Learning Influence Adoption in Heterogeneous Networks

Abstract

We study the problem of learning influence adoption in networks. In this problem, a communicable entity (such as an infectious disease, a computer virus, or a social media meme) propagates through a network, and the goal is to learn the state of each individual node by sampling only a small number of nodes and observing/testing their states. We study this problem in heterogeneous networks, in which each individual node has a set of distinct features that determine how it is affected by the propagating entity. We give an efficient algorithm with nearly optimal sample complexity for two variants of this learning problem, corresponding to symptomatic and asymptomatic spread. In each case, the optimal sample complexity naturally generalizes both the complexity of learning how nodes are affected in isolation, and the complexity of learning influence adoption in a homogeneous network.

1 Introduction

The spread of contagious entities such as biological infections, social media memes, or computer viruses in a population via network propagation is a central object of study in the field of network science. Consider, for example, the spread of an infectious disease. Quite often, a new disease originates by genetic mutation, and begins to propagate by first infecting a group of people in close contact with the original source of the disease. After the initial infections, the disease continues to propagate as carriers of the disease interact with others: each carrier exposes colleagues, family, and friends to the disease, who may contract the disease and become spreaders themselves.

While there is substantive research on how to *control* (or *maximize*, say in viral advertising) such spread (Kempe, Kleinberg, and Tardos 2003; Chen, Wang, and Yang 2009; Tang, Xiao, and Shi 2014), and to *infer* network parameters from it (Liben-Nowell and Kleinberg 2007; Chierichetti, Liben-nowell, and Kleinberg 2011; Du et al. 2012), much less is known about *learning the state of individuals in the network* based on the propagation process. In the latter problem, we observe the outcome of the propagation process for a *sample* of individuals in the network, and use these observations to train a learning algorithm that can then infer the outcome for an individual that we have not observed. For example, suppose a university is trying to identify individuals affected by an infectious disease in its student population. Due to limited resources, it might be infeasible to test *every* student, at least not frequently; instead, it can test a sample of students and use additional information such as shared dorm residency, class registration, etc., to identify the students who are at risk of contracting the disease. We call this the *influence adoption* phenomenon in a network, and seek to study its learnability.

Recently, Conitzer, Panigrahi, and Zhang (2020) introduced this problem and studied it in the *homogeneous* model, where *any* individual who comes in contact with an infected individual gets infected as well. The state of an individual is then determined by the initial set of infected individuals and the propagation network itself (which may be random). In practice, however, individuals differ in their characteristics, and have varying levels of susceptibility to a spreading contagion. For instance, some diseases infect only people of particular age groups, some computer viruses infect only computers with specific system configurations, and some memes are meaningful only to people in particular professions. In many cases, if an individual is not infected, she is not a spreader either.

Indeed, this heterogeneity significantly changes the learning task at hand. For instance, if a disease only infects adults, and the community we care about consists exclusively of children, then we only need to learn the fact that children are unaffected, rather than the status of any particular individual. Or, if a disease were to infect people of a particular blood group, the learning task becomes more challenging since the spread of the disease is now controlled by two sets of information – the initial set of infected individuals *and the individual features of members of the population* – in addition to the propagation network itself (which may again be random). This leads to the following fundamental question:

What is the impact of heterogeneity among individuals on the learnability of influence adoption in a network?

1.1 Our Contributions

Modeling heterogeneity. We model the community of interest as a network of heterogeneous nodes representing individuals and edges representing connections between them.

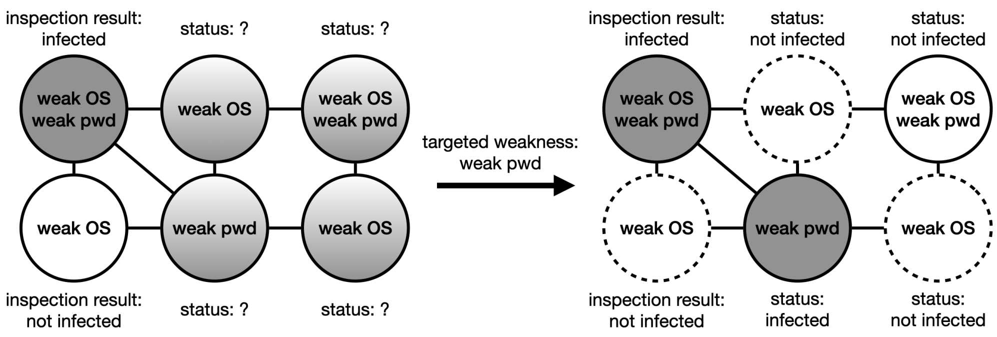

Figure 1: An example computer network, where the two computers on the left were exposed to, and therefore have been inspected for a new virus. Suppose the virus targets an unknown set of weaknesses, and suppose the presence of any one of these weaknesses allows the computer to be infected. Since the top left computer, with a weak OS and a weak password, is infected, the weaknesses targeted must include a weak OS, a weak password, or both (meaning that either would suffice for infection). On the other hand, the bottom left computer is not infected, which means the virus does not target a weak OS. So given the prior knowledge and the inspection results, we know for sure that the virus targets a weak password. Given this, we can infer the complete adoption pattern in the network — the top middle computer does not have a weak password, and therefore is invulnerable to the virus. The same is true for the bottom right computer. The bottom middle one, on the other hand, is vulnerable and in contact with the virus at the same time, so it must be infected. The top right computer is vulnerable, but the virus cannot reach it, since neither of its neighbors is infected. So the top right computer is not infected either.

Each node has a set of *features* belonging to a common feature space $\mathcal{X}$; these features determine whether the node is susceptible to the propagating entity. In addition to the features of individual nodes, the overall infection pattern also depends on the nodes that are originally exposed to the propagating entity: we call these nodes *generators*.

For example, if a computer virus is propagating through a network, then features that determine susceptibility of individual computers include the nature of the OS, the strength of passwords, etc. While for a new virus, we may not know exactly what vulnerabilities it targets, we do know that there is an underlying mapping from the space of relevant features to the susceptibility of a computer to this virus. In fact, this yields a hypothesis class of possible sets of vulnerable feature combinations that a virus may target, and makes it possible to reason about the entire infection pattern of the virus by inspecting only some computers. Similarly, we denote the set of all patterns of generators, i.e., the computers infected externally, as a hypothesis class $\mathcal{H}_g$. (See Fig. 1 for a detailed instantiation of this example.)

We also distinguish between two common modes in which a communicable entity can spread in a network. The first, which we call *symptomatic spread*, refers to the setting where a node transmits the entity to a neighbor only if it received the entity *and is susceptible to it* (in the context of epidemiology, this roughly corresponds to an SI process spreading on a graph on which some nodes are resistant/removed). This model corresponds to the analysis in the caption of Figure 1. In the second model, which we call *asymptomatic spread*, a node can transmit the entity to a neighbor as long as it received the entity itself, *irrespective of whether it is susceptible* (which roughly corresponds to an SI process where some nodes are asymptomatic). In this model, for the example in Figure 1, we would conclude that the top right node is also infected (i.e., shows symptoms), as it would be exposed and also must be susceptible. (This would correspond to the case where the virus spreads through all computers on the network but is only able to affect — say, encrypt the hard drive of — some of them. In our language, only the computers with encrypted hard drives would be 'infected.') We denote the corresponding hypothesis class for susceptibility of individual nodes $\mathcal{H}_s$.

Optimal sample complexity bounds. Our main results in this paper are asymptotically tight bounds for the *sample complexity* (i.e., the number of random nodes that need to be inspected) of learning influence adoption in heterogeneous networks, for both the symptomatic spread and asymptomatic spread models.

In particular, we show the following bounds. In these bounds, ρ is the maximum *width* of the network, n is the number of nodes in it, and d_g and d_s are the VC dimensions of the generator and susceptibility hypothesis classes respectively (we will formally define the sample complexity and the width later).

- the sample complexity of learning in the symptomatic spread model is $O(d_s \log n + \min(\rho, d_g \log n))$.
- the sample complexity of learning in the asymptomatic spread model is $O(d_s + \min(\rho, d_g \log n))$.

To gain some intuition about these bounds, let us first compare them to the homogeneous setting studied by Conitzer, Panigrahi, and Zhang (2020). In that model, all nodes are susceptible, so that $d_s = 0$, and any subset of them can be generators, so that $d_g = n$. Thus, the above bounds reduce to the width of the network ρ, which is exactly the result in Conitzer, Panigrahi, and Zhang (2020). Next, suppose every individual in a network is externally infected, so that $d_g = 0$. Then, the complexity of learning the state of each individual reduces to that of learning in the hypothesis class $\mathcal{H}_s$, which demonstrates that the sample complexity bounds must depend on d_s. Finally, suppose the network is an empty graph, so that $\rho = n$, and everybody is susceptible, so that $d_s = 0$. In this case, an individual is infected if and only if that individual is externally exposed, which shows that the dependence on d_g is also necessary.

While the discussion above establishes that the sample complexity bounds must depend on d_s, d_g, and ρ, we show further in later sections that the specific bounds given above are also asymptotically tight. For this purpose, we construct a set of "hard" instances of the problem, in which effective learning using fewer samples than these bounds can be ruled out from an information-theoretic perspective.

Random propagation. The propagation of communicable entities, such as the spread of disease between an infected and a healthy person who comes in contact, is often inherently a random event. In some cases, we may know who has come into contact with whom, but we generally still do not know for certain if there was transmission – though we may have a better idea of the probability thereof. In other cases, who came into contact with whom itself might only be known through a Bayesian process: it might not be known if two people came in contact with one another, but the probability of doing so can be inferred from their activities. We abstract from both these cases by considering a random propagation network, and give a generic reduction from sample complexity bounds for learning in deterministic environments to learning in random environments. Using this reduction and our sample complexity bounds for deterministic propagation, we obtain similar bounds for *stable* random networks. The stability condition assumes that the target error rate and failure probability of the learner are above the *resolution* of the random network, a condition that is necessary even for homogeneous networks (see (Conitzer, Panigrahi, and Zhang 2020)).

1.2 Further Related Work

Influence propagation. The phenomenon of influence propagation in a network was first studied by Kempe, Kleinberg, and Tardos (2003). Various models of influence propagation have been investigated (Gruhl et al. 2004; Chen, Wang, and Wang 2010; Chen et al. 2011; Myers, Zhu, and Leskovec 2012). Several aspects of influence propagation have been extensively studied, among which a topic particularly related to our work is learning *from* influence propagation, i.e., learning structures/parameters of networks. There, the goal is to learn parameters of the network in which the propagation happens given outcomes of the propagation procedure. As such, it can be regarded as an inverse problem of the one we study. Some representative results on the topic include (Liben-Nowell and Kleinberg 2007; Goyal, Bonchi, and Lakshmanan 2010; Chierichetti, Liben-nowell, and Kleinberg 2011; Gomez Rodriguez, Balduzzi, and Schölkopf 2011; Saito et al. 2011; Du et al. 2012; Guille and Hacid 2012; Abrahao et al. 2013; Cheng et al. 2014; Daneshmand et al. 2014; Du et al. 2014; Narasimhan, Parkes, and Singer 2015; He et al. 2016; Kalimeris et al. 2018). A related and more recent line of work is on representation learning for influence propagation (Bourigault, Lamprier, and Gallinari 2016; Li et al. 2017; Wang et al. 2017). Being an inverse problem, our results are not directly comparable to all the above.

Epidemic estimation. There is a massive body of research on epidemic estimation, where the main focus is on *macroscopic* prediction tasks, e.g., whether a meme will go viral, or the number of infections over time (Myers, Zhu, and Leskovec 2012; Weng, Menczer, and Ahn 2013). In contrast, our results aim to recover the infection status of every single node, as opposed to aggregate statistics.

Learning theory. Valiant (1984) introduced the probably approximately correct (PAC) framework for passive learning. The Vapnik-Chervonenkis (VC) theory (see (Vapnik 2013)) nicely characterizes the learnability of different concepts via the VC dimension, following which various measures of complexity have been considered (Alon et al. 1997; Bartlett and Mendelson 2002; Pollard 2012; Daniely et al. 2015), and tighter generalization bounds have also been developed (Talagrand 1994; Hanneke 2016). While these general results are powerful, as discussed in later sections, they cannot be directly applied to the specific problem we study.

