Algorithms and Uncertainty, Winter 2018/19 Lecture 10 (4 pages)

Stochastic Set Cover

Instructor: Thomas Kesselheim

Last time, we introduced stochastic multi-stage optimization. Our goal is to compute a good policy, the difficulty being the enormous number of states and actions. In fact, an optimal policy solves the Vertex Cover problem optimally and therefore cannot be computed efficiently unless P = NP.

The technique that we used for stochastic Vertex Cover generalizes far beyond this single application. We will see this today in the context of Set Cover.

1 Stochastic Set Cover

Recall that in the offline Set Cover problem, there is a universe of m elements U and a family of subsets $\mathcal{S} \subseteq 2^U$. Each set $S \in \mathcal{S}$ has a cost c_S. We have to select a cover $\mathcal{C} \subseteq \mathcal{S}$ such that for all $j \in U$ there is some $S \in \mathcal{C}$ with $j \in \mathcal{S}$. We want to minimize the cost $\sum_{S \in \mathcal{C}} c_S$. We have already seen the LP relaxation before:

$$\begin{aligned} \text{minimize} \quad & \sum_{S \in \mathcal{S}} c_S x_S & & \\ \text{subject to} \quad & \sum_{S\colon e \in S} x_S \geq 1 & & \text{for all } e \in U \\ & x_S \geq 0 & & \text{for all } S \in \mathcal{S} \end{aligned}$$

In the stochastic version, only a subset $A \subseteq U$ has to be covered. That is, only for $j \in A$, there has to be $S \in \mathcal{C}$ with $j \in \mathcal{S}$. It is uncertain which set A it is. The set A is drawn from a known probability distribution. We denote by p_A, $A \subseteq U$, the probability that A has to be covered.

Eventually, we will have to cover all of A. We have two opportunities to select sets: Before A is revealed and afterwards. Before A is revealed (stage I), adding $S \in \mathcal{S}$ costs c_S^{I}; after A is revealed (stage II), it costs $c_S^{\mathrm{II}} \geq c_S^{\mathrm{I}}$.

We formulate an LP relaxation as follows. Given an arbitrary policy, let $x_S = 1$ if set S is selected in the first stage, 0 otherwise. Let $y_{A,S} = 1$ if set S is selected in the second stage if set A has to be covered, 0 otherwise. Based on these variables, we can extend the idea of Stochastic Vertex Cover by

$$\begin{aligned} \text{minimize} \quad & \sum_{S \in \mathcal{S}} c_S^{\mathrm{I}} x_S + \sum_{A \subseteq U} p_A \sum_{S \in \mathcal{S}} c_S^{\mathrm{II}} y_{A,S} & & \\ \text{subject to} \quad & \sum_{S\colon e \in S} x_S + \sum_{S\colon e \in S} y_{A,S} \geq 1 & & \text{for all } A \subseteq U,\ e \in A \\ & x_S, y_{A,S} \geq 0 & & \text{for all } S \in \mathcal{S},\ A \subseteq U \end{aligned}$$

It is easy to observe that every policy corresponds to an LP solution whose value is the expected cost of this policy. So, the optimal LP solution can only be cheaper than the optimal policy. Again, if $p_A > 0$ only for a small number of sets, we can solve this linear program in polynomial time. The results on sample-average approximation hold here as well.

Our approach for Vertex Cover does not work anymore: A constraint can have many variables. So, not necessarily any variable gets a values of at least $\frac{1}{4}$ (or any other other constant

value). This effect already takes place in the offline setting. So, let us understand this problem first.

2 Offline Set Cover and the Greedy Algorithm

Every solution to the Set Cover problem also corresponds to a feasible solution to the LP relaxation. However, the best *fractional* solution can be cheaper but not arbitrarily so. We have seen before how to round LP solutions to feasible Set Cover solutions. But there is an even easier approach: Run a simple greedy algorithm. We will show that its solution cost is also bounded in terms of the cheapest LP solution.

The greedy algorithm is truly simple. It works as follows

- Initially, set $U' := U$
- While $U' \neq \emptyset$
  - Let S be the set that minimizes $\frac{c_S}{|S \cap U'|}$
  - Add S to $\mathcal{C}$, set $U' := U' \setminus S$.

So, in every step, the algorithm chooses the set S of minimum cost *per newly covered element.*

Theorem 10.1. *Let $\mathcal{C}$ be the cover computed by the greedy algorithm, let x^* be the optimal solution to the LP relaxation. Then $\sum_{S \in \mathcal{C}} c_S \leq O(\log m) \sum_{S \in \mathcal{S}} c_S x^*_S$, where $m = |U|$.*

Proof. Every element gets removed from U' at some point. Let e_k be the k^{th} element that is removed from the set U', breaking ties arbitrarily. Element e_k gets removed from U' because it is covered by some S for the first time; later more sets covering e_k can follow, which we ignore. Let S_k denote this set S it is first covered by and let U'_k denote the state of U' at the beginning of the iteration in which e_k is removed.

