Sensor Placement for Outbreak Detection in Computer Security

Andreas Krause SCS, CMU
H. Brendan McMahan Google Inc.
Carlos Guestrin SCS, CMU
Anupam Gupta SCS, CMU

Abstract

We consider the important computer security problem of outbreak detection, where we want to place sensors (monitoring stations, probes) for detecting events (computer viruses) spreading over a network. We show that such problems can be modeled by the problem of simultaneously maximizing a collection of submodular set functions. We show, how the SATURATE algorithm [3] performs near-optimally in this setting, even if sensors can (accidentally or through adversarial manipulation) fail.

1 Introduction

An important problem in computer security is outbreak detection in networks. In this problem, we are given a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and a process spreading dynamically over the network. We can place a set of monitoring stations, which detect the events. Examples include detecting viruses spreading over computer networks, monitoring municipal water distribution networks for contamination detection [5], and even problems like selecting informative weblogs to read in order to detect citation cascades [6].

More formally, we are given a set of outbreak scenarios $\mathcal{I}$. Each scenario $i \in \mathcal{I}$ models an event starting at a node $s \in \mathcal{V}$ and spreading over the graph. With each node $v \in \mathcal{V}$, we associate the detection time $T(i, v)$ as the earliest time at which the event reaches node v. $T(i, v) = \infty$ if v is never reached. We can place a set of k sensors at a set of nodes $\mathcal{A} \subseteq \mathcal{V}$, $|\mathcal{A}| = k$. These nodes detect the event i at time $T(i, \mathcal{A}) = \min_{v \in \mathcal{A}} T(i, v)$. With each scenario i, we also associate a *penalty function* π_i, where $\pi_i(t)$ quantifies our loss if the outbreak is detected at time t. For example, we can set $\pi_i(t)$ to model the monetary loss associated with servers failing due to the virus infection or to model the amount of contaminated water consumed, if the outbreak is detected at time t. With each scenario, we can then associate the reward function $R_i(\mathcal{A}) = \pi_i(\infty) - \pi_i(T(i, \mathcal{A}))$, which is defined over all subsets $2^{\mathcal{V}}$, and quantifies the utility for placing sensors at locations $\mathcal{A}$. If no sensors are placed, then no utility is obtained.

The goal in outbreak detection is then to place a set of sensors, such that the utility R_i is simultaneously maximized over all R_i. From this goal, one can formalize different optimization problems. If one believes that outbreaks happen at random, then one can define an *average case* objective $R_{avg}(\mathcal{A}) = \sum_{i \in \mathcal{I}} P(i) R_i(\mathcal{A})$. If an adversary selects the outbreak scenario i knowing about our sensor placement (and hence picking the worst possible scenario i), our objective is $R_{adv}(\mathcal{A}) = \min_{i \in \mathcal{I}} R_i(\mathcal{A})$.

Our goal is, given a budget k on the number of sensors we can place, to find a placement

$$\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}| \leq k} R(\mathcal{A}), \tag{1}$$

where R is either R_{avg} or R_{adv}.

2 Sensor placement algorithms

Unfortunately, both problems are hard [3]. The key to obtaining approximate solutions is to realize that the objective functions R_i satisfy an important property, which we proved in [5]: Adding a sensor helps more if we have placed few sensors so far, and less if we already have placed many sensors. This property is formalized by the combinatorial concept of submodularity (*c.f.,* [7]). A set function F is called submodular, if for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $s \in \mathcal{V} \setminus \mathcal{B}$ it holds that $F(\mathcal{A} \cup \{s\}) -$

$F(\mathcal{A}) \geq F(\mathcal{B} \cup \{s\}) - F(\mathcal{B})$, i.e., adding s to $\mathcal{A}$ helps more than adding s to a superset $\mathcal{B}$. F is called nondecreasing, if for all $\mathcal{A} \subseteq \mathcal{B}$ it holds that $F(\mathcal{A}) \leq F(\mathcal{B})$.

A key result about submodular functions states that the greedy algorithm, which iteratively adds the sensor s to the set $\mathcal{A}$ of chosen locations such that $F(\mathcal{A} \cup \{s\})$ is maximized, is near-optimal: It is guaranteed to obtain a solution $\mathcal{A}_G$, which achieves at least a constant fraction of $(1 - 1/e) \approx 63\%$ of the optimal solution [7] – in fact, this is the best possible guarantee achievable in polynomial time unless $P = NP$ [1]. Since submodular functions are closed under nonnegative linear combinations, the average case objective R_{avg} is submodular as well, and hence, the greedy algorithm solves problem (1) near-optimally. It is also possible to use submodularity to obtain online bounds and speed up algorithms [5].