2 Preliminaries

2.1 Useful Tools from Learning Theory

Definition 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 $\mathcal{X}$, denoted $d(\mathcal{H})$, is the cardinality of the largest set $S \subseteq X$ shattered by $\mathcal{H}$.

The sample complexity of PAC learning is given by the following theorem:

Theorem 2 (VC Theorem (the realizable case))**.** *Fix $\mathcal{X}$ and $\mathcal{H}$. Consider any distribution $\mathcal{D}$ over $\mathcal{X}$, $f \in \mathcal{H}$, $\delta > 0$, and $\varepsilon > 0$. Now, given $m = O((d(\mathcal{H}) \log(1/\varepsilon) + \log(1/\delta))/\varepsilon)$ i.i.d. samples from $\mathcal{D}$, the following holds with probability at least $1 - \delta$: any hypothesis $h \in \mathcal{H}$ that is consistent with all the m samples (i.e., $h(x_i) = y_i$ for all $i \in [m]$) satisfies*

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

Moreover, this bound is tight in the sense that any algorithm achieving this guarantee requires $\Omega((d(\mathcal{H}) + \log(1/\delta))/\varepsilon)$ samples.

2.2 Learning Influence Propagation in Networks

Definition 3 (Implicit Hypothesis Class)**.** Given a directed network $G = (V, E)$, the implicit hypothesis class associated with G, $\mathcal{H}(G)$, is defined to be the family of all subsets S of V, where for any two vertices $u, v \in V$, if $u \in S$ and u can reach v in G (i.e., $u \to_G v$), then $v \in S$.

In words, the implicit hypothesis class is the family of all sets that are closed under reachability in G.

Definition 4 (Width of Directed Networks)**.** The width $\rho(G)$ of a directed network $G = (V, E)$ is the size of the maximum set $S \subseteq V$ of vertices, such that for any $u, v \in S$ where $u \neq v$, there is no u to v path in G, i.e., $u \not\to_G v$.

Lemma 5 (VC Dimension of Implicit Hypothesis Class (Conitzer, Panigrahi, and Zhang 2020))**.** *Given a directed network G, the VC dimension of the implicit hypothesis class $\mathcal{H}(G)$ associated with G is $d(\mathcal{H}(G)) = \rho(G)$.*

3 Learning under Symptomatic Spread

The propagation model. In a directed network $G = (V, E)$, each node $u \in V$ has features $x(u) \in \mathcal{X}$ given by a feature mapping $x : V \to \mathcal{X}$. Given a generator concept $f_g : \mathcal{X} \to \{0, 1\}$ and a susceptibility concept $f_s : \mathcal{X} \to \{0, 1\}$, the final adoption pattern $f_a : V \to \{0, 1\}$ induced by f_g and f_s is given by: $f_a(v) = 1$ iff there exists $u \in V$, such that $f_g(x(u)) = 1$, and u can reach v in the sub-network $G(x, f_s)$ induced by $V(x, f_s) = \{w \in V \mid f_s(x(w)) = 1\}$, i.e., $u \to_{G(x,f_s)} v$. In the rest of the paper, we will abuse notation and let $f_g(u) = f_g(x(u))$, $f_s(u) = f_s(x(u))$, etc.

In words, a node u is initially in contact with the influence iff it is assigned label $f_g(u) = 1$ by the generator concept f_g. Moreover, if a node u has label $f_s(u) = 1$, then it is fully susceptible to the influence, meaning that if in contact with the contagious entity, u will adopt and subsequently spread the influence. Otherwise (i.e., when $f_s(u) = 0$), u is fully immune to the influence, meaning u does not interact with the contagious entity in any way.

Given the propagation model, we are ready to define the learning problem. Roughly speaking, the learning problem asks to design an algorithm, which given access to a number of labeled sample nodes (where the label is whether the node has adopted the influence), predicts the label of a random node correctly with probability at least $1-\varepsilon$. Moreover, the algorithm is allowed to fail with probability δ, in which case it does not need to satisfy any requirements. Below we define the problem formally.

The learning problem. Fix a network G, a feature mapping x, a generator hypothesis class $\mathcal{H}_g$ and a susceptibility hypothesis class $\mathcal{H}_s$. The learning problem asks to design an algorithm, which for any generator concept $f_g \in \mathcal{H}_g$, susceptibility concept $f_s \in \mathcal{H}_s$, distribution $\mathcal{D}$ over V, $\varepsilon > 0$ and $\delta > 0$, with $m = m(\varepsilon, \delta)$ labeled (i.e., adoption vs no adoption) samples from $\mathcal{D}$, outputs a hypothesis $h : V \to \{0, 1\}$ satisfying: with probability at least $1 - \delta$ (over the samples and the algorithm's internal randomness), $\Pr_{v\sim\mathcal{D}}[h(v) \neq f_a(v)] \leq \varepsilon$, where f_a is the adoption pattern induced by f_g and f_s.

In the above definition, we assume that the algorithm has full access to the generator and susceptibility hypothesis classes. However, being families of subsets of the feature space $\mathcal{X}$, these hypothesis classes can be quite large or even infinite, and in fact, they may not even admit a succinct representation. To this end, throughout the paper, we assume oracle access to $\mathcal{H}_g$ and $\mathcal{H}_s$ for empirical risk minimization. More specifically, we assume the following can be done in polynomial time: for $\mathcal{H} \in \{\mathcal{H}_g, \mathcal{H}_s\}$, given k labeled data points $\{(x_i, y_i)\}_{i\in[k]}$, one can find a concept h in $\mathcal{H}$ that best fits the labels, i.e., $h \in \operatorname{argmin}_{h'\in\mathcal{H}} \sum_{i\in[k]} |h'(x_i) - y_i|$.

In homogeneous networks, as shown by Conitzer, Panigrahi, and Zhang (2020), the key parameter which controls the sample complexity of learning influence adoption is the width. However, in heterogeneous networks, the width by itself is no longer an effective measure of the complexity of the network. For example, the notion of the width does not capture the phenomenon that some members of the network may not be susceptible to the influence, making the network less well-connected than it appears to be. To this end, we instead use the maximum width, a generalization of the width to heterogeneous networks, to measure the complexity of a network.

Definition 6 (Maximum Width of Directed Networks)**.** For a network G, a feature mapping x, and susceptibility hypothesis class $\mathcal{H}_s$, the maximum width is defined to be $\rho(G, x, \mathcal{H}_s) = \max_{f_s\in\mathcal{H}_s} \rho(G(x, f_s))$.

We remark that the maximum width $\rho(G, x, \mathcal{H}_s)$ can be either larger or smaller than $\rho(G)$. For example, suppose we know there is precisely one insusceptible node (but not which one). Then, when G has no edges, $\rho(G, x, \mathcal{H}_s) = \rho(G) - 1$, and when G is a chain, $\rho(G, x, \mathcal{H}_s) = \rho(G) + 1$.

Now we are ready to state our main result for the symptomatic spread model. The following theorem gives a sample complexity upper bound for the learning problem with deterministic networks — we will generalize this to random networks in Section 5.

Theorem 7. *In the symptomatic spread model, for any $G = (V, E)$, x, $\mathcal{H}_g$, and $\mathcal{H}_s$, let $\mathcal{H}_a$ be the class of all possible adoption patterns induced by generator concepts in $\mathcal{H}_g$ and susceptibility concepts in $\mathcal{H}_s$. Let $n = |V|$, $\rho = \rho(G, x, \mathcal{H}_s)$, $d_g = d(\mathcal{H}_g)$ and $d_s = d(\mathcal{H}_s)$. Then we have*

$$d(\mathcal{H}_a) = O(d_s \log n + \min(\rho, d_g \log n)).$$

As a result, there is an algorithm that learns the adoption pattern with any error rate $\varepsilon > 0$ and failure probability $\delta > 0$ using

$$O\left(\frac{(d_s \log n + \min(\rho, d_g \log n)) \cdot \log \varepsilon^{-1} + \log \delta^{-1}}{\varepsilon}\right)$$

samples. Moreover, the algorithm runs in polynomial time when d_s is a constant.

The proof of the theorem, as well as all other proofs, are deferred to the appendices. Here we provide some intuition for the essence of the theorem: the upper bound on the VC dimension of the adoption hypothesis class $\mathcal{H}_a$. Since each member of $\mathcal{H}_a$ is generated by a member of $\mathcal{H}_g$ and a member of $\mathcal{H}_s$, if we can show that the numbers of "effectively different" members in $\mathcal{H}_g$ and $\mathcal{H}_s$ (i.e., members which can induce different adoption patterns) are both reasonably small, then the cardinality of $\mathcal{H}_a$, upper bounded by the product of the two numbers, must be small too. This would imply an upper bound on the VC dimension, since in order to shatter a set of size d, $\mathcal{H}_a$ must have at least 2^d members. Given the above observations, a weaker version of the bound can be derived immediately from the Sauer-Shelah lemma, stated below.

Lemma 8 (Sauer-Shelah Lemma (rephrased))**.** *Let $\mathcal{F}$ be a family of sets with VC dimension $d(\mathcal{F}) = d$. Then for any set S with cardinality $|S| = n$, $|\mathcal{F}^{\cap S}| \leq (2en/d)^d$, where $\mathcal{F}^{\cap S} = \{T \cap S \mid T \in \mathcal{F}\}$.*

Applying the above lemma to $\mathcal{H}_g$ and $\mathcal{H}_s$ restricted to the nodes V, we have that $|\mathcal{H}_g^{\cap V}| \leq (2en/d_g)^{d_g}$ and $|\mathcal{H}_s^{\cap V}| \leq (2en/d_s)^{d_s}$. So $|\mathcal{H}_a| \leq |\mathcal{H}_g^{\cap V}| \cdot |\mathcal{H}_s^{\cap V}| = O(n^{d_g+d_s})$, which implies $d(\mathcal{H}_a) = O(\log |\mathcal{H}_a|) = O((d_g + d_s) \log n)$.

In order to obtain the full bound in Theorem 7, we also need to show that $d(\mathcal{H}_a) = O(d_s \log n + \rho)$, which requires more effort. To prove this bound, we consider the susceptibility hypothesis class $\mathcal{H}_s$, and the implicit hypothesis class associated with the sub-network induced by susceptible nodes, i.e., $\mathcal{H}(G(x, f_s))$, for each $f_s \in \mathcal{H}_s$. Then, a similar counting argument to the one above gives $d(\mathcal{H}_a) = O((d_s + \rho) \log n)$. However, it turns out one can do better via a refined argument, formalized in the following key lemma.

Lemma 9. *For any integer $k > 0$ and k families of sets $\mathcal{F}_1, \dots, \mathcal{F}_k$, if for any $i \in [k]$, $d(\mathcal{F}_i) \leq d$, then $d\left(\bigcup_{i \in [k]} \mathcal{F}_i\right) = O(d + \log k)$.*

Applying the lemma to $\mathcal{H}(G(x, f_s))$ for all "effectively different" $f_s \in \mathcal{H}_s$ then gives $d(\mathcal{H}_a) = O(d_s \log n + \rho)$, thereby finishing the proof of Theorem 7. One may suspect that with a further improved analysis, one can actually remove the $\log n$ factor from all terms in the upper bound. Next we show that this logarithmic dependency on n is unavoidable — and in fact, all terms in our upper bound are tight up to constant factors — even when restricted to undirected networks. See Appendix A for proof sketches.

Theorem 10. *In the symptomatic spread model, there exist families of hard instances of the learning problem in the symptomatic spread model, on which achieving constant target error rate ε and failure probability δ requires (1) $\Omega(d_s \log n)$ samples, when d_g and ρ are constant, (2) $\Omega(d_g \log n)$ samples, when d_s is constant, or (3) $\Omega(\rho)$ samples, when d_s is constant. In other words, all terms in Theorem 7 are necessary. Moreover, all hard instances only involve undirected networks.*

Finally, we show that when the learning algorithm cannot access the feature mapping, no nontrivial sample complexity bound is possible.

