PAC-learnability of influence functions in social networks.

Aravind Sankar (asankar3)

May 9, 2019

1 Abstract

Recent years have seen tremendous interest in understanding and predicting the spread or diffusion of information on social media platforms such as Twitter, Facebook, etc. Modeling nformation spread involves learning influence functions that map a set of initially activated seed nodes, to the resulting set of influenced nodes in the social network. In this project, we review and analyze the PAC-learnability results derived by [4] of two classical well-studied influence functions: Linear Threshold (LT), Independent Cascade (IC) models under the *partial observation* setting, *i.e.*, we only know the final set of influenced nodes, while the exact time steps of activation are unknown.
To analyze LT, the influencing process over multiple time steps is viewed as a multi-layer neural network with deterministic binary valued outputs. Classical VC dimension bounds are applied to obtain sample complexity guarantee through uniform convergence arguments. We analyze a special case of k-regular graphs to derive precise bounds. On the other hand, the IC model is inherently stochastic and continuous valued, which precludes a similar VC-based analysis. To address this issue, the IC influence function is interpreted as an expectation over random draws of subgraphs and the edge weights of social network are assumed to satisfy certain mild regularity conditions. This facilitates the application of standard uniform convergence arguments based on covering numbers. In this project, we briefly introduce the concept of covering numbers in the context of bounding Rademacher averages to contrast the proof techniques with standard VC-based analyses.

2 Preliminaries

Definition 1 (Social Network)**.** *A social network is a finite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $V = \{1, \ldots, n\}$ and edges $E \subseteq V^2$, $|E| = r$. Each edge (u, v) has a non-negative weight $w_{uv} \in \mathbb{R}^+$, that indicates the strength of influence of node v on node u.*

Influence propagation in a social network starts from a small set of users, named *seeds*, which have opinion 1, while the rest have 0. For simplicitly, we assume a step-wise propagation process with discrete time steps, while ignoring the exact timestamps of activation/influence. At each step, a node may change its opinion from 0 to 1 based on the opinion of its neighbors. Once influenced, the opinion never changes back to 0.

Definition 2 (Social Network)**.** *A diffusion cascade $(X_i, Y_i^{1:n})$ consists of an ordered sequence of at most n node activations in ascending order of time, where X_i is the seed set of the cascade, $Y_i^{1:n} = \{Y_i^1, \ldots, Y_i^n\}$ and Y_i^t is the set of nodes influenced at time step t.*

Here, we consider a *partial observation* scenario, where the exact order of influence is not available to learn an influence model, *i.e.*, we only know the final set of influenced nodes in each cascade (agnostic to activation orders). In this scenario, the training set of samples can be defined as:

Definition 3 (Training Samples). *$Z^m = \{(X_1, Y_1), \ldots, (X_m, Y_m)\}$ where X_i is the seed set and Y_i is the set of influenced nodes in training sample/cascade i.*

An influence function learns a model from the training samples Z^m to map a set of seed nodes X to the set of final influenced nodes Y. Formally, an influence function f can defined as:

Definition 4 (Influence Function). *$f : 2^{\mathcal{V}} \mapsto [0,1]^n$ which maps an input seed set X to activation probabilities $[f_1(X), \ldots, f_n(X)] \in [0,1]^n$ where $f_u(X)$ indicates the probability of influencing node $u \in \mathcal{V}$ during any time step of propagation.*

We denote the class of all influence functions under an influence model over $\mathcal{G}$ by $\mathcal{F}_G$. We measure the performance of any influence function f using a loss function $l : 2^{\mathcal{V}} \times [0,1]^n \mapsto \mathbb{R}^+$ that measures the mismatch between the target set of influenced nodes $Y \subseteq \mathcal{V}$ and predicted activation probabilities $f(X) \in [0,1]^n$. The empirical and generalization risk for an arbitrary algorithm f under a loss function $l(\cdot)$ are denoted by $L_n(f)$ and $L(f)$ respectively, which are defined as:

$$L_n(f) = \frac{1}{m} \sum_{i=1}^{m} l(Y_i, f(X_i)) \qquad L(f) = \mathbb{E}_{X,Y}[l(Y, f(X))]$$

Definition 5 (PAC-learnability of $\mathcal{F}_G$). *We say that the class of influence functions $\mathcal{F}_G$ is PAC-learnable if with probability $1-\delta$, $L(f) - L^*(\mathcal{F}_G) \leq \epsilon \;\; \forall m \geq m(\epsilon, \delta)$ where $L^*(\mathcal{F}_G) = \inf_{f \in \mathcal{F}_G} (L(f))$.*

Here, we restrict our analysis to defining PAC learnability, rather than the efficiency of the algorithm. We are only concerned about the existence of a learnable algorithm, while a polynomial time algorithm may not be possible. Now, we define two specific influence models that have been well-studied in the literature of social network analysis, Linear Threshold [2] and Independent Cascade [3] models. Let A_t denote the set of nodes that are influenced at time step t.

