Learning Opinions in Social Networks

Vincent Conitzer * 1  Debmalya Panigrahi * 1  Hanrui Zhang * 1

Abstract

We study the problem of learning opinions in social networks. The learner observes the states of some sample nodes from a social network, and tries to infer the states of other nodes, based on the structure of the network. We show that sample-efficient learning is impossible when the network exhibits strong noise, and give a polynomial-time algorithm for the problem with nearly optimal sample complexity when the network is sufficiently stable.

1. Introduction

Suppose we are a social media company. A new product is about to come out, and we would like to learn whether each individual user has heard about it or not (we will refer to this as the current opinion of the user). This information may be useful for further marketing of the product, or for other purposes. To achieve this goal, we decide to run a poll, by asking each user visiting our website a few questions (i.e., inspecting the user). We hope to inspect as few users as possible, because we would rather let them engage in other activities on the site. Moreover, since we have no control over which users will visit our website, the only thing we can decide is the time interval during which we run the poll, or equivalently, the rough number of users to inspect.

If our goal is to make sure that our estimates for individual users are uniformly accurate, without further knowledge about users, we would have to inspect almost all users. However, in our role as a social media company, we have access to the social network formed by our users. In particular, we know which users are likely to be affected by which other users. This enables us to infer the opinion of some users without inspecting them at all, as long as we know the opinions of certain other users. For example, if user $u$ strongly affects user $v$, and we know that user $u$ is aware of our product, then we are quite sure that user $v$ is also aware, because most likely $u$ has told $v$ about the product. In an extreme case, suppose we know that all users share all information with each other at all times. Then inspecting only one user is enough for our purpose, because either everyone is aware of the new product, or no one is. In general, how many users we need to inspect depends heavily on the structure of the network formed by users. So, given the structure of the network, the questions we are interested in are:

- How many users do we need to inspect in order to have enough information to make an accurate estimate of the current state? In other words, what is the sample complexity of learning opinions in a given network?
- Given the opinions of the users inspected, how can we infer the opinions of other users who have not been inspected?

While we have illustrated the problem using the example of marketing a product, the same model can be used for many other purposes. For instance, we may want to learn whether users are aware of a particular political candidate, or of a particular news item. We may want to learn whether an HIV awareness campaign has reached individual homeless youth (Wilder et al., 2018). Or, rather than the spread of information, we may consider the spread of a biological or computer virus in a network, learning where it is likely to have spread (Romano et al., 2010).

1.1. Our Results

In this paper, we give (1) an asymptotically tight bound (up to a logarithmic factor) on the sample complexity of learning opinions in social networks, and (2) a polynomial time algorithm which, given the samples, outputs an approximately correct estimate of the state of the network, with high probability. Our results are summarized in the following theorem, which roughly states that there is an efficient algorithm that learns the state of the network with any desired error rate $\varepsilon$ and failure probability $\delta$ (up to the resolution of the network), by observing a nearly optimal number of labelled sample nodes only.

Theorem 1.1 (Main Result, Informal). For any given (possibly random) network, there exists an efficient algorithm, which, for any $\varepsilon \in [\varepsilon_0, 1]$ and $\delta \in [\delta_0, 1]$, given access to
Learning Opinions in Social Networks

\[ m = \tilde{O}(d/\varepsilon) \text{ iid samples,} \] with probability \( 1 - \delta \), outputs an estimate of the state of any network that is accurate for a \( 1 - \varepsilon \) fraction of the members of the network. Here, \( d \) is the expected complexity of the network, and \( \varepsilon_0 \) and \( \delta_0 \) are network-dependent constants modelling the intrinsic resolution of the network, which depends on the strength of the noise. Moreover, the above number of samples is minimum possible, up to a factor of \( O(\log(1/\varepsilon)) \).

1.2. Related Work

Most closely related to our work is the line of research on learning structures of social networks. The problem of interest there can be considered as an inverse problem of the one studied in this paper: given outcomes of some propagation procedure, the goal is to recover parameters of the network governing the propagation. Some representative results include (Liben-Nowell & Kleinberg, 2007; Goyal et al., 2010; Chierichetti et al., 2011; Gomez Rodriguez et al., 2011; Saito et al., 2011; Du et al., 2012; Guille & Hacid, 2012; Abrahao et al., 2013; Cheng et al., 2014; Daneshmand et al., 2014; Du et al., 2014; Narasimhan et al., 2015; He et al., 2016; Kalimeris et al., 2018). A similar and more recent line of work is on representation learning for information propagation (Bourigault et al., 2016; Li et al., 2017; Wang et al., 2017). By virtue of being an inverse problem, our results are not directly comparable to all these.

The study of information propagation in a network was initiated by Kempe et al. (2003). Since then, various models of information propagation have been proposed (Gruhl et al., 2004; Chen et al., 2010; 2011; Myers et al., 2012). Several research topics have drawn significant attention, such as influence maximization (Massel & Roch, 2007; Chen et al., 2009; 2010; 2011; Borgs et al., 2014; Tang et al., 2014), identification of influential nodes (Agarwal et al., 2008; Pal & Counts, 2011), and community detection (Faloutsos et al., 2004; Coscia et al., 2011).

Since the introduction of probably approximately correct (PAC) learning by Valiant (1984), a series of remarkable results have provided a rather complete picture for passive learning from observations. Following the groundbreaking Vapnik-Chervonenkis (VC) theory (Vapnik, 1993), various measures of complexity have been considered (Alon et al., 1997; Bartlett & Mendelson, 2002; Pollard, 2012; Daniely et al., 2015), based on which tighter generalization bounds have also been developed (Hanneke, 2016). While these general results are powerful, as discussed in later sections, they cannot be directly applied to the specific problem considered in this paper.

2. Preliminaries

In this section, we review relevant concepts from the areas of learning theory and social network analysis, which set up the context for our results. We then formally define the problem investigated in this paper.

