A. The Hyperparametric Model, Discussion

As discussed in Section 2, the essence of the hyperparametric model is that the diffusion probability of every edge is dictated by the features of its endpoints. Node features have been used in several studies and been proved to impact network formation (see e.g. (Lazarsfeld & Merton; McPherson et al., 2001)). Specifically, as a general principle, we tend to be friends with people that are "similar" to us, a phenomenon called *homophily* ("birds of a color flock together"). A recent line of work (Anderson et al., 2015) studies the effect of homophily in diffusion and in cascading behavior in social networks. Of course in real social networks there are connections between very diverse people as well.

The hyperparametric model that we propose in this paper can be viewed as an extension of these observations to the IC model. The intuition is that two nodes with similar features (interests, age, country of residence, etc.) will in principle be more influential to each other than diverse nodes, or similar medical characteristics between two nodes will increase the likelihood of transmitting a disease. This kind of observations are not captured by the traditional IC model, as it is unaware of the individual characteristics of each node and the homophily is present only in the network structure.

A natural question that one can ask is whether the sigmoid function is an appropriate function to use in order to encode the influence probabilities. In principle any function that takes as input two vectors and outputs a value in $[0, 1]$ could work, however the definition of the sigmoid function is very relevant for our purposes, since we can adjust the hyperparameters to capture the changes in the diffusion probabilities as a result of the agreement or the disagreement between the features of different nodes. Specifically, if two nodes have similar value in an important feature, then by choosing the respective coordinates of θ to be small we are increasing the influence probability. Similarly we can decrease the influence probability if the features are very dissimilar. The hyperparametric assumption tells us that there is a θ that is a good compromise over all the nodes in the network. Additionally, assuming that the diffusion probabilities are generated by the sigmoid function, our MLE optimization problem reduces to logistic regression, which is well-understood, in the case where every active node has only one active parent.

B. Definitions

Definition 1 (PAC learnability)**.** A hypothesis class $\mathcal{H}$ is *Probably Approximately Correct* (PAC) learnable with respect to some reward function r, if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm $\mathcal{A}$ such that for every $\epsilon, \delta \in (0,1)$ and every data-generating distribution $\mathcal{D}$, if we run $\mathcal{A}$ on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. samples generated by $\mathcal{D}$, it returns a hypothesis $\hat{h}$ such that, with probability at least $1 - \delta$ over the choice of the samples, it holds:

$$\mathbb{E}_{s\sim\mathcal{D}}[r(\hat{h}, s)] \geq \max_{h\in\mathcal{H}} \mathbb{E}_{s\sim\mathcal{D}}[r(h, s)] - \epsilon.$$

Definition 2 (Rademacher Complexity)**.** The Rademacher complexity of a hypothesis class $\mathcal{H}$, with respect to a reward function r and a training set S of size m, drawn from a distribution $\mathcal{D}$ is defined as:

$$\mathcal{R}(\mathcal{H}, S) = \frac{2}{m} \mathbb{E}_{\vec{\sigma}\sim\{-1,1\}^m} \left[\sup_{h\in\mathcal{H}} \sum_{i=1}^{m} \sigma_i r(h, s_i) \right]$$

where $\{-1, 1\}$ symbolizes the uniform distribution with that support.

Definition 3 (Covering Number (Shalev-Shwartz & Ben–David, 2014))**.** Let $A \subseteq \mathbb{R}^d$ be a set of vectors. We say that A is ϵ-covered by a set A_ϵ with respect to some norm p, if for all $\vec{a} \in A$ there exists an $\vec{a}_\epsilon \in A_\epsilon$ such that $||\vec{a} - \vec{a}_\epsilon||_p \leq \epsilon$. The covering number of A is the cardinality of the smallest A_ϵ that ϵ-covers A.

Definition 4 (Lipschitz function)**.** A function $f : A \to B$ is ρ-Lipschitz over A, with respect to some norm α, if for all $x_1, x_2 \in A$ it holds:

$$|f(x_1) - f(x_2)| \leq \rho \, ||x_1 - x_2||_\alpha .$$

C. The Distribution

As discussed in Section 2, a cascade C is a sequence of disjoint subsets of nodes $\{V_0, V_1, \dots, V_{n-1}\}$ that become active in each time step, where V_0 is the initial seed. Each cascade C is associated with some probability $\mathbb{P}[C]$ that depends on the seed-generating distribution $\mathcal{D}_0$, the structure of the graph and the diffusion probabilities of the edges ($\mathbb{P}[C] = \mathbb{P}[V_0] \cdot \mathbb{P}[V_1|V_0] \cdots \mathbb{P}[V_{n-1}|V_{\tau<n-1}]$).