Theorem 11. *In the symptomatic spread model, when the feature mapping x is unknown (but the features of labeled sample nodes and nodes to make predictions for are known), there is a family of instances where d_g, d_s, and ρ are all constant, but the number of samples required to achieve constant error rate ε and failure probability δ is $\Omega(n)$.*

4 Learning under Asymptomatic Spread

The propagation model. In a directed network $G = (V, E)$, each node $v \in V$ has features $x(v) \in \mathcal{X}$ given by a feature mapping $x : V \to \mathcal{X}$. Given a generator concept $f_g : \mathcal{X} \to \{0, 1\}$ and a susceptibility concept $f_s : \mathcal{X} \to \{0, 1\}$, the final adoption pattern $f_a : V \to \{0, 1\}$ induced by f_g and f_s is given by: $f_a(v) = 1$ iff there exists $u \in V$ such that $f_g(x(u)) = 1$ and u can reach v in G, and $f_s(x(v)) = 1$. Again, we will abuse notation and let $f_g(u) = f_g(x(u))$, $f_s(u) = f_s(x(u))$, etc.

The asymptomatic spread model is similar to the symptomatic spread model discussed in Section 3, except that here, a node that is not susceptible cannot adopt, but can still spread, the influence. Given the propagation model, we can define the learning problem in a similar way.

The learning problem. Fix a network G, a feature mapping x, a generator hypothesis class $\mathcal{H}_g$ and a susceptibility hypothesis class $\mathcal{H}_s$. The learning problem asks to design an algorithm, which for any generator concept $f_g \in \mathcal{H}_g$, susceptibility concept $f_s \in \mathcal{H}_s$, distribution $\mathcal{D}$ over V, $\varepsilon > 0$ and $\delta > 0$, with $m = m(\varepsilon, \delta)$ labeled (i.e., adoption vs no adoption) samples from $\mathcal{D}$, outputs a hypothesis $h : V \to \{0, 1\}$ satisfying: with probability at least $1 - \delta$ (over the samples and the algorithm's internal randomness), $\Pr_{v \sim \mathcal{D}}[h(v) \neq f_a(v)] \leq \varepsilon$, where f_a is the adoption pattern induced by f_g and f_s.

The following theorem establishes our sample complexity upper bound for the asymptomatic spread model. The proof techniques are similar to those used to establish Theorem 7.

Theorem 12. *In the asymptomatic spread model, for any $G = (V, E)$, x, $\mathcal{H}_g$, and $\mathcal{H}_s$, let $\mathcal{H}_a$ be the class of all possible adoption patterns induced by generator concepts in $\mathcal{H}_g$ and susceptibility concepts in $\mathcal{H}_s$. Let $n = |V|$, $\rho = \rho(G)$, $d_g = d(\mathcal{H}_g)$ and $d_s = d(\mathcal{H}_s)$. Then we have $d(\mathcal{H}_a) = O(d_s + \min(\rho, d_g \log n))$. As a result, there is an algorithm that learns the adoption pattern with any error rate $\varepsilon > 0$ and failure probability $\delta > 0$ using*

$$O\left(\frac{(d_s + \min(\rho, d_g \log n)) \cdot \log \varepsilon^{-1} + \log \delta^{-1}}{\varepsilon}\right)$$

samples. Moreover, the algorithm runs in polynomial time when d_g and d_s are constant.

The following theorem shows that all terms in our sample complexity upper bound are necessary. The hard instances are similar to those in the symptomatic spread model.

Theorem 13. *In the symptomatic spread model, there exist families of hard instances of the learning problem in the asymptomatic spread model, on which achieving constant target error rate ε and failure probability δ requires (1) $\Omega(d_s)$ samples, when d_g and ρ are constant, (2) $\Omega(d_g \log n)$ samples, when d_s is constant, or (3) $\Omega(\rho)$ samples, when d_s is constant. In other words, all terms in Theorem 12 are necessary. All hard instances only involve undirected networks.*

We remark that the above bound has weaker dependency on n than the one in Theorems 7 and 10. This suggests that with the same amount of prior knowledge, it is easier to learn under asymptomatic than under symptomatic spread.

5 Random Networks

In this section, we discuss how our sample complexity bounds can be generalized to random networks. We first present a general reduction for PAC learning with random hypothesis classes, and then show how the reduction can be applied to the problem of learning influence adoption, which yields sample complexity bounds in random networks.

The random hypothesis class model. Let $\mathcal{S}$ be the set of possible states of the world. Each $s \in \mathcal{S}$ corresponds to a random concept $f_{s,r}$ over a feature space $\mathcal{X}$, given by a random mapping $r \sim \mathcal{R}$ (in other words, $f_{s,r} = r(s)$ is a random subset of $\mathcal{X}$). Moreover, r also maps $\mathcal{S}$ to a random hypothesis class $\mathcal{H}_r = \{f_{s,r} \mid s \in \mathcal{S}\}$.

Learning with random hypothesis classes. Fix a state space $\mathcal{S}$, a feature space $\mathcal{X}$, and a distribution $\mathcal{R}$ over mappings from $\mathcal{S}$ to $2^{\mathcal{X}}$. The learning problem asks to design an algorithm, which for any state $s \in \mathcal{S}$, distribution $\mathcal{D}$ over $\mathcal{X}$, $\varepsilon > 0$ and $\delta > 0$, when $r \sim \mathcal{R}$ is an unobservable random mapping drawn from $\mathcal{R}$, with $m = m(\varepsilon, \delta)$ labeled samples from $\mathcal{D}$ (i.e., $\{(x_i, y_i)\}_{i \in [m]}$ where $y_i = f_{s,r}(x_i)$ for each i), outputs a hypothesis $h : \mathcal{X} \to \{0, 1\}$ satisfying: with probability at least $1 - \delta$ (over the randomness in r, the labeled samples, and the learning algorithm), $Pr_{x \sim \mathcal{D}}[h(x) \neq f_{s,r}(x)] \leq \varepsilon$. We say such an algorithm (ε, δ)-learns the random hypothesis class $\mathcal{H}_r$.

Below we generalize the notion of stable networks by Conitzer, Panigrahi, and Zhang (2020) to PAC learning with random hypothesis classes, and show that stability enables efficient learning in the more general problem that we study.

Definition 14 ($(\varepsilon_0, \delta_0)$-Stability)**.** Fix a state space $\mathcal{S}$, a feature space $\mathcal{X}$, and a distribution $\mathcal{R}$ over mappings from $\mathcal{S}$ to $2^{\mathcal{X}}$. Let r and r' be two independent sample mappings drawn from $\mathcal{R}$. We say $(\mathcal{S}, \mathcal{X}, \mathcal{R})$ is $(\varepsilon_0, \delta_0)$-stable with respect to a distribution $\mathcal{D}$ over $\mathcal{X}$, if for any $s \in \mathcal{S}$, the following is true: with probability at least $1 - \delta_0$ over r, $\Pr_{r' \sim \mathcal{R}, x \sim \mathcal{D}}[f_{r,s}(x) \neq f_{r',s}(x)] \leq \varepsilon_0$.

Theorem 15. *Fix a state space $\mathcal{S}$, a feature space $\mathcal{X}$, and a distribution $\mathcal{R}$ over mappings from $\mathcal{S}$ to $2^{\mathcal{X}}$. Suppose $(\mathcal{S}, \mathcal{X}, \mathcal{R})$ is $(\varepsilon_0, \delta_0)$-stable with respect to some distribution $\mathcal{D}$ over $\mathcal{X}$. Then there is an algorithm that, for any $\varepsilon > C_1 \cdot \varepsilon_0$ and $\delta > C_2 \cdot \delta_0$, (ε, δ)-learns the random hypothesis class $\mathcal{H}_r$ using*

$$m = O\left(\frac{d \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}\right)$$

samples, where $d = \mathbb{E}_r[d(\mathcal{H}_r)]$ is the expected VC dimension of the random hypothesis class $\mathcal{H}_r$.

We note that the stability condition is necessary, since Conitzer, Panigrahi, and Zhang (2020) show (in Proposition 4.1) that it is statistically hard to learn anything nontrivial even in their simpler model without any assumptions on the stability of the network. Intuitively, this is because the learning procedure can be viewed as inferring the seed set given the outcome of the propagation, and if the outcome and the seed set are not correlated well enough, then information-theoretically there is no hope for inference.

Below we formally define the problem of learning influence adoption in random networks, and apply Theorem 15 to obtain sample complexity bounds. We start with the case of symptomatic spread; the case of asymptomatic spread is defined similarly.

Let $\mathcal{G}$ be a distribution over directed networks defined over vertices V (also known as a random live-edge graph in the context of influence propagation). Each node v has features $x(v)$ given by a feature mapping x. Given a generator concept f_g and a susceptibility concept f_s, the final random adoption pattern f_a induced by f_g and f_s is given by the following procedure: first draw a random network $G \sim \mathcal{G}$. Then, $f_a(v) = 1$ iff there exists $u \in V$ such that $f_g(x(u)) = 1$, and u can reach v in the sub-network $G(x, f_s)$ induced by $V(x, f_s) = \{w \in V \mid f_s(x(w)) = 1\}$, i.e., $u \to_{G(x,f_s)} v$. That is, f_a is the induced adoption pattern by f_g and f_s in a random network $G \sim \mathcal{G}$.

Observe that the random network G induces a random mapping r_G from $\mathcal{H}_g \times \mathcal{H}_v$ to $2^{\mathcal{X}}$, where $r_G(f_g, f_s)$ is the adoption pattern induced by f_g and f_s in G. We say this is the implicit mapping induced by the distribution $\mathcal{G}$, and let $\mathcal{R}(\mathcal{G})$ denote the distribution of this mapping. Moreover, let $\mathcal{H}_G = r_G(\mathcal{H}_g \times \mathcal{H}_s)$ be the class of possible adoption patterns induced by $\mathcal{H}_g$ and $\mathcal{H}_s$ in G.

Fix a distribution over networks $\mathcal{G}$, a feature mapping x, a generator hypothesis class $\mathcal{H}_g$ and a susceptibility hypothesis class $\mathcal{H}_s$. The learning problem asks to design an algorithm, which for any generator concept $f_g \in \mathcal{H}_g$, susceptibility concept $f_s \in \mathcal{H}_s$, distribution $\mathcal{D}$ over V, $\varepsilon > 0$ and $\delta > 0$, with $m = m(\varepsilon, \delta)$ labeled samples from $\mathcal{D}$, outputs a hypothesis $h : V \to \{0, 1\}$ satisfying: with probability at least $1 - \delta$, $\Pr_{v \sim \mathcal{D}}[h(v) \neq f_a(v)] \leq \varepsilon$, where f_a is the random adoption pattern induced by f_g and f_s in an unobservable random network $G \sim \mathcal{G}$.

Corollary 16. *In the symptomatic spread model with random networks, fix $\mathcal{G}$, $\mathcal{X}$, $\mathcal{H}_g$, $\mathcal{H}_s$, and x. Suppose for a distribution $\mathcal{D}$ over $\mathcal{X}$, $(\mathcal{H}_g \times \mathcal{H}_s, \mathcal{X}, \mathcal{R}(\mathcal{G}))$ is $(\varepsilon_0, \delta_0)$-stable with respect to $\mathcal{D}$. Then the random hypothesis class $\mathcal{H}_G$ is (ε, δ)-learnable for any $\varepsilon > C_1 \cdot \varepsilon_0$, $\delta > C_2 \cdot \delta_0$ using*

$$O\left(\frac{(d_s \log n + \min(\rho, d_g \log n)) \cdot \log \varepsilon^{-1} + \log \delta^{-1}}{\varepsilon}\right)$$

samples, where $\rho = \mathbb{E}_{G \sim \mathcal{G}}[\rho(G, x, \mathcal{H}_s)]$.

Corollary 17. *In the asymptomatic spread model with random networks, fix $\mathcal{G}$, $\mathcal{X}$, $\mathcal{H}_g$, $\mathcal{H}_s$, and x. Suppose for a distribution $\mathcal{D}$ over $\mathcal{X}$, $(\mathcal{H}_g \times \mathcal{H}_s, \mathcal{X}, \mathcal{R}(\mathcal{G}))$ is $(\varepsilon_0, \delta_0)$-stable with respect to $\mathcal{D}$. Then the random hypothesis class $\mathcal{H}_G$ is (ε, δ)-learnable for any $\varepsilon > C_1 \cdot \varepsilon_0$, $\delta > C_2 \cdot \delta_0$ using*

$$O\left(\frac{(d_s + \min(\rho, d_g \log n)) \cdot \log \varepsilon^{-1} + \log \delta^{-1}}{\varepsilon}\right)$$

samples, where $\rho = \mathbb{E}_{G \sim \mathcal{G}}[\rho(G)]$.

