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  
jq.ma1216@gmail.com

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. It can affect people’s daily life in different aspects [8], [1]. The recent success of viral marketing is a strong evidence of social influence [12], [22], [17]. Much research has been conducted about social influence including pairwise influence [14], [23], topic influence [25], community influence [7], [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 little 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-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 makes 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× 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, 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 formu-

---

1http://weibo.com, the largest Chinese microblogging service.

2http://aminer.org, an author-centric search and mining system.
probability $I_t$ that $v$ is activated by $u$ indirectly is
\[ I_t = 1 - \prod_{i=0}^{N} (1 - I_t(p_i)) = \sum_{i=0}^{N} I_t(p_i) + o(I_t(p_i)) \] (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 $\mathcal{P}$,
\[ I_t = \max_{Q \subset \mathcal{P}} \sum_{p_i \in Q} I_t(p_i) \] s.t. $|Q| = k$ (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 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,
\[ \hat{I}_t(D_t) = \sum_{p_i \in D_t} I_t(p_i) \] (4)
where $D_t \subset \mathcal{P}, |D_t| = k$ 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,
\[
\frac{1}{T} \left( \max_{Q \subset \mathcal{P}} \sum_{p_i \in Q} \sum_{t=1}^{T} I_t(p_i) - \sum_{t=1}^{T} \hat{I}_t(D_t) \right) \] s.t. $|Q| = k$ (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 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
\[
\min \frac{1}{T} \left( \max_{Q \subset \mathcal{P}} \sum_{p_i \in Q} \sum_{t=1}^{T} I_t(p_i) - \sum_{t=1}^{T} \hat{I}_t(D_t) \right) \] s.t. $|Q| = k$ (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 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 the product 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,
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 experiments.

There are many bandit algorithms which 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 those are even not chosen. Therefore, using edges as bandits will reduce the cost of exploration.

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

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.

Summary. 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 $C$ instead of the full path $P$.

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, where $p_{i,t}$ and $g_{e,t}$ denote the sampling distribution of path $i$ and edge $e$ at time $t$ respectively.

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
We observe that, and Inequality (10), which can be represented in the facts showing in Equation (9) boundedness is rooted in the relationship between the gain estimate and the mixing coefficient $\gamma > 0$, learning rate $\eta > 0$.

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),

$$\sum_{i \in P} p_{i,t} g_{i,t}^2 \leq K \sum_{e \in E} g_{e,t}^2$$  \hfill (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.

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 $\ln \frac{W_T}{W_0}$. We can easily obtain that, for any $i \in \mathcal{P}$, a lower bound is,

$$\ln \frac{W_T}{W_0} = \ln \sum_{i \in \mathcal{P}} e^{\eta G_i,T} - \ln N \geq \eta \max_{i \in \mathcal{P}} G_i,T - \ln N$$  \hfill (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 $\ln \frac{W_T}{W_0}$ as a lemma and leave the proof to the Appendix.

Lemma 2. As long as $\eta g_{i,t}^2 \leq 1$,

$$\ln \frac{W_T}{W_0} \leq \frac{\eta}{1-\gamma} \hat{G}_T + \frac{(e-2)\eta^2 K|\mathcal{C}|}{1-\gamma} \max_{i \in \mathcal{P}} G_i,T$$  \hfill (13)

Since we set $\eta = \gamma / K|\mathcal{C}|$ in E-EXP3 algorithm, we can get the inequality $\eta g_{i,t}^2 \leq 1$. However, the magnitude of $\eta$ calculated for the large scale data is much smaller than 1, so inequality $\eta g_{i,t}^2 \leq 1$ can be satisfied easily in reality. Moreover, since the meaning of $\eta$ in our algorithm is learning rate, we expect that the value of $\eta$ is as large as possible in the limitation of $\eta g_{i,t}^2 \leq 1$, so as to accelerate the speed of convergence of the algorithm. Therefore, we can adjust the value of $\eta$ if necessary.

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

$$\hat{G}_T \geq (1 - \gamma - (e-2)\eta K|\mathcal{C}|) \max_{i \in \mathcal{P}} G_i,T$$

$$- \frac{1 - \gamma}{\eta} \ln N$$

$$\geq (1 - (e-1)\gamma) \max_{i \in \mathcal{P}} G_i,T$$  \hfill (14)

Hence,

$$\max_{i \in \mathcal{P}} G_i,T - \hat{G}_T \leq (e-1)\gamma \max_{i \in \mathcal{P}} G_i,T + \frac{K|\mathcal{C}| \ln N}{\gamma}$$  \hfill (15)

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

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

\footnote{http://www.jiaqima.me/papers/learning-cascaded-influence.pdf}
Algorithm 2: Preprocessing Schedule of RE-EXP3

Input: Preprocessing Round \(T_p\), \(\gamma\), \(K\), \(|C|\)

Output: \(\eta\)

1. \(\eta \leftarrow \gamma/K|C|\)
2. \(\mathcal{G} \leftarrow \emptyset\)
3. \(\text{foreach } t \text{ in range}(T_p) \text{ do}\)
   4. Choose \(D_t\) with E-EXP3
   5. \(\mathcal{G} \leftarrow \mathcal{G} \cup \{g^t_{i,t} : i \in D_t\}\)
   6. \(\eta \leftarrow \eta \times \min\left\{\frac{1}{\text{mean}(\mathcal{G}) + 3\text{var}(\mathcal{G})}, 1\right\}\)

---

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.

2) Weibo\(^4\): 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\(^6\)\(^7\): 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:

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.

---

\(^4\)Weibo data source: https://aminer.org/billboard/id:55af4227dabfa1ce3ed1235
\(^5\)Weibo website: http://weibo.com/
\(^6\)AMiner data source: https://aminer.org/billboard/id:56d7ef72c35f4f94cd5238a7
\(^7\)AMiner website: https://aminer.org/
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 stamp, \( T \) 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.

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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF</td>
<td>0.55</td>
<td>0.58</td>
<td>0.45</td>
<td>0.51</td>
</tr>
<tr>
<td>P-EXP3</td>
<td>0.67</td>
<td>0.68</td>
<td>0.55</td>
<td>0.73</td>
</tr>
<tr>
<td>E-EXP3</td>
<td>0.79</td>
<td>0.71</td>
<td>0.65</td>
<td>0.77</td>
</tr>
<tr>
<td>RE-EXP3</td>
<td>0.64</td>
<td>0.65</td>
<td>0.65</td>
<td>0.67</td>
</tr>
<tr>
<td>FO</td>
<td>0.70</td>
<td>0.77</td>
<td>0.60</td>
<td>0.68</td>
</tr>
</tbody>
</table>

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. [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 investigating such kind
of influence but they mainly discussed the existence and prototypical explanation of indirect influence without giving detailed measurements of a given pair of users. We propose the first feasible quantitative social influence measurement between users who are not connected directly.

**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**