Given a distribution $\mathcal{D}_0$ that generates the seed we want to define a distribution $\mathcal{D}$ that generates samples of the form $s = ((X, u), y)$ as described in Section 2. We define $\mathcal{D}$ to be the distribution that picks a cascade $C = \{V_0, V_1, \dots, V_{n-1}\}$ with probability proportional to $\mathbb{P}[C]$ and decomposes it into simpler samples of the form $s = ((X, u), y)$. This decomposition is simple (we already described it in Section 2): for every $\tau \in \{0, 1, \dots, n-1\}$, consider all the nodes $v \notin \cup_{t=0}^{\tau-1} V_t$ that are within distance of 1 from V_τ. For every v that became activated by V_τ (i.e. $v \in V_{\tau+1}$) create the sample $((V_\tau, v), 1)$, and for every v

that remained inactive create the sample $((V_\tau, v), 0)$. Once the entire list of samples for the cascade C is produced, $\mathcal{D}$ returns one of them uniformly at random.

In other words, every possible sample $s = ((X, u), y)$ is assigned probability $\mathbb{P}[s] = \sum_{C: s \in C} \frac{\mathbb{P}[C]}{\#\text{ samples in } C}$ by $\mathcal{D}$.

Essentially, $\mathcal{D}$ can be thought as a distribution that agrees with $\mathcal{D}_0$ in the creation of the cascade and then picks a single sample out of it that is representative of the entire cascade.

D. Learnability with Respect to the Diffusion Probabilities

The first thing that one might think of is to use the samples provided in order to learn the diffusion probabilities of the edges or the hyperparameter θ itself. However, it is easy to see that this approach will fail. Consider a network with only three nodes, u_1, u_2, v and two edges $(u_1, v), (u_2, v)$ with probabilities 1 and 0 respectively. Consider a sample generating distribution $\mathcal{D}$ that always activates both u_1 and u_2. Then, no matter how many samples drawn from $\mathcal{D}$ we will see, we cannot learn $p_{u_1,v}$ and $p_{u_2,v}$, while we can learn the outcome, i.e. that node v will be influenced. A simple modification shows that the same holds for the learnability of the hyperparameter θ.

Hence, if one wants to achieve convergence to the true diffusion probabilities or the true hyperparameter θ, extra assumptions on the distribution $\mathcal{D}$ as well as the feature vectors of the nodes are required. This is why in this work we are instead interested in a `PAC` learning guarantee and we use the log-likelihood function, defined in Section 2, as our reward function that allows us to interpret and utilize samples generated by any distribution $\mathcal{D}$.

E. Ommited Lemmas and Proofs, Section 3.1

As we discussed in Section 3.1, the proof of the sample complexity involves covering numbers, but we first need to go through the following two lemmas, that prove the Lipschitz continuity of the local influence function and the log-likelihood. Intuitively, this means that a small change in the argument (hyperparameter) will only impose a small change in the respective log-likelihood function. Hence if we can find a cover for the space of the hyper-parameters, we can convert it into a cover of the space of log-likelihoods with a small increase on its size.

Lemma 2 (Lipschitz Continuity of the Local Influence Function)**.** *The local influence function of any node $v \in V$, $f_v^\theta(X)$, is ρ-Lipschitz for any $X \subseteq V \setminus \{v\}$, i.e. for all $\theta, \theta' \in \mathcal{H} : ||\theta - \theta'||_1 \leq \epsilon \Rightarrow |f_v^\theta(X) - f_v^{\theta'}(X)| \leq \rho\epsilon$, where ρ depends on λ.*

Proof. Fix a node $v \in V$ and an $X \subseteq V \setminus \{v\}$. If we bound the infinite norm of the gradient of $f_v^\theta(X)$ by ρ then this would imply that f is ρ-Lipschitz with respect to the dual norm, i.e. the ℓ_1-norm.

$$
\begin{aligned}
\left|\frac{\theta f_v^\theta(X)}{\theta\theta_\ell}\right| &= \left|\frac{\theta}{\theta\theta_\ell}\Big(1 - \prod_{u \in X \cap N(v)} (1 - \sigma(\theta, x_{uv}))\Big)\right| \\
&= \left|\sum_{u \in X \cap N(v)} \frac{\theta\sigma(\theta, x_{uv})}{\theta\theta_\ell} \prod_{u' \in X \cap N(v) : u' \neq u} (1 - \sigma(\theta, x_{u'v}))\right| \\
&\leq \sum_{u \in X \cap N(v)} \left|\frac{\theta\sigma(\theta, x_{uv})}{\theta\theta_\ell}\right| \cdot \left|\prod_{u' \in X \cap N(v) : u' \neq u} (1 - \sigma(\theta, x_{u'v}))\right| \\
&\leq \sum_{u \in X \cap N(v)} \prod_{u' \in X \cap N(v) : u' \neq u} (1 - \lambda) \\
&= |X \cap N(v)|(1 - \lambda)^{|X \cap N(v)| - 1}
\end{aligned}
$$

where we used the fact that $\left|\frac{\theta\sigma(\theta x_{uv})}{\theta\theta_\ell}\right| \leq 1$, since the sigmoid function σ is 1-Lipschitz as for the ℓ_∞ norm with respect to θ. We also used the fact that the influence probabilities are bounded away from 1.