6 Experimental Evaluation

In this section, we instantiate our algorithms in a concrete setup with random networks. Arguably the simplest class of random networks is Erdős-Rényi random graphs. However, such networks are infeasible for illustrating the efficacy of our approach: since all nodes are symmetric, these networks (before realizing) carry almost no structural information that can be utilized in learning. In order to make the learning problem as *nontrivial* as possible, we instead consider the following setup. Fixing the number of nodes n, each edge (u, v) realizes independently with probability p_{uv}. These probabilities satisfy: for any $u, v \in [n]$, $p_{uv} = 1/|u - v|$ if $|u - v| \leq C$, and 0 otherwise. Intuitively, this models the case where all nodes are located on a line, and nodes closer to each other interact more frequently. This is a one-dimensional instance of the small-world model by Kleinberg (2000), except that we truncate the probabilities at distance C to make the learning problem harder. In our experiments, we choose $C = 30$. For the generator and susceptibility hypothesis classes, we consider the setting discussed in our introductory example (Figure 1): there are k potential weaknesses, and each susceptibility hypothesis (corresponding to

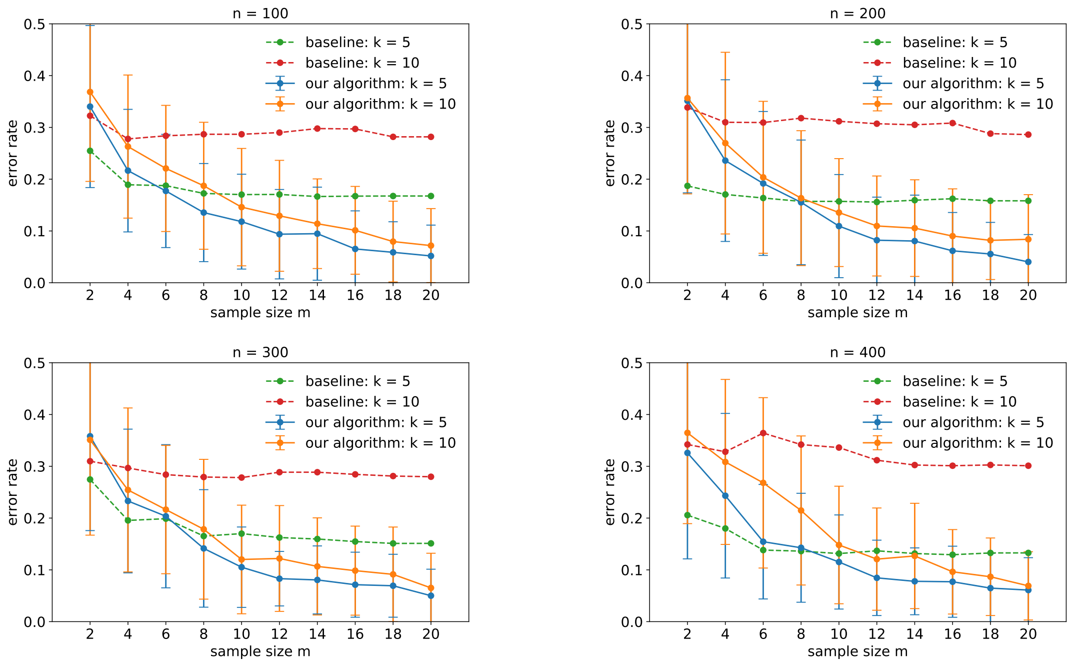

Figure 2: Error rates of our algorithm and the feature-oblivious baseline (Conitzer, Panigrahi, and Zhang 2020) on instances with different parameters. Each point is the average of 40 independent runs, and each run takes less than a minute on a laptop.

a virus) is induced by a subset of the k weaknesses, corresponding to those the virus targets. Recall that a node is susceptible as long as it has some weakness that the virus targets. For simplicity, we assume the generator hypothesis class is singletons of nodes, meaning that exactly one (unknown) computer has been initially exposed to the virus. In our experiments, we choose the ground truth generator node uniformly at random, and the ground truth susceptibility hypothesis by adding each weakness to the set targeted independently with probability $3/k$ (so in expectation the virus targets 3 weaknesses). Then we generate the set of weaknesses of each node by adding each weakness with probability 0.2. All these numbers are chosen specifically to make the instance rich and the learning problem nontrivial.

Our results are shown in Figure 2. First note that the hypothesis classes we choose satisfy $d(\mathcal{H}_g) = 1$ and $d(\mathcal{H}_s) = k$. So applying Corollary 16, up to the resolution ε_0, given m labeled samples, the error rate of our learner is $O((\log n + k)/m)$. This roughly aligns with our experimental observations: as m grows, the error rate drops roughly as the bound indicates, and tends to plateau as it approaches the stability parameter ε_0 of the network, which appears to be quite small in this setup. For all 4 values of n the curves are very similar, which corresponds to the logarithmic dependence on n. The dependence on k is more significant, as suggested by the theoretical results. For comparison, we also include the error rates of the feature-oblivious baseline algorithm given in (Conitzer, Panigrahi, and Zhang 2020), which ignores the sets of weaknesses and simply assume all nodes are susceptible (there are two curves because the networks are generated differently for different values of k). As the figure shows, the baseline algorithm fails to exploit the heterogeneity of the network, and suffers significantly higher error rates.

7 Conclusion

In this paper, we studied the problem of influence adoption in the context of heterogeneous networks. There are several interesting follow-up directions from this study. First, there are situations, particularly in the context of infectious diseases, where the state of being infected or healthy is transient. In this case, can the state of individual nodes be learned over *time*, by periodically sampling nodes for their current state? It is also important to study the practical implications of this learning problem, since it has immediate applications to fields like epidemiology. For instance, do the sample complexity bounds shown in this paper lead to practical algorithms that can be used in field studies? Finally, while we assumed that the nodes we inspect are a random sample from an underlying distribution (in the spirit of PAC learning), a different direction is to consider an active learning approach in which the algorithm can choose the nodes it wants to inspect. It is plausible that this additional selectivity can help reduce the size of the training set significantly in some situations. In fact, in, for example, epidemiological applications, we often see a combination of the two approaches: inspect a random sample but also inspect some selected nodes, such as those that are at high risk of contracting the disease because they neighbor infected nodes from the random sample. Such hybrid models of practical significance constitute an interesting direction for future research.

References

Abrahao, B.; Chierichetti, F.; Kleinberg, R.; and Panconesi, A. 2013. Trace complexity of network inference. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 491–499. ACM.

Alon, N.; Ben-David, S.; Cesa-Bianchi, N.; and Haussler, D. 1997. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM (JACM)*, 44(4): 615–631.

Bartlett, P. L.; and Mendelson, S. 2002. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov): 463–482.

Bourigault, S.; Lamprier, S.; and Gallinari, P. 2016. Representation learning for information diffusion through social networks: an embedded cascade model. In *Proceedings of the Ninth ACM international conference on Web Search and Data Mining*, 573–582. ACM.

Chen, W.; Collins, A.; Cummings, R.; Ke, T.; Liu, Z.; Rincon, D.; Sun, X.; Wang, Y.; Wei, W.; and Yuan, Y. 2011. Influence maximization in social networks when negative opinions may emerge and propagate. In *Proceedings of the 2011 siam international conference on data mining*, 379–390. SIAM.

Chen, W.; Wang, C.; and Wang, Y. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1029–1038. ACM.

Chen, W.; Wang, Y.; and Yang, S. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 199–208. ACM.

Cheng, J.; Adamic, L.; Dow, P. A.; Kleinberg, J. M.; and Leskovec, J. 2014. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, 925–936. ACM.

Chierichetti, F.; Liben-nowell, D.; and Kleinberg, J. M. 2011. Reconstructing patterns of information diffusion from incomplete observations. In *Advances in neural information processing systems*, 792–800.

Conitzer, V.; Panigrahi, D.; and Zhang, H. 2020. Learning Opinions in Social Networks. In *International Conference on Machine Learning*, 2122–2132. PMLR.

Daneshmand, H.; Gomez-Rodriguez, M.; Song, L.; and Schölkopf, B. 2014. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *International Conference on Machine Learning*, 793–801.

Daniely, A.; Sabato, S.; Ben-David, S.; and Shalev-Shwartz, S. 2015. Multiclass learnability and the erm principle. *The Journal of Machine Learning Research*, 16(1): 2377–2404.

Du, N.; Liang, Y.; Balcan, M.; and Song, L. 2014. Influence function learning in information diffusion networks. In *International Conference on Machine Learning*, 2016–2024.

Du, N.; Song, L.; Yuan, M.; and Smola, A. J. 2012. Learning networks of heterogeneous influence. In *Advances in Neural Information Processing Systems*, 2780–2788.

Gomez Rodriguez, M.; Balduzzi, D.; and Schölkopf, B. 2011. Uncovering the Temporal Dynamics of Diffusion Networks. In *28th International Conference on Machine Learning (ICML 2011)*, 561–568. International Machine Learning Society.

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

Gruhl, D.; Guha, R.; Liben-Nowell, D.; and Tomkins, A. 2004. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, 491–501. ACM.

Guille, A.; and Hacid, H. 2012. A predictive model for the temporal dynamics of information diffusion in online social networks. In *Proceedings of the 21st international conference on World Wide Web*, 1145–1152. ACM.

Hanneke, S. 2016. The optimal sample complexity OF PAC learning. *The Journal of Machine Learning Research*, 17(1): 1319–1333.

He, X.; Xu, K.; Kempe, D.; and Liu, Y. 2016. Learning influence functions from incomplete observations. In *Advances in Neural Information Processing Systems*, 2073–2081.

Kalimeris, D.; Singer, Y.; Subbian, K.; and Weinsberg, U. 2018. Learning diffusion using hyperparameters. In *International Conference on Machine Learning*, 2425–2433.

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

Kleinberg, J. 2000. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, 163–170.

Li, C.; Ma, J.; Guo, X.; and Mei, Q. 2017. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th international conference on World Wide Web*, 577–586. International World Wide Web Conferences Steering Committee.

Liben-Nowell, D.; and Kleinberg, J. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7): 1019–1031.

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

Narasimhan, H.; Parkes, D. C.; and Singer, Y. 2015. Learnability of influence in networks. In *Advances in Neural Information Processing Systems*, 3186–3194.

Pollard, D. 2012. *Convergence of stochastic processes*. Springer Science & Business Media.

Saito, K.; Ohara, K.; Yamagishi, Y.; Kimura, M.; and Motoda, H. 2011. Learning diffusion probability based on node attributes in social networks. In *International Symposium on Methodologies for Intelligent Systems*, 153–162. Springer.

Talagrand, M. 1994. Sharper bounds for Gaussian and empirical processes. *The Annals of Probability*, 28–76.

Tang, Y.; Xiao, X.; and Shi, Y. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 75–86. ACM.

Valiant, L. G. 1984. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 436–445. ACM.

Vapnik, V. 2013. *The nature of statistical learning theory*. Springer science & business media.

Wang, J.; Zheng, V. W.; Liu, Z.; and Chang, K. C.-C. 2017. Topological recurrent neural network for diffusion prediction. In *2017 IEEE International Conference on Data Mining (ICDM)*, 475–484. IEEE.

Weng, L.; Menczer, F.; and Ahn, Y.-Y. 2013. Virality prediction and community structure in social networks. *Scientific reports*, 3(1): 1–6.

A Proof Sketches of the Lower Bounds

Here we briefly describe the hard instances for each of the lower bounds. We start from the less complex $\Omega(\rho)$ bound. Consider n unconnected nodes, each with a distinct feature combination. Suppose all nodes are susceptible (so $\mathcal{H}_s$ is a singleton with VC dimension $d_s = 0$), and the set of generators is a uniformly random subset of all nodes. Here, a node adopts the influence (i.e., has label 1) iff it is a generator, and whether a node adopts the influence is independent of everything else. It is easy to show that learning anything nontrivial requires $\Omega(n) = \Omega(\rho)$ samples.

