Algorithms and Uncertainty, Winter 2018/19 Lecture 9 (5 pages)

Stochastic Vertex Cover

Instructor: Thomas Kesselheim

In the analysis of online algorithms, we assumed that we have to make commitments right away. In practice often restrictions are not as strict. Just suppose you have to fly to New York City two months from now. You could either buy the ticket now for a cheap price or later on. Now the ticket is cheap but there is a chance that you actually cannot go on the trip. So, it might also make sense to wait and buy the ticket for a higher price when it is certain that you have to go.

This is a typical example of a multi-stage optimization problem. These are problems in which the optimization instance gets more and more concrete over time and decisions can be made on the way. There are both models with stochastic as well as adversarial inputs. Today, we will consider simple examples of such stochastic problems.

1 Stochastic Vertex Cover

Recall the standard offline weighted Vertex Cover problem: We are given a graph $G = (V, E)$ and vertex weights $(c_v)_{v \in V}$. We have to choose a subset $F \subseteq V$ of the vertices such that for each edge at least one endpoint is contained in F. That is, for all $\{u, v\} \in E$, we have $u \in F$ or $v \in F$. The objective is to minimize the sum of weights of selected vertices $\sum_{v \in F} c_v$.

In the stochastic version, the edge set E is uncertain. It is drawn from a known probability distribution. The probability that the edge set is E is given as p_E. Our algorithm knows the entire vector $(p_E)_E$ from the start. We assume that $p_E = 0$ for all except polynomially many sets E.

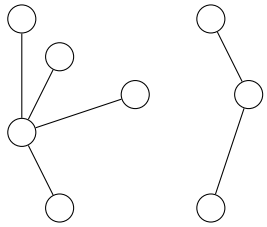
We can pick vertices at two points in time: Before the edge set E is revealed and afterwards. In the first stage, vertices are cheaper: For vertex v, we have to pay c_v^{I}. In the second stage, for vertex v, we have to pay $c_v^{\mathrm{II}} \geq c_v^{\mathrm{I}}$.

Important special cases are as follows. We might have $c_v^{\mathrm{I}} = c_v^{\mathrm{II}}$ for all v. In this case, choosing sets in the first stage does not make any sense and we might as well wait until the second stage. If $c_v^{\mathrm{II}} = \infty$, then we want to cover all edges that can possibly show up already in the first stage.

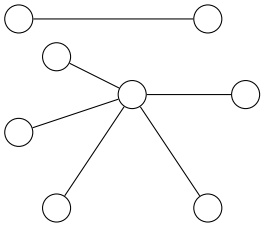
We know the distribution $(p_E)_E$ and well as both cost vectors $(c_v^{\mathrm{I}})_{v \in V}$ and $(c_v^{\mathrm{II}})_{v \in V}$ in advance. The goal is to minimize the expected cost

$$\sum_{v \text{ selected in first stage}} c_v^{\mathrm{I}} + \mathbf{E}\left[\sum_{v \text{ selected in second stage}} c_v^{\mathrm{II}}\right].$$

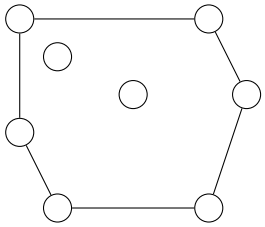
Example 9.1. *An example instance could look as follows. There is a fixed set of vertices, there are three scenarios, corresponding to different edges. The problem is already interesting if in the first stage every vertex costs $c_v^I = 1$ and in the second stage every vertex costs $c_v^{II} = \lambda$.*



$p_E = 0.2$



$p_E = 0.5$



$p_E = 0.3$

2 Our Goal

Observe that the stochastic vertex-cover problem can be modeled as a Markov decision process with time horizon $T = 2$. So, we could in principle use the algorithm based on dynamic programming to compute an optimal policy. However, the number of states will be huge. Computing it is at least as hard as solving the Vertex Cover problem optimally because one special case is that $p_E = 1$ for one set E. Vertex Cover is an NP-hard problem, so we cannot hope to find an exact algorithm that runs in polynomial time. Therefore, we will be interested in *approximating the optimal policy* in polynomial time.