Definition 6 (Linear Threshold). *Each node $u \in \mathcal{V}$ has non-negative threshold $k_u \in \mathbb{R}^+$ and is influenced at time step t if the sum of the edge weights from its previously influenced neighbors at time step $t-1$ exceeds threshold k_u, i.e., $\sum_{v \in N(v) \cap A_{t-1}} w_{uv} \geq k_u$.*

Definition 7 (Independent Cascade). *Each node $u \in \mathcal{V}$ is influenced at time t independently by each previously influenced neighbor v with probability p_{uv} at time step $t-1$. Each node can then influence its neighbors for one time step, and never changes its opinion to 0. Here, the influence probability along an edge is defined as: $p_{uv} = w_{uv} / \sum_{v' \in N(u) \cup \{u\}} w_{uv'}$.*

3 Learnability of Linear Threshold (LT) Model

In this section, analyze PAC-learnability of LT model under a realizable setting, *i.e.*, the training samples Z^m are generated using an LT model. Since the influence process is deterministic with binary-valued outputs, the 0-1 loss is used to evaluate performance. Specifically, for target set $Y \subseteq \mathcal{V}$ and predictions $q \in \{0,1\}^n$, the 0-1 loss function is defined as: $l_{0\text{-}1}(Y, q) = \frac{1}{n} \sum_{u=1}^{n} \left(q_u \neq \mathbf{1}_{\{u \in Y\}}\right)$.

Let $f_w \in \mathcal{F}_G$ denote an arbitrary influence function with parameters $w \in \mathbb{R}^{r+n}$ (edge weights and thresholds). There exists an ERM algorithm that outputs an influence function $\hat{f}_w$ with zero error on training samples Z^m, *i.e.*, $L_m(\hat{f}_w) = \frac{1}{m}\sum_{i=1}^{m} l(Y_i, \hat{f}_w(X_i)) = 0$, since LT is deterministic and training samples are generated via an LT model. Note that we are only concerned about the existence of an ERM algorithm and not the computational feasibility of implementation. We now state the key PAC-learnability result of the LT model.

Theorem (PAC learnability under LT model). The class of influence functions $\mathcal{F}_G$ under the LT model is PAC-learnable *wrt* 0-1 loss with sample complexity $\tilde{O}(r + n/\epsilon)$.

Since learnability under binary-valued functions is closely tied to the VC-dimension of the function class $\mathcal{F}_G$, we state another lemma that is crucial to proving this theorem. For a given node u that has not been influenced yet, let f^u be the influence function predicting the activation of u.

Lemma (VC-dimension of LT influence functions). For a fixed node $u \in \mathcal{V}$, the class of all LT influence functions $f^u : 2^V \mapsto \{0, 1\}$ has a VC-dimension of at most $\tilde{O}(r + n)$

We first briefly sketch the key ideas towards proving this lemma and then illustrate the proof for the special case of k-regular graphs.

Single-step LT: First, we analyze a single-step of activation for node $u \in \mathcal{V}$ at time step t assuming Z is the set of influenced nodes at time step $t-1$. At time step t, the output of influence function f^u_w is defined as: $f^u_w(Z) = \mathbf{1}_{\left\{\sum_{v \in N(u) \cap Z} w_{uv} \geq k_u\right\}}$. This function can be interpreted as a two-layer neural network (NN) with linear activations, *i.e.*, the input layer has n units (one per node) with binary values indicating whether each node belongs to the set of influenced nodes at $t-1$, Z. The output layer (for node u) is a single binary unit with activation threshold k_u that takes 1 if u is influenced at time step t and 0 otherwise. The connections between the input and output layer are determined based on the edges in $\mathcal{G}$, with connection weights given by w_{uv}. In total, there are n output units, each corresponding to a node in $\mathcal{G}$, *i.e.*, the output layer returns a binary vector indicating activation of each node at time step t. Next, we generalize this formulation to multiple propagation steps.