Now we proceed to the more complex constructions. First consider the $\Omega(d_s \log n)$ bound. The plan is to prove an $\Omega(\log n)$ bound for the case of $d_s = 1$, and then lift the construction to give $\Omega(d_s \log n)$ by making d_s parallel copies of the construction. The construction is illustrated in Figure 3. Suppose we care about k nodes (we will see below that our construction requires that $n > k$). Ideally, we would like to let all these nodes be susceptible, fix a single node (say node 1) as a generator, and partition all nodes into 2 disjoint cliques (so $\rho = 2$). Each node is in one of the two cliques independently and uniformly at random. Then, a node adopts the influence (i.e., has label 1) iff it is in the same clique as node 1, which happens with probability $1/2$ independent of everything else. This would give a lower bound of $\Omega(k)$. However, this idea implemented in the straightforward way requires edges to be heterogeneous too. In particular, we want only edges within the same clique to exist, which is impossible under the symptomatic spread model.

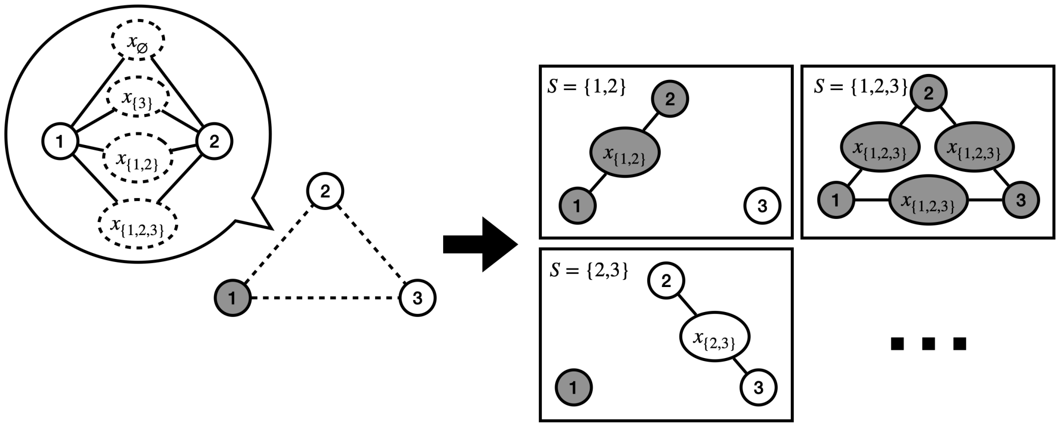

Figure 3: Example construction for the $\Omega(d_s \log n)$ bound using extended edges when $k = 3$. There are $2^3 = 8$ groups of edges, one for each partition $(S, \bar{S})$. The left graph is the complete network, where node 1 (the dark node) is the only generator. Depending on S, the complete network may realize into different sub-networks, out of which we show 3 examples on the right (corresponding to $S = \{1, 2\}$, $S = \{2, 3\}$, and $S = \{1, 2, 3\}$). The dark nodes are those with label 1 (because they are connected to node 1).

In order to circumvent the above issue, we simulate heterogeneous edges by placing a dummy node, whose label we do not care about, in the middle of every edge. We call such a gadget an extended edge. Then effectively, an extended edge exists iff the corresponding dummy node is susceptible. Moreover, we create 2^k groups of extended edges, one for each partition of the k nodes into two cliques. Within each group, all extended edges share the same feature combination unique to this group. Then we can make precisely one group of extended edges exist, by making the corresponding feature combination susceptible (a susceptibility hypothesis class allowing precisely this has VC dimension $d_s = 1$). The number of extended edges (and hence dummy nodes) in the above implementation is at most $2^k \cdot k^2$, so the total number of nodes is $n = O(2^{O(k)})$, which gives a lower bound of $\Omega(k) = \Omega(\log n)$.

Finally, consider the $\Omega(d_g \log n)$ bound. Again, we show an $\Omega(\log n)$ bound for the $d_g = 1$ case, and then lift to $\Omega(d_g \log n)$ by making d_g parallel copies. The construction is illustrated in Figure 4. The idea is to create 2^k cliques, where k is the number of feature combinations that matter. All nodes are susceptible (so $d_s = 0$), and among the k feature combinations, exactly one is a generator feature combination (a generator hypothesis class allowing exactly this has VC dimension $d_g = 1$). Suppose the r-th feature combination is the generator feature combination, and the cliques are numbered from 0 to $2^k - 1$. We would like each clique i to have label 1 (i.e., to contain a generator) iff the r-th digit in the binary representation of i is 1. Then, choosing r uniformly at random gives a lower bound of $\Omega(k)$. In order to implement this, for each $j \in [k]$, we add a node with feature combination x_j in the i-th clique iff the j-th digit of i is 1. We also add a node to clique 0 with a dummy feature combination x_0, which is never a generator feature combination, since otherwise clique 0 would be empty. The number of nodes in this construction is at most $n \leq 2^k \cdot k$, which implies a lower bound of $\Omega(k) = \Omega(\log n)$.

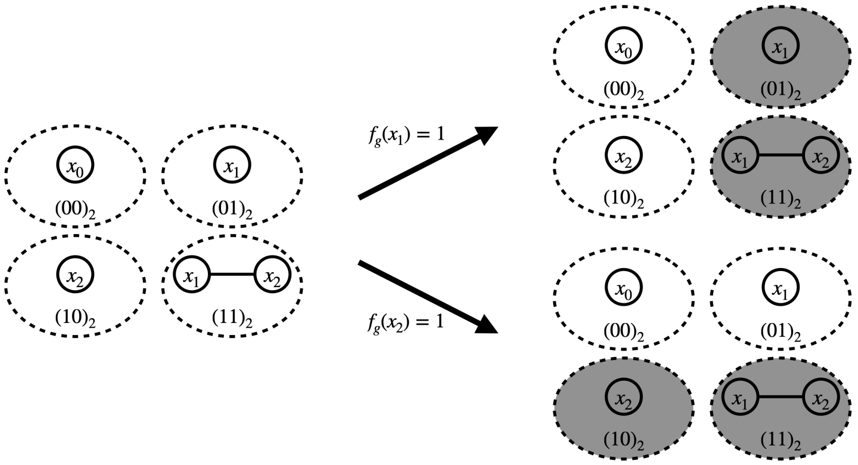

Figure 4: Example construction for the $\Omega(d_g \log n)$ bound: the $k = 2$ case with $2^2 = 4$ cliques. The left graph illustrates the complete network in the construction. When x_1 is the generator feature combination, the construction realizes into the top right network, where the dark cliques are those with label 1. Similarly, when x_2 is the generator feature combination, the construction realizes into the bottom right network. These are the only two possible realizations for the construction.

B Omitted Proofs from Section 3

Proof of Lemma 9. Suppose $\mathcal{F} = \bigcup_{i\in[k]} \mathcal{F}_i$ shatters a set A where $|A| = D$. We only need to show that $D = O(d + \log k)$. By the Sauer-Shelah lemma (Lemma 8), $\mathcal{F}_i^{\cap A} = \{S \cap A \mid S \in \mathcal{F}_i\}$ (that is, $\mathcal{F}_i$ projected to A) has cardinality at most

$$|\mathcal{F}_i| \le (2e|A|/d)^d = (2eD/d)^d,$$

and $\mathcal{F}|_A = \bigcup_{i\in[k]} \mathcal{F}_i|_A$ has cardinality at most

$$|\mathcal{F}^{\cap A}| \le \sum_{i\in[k]} |\mathcal{F}_i^{\cap A}| \le \sum_{i\in[k]} (2eD/d)^d = k(2eD/d)^d.$$

Since A is shattered by $\mathcal{F}$, we have

$$2^D = 2^{|A|} = |\mathcal{F}^{\cap A}| \le k(2eD/d)^d,$$

which immediately gives

$$D = O(d \log((\log k)/d) + \log k).$$

Moreover, when $d = \Omega(\log k)$, the above bound becomes $O(d) = O(d + \log k)$. Otherwise (i.e., when $d = o(\log k)$), let $t = (\log k)/d = \omega(1)$. The above bound then becomes $O(d \log t + dt) = O(dt) = O(\log k) = O(d + \log k)$. So putting the two cases together, we conclude that $D = O(d + \log k)$, which implies $d(\mathcal{F}) = O(d + \log k)$. □

Proof of Lemma 9. Suppose $\mathcal{F} = \bigcup_{i\in[k]} \mathcal{F}_i$ shatters a set A where $|A| = D$. We only need to show that $D = O(d + \log k)$. By the Sauer-Shelah lemma (Lemma 8), $\mathcal{F}_i^{\cap A} = \{S \cap A \mid S \in \mathcal{F}_i\}$ (that is, $\mathcal{F}_i$ projected to A) has cardinality at most

$$|\mathcal{F}_i| \le (2e|A|/d)^d = (2eD/d)^d,$$

and $\mathcal{F}|_A = \bigcup_{i\in[k]} \mathcal{F}_i|_A$ has cardinality at most

$$|\mathcal{F}^{\cap A}| \le \sum_{i\in[k]} |\mathcal{F}_i^{\cap A}| \le \sum_{i\in[k]} (2eD/d)^d = k(2eD/d)^d.$$

Since A is shattered by $\mathcal{F}$, we have

$$2^D = 2^{|A|} = |\mathcal{F}^{\cap A}| \le k(2eD/d)^d,$$

which immediately gives

$$D = O(d \log((\log k)/d) + \log k).$$

Moreover, when $d = \Omega(\log k)$, the above bound becomes $O(d) = O(d + \log k)$. Otherwise (i.e., when $d = o(\log k)$), let $t = (\log k)/d = \omega(1)$. The above bound then becomes $O(d \log t + dt) = O(dt) = O(\log k) = O(d + \log k)$. So putting the two cases together, we conclude that $D = O(d + \log k)$, which implies $d(\mathcal{F}) = O(d + \log k)$. □

Proof of Theorem 7. We first prove the upper bound on the VC dimenion of $\mathcal{H}_a$, the class of all possible adoption patterns. First we show the easy part, i.e.,

$$d(\mathcal{H}_a) = O((d_s + d_g)\log n).$$

Suppose $d(\mathcal{H}_a) = D$, and moreover, for some set $A \subseteq V$ of size $|A| = D$, $\mathcal{H}_a$ shatters A. In other words, $|\mathcal{H}_a^{\cap A}| = 2^{|A|} = 2^D$. We only need to show $D = O((d_s + d_g)\log n)$, which requires upper bounding $|\mathcal{H}_a^{\cap A}|$. Since every $f_a \in \mathcal{H}_a$ is induced by a pair (f_g, f_s) where $f_g \in \mathcal{H}_g$ and $f_s \in \mathcal{H}_s$, we have

$$|\mathcal{H}_a^{\cap A}| \le |\mathcal{H}_g^{\cap A}| \cdot |\mathcal{H}_s^{\cap A}|.$$

By the Sauer-Shelah lemma (Lemma 8),

$$|\mathcal{H}_g^{\cap A}| \le \left(\frac{2eD}{d_g}\right)^{d_g} \quad \text{and} \quad |\mathcal{H}_s^{\cap A}| \le \left(\frac{2eD}{d_s}\right)^{d_s},$$

which implies

$$|\mathcal{H}_a^{\cap A}| \le |\mathcal{H}_g^{\cap A}| \cdot |\mathcal{H}_s^{\cap A}| \le \left(\frac{2eD}{d_g}\right)^{d_g}\left(\frac{2eD}{d_s}\right)^{d_s} \le (2eD)^{d_g + d_s}.$$

Since $A \subseteq V$, we have $|A| \le n$, and therefore

$$|\mathcal{H}_a^{\cap A}| \le (2en)^{d_g + d_s},$$

which gives

$$D \le \log|\mathcal{H}_a^{\cap A}| = O((d_g + d_s)\log n).$$

Now we prove the harder part of the upper bound for the VC dimension, i.e.,

$$d(\mathcal{H}_a) = O(\rho + d_s \log n).$$

The key step of the proof is to apply Lemma 9 to $\mathcal{H}_a$. Observe that

$$\mathcal{H}_a \subseteq \bigcup_{f_s \in \mathcal{H}_s} \mathcal{H}(G(x, f_s)) = \bigcup_{f'_s \in \mathcal{H}_s^{\cap V}} \mathcal{H}(G(x, f'_s)).$$

And the above is true even if there is no structure in the generator class $\mathcal{H}_g$ (and therefore all subsets of V might be the set of generators). By the Sauer-Shelah lemma (Lemma 8),

$$|\mathcal{H}_s^{\cap V}| \le \left(\frac{2e|V|}{d_s}\right)^{d_s} = \left(\frac{2en}{d_s}\right)^{d_s}.$$

Moreover, by Lemma 5, for each $f'_s \in \mathcal{H}_s^{\cap V}$,

$$d(\mathcal{H}(G(x, f'_s))) = \rho(G(x, f'_s)) \le \rho(G, x, \mathcal{H}_s) = \rho.$$

Now we can apply Lemma 9 to $\mathcal{H}_a$, with parameters

$$k = |\mathcal{H}_s^{\cap V}| \le \left(\frac{2en}{d_s}\right)^{d_s} \quad \text{and} \quad d = \rho,$$

which gives

$$d(\mathcal{H}_a) = O(d + \log k) = O(\rho + d_s \log n).$$

This finishes the proof of the upper bound on $d(\mathcal{H}_a)$.

