom, C. Marlow, and J. Kleinberg. Structural diversity in social contagion. *Proceedings of the National Academy of Sciences*, 109(16):5962–5966, 2012.

[29] R. Yan, I. E. Yen, C.-T. Li, S. Zhao, and X. Hu. Tackling the achilles heel of social networks: Influence propagation based language model smoothing. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1318–1328. International World Wide Web Conferences Steering Committee, 2015.

[30] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. Social influence locality for modeling retweeting behaviors. In *IJCAI*, volume 13, pages 2761–2767, 2013.

VII. Appendix

Proof of Equation (9)

Proof.

$$\begin{aligned}
&\sum_{i\in\mathcal{P}} p_{i,t} g_{i,t}^{'} \\
&= \sum_{i\in\mathcal{P}} p_{i,t} \sum_{e\in i} g_{e,t}^{'} \\
&= \sum_{e\in E} g_{e,t}^{'} \sum_{i\in\mathcal{P}:e\in i} p_{i,t} \\
&= \sum_{e\in E} g_{e,t}^{'} q_{e,t} \\
&= g_{\mathcal{D}_t,t}
\end{aligned}$$

□

Proof of Inequality (11)

Proof.

$$\begin{aligned}
&\sum_{i\in\mathcal{P}} p_{i,t} g_{i,t}^{'2} \\
&= \sum_{i\in\mathcal{P}} p_{i,t} (\sum_{e\in i} g_{e,t}^{'})^2 \\
&\le \sum_{i\in\mathcal{P}} p_{i,t} K \sum_{e\in i} g_{e,t}^{'2} \\
&= K \sum_{e\in E} g_{e,t}^{'2} q_{e,t} \\
&\le K \sum_{e\in E} q_{e,t} g_{e,t}^{'} \frac{1}{q_{e,t}} \\
&= K \sum_{e\in E} g_{e,t}^{'}
\end{aligned}$$

□

Proof of Lemma 2

Proof. For all $t = 1, 2, \ldots, T$ we have

$$\begin{aligned}
&\ln \frac{\overline{W}_t}{\overline{W}_{t-1}} \\
&= \ln \sum_{i\in\mathcal{P}} \frac{\overline{w}_{i,t-1}}{\overline{W}_{t-1}} e^{\eta g_{i,t}^{'}} \\
&\le \ln \sum_{i\in\mathcal{P}} \frac{\overline{w}_{i,t-1}}{\overline{W}_{t-1}} (1 + \eta g_{i,t}^{'} + (e-2)\eta^2 g_{i,t}^{'2}) \\
&\le \ln \left(1 + \sum_{i\in\mathcal{P}} \frac{\overline{w}_{i,t-1}}{\overline{W}_{t-1}} (\eta g_{i,t}^{'} + (e-2)\eta^2 g_{i,t}^{'2}) \right) \\
&\le \sum_{i\in\mathcal{P}} \frac{\overline{w}_{i,t-1}}{\overline{W}_{t-1}} (\eta g_{i,t}^{'} + (e-2)\eta^2 g_{i,t}^{'2})
\end{aligned} \tag{18}$$

where the first inequality will hold as $\eta g_{i,t}^{'} \le 1$ and the last inequality holds easily since $\ln(1+x) \le x$ follows for all $x > -1$. Notice that

$$\begin{aligned}
\eta g_{i,t}^{'} &= \eta \sum_{e\in i} g_{e,t}^{'} \le \eta \sum_{e\in i} \frac{1}{q_{e,t}} \le \frac{\eta K}{q_{e,t}} \\
&\le \frac{\eta K |\mathcal{C}|}{\gamma} = 1
\end{aligned}$$

Next, using the definition of the distribution p_i and the structure of network, we obtain that for all $t = 1, 2, \ldots$

$$\begin{aligned}
&\ln \frac{\overline{W}_t}{\overline{W}_{t-1}} \\
&\le \sum_{i\in\mathcal{P}} \frac{\overline{w}_{i,t-1}}{\overline{W}_{t-1}} (\eta g_{i,t}^{'} + (e-2)\eta^2 g_{i,t}^{'2}) \\
&\le \frac{\eta}{1-\gamma} \sum_{i\in\mathcal{P}} p_{i,t} g_{i,t}^{'} + (e-2)\frac{\eta^2}{1-\gamma} \sum_{i\in\mathcal{P}} p_{i,t} g_{i,t}^{'2} \\
&\le \frac{\eta}{1-\gamma} g_{\mathcal{D}_t,t} + (e-2)\frac{\eta^2 K}{1-\gamma} \sum_{e\in E} g_{e,t}^{'}
\end{aligned} \tag{19}$$

So we Finally get the upper bound of $\ln \frac{\overline{W}_T}{\overline{W}_0}$

$$\begin{aligned}
&\ln \frac{\overline{W}_T}{\overline{W}_0} \\
&= \ln \prod_{t=1}^{T} \frac{\overline{W}_t}{\overline{W}_{t-1}} = \sum_{t=1}^{T} \ln \frac{\overline{W}_t}{\overline{W}_{t-1}} \\
&\le \frac{\eta}{1-\gamma} \hat{G}_T + \frac{(e-2)\eta^2 K}{1-\gamma} \sum_{e\in E} G_{e,t}^{'} \\
&\le \frac{\eta}{1-\gamma} \hat{G}_T + \frac{(e-2)\eta^2 K |\mathcal{C}|}{1-\gamma} \max_{i\in\mathcal{P}} G_{i,t}^{'}
\end{aligned} \tag{20}$$

□