Graded Homework 1, exercise 4

Marco Milanta

December 6, 2021

Number of Minimum Cuts (25 points)

In the class, it was shown that the number of minimum cuts of any unweighted graph is at most $\binom{n}{2}$. Here we want to discuss two generalizations of this result.

1. In the lecture we proved that the number of minimum cuts in any unweighted graph is at most $\binom{n}{2}$. Prove the same result for *weighted graphs.* All the weights are positive.

2. For a graph $G = (V, E)$, a subset of edges $E' \subseteq E$ is a k-cut if $G' = (V, E \setminus E')$ has at least k connected components. Prove that the number of minimum k-cuts in any unweighted graph is at most $2^{O(k)} n^{2k-2}$.

Solutions

1. Let $G = (V, E, w)$ be our weighted graph, $w : E \to \mathbb{R}^+$. Here $w(e)$ is the weight of the edge e. Let $n = \#V$. Let M be the minimum cut size.

   To prove that the number of minimum cut is at most $\binom{n}{2}$, we use a modified version of Karger's random contraction algorithm. When we sample edges, we don't do it uniformly, but the probability of sampling e from E' will be

   $$\Pr(\text{samplig } e \text{ from } E') = \frac{w(e)}{\sum_{e \in E'} w(e)}.$$

   We now prove the following lemma

   Lemma 1 (Karger's efficency for weighted graphs)**.** *Given a minimum cut $\{S, V \setminus S\}$ of size M. The probability that the modified Karger's algorithm selects it is at least $\frac{1}{\binom{n}{2}}$*

   Proof. Let $\tilde{E}$ be the set of edges that cut $\{S, V \setminus S\}$. We know that Krager's algorithm runs in no more than $n - 2$ steps. We say that the algorithm fails if at any timestap it samples an element from $\tilde{E}$. Note that if the algorithm doesn't fail it outputs $\{S, V \setminus S\}$. To have more insight about why this is true one can look into Karger's algorithm on the lecture notes chapter 13.

   Failing at step i: We now look at the probability of failing at step i assuming we didn't fail before.

   The probability of failing corresponds to the probability of sampling $e \in \tilde{E}$ from E_i. E_i is the set of edges in the contracted graph, V_i is the set of vertices which have not been contradiction. Note that $\#V_i = n - i + M$. Note that $\sum_{e \in E_i} w(e) \geq \frac{(n-i+1)M}{2}$. To show this we rewrite it as

$$\begin{aligned}
\sum_{e \in E_i} w(e) &= \frac{1}{2} \sum_{v \in V_i} \sum_{e \text{ adj. to } v} w(e) \\
&\geq \frac{1}{2} \sum_{v \in V_i} M \\
&= \frac{1}{2} \#V_i M = \frac{1}{2}(n - i + 1)M
\end{aligned}$$

Where in the first equality we rewrite the sum looking vertex by vertex at its outgoing edges. In this way, however, we count every vertex twice, therefore we add a $\frac{1}{2}$ factor to compensate. In the first line we use the fact that $\sum_{e \text{ adj. to } v} w(e)$ is just the size of the cut $\{\{e\}, V \setminus \{e\}\}$, therefore it must be larger than M.

Now, the probability of failing is

$$\begin{aligned}
\Pr\left(\text{sampling } e \in \tilde{E} \mid e \in E_i\right) &= \sum_{e \in \tilde{E}} \Pr\left(\text{sampling } e \mid e \in E_i\right) \\
&= \sum_{e \in \tilde{E}} \frac{w(e)}{\sum_{e \in E'} w(e)} && \text{modified algoritm} \\
&\leq \frac{\sum_{e \in \tilde{E}} w(e)}{\frac{(n-i+1)M}{2}} = \frac{M}{\frac{(n-i+1)M}{2}} = \frac{2}{n - i + 1}. && \text{using the result above}
\end{aligned}$$

Never failing: The probability of never failing is

$$\begin{aligned}
\Pr(\text{never failing}) &= \Pr(\text{not fail