Algorithms and Uncertainty, Winter 2017/18 Lecture 13 (4 pages)

Stochastic Steiner Tree

Instructor: Thomas Kesselheim

Today, we will consider another example of stochastic two-stage optimization. Last time, we considered LP-based approaches. One the the drawbacks of these approaches is that one always has to solve a linear program whose size depends on the number of scenarios. Today, we will consider a different kind of algorithm that is more combinatorial. The number of scenarios does not matter at all. Indeed, we only need to be able to draw samples from the same distribution that the scenario is generated from.

1 Problem Formulation

We consider a stochastic variant of the following rooted Steiner tree problem. In the deterministic offline problem, we are given a graph $G = (V, E)$, edge weights $w_e \geq 0$ for $e \in E$, a root $r \in V$, and a set of terminals $T \subseteq V$. Our task is to select a subset of the edges $S \subseteq E$ such that $\{r\} \cup T$ is connected in $G' = (V, S)$ and $\sum_{e \in S} w_e$ is minimized. Observe that if $T = V$ then this problem is exactly the minimum spanning tree problem. It is an NP-hard problem. Without loss of generality, $G = (V, E)$ is a complete graph. We can also assume that the weights w_e fulfill the triangle inequality. That is, $w_{\{u,v\}} \leq w_{\{u,x\}} + w_{\{x,v\}}$ for all $u, v, x \in V$. This is without loss of generality because we could instead take the detour via x instead of the edge $\{u, v\}$.

In the stochastic variant, we do not know the set T in advance but only the distribution it is drawn from. In the first stage, we do not yet know the set T but we can already pick edges e at costs w_e. In the second stage, we know the set T but edges are more expensive now: Picking edge e costs $\lambda \cdot w_e$ for $\lambda \geq 1$.

We know the probability distribution that T is drawn from. More precisely, it will only be necessary to be able to draw samples from the same distribution. Our goal is to minimize the expected cost

$$\sum_{e \text{ selected in first stage}} w_e + \mathbf{E}\left[\sum_{e \text{ selected in second stage}} \lambda \cdot w_e\right] \enspace .$$

Let us understand the limiting cases first: In the case $\lambda = 1$ it does not make sense to select anything in the first stage because it does not get more expensive in the second one. For $\lambda \to \infty$, the second stage gets extremely expensive, so we buy edges connecting every possible T in the first stage.

Again, even the basic Steiner tree problem is NP hard. Therefore, we cannot compute the optimal policy in polynomial time and we want to approximate it instead. More formally, let E_0^* we the set of edges selected by the optimal policy in the first stage, and let E_T^* we the set of edges selected by the optimal policy in the second stage if the set of terminals is T. We are looking for a policy whose expected cost is as close as possible to

$$Z^* := \sum_{e \in E_0^*} w_e + \mathbf{E}\left[\sum_{e \in E_T^*} \lambda \cdot w_e\right] \enspace .$$

2 Steiner Trees and Spanning Trees

Before coming to our algorithm, let us first prove a well-known result that Steiner trees can be approximated by minimum spanning trees. Such a spanning tree only uses edges between $\{r\} \cup T$ and no edges to other vertices (called Steiner vertices). Let $\mathrm{MST}(T) \subseteq E$ be the minimum spanning tree on $G|_{\{r\} \cup T}$ and let $\mathrm{Steiner}(T) \subseteq E$ be the optimal Steiner tree connecting $\{r\} \cup T$.

Lemma 13.1. *A minimum spanning tree on $G|_{\{r\} \cup T}$ is a 2-approximation for the min-cost Steiner tree on $\{r\} \cup T$, formally*

$$w(\mathrm{MST}(T)) \leq 2 \cdot w(\mathrm{Steiner}(T))$$

Proof. The idea is as follows: Traverse the optimal Steiner tree in a depth-first-search manner. You cross each edge twice: Once when entering the subtree and once when exiting it again. Equivalently, you can double each edge in the tree and consider an Euler tour through these duplicated tree edges. As each edge is crossed twice, the sum of edge costs on this run is $2 \cdot w(\mathrm{Steiner}(T))$.

We get a sequence of vertices that contains r and each terminal from T at least once. Consider the path that shortcuts this sequence by only visiting r and the vertices in T exactly once. By triangle inequality, this path can only be shorter, so the sum of edge costs is at most $2 \cdot w(\mathrm{Steiner}(T))$.