Now, since the function $h(x) = x(1 - \lambda)^{x-1}$ is maximized for $x = -\frac{1}{\ln(1-\lambda)}$, we get that $f_v^\theta(X)$ is ρ-Lipschitz, for $\rho := \frac{-1}{\ln(1-\lambda)}(1 - \lambda)^{-\left(\frac{1}{\ln(1-\lambda)} + 1\right)}$[5].

□

The Lipschitzness of the local influence function f_v^θ easily implies the Lipschitzness of the log-likelihood of the respective sample.

Lemma 3 (Boundness and Lipschitz continuity of the log–likelihood function)**.** *Fix a hyperparameter $\theta \in \mathbb{R}^d : ||\theta||_\infty \leq B$. Then, for any valid sample $s = (X, v, y)$ it holds:*

1. $\lambda \leq f_v^\theta(X) \leq 1 - \lambda^{|X \cap N(v)|}$,
2. $|\mathcal{L}(s, \theta)| \leq |X \cap N(v)| \cdot \ln(1/\lambda)$,
3. *$\mathcal{L}(s, \theta)$ is $\frac{\rho}{\lambda^{|X \cap N(v)|}}$-Lipschitz in θ with respect to the ℓ_1 norm.*

Proof. 1. The lower bound is immediate. Since X contains at least one neighbor of v and the minimum influence probability of any edge is λ, node v is influenced

[5] Notice that there is a smooth tradeoff here, if $|X \cap N(v)|$ is very small then a small change in θ will impose a small change in $f_v^\theta(X)$ since there are not many nodes trying to influence v. If $|X \cap N(v)|$ on the other hand is very large then there are so many nodes trying to do so already, that a small change in θ will not have a significant effect on $f_v^\theta(X)$ either.

by the seed X with probability at least λ. For the upper bound, note that in the best case, all the influence probabilities between v and its neighbors would be the maximum possible, i.e. $1-\lambda$. Now, remember that $f_v^\theta(X) = 1-\prod_{u\in X\cap N(v)}(1-p_{u,v}(\theta)) \Rightarrow f_v^\theta(X) \leq 1-\prod_{u\in X\cap N(v)}\lambda \leq 1-\lambda^{|X\cap N(v)|}$.

2. For a sample $s = ((X,v),y)$ we have:

$$\begin{aligned}|\mathcal{L}(s,\theta)| &= |y\ln\left(f_v^\theta(X)\right) + (1-y)\ln\left(1-f_v^\theta(X)\right)|\\ &\leq y|\ln\left(f_v^\theta(X)\right)| + (1-y)|\ln\left(1-f_v^\theta(X)\right)|\\ &\leq y|\ln\lambda| + (1-y)|\ln\left(1-(1-\lambda^{|X\cap N(v)|})\right)|\\ &\leq |\ln\lambda^{|X\cap N(v)|}| = |X\cap N(v)|\cdot\ln(1/\lambda)\end{aligned}$$

where the second inequality holds since $\lambda \leq f_v^\theta(X) < 1 \Rightarrow \ln\lambda \leq \ln\left(f_v^\theta(X)\right) < 0 \Rightarrow |\ln\lambda| \geq |\ln\left(f_v^\theta(X)\right)|$, and the third inequality since $|X\cap N(v)| \geq 1 \Rightarrow \lambda^{|X\cap N(v)|} \leq \lambda < 1$.

3. Similarly to Lemma 2, we need to bound the ℓ_∞ norm of the gradient of $\mathcal{L}$ with respect to θ. Hence:

$$\begin{aligned}&||\nabla_\theta\mathcal{L}(s,\theta)||_\infty\\ &= ||\nabla_\theta[y\ln\left(f_v^\theta(X)\right) + (1-y)\ln\left(1-f_v^\theta(X)\right)]\,||_\infty\\ &= \left|\left|y\cdot\frac{\nabla_\theta f_v^\theta(X)}{f_v^\theta(X)} - (1-y)\cdot\frac{\nabla_\theta f_v^\theta(X)}{1-f_v^\theta(X)}\right|\right|_\infty\\ &\leq \left(\left|y\cdot\frac{1}{f_v^\theta(X)}\right| + \left|(1-y)\cdot\frac{1}{1-f_v^\theta(X)}\right|\right)\cdot||\nabla_\theta f_v^\theta(X)||_\infty\\ &\leq \rho\left(\frac{y}{\lambda} + \frac{1-y}{\lambda^{|X\cap N(v)|}}\right) \leq \frac{\rho}{\lambda^{|X\cap N(v)|}}\end{aligned}$$

where the bound on the gradient of f follows from Lemma 2.