2.1. Learning Theory

The problem studied in this paper is an extension to the classical problem of probably approximately correct (PAC) learning (Valiant, 1984). The problem is defined by the following parameters: a space \( X \) of data points, a distribution \( D \) over \( X \), and a hypothesis class \( \mathcal{H} \subseteq 2^X \). As a shorthand, for any \( x \in X \) and \( h \in \mathcal{H} \), let \( h(x) = \mathbb{I}(x \in h) \) denote the indicator that \( x \in h \). The goal is to find an algorithm, which, for any ground truth \( c \in \mathcal{H} \) and desired failure probability \( \delta > 0 \) and error rate \( \varepsilon > 0 \), given \( m \) iid samples \( \{(x_i, y_i)\}_{i \in [m]} \) where \( x_i \sim D \) and \( y_i = c(x_i) \) for any \( i \in [m] \), returns a hypothesis \( h \in \mathcal{H} \) such that with probability at least \( 1 - \delta \),

\[ \Pr_{x \sim D}[h(x) \neq c(x)] \leq \varepsilon. \]

Here, we are often interested in minimizing the number of samples \( m \), i.e., the sample complexity of the algorithm.

It is known that up to a logarithmic factor, the sample complexity of a PAC learning instance depends only on the VC dimension of the class, defined below (see, e.g., (Kearns & Vazirani, 1994)).

**Definition 2.1** (VC Dimension). A set \( S \) is shattered by a family of sets \( \mathcal{F} \), if for any \( T \subseteq S \), there exists \( U \in \mathcal{F} \), such that \( S \cap U = T \). The VC dimension of a hypothesis class \( \mathcal{H} \) over a space \( X \), \( \text{VC}(\mathcal{H}) \), is the size of the minimum set \( S \subseteq X \) shattered by \( \mathcal{H} \). That is,

\[ \text{VC}(\mathcal{H}) = \min\{|S| \mid S \subseteq X, \{S \cap U \mid U \in \mathcal{H}\} = 2^S \}. \]

Fixing a hypothesis class, the sample complexity of PAC learning is given by the following theorem (see, e.g., (Kearns & Vazirani, 1994)), which characterizes the number of samples \( m \) required to learn a concept with desired error rate \( \varepsilon \) and failure probability \( \delta \).

**Theorem 2.1** (VC Theorem, the Realizable Case). Fix \( X \) and \( \mathcal{H} \). For any \( D \) over \( X \), \( c \in \mathcal{H} \), \( \delta > 0 \) and \( \varepsilon > 0 \), given \( m = O((\text{VC}(\mathcal{H}) \log(1/\varepsilon) + \log(1/\delta))/\varepsilon) \) samples, with probability at least \( 1 - \delta \), any hypothesis \( h \in \mathcal{H} \) which is consistent with all the samples (i.e., \( h(x_i) = y_i \) for all \( i \in [m] \)) satisfies

\[ \Pr_{x \sim D}[h(x) \neq c(x)] \leq \varepsilon. \]

Moreover, this bound is tight in the sense that any algorithm achieving this guarantee requires \( \Omega((\text{VC}(\mathcal{H}) + \log(1/\delta))/\varepsilon) \) samples.
2.2. Social Network Analysis

We now review a few basic concepts in social network analysis, which provide the language for describing how members (henceforth nodes) of a network interact with each other. The presence of such interactions enable inference of the opinions of nodes that are not inspected. Each node has two possible opinions or states, active and inactive, encoding whether the node is aware of an idea (e.g., a new product) or not. Following conventions in social network analysis, we assume the final opinions of nodes are formed via the following information propagation process: a subset of nodes (the seed set) is active at the beginning. Over time, active nodes make other (previously inactive) nodes active, in a process governed by the propagation model. But, once a node is active, it never becomes inactive again. As a result, the propagation stops eventually, and the final opinions of nodes are the outcome of this propagation process.

Throughout the paper, we consider a general propagation model on live-edge graphs (see, e.g., Chen et al., 2010)). In the live-edge graph model, a network over nodes $V$ (where $|V| = n$) is modelled by a distribution $\mathcal{G}$ over possible realizations of the network, each of which is a simple directed graph over the same set of nodes $V$. Given the seed set $S_0 \subseteq V$, the propagation happens in the following probabilistic way: A realization $G = (V, E) \sim \mathcal{G}$ is drawn from the distribution $\mathcal{G}$. Information then propagates according to reachability between nodes in $G$, where an edge $\{u, v\} \in E$ indicates that $v$ is activated by $u$. The propagation happens in steps. At step $i > 0$, all inactive nodes that are directly reachable from an active node at the previous step become active. That is, the new set of active nodes is

$$S_i = S_{i-1} \cup \{u \mid \exists v \in S_{i-1}, s.t. (v, u) \in E\}.$$

The above procedure defines a monotone sequence of sets of active nodes $\{S_i\}_{i \leq n}$, where for any $i > 0$, $S_i \supseteq S_{i-1}$. On the other hand, since $S_i \subseteq V$ for any $i$, there exists some $i < n$, such that for any $j > i$, $S_i = S_j$. In particular, for any $i \geq n - 1$, $S_i = S_{n-1}$. In light of this, we say $S_\infty = S_{n-1}$ is the final outcome of the propagation procedure. Equivalently, one may define $S_\infty$ as the set of nodes reachable from the seed set. Note that in general, fixing the seed set $S_0$, the final outcome $S_\infty$ is a random variable depending on the realization of the network. Again, for brevity, for any set of nodes $S \subseteq V$ and node $u \in V$, let $S(u) = \mathbb{I}[u \in S]$ be the indicator that $u \in S$.

2.3. Learning Opinions in Social Networks

As illustrated in the introductory example, the problem we aim to study is roughly the following. An unknown seed set $S_0$ is chosen (possibly adversarially), and a realization $G \sim \mathcal{G}$ of the network is drawn, but remains invisible to the algorithm. Propagation then happens from $S_0$ in $G$, resulting in an outcome $S_\infty$. The algorithm observes a number of iid nodes, and learns whether each of them is in $S_\infty$. The algorithm then tries to infer $S_\infty$, aiming to guarantee error at most $\varepsilon$, with probability at least $1 - \delta$. Below we formally state the problem of learning opinions in social networks.

