Scalable Vaccine Distribution in Large Graphs given Uncertain Data

Yao Zhang
Department of Computer Science
Virginia Tech
yaozhang@cs.vt.edu

B. Aditya Prakash
Department of Computer Science
Virginia Tech
badityap@cs.vt.edu

ABSTRACT

Given an noisy or sampled snapshot of a network, like a contact-network or the blogosphere, in which an infection (or meme/virus) has been spreading for some time, what are the best nodes to immunize (vaccinate)? Manipulating graphs via node removal by itself is an important problem in multiple different domains like epidemiology, public health and social media. Moreover, it is important to account for uncertainty as typically surveillance data on who is infected is limited or the data is sampled. Efficient algorithms for such a problem can help public-health experts take more informed decisions.

In this paper, we study the problem of designing vaccine-distribution algorithms under an uncertain environment, with known information consisting of confirmed cases as well as a probability distribution of unknown cases. We formulate the NP-Hard Uncertain Data-Aware Vaccination problem, and design multiple efficient algorithms for factorizable distributions (including a novel sub-quadratic algorithm) which naturally take into account the uncertainty, while providing robust solutions. Finally, we show the effectiveness and scalability of our methods via extensive experiments on real datasets, including large epidemiological and social networks.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications–*Data mining*

Keywords

Graph Mining; Uncertainty; Immunization; Diffusion

1. INTRODUCTION

What is the best way to distribute vaccines to prevent the spread of diseases on a socio-contact network? Most previous works (see Related Work) for controlling propagation have concentrated on developing strategies for vaccination (node/edge removal) pre-emptively before the start of an epidemic. While very useful to provide insights in to which baseline policies can best control an infection, they may not be ideal to help make real-time decisions as the infection is progressing. Consider also social media and cyber security. Popular phrases or links or rumors are re-posted/re-tweeted on Facebook/Twitter, 'infecting' followers to do the same. How should Twitter decide which accounts to suspend/delete to stop active rumors/spam/malware as much as possible? Which machines should install patches first, in presence of malware attacks? All these problems can be thought of as immunization/vaccination in a network, in presence of already infected nodes [46].

However, in reality contagions usually spread over uncertain environments and the sources of such uncertainty are many. For example, in public health, due to the so-called multi-layered surveillance pyramid [39, 16, 30] at each layer the number of *detected* infections is a fraction of the infections in the layer below it. Hence the total detected infections at the top of the pyramid is a fraction of the actual infections in the population at the bottom. Another example is the likelihood ratios used in diagnostic testing [13]. For each a person who gets the negative test outcome, she has some probability that her test was a false-negative. In social media, as externals we rarely get access to the complete cascade. Researchers usually have access to only a uniform sample of cases (e.g. the Twitter API). In Facebook, most users keep their activity and profiles private. Moreover, if only because of the extreme velocity of social media data, one has to resort to using only a sample of the data. Hence this implies that we will have to make do with only an uncertain snapshot.

In this paper, we study the problem of how to best distribute vaccines to nodes in large networks, in presence of uncertain prior information. Our goal is *not* to fill-in the missing information; instead we want to take robust decisions *in presence* of uncertain information. Our contributions include:

1. **Problem Formulation**: We formulate the Uncertain Data-Aware Vaccination problem, which takes into account multiple natural uncertainty models arising from social media and epidemiology.
2. **Efficient Algorithms**: As the problem is NP-hard and hard to approximate within absolute error, we develop multiple polynomial-time algorithms of varying efficiency, namely (a) SAMPLE-CAS, based on the sample average approximation; and (b) EXPECT-MAX, a faster hybrid algorithm which leverages the so-called

CIKM'14, November 3–7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2661829.2662088.

expected graph and two complementary approaches to estimate benefits.

3. **Extensive Experiments**: We demonstrate the effectiveness and scalability of our algorithms on multiple real datasets including large epidemiological and social networks, over different uncertainty distributions and initial conditions. Our algorithms outperform several other competitor algorithms, getting substantial gains in both number of nodes saved, and running time.

The rest of the paper is organized as follows. Section 2 presents some preliminaries while Section 3 sets up the Uncertain Data-Aware Vaccination problem, and discusses the computational complexity of our problem, Section 4 presents our algorithms and Section 5 presents experimental results on several datasets. We give related work in Section 6, and finally conclude in Section 7.

2. PRELIMINARIES

Table 1 lists the main symbols used in this paper. There exists an underlying contact network G on which the contagion (disease/virus/meme etc.) can spread. We assume that our network is weighted and undirected, but all our methods can be naturally generalized to directed graphs.

Table 1: Terms and Symbols

Symbol	Definition and Description
UDAV	Uncertain Data-Aware Vaccination problem
IC	Independent Cascade Model
SIR	Susceptible-Infected-Recovered Model
footprint	number of infected nodes at the end
benefit	number of nodes saved
$G(V, E)$	graph G with nodes set V and edges set E
$\mathcal{U}$	uncertainty model
$\beta_{i,j}$	propagation probability from node i to j (weight over edges)
p_i	probability that i is infected at the start
k	the budget (i.e., the number of vaccines available)
S	set of nodes to give vaccines to
$\mathbb{E}_S(F)$	the expected footprint after vaccinating S
$\delta_Z(S)$	given graph Z, the expected benefit of vaccinating S in Z
l	number of samples
α	percentage of nodes that have $p_i > 0$ in $\mathcal{U}$

We use two widely used propagation models to describe how the virus spreads on the network: the Independent Cascade (IC) model and the Susceptible-Infected-Recovered (SIR) model. SIR is a well-known epidemiological model to model mumps-like infections [17, 2]. A node in this model can be healthy (susceptible), infectious or recovered. When a node u becomes infected at the timestamp t, it will try to infect each of its direct healthy neighbors v with the propagation probability $\beta_{u,v}$. If u succeeds, v will become infectious at the timestamp $t+1$. At the end of each timestamp t, each infected node u has a curing probability ρ to become 'recovered' at the next timestamp $t+1$. Once recovered, u will never be infected further. The process stops when no additional node becomes infectious. The IC model [21], a special case of SIR, has been extensively studied in the social media to model the viral marketing. Unlike SIR, a node u in IC has only single chance to infect its healthy neighbors (hence the curing probability, $\rho = 1$ here).

3. OUR PROBLEM FORMULATION

3.1 Uncertainty model

In this paper, we are concerned with the scenario when we know the underlying contact network, but we do not know the exact current infected state of the network. One source of uncertainty is public-health surveillance [39, 16, 34, 30, 12, 5]. Generally there are three types of surveillance: population-based, health provider-based and lab-based. Although different types of surveillance may have different probabilities to miss the truly infected person, we can simply use a set of probabilities $\mathcal{P}$ (over the nodes) to model such uncertainty. Another example is the likelihood ratios used in diagnostic testing; each a person has a probability p that her test was a false-negative. In Twitter, each relevant 'infected' tweet can be modeled as having some probability of being missed (because of uniform samples [32]).

Table 2 summarizes common probability distributions models $\mathcal{U}$ we use in this paper to model the uncertainty in observed infections. Each gives the probability of a node i not observed as infected being truly infected. We focus on fully factorizable distributions (over nodes) for simplicity[1]. Hence, if G_j denotes a particular configuration of infections in the network (i.e. a 'possible world'), then $\Pr(G \equiv G_j) = \prod_{a \in I} p_a \prod_{b \in H} (1 - p_b)$ where I and H are the set of infected and healthy nodes in G_j, and the probabilities p_i for any node i come from $\mathcal{U}$.