□

We are now ready to prove the covering number for the space of the log-likelihood functions. We state Lemma 1 again for completion.

Lemma 1. Let $S = \{((X_i,v_i),y_i)\}_{i=1}^m$ be a non-empty set of samples and let $\Delta_S = \max_{s\in S}|X\cap N(v)|$ (maximum in-degree of a node that was activated across all samples). The covering number of the class of all log-likelihood functions for S is $O\left(\left(\frac{B\rho d}{\lambda^{\Delta_S}\epsilon}\right)^d\right)$, i.e. we can choose a discrete cover $\mathcal{H}_\epsilon \subseteq \mathcal{H}$ of size $O\left(\left(\frac{B\rho d}{\lambda^{\Delta_S}\epsilon}\right)^d\right)$, such that for all $\theta \in \mathcal{H}$, there exists a $\theta_\epsilon \in \mathcal{H}_\epsilon$ with

$$\sup_{s\in S}|\mathcal{L}(s,\theta) - \mathcal{L}(s,\theta_\epsilon)| \leq \epsilon.$$

Proof. Remember that the unknown hyperparameter θ lies in $\mathcal{H} = [-B,B]^d$. Hence, the space of the hyperparameter is a d-dimensional hypercube and it is known that it can be covered by $\left(\frac{Bd}{\epsilon}\right)^d$ ℓ_1-balls of radius ϵ.

Also, in Lemma 3 we proved that for any sample $s = ((X,v),y)$ and for all $\theta,\theta' \in \mathcal{H}$ it holds:

$$|\mathcal{L}(s,\theta) - \mathcal{L}(s,\theta')| \leq \frac{\rho}{\lambda^{|X\cap N(v)|}}||\theta-\theta'||_1$$

This says that if the hyperparameters are separated by a distance of ϵ in the ℓ_1 space then, for any sample s the likelihoods of it with respect to θ and θ' are within a distance of $\frac{\rho}{\lambda^{|X\cap N(v)|}}\epsilon \leq \frac{\rho}{\lambda^{\max_{s\in S}|X\cap N(v)|}}\epsilon = \frac{\rho}{\lambda^{\Delta_S}}\epsilon$ from each other. Clearly, an ℓ_1 cover of radius ϵ over the parameter space can be translated to a cover of the space of likelihood functions. In particular, if the parameter space is covered by R ℓ_1-balls of radius ϵ and centers $\theta_1,\ldots,\theta_R$, then the likelihood functions $\mathcal{L}^{\theta_1},\ldots,\mathcal{L}^{\theta_R}$ form a $\frac{\rho}{\lambda^{\Delta_S}}\epsilon$-cover of the space of all the likelihood functions. Thus one can easily see that in order to have an ϵ-cover of the space of the log-likelihoods, we require at most $O\left(\left(\frac{B\rho d}{\lambda^{\Delta_S}\epsilon}\right)^d\right)$ discrete θs. Hence, given the set S, the covering number of the class is $O\left(\left(\frac{B\rho d}{\lambda^{\Delta_S}\epsilon}\right)^d\right)$.

□

The covering number allows us to consider a discrete hypothesis class instead of a continuous one, and hence we can bound its Rademacher complexity, using *Massart's lemma* for finite hypothesis classes. Subsequently, we need to associate the Rademacher complexity of the discretized class with the Rademacher complexity of the continuous one, something that can be done using the following lemma.

Lemma 4 (Discretization Lemma)**.** *Let $\mathcal{H}$ be any hypothesis class and S be a set of m samples drawn from some distribution $\mathcal{D}$, and suppose that $\mathcal{H}_\epsilon$ is an ϵ-cover of S, i.e. for any $h \in \mathcal{H}$ there exists $h_\epsilon \in \mathcal{H}_\epsilon$ such that:*

$$\sup_{s\in S}|r(h,s) - r(h_\epsilon,s)| \leq \epsilon$$

where $r(\cdot,\cdot)$ is some reward function (in our case the log-likelihood). Then it holds:

$$\mathcal{R}(S,\mathcal{H}) \leq \mathcal{R}(S,\mathcal{H}_\epsilon) + 2\epsilon$$

Proof. For any $h \in \mathcal{H}$, let h_ϵ be a hypothesis that covers it. Then, by the definition of the Rademacher complexity it holds:

$$\mathcal{R}(S,\mathcal{H}) = \mathbb{E}_{\vec{\sigma}\sim\{-1,1\}^m}\left[\sup_{h\in\mathcal{H}}\frac{2}{m}\sum_{i=1}^m\sigma_i r(h,s_i)\right]$$