The problem is defined by the following parameters: a set of nodes $V$ (where $|V| = n$), a network $G$ over $V$ (in the form of a live-edge graph defined above), and a distribution $\mathcal{D}$ over $V$. Given these, the goal is to design an algorithm, which for any seed set $S_0 \subseteq V$, $\delta > 0$, and $\varepsilon > 0$, given $m$ iid labelled samples $\{(u_i, o_i)\}_{i \in [m]}$ where for any $i \in [m]$, $u_i \sim \mathcal{D}$ and

$$o_i = S_\infty(u_i) = \mathbb{I}[u_i \in S_\infty]$$

computes a hypothesis set $H \subseteq V$ of active nodes, such that with probability at least $1 - \delta$,

$$\Pr_{u \sim \mathcal{D}}[S_\infty(u) \neq H(u)] \leq \varepsilon.$$

We are interested in minimizing the sample complexity $m$.

In words, given a network modelled by a live-edge graph, the problem asks for an algorithm, which, with high probability, approximately recovers the final outcome of a propagation procedure starting from any seed set, by observing iid samples only. In particular, the algorithm is not aware of the realization of the network — this additional randomness from the propagation procedure prevents us from applying existing results from learning theory directly.

We also remark that the problem becomes trivial with $\Omega(n/\varepsilon)$ samples. In this case, one can learn the state of the network irrespective of its structure. To this end, we aim to find a parameter of the network that tightly characterizes the minimum number of samples required to recover the outcome of the propagation.

3. Warmup: Deterministic Networks

To develop some intuition, we first investigate a special case of the problem where the network is deterministic, i.e., the support of $\mathcal{G}$ is a singleton. We present a few observations, given which the problem can be solved by applying the classical VC theorem.

Let $G = (V, E)$ be the only possible realization of $\mathcal{G}$. The key observation is that the effective hypothesis class to be considered might be much smaller than $2^V$. In fact, for any $u \in V$ and $v \in V$ where $v$ is reachable from $u$, for any seed set $S_0$, it is impossible that in the final outcome of the propagation, $u \in S_\infty$ but $v \notin S_\infty$. In general, we only need to

\footnote{Recall that $S_\infty$ is the (random) final set of active nodes according to the propagation procedure starting from $S_0$, defined by $\mathcal{G}$.}
consider hypotheses where no such contradictions happen. In other words, the propagation procedure on $G$ induces a hypothesis class $H \subseteq 2^V$. In light of the VC theorem, we now consider the VC dimension of this associated hypothesis class $H$, which we define to be the VC dimension of the graph $G$. Projecting the definition of the VC dimension to graphs, we get directly the following definition.

**Definition 3.1 (VC Dimension of Directed Graphs).** The VC dimension $\text{VC}(G)$ of a given graph $G = (V, E)$ is the size of the maximum set $S \subseteq V$ of nodes, such that for any $u, v \in S$ where $u \neq v$, $u$ cannot reach $v$ in $G$.

One may show that the above definition in fact coincides with the VC dimension of the hypothesis class associated with the graph (see the appendix in the full version of the paper for a proof).

**Proposition 3.1.** Let $G$ be any directed graph, and $H$ be its associated hypothesis class defined above. Then $\text{VC}(G) = \text{VC}(H)$.

We make a few remarks regarding the above definition.

- Restricted to deterministic networks, one can always assume without loss of generality that the unique realization $G$ is acyclic. This is because fixing any seed set, for any strongly connected component in $G$, either all nodes in the component are in $S_\infty$, or no node is. One can therefore effectively contract any such component into a single node. Moreover, this contraction does not affect the VC dimension $\text{VC}(G)$ of $G$.

- Given the observation that in our problem, any graph is effectively acyclic, the above definition of the VC dimension of a graph coincides precisely with the concept of the width of a graph or a partially ordered set. As a result, the VC dimension of any graph can be computed in polynomial time.

Based on the above observations, as long as the network $G$ is deterministic, we may apply the classical VC theory to the problem of learning opinions in social networks. The VC theorem then directly gives the following.

**Theorem 3.1 (Learning Opinions in Social Networks, the Deterministic Case).** Fix a deterministic network $G = (V, E)$. For any seed set $S_0$, $D$ over $V$, $\delta > 0$ and $\varepsilon > 0$, given $m = O((\text{VC}(G) \log(1/\varepsilon) + \log(1/\delta))/\varepsilon)$ samples, with probability at least $1 - \delta$, any hypothesis set $H \in H$ in the associated hypothesis class that is consistent with all the samples satisfies

$$\Pr_{u \sim D}[S_\infty(u) \neq H(u)] \leq \varepsilon.$$ 

Moreover, this result is tight in that any algorithm providing this guarantee requires $\Omega((\text{VC}(G) + \log(1/\delta))/\varepsilon)$ samples.

To compute a hypothesis $H$ consistent with the samples, one can simply take all sampled nodes that are active (denoted $A$), and let $H$ be the set of all nodes reachable from $A$ in $G$. It is straightforward to check that $H$ is in fact consistent with $G$ and the sample nodes. This gives an algorithm that is both sample efficient and computationally efficient.

### 4. The General Case: Random Networks

In the previous section, we demonstrated an interesting connection between the problem of learning opinions in social networks and the classical VC theory for passive learning. In particular, we identified a network dependent parameter, namely the width, which coincides with the VC dimension of the induced hypothesis class, and therefore dictates the sample complexity of learning in a given deterministic network. We now try to generalize these results to the general case of the problem.

In general, the network $G$ we consider may exhibit randomness. In such cases, it is no longer possible to apply the VC theorem directly to obtain learning algorithms with a non-trivial sample complexity bound. In fact, the VC dimension $\text{VC}(G)$ of the realization $G \sim G$ itself is random, which makes it infeasible for measuring the complexity of $G$. Still, one may hope for a bound that depends on the expectation, or worst case upper bound, of $\text{VC}(G)$ when $G \sim G$. In this section, we first show that this is, in general, not possible. We then identify a condition under which sample efficient learning is possible, and give a nearly optimal algorithm when the condition holds.