Given any instance $\mathcal{I}$ of the problem, that is the probability distribution over edge sets and the different cost vectors, let $C_{\mathcal{I}}(\pi)$ denote the expected cost of policy π. There is an optimal policy $\pi^*_{\mathcal{I}}$ such that $C_{\mathcal{I}}(\pi^*_{\mathcal{I}})$. Our goal is to design a polynomial time algorithm with the following property. It is given an instance $\mathcal{I}$ and it is supposed to compute a policy π such that $C_{\mathcal{I}}(\pi) \leq \alpha \cdot C_{\mathcal{I}}(\pi^*_{\mathcal{I}})$, where $\alpha > 1$ is as small as possible.

Note that said $\pi^*_{\mathcal{I}}$ is *not* the offline optimum. Indeed, there is not a lot we can do if we are compared to the offline optimum. Suppose we have only a single edge e, which has to be covered with probability ϵ. Covering it in the first phase costs ϵ; in the second phase costs 1. Any policy has expected cost ϵ but the offline optimum has expected cost ϵ^2.

3 LP Relaxation

Our approach to approximating the optimal policy will be to first formulate a linear program that any policy has to fulfill but not every solution corresponds to a feasible policy. For the stochastic vertex-cover problem, we can write the following LP.

$$\begin{aligned} \min \sum_{v \in V} c^{\mathrm{I}}_v x_v + \sum_E p_E \sum_{v \in V} c^{\mathrm{II}}_v y_{E,v} & \\ \text{subject to } x_u + y_{E,u} + x_v + y_{E,v} \geq 1 & \qquad \text{for all } E,\ \{u,v\} \in E \\ x_u, y_{E,u} \geq 0 & \qquad \text{for all } E,\ u \in V \end{aligned}$$

Observe that we get a feasible solution by we setting $x_v = 1$ if the optimal policy chooses vertex v in the first stage and $y_{E,v} = 1$ if the optimal policy chooses vertex v in the second stage when the edge set is E. The objective function value is exactly the expected cost of the optimal policy.

4 A Simple Algorithm

Our approximation algorithm computes an optimal solution (x^*, y^*) to this LP. This can be done in polynomial time if $p_E > 0$ for only polynomially many sets E. This solution does not

necessarily correspond to a feasible policy because values can be fractional. We derive a feasible policy as follows.

- In the first stage, pick all vertices for which $x^*_v \geq \frac{1}{4}$.
- In the second stage, when knowing E, pick all vertices for which $y^*_{E,v} \geq \frac{1}{4}$.

Theorem 9.2. *The algorithm computes a feasible policy whose expected cost is at most* 4*-times the cost of the optimal policy.*

Proposition 9.3. *The algorithm always computes a feasible policy.*

Proof. Consider any scenario E and $e = \{u, v\} \in E$. As (x^*, y^*) is a feasible LP solution, we have

$$x^*_u + y^*_{E,u} + x^*_v + y^*_{E,v} \geq 1 \ .$$

This means that one of x^*_u, $y^*_{E,u}$, x^*_v, and $y^*_{E,v}$ is at least $\frac{1}{4}$. This means that edge e is covered in scenario E. □

Proposition 9.4. *The expected cost of the computed policy is at most* 4*-times the expected cost of the optimal policy.*

Proof. Let F_0 be the set of vertices picked by the computed policy in the first stage, F_E be the set of vertices picked in the second stage if the edge set is E.