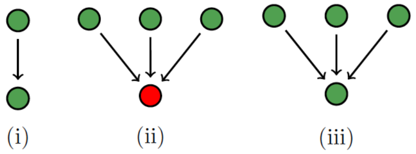

Figure 7. Three different categories of a sample. Nodes in green are active and in red inactive. In case (iii) we know that at least one of the three edges became activated but not which one(s).

$$
\begin{aligned}
&= \mathbb{E}_{\vec{\sigma}}\left[\sup_{h\in\mathcal{H}}\left(\frac{2}{m}\sum_{i=1}^{m}\sigma_i r(h_\epsilon, s_i)\right.\right.\\
&\qquad\left.\left. + \frac{2}{m}\sum_{i=1}^{m}\sigma_i(r(h,s_i) - r(h_\epsilon, s_i))\right)\right]\\
&\le \mathbb{E}_{\vec{\sigma}}\left[\sup_{h\in\mathcal{H}}\frac{2}{m}\sum_{i=1}^{m}\sigma_i r(h_\epsilon, s_i)\right.\\
&\qquad\left. + \sup_{h\in\mathcal{H}}\frac{2}{m}\sum_{i=1}^{m}\sigma_i(r(h,s_i) - r(h_\epsilon, s_i))\right]\\
&\le \mathbb{E}_{\vec{\sigma}}\left[\sup_{h_\epsilon\in\mathcal{H}_\epsilon}\frac{2}{m}\sum_{i=1}^{m}\sigma_i r(h_\epsilon, s_i)\right.\\
&\qquad\left. + \sup_{h\in\mathcal{H}}\frac{2}{m}\sum_{i=1}^{m}\sigma_i(r(h,s_i) - r(h_\epsilon, s_i))\right]\\
&= \mathcal{R}(S,\mathcal{H}_\epsilon) + \mathbb{E}_{\vec{\sigma}}\left[\sup_{h\in\mathcal{H}}\frac{2}{m}\sum_{i=1}^{m}\sigma_i(r(h,s_i) - r(h_\epsilon, s_i))\right]\\
&\le \mathcal{R}(S,\mathcal{H}_\epsilon) + \mathbb{E}_{\vec{\sigma}}\left[\sup_{h\in\mathcal{H}}\frac{2}{m}\sum_{i=1}^{m}|r(h,s_i) - r(h_\epsilon, s_i)|\right]\\
&\le \mathcal{R}(S,\mathcal{H}_\epsilon) + 2\epsilon
\end{aligned}
$$

□

Lemma 5 (Massart's lemma for finite hypothesis classes)**.** *Let $A = \{\vec{\alpha}_1, \vec{\alpha}_2, \ldots, \vec{\alpha}_N\}$ be a finite set of vectors in $\mathbb{R}^m$. Then:*

$$\mathbb{E}_{\vec{\sigma}\sim\{-1,1\}^m}\left[\max_{\vec{\alpha}\in A}\frac{2}{m}\sum_{t=1}^{m}\sigma_t\alpha_t\right] \le 2\cdot\max_{\alpha\in A}||\vec{\alpha}||\frac{\sqrt{2\log N}}{m}$$

F. Solving the Optimization Problem

As we mentioned in Section 4 there are three distinct cases for a sample $s = ((X,v),y)$ in the training set S: (i) node v was not influenced, (ii) node v was influenced and there is only one neighbor of v in X ($|X\cap N(v)| = 1$) and (iii) node v was influenced and there are more than one neighbors of v in X ($|X\cap N(v)| > 1$), as shown in Figure F.

Notice that the likelihood functions corresponding in samples of the kinds (i) and (ii) are concave (by the definition of the log-likelihood, equation (1)). Hence, if there were no obfuscated samples the optimization problem would be concave and thus efficiently solvable via iterative optimization methods such as *Gradient Descent*.

Partitioning S into S_o that contains the obfuscated samples and $S\setminus S_o$ that contains the samples of concave likelihoods, we can express our objective function as $\tilde{f}(\theta) := \frac{1}{m}\sum_{s\in S}\mathcal{L}(s,\theta) = \frac{1}{m}\sum_{s\in S\setminus S_o}\mathcal{L}(s,\theta) + \frac{1}{m}\sum_{s\in S_o}\mathcal{L}(s,\theta) =: f(\theta)+\xi(\theta)$. Optimizing $\tilde{f}$ can be perceived as optimizing a concave function f under noise ξ over a convex set.