We define

$$p_k = \frac{c_{S_k}}{|S_k \cap U'_k|}$$

as the cost per newly-covered element that we incur when covering element e_k. Note that while covering e_k we may cover elements for the first time as well and we split up the cost of set c_{S_k} evenly among them. By this definition, we can write the cost that our algorithm incurs as

$$\sum_{S \in \mathcal{C}} c_S = \sum_{k=1}^{m} p_k \ .$$

We claim that

$$p_k \leq \frac{\sum_{S \in \mathcal{S}} c_S x^*_S}{m - k + 1} \ . \tag{1}$$

This then implies

$$\sum_{S \in \mathcal{C}} c_S = \sum_{k=1}^{m} p_k \leq \sum_{k=1}^{m} \frac{\sum_{S \in \mathcal{S}} c_S x^*_S}{m - k + 1} = \sum_{S \in \mathcal{S}} c_S x^*_S \sum_{k=1}^{m} \frac{1}{k} = O(\log m) \sum_{S \in \mathcal{S}} c_S x^*_S \ ,$$

which is exactly what we claimed.

So, it only remains to show (1). Recall that S_k minimizes $\frac{c_S}{|S \cap U'_k|}$. That is, we can write

$$p_k = \min_S \frac{c_S}{|S \cap U'_k|} = \min_{e \in U'_k} \min_{S : e \in S} \frac{c_S}{|S \cap U'_k|} \ .$$

The last step looks a bit redundant but now we have two minimum operators that we can talk about. Note that any minimum is always upper-bounded by any (weighted) average. That is, we have

$$\min_{e\in U'_k} \min_{S:e\in S} \frac{c_S}{|S\cap U'_k|} \le \frac{1}{|U'_k|}\sum_{e\in U'_k} \min_{S:e\in S} \frac{c_S}{|S\cap U'_k|} \ .$$

Furthermore, because x^* is a feasible LP solution $\sum_{S:e\in S} x^*_S \ge 1$ for all e. So,

$$\min_{S:e\in S} \frac{c_S}{|S\cap U'_k|} \le \sum_{S:e\in S} x^*_S \frac{c_S}{|S\cap U'_k|} \ .$$

In combination, we have

$$p_k \le \frac{1}{|U'_k|}\sum_{e\in U'_k}\sum_{S:e\in S} x^*_S \frac{c_S}{|S\cap U'_k|} = \frac{1}{|U'_k|}\sum_{S:S\cap U'_k\neq\emptyset} c_S x^*_S \le \frac{1}{|U'_k|}\sum_{S\in\mathcal{S}} c_S x^*_S \ ,$$

where the second-to-last step is only a re-ordering of the sum.

Equation (1) now follows because $|U'_k| = m-k+1$ because before the k^{th} element is removed, there are at least $m-k+1$ left. There might be even more because other elements get removed in the same iteration. □

3 Algorithm for Stochastic Set Cover

Now, we can proceed to the multi-stage variant. As said before, our algorithm first solves the LP relaxation and obtain an optimal solution (x^*, y^*). To turn it into a policy as follows.

- Let U_0 be the set of all elements e such that $\sum_{S:\, e\in S} x^*_S \ge \frac{1}{2}$. Make sure to cover these elements in the first stage, e.g., by running the greedy algorithm on U_0 with costs $(c^{\text{I}}_S)_{S\in\mathcal{S}}$.
- Cover $A\setminus U_0$ in the second stage, e.g., by running the greedy algorithm on $A\setminus U_0$ with costs $(c^{\text{II}}_S)_{S\in\mathcal{S}}$.

The policy is clearly feasible because whatever A is drawn, each $e\in A$ is covered in the second stage at the latest.

Theorem 10.2. *The algorithm turns any fractional solution to the LP into a feasible policy of at most $O(\log m)$-times the cost in polynomial time.*

Proof. Let us understand the cost of the first stage of our policy. We defined it to cover U_0. The (deterministic) LP relaxation of this problem is the following.

$$\begin{aligned} \text{minimize} \sum_{S\in\mathcal{S}} c^{\text{I}}_S x_S & \\ \text{subject to} \sum_{S:\, e\in S} x_S \ge 1 & \qquad \text{for all } e\in U_0 \\ x_S \ge 0 & \qquad \text{for all } S\in\mathcal{S} \end{aligned}$$

Observe that $2x^*$ is a feasible solution by the way we defined U_0. So the optimal value is at most $2\sum_{S\in\mathcal{S}} c^{\text{I}}_S x^*_S$. This means that, by Theorem 10.1, our first-stage selection has a cost of at most

$$O(\log m)\cdot 2\sum_{S\in\mathcal{S}} c^{\text{I}}_S x^*_S \ .$$

In the second stage, we only have to cover $A \setminus U_0$. Observe that for all $e \in A \setminus U_0$

$$\sum_{S\colon e \in S} y^*_{A,S} \geq \frac{1}{2}$$

because (x^*, y^*) is a feasible LP solution. So, we can follow just the same idea as above and get a cover of cost at most

$$O(\log m) \cdot 2 \sum_{S \in \mathcal{S}} c^{\mathrm{II}}_S y^*_{A,S} \ .$$

In combination, our cover will cost in expectation

$$O(\log m) \cdot \left(\sum_{S \in \mathcal{S}} c^{\mathrm{I}}_S x^*_S + \sum_A p_A \sum_{S \in \mathcal{S}} c^{\mathrm{II}}_S y^*_{A,S} \right) \ .$$

□

References

- Stochastic optimization is (almost) as easy as deterministic optimization, D. Shmoys, C. Swamy, FOCS 2004 (Set Cover and generalizations)