3.2 Problem Definition

Now we are ready to state our problem formally. We assume that a contagion can travel in principle from any node to any other node i.e. the graph is connected (strongly connected if directed). We are given a fixed-set I_0 with infected nodes, and an uncertainty model $\mathcal{U}$ as above. We are also given a budget k of vaccines. Giving a vaccine to a node renders it immune to the virus and hence it can not get infected further (effectively removing it from the network). Our goal is to find the 'best' set S of nodes to vaccinate to minimize the spread of the contagion, which can be measured by the so-called 'footprint', the number of infected nodes at the end. A subtle point is that vaccination is meaningful *only for healthy nodes*. Hence when we select a node-set S, not all the nodes in S can be vaccinated (removed) in all possible sampled graphs: if a node i is infected in a possible world G_j, then it can not be vaccinated in G_j, and it does not give us any benefit there.

More specifically, suppose S is the set of nodes selected initially for vaccination and G_i is a particular realization ('world') sampled from $\mathcal{U}$. There only exist infected or healthy nodes in G_i. Denote $S_i \subseteq S$ as the subset of nodes

[1]Extending our results to more general forms e.g. distributions factorizing over groups of nodes being infected is interesting future work.

Table 2: Uncertainty models for initial infections used in this paper.

Name	Distribution	Description
`UNIFORM`	$p_i = p$	All nodes have identical probability to be infected. Can be thought of sampling rate in case of Twitter API [32].
`SURVEILLANCE`	$p_i \in \mathcal{P}$	Each node takes a probability from $\mathcal{P}$ (which is a finite set of probabilities like $\{0.1, 0.5, 0.9\}$). See Surveillance pyramid [39, 34, 30].
`PROP-DEG`	$p_i \propto d_i$	The probability to be infected for each node is proportional to its degree, i.e., people with larger number of connections have higher probabilities to be infected.
`GENERAL`	p_i	Each node has its own infected probability.

in S that are healthy in G_i—these are the nodes which will be vaccinated in G_i. Denote $\sigma_{G_i}(S_i)$ as the expected number of infected nodes after running the epidemiology model e.g. IC, on G_i starting from the infected nodes in G_i but *after removing* nodes in set S_i. Let F be the random variable denoting the number of infected nodes after choosing set S under $\mathcal{U}$. Then $\mathbb{E}_S(F) = \sum_{G_i \sim \mathcal{U}} \Pr(G_i)\sigma_{G_i}(S_i)$ and we are trying to find the best set S to minimize $\mathbb{E}_S(F)$. Formally:

PROBLEM 1: Uncertain Data-Aware Vaccination Problem: $UDAV(G, \mathcal{U}, I_0, k)$.

GIVEN: *A graph $G(V, E)$ with node set V and edge set E, the uncertainty model $\mathcal{U}$, the infected node set I_0, propagation probability on each edge $\{i, j\}$ $\beta_{i,j}$, and an integer (budget) k.*

FIND: *A set of nodes* $S^* = \underset{S}{\operatorname{argmin}}\ \mathbb{E}_S(F)$ s.t. $|S| = k$.

Note that as vaccination will be applied only to healthy nodes in a possible world, this formulation also naturally generalizes the corresponding deterministic version of this problem (data-ware vaccination problem studied in *DAV* [46]).

Complexity. *UDAV* is NP-hard, and cannot be approximated within an absolute error since its deterministic counterpart *DAV* is itself NP-hard, and cannot be approximated within an absolute error [46].

4. OUR PROPOSED METHODS

Overview. In this section, we first present a sampling algorithm SAMPLE-CAS for *UDAV*, which is a stochastic algorithm under the SAA framework. However, SAMPLE-CAS is not scalable to large networks. Hence, we propose two faster algorithms: EXPECT-DOM and EXPECT-EIG, which are based on the expected graph and measuring benefits of vaccinations. After analyzing the performance of EXPECT-DOM and EXPECT-EIG, we show that these two algorithms are complementary w.r.t. the support of the uncertainty model, and hence we present a hybrid algorithm called EXPECT-MAX with sub-quadratic running time.

We assume the `GENERAL` model everywhere in this section (as the rest in Table 2 are just special cases of `GENERAL`). Further we describe the algorithms assuming the IC model first(Section 4.1 and 4.2)—later, we will discuss how to extend to the SIR model (Section 4.3).

4.1 The Sample-Cascade Algorithm

Main Idea. Since *UDAV* is a stochastic optimization problem, we try to apply the SAA (Sample Average Approximation) [22] framework to solve it. The idea is to reduce the stochastic optimization problem to the deterministic version by sampling the uncertainty distribution to generate a finite number of deterministic cases. Unfortunately, as we mentioned in the previous section, even the deterministic version of *UDAV* is NP-hard. Hence we leverage the solution in [46], which utilizes a spanning tree called dominator tree, and then find a suitable *sub-modular* structure to solve *UDAV* approximately.

Details. Let $\delta_{G_i}(S_i)$ be the expected benefit after vaccinating the healthy node set S_i in a graph G_i, i.e.:

$$\delta_{G_i}(S_i) = \sigma_{G_i}(\emptyset) - \sigma_{G_i}(S_i) \quad (1)$$

So

$$\begin{aligned} \mathbb{E}_S(F) &= \sum_{G_i} \Pr(G_i)\sigma_{G_i}(S_i) \\ &= \sum_{G_i} \Pr(G_i)(\sigma_{G_i}(\emptyset) - \delta_{G_i}(S_i)) \quad (2) \end{aligned}$$

Since $\sum_{G_i} \Pr(G_i)\sigma_{G_i}(\emptyset)$ is constant, *UDAV* (Problem 1) can be rewritten as:

$S^* = \underset{S}{\operatorname{argmax}}\ \sum_{G_i} \Pr(G_i)\delta_{G_i}(S_i)$ s.t. $|S| = k$.

So we need to compute $\delta_{G_i}(S_i)$ for each G_i, which is essentially the deterministic problem on graph G_i. Hence we re-purpose the solution from [46]: first merge all the infected nodes in G_i into a super node I_0 (I_0 is infected). If a healthy node has multiple infected neighbors, I_0 will connect to the node with the probability that is the logical-OR of the individual probabilities (so if a node u has two infected neighbors x and y, $\beta_{I_0,u} = 1 - (1 - \beta_{x,u})(1 - \beta_{y,u})$). Secondly, build a dominator tree Dom_{G_i} on this merged graph, and properly weight it. Briefly, given a source node I_0, a node v dominates another node u if every path from I_0 to u contains v. Node v is the immediate dominator of u, denoted by $v = idom(u)$, if v dominates u and every other dominator of u dominates v. We can build a dominator tree rooted at I_0 by adding an edge between the nodes u and v if $v = idom(u)$ (totally in near-linear time [7, 26]). Finally we approximate $\delta_{G_i}(S_i)$ as $\delta_{Dom_{G_i}}(S_i)$ (i.e. the benefit after removing nodes in S_i in Dom_{G_i}).

In fact, can further prove that the real benefit of removing S_i from graph G_i is lower-bounded by $\delta_{Dom_{G_i}}(S_i)$.

LEMMA 1. *(Lower Bound of $\delta_{G_i}(S_i)$) The nodes we can save from G must be greater than nodes we save from its dominator tree, that is, $\delta_{Dom_{G_i}}(S_i) \le \delta_{G_i}(S_i)$ (where the inequality is saturated when the merged graph is a tree).*

PROOF. (Sketch) For any node u in S_i, the benefit we can get on G_i is at least all nodes under the subtree of u in the dominator tree of G_i (because there is no path from I_0 to those nodes). Hence, $\delta_{Dom_{G_i}}(S_i) \le \delta_{G_i}(S_i)$. □

Let $Q(S) = \sum_{G_i} \Pr(G_i)\delta_{Dom_{G_i}}(S_i)$; then using Lemma 1, we get $Q(S) \le \sum_{G_i} \Pr(G_i)\delta_{G_i}(S_i)$. The gap between $Q(S)$

and $\sum_{G_i} \Pr(G_i)\delta_{G_i}(S_i)$ depends on the structure of the graph (if the merged-graph is a tree, there is no gap). Hence, Lemma 1 suggests that we can use $Q(S)$ to approximate $\sum_{G_i} \Pr(G_i)\delta_{G_i}(S)$. Hence we formulate Problem 2 next, to *approximate UDAV* (Problem 1).