The first approach to this problem is to ignore the obfuscated samples (i.e. the noise) and optimize f instead of $\tilde{f}$ using Gradient Descent. The success of this approach lies in the fact that the log-likelihood of each sample is bounded hence, if we have a small number of obfuscated samples the maximizer of f will approximately maximize $\tilde{f}$ as well.

Lemma 6. *Let m_o denote the number of obfuscated samples in a training set S of m i.i.d. samples drawn from $\mathcal{D}$. If $\frac{m_o}{m} \le \frac{\epsilon}{\Delta_S\ln(1/\lambda)}$, and we use Gradient Descent on $f(\theta) = \frac{1}{m}\sum_{s\in S\setminus S_o}\mathcal{L}(s,\theta)$ for $T \ge \left(\frac{Bd\rho}{\lambda^{\Delta_S}\epsilon}\right)^2$ iterations with a learning rate of $\eta = \sqrt{\frac{B^2\lambda^{2\Delta_S}}{\rho^2 T}}$, we can recover $\hat{\theta}\in[-B,B]^d$ such that:*

$$\tilde{f}(\hat{\theta}) \ge \max_{\theta\in\mathcal{H}}\tilde{f}(\theta) - 2\epsilon.$$

Proof. To simplify the notation in this proof let $\theta^* = \arg\max_\theta f(\theta)$ and $\tilde{\theta} = \arg\max_\theta \tilde{f}(\theta)$.

The first thing to notice is that, as we argued before, f is a concave function over a convex set, hence it can be approximately optimized using GD (note that GD is used for minimization problems but since f is concave, $-f$ is a convex function over a convex set and the minimum of $-f$ is the same as the maximum of f).

Also, since the function f is $\frac{\rho}{\lambda^{\Delta_S}}$-Lipschitz with respect to the ℓ_1 norm, it is $\frac{\rho\sqrt{d}}{\lambda^{\Delta_S}}$-Lipschitz with respect to the ℓ_2 norm. Additionally, it holds: $||\theta||_2 \le B\sqrt{d}$.

Known results on the convergence of GD (see e.g. (Shalev-Shwartz & Ben-David, 2014)), imply that running GD for $T \ge \left(\frac{Bd\rho}{\lambda^{\Delta_S}\epsilon}\right)^2$ iterations using a learning rate of $\eta = \sqrt{\frac{B^2\lambda^{2\Delta_S}}{\rho^2 T}}$ will return a $\hat{\theta}\in\mathcal{H}$ such that:

$$f(\hat{\theta}) \ge f(\theta^*) - \epsilon.$$

Now, we will use the fact that the noise ξ is small to show that the maximizer of the concave function f, is an approximate maximizer for the approximate concave function $\tilde{f}$:

$$\begin{aligned}\tilde{f}(\tilde{\theta}) = f(\tilde{\theta}) + \xi(\tilde{\theta}) &\leq f(\theta^*) + \xi(\tilde{\theta}) \\ &\leq f(\hat{\theta}) + \epsilon + \xi(\tilde{\theta}) \\ &\leq f(\hat{\theta}) + \xi(\hat{\theta}) - \xi(\hat{\theta}) + \epsilon \\ &= \tilde{f}(\hat{\theta}) - \xi(\hat{\theta}) + \epsilon \\ \Rightarrow \tilde{f}(\hat{\theta}) &\geq \tilde{f}(\tilde{\theta}) - \frac{m_o}{m}\Delta_S \ln\frac{1}{\lambda} - \epsilon \\ &\leq \tilde{f}(\tilde{\theta}) - 2\epsilon\end{aligned}$$

where the first inequality holds since θ^* is the maximizer of f, the second because of the GD guarantee and the third because the log-likelihood of any sample is always negative, hence $\xi(\theta) < 0$, for every $\theta \in \mathcal{H}$. From Lemma 3 we know that $\forall s \in S, \forall \theta \in \mathcal{H}: |\mathcal{L}(s, \theta)| \leq \Delta_S \ln \frac{1}{\lambda}$. Hence, using triangle inequality we can get:

$$\left|\xi(\hat{\theta})\right| = \left|\frac{1}{m}\sum_{s \in S_o} \mathcal{L}(s, \hat{\theta})\right| \leq \frac{m_o}{m}\Delta_S \ln\frac{1}{\lambda}.$$

Finally, the last inequality holds because of the assumption that: $\frac{m_o}{m} \leq \frac{\epsilon}{\Delta_S \ln(1/\lambda)}$. □

Note that in real social networks Δ_S is constant so the running time of GD is $\mathcal{O}\left(\frac{d^2}{\epsilon^2}\right)$. However, even in cases where we have a few nodes with super-constant degree, we can consider the respective samples as noise (hence add them to S_o) and still run GD in polynomial time at the price of slightly increased error, due to the increase in the noise.