We now have

$$\sum_{v \in F_0} c^{\mathrm{I}}_v \leq 4 \sum_{v \in V} c^{\mathrm{I}}_v x^*_v \qquad \text{and} \qquad \sum_{v \in F_E} c^{\mathrm{II}}_v \leq 4 \sum_{v \in V} c^{\mathrm{II}}_v y^*_{E,v} \ .$$

Therefore

$$\sum_{v \in F_0} c^{\mathrm{I}}_v + \mathbf{E}\left[\sum_{v \in F_E} c^{\mathrm{II}}_v\right] = \sum_{v \in F_0} c^{\mathrm{I}}_v + \sum_E p_E \sum_{v \in F_E} c^{\mathrm{II}}_v \leq 4 \left(\sum_{v \in V} c^{\mathrm{I}}_v x^*_v + \sum_E p_E \sum_{v \in V} c^{\mathrm{II}}_v y^*_{E,v}\right) \ .$$

As observed above, the cost of the optimal LP-solution is upper bounded by the expected cost of the optimal policy. □

5 Challenge: Large Number of Scenarios

One major challenge of the LP-based approach is that the the LP enumerates all scenarios explicitly. This way, the number of variables and number of constraints in the LP grows linearly in the number of scenarios. Having many scenarios is not as hypothetical as it might sound. For example, if each edge is present with probability $\frac{1}{2}$ independently, we would have $2^{\frac{n(n-1)}{2}}$ different scenarios and the LP gets huge. This happens despite the fact that the probability distribution over scenarios can be described very easily.

The first question that one should ask is: How should such a probability distribution be represented? The most general approach is to say that the algorithm does not have access to the scenarios explicitly. Instead, it has sample access to the distribution: It may draw from it as often as necessary and will always see only the drawn set E.

A standard algorithmic approach is called *sample-average approximation*. Draw N times from the distribution and set $\hat{p}_E$ to the fraction of times that scenario E was drawn. Now, run the algorithm pretending that the distribution is actually given by $(\hat{p}_E)_E$.

The key question is: How large do we have to choose N so that the sample is a good representative of the distribution? There are many results giving answers to this question, often in a much more general form. Here, we will give an example calculation, which has some weaknesses. See the paper by Charikar, Chekuri, and Pál for a stronger bound.

To formalize the question, let $X = \{(x_v)_{v\in V} \mid 0 \le x_v \le 1 \text{ for all } v\}$ be the set of all possible first-stage decision vectors x. For an assignment of the variables x in the LP, we define $f(x)$ to be the optimal LP value with respect probability distribution $(p_E)_E$, keeping x fixed. We let $\hat{f}(x)$ be the same quantity but with respect to the probability distribution $(\hat{p}_E)_E$.

Our algorithm uses a point x that minimizes $\hat{f}$, although it should actually minimize f.

Theorem 9.5. *Let $M = \max_{v\in V} c_v^I + c_v^{II}$. For all $\epsilon, \delta > 0$, if $N \ge \frac{9n^2M^2}{2\epsilon^2} \ln\left(\frac{2}{\delta}\left(\frac{3nM}{\epsilon}+1\right)^n\right)$, then*

$$\mathbf{Pr}\left[\textit{There is } x \in X \textit{ with } |\hat{f}(x) - f(x)| \ge \epsilon\right] \le \delta \ .$$

This means that minimizing $\hat{f}$ instead of f gives an additive error of less then 2ϵ with probability at least $1-\delta$. The biggest weakness is that the bound depends on M. So, it is only pseudo-polynomial.

Proof. We will proceed in three steps.

Step 1: The first step is to consider only a fixed $x \in X$. Let $g(x, E)$ be the cheapest way to cover all of E given that the (fractional) first-stage decision is fixed to x. Let $E_1, \ldots, E_N$ be the scenarios drawn for the sample-average approximation. By this definition, we have

$$\hat{f}(x) = \frac{1}{N}\sum_{i=1}^{N} g(x, E_i) \ .$$

Furthermore, $f(x) = \mathbf{E}[g(x, E)]$, where the expectation is over E. So, we can interpret $\hat{f}(x)$ as an average of N independent real-valued random variables. Their expectations are exactly $f(x)$. This is a clear case for Hoeffding's inequality.