PROBLEM 2: GIVEN: $G(V, E)$, $\mathcal{U}$, I_0, *and* k.
FIND: *A set of nodes* $S^* = \operatorname{argmax}_S Q(S)$ *s.t.* $|S| = k$.

Interestingly, $Q(S)$ is a submodular function, while $\delta_{G_i}(S)$ is not submodular [46].

LEMMA 2. *(Submodularity of $Q(S)$) $Q(S)$ is a submodular function.*

PROOF. (Sketch) First of all, we prove that given a set S, $\delta_{Dom_{G_i}}(S)$ is a submodular function of S (this can be done by a case analysis). Then $Q(S)$ is a submodular because the linear combination of submodular functions is still a submodular function. □

We now apply the SAA framework: sample l graphs $G_1, G_2, \ldots, G_l$ from $\mathcal{U}$ and define $Q_l(S) = \frac{1}{l}\sum_{i=1}^{l} \delta_{Dom_{G_i}}(S_i)$. As $Q_l(S)$ is a submodular function, we apply the greedy algorithm [33] to obtain the $(1-1/e)$-approximation for Problem 2 (under the l samples). We call this algorithm SAMPLE-CAS (Algorithm 1). Note that we can speed up Algorithm 1 using CELF optimization [27].

Algorithm 1 The SAMPLE-CAS algorithm

Require: Input G, $\mathcal{U}$,I_0, k and l
1: Sample $G_1, \ldots, G_l$ from $\mathcal{U}$ and G
2: Merge infected nodes into I_i^0 for each G_i
3: Build dominator trees $Dom_{G_1}, \ldots, Dom_{G_l}$ rooted at I_i^0 for G_i
4: $S = \emptyset$
5: **for** $i \leftarrow 1$ **to** k **do**
6: $\quad a^* = \arg\max_a \frac{1}{l}\sum \delta_a^{Dom_{G_i}}$
7: $\quad$ Remove a^* from each of $Dom_{G_1}, \ldots, Dom_{G_l}$.
8: $\quad S = S \cup \{a^*\}$.
9: **end for**
10: **return** S

LEMMA 3. *(Running Time of SAMPLE-CAS) The time complexity for Algorithm 1 is $O(l(k|V|+k|E|+|V|\log|V|))$.*

PROOF. (Sketch) Sample G_i needs $O(l|V|)$ time, and build l dominator trees and weight it need $O(l|V|\log|V|)$ time. Selecting a node a needs $l(|E|+|V|)$ time. So in general the time complexity is $O(l(k|V|+k|E|+|V|\log|V|))$. □

How many samples? The next lemma estimates the number of samples l needed so that $Q_l(S)$ is a good estimate of $Q(S)$.

LEMMA 4. *(Number of samples) For any $\epsilon > 0$, to estimate $Q(S)$ within absolute error ϵ with probability $\gamma = 1 - 2\exp(-\frac{2l\epsilon^2}{\Delta^2})$, we need $l \geq \frac{\Delta^2}{2\epsilon^2} ln\frac{2}{1-\delta}$, where Δ is the upper bound for $\delta_{Dom_{G_i}}(S)$ in the dominator tree.*

PROOF. (Sketch) It follows from using the well-known Hoeffding's Inequality [31]. □

As Δ can be $O(|V|)$, Lemma 4 shows that we need worst-case $O(|V|^2)$ samples to get accurate estimates. Hence SAMPLE-CAS does not scale to large networks.

4.2 Expect-Max: a faster algorithm

Since SAMPLE-CAS is not scalable for large networks, we next develop another faster algorithm EXPECT-MAX. We give the main idea, and then describe the details in the subsequent subsections.

Main Idea. We first formulate an equivalent problem which uses the concept of a so-called 'expected graph' $G_{\mathbb{E}}$. Based on that, we propose two different methods EXPECT-DOM and EXPECT-EIG, measuring expected benefits of vaccinations. We show that these two methods are in fact complementary, and hence then propose EXPECT-MAX, which is sub-quadratic in running time (in nodes/edges).

4.2.1 Expected Graph: An Equivalent Formulation

Here, we formulate an equivalent formulation of Problem 1 based on the concept of an 'expected graph'.

DEFINITION 1 (EXPECTED GRAPH): *The expected graph $G_{\mathbb{E}}$ is constructed as follows: start with G; add a 'super node' I_0; connect I_0 to any node i where $p_i > 0$ with the edge weight $\beta_{I_0,i} = p_i$ ($p_i \in \mathcal{U}$, the uncertainty model); and then mark all nodes except I_0 as healthy nodes.*

As we show next, this construction transforms the uncertainty model from nodes to edges without losing any information. Hence, we can focus on a *single* graph $G_{\mathbb{E}}$ instead of sampling graphs (the main reason why SAMPLE-CAS was slow).

More specifically, we show in Lemma 5, an equivalent formulation of Problem 1 based on expected graph $G_{\mathbb{E}}$, under `GENERAL` and for budget $k = 1$. The main idea is that, crucially, as `GENERAL` is *factorizable* (i.e. for a particular configuration G_j, $\Pr(G \equiv G_j) = \prod_{a\in I} p_a \prod_{b\in H}(1-p_b)$, see Section 3 for details), after running the first step of the diffusion model on the expected graph, we will get the same configurations like sampling from the uncertainty model in Problem 1. A subtle point is that Lemma 5 also takes into account the fact that nodes can not be vaccinated in all 'possible-worlds' (wherever they are already infected), by correcting the estimate got from $G_{\mathbb{E}}$ by an appropriate factor.

LEMMA 5. *(Equivalent formulation of UDAV when $k = 1$) When the budget $k = 1$, for the UDAV problem, the best node $a^* = \arg\min_a \mathbb{E}_{\{a\}}(F)$ can be equivalently wrriten as $a^* = \arg\max_a (1-p_a)\delta_{G_{\mathbb{E}}}(\{a\})$.*

PROOF. (Sketch) We first prove $\delta_{G_{\mathbb{E}}}(a) = \sum_{G_i} \Pr(G_i)\delta_{G_i}(a)$ using Definition 1 and the factorizability of `GENERAL`. Based on this, we can prove that $\mathbb{E}_{\{a\}}(F) = \sum_{G_i} \Pr(G_i)\ \sigma_{G_i}(\emptyset) - (1-p_a)\delta_{G_{\mathbb{E}}}(\{a\})$, hence minimizing $\mathbb{E}_{\{a\}}(F)$ is equivalent to maximizing $(1-p_a)\delta_{G_{\mathbb{E}}}(\{a\})$. □

Lemma 5 shows that when the budget $k = 1$, we can get an equivalent formulation of Problem 1 based on the expected graph. Furthermore, note that *UDAV* is a stochastic problem, while Lemma 5 is based on calculating 'benefits' $\delta_{G_{\mathbb{E}}}(\{a\})$ on a *deterministic* graph $G_{\mathbb{E}}$. Next we propose two heuristics to estimate $\delta_{G_{\mathbb{E}}}(\{a\})$ on $G_{\mathbb{E}}$ which are complementary methods based on α, the support of the uncertainty model (see Section 4.2.4 for more details).