This path is a spanning tree of $G|_{\{r\} \cup T}$. The minimum spanning tree has at most its cost. □

3 Algorithm "Boosted Sampling"

We will consider the following algorithm called Boosted Sampling:

- In the first stage, draw λ times from the known distribution, call these sets $S_1, \ldots, S_\lambda$. Compute a minimum spanning tree on $\{r\} \cup S_1 \cup \ldots \cup S_\lambda$, let E_0 be the set of edges contained in it and pick them.
- In the second stage, set $w_e = 0$ for all $e \in E_0$ and compute a minimum spanning tree on $\{r\} \cup T$, let E_T be the set of contained edges not picked so far and pick them.

This algorithm only needs to sample λ times and calculate two minimum spanning trees. It therefore runs in polynomial time if λ is polynomially bounded.

Theorem 13.2. *The expected cost of the algorithm is at most $4Z^*$. That is,*

$$\mathbf{E}\left[\sum_{e \in E_0} w_e + \sum_{e \in E_T} \lambda \cdot w_e\right] \leq 4Z^*.$$

4 Analysis of First Stage

Lemma 13.3. *The expected first-stage cost of the algorithm is at most $2Z^*$. That is,*

$$\mathbf{E}\left[\sum_{e \in E_0} w_e\right] \leq 2Z^*.$$

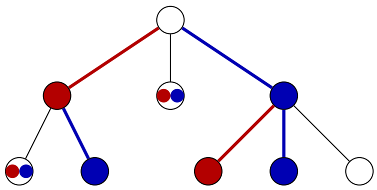

Figure 1: Illustration of Lemma 13.5 with two sets U_1 and U_2. Using only the red edges, each red vertex is connected to the root or a blue vertex, which we can connect for free, or is blue itself. The same holds if we swap red and blue.

Proof. Observe that $E_0^* \cup E_{S_1}^* \cup \ldots \cup E_{S_\lambda}^*$ is a feasible Steiner tree connecting all of $S_1 \cup \ldots \cup S_\lambda$ to the root r.

Our choice, $E_0 = \mathrm{MST}(S_1 \cup \ldots \cup S_\lambda)$ can have at most twice the cost, so

$$w(E_0) \leq 2w(E_0^* \cup E_{S_1}^* \cup \ldots \cup E_{S_\lambda}^*) = 2w(E_0^*) + 2\sum_{i=1}^{\lambda} w(E_{S_i}^*) \enspace .$$

By linearity of expectation, we have

$$\mathbf{E}\left[w(E_0)\right] \leq 2w(E_0^*) + 2\sum_{i=1}^{\lambda} \mathbf{E}\left[w(E_{S_i}^*)\right] \enspace .$$

Furthermore, observe that $\mathbf{E}\left[w(E_{S_i}^*)\right] = \mathbf{E}\left[w(E_T^*)\right]$ for all i because S_i and T are drawn from the same distribution. So

$$\mathbf{E}\left[w(E_0)\right] \leq 2w(E_0^*) + 2\lambda\mathbf{E}\left[w(E_T^*)\right] = 2Z^* \enspace .$$

□

5 Analysis of Second Stage

Lemma 13.4. *The expected second-stage cost of the algorithm is at most $2Z^*$. That is,*

$$\mathbf{E}\left[\sum_{e \in E_T} \lambda \cdot w_e\right] \leq 2Z^*.$$

To bound the cost incurred in the second stage, we have to understand how expensive it is to "augment" a spanning tree. Given $A, B \subseteq V$ let $\delta(A, B)$ be the cost of a minimum spanning tree on the graph $G|_{\{r\} \cup A \cup B}$ when setting $w_{\{u,v\}} = 0$ for all $u, v \in \{r\} \cup A$.

Lemma 13.5. *For any $U_1, \ldots, U_k \subseteq V$, we have*

$$\sum_{i=1}^{k} \delta\left(\bigcup_{j \neq i} U_j, U_i\right) \leq w(\mathrm{MST}(U_1 \cup \ldots \cup U_k)) \enspace .$$

Proof. Consider $\mathrm{MST}(U_1 \cup \ldots \cup U_k)$. Recall that this is a tree rooted at r. For $v \in U_1 \cup \ldots \cup U_k$, $v \neq r$, let a_v be the weight of the edge connecting v to its parent node in this tree.