Lemma 9.6 (Hoeffding's inequality)**.** *Let $Z_1, \ldots, Z_N$ be independent random variables such that $a_i \le Z_i \le b_i$ with probability 1. Let $\bar{Z} = \frac{1}{N}\sum_{i=1}^{N} Z_i$ be their average. Then for all $t \ge 0$*

$$\mathbf{Pr}\left[\left|\bar{Z} - \mathbf{E}\left[\bar{Z}\right]\right| \ge t\right] \le 2\exp\left(-\frac{2N^2t^2}{\sum_{i=1}^{N}(b_i-a_i)^2}\right) \ .$$

Setting $a_i = 0$, $b_i = nM$ for all i, we get for all $x \in X$ and all $t > 0$

$$\mathbf{Pr}\left[|\hat{f}(x) - f(x)| \ge t\right] \le 2\exp\left(-\frac{2Nt^2}{n^2M^2}\right) \ .$$

Step 2: This probability bound holds for every fixed $x \in X$ but we want the sums to be close for all x simultaneously. To get such a bound, we first approximate X by a *mesh* X'. The mesh X' contains only the points $x \in X$ for which x_v is a multiple of γ for every $v \in V$. Here, γ is chosen appropriately small. By this definition, X' is finite. More precisely, $|X'| = \left(\frac{1}{\gamma}+1\right)^n$.

Recall the union bound.

Lemma 9.7 (Union Bound)**.** *For any sequence of not necessarily disjoint events $\mathcal{E}_1, \mathcal{E}_2 \ldots$, we have*

$$\mathbf{Pr}\left[\mathcal{E}_1 \cup \mathcal{E}_2 \cup \ldots\right] \le \mathbf{Pr}\left[\mathcal{E}_1\right] + \mathbf{Pr}\left[\mathcal{E}_2\right] + \ldots \ .$$

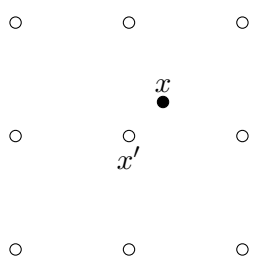


Figure 1: To bound the difference of $\hat{f}(x)$ and $f(x)$, we use x'. We know that x' is close, $\hat{f}(x')$ and $f(x')$ differ by at most t, and that f and $\hat{f}$ do not change drastically.

We can interpret $\exists x' \in X' : |\hat{f}(x') - f(x')| \geq t$ as such a sequence of events and get that for all $\gamma > 0$, $t > 0$

$$\mathbf{Pr}\left[\exists x' \in X' : |\hat{f}(x') - f(x')| \geq t\right] \leq \sum_{x' \in X'} \mathbf{Pr}\left[|\hat{f}(x') - f(x')| \geq t\right] \leq |X'| 2\exp\left(-\frac{2Nt^2}{n^2M^2}\right) \ .$$

Step 3: Now, we can move to all points. Given any $x \in X$, let $x' \in X'$ be the closest point in X'. By definition of M, we have $|f(x) - f(x')| \leq n\gamma M$ and also $|\hat{f}(x) - \hat{f}(x')| \leq n\gamma M$. By triangle inequality, if $|\hat{f}(x') - f(x')| < t$ for all $x' \in X'$, then we also have (see also Figure 1)

$$\begin{aligned} |\hat{f}(x) - f(x)| &= |\hat{f}(x) - \hat{f}(x') + \hat{f}(x') - f(x') + f(x') - f(x)| \\ &\leq |\hat{f}(x) - \hat{f}(x')| + |\hat{f}(x') - f(x')| + |f(x') - f(x)| \\ &\leq 2n\gamma M + t \ . \end{aligned}$$

Overall, this gives us that for all $t > 0$ and $\gamma > 0$

$$\mathbf{Pr}\left[\exists x \in X : |\hat{f}(x) - f(x)| \geq 2n\gamma M + t\right] \leq \left(\frac{1}{\gamma} + 1\right)^n 2\exp\left(-\frac{2Nt^2}{n^2M^2}\right)$$

Now, setting $t = \frac{\epsilon}{3}$ and $\gamma = \frac{\epsilon}{3nM}$, the bound follows. □

References

- On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems, N. Immorlica, D. Karger, M. Minkoff. V. Mirrokni, SODA 2004 (Vertex Cover)
- Sampling Bounds for Stochastic Optimization, M. Charikar, C. Chekuri, M. Pál, APPROX/RANDOM 2005 (Sample-Average Approximation)