4.2.2 The EXPECT-DOM Algorithm

One of the ways we can estimate the benefits is by using our Lemma 1 on $G_{\mathbb{E}}$. The main idea is that we estimate $\delta_{G_{\mathbb{E}}}(\{a\})$ by its lowerbound $\delta_{Dom_{G_{\mathbb{E}}}}(\{a\})$ via the dominator tree on the expected graph. Motivated by the equivalent

formulation of Problem 1 (Lemma 5), we propose that at each step select a node with the maximum value of $(1-p_a)\delta_{Dom_{G_{\mathbb{E}}}}(\{a\})$ after building the dominator tree $Dom_{G_{\mathbb{E}}}$ of $G_{\mathbb{E}}$. We call this algorithm EXPECT-DOM (Algorithm 2).

Algorithm 2 The EXPECT-DOM algorithm

Require: Input G, $\mathcal{U}$,I_0 and k
1: Construct $G_{\mathbb{E}}$
2: $S = \emptyset$
3: Build a dominator tree $Dom_{G_{\mathbb{E}}}$ on $G_{\mathbb{E}}$
4: **for** $i \leftarrow 1$ **to** k **do**
5: $a^* = \arg\max_a (1-p_a)\delta_{Dom_{G_{\mathbb{E}}}}(\{a\})$
6: $S = S \cup \{a^*\}$
7: Remove a^* from $G_{\mathbb{E}}$
8: **end for**
9: **return** S

LEMMA 6. *(Running Time of* EXPECT-DOM*) The time complexity for Algorithm 2 is* $O(k(|V|+|E|)+|V|\log|V|)$.

PROOF. (Sketch) Creating an expected graph $G_{\mathbb{E}}$ costs $O(|V|)$ time, building a dominator tree and weight it need $|V|\log|V|$ time. Updating dominator tree costs $O(|V|+|E|)$ time. Hence, the time complexity of EXPECT-DOM is $O(k(|V|+|E|)+|V|\log|V|)$. □

4.2.3 The EXPECT-EIG Algorithm

Another approach we propose is to estimate $\delta_{G_{\mathbb{E}}}(\{a\})$ is via the change in the largest eigenvalue of $G_{\mathbb{E}}$, $\Delta\lambda_1(a)$, after removing node a. The largest eigenvalue of the adjacency matrix of a graph is related to the so-called 'epidemic threshold' of the graph under several epidemic models [37, 36]. If the largest eigenvalue is very small, a virus will get extinguished quickly. Next we will explain why $\Delta\lambda_1(a)$ is crucial to the benefits. In addition to that, we will show how to estimate $\delta_{G_{\mathbb{E}}}(\{a\})$ using the greedy algorithm in [43] as well.

Justification of $\Delta\lambda_1(a)$. Let $\lambda_i/\mathbf{u_i}$ be the i-th largest eigenvalue/ eigenvector of $G_{\mathbb{E}}$, and $\mathbf{f_t}$ be the vector of probability of each node being infected at time t. The next lemma will show that the expected number of newly infected nodes is upper-bounded by a function of λ_1. Hence, reducing λ_1 (maximizing $\Delta\lambda_1(a)$) by removing node a, can effectively minimize the expected number of newly infected nodes, and eventually minimize $\mathbb{E}_{\{a\}}(F)$ (the expected number of infected nodes at the end). According to Equation 2 (in Section 4.1), minimizing $\mathbb{E}_{\{a\}}(F)$ is equivalent to maximizing the benefit $\delta_{G_{\mathbb{E}}}(\{a\})$. Hence we can estimate $\delta_{G_{\mathbb{E}}}(\{a\})$ using $\Delta\lambda_1(a)$.

LEMMA 7. *The expected number of newly infected nodes at timestep* $t+1$*, is upper-bounded by* $h = \mathbf{e}'(\sum_{j=1}^{|V|}\lambda_j^t \mathbf{u_j}\mathbf{u_j}')\mathbf{f_1}$. *Furthermore,* $h \le \lambda_1^t \mathbf{e}'(\sum_{j=1}^{|V|}\mathbf{u_j}\mathbf{u_j}')\mathbf{f_1}$ *where* $\mathbf{e} = (1,\ldots,1)'$ *and* $\mathbf{f_1} = (p_1,\ldots,p_n)'$ *(the initial infection probabilities of the nodes, which essentially comes from the uncertainty model).*

PROOF. (Sketch) First, following steps of Lemma 1 in [36], we can get that the expected number of newly infected nodes at timestep $t+1$ is upper-bounded by $\mathbf{e}'(\sum_{j=1}^{|V|}\lambda_j^t \mathbf{u_j}\mathbf{u_j}')\mathbf{f_1}$. Second, since λ_1 is real and positive (using Perron-Frobenius theorem), we get $h \le \lambda_1^t \mathbf{e}'(\sum_{j=1}^{|V|}\mathbf{u_j}\mathbf{u_j}')\mathbf{f_1}$. □

The Expect-Eig Algorithm. Motivated by Lemma 5 (the equivalent formation of Problem 1) and Lemma 7, we can greedily select a node with the maximum value of $(1-p_a)\Delta\lambda_1(a)$ at each step, using $\Delta\lambda_1(a)$ as an estimate of the benefit of removing a node. We call this algorithm EXPECT-EIG (Algorithm 3).

Comment. [43] gives a fast greedy algorithm for this task, by approximating $\Delta\lambda_1(a) \approx 2\lambda_1 u_a^2$ (based on the first-order matrix perturbation theory). Here we use it in Algorithm 3 (Line 5).

LEMMA 8. *(Running Time of* EXPECT-EIG*) The time complexity for Algorithm 3 is* $O(k(|V|+|E|))$.

PROOF. (Sketch) Calculating $\mathbf{u_1}$ costs $O(|E|)$ time using the power method. Hence, Algorithm 3 takes $O(k(|V|+|E|))$ time. □

Algorithm 3 The EXPECT-EIG algorithm

Require: Input G, $\mathcal{U}$,I_0 and k
1: Construct $G_{\mathbb{E}}$
2: Get λ_1 and $\mathbf{u_1} = (u_1,\ldots,u_n)'$ from $G_{\mathbb{E}}$.
3: $S = \emptyset$
4: **for** $i \leftarrow 1$ **to** k **do**
5: $\Delta\lambda_1(a) = 2\lambda_1 u_a^2$
6: $a^* = \arg\max_a (1-p_a)\Delta\lambda_1(a)$
7: $S = S \cup \{a^*\}$
8: Remove a^* from $G_{\mathbb{E}}$ and update λ_1 and $\mathbf{u_1}$.
9: **end for**
10: **return** S

4.2.4 The Hybrid Algorithm: Expect-Max

Although EXPECT-DOM and EXPECT-EIG are both fast algorithms compared to SAMPLE-CAS, they may not work well all the time. Next we will discuss how uncertainty models affect their performances, and present a hybrid algorithm combining both of them.

Discussion about Expect-Dom. Denote α as the support of the uncertainty model (the percentage of nodes that are possibly infected). When $\alpha = 0$, the *UDAV* problem becomes exactly the *DAV* problem [46] (the deterministic case of *UDAV*) and EXPECT-DOM reduces to the algorithm in [46], which was shown to perform well. However consider the opposite case $\alpha = 1$. In this case, I_0 connects to the rest of nodes. Hence the dominator tree of $G_{\mathbb{E}}$ becomes a star. For any node a, $\delta_{Dom_{G_{\mathbb{E}}}}(a)$ will only depend on the propagation probability from I_0 to a (i.e., p_a). We cannot utilize any other information from the original graph, hence we would choose nodes essentially randomly. This also gives us the intuition that as α increases, the performance of EXPECT-DOM will become worse (a fact we demonstrate in experiments as well).

Discussion about Expect-Eig. As we discussed in Section 4.2.3, the expected number of newly infected nodes at timestep $t+1$ is upperbounded by $h = \mathbf{e}'(\sum_{j=1}^{|V|}\lambda_j^t \mathbf{u_j}\mathbf{u_j}')\mathbf{f_1} \le \lambda_1^t \mathbf{e}'(\sum_{j=1}^{|V|}\mathbf{u_j}\mathbf{u_j}')\mathbf{f_1} = h_1$ ($\mathbf{f_1}$ essentially comes from the uncertainty model). We first demonstrate that this inequality ($h \le h_1$) saturates when $\mathbf{f_1}$ is parallel to $\mathbf{u_1}$. Then, we will show how to maximize our chance to achieve this, which will lead us to the discussion about the performance of EXPECT-EIG in terms of α.