Now we show the existence of a learning algorithm with target error rate ε and failure probability δ using the desired number of samples

$$m = O\left(\frac{(d_s \log n + \min(\rho, d_g \log n)) \cdot \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}\right).$$

By the VC theorem (Theorem 2), with m samples, any algorithm that finds an adoption hypothesis $h_a \in \mathcal{H}_a$ that is consistent with the samples satisfies the above condition. Therefore, a simple (but inefficient) algorithm is to enumerate a generator hypothesis $h'_g \in \mathcal{H}_g^{\cap V}$ and a susceptibility hypothesis $h'_s \in \mathcal{H}_s^{\cap V}$, and check whether the induced adoption hypothesis by h'_g and h'_s is consistent with the m samples. Such enumeration, implemented in the straightforward way, could take $\Omega(2^{2n})$ time. To design a more efficient algorithm, we instead take the following steps: first we generate $\mathcal{H}_s^{\cap V}$ in time $O(|\mathcal{H}_s^{\cap V}|) = O(n^{d_s})$ by exploiting the ERM oracle to $\mathcal{H}_s$. Then, for each $h'_s \in \mathcal{H}_s^{\cap V}$, we check whether there exists an $h_g \in \mathcal{H}_g$ such that the adoption hypothesis induced by h'_s and h_g is consistent with the m samples. This can be done using one call to the ERM oracle for $\mathcal{H}_g$ (when $d_g \log n \le \rho$), or one call to the ERM algorithm for the implicit hypothesis class associated with $G(x, h'_s)$ (when $d_g \log n > \rho$). The overall algorithm runs in time $O(n^{d_s} \cdot \mathrm{poly}(n))$, which is polynomial in all other relevant parameters when d_s is constant.

To efficiently generate $\mathcal{H}_s^{\cap V}$, we apply the following lemma.

Lemma 18. *For any hypothesis class $\mathcal{H}$ and set S where $|S| = s$, $\mathcal{H}^{\cap S}$ can be explicitly listed using $O(s \cdot |\mathcal{H}^{\cap S}|)$ calls to the ERM oracle for $\mathcal{H}$, with $O(s \cdot |\mathcal{H}^{\cap S}|)$ additional operations.*

Proof. Without loss of generality suppose $S = [s]$. Consider the following binary tree of depth $s + 1$, constructed from $\mathcal{H}^{\cap S}$ (i.e., the prefix tree of $\mathcal{H}^{\cap S}$):
- Each node in the tree corresponds bijectively to a prefix of a set in $\mathcal{H}^{\cap S}$. That is, each node is associated with a unique member of the following set
$$\{(i, T \cap [i]) \mid T \in \mathcal{H}^{\cap S}, i \in \{0, 1, \ldots, s\}\}.$$
Below we will use the associated pair to refer to a node in the tree.
- The root of the tree is $(0, \emptyset)$. The parent of any node (i, T) where $i > 0$ is $(i - 1, T \cap [i - 1])$.

Observe that the leaves of the above tree correspond bijectively to members of $\mathcal{H}^{\cap S}$, and the size of the tree is $O(s \cdot |\mathcal{H}^{\cap S}|)$. So, listing members of $\mathcal{H}^{\cap S}$ only requires efficiently traversing the above tree.

In order to do that, observe that a pair (i, T) is a node of the tree, iff there exists $h \in \mathcal{H}$, such that for all $j \in [i]$, $h(j) = \mathbb{I}[j \in T]$. So, to check whether (i, T) is a node of the tree, we only need to call the ERM oracle to $\mathcal{H}$ on $\{(j, \mathbb{I}[j \in T])\}_{j\in[i]}$. Then, (i, T) is a node of the tree iff the empirical risk minimizer returned by the oracle is perfectly consistent with the input data points. Now to traverse the tree, we simply start from the root $(0, \emptyset)$, and check if the two potential children of the root are in fact members of the tree. Then we recursively traverse the subtree rooted at each child of the root. This can be done using $O(s \cdot |\mathcal{H}^{\cap S}|)$ ERM calls and $O(s \cdot |\mathcal{H}^{\cap S}|)$ additional operations. □

Recall that the learning algorithm takes as input m labeled samples $\{(x_i, y_i)\}_{i\in[m]}$, and outputs a hypothesis h that approximates the actual adoption pattern f_a. Given Lemma 18, the overall algorithm works in the following way.
1. Apply Lemma 18 to generate $\mathcal{H}_s^{\cap V}$ in time $O(n \cdot |\mathcal{H}_s^{\cap V}|) = O(n^{d_s+1})$.
2. For each $h'_s \in \mathcal{H}_s^{\cap V}$, perform the following steps.
   (a) Let $G' \leftarrow G(x, h'_s)$, and $S \leftarrow \{u \in V \mid \exists i \in [m] : y_i = 1 \text{ and } x_i \to_{G'} u\}$. That is, G' is the sub-network induced by all susceptible nodes according to h'_s, and S is the set of vertices reachable in G' from some sample node x_i with label $y_i = 1$.
   (b) If there exists $i \in [m]$, such that $x_i \in S$ and $y_i = 0$, then abandone the current h'_s and continue with the next susceptibility hypothesis in $\mathcal{H}_s^{\cap V}$.
   (c) Let $P \leftarrow \{u \in S \mid \forall v \in S, v \neq u : v \not\to_{G'} u\}$, $N \leftarrow \{u \in V \mid \exists i \in [m] : u \to_{G'} x_i, y_i = 0\}$. That is, P is the set of vertices in S not reachable from any other vertices in S, and N is the set of vertices that can reach some sample x_i with label $y_i = 0$.
   (d) Call the ERM oracle to $\mathcal{H}_g$ with data points $\{(u, 1) \mid u \in P\} \cup \{(u, 0) \mid u \in N\}$, and let h_g be the output hypothesis. If h_g is consistent with all input data points, then output the adoption pattern h_a induced by h_g and h'_s and terminate the algorithm. Otherwise, abandon the current h'_s and continue with the next susceptibility hypothesis in $\mathcal{H}_s^{\cap V}$.

It is easy to check the above algorithm in fact runs in time $O(n^{d_s} \cdot \mathrm{poly}(n))$, and outputs a hypothesis adoption pattern h_a consistent with all labeled samples. □

Proof of Theorem 10. We first prove the $\Omega(d_s \log n)$ bound. We start with a family of instances where $d_s = 1$, $d_g = 0$, and $\rho = 2$, and to achieve constant ε and δ one needs $\Omega(\log n)$ samples. The vertices are divided into two categories: "real" ones and "dummy" ones. Let $R = [k]$ for some parameter k be the real vertices — these are the vertices whose labels we care about. The idea is to subdivide R into two random connected components, and each real node is independently in each of the two components with probability $1/2$. In one of the two components, every real node has label 1, and in the other, every node has label 0. So, effectively the label of a real node (except for node 1 — see below) is an unbiased coin independent of the labels of all other real vertices.

In order to implement the above idea, we create 2^k groups of edges and dummy vertices, one for each subset of R. Suppose the two components we want to create is C and $R \setminus C$. Let D_C and E_C be the dummy vertices and edges corresponding to $(C, R \setminus C)$. For each (unordered) pair of vertices u and v in the same component (i.e., $\{u, v\} \subseteq C$ or $\{u, v\} \subseteq R \setminus C$), we add the following dummy vertices and edges:
- a dummy node $d^C_{u,v} \in D_C$, and
- two undirected edges $(u, d^C_{u,v}) \in E_C$ and $(v, d^C_{u,v}) \in E_C$.

Then $V = R \cup \bigcup_{C\subseteq R} D_C$, and $E = \bigcup_{C\subseteq R} E_C$. Note that for each $C \subseteq R$, $|D_C| \leq k^2$, so we have
$$n = |V| \leq k + 2^k \cdot k^2,$$
and as a result, $k = \Omega(\log n)$.

Observe that among the dummy vertices, if those in D_C are susceptible and all other dummy vertices are not, then in the induced sub-network R is divided exactly into two components, C and $R \setminus C$. To implement this, let $\mathcal{X} = \{x_0, x_1\} \cup \{x_C \mid C \subseteq R\}$. Moreover,
- real node $1 \in R$ has features x_1 and all other vertices in R have feature x_0, and
- all dummy vertices in D_C have features x_C.

The generator hypothesis class consists of a single concept f_g, where $f_g(x_1) = 1$ and $f_g(x) = 0$ for any $x \in \mathcal{X} \setminus \{x_1\}$. Clearly $d_g = d(\mathcal{H}_g) = 0$. The susceptibility hypothesis class has 2^k members $\mathcal{H}_s = \{f_s^C \mid C \subseteq R\}$. For any $C \subseteq R$, f_s^C assigns 1 to x_0, x_1, and x_C,and 0 to all other feature combinations. One can check that $d_s = d(\mathcal{H}_s) = 1$.

Now consider the learning problem. Let the population distribution $\mathcal{D}$ be uniform over R. Observe that the adoption pattern f_a^C induced by f_g and f_s^C satisfies: for any $u \in R$, $f_a^C(u) = 1$ iff $1 \in C$ and $u \in C$, or $1 \notin C$ and $u \notin C$. Suppose the actual susceptibility concept is uniformly at random from $\mathcal{H}_s$. That is, $f_s = f_s^C$ for a uniformly random subset C of R. Then except for node 1, any $u \in R$ is independently in the same component as 1 with probability $1/2$, and so $f_a(u) = 1$ with probability $1/2$. So, for the learning algorithm, if $u \in R$ is not among the labeled samples, then the algorithm has absolutely no knowledge about $f_a(u)$, and has to make mistakes with probability $1/2$. Now in order to achieve target error rate $1/4$, the algorithm has to observe about $1/2$ of the real vertices, and the number of samples required is $\Omega(k) = \Omega(\log n)$. This leads to an $\Omega(\log n)$ lower bound when $d_g = 0$ and $d_s = 1$. Now to generalize the above argument for large d_s, one can simply make independent copies of the above hard instance, where d_s is the number of such copies. Then, essentially the same argument gives a lower bound of $\Omega(d_s \log n)$.

We now prove the $\Omega(d_g \log n)$ bound. Again, we start with a family of instances where $d_g = 1$ and $d_s = 0$, and to achieve constant ε and δ one needs $\Omega(\log n)$ samples. Let the feature space be $\mathcal{X} = \{0, \dots, k\}$. Consider an undirected network consisting of 2^k cliques (numbered 0 through $2^k - 1$) and $n \le k \cdot 2^k$ vertices. The susceptibility hypothesis class $\mathcal{H}_s$ consists of a single concept f_s, which assigns 1 to any feature combination $x \in \mathcal{X}$, i.e., $\forall x \in \mathcal{X}$, $f_s(x) = 1$. Clearly $d_s = d(\mathcal{H}_s) = 0$. The generator hypothesis class $\mathcal{H}_g$ consists of k concepts $\{f_g^1, \dots, f_g^k\}$. For each $i \in [k]$, f_g^i assigns 1 to the feature combination $i \in \mathcal{X}$, and 0 to all other feature combinations. One can check that $d_g = d(\mathcal{H}_g) = 1$.