#### 4.1. Obstacles to Learning in General Networks

We first discuss a key difference between learning in deterministic and random networks, which makes it impossible to design sample efficient learning algorithms that work for arbitrary random networks. To be specific, recall that the learning algorithm has no access to the actual realization of the random network. As a result, the outcome of the propagation itself, as well as the labelled sample nodes, carries information about both the seed set and the actual realization, where the latter in some sense introduces noise to the problem. Informally speaking, this becomes problematic when the randomness of the network overwhelms the amount of information encoded in the seed set. In such cases, one would have to spend most of the samples to recover the noise, which has no intrinsic structure to be exploited.

To make this concrete, we show constructively that in certain pathological cases where $G$ is poorly concentrated, even if the VC dimension of the realization, $\text{VC}(G)$, is always 1, we may still need $\Omega(n)$ samples to learn the label of a single fixed node with constant probability.
Consider the following construction of $G$. To generate $G \sim G$, we first generate a uniform random permutation $\sigma : V \rightarrow [v]$ of $V$, where $\sigma(u)$ is the rank of $u \in V$. $G = (V, E)$ is then defined by the following: for any $u, v \in V$,

$$(u, v) \in E \iff \sigma(u) + 1 = \sigma(v).$$

That is, $G$ is a chain formed by connecting consecutive nodes in the permutation $\sigma$. Clearly $G$ always has VC dimension 1.

Now fix the seed set $S_0$ to be the singleton set containing a fixed node $u_0$, and let $D$ be the uniform distribution over $V$. Observe that in the final outcome of the propagation, for any node $u \neq u_0$,

$$u \in S_\infty \iff \sigma(u_0) \leq \sigma(u),$$

where $\sigma(\cdot)$ is the permutation defining $G$. We can show the following lower bound (proof in appendix): 

**Proposition 4.1.** By observing iid labelled samples, for any $u \neq u_0$, an algorithm needs $\Omega(n)$ samples to recover $S_\infty(u)$ with probability $9/10$.

In other words, one needs to inspect, in expectation, a constant fraction of all nodes in order to recover anything nontrivial about the network.

### 4.2. Stability of Networks

The above discussion rules out the possibility of any nontrivial sample complexity bound without further assumptions on the network. However, the difficulty of learning in the above network comes solely from the fact that $G$ does not concentrate for any reasonable notion of concentration. Real-world networks, on the other hand, rarely exhibit such extreme randomness. To this end, we identify a mild condition that captures the intrinsic resolution of random networks, and give an algorithm that learns opinions in social networks up to this resolution.

**Definition 4.1 (($\varepsilon_0, \delta_0$)-Stable Networks).** A network $G$ is ($\varepsilon_0, \delta_0$)-stable with respect to a distribution $D$ defined over the vertex set $V$, if for any seed set $S_0$, the final outcome $S_\infty$ of the propagation satisfies, with probability at least $1 - \delta_0$,

$$\Pr_{u \sim D, S_\infty} [S_\infty(u) \neq S'_\infty(u)] \leq \varepsilon_0,$$

where $S'_\infty$ is an independent copy of $S_\infty$.

In words, a network is ($\varepsilon_0, \delta_0$)-stable if the outcome $S_\infty$ of the propagation from $S_0$ does not lie in the “tail” (which means $S_\infty$ is on average close to an independent copy of itself) with probability at least $1 - \delta_0$. To be more precise, if this happens, then the expected distance from the outcome $S_\infty$ to another identically distributed outcome $S'_\infty$ is at most $\varepsilon_0$. This is rather mild, since it does not even require the expected distance between two independent outcomes of propagation to have small second moment, even conditioning on one of them being not in the tail. Technically, with a small second moment, one would expect good concentration behavior of $S_\infty$, making it practically deterministic. This is not the case for the notion of ($\varepsilon_0, \delta_0$)-stability. As we will see, our algorithm is independent of any concentration property of $S_\infty$.

We also remark that the notion of ($\varepsilon_0, \delta_0$)-stability may not be the only nontrivial condition that permits efficient learning. On the other hand, it does provide a way of parameterizing random networks, which allows us to derive almost matching upper and lower bounds on the sample complexity. To this end, ($\varepsilon_0, \delta_0$)-stability appears to be one of the most natural notions that capture the intrinsic resolution of random networks, and may even be generalized to other learning problems.

### 4.3. Efficient Algorithm for Stable Networks

Despite the mildness of the stability condition, we show that it enables an efficient learning algorithm, Algorithm 1, up to the intrinsic resolution of the network.

Algorithm 1 works in the following way. Without access to the actual realization of the network, the algorithm draws a number of iid sample realizations from $G$. It then tries to fit the labelled sample nodes on each of these sample realizations, by finding the hypothesis set that is consistent with the realization, and moreover minimizes the number of sample nodes whose labels are inconsistent with the hypothesis. In other words, the algorithm finds the empirical risk minimizer for each of the sample realizations. Note that here, the best hypothesis set for a sample realization may not be perfectly consistent with the sample nodes — this is because the sample nodes are part of the outcome of the actual propagation process on the realized network, which is likely different from the sample realization. With these empirical risk minimizers, the algorithm then computes and outputs the node-wise majority vote, i.e., a node is in the output hypothesis, iff it is in more than half of these empirical risk minimizers.

To efficiently compute empirical risk minimizers on sample realizations, Algorithm 1 calls a subroutine, Algorithm 2. Algorithm 2 treats the empirical risk minimization problem as a constrained combinatorial optimization problem. Concretely, it models the problem in the following way: each sample node with label 1 has weight 1, and each sample node with label 0 has weight $-1$. The algorithm tries to find a subset of all nodes with maximum weight, subject to the constraint that for any edge $(u, v) \in E$, if $u$ is in the subset, then $v$ must be in the subset. This problem turns out to be solvable by further formulating the above as a min-cut problem, which can be solved in polynomial time.
We now analyze the efficiency and correctness of Algorithm 1.

**Theorem 4.1** (Learning Opinions in Social Networks, the General Case). Let $\mathcal{G}$ be a network that is $(\varepsilon_0, \delta_0)$-stable. For any seed set $S_0$, $\mathcal{D}$ over $V$, $\delta \geq C_1 \cdot \delta_0$ and $\varepsilon \geq C_2 \cdot \varepsilon_0$, given

$$m = O \left( \frac{\mathbb{E}_{G \sim \mathcal{G}}[\text{VC}(G)] \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon} \right)$$

samples, with probability at least $1 - \delta$, Algorithm 1 outputs in time $\text{poly}(n, 1/\varepsilon, \log(1/\delta))$ a hypothesis set $H \subseteq V$ that satisfies

$$\Pr_{\mathcal{D}}[S_\infty(u) \neq H(u)] \leq \varepsilon.$$  

$C_1 > 0$ and $C_2 > 0$ are absolute constants.