LEMMA 9. *(h-*h_1 *Gap) As the inner product of* $\mathbf{u_1}$ *and* $\mathbf{f_1}$ *increases,* $h_1 - h$ *decreases. When* $\mathbf{f_1}$ *is parallel to* $\mathbf{u_1}$*,* $h_1 = h$.

PROOF. (Sketch) As $\mathbf{u_1'f_1}$ increases, $\mathbf{f_1}$ becomes more parallel to $\mathbf{u_1'}$, and $\mathbf{u_j}'\mathbf{f_1}$ $(j \neq 1)$ becomes smaller (because $\mathbf{u_i}$ and $\mathbf{u_j}$ are orthogonal). Hence $h_2 - h_1$ decreases. And when $\mathbf{u_j}'\mathbf{f_1} = 0$ $(j \neq 1)$, $h_1 = h$. □

This shows that closer the uncertainty model is to $\mathbf{u_1}$, the better bound h_1 is of h: as a result of which we expect $\Delta\lambda_1(a)$ to become a better estimate, and hence EXPECT-EIG to perform better. How is this related to α? The following analysis shows a preliminary justification. Apriori we do not know the graph, hence we do not know $\mathbf{u_1}$: so reasonably we can assume it is randomly uniformly picked from a n-dimensional space. Let us denote $\mathbf{x}$ as the random variable of the first eigenvector. To make $\mathbf{f_1}$ more parallel to $\mathbf{x}$, we need to maximize the expectation of $\mathbf{f_1'x}$ (i.e., $\mathbb{E}_{\mathbf{x}}[\mathbf{f_1'x}]$). It is not hard to see that as we increase α, $\mathbb{E}_{\mathbf{x}}[\mathbf{f_1'x}]$ will increase.

LEMMA 10. *(Expected gap) When α increases, $\mathbb{E}_{\mathbf{x}}[\mathbf{x'f_1}]$ increases as well.*

PROOF. (Sketch) $\mathbb{E}_{\mathbf{x}}[\mathbf{x'f_1}] = \mathbf{f_1'}\mathbb{E}_{\mathbf{x}}[\mathbf{x}]$, and all elements in $\mathbb{E}_{\mathbf{x}}[\mathbf{x}]$ are non-negative. As α increases, more elements in $\mathbf{f_1}$ become non-zero, hence $\mathbb{E}_{\mathbf{x}}[\mathbf{x'f_1}]$ increases as well. □

Lemma 10 suggests that when α increases, we expect $\mathbf{f_1}$ and $\mathbf{u_1}$ to become more parallel, and so the gap to decrease, as a result of which $\Delta\lambda_1(a)$ becomes a better estimate. Thus even this preliminary analysis immediately suggests that as α becomes larger, EXPECT-EIG should perform better. Again we demonstrate this through experiments as well.

The Expect-Max Algorithm. The above discussion suggests a complementary picture: when α is low, we expect EXPECT-DOM to be better, and when α is high, we expect EXPECT-EIG to be better. Unfortunately, we don't know exactly when which algorithm is better: this likely depends not only on α but also the graph, and the distribution. However, we can still leverage this insight to propose a hybrid algorithm called EXPECT-MAX, which maintains the scalability and quality of EXPECT-DOM and EXPECT-EIG. EXPECT-MAX chooses either EXPECT-DOM or EXPECT-EIG based on their performances, that is,

$$S_{\text{EXPECT-MAX}} = \underset{S=\{S_{\text{EXPECT-DOM}}, S_{\text{EXPECT-EIG}}\}}{\text{argmax}} \mathbb{E}_S(F)$$

Comment. S is the output either of EXPECT-DOM or EXPECT-EIG, and $\mathbb{E}_S(F)$ can be obtained by via simulation of the IC model (*not* sampling from the uncertainty model). Also note that EXPECT-MAX is *not* the greedy algorithm that picks one node from either EXPECT-DOM or EXPECT-EIG in each step. Instead, it chooses S just once after running EXPECT-DOM and EXPECT-EIG. Hence the time complexity for EXPECT-MAX is $O(k(|V|+|E|)+|V|\log|V|+T)$ where T is the time to run IC model (which should be sub-quadratic in edges).

4.3 Extending to SIR model

Note that in SIR, the footprint is the total number of *recovered* nodes at the end (in contrast to the IC model). Nevertheless, leveraging the method in [46], we can directly extend our algorithms to SIR model by changing SIR model to IC model with the propagation probability $1-(1-\beta_{i,j})^{\frac{1}{\rho}}$. This does not change any of our algorithms/results.

5. EXPERIMENTS

We present a detailed experimental evaluation in this section.

5.1 Experimental Setup

We briefly describe our set-up next. We implemented the algorithms in Python[2], and conducted the experiments using a 4 Xeon E7-4850 CPU with 512GB of 1066Mhz main memory.

Datasets. We ran our experiments on multiple datasets using both IC and SIR. Table 3 summarizes the datasets, which were chosen for their size as well as the applicability to the *UDAV* problem (from social media to epidemiology).

1. `KARATE` is a social network of friendships with 34 members in a karate club at a US university in the 1970s [45].
2. `OREGON`[3] is the Oregon AS router graph collected from the Oregon router views. The contagion here can be thought of malware and computer-network viruses, which we want to control by shutting-off or patching relevant routers.
3. `STANFORD`[4] is the Stanford CS hyperlink network, in which a web page links to another page. Contagions here can be false information spreading through the webspace, and we want to prevent their spread by posting true information at strategic web pages.
4. `GNUTELLA`[5] is a peer-to-peer network showing the snapshot of the Gnutella P2P file sharing network from August 2012. Similar to `OREGON`, we can control the spread of malware and harmful files by patching some important peers.
5. `BRIGHTKITE`[5] is a friendship network from a location-based social networking service provider Brightkite. As friends regularly frequent the same places, such location-based networks can be useful for the public-health.
6. `PORTLAND` and `MIAMI` are social-contact graphs based on detailed microscopic simulations of large US cities. Edge weights here represent the expected contact time between people. Versions of these have been used in national smallpox and influenza modeling studies using the SIR model [15].

Table 3: Datasets

Dataset	Nodes(V)	Edge(E)	Model
`KARATE`	34	156	IC
`OREGON`	633	2172	IC
`STANFORD`	8929	53829	IC
`GNUTELLA`	10876	39994	IC
`BRIGHTKITE`	59228	0.2 million	IC
`PORTLAND`	0.5 million	1.6 million	SIR
`MIAMI`	0.6 million	2.1 million	SIR

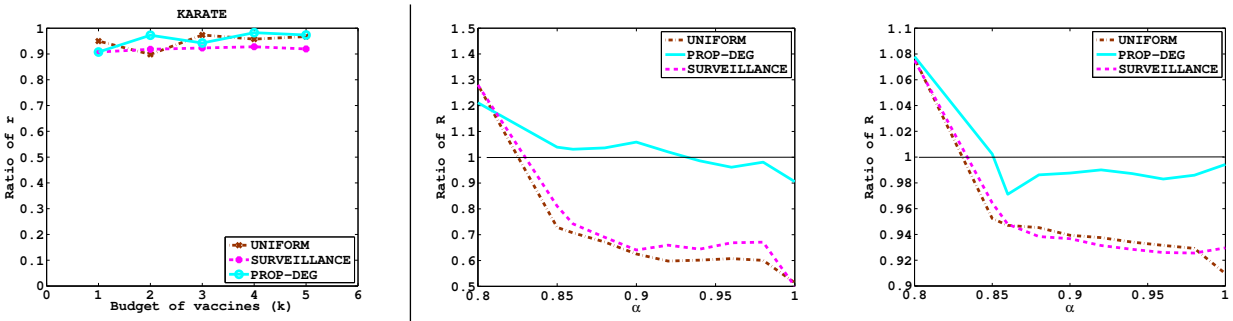
Uncertainty models. We used three types of uncertainty models: (a) `UNIFORM`: $p = 0.6$; (b) `SURVEILLANCE`: p_i uniformly randomly chosen from $\{0.1, 0.5\}$ for each node i (following different levels of the surveillance pyramid, e.g: 10%

[2]Code can be downloaded from `http://people.cs.vt.edu/~yaozhang/code/udav`

[3]`http://topology.eecs.umich.edu/data.html`.