Now, we can bound

$$\delta\left(\bigcup_{j \neq i} U_j, U_i\right) \leq \sum_{v \in U_i \setminus \bigcup_{j \neq i} U_j} a_v$$

because by connecting each $v \in U_i \setminus \bigcup_{j \neq i} U_j$ to its parent node and using the zero-weight edges all of $U_1 \cup \ldots \cup U_k$ is connected.

Therefore, we now have

$$\sum_{i=1}^{k} \delta\left(\bigcup_{j \neq i} U_j, U_i\right) \leq \sum_{i=1}^{k} \sum_{v \in U_i \setminus \bigcup_{j \neq i} U_j} a_v \leq \sum_{v \in \bigcup_i U_i} a_v = w(\mathrm{MST}(U_1 \cup \ldots \cup U_k)) \ . \qquad \square$$

Proof. In the second stage, we connect the set T by augmenting a minimum spanning tree on $\{r\} \cup S_1 \cup \ldots \cup S_\lambda$ to one that also includes the set T. Therefore

$$\sum_{e \in E_T} \lambda \cdot w_e = \lambda \cdot \delta(S_1 \cup \ldots \cup S_\lambda, T) \ .$$

We now perform a thought experiment: Note that $S_1, \ldots, S_\lambda$ and T are $\lambda + 1$ independent draws from the same distribution. So, equivalently, we might also draw $U_1, \ldots, U_{\lambda+1}$ from this distribution and then draw K uniformly from $\{1, \ldots, \lambda + 1\}$ and set $T = U_K$ and assign the other U_i sets arbitrarily to $S_1, \ldots, S_\lambda$.

Therefore, we can write

$$\mathbf{E}\left[\delta(S_1 \cup \ldots \cup S_\lambda, T)\right] = \mathbf{E}\left[\delta\left(\bigcup_{j \neq K} U_j, U_K\right)\right] = \mathbf{E}\left[\frac{1}{\lambda + 1} \sum_{i=1}^{\lambda+1} \delta\left(\bigcup_{j \neq i} U_j, U_i\right)\right] \ .$$

By Lemma 13.5, we have

$$\sum_{i=1}^{\lambda+1} \delta\left(\bigcup_{j \neq i} U_j, U_i\right) \leq w(\mathrm{MST}(U_1 \cup \ldots \cup U_{\lambda+1})) \ .$$

So, combining these arguments, the second-stage cost of our algorithm can be bounded by

$$\mathbf{E}\left[\sum_{e \in E_T} \lambda \cdot w_e\right] \leq \frac{\lambda}{\lambda + 1} \mathbf{E}\left[w(\mathrm{MST}(U_1 \cup \ldots \cup U_{\lambda+1}))\right] \ .$$

Again, $E_0^* \cup E_{U_1}^* \cup \ldots \cup E_{U_{\lambda+1}}^*$ is a feasible Steiner tree connecting $U_1 \cup \ldots \cup U_{\lambda+1}$ to the root, so the minimum spanning tree can have at most twice the cost, formally

$$\begin{aligned} w(\mathrm{MST}(U_1 \cup \ldots \cup U_{\lambda+1})) &\leq 2w(E_0^* \cup E_{U_1}^* \cup \ldots \cup E_{U_{\lambda+1}}^*) \\ &= 2w(E_0^*) + 2 \sum_{i=1}^{\lambda+1} w(E_{U_i}^*) \ . \end{aligned}$$

Again use linearity of expectation and $\mathbf{E}\left[w(E_{U_i}^*)\right] = \mathbf{E}\left[w(E_T^*)\right]$ to get

$$\mathbf{E}\left[w(\mathrm{MST}(U_1 \cup \ldots \cup U_{\lambda+1}))\right] \leq 2w(E_0^*) + 2(\lambda + 1)\mathbf{E}\left[w(E_T^*)\right] \leq 2\frac{\lambda + 1}{\lambda}\left(w(E_0^*) + \lambda \mathbf{E}\left[w(E_T^*)\right]\right) \ .$$

$\square$

Reference

Boosted sampling: approximation algorithms for stochastic optimization, A. Gupta, M. Pál, R. Ravi, A. Sinha, STOC 2004