Before proving the theorem, we remark that the above sample complexity bound is optimal up to a factor of $O(\log(1/\varepsilon))$. This can be seen by restricting $\mathcal{G}$ to be deterministic, and comparing the sample complexity bound against the lower bound in Theorem 3.1.

**Proof of Theorem 4.1.** First observe that Algorithm 1 runs in time $\text{poly}(n, 1/\varepsilon, \log(1/\delta))$. This is because the algorithm takes $m = \text{poly}(n, 1/\varepsilon, \log(1/\delta))$ labelled sample nodes, and performs empirical risk minimization with these nodes on $\emptyset = \text{poly}(n, 1/\varepsilon, \log(1/\delta))$ sample realizations of the network. In each of these, the algorithm calls Algorithm 2 on an input of polynomial size. The latter simply computes a max flow in time polynomial in the input size, which in turn is $\text{poly}(n, 1/\varepsilon, \log(1/\delta))$. Finally, the algorithm scans through the empirical risk minimizers and the nodes, and computes the majority vote in polynomial time.

Now we show Algorithm 1 in fact outputs a hypothesis with the desired error rate and failure probability. We first show that Algorithm 2 does find empirical risk minimizers.

**Lemma 4.1.** Given a realization $G = (V, E)$ and labelled sample nodes $\{(u_i, o_i)\}_{i \in [m]}$, Algorithm 2 finds a set $H \subseteq V$ consistent with $G$ minimizing the empirical risk, i.e.,

$$\sum_{i \in [m]} I[H(u_i) \neq o_i].$$

**Proof.** Let $w : V \rightarrow \{-1, 0, 1\}$ be a weight function over nodes. We construct $w$ such that for any $u \in V$,

$$w(u) = \sum_{i \in [m]} I[u = u_i] \cdot (2o_i - 1).$$

Observe that minimizing the empirical risk is equivalent to finding a set $H \subseteq V$ consistent with $G$, maximizing the total weight $\sum_{u \in H} w(u)$. In fact,

$$\sum_{u \in H} w(u) = \sum_{u \in H} \sum_{i \in [m]} I[u = u_i] \cdot (2o_i - 1)$$

$$= \sum_{i \in [m]} (2o_i - 1) \sum_{u \in H} I[u = u_i]$$

$$= \sum_{i \in [m]} (2o_i - 1) \cdot H(u_i)$$

$$= \sum_{i \in [m]} (o_i - I[H(u_i) \neq o_i])$$

$$= \sum_{i \in [m]} o_i - \sum_{i \in [m]} I[H(u_i) \neq o_i].$$
We further argue that the latter problem can be solved precisely by computing the min-cut in the way performed by Algorithm 2. Observe that in any finite capacity $s$-$t$ cut in $G', \forall u \in V$, when this happens, we have to cut all edges from $u$ to each of which corresponds to a sample node $(u_i, o_i)$ where $u_i = u$ and $o_i = 0$. By cutting these edges, we incur cost

$$
\sum_{i \in [m]} \|u_i = u\|(H(u) - o_i).
$$

When $u \in H$, we have to cut all edges from $u$ to $t$, each of which corresponds to a sample node $(u_i, o_i)$ where $u_i = u$ and $o_i = 1$. By cutting these edges, we incur cost

$$
\sum_{i \in [m]} \|u_i = u\|(o_i - H(u)).
$$

So the total cost we incur can be written as

$$
\sum_{u \in V} \sum_{i \in [m]} \|u_i = u\|(H(u) - o_i) \cdot (2H(u) - 1)
= \sum_{u \in V} \sum_{i \in [m]} \|u_i = u\|(2H(u)^2 - H(u) - 2H(u)o_i + o_i)
= \sum_{u \in V} \sum_{i \in [m]} \|u_i = u\|(H(u)(1 - 2o_i) + o_i)
= \sum_{u \in V} -w(u) + \sum_{i \in [m]} o_i.
$$

Observe that the min-cut minimizes the above cost, which is equivalent to maximizing $\sum_{u \in H} w(u)$. This concludes the proof of the lemma.

Having proved that Algorithm 2 in fact computes an empirical risk minimizer, we can then apply the following lemma about the error rate of empirical risk minimizers.

**Lemma 4.2.** Fix a space $X$ and a hypothesis class $\mathcal{H} \subseteq 2^X$. Fix any distribution $\mathcal{D}$ over $X$, $\epsilon \leq X$, $\delta' > 0$ and $\epsilon' > 0$. Moreover, suppose there is a hypothesis $h^* \in \mathcal{H}$ such that $Pr_{x \sim \mathcal{D}}[h^*(x) \neq c(x)] \leq \epsilon'/2$. Given at least $m' = O((|\mathcal{H}| \log(1/\epsilon') + \log(1/\delta'))/\epsilon')$ samples (denoted $\{(x_i, y_i)\}_{i \in [m']}$, with probability at least $1 - \delta'$, any empirical risk minimizer $h \in \mathcal{H}$, i.e., any $h$ satisfying

$$
Pr_{x \sim \mathcal{D}}[h(x) \neq c(x)] \leq \epsilon'.
$$

The lemma is a straightforward adaptation of the classical VC theorem, and can be proved by modifying the original proof, replacing the additive concentration bounds by their multiplicative versions. We defer the proof to the appendix in the full version of the paper.

We now bound the error rate and failure probability of Algorithm 1. Let $S_{\infty}$ be the outcome of the actual propagation from $S_0$ on the actual realization (denoted $G^*$) of the network $G$. Moreover, for any $k \in [\theta]$, let $S_{\infty}^k$ be the outcome of the propagation from $S_0$ on the sample realization $G^k$. Since $G$ is $(\varepsilon_0, \delta_0)$-stable, with probability at least $1 - \delta_0$ over $G^*$, the following holds: for any $k \in [\theta]$,

$$
Pr_{G^* \sim G, u \sim D}[S_{\infty}(u) \neq S_{\infty}^k(u)] \leq \varepsilon_0.
$$

We condition on the above event from now on. Otherwise, when $G^*$ lies in the tail (which happens with probability at most $\delta_0$), we consider the algorithm to have failed.