[4]`http://www.cise.ufl.edu/research/sparse/matrices/Gleich/`.

[5]`GNUTELLA` and `BRIGHTKITE` are from `http://snap.stanford.edu/data/index.html`.



(a) Sample-Cas vs. Optimal (b) STANFORD (c) BRIGHTKITE

Figure 1: (a). Quality of Sample-Cas. Comparison between Sample-Cas and Optimal on KARATE over different distributions ($r = \frac{\#healthy\ nodes\ saved\ by\ \textbf{Sample-Cas}}{\#healthy\ nodes\ saved\ by\ \textbf{Optimal}}$ and $\alpha = 0.5$). (b) and (c). Comparison between Expect-Dom and Expect-Eig. Ratio of R ($R = \frac{\#healthy\ nodes\ saved\ by\ \textbf{Expect-Dom}}{\#healthy\ nodes\ saved\ by\ \textbf{Expect-Eig}}$) vs. α. Expect-Dom performs better than Expect-Eig when $R > 1$, otherwise Expect-Eig is better.

of the total population is infected and is in the hospital, and only $\sim 33\%$ of infected people go to a hospital, which together imply 50% of the total population is infected and does not go to a hospital); (c) PROP-DEG: $p_i = d_i/d_{max}$ for each node i (d_{max} is the maximum degree of the graph G).

Parameters. For IC model, $\beta_{u,v}$ is uniformly randomly chosen from $\{0.1, 0.5, 0.9\}$ [9]. For SIR model, we use the normalized contact time as the propagation probability $\beta_{u,v}$, and set a uniform curing probability $\rho = 0.6$. We uniformly randomly pick 5% of nodes as infected nodes. For Sample-Cas, we set the number of samples $l = 200$. For robustness, each data point we show is the mean of 1000 runs of the diffusion/epidemiological model.

Baselines. We compare our algorithms against various intuitive and non-trivial competitors to better judge their performance. Recall that I_0 is the infected node set ($I_0 = \{u | u \in V, p_u = 1\}$). Let us denote $W = V - I_0$ (so W is the set of nodes that are not certainly infected at the start).

1. Optimal: a brute-force algorithm that tries all combinations of possible solutions. As it is very slow, we only run it on very small graph (KARATE).
2. Random: uniformly randomly select k nodes from W.
3. Degree: choose the top-k nodes from W according to their weighted degree.
4. PageRank: pick the top-k healthy nodes from W with the highest pagerank. We use the restart probability of 0.15.
5. Per-PRank: we first merge all infected nodes into one supernode as the preferred node, and then choose the top-k nodes from W with the highest personalized pagerank with respect to the supernode [18]. We use the restart probability of 0.15.
6. Dava-fast: This is a fast immunization algorithm [46], which aims to control the epidemic in presence of already infected nodes (without uncertainty in the data). We apply Dava-fast as if any node from W on G is a healthy node. We take the top-k nodes from W according to the algorithm.

5.2 Results

In short, we demonstrate that Sample-Cas and Expect-Max outperform other baselines on all datasets. Sample-Cas provides very accurate results, but does not scale to large networks, while Expect-Max is fast, scalable and effective. We also show the behaviors of Expect-Dom and Expect-Eig as α varies.

5.2.1 *Accuracy of* Sample-Cas

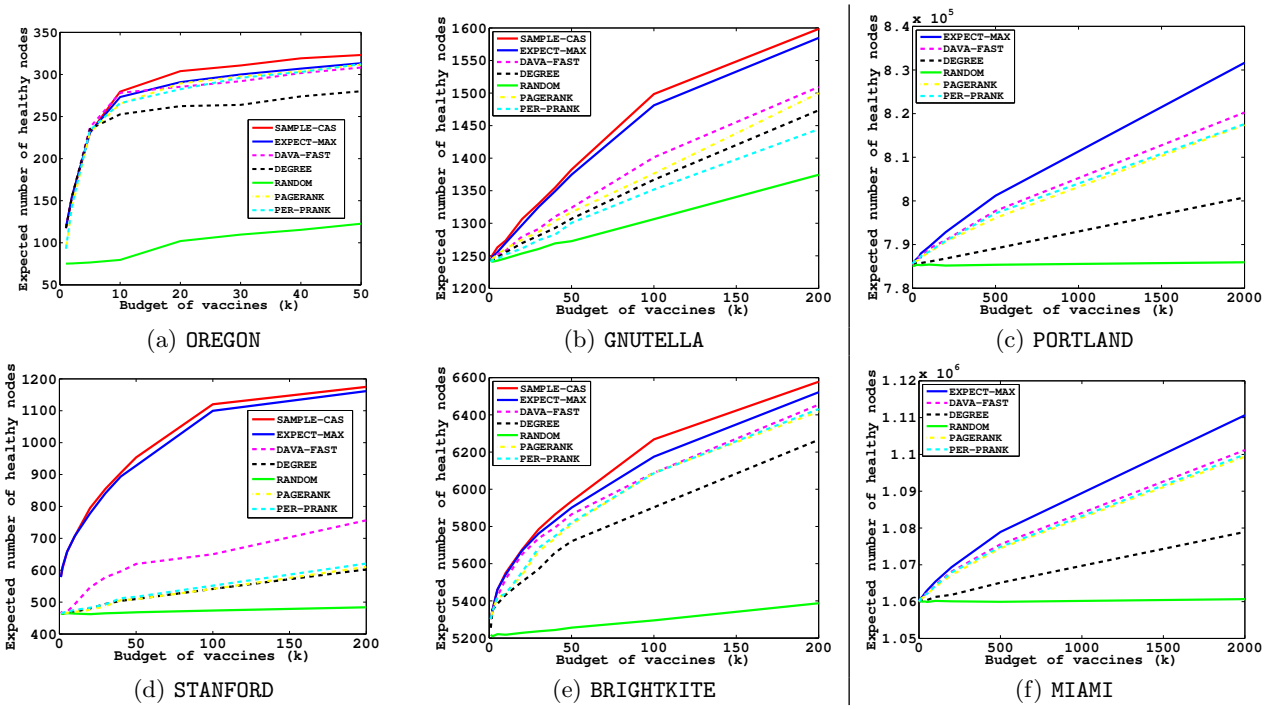
First of all, we compare Sample-Cas with Optimal on KARATE to demonstrate its accuracy (because Optimal is too slow, we chose KARATE so that we can run Optimal completely). As Figure 1(a) shows, for all uncertainty models, Sample-Cas saves at least 90% of nodes compared to Optimal no matter how k changes. We also found as expected, Sample-Cas's performance gets better as number of samples increases (not shown here).

5.2.2 *Justification of* Expect-Max

We compared Expect-Dom with Expect-Eig as α changes on multiple datasets under three uncertainty models (see Figure 1(b) and (c)). For all networks, as expected from our discussion in Section 4.2.4, clearly as α increases, Expect-Eig becomes better while Expect-Dom becomes worse. In addition to that, there does exist a 'cross-over point' for each network where the algorithms switch in performance ($R = 1$ in Figure 1(b) and (c)). However, this cross-over point is different for different networks *and* for different distributions, which is the reason why we propose the Expect-Max algorithm (as we do not know exactly when we should use either Expect-Dom or Expect-Eig as α changes).