Multiple Steps: To analyze multiple propagation steps, an intuitive strategy is to extend the above two-layer neural network by replicating the second layer for multiple time steps. However, the LT model forces each node to be influenced only once, which implies further constraints on the structure of the neural network, *e.g.*, a self-loop can be set to have a weight exceeding threshold k_u so as to enforce that u remains active forever, once activated. However, such conditions only constrain the expressive power of the neural network. Thus, f^w is a neural network with $n+1$ layers, where each layer has $r+n$ parameters. The overall influence function can be represented as a neural network with $n+1$ layers, with each layer containing $r+n$ parameters. A naive application of classic VC bounds bounds result in $n(r+n)$ parameters with VC dimension $O(n(r+n)\log(n(r+n))$. Since parameters in each layer are shared, the bound can be improved to $O((r+n)\log(r+n)$. We illustrate a special case of this bound in the following example.

Example k-regular graphs. *VC dimension of all functions f^w_u for node u satisfied $VC(f^w_u) \leq 4(k+1)\log((k+1)n)$.*

Consider the case of k-regular graphs where each node in $\mathcal{G}$ has exactly k neighbors. Let us denote by $f_w^{t,u} : \{0,1\}^n \mapsto \{0,1\}$ the function computed at node u in layer $t+1$ for a given seed set X. Also, let $\mathcal{F}^{t,u}$ denote the class of LT influence functions for different values of parameter w. Since the layer 1 is the input layer, we examine the functions at $t = 2$, denoted as $f^{1,u}$, which correspond to a half-space classifiers. Since every node u has k neighbors, $\text{VC}(f^{1,u}) = k+1$. Similarly, n such units (for each node) are computed in layer 1 with linear threshold activations, which are required as input to the third layer. Thus, the second layer has n half-space classifiers (of dimension k) to produce n outputs (which is a cartesian product). We state the following two propositions without proof:

Proposition Shatter coefficient of cartesian products. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two function classes, and let $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2$ be their cartesian product. Then, we have $S_m(\mathcal{F}) \leq S_m(\mathcal{F}_1) \cdot S_m(\mathcal{F}_2)$.

Proposition Shatter coefficient of compositions. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two function classes, and let $\mathcal{F} = \mathcal{F}_2 \circ \mathcal{F}_1$ be their composition. Then, we have $S_m(\mathcal{F}) \leq S_m(\mathcal{F}_1) \cdot S_m(\mathcal{F}_2)$.

By Sauer-Shelah lemma and the above two propositions,

$$S_m(f^{1,u}) \leq \prod_{j=1}^{n} \left(\frac{me}{k+1}\right)^{k+1} \quad \forall m \geq k+1 \tag{1}$$

Here, our key claim is that with each new layer having the same connection weights, the ability of a neural network to shatter a subset of points can only reduce. We formalize this argument as follows: For any set of points of a given size shattered by $f^{t,u} : t \geq 2$, we prove that there exists a set of points of the same size shattered by $f^{t-1,u}$.
Consider a set of points $\{x_1, \ldots, x_N\}$ shattered by $f^{t,u} : t \geq 2$, *i.e.*, let $|f^{t,u}(x_1), \ldots, f^{t,u}(x_N)| = 2^N$ over all possible parameters w. The key observation is that we can write each $f^{t,u}(x_j) = f^{t-1,u}(z_j)$ where $z_j = (f^{1,1}(x_j), \ldots, f^{1,n}(x_j))$. Here, each z_j corresponds to some point (obtained after the application of the first layer) and the operations of the remaining $t-1$ layers on z_j can be written as $f^{t-1,u}(\cdot)$ since the connection weights are shared. Since $|f^{t,u}(x_1), \ldots, f^{t,u}(x_N)| = 2^N$, it is necessarily the case that $|f^{t-1,u}(z_1), \ldots, f^{t-1,u}(z_N)|$ over all possible parameters w, which implies that the set of points $z_1, \ldots, z_N$ are shattered by $f^{t-1,u}$. Thus, we get $VC(f^{t,u}) \leq VC(f^{t-1,u})$. Now, we use this result along with Eqn. 1 to get

$$S_m(f^{t,u}) \leq \left(\frac{me}{k+1}\right)^{(k+1)n} \quad \forall m \geq k+1$$

We can easily see that for $k \geq 2$, $m = 4(k+1)\log((k+1)n)$ is sufficient to make the RHS of the above equation $\leq 2^m$. Thus, we get a bound of the VC dimension, *i.e.*, $VC \leq 4(k+1)\log((k+1)n)$.