For $k \in [\theta]$, let $e^k$ be the difference between $S_{\infty}$ and $S_{\infty}^k$:

$$
e^k = Pr_{u \sim D}[S_{\infty}(u) \neq S_{\infty}^k(u)].
$$

Observe that $\{e^k\}_{k \in [\theta]}$ are iid random variables in $[0, 1]$, whose common expectation does not exceed $\varepsilon_0$.

We now apply Lemma 4.2 with different parameters (particularly, different $\epsilon'$, $\delta'$, $\delta$) for each $G^k$, to bound the error rate of $H_k$ w.r.t. the actual outcome $S_{\infty}$. For any $k \in [\theta]$, apply Lemma 4.2 with $m' = m$, $\delta' = \delta/3$ and

$$
\epsilon' = \epsilon^k = \max \left(2e^k, \frac{\varepsilon}{8} \cdot \left(1 + \frac{\text{VC}(G^k)}{\mathbb{E}_{G \sim \mathcal{D}}[\text{VC}(G)]}\right)\right).
$$

Note that the condition of Lemma 4.2 is satisfied. In particular, there is a hypothesis $S_{\infty}^k$ consistent with $G^k$ that has error rate $e^k \leq \epsilon^k/2$, and with a properly chosen constant factor in $m$,

$$
m' = O \left(\frac{\text{VC}(G^k) \log(1/\epsilon') + \log(1/\delta')}{\epsilon'}\right)
\leq O \left(\frac{(8 + o(1))(\text{VC}(G^k) \log(1/\epsilon) + \log(1/\delta))}{\epsilon(1 + \text{VC}(G^k)/\mathbb{E}_{G \sim \mathcal{D}}[\text{VC}(G)])}\right)
\leq O \left(\frac{(8 + o(1))(\mathbb{E}_{G \sim \mathcal{D}}[\text{VC}(G)] \log(1/\epsilon) + \log(1/\delta))}{\epsilon}\right)
\leq m.
$$
Taking a union bound over \( k \in [\theta] \), the above holds simultaneously for all \( H^k \) with probability at least \( 1 - \theta \cdot \delta' \geq 1 - \delta / 3 \). Again, we condition on the above event from now on. Whenever the above does not happen (with probability at most \( \delta / 3 \)), we consider the algorithm to have failed.

Observe that the expected error rate of \( \{ H^k \}_k \) is already low (i.e., on the order of \( \epsilon \)). To be specific, note that \( \{ G^k \}_k \) are iid variables with distribution \( G \) even if we condition on the event that Algorithm 2 succeeds for all \( \{ G^k \}_k \), since the latter depends only on the randomness in the labelled sample nodes. As a result, for any \( k \in [\theta] \),

\[
\mathbb{E}_{G^k}[\epsilon^k] = \mathbb{E}_{G^k}\left[ \max\left( 2\epsilon^k + \epsilon^0, \frac{e}{8} \left( 1 + \frac{\text{VC}(G^k)}{\mathbb{E}_{G \sim G}[\text{VC}(G)]} \right) \right) \right] \\
\leq \mathbb{E}_{G^k}\left[ 2e^k + \epsilon^0, \frac{e}{8} \left( 1 + \frac{\text{VC}(G^k)}{\mathbb{E}_{G \sim G}[\text{VC}(G)]} \right) \right] \\
\leq 2\mathbb{E}_{G^k}[\epsilon^k] + \epsilon^0 + \frac{e}{8} \cdot \frac{\text{VC}(G^k)}{\mathbb{E}_{G \sim G}[\text{VC}(G)]} \\
\leq 2\epsilon_0 + \frac{e}{4}.
\]

Since \( \epsilon \geq C_1 \epsilon_0 \), the above is upper bounded by \( \epsilon / 3 \) whenever \( C_1 \geq 24 \). The problem is that each \( H^k \) may still have error rate larger than \( \epsilon \) with probability larger than \( \delta \). To this end, we apply majority voting to boost the probability of success.

First, we bound the average error rate of \( \{ H^k \}_k \) using concentration inequalities, and show that with high probability (i.e., at least \( 1 - \delta / 3 \)), it is at most \( \epsilon / 2 \). Observe that \( \{ \epsilon^k \}_k \) are iid variables in \([0, 1] \). For small enough \( \delta \), by the multiplicative Chernoff bound,

\[
\Pr\left[ \frac{1}{\theta} \sum_{k \in [\theta]} \epsilon^k \geq \frac{\epsilon}{2} \right] = \Pr\left[ \frac{1}{\theta} \sum_{k \in [\theta]} \epsilon^k \geq \left( 1 + \frac{1}{2} \right) \cdot \frac{\epsilon}{3} \right] \\
\leq \exp\left( -\frac{(1/2)^2 \cdot (\theta \epsilon / 3)}{2 + 1/2} \right) \\
= \exp\left( -\frac{10 \log(\delta)}{3} \right) \leq \delta^{\frac{1}{3}}.
\]

We remark that the actual mean of \( \epsilon^k \) may be smaller than \( \epsilon / 3 \), but that only makes the probability smaller. So with probability at least \( 1 - \frac{\delta}{3} \),

\[
\frac{1}{\theta} \sum_{k \in [\theta]} \epsilon^k \leq \frac{\epsilon}{2}.
\]

The final step is to show that the majority vote amplifies the average error rate of the “voters” at most by a factor of 2.

**Lemma 4.3.** Fix a space \( X \), a distribution \( D \) over \( X \), and \( c \subseteq X \). Suppose there are \( \theta \) subsets of \( X \), \( \{ h^k \}_k \), satisfying

\[
1 / \theta \sum_{k \in [\theta]} \Pr_{x \sim D}[c(x) \neq h^k(x)] \leq \epsilon',
\]

for some \( \epsilon' > 0 \). Then the pointwise majority vote \( h \) of \( \{ h^k \}_k \), defined such that for any \( x \in X \),

\[
h(x) = \mathbb{I}\left[ \sum_{k \in [\theta]} h^k(x) \geq \theta / 2 \right],
\]

satisfies \( \Pr_{x \sim D}[h(x) \neq c(x)] \leq 2 \epsilon' \).