Unfortunately, the adversarial objective R_{adv}, which is far more relevant for computer security, is *not* submodular. In fact, it can be shown, that in this setting, the greedy algorithm performs arbitrarily badly. In [3], we consider the problem of solving Problem (1) for *arbitrary* nondecreasing submodular functions R_i. We develop the SATURATE algorithm, which is guaranteed to find a sensor placement $\mathcal{A}$, for which $R_{adv}(\mathcal{A}) \geq \max_{|\mathcal{A}'| \leq k} R_{adv}(\mathcal{A}')$, and $|\mathcal{A}| \leq \alpha k$ for some small α, i.e., finds a solution which obtains adversarial score at least as much as the optimal solution, at slightly increased cost. Similarly to the greedy algorithm, SATURATE is shown to be best-possible under reasonable complexity-theoretic assumptions [3].

3 Other applications and connection to machine learning

Sensor failures. The problem of maximizing the minimum over a set of submodular functions arises in other settings as well. For example, in the outbreak detection problem, sensors might *fail*, due to hardware failures or manipulation by an adversary. We can model this problem in the following way: Given a submodular function F (e.g., the utility for placing a set of sensors), and a set $\mathcal{B} \subseteq \mathcal{V}$, we define a new function $F_{\mathcal{B}}(\mathcal{A}) = F(\mathcal{A} \setminus \mathcal{B})$. This set function corresponds to the (reduced) utility if all the sensors at locations in $\mathcal{B}$ fail. It is easy to show that if F is nondecreasing and submodular, so is $F_{\mathcal{B}}$. Hence, the problem of optimizing sensor placements which are robust to sensor failures results in a problem of simultaneously maximizing a collection of submodular functions, e.g., for the worst-case failure of $k' < k$ sensors we solve $\max_{|\mathcal{A}| \leq k} \min_{|\mathcal{B}| \leq k'} F_{\mathcal{B}}(\mathcal{A})$.

In fact, we can combine probabilistic/adversarial outbreak scenarios with probabilistic/adversarial sensor failures in an arbitrary manner. For example, we can try to optimize for placements which are robust against an adversarial virus infection, with probabilistic sensor failures, and vice versa. The SATURATE algorithm can be applied to any such combination.

Connection to machine learning. One important problem in machine learning is feature selection. In feature selection, the goal is to select a subset of features which are informative with respect to, e.g., a given classification task. One objective frequently considered is the problem of selecting a set of features which maximize the information gained about the class variable Y after observing the features $\mathcal{A}$, $F(\mathcal{A}) = H(Y) - H(Y \mid \mathcal{A})$, where H denotes the Shannon entropy. In [4], it was shown, that in a large class of graphical models, the information gain $F(\mathcal{A})$ is in fact a submodular function. Now we can consider a setting, where an adversary can delete features which we selected (as considered, e.g., in [2]). The problem of selecting features robustly against such arbitrary deletion of, e.g., m features, is hence equivalent to the problem of maximizing $\min_{|\mathcal{B}| \leq m} F_{\mathcal{B}}(\mathcal{A})$, where $\mathcal{B}$ are the deleted features.

In [3], we draw other connections, e.g., to the problem of minimizing the maximum posterior variance in Gaussian Process regression and robust experimental design. We believe that the problem of maximizing an adverarially chosen submodular objective function is relevant to a variety of security and machine learning problems.

References

[1] Uriel Feige, *A threshold of ln n for approximating set cover*, Journal of the ACM **45** (1998), no. 4, 634 – 652.
[2] Amir Globerson and Sam Roweis, *Nightmare at test time: Robust learning by feature deletion*, ICML, 2006.
[3] A. Krause, B. McMahan, C. Guestrin, and A. Gupta, *Selecting observations against adversarial objectives*, Advances in Neural Information Processing Systems (Vancouver, Canada), 2007.
[4] Andreas Krause and Carlos Guestrin, *Near-optimal value of information in graphical models*, UAI, 2005.
[5] Andreas Krause, Jure Leskovec, Carlos Guestrin, Jeanne VanBriesen, and Christos Faloutsos, *Efficient sensor placement optimization for securing large water distribution networks*, Submitted to the Journal of Water Resources Planning an Management (2007).
[6] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance, *Cost-effective outbreak detection in networks*, 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007.
[7] G. Nemhauser, L. Wolsey, and M. Fisher, *An analysis of the approximations for maximizing submodular set functions*, Mathematical Programming **14** (1978), 265–294.