5.2.3 *Effectiveness of* Sample-Cas *and* Expect-Max

Figure 2(a), (b), (d) and (e) show experimental results under IC model for UNIFORM. In all networks, Sample-Cas and Expect-Max consistently outperform other competitors. OREGON contains only 600 nodes, hence we varied k till 50. Due to a jelly-fish-type structure of OREGON, for lower k, most algorithms perform well by targeting the nodes in the core. However, for larger k, Sample-Cas provides the best solution, while Expect-Max outperforms other competitors as well, getting solutions almost as good as Sample-Cas. For GNUTELLA, STANFORD and BRIGHTKITE (much larger than OREGON), the difference of Sample-Cas and Expect-Max from the other algorithms is clearer: they save upto 2.5 times the nodes than other algorithms, yet Expect-Max took a fraction of the running time of Sample-Cas (see Ta-



(a) OREGON (b) GNUTELLA (c) PORTLAND

(d) STANFORD (e) BRIGHTKITE (f) MIAMI

Figure 2: Effectiveness ($\alpha = 0.5$, UNIFORM). Expected number of healthy nodes after distributing vaccines vs. budget k. Higher is better. (a), (b), (d), (e): IC model; (c), (f): SIR model. Sample-Cas and Expect-Max outperform other baseline algorithms.

ble 4). Note that although Dava-fast contains information of infected nodes, it doesn't perform well (especially on STANFORD) because it fails to take into account the uncertainty model.

We got similar result under SIR model on PORTLAND and MIAMI for UNIFORM (see Figure 2(c) and (f)). Since PORTLAND and MIAMI have more than $0.5million$ nodes, Sample-Cas did not finish even in a day, and we do not show it on the plots. We notice that the larger k becomes, the better Expect-Max performs than other competitors. When $k = 2000$, the difference of Expect-Max from other algorithms is clearer: it saves more than $10,000$ nodes than the second best algorithm Dava-fast.

For SURVEILLANCE and PROP-DEG, the results are the same: Sample-Cas and Expect-Max always outperform other algorithms (see Figure 3). We do not show the plots of other datasets and other values of α due to lack of space, but the results are similar: Sample-Cas and Expect-Max provide the best solution.

5.2.4 Scalability

Although both Sample-Cas and Expect-Max are polynomial time (in particular Expect-Max is subquadratic in nodes and edges), we show some running time results to evaluate scalability. Table 4 shows the running times of our algorithms under UNIFORM: Expect-Max is much faster than Sample-Cas. Expect-Max takes only **45 seconds** on STANFORD while Sample-Cas takes about **an hour**. The larger networks are, the faster Expect-Max is than Sample-Cas. On BRIGHTKITE with $60K$ nodes, Expect-Max is more than **50 times** faster than Sample-Cas. Furthermore, on the largest network MIAMI, Expect-Max takes about **2.5 hours** to select 100 nodes while Sample-Cas did not finish even in *one day*. Hence, Expect-Max is scalable for large networks.

Table 4: Running times (sec.) when $k = 100$ and $l = 200$ ($\alpha = 0.5$). Runs terminated when running time $t > 24$ hours. (shown by '-')

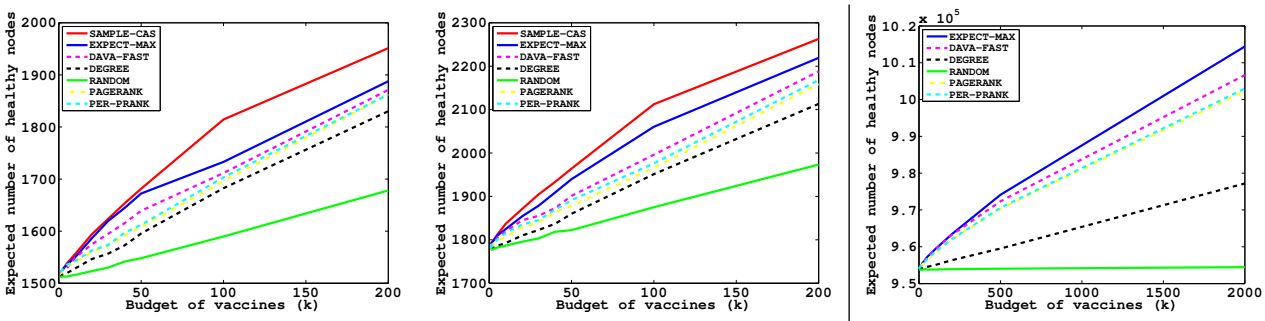
	Sample-Cas	Expect-Max
OREGON	241.6	3.1
STANFORD	3401.7	45.2
GNUTELLA	4221.1	59.8
BRIGHTKITE	19072.0	371.5
PORTLAND	-	6930.2
MIAMI	-	9231.4

6. RELATED WORK

We now review the most closely related work here.

Stochastic Optimization. Extensive surveys and textbooks [40, 20] exist on this topic. Dyer et al. [14] showed that two-stage stochastic programming problems are $\#P$-hard. Sample Average Approximation (SAA) is a well-known framework to approach these problems, which yields strong approximation results [22, 41]. We leveraged the SAA framework for Sample-Cas in this paper. However, it is not scalable to large networks because of its computational complexity.

Handling Uncertainty. Many studies in epidemiology try to estimate the total infections using the surveillance pyramid [39, 16, 34, 30, 12, 5]. More generally, missing data in



(a) SURVEILLANCE (GNUTELLA) (b) PROP-DEG (GNUTELLA) (c) SURVEILLANCE (PORTLAND)

Figure 3: Effectiveness ($\alpha = 0.5$). Expected number of healthy nodes after distributing vaccines vs. budget k. Higher is better. (a), (b): IC model; (c): SIR model. Sample-Cas and Expect-Max outperform other baseline algorithms.

networks is an important yet relatively poorly understood problem. A line of work in databases studies several querying problems on uncertain graphs, including the k-nearest neighbors query [35], discovering reliable subgraphs [19] and efficient subgraph search [44]. Another related line of work studies the effect of sampling on measured structural properties [11, 23, 4] or network construction [25, 29]. Correcting for the effects of missing data in cascades in general has not seen much attention—the exceptions are Sadikov et al. [38] (who try to correct metrics like cascade size for sampling), and Adiga et al. [1] (who study the effect of more general noise in the network *structure* on metrics like expected footprint in the IC and LT models). Here we study a specific *algorithmic* task (immunization) under uncertainty in *observed infections*.

Immunization Algorithms. Most existing studies focus on finding optimal strategies for vaccine allocation under perfect information [10, 6, 28, 8]. Using game theory, Aspnes et al. [3] developed inoculation strategies for victims of viruses under random starting points. Kuhlman et al. [24] studied two formulations of the problem of blocking a contagion through edge removals under the model of discrete dynamical systems. Tong et al. [43, 42] and Prakash et al. [36] proposed various node-based and edge-based immunization algorithms based on minimizing the largest eigenvalue of the graph. Zhang and Prakash [46] studied the problem of immunizing healthy nodes in presence of already infected nodes.

To summarize, none of the above works studies the problem of distributing vaccines given *uncertain surveillance* data.

7. CONCLUSIONS

This paper addresses the problem of distributing vaccines given uncertain data over large networks with applications to cascade-like processes on networks in several areas. The main contributions are:

(a) *Problem Formulation:* Motivated by multiple natural uncertainty models from social media and epidemiology, we first formulated the Uncertain Data-Aware Vaccination (*UDAV*) problem.