Corollary 1 (Efficient Learnability)**.** *Let $G = (V, E)$ be a directed graph and $\mathcal{D}$ be a distribution that generates samples of the form $s = ((X, v), y)$. Let $\Delta = \max_{s\sim\mathcal{D}} |X \cap N(v)|$. Then, for any $\epsilon, \delta \in (0, 1)$, if we use Maximum Likelihood Estimation on a training set of size $m \geq m(\epsilon, \delta) = \mathcal{O}\Big(\Delta^2 \log^2(1/\lambda)\frac{d\log(B\rho d/\lambda^\Delta \epsilon)+\log(1/\delta)}{\epsilon^2}\Big)$ samples drawn i.i.d. from $\mathcal{D}$, and $\frac{m_o}{m} \leq \frac{\epsilon}{\Delta_S \ln(1/\lambda)}$, then with probability at least $1 - \delta$ (over the draw of the training set) it holds:*

$$\sup_{\theta \in \mathcal{H}} \mathbb{E}_{s\sim\mathcal{D}}[\mathcal{L}(s, \theta)] - \mathbb{E}_{s\sim\mathcal{D}}[\mathcal{L}(s, \hat{\theta})] \leq 3\epsilon.$$

Moreover, the MLE runs in time polynomial in d and ϵ.

Proof. Follows from the proof of Theorem 1 and the fact that $\hat{\theta}$ approximately optimizes the cummulative log-likelihood over S up to an additive term of 2ϵ, according to Lemma 7. □

We now focus on the second approach: optimize $\tilde{f}$ directly.

Corollary 2 (Using (Belloni et al., 2015))**.** *Let m_o denote the number of obfuscated samples in a training set S of m i.i.d. samples. Then if $\frac{m_o}{m} \leq \frac{\epsilon}{d\Delta_S \ln(1/\lambda)}$, there is a randomized algorithm that can recover $\hat{\theta} \in [-B, B]^d$ such that:*

$$\mathbb{E}\left[\tilde{f}(\hat{\theta})\right] \geq \max_{\theta \in \mathcal{H}} \tilde{f}(\theta) - 2\epsilon.$$

Proof. From Lemma 3 we know that $\forall s \in S, \forall \theta \in \mathcal{H}$: $|\mathcal{L}(s, \theta)| \leq \Delta_S \ln \frac{1}{\lambda}$. Hence, using triangle inequality we can get:

$$|\xi(\theta)| = \left|\frac{1}{m}\sum_{s \in S_o} \mathcal{L}(s, \theta)\right| \leq \frac{m_o}{m}\Delta_S \ln\frac{1}{\lambda}.$$

So, for $\frac{m_o}{m} \leq \frac{\epsilon}{d\Delta_S \ln(1/\lambda)}$ it holds $|\xi(\theta)| \leq \frac{\epsilon}{d}$, for all $\theta \in \mathcal{H}$. Also note that the convex set $\mathcal{H} = [-B, B]^d$ is well-rounded, according to the definition of (Belloni et al., 2015), because it is contained between the d-dimensional ball of radius B and the one of radius $B\sqrt{d}$, and that there is a trivial membership oracle to $\mathcal{H}$ (just check whether all coordinates of a vector are in $[-B, B]$).

Finally, note that since $\mathcal{L}$ is $\frac{\rho}{\lambda^\Delta}$-Lipschitz with respect to the ℓ_1 norm, f is also $\frac{\rho}{\lambda^\Delta}$-Lipschitz with respect to the ℓ_1 norm and, as a consequence, $\frac{d\rho}{\lambda^\Delta}$–Lipschitz with respect to the ℓ_∞ norm. Hence, all the requirements of the algorithm of (Belloni et al., 2015) are satisfied, and we can apply Simulated Annealing to recover a vector of hyperparameters $\hat{\theta} \in \mathcal{H}$ such that on expectation it holds:

$$\tilde{f}(\hat{\theta}) \geq \max_{\theta \in \mathcal{H}} \tilde{f}(\theta) - 2\epsilon$$

which completes the proof. □

Since for large enough training set S, the value $\frac{m_o}{m}$ will converge to the real probability p_o of seeing an obfuscated sample, the results above essentially tell us that if p_o is small enough we can still optimize the function and recover the hyperparameter despite the non-concavity of the objective function. Hence p_o quantifies the "difficulty" of the optimization problem. It depends on the distribution that generates the samples and it can be bounded in simple cases.

The following lemma provides an upper bound on that probability for the case where each node is chosen to participate in X independently with probability p_X. More involved analysis is possible for different sample-generating distributions.