See the appendix for a proof of the lemma. Now we apply Lemma 4.3 to \( \{ H^k \}_k \), with \( \epsilon' = \epsilon / 2 \). The condition is satisfied, since the error rate of \( H^k \) is upper bounded by \( \epsilon^k \), and the average of these \( \{ \epsilon^k \}_k \) is at most \( \epsilon / 2 \). As a result, the majority vote \( H \), which is the output of Algorithm 1, has error rate at most \( \epsilon \). As for the failure probability, recall that the algorithm may fail in 3 cases: (1) the actual realization \( G^* \) lies in the tail, so no property can be guaranteed by the \((\epsilon_0, \delta_0)\)-stability of \( G \), which happens with probability at most \( \delta_0 \); (2) one of the calls to Algorithm 2 fails, which happens with probability at most \( \delta / 3 \) over the labelled sample nodes; and (3) the upper bound on the average error rate of \( \{ H^k \}_k \) exceeds \( \epsilon / 2 \), which happens with probability at most \( \delta / 3 \) over the sample realizations \( \{ G^k \}_k \). So whenever \( C_2 \geq 3 \), \( \delta_0 \leq \delta / 3 \), and taking a union bound over the 3 cases of failure, the total probability of failure does not exceed \( \delta \). This concludes the proof of Theorem 4.1. \( \square \)

### 5. Conclusion and Future Research

While various aspects of information propagation in social networks have been extensively studied, the problem of inferring the state of a network based on the structure induced by such propagation procedures has remained largely unexplored. In this paper, we study the algorithmic and statistical aspects of learning opinions in social networks. Our results show that in the classical live-edge graph model, nontrivial (and in fact, nearly optimal) inference is possible, if and only if the noise exhibited by the network is not overwhelmingly large. Future research directions include generalizing our results to other models of social networks, as well as other notions of information propagation. Also, our results can be interpreted as a generalization of the VC theorem to random hypothesis classes with some specific properties. Another interesting research question is whether such generalization is possible for more general hypothesis classes. Answering these questions would broaden the understanding of both social network analysis and the theory of passive learning.
Learning Opinions in Social Networks

References


A. Omitted Proofs

**Proof of Proposition 3.1.** Let $G = (V, E)$ be any directed graph, and $\mathcal{H}$ be its associated hypothesis class. We show below that a set of nodes $S \subseteq V$ is shattered by $\mathcal{H}$, if and only if no node in $S$ can reach another node in $S$.

Suppose $S$ satisfies for any $u, v \in S$ where $u \neq v$, $u$ cannot reach $v$. Consider any $T \subseteq S$. We show that there exists some $S_0 \subseteq V$ such that the corresponding outcome of propagation $S_\infty \in \mathcal{H}$ satisfies

$$S_\infty \cap S = T.$$

In fact, let $S_0 = T$. Since for any $u \in T$ and $v \in S \setminus T$, $u$ cannot reach $v$, we always have

$$(S \setminus T) \cap S_\infty = \emptyset,$$

which implies

$$S_\infty \cap S = T.$$

Now suppose $S$ is shattered by $\mathcal{H}$. Let $u, v \in S$ be any two nodes in $S$ where $u \neq v$. We argue that $u$ cannot reach $v$. To see this, let $S_\infty \in \mathcal{H}$ be a feasible outcome of propagation, where $u \in S_\infty$ and $v \notin S_\infty$. Such an outcome always exists, because $S$ is shattered by $\mathcal{H}$. Then if $u$ can reach $v$, we have

$$u \in S_\infty \implies v \in S_\infty,$$

a contradiction.

Putting the above together, we see immediately that $VC(G) = VC(H)$.

**Proof of Proposition 4.1.** Observe that the Bayesian optimal algorithm for recovering $S_\infty(u)$ is the following. If $u$ is among the sample nodes, then output its label. Otherwise output the majority label of the sample nodes. There are two possible ways of correctly recovering $S_\infty(u)$.

- $u$ happens to be one of the labelled samples.
- $u$ happens to have the true majority label, and the true majority label happens to be the same as the sample majority label.

Suppose we observe $m = cn$ samples for some $c > 0$. The probability that the first case happens is $1 - (1 - 1/n)^m \approx 1 - 1/e^c$. For the second case, we allow the algorithm the additional advantage, that it always knows precisely the true majority label. Even then, the probability that the majority label coincides with $S_\infty(u)$ is only about

$$\sum_{n/2 \leq i \leq n} \frac{2}{n} \cdot \frac{i}{n} \approx \frac{3}{4}.$$

Taking a union bound, the probability of the optimal algorithm outputting the correct label is only about

$$\frac{7}{4} - \frac{1}{e^c}.$$

Taking $c = 0.01$, the above probability is clearly smaller than $9/10$.

**Proof of Lemma 4.2.** We first show the easy part of the claim, i.e., with high probability $h^*$ has empirical error at most $\frac{3}{5} \varepsilon'$. By the multiplicative Chernoff bound, we have

$$\Pr \left[ \frac{1}{m'} \sum_i \mathbb{I}[h^*(x_i) \neq y_i] \geq \frac{3}{5} \varepsilon' \right] \leq \exp \left( -\frac{(1/5)^2 m' (\varepsilon'/2)}{2 + 1/5} \right).$$

For a properly chosen constant in $m'$, this is at most $\delta'/2$. 
Now consider the harder part, i.e., with high probability, any hypothesis $h$ where $\Pr[h(x) \neq c(x)] \geq \varepsilon'$ has empirical error at least $\frac{3}{4}\varepsilon' > \frac{3}{2}\varepsilon'$. We apply the double sampling technique. That is, we first sample $2m'$ iid samples, denoted $S$, and then choose uniformly at random $m'$ elements from $S$ to form $S_1$. Clearly $S_1$ is identically distributed as $\{(x_i, y_i)\}_{i \in [m]}$. Let $\mathcal{E}_1$ be the event that there exists some $h \in \mathcal{H}$ where $\Pr[h(x) \neq c(x)] \geq \varepsilon'$, such that

$$\frac{1}{m'} \sum_{(x,y) \in S_1} \mathbb{I}[h(x) \neq y] \leq \frac{3}{4}\varepsilon'.$$