We now describe the vertices V and edges E. For each feature combination $i \in \mathcal{X}$, and each $j \in \{0, \dots, 2^k - 1\}$, we create a node with feature combination i in clique j iff the i-th digit in the binary representation of j is 1. We also create a node in clique 0 with feature combination $0 \in \mathcal{X}$, simply to make clique 0 nonempty (note that $f_g^i(0) = 0$ for all $i \in [k]$). Since every feature combination corresponds to at most 2^k vertices, the total number of vertices is $n \le k \cdot 2^k$. Then E is the set of all intra-clique edges.

Now consider the learning problem. Let $V' \subseteq V$ be any set containing exactly one node from each of the k cliques, and the population distribution $\mathcal{D}$ be uniform over V'. Suppose the ground truth generator concept f_g is chosen uniformly at random from $\mathcal{H}_g$, so $f_g = f_g^r$ where r is a uniformly random integer from $[k]$. When we draw a labeled sample (x, y), the label is $y = f_a(x) = 1$ iff the r-th digit in the number of the clique containing x is 1. Observe that the posterior distribution of r after observing some labeled samples is always uniform over a subset of $[k]$. We argue that in order to achieve error probability less than $1/4$, one has to be completely sure about r. In fact, even if we know r is (wlog) either 1 or 2, each with probability $1/2$, we still make mistakes with probability $1/4$. This is because for a random node drawn from $\mathcal{D}$, with probability $1/2$, exactly one of the first two digits in the number of the clique containing the node is 1. When this happens, we have absolutely no knowledge about the label of the node, and therefore we make mistakes with probability $1/2$. This gives an overall error probability of $1/4$. Now it is easy to show that in order to learn the precise value of r with constant (say $9/10$) probability, one needs $\Omega(k) = \Omega(\log n)$ samples. This leads to an $\Omega(\log n)$ lower bound when $d_g = 1$ and $d_s = 0$. Now to generalize the above argument for large d_g, again one can simply make independent copies of the above hard instance, where d_g is the number of such copies. Then, essentially the same argument gives a lower bound of $\Omega(d_g \log n)$.

Finally we prove the $\Omega(\rho)$ bound. The family of hard instances here is considerably simpler compared to those for the previous two bounds. Let $\mathcal{X} = V = [n]$ and $E = \emptyset$, and as a result, $\rho = n$. Each node $i \in V$ has a distinct feature combination $x(i) = i$. Let f_s be the only member of $\mathcal{H}_s$, which assigns 1 to every feature combination (i.e., all vertices are always susceptible). Let $\mathcal{H}_g$ be the set of all possible mappings from $\mathcal{X}$ to $\{0, 1\}$ (i.e., any node can be a generator).

Now consider the learning problem. Let $\mathcal{D}$ be uniform over V. Suppose the actual generator concept f_g is chosen uniformly at random from $\mathcal{H}_g$. In other words, the induced adoption pattern f_a assigns label 1 to each node independently with probability $1/2$. It is easy to show that the algorithm has to observe $\Omega(n) = \Omega(\rho)$ samples in order to achieve a constant error rate. This gives the $\Omega(\rho)$ lower bound. □

Proof of Theorem 11. The idea is to construct a graph consisting of two parallel chains that are randomly swapped in each layer. In one of the chains, all vertices have label 1, and in the other, all vertices have label 0. Because of the random swaps, it is impossible for the learning algorithm to tell which chain a node is in. As a result, before observing the label of one of the two vertices in the same layer, the learning algorithm has no knowledge about the label of a node.

Let the feature space be $\mathcal{X} = \{x_1, x_2, x_3\}$. The generator hypothesis class $\mathcal{H}_g$ consists of a single concept f_g, where $f_g(x_1) = 1$ and $f_g(x_2) = f_g(x_3) = 0$. The susceptibility hypothesis class $\mathcal{H}_s$ also has a single member f_s, where $f_s(x_1) = f_s(x_2) = 1$ and $f_s(x_3) = 0$. In other words, a node with features x_1 is both a generator and susceptible, a node with features x_2 is a generator but not susceptible, and a node with features x_3 is neither a generator nor susceptible. The vertices are divided into "real" ones and "dummy" ones. Let $R = \{0, 1, \dots, 2k\}$ for some parameter k be the real vertices — these are the ones whose labels we care about. We say node 0 is in layer 0, and for each $i \in [k]$, vertices $2i - 1$ and $2i$ are in layer i. The feature mapping assigns x_1 to node 0, and x_2 to all other vertices in R. The idea is to make node 0 a generator, and create a random edge from 0 either to 1 or 2 with equal probability. Then between layer i and $i + 1$, we either (with probability $1/2$) create edges from $2i-1$ to $2i+1$ and from $2i$ to $2i+2$ (i.e., we do not swap the two chains), or (with probability $1/2$) create edges from $2i-1$ to $2i+2$ and from $2i$ to $2i+1$ (i.e., we swap the two chains). So as a result, in layer i, either $2i-1$ or $2i$ has label 1, and this is independent of all other labels of vertices in R.

To implement the above idea, let

$$D_0 = \{d_0^0, d_1^0\}$$

be the dummy vertices associated with layer 0, and

$$E_0 = \{(0, d_0^0), (0, d_0^1), (d_0^0, 1), (d_1^0, 1)\}$$

be the edges associated with layer 0. For each $i \in [k-1]$, let

$$D_i = \{d_{00}^i, d_{01}^i, d_{10}^i, d_{11}^i\}$$

be the dummy vertices associated with layer i, and

$$E_i = \{(2i-1, d_{00}^i), (2i-1, d_{01}^i), (2i, d_{10}^i), (2i, d_{11}^i), (d_{00}^i, 2i+1), (d_{01}^i, 2i+2), (d_{10}^i, 2i+1), (d_{11}^i, 2i+2)\},$$

be the edges associated with layer i. Then, in layer 0, the feature mapping assigns $x(d_0^0) = x_2$ and $x(d_1^0) = x_3$ with probability $1/2$, and $x(d_1^0) = x_2$ and $x(d_0^0) = x_3$ with probability $1/2$. Similarly, in layer $i \in [k-1]$, with probability $1/2$, $x(d_{00}^i) = x(d_{11}^i) = x_2$ and $x(d_{10}^i) = x(d_{01}^i) = x_3$, and with probability $1/2$, $x(d_{01}^i) = x(d_{10}^i) = x_2$ and $x(d_{00}^i) = x(d_{11}^i) = x_3$. This implements precisely the above idea. Also note that the total number of vertices is

$$n = |V| = |R| + |D_0| + \sum_{i \in [k-1]} |D_i| = 2k + 1 + 2 + 4(k-1) = 6k - 1.$$

Now consider the learning problem. Let $\mathcal{D}$ be uniform over $[2k] \subseteq R$. Observe that for any $i \in [k]$, in order to say anything about $f_a(2i-1)$ or $f_a(2i)$, the algorithm needs to observe at least one labeled sample from the same layer. As a result, to achieve error rate $1/4$ and failure probability $1/4$, the number of samples needed is $\Omega(k) = \Omega(n)$. This gives the desired lower bound. □

C Omitted Proofs from Section 4

Proof of Theorem 12. We first show that $d(\mathcal{H}_a) = O(d_s + d_g \log n)$. The plan is to apply Lemma 9 to $\mathcal{H}_a$. For each $f_g \in \mathcal{H}_g$, let V_{f_g} be the set of all vertices in V that is reachable from some node u where $f_g(u) = 1$. That is,

$$V_{f_g} = \{v \in V \mid \exists u \in V : f_g(u) = 1, u \to_G v\}.$$

Suppose f_g and f_s induce f_a. Observe that for any $u \in V$, $f_a(u) = 1$ iff $u \in V_{f_g}$ and $f_s(u) = 1$. Let $\mathcal{H}_{a,f_g}$ be the family of all adoption patterns induced by $f_g \in \mathcal{H}_g$ and all concepts in $\mathcal{H}_s$, i.e.,

$$\mathcal{H}_{a,f_g} = \{f_s \cap V_{f_g} \mid f_s \in \mathcal{H}_s\}.$$

By the construction of $\mathcal{H}_a^{f_g}$, one can check that

$$d(\mathcal{H}_{a,f_g}) \le d(\mathcal{H}_s) = d_s.$$

Observe that

$$\mathcal{H}_a = \bigcup_{f_g \in \mathcal{H}_g} \mathcal{H}_{a,f_g} = \bigcup_{f'_g \in \mathcal{H}_g^{\cap V}} \mathcal{H}_{a,f'_g}.$$

Then applying Lemmas 8 and 9, we have

$$d(\mathcal{H}_a) = O(d_s + \log |\mathcal{H}_g^{\cap V}|) = O(d_s + d_g \log n).$$

Now we show that $d(\mathcal{H}_a) = O(d_s + \rho)$. Again we apply Lemma 9. Suppose $d(\mathcal{H}_a) = D$, and $\mathcal{H}_a$ shatters $S \subseteq V$ where $|S| = D$, so $d(\mathcal{H}_a^{\cap S}) = D$. Recall that $\mathcal{H}(G)$ is the implicit hypothesis class associated with G, and $\rho = d(\mathcal{H}(G))$. Observe that regardless of $\mathcal{H}_g$,

$$\mathcal{H}_a^{\cap S} \subseteq \{f \cap f'_s \mid f \in \mathcal{H}(G), f'_s \in \mathcal{H}_s^{\cap S}\} = \bigcup_{f'_s \in \mathcal{H}_s^{\cap S}} \{f \cap S \cap f'_s \mid f \in \mathcal{H}(G)\}.$$

Now by Lemmas 8 and 9,

$$D = d(\mathcal{H}_a^{\cap S}) = O(d(\mathcal{H}(G)) + \log |\mathcal{H}_s^{\cap S}|) = O(\rho + d_s \log(D/d_s)).$$

This immediately gives

$$D = O(\rho + d_s \log(1 + \rho/d_s)).$$

In fact, this bound can be further simplified: when $\rho = O(d_s)$, the above bound becomes $O(d_s) = O(\rho + d_s)$. When $\rho = \omega(d_s)$, let $k = \rho/d_s$, and we have

$$D = O(d_s \cdot (k + \log(1+k))) = O(d_s \cdot k) = O(\rho) = O(\rho + d_s).$$

So in either case, we can conclude that

$$d(\mathcal{H}_a) = D = O(\rho + d_s).$$

Now consider algorithms for learning $\mathcal{H}_a$. We give below a simple algorithm that runs in time $O(n^{d_g+d_s}\mathrm{poly}(n))$ which achieves target error rate ε and failure probability δ using

$$m = O\left(\frac{(d_s + \min(\rho, d_g \log n)) \cdot \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}\right)$$

samples. In light of Theorem 2, we only need to find a hypothesis $h_a \in \mathcal{H}_a$ that is consistent with all m labeled samples. Recall that

$$\mathcal{H}_a = \bigcup_{f'_g \in \mathcal{H}_g^{\cap V}} \{f_s \cap V_{f'_g} \mid f_s \in \mathcal{H}_s\} = \bigcup_{f'_g \in \mathcal{H}_g^{\cap V}} \{f'_s \cap V_{f'_g} \mid f'_s \in \mathcal{H}_s^{\cap V}\}.$$

So, given Lemma 18, we can enumerate $h'_g \in \mathcal{H}_g^{\cap V}$ in time $O(n^{d_g})$ and $h'_s \in \mathcal{H}_s^{\cap V}$ in time $O(n^{d_s})$. We then output the hypothesis h_a induced by h'_g and h'_s if h_a is consistent with the labeled samples. □

Proof of Theorem 13. For the $\Omega(d_g \log n)$ and $\Omega(\rho)$ bounds, the family of hard instances for the same bounds in the proof of Theorem 10 work in exactly the same way (except that here f_s is the only member of $\mathcal{H}_s$, where $f_s(u) = 1$ for all $u \in V$) We therefore refrain from reiterating the same arguments. Below we prove the $\Omega(d_s)$ bound. Let $\mathcal{X} = V = [n]$, $E = \emptyset$, and f_g be the only member of $\mathcal{H}_g$ where $f_g(i) = 1$ for all $i \in [n]$. Moreover, let $\mathcal{H}_s$ be an arbitrary hypothesis class over $\mathcal{X}$ where $d(\mathcal{H}_s) = d_s$. Then for any $u \in V$, $f_a(u) = 1$ iff $f_s(u) = 1$, and the learning problem is equilavent to learning the susceptibility hypothesis class $\mathcal{H}_s$. By Theorem 2, it requires $\Omega(d_s)$ samples to achieve constant ε and δ in the above learning task. □

D Omitted Proofs from Section 5

Proof of Theorem 15. Given m labeled samples $\{(x_i, y_i)\}_{i \in [m]}$, the algorithm works in the following way.