Lemma 7. *Let $G = (V, E)$ be a graph and $s = ((X, v), y)$ be a sample where each node of V is chosen to participate in X independently with probability p_X. The probability p_o that s is obfuscated, is upper bounded by $1 - \big(1 - p_X \cdot$*

$(1-\lambda)\big)^{\Delta} - p_X(1-p_X)^{\Delta-1}\cdot\lambda$, *where* Δ *is the maximum degree in the graph.*

Proof. We want to bound the probability that we will get an obfuscated sample, i.e. a sample for which $y = 1$ and $|X \cap N(v)| > 1$ assuming that $v \notin X$. It holds:

$$\mathbb{P}[y=1|v\notin X] = \mathbb{P}[(y=1)\cap(|N(v)\cap X|>1)|v\notin X]$$
$$+\mathbb{P}[(y=1)\cap(|N(v)\cap X|\leq 1)|v\notin X]$$
$$\Downarrow$$
$$\mathbb{P}[(y=1)\cap(|N(v)\cap X|>1)|v\notin X] = \mathbb{P}[y=1|v\notin X]$$
$$-\mathbb{P}[(y=1)\cap(|N(v)\cap X|\leq 1)|v\notin X]$$

So to compute the probability of getting an obfuscated sample, we need to compute the probability of a node becoming active given that it does not belong in X, and the probability of becoming active while having at most one parents in X (given that it does not belong in X).

Let's fist upper bound the probability of node v becoming active. Remember that each node is selected to participate in X with probability p_X, and that for the influence probability of each edge $e \in E$ holds $p_e \in [\lambda, 1-\lambda]$. Hence:

$$\mathbb{P}[y=1|v\notin X] = 1 - \prod_{u\in N(v)}\big(1-\mathbb{P}[u \text{ activates } v]\big)$$
$$= 1 - \prod_{u\in N(v)}\big(1-p_X\cdot p_{u,v}\big)$$
$$\leq 1 - \big(1-p_X\cdot(1-\lambda)\big)^{\Delta}$$

It remains to lower bound the probability that v becomes active while having only one active parent. It is:

$$\mathbb{P}[(y=1)\cap(|N(v)\cap X|\leq 1)|v\notin X]$$
$$= \mathbb{P}[(y=1)\cap(|N(v)\cap X|=0)|v\notin X]$$
$$+\mathbb{P}[(y=1)\cap(|N(v)\cap X|=1)|v\notin X]$$
$$= 0 + \mathbb{P}[(y=1)\cap(|N(v)\cap X|=1)|v\notin X]$$
$$= \sum_{u\in N(v)} p_X(1-p_X)^{|N(v)|-1}\cdot p_{u,v}$$
$$= p_X(1-p_X)^{|N(v)|-1}\cdot\sum_{u\in N(v)} p_{u,v}$$
$$\geq p_X(1-p_X)^{\Delta-1}\cdot\lambda$$

Putting everything together we get the desired upper bound:

$$\mathbb{P}[(y=1)\cap(|N(v)\cap X|>1)|v\notin X] \leq$$
$$1-\big(1-p_X\cdot(1-\lambda)\big)^{\Delta} - p_X(1-p_X)^{\Delta-1}\cdot\lambda$$

□

G. Omitted Details from the Experiments

Synthetic Graphs: As we discussed in Section 5.1 different graph models yield graphs with different topological properties. The ones we selected for our experiments are the following:

- *Barabási-Albert:* The degree distribution of this model is a power law and hence captures interesting properties of the real-world social networks. We took 10 initial vertices and added 10 edges at each step, using the preferential attachment model, until we reached 1000 vertices.
- *Kronecker graphs:* This model for social networks was introduced in (Leskovec et al., 2005). The adjacency matrix of a Kronecker graph is generated by repeated applications of the Kronecker product to an initial seed matrix. In this case we started from a star graph with 4 vertices and computed the Kronecker product till we reached 1000 vertices.
- *Configuration model:* The configuration model allows us to construct a graph with a given degree distribution. We chose 1000 vertices and a power-law degree distribution with parameter $\alpha = 2$.
- *Erdös-Rényi:* We used the celebrated $G(n, m)$ model to create a graph with 1000 vertices and 20000 edges. $G(n, m)$ does not capture some of the properties of real social networks, however it is a very impactful model with variety of applications in several areas of science.

Training Set. We randomly activate an initial seed X of size 10% of the size of the network. X is chosen large to ensure that there exist nodes with multiple active parents and study whether convergence occurs even when (2) is indeed non-concave. We choose one node v reachable from the seed X uniformly at random. If v becomes influenced by X its label y is set to 1, and to 0 otherwise. The seed X, together with v and the label y form one sample $s = ((X, v), y)$ as described in Section 3. We generate 100,000 such samples and attempt to solve the optimization problem (2) using SGD, initializing the hyperparameters to 0 and using a learning rate of $1/\sqrt{T}$, where T is the number of iterations.