Thus, we illustrate the proof to derive a bound on the VC dimension for the LT influence function given a node u in the special case of k-regular graphs. The PAC-learnability of LT influence functions directly follows with VC-based bounds for binary-valued functions.

4 Learnability of Independent Cascade (IC) Model

In this section, analyze PAC-learnability of the IC model, which has probabilistic outputs. Instead of 0-1 loss, the squared loss is used to measure performance. Specifically, for target set $Y \subseteq \mathcal{V}$ and

predictions $q \in \{0,1\}^n$, the squared loss function is defined as:

$$l_{\text{sq}}(Y,q) = \frac{1}{n}\sum_{u=1}^{n} \mathbf{1}_{\{u\in Y\}}\,(1-q_u)^2 + \mathbf{1}_{\{u\notin Y\}}\,(q_u)^2$$

A key assumption that is crucial to prove learnability of IC functions is to assume that the edge probabilities are bounded away from 0 and 1, *i.e.*, $w \in [\lambda, 1-\lambda]^r$ for some $0 \leq \lambda \leq 0.5$.
First, the inherently stochastic IC function is given a closed-form interpretation as an expectation over a randomly drawn subset of edges (subgraph) A from the graph G [3], *i.e.*, the active edges can be viewed as having been chosen using independent Bernoulli draws.

$$f_u^w(X) = \sum_{A\subseteq E} \prod_{(a,b)\in A} w_{ab} \prod_{(a,b)\notin A} (1-w_{ab})\sigma_u(A,X)$$

where $\sigma(A,X)$ evaluates to 1 if u is reachable from X via edges in randomly drawn subgraph A. We consider a surrogate loss function, defined by the log-likelihood, which is a simpler function to analyze, defined as:

$$L(X,Y,w) = \sum_{u=1}^{n} \mathbf{1}_{\{u\in Y\}} \log(\hat{f}_w^u(X)) + \mathbf{1}_{\{u\notin Y\}} \log(1-\hat{f}_w^u(X))$$

The algorithm we consider for PAC-learnability is obtained through maximum likelihood (ML) estimation of the IC influnece function $\hat{f}_w^u(X)$ using the above surrogate objective, *i.e.*, the parameters $\bar{w}$ are learned by optimizing:

$$\bar{w} = \max_{w\in[\lambda,1-\lambda]^r} \sum_{i=1}^{m} L(X_i,Y_i,w)$$

It can also be easily verified that under the assumptions of the parameters w, the log-likelihood function is bounded and lipschitz continuous. We leave the reader to refer to the proof in the original paper [4].

Theorem (PAC learnability under IC model). The class of influence functions $\mathcal{F}_G$ under the IC model is PAC-learnable *wrt* squared loss with sample complexity $\tilde{O}(n^3 r/\epsilon^2)$.

Before delving into the proof, we first introduce the basics of covering numbers and their application to PAC-learnability through uniform convergence of empirical means [1]. Informally, covering number defines the number of L_p balls of size ϵ needed to completely a given space.

Definition 8 (Covering Number)**.** *Let S be a metric space with L_p norm and $T \subset S$. We define $T' \subset S$ as an ϵ-cover for T, if for all $x \in T$, there exists $y \in T'$ such that $||x-y||_p \leq \epsilon$. The ϵ-covering number of (T, L_p), denoted by $\mathcal{N}(\alpha, T, L_p)$ is the size of the smallest ϵ-covering.*

We now state and prove an extension of finite class lemma (that is applicable only to finite hypothesis classes) to bound the rademacher averages based on covering numbers.

Theorem (Bounding Rademacher averages using covering numbers).. For any $\mathcal{A} \subset \mathbb{R}^n$ such that $||a||_2 \leq L\ \forall a \in \mathcal{A}$, we have $R(\mathcal{A}) \leq \inf_{\alpha>0} \left\{ \max_a ||a|| \frac{\sqrt{2\log \mathcal{N}_2(\sqrt{n}\alpha, \mathcal{A}, L_2)}}{n} + \alpha \right\}$