1. Draw $\theta = \frac{100}{\varepsilon}\log(1/\delta)$ iid sample hypothesis classes $\{\mathcal{H}_k\}_{k \in [\theta]}$, where $\mathcal{H}_k = \mathcal{H}_{r_k}$ for $r_k \sim \mathcal{R}$.
2. For each $k \in [\theta]$, compute an empirical risk minimizer h_k in $\mathcal{H}_k$ with respect to the labeled samples $\{(x_i, y_i)\}$.
3. Output the following pointwise majority hypothesis h: for each $x \in \mathcal{X}$, $h(x) = 1$ iff

$$\sum_{k \in [\theta]} h_k(x) \geq \frac{\theta}{2}.$$

The proof relies on the following key lemma about the multiplicative relative error of empirical risk minimizers.

Lemma 19 ((Conitzer, Panigrahi, and Zhang 2020)). *Fix a feature space $\mathcal{X}$ and a hypothesis class $\mathcal{H}$ over $\mathcal{X}$. Fix any distribution $\mathcal{D}$ over $\mathcal{X}$, $f : \mathcal{X} \to \{0,1\}$, $\delta' > 0$ and $\varepsilon' > 0$. Moreover, suppose there is a hypothesis $h^* \in \mathcal{H}$ such that $\Pr_{x\sim\mathcal{D}}[h^*(x) \neq f(x)] \leq \varepsilon'/2$. Now, consider any empirical risk minimizer $h \in \mathcal{H}$ for at least $m' = O((d(\mathcal{H})\log(1/\varepsilon') + \log(1/\delta'))/\varepsilon')$ samples $\{(x_i, y_i)\}_{i\in[m']}$, i.e.,*

$$h \in \operatorname*{argmin}_{h' \in \mathcal{H}} \sum_{i \in [m']} \mathbb{I}[h'(x_i) \neq y_i],$$

where $y_i = f(x_i)$ for each $i \in [m']$. Then, with probability at least $1 - \delta'$, h has error rate at most ε', i.e., $\Pr_{x\sim\mathcal{D}}[h(x) \neq f(x)] \leq \varepsilon'$ with respect to f.

We first try to apply Lemma 19 to the sample hypothesis classes $\{\mathcal{H}_k\}$. Suppose $s^* \in \mathcal{S}$ is the ground truth state of the world, and r^* is the realized random mapping. s^* and r^* together induce the ground truth concept f_{s^*,r^*}. Recall that $(\mathcal{S}, \mathcal{X}, \mathcal{R})$ is $(\varepsilon_0, \delta_0)$-stable with respect to $\mathcal{D}$. So with probability at least $1 - \delta_0$, for all $k \in [\theta]$,

$$\Pr_{r_k \sim \mathcal{R}, x \sim \mathcal{D}}[f_{s^*,r^*}(x) \neq f_{s^*,r_k}(x)] \leq \varepsilon_0.$$

We condition on the above event from now on. Whenever the above does not hold (which happens with probability at most δ_0), we consider the algorithm to have failed.

For each $k \in [\theta]$, let e_k be the difference between f_{s^*,r^*} and f_{s^*,r_k}, i.e.,

$$e_k = \Pr_{x\sim\mathcal{D}}[f_{s^*,r^*}(x) \neq f_{s^*,r_k}(x)].$$

Observe that $\{e_k\}_{k\in[\theta]}$ are iid random variables in $[0,1]$, whose expectation does not exceed ε_0. We now apply Lemma 19 with different parameters (particularly, different $\varepsilon' = \varepsilon_k$) for each $\mathcal{H}_k$, to bound the error rate of h_k with respect to the ground truth f_{s^*,r^*}. For any $k \in [\theta]$, apply Lemma 19 with $m' = m$, $\delta' = \delta/3\theta$ and

$$\varepsilon' = \varepsilon_k = \max\left(2e_k, \frac{\varepsilon}{8}\cdot\left(1 + \frac{d(\mathcal{H}_k)}{\mathbb{E}_{r\sim\mathcal{R}}[d(\mathcal{H}_r)]}\right)\right).$$

Note that the condition of Lemma 19 is satisfied. In particular, there is a hypothesis f_{s^*,r_k} consistent with $\mathcal{H}_k$ that has error rate $e_k \le \varepsilon_k/2$. Then,

$$\begin{aligned}
&m' \\
&= O\left(\frac{d(\mathcal{H}_k)\log(1/\varepsilon') + \log(1/\delta')}{\varepsilon'}\right) \\
&\le O\left(\frac{((8+o(1))(d(\mathcal{H}_k)\log(1/\varepsilon) + \log(1/\delta))}{\varepsilon(1 + d(\mathcal{H}_k)/\mathbb{E}_{r\sim\mathcal{R}}[d(\mathcal{H}_r)])}\right) \\
&\le O\left(\frac{(8+o(1))(\mathbb{E}_{r\sim\mathcal{R}}[d(\mathcal{H}_r)]\log(1/\varepsilon) + \log(1/\delta))}{\varepsilon}\right) \\
&\le m,
\end{aligned}$$

where the last inequality holds when the constant in the choice of m is large enough. So, by Lemma 19, we get the following: with probability at least $1-\delta'$, h_k satisfies

$$\Pr_{x\sim\mathcal{D}}[f_{s^*,r^*}(x) \ne h_k(x)] \le \varepsilon_k.$$

Taking a union bound over $k \in [\theta]$, this inequality holds simultaneously for all h_k with probability at least

$$1 - \theta\cdot\delta' \ge 1 - \delta/3.$$

Again, we condition on the above event from now on, and consider the algorithm to have failed otherwise (which happens with probability at most $\delta/3$).

Observe that the expected error rate of $\{h_k\}_k$ is already low (i.e., on the order of ε). To be specific, note that $\{r_k\}_k$, and therefore $\{\mathcal{H}_k\}_k$, are still iid even if we contidion on the algorithm has not failed so far, which depends only on the randomness in the labeled sample nodes. As a result, for any $k \in [\theta]$, we have:

$$\begin{aligned}
&\mathbb{E}_{r_k}[\varepsilon_k] \\
&= \mathbb{E}_{r_k}\left[\max\left(2e_k, \frac{\varepsilon}{8}\left(1 + \frac{d(\mathcal{H}_k)}{\mathbb{E}_{r\sim\mathcal{R}}[d(\mathcal{H}_r)]}\right)\right)\right] \\
&\le \mathbb{E}_{r_k}\left[2e_k + \frac{\varepsilon}{8}\left(1 + \frac{d(\mathcal{H}_k)}{\mathbb{E}_{r\sim\mathcal{R}}[d(\mathcal{H}_r)]}\right)\right] \\
&\le 2\mathbb{E}_{r_k}[e_k] + \frac{\varepsilon}{8} + \varepsilon\cdot\frac{\mathbb{E}_{r_k}[d(\mathcal{H}_k)]}{8\mathbb{E}_{r\sim\mathcal{R}}[d(\mathcal{H}_r)]} \\
&\le 2\varepsilon_0 + \frac{\varepsilon}{4}.
\end{aligned}$$

Since $\varepsilon \ge C_1\cdot\varepsilon_0$, the above is upper bounded by $\varepsilon/3$ whenever $C_1 \ge 24$. But, each h_k may still have error rate larger than ε with probability larger than δ. Here, 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 $\varepsilon/2$. Observe that $\{\varepsilon_k\}_k$ are iid variables in $[0,1]$.[1] For small enough δ, by the multiplicative

[1]The above choice of ε_k itself may exceed 1. However, since ε_k is an upper bound of a probability, one can always truncate ε_k at 1, which does not increase the mean. We omit this in the proof for the sake of brevity.

Chernoff bound,

$$\begin{aligned}
&\Pr\left[\frac{1}{\theta}\sum_{k\in[\theta]}\varepsilon_k \geq \frac{\varepsilon}{2}\right] \\
=\ &\Pr\left[\frac{1}{\theta}\sum_{k\in[\theta]}\varepsilon_k \geq \left(1+\frac{1}{2}\right)\cdot\frac{\varepsilon}{3}\right] \\
\leq\ &\exp\left(-\frac{(1/2)^2\cdot(\theta\varepsilon/3)}{2+1/2}\right) \\
=\ &\exp\left(\frac{10\log(\delta)}{3}\right) \leq \frac{\delta}{3}.
\end{aligned}$$

We remark that the actual mean of ε_k may be smaller than $\varepsilon/3$, but that only makes the probability smaller. So with probability at least $1-\frac{\delta}{3}$,

$$\frac{1}{\theta}\sum_{k\in[\theta]}\varepsilon_k \leq \frac{\varepsilon}{2}.$$

The following lemma then guarantees that the majority vote amplifies the average error rate of the empirical risk minimizers at most by a factor of 2.

Lemma 20 ((Conitzer, Panigrahi, and Zhang 2020))**.** *Fix a feature space $\mathcal{X}$, a distribution $\mathcal{D}$ over $\mathcal{X}$, and $f\subseteq\mathcal{X}$. Suppose there are θ subsets of $\mathcal{X}$, $\{h_k\}_{k\in[\theta]}$, satisfying*

$$\frac{1}{\theta}\sum_{k\in[\theta]}\Pr_{x\sim\mathcal{D}}[f(x)\neq h_k(x)]\leq\varepsilon',$$

for some $\varepsilon'>0$. Then the pointwise majority vote h of $\{h_k\}_{k\in[\theta]}$, defined such that for any $x\in\mathcal{X}$,

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

satisfies

$$\Pr_{x\sim\mathcal{D}}[h(x)\neq f(x)]\leq 2\varepsilon'.$$

We apply Lemma 20 to $\{h_k\}_{k\in[\theta]}$, with $\varepsilon'=\varepsilon/2$. The condition is satisfied, since the error rate of h_k is upper bounded by ε_k, and the average of these $\{\varepsilon_k\}_k$ is at most $\varepsilon/2$. As a result, the majority vote h, which is the final output of the learning algorithm, has error rate at most ε. As for the failure probability, recall that the algorithm may fail in 3 cases: (1) the realized mapping r^* is an outlier, so no property can be guaranteed by the (ε_0,δ_0)-stability of $(\mathcal{S},\mathcal{X},\mathcal{R})$, which happens with probability at most δ_0, (2) one of the calls to the ERM oracles fails, which happens with probability at most $\delta/3$ over the labeled sample nodes, and (3) the upper bound on the average error rate of $\{h_k\}_k$ exceeds $\varepsilon/2$, which happens with probability at most $\delta/3$ over the sampled hypothesis classes $\{\mathcal{H}_k\}_k$. So, taking a union bound over these three cases, we infer that the total probability of failure does not exceed $\delta\geq 3\delta_0$ for any $C_2\geq 3$. This concludes the proof of the theorem. ☐

Proof of Corollary 16. We apply Theorem 15 with $\mathcal{S}=\mathcal{H}_g\times\mathcal{H}_v$, and $\mathcal{R}=\mathcal{R}(\mathcal{G})$, which directly gives (ε,δ)-learnability with

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

samples. Now by Theorem 7,

$$\begin{aligned}
m&=O\left(\frac{\mathbb{E}_{G\sim\mathcal{G}}[d_v\log n+\min(\rho(G),d_g\log n)]\cdot\log(1/\varepsilon)+\log(1/\delta)}{\varepsilon}\right)\\
&=O\left(\frac{(d_v\log n+\min\left(\mathbb{E}_{G\sim\mathcal{G}}[\rho(G)],d_g\log n\right))\cdot\log(1/\varepsilon)+\log(1/\delta)}{\varepsilon}\right),
\end{aligned}$$

as desired. ☐