(b) *Efficient Algorithms:* Due to its computational complexity, we presented two main novel algorithms: Sample-Cas and Expect-Max. Sample-Cas is an accurate stochastic algorithm under the SAA framework, while Expect-Max is a fast hybrid algorithm with sub-quadratic time complexity, which utilizes the expected graph and two complementary methods to estimate benefits.

(c) *Extensive Experiments:* Experimental results demonstrate that our algorithms outperform several other baseline algorithms on multiple diverse real datasets (from social, cyber, and epidemiological domains) over multiple different uncertainty models.

Future work can include extending our results to other models in epidemiology such as SIS (where nodes can get infected multiple times), and generalizing Expect-Max to any uncertainty distribution (not just factorizable distributions).

Acknowledgements. The authors would like to thank the anonymous reviewers for their comments, and Virginia Bioinformatics Institute for sharing with us the PORTLAND and MIAMI datasets. This material is based upon work supported by the NSF under Grant No. IIS-1353346 and by the Maryland Procurement Office under contract H98230-14-C-0127. Any opinions, findings and conclusions or recommendations express in this material are those of the author(s) and do not necessarily reflect the views of the respective funding agencies.

8. REFERENCES

[1] A. Adiga, C. J. Kuhlman, H. S. Mortveit, and A. K. S. Vullikanti. Sensitivity of diffusion dynamics to network uncertainty. In *AAAI*, 2013.

[2] R. M. Anderson and R. M. May. *Infectious Diseases of Humans*. Oxford University Press, 1991.

[3] J. Aspnes, K. Chang, and A. Yampolskiy. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. SODA '05, pages 43–52, 2005.

[4] S. P. Borgatti, K. M. Carley, and D. Krackhardt. On the robustness of centrality measures under conditions of imperfect data. *Soc. Netw.*, 28(2):124 – 136, 2006.

[5] M. Braks, J. van der Giessen, M. Kretzschmar, W. van Pelt, E. Scholte, C. Reusken, H. Zeller, W. van Bortel, and H. Sprong. Towards an integrated approach in surveillance of vector-borne diseases in europe. *Parasit Vectors*, 4(192), 2011.

[6] L. Briesemeister, P. Lincoln, and P. Porras. Epidemic profiles and defense of scale-free networks. *WORM 2003*, Oct. 27 2003.

[7] A. L. Buchsbaum, H. Kaplan, A. Rogers, and J. R. Westbrook. A new, simpler linear-time dominators algorithm. *ACM Trans. Program. Lang. Syst.*, 20(6):1265–1296, Nov. 1998.

[8] P.-A. Chen, M. David, and D. Kempe. Better vaccination strategies for better people. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, pages 179–188, New York, NY, USA, 2010. ACM.

[9] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. *KDD*, 2010.

[10] R. Cohen, S. Havlin, and D. ben Avraham. Efficient immunization strategies for computer networks and populations. *Physical Review Letters*, 91(24), Dec. 2003.

[11] E. Costenbader and T. W. Valente. The stability of centrality measures when networks are sampled. *Soc. Netw.*, 25(4):283–307, 2003.

[12] J. Crump, F. Youssef, S. Luby, M. Wasfy, J. Rangel, M. Taalat, S. Oun, and F. Mahoney. Estimating the incidence of typhoid fever and other febrile illnesses in developing countries. *Emerging Infectious Diseases*, 9(5):539–544, 2003.

[13] J. J. Deeks and D. G. Altman. Statistics notes: Diagnostic tests 4: likelihood ratios. *BMJ: British Medical Journal*, 329(7458):168, 2004.

[14] M. Dyer and L. Stougie. Computational complexity of stochastic programming problems. *Math. Program.*, 106(3):423–432, May 2006.

[15] S. Eubank, H. Guclu, V. S. Anil Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429(6988):180–184, May 2004.

[16] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2009.

[17] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42, 2000.

[18] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 271–279, New York, NY, USA, 2003. ACM.

[19] R. Jin, L. Liu, and C. C. Aggarwal. Discovering highly reliable subgraphs in uncertain graphs. In *KDD*, pages 992–1000. ACM, 2011.

[20] P. Kall and S. Wallace. *Stochastic programming*. Wiley-Interscience series in systems and optimization. Wiley, 1994.

[21] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Conference of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, New York, NY, 2003. ACM Press.

[22] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM J. on Optimization*, 12(2):479–502, Feb. 2002.

[23] G. Kossinets. Effects of missing data in social networks. *Soc. Netw.*, 28(3):247–268, 2006.

[24] C. J. Kuhlman, G. Tuli, S. Swarup, M. V. Marathe, and S. S. Ravi. Blocking simple and complex contagion by edge removal. In *ICDM*, pages 399–408, 2013.

[25] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie. Sampling biases in ip topology measurements. In *In IEEE INFOCOM*, pages 332–341, 2003.

[26] T. Lengauer and R. E. Tarjan. A fast algorithm for finding dominators in a flowgraph. *ACM Trans. Program. Lang. Syst.*, 1(1):121–141, Jan. 1979.

[27] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, San Jose, California, USA, 2007. ACM. http://doi.acm.org/10.1145/1281192.1281239.

[28] N. Madar, T. Kalisky, R. Cohen, D. ben Avraham, and S. Havlin. Immunization and epidemic dynamics in complex networks. *Eur. Phys. J. B*, 38(2):269–276, 2004.

[29] A. Maiya and T. Berger-Wolf. Benefits of bias: Towards better characterization of network sampling. In *KDD*, 2011.

[30] M. Marathe and A. K. S. Vullikanti. Computational epidemiology. *Commun. ACM*, 56(7):88–96, July 2013.

[31] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.

[32] F. Morstatter, J. Pfeffer, H. Liu, and K. M. Carley. Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. In *ICWSM*, 2013.

[33] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, (1):265–294, 1973.

[34] H. Nishiura, G. Chowell, and C. Castillo-Chavez. Did modeling overestimate the transmission potential of pandemic (h1n1-2009)? sample size estimation for post-epidemic seroepidemiological studies. *PLoS ONE*, 6(3):e17908, 03 2011.

[35] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. K-nearest neighbors in uncertain graphs. *Proceedings of the VLDB Endowment*, 3(1-2):997–1008, 2010.

[36] B. A. Prakash, L. A. Adamic, T. J. Iwashyna, H. Tong, and C. Faloutsos. Fractional immunization in networks. In *SDM*, pages 659–667, 2013.

[37] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. In *ICDM*, 2011.

[38] E. Sadikov, M. Medina, J. Leskovec, and H. Garcia-Molina. Correcting for missing data in information cascades. In *WSDM*. ACM, 2011.

[39] G. J. Smith, D. Vijaykrishna, J. Bahl, S. J. Lycett, M. Worobey, O. G. Pybus, S. K. Ma, C. L. Cheung, J. Raghwani, S. Bhatt, et al. Origins and evolutionary genomics of the 2009 swine-origin h1n1 influenza a epidemic. *Nature*, 459(7250):1122–1125, 2009.

[40] J. Spall. Stochastic optimization. *Handbook of Computational Statistics*, pages 173–201, 2012.

[41] C. Swamy and D. B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *SIGACT News*, 37(1):33–46, Helen Martin 2006.

[42] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos. Gelling, and melting, large graphs by edge manipulation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 245–254, New York, NY, USA, 2012.

[43] H. Tong, B. A. Prakash, C. E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau. On the vulnerability of large graphs. In *ICDM*, 2010.

[44] Y. Yuan, G. Wang, H. Wang, and L. Chen. Efficient subgraph search over large uncertain graphs. *Proc. VLDB Endow*, 4(11), 2011.

[45] W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[46] Y. Zhang and B. Prakash. Dava: Distributing vaccines over large networks under prior information. In *2014 SIAM International Conference on Data Mining (SDM14)*, SDM'14, 2014.