Proof: Let $\tilde{A}$ be the α-cover of $\mathcal{A}$ in L_2 norm of size $|\tilde{\mathcal{A}}| = \mathcal{N}(\sqrt{n}\alpha, \mathcal{A}, L_2)$. For element a, let $\tilde{a}$ be the covering element. Now, we can rewrite the rademacher average as:

$$R(\mathcal{A}) = \mathbb{E}_\epsilon \sup_a \frac{1}{n}\langle \epsilon, a\rangle = \mathbb{E}_\epsilon \sup_a \frac{1}{n}\langle \epsilon, a - \tilde{a}\rangle + E_\epsilon \sup_a \frac{1}{n}\langle \epsilon, \tilde{a}\rangle$$
$$\leq R(\tilde{\mathcal{A}}) + \frac{1}{n}||\epsilon||_2||a - \tilde{a}||_2 \leq R(\tilde{\mathcal{A}}) + \alpha$$

To bound $R(\tilde{A})$, we use finite class lemma to get: $R(\tilde{A}) \leq \max_a ||a||^2 \frac{\sqrt{2\log \mathcal{N}_2(\sqrt{n}\alpha, \mathcal{A}, L_2)}}{n}$. Since α is arbitrary, we take an infimum over α, which completes the proof. This theorem implies that the rademacher complexity of a function class can be bounded based on its covering number. To demonstrate learnability of IC functions, we will bound the covering number of IC functions, given by the key lemma:

Lemma (Covering number of IC influence functions). . The L_1 covering number of the class of all IC influence functions f^u for radius ϵ is $O((r/\epsilon)^r)$.

Proof:

We sketch the proof of the above lemma in two parts:

1. We first show that the IC influence function f^u_w is 1-Lipschitz *wrt* the L_1 norm, *i.e.*, For a given $X \subseteq \mathcal{V}$, for any $w, w^{'} \in \mathbb{R}^r$ with $||w - w^{'}|| \leq \epsilon$, $|f^u_w(X) - f^u_{w^{'}}(X)| \leq \epsilon$

   We exclude this proof since it directly follows from definition of the influence function.

2. Then, we use the lipschtitz property to define an ϵ-cover over the space of IC functions.

   The influence function f^u_w is parameterized by $w \in [0,1]^r$, which is bounded. It can be easily shown that the space of parameters w can be covered by $(r/\epsilon)^r$ balls of radius ϵ. Since f^u_w is 1-Lipschitz *wrt* L_1 norm, $\max_{Z\subseteq\mathcal{V}} |f^u_w(Z) - f^u_{w^{'}}(Z)| \leq ||w - w^{'}||_1$ This implies that the space of parameters w is covered by R L_1 balls if radius ϵ, then the corresponding influence functions (with parameters as the ball centers) form an L_∞ cover in the space of influence functions.

Now, that we have established a bound on the covering number of influence functions, we have all the necessary tools to establish PAC-learnability of IC influence functions. The full proof makes use of covering number based uniform convergence results for empirical (or equivalently surrogate) risk minimization over a real-valued function class. It builds upon the above stated theorem that bounds the rademacher complexity based on the covering number. We exclude the complete proof for the sake of brevity.

5 Discussion and Future Work

Although IC requires estimation of fewer parameters than LT (r versus $r + n$), we find that the sample complexity of LT model is proportional to $\frac{n}{\epsilon}$, while IC varies as the inverse of $\frac{n^3}{\epsilon^2}$. Clearly, this indicates that IC requires much more samples to learn on average, which is expected due to the stochastic nature of IC, in comparison to deterministic binary-valued LT functions. Furthermore, it is worth noting that unlike the LT model which has zero empirical risk, the optimal empirical risk for IC is non-zero in general.

The IC model makes a crucial regularity assumption of the edge probability space, *i.e.*, $w \in [\lambda, 1-\lambda]^r$. This assumption may not be completely realistic, *e.g.*, even when all the neighbors of a node are influenced at a time step, there is a small non-zero probability of the node not being influenced in the next step.

This analysis does take into account the specific input distribution of seed users, which indeed is not *i.i.d* in real world scenarios. Typically, the support of the seed distribution only covers a subset of high-degree popular nodes in a social network. In future, it would be interesting to see how these learnability results extend to a scenario with specific prior seed distributions.

References

[1] Martin Anthony and Peter L Bartlett. Neural network learning: Theoretical foundations. cambridge university press, 2009.

[2] Mark Granovetter. Threshold models of collective behavior. American journal of sociology, 83 (6):1420–1443, 1978.

[3] David Kempe, Jon Kleinberg, and Éva 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.

[4] Harikrishna Narasimhan, David C Parkes, and Yaron Singer. Learnability of influence in networks. In Advances in Neural Information Processing Systems, pages 3186–3194, 2015.