We want to show that $\Pr[\mathcal{E}_1] \leq \delta'/2$. Let $S_2 = S \setminus S_1$, and $\mathcal{E}_2$ be a proxy of $\mathcal{E}_1$, i.e., the event that there exists $h \in \mathcal{H}$ where $\Pr[h(x) \neq c(x)] \geq \varepsilon'$, such that

$$\frac{1}{m'} \sum_{(x,y) \in S_1} \mathbb{I}[h(x) \neq y] \leq \frac{3}{4}\varepsilon', \tag{1}$$

and moreover,

$$\frac{1}{m'} \sum_{(x,y) \in S_2} \mathbb{I}[h(x) \neq y] \geq \frac{7}{8}\varepsilon'. \tag{2}$$

One may show that $\Pr[\mathcal{E}_2 | \mathcal{E}_1] = \Omega(1)$ for the choice of $m'$ in the lemma. So to upper bound $\Pr[\mathcal{E}_1]$, we only have to show $\Pr[\mathcal{E}_2] \leq C \cdot \delta'$, where

$$C = \frac{1}{2} \Pr[\mathcal{E}_2 | \mathcal{E}_1] = \Omega(1).$$

We argue below that for any choice of $S$, $\Pr[\mathcal{E}_2 | S] \leq C \cdot \delta'$, and therefore $\Pr[\mathcal{E}_2] \leq C \cdot \delta'$. Intuitively, fixing $S$, $\mathcal{E}_2$ is the event that for some $h \in \mathcal{H}$, the sample points misclassified by $h$ is distributed in an unbalanced way into $S_1$ and $S_2$. By Sauer’s Lemma (see, e.g., (Kearns & Vazirani, 1994)), fixing $S$ where $|S| = 2m'$, the number of effectively different hypotheses on $S$ in $\mathcal{H}$, i.e.,

$$N = |\{h \cap S | h \in \mathcal{H}\}|,$$

is at most $(2em'/d)^d$, where $d = \text{VC}(\mathcal{H})$. Therefore we only need to consider these $N$ hypotheses. The plan is, for each $h \cap S$, we bound the probability that the points in $S$ misclassified by $h \cap S$ are distributed in a sufficiently unbalanced way into $S_1$ and $S_2$, and then take a union bound over these $N$ hypotheses.

Fix any $h|_S = h \cap S$. Let $p(h|_S)$ be the probability that $h|_S$ satisfies (1) and (2) simultaneously. If

$$\frac{1}{m'} \sum_{(x,y) \in S} \mathbb{I}[h|_S(x) \neq y] < \frac{13}{16}\varepsilon',$$

then $h|_S$ cannot possibly satisfy (1) and (2) simultaneously, i.e., $p(h|_S) = 0$. Otherwise, suppose

$$\frac{1}{m'} \sum_{(x,y) \in S} \mathbb{I}[h|_S(x) \neq y] \geq \frac{13}{16}\varepsilon'.$$

We bound below the probability that $h|_S$ satisfies (1), which clearly upper bounds $p(h|_S)$.

Note that by the choice of $S_1$, the random variables $\{\mathbb{I}[h|_S(x) \neq y]\}_{(x,y) \in S_1}$ are negatively correlated. Therefore we only need to bound

$$\Pr \left[ \frac{1}{m'} \sum_{i \in [m']} z_i \leq \frac{3}{4}\varepsilon' \right],$$

where $\{z_i\}_{i \in [m']}$ are iid Bernoulli variables with mean at least $\frac{13}{16}\varepsilon' > \frac{3}{2}\varepsilon'$. Again, by the multiplicative Chernoff bound,

$$\Pr \left[ \frac{1}{m'} \sum_{i \in [m']} z_i \leq \frac{3}{4}\varepsilon' \right] \leq \exp \left( -\frac{(1/12)^2 m' \cdot (3/4)\varepsilon'}{2} \right) = \exp(-\Omega(m\varepsilon')).$$
Learning Opinions in Social Networks

Now taking a union bound over the $N$ possible choices of $h|_S$, we get

$$\Pr[\mathcal{E}_2] \leq N \cdot \exp(-\Omega(m\epsilon')) \leq (em/d)^d \exp(-\Omega(m\epsilon')).$$

Given a properly chosen constant in $m'$, the above is upper bounded by $C \cdot \delta'$, and therefore

$$\Pr[\mathcal{E}_1] \leq \frac{1}{2}\delta'.$$

Putting the above together, the lemma follows immediately. \qed

**Proof of Lemma 4.3.** Fix some $x \in X$. Suppose $c(x) \neq h(x)$. Then it has to be the case that

$$\sum_{k \in [\theta]} I[h^k(x) \neq c(x)] \geq \frac{\theta}{2}.$$

So it always holds that

$$\frac{1}{\theta} \sum_k I[h^k(x) \neq c(x)] \geq \frac{1}{2} \cdot I[h(x) \neq c(x)].$$

Now one may bound the difference between $h$ and $c$ in the following way.

$$\Pr_{x \sim \mathcal{D}}[h(x) \neq c(x)] = \mathbb{E}_{x \sim \mathcal{D}}[I[h(x) \neq c(x)]]$$

$$\leq \mathbb{E}_{x \sim \mathcal{D}} \left[ \frac{2}{\theta} \sum_{k \in [\theta]} I[h^k(x) \neq c(x)] \right]$$

$$= \frac{2}{\theta} \sum_{k \in [\theta]} \mathbb{E}_{x \sim \mathcal{D}}[I[h^k(x) \neq c(x)]]$$

$$= \frac{2}{\theta} \sum_{k \in [\theta]} \Pr_{x \sim \mathcal{D}}[h^k(x) \neq c(x)]$$

$$\leq 2\epsilon',$$

which gives the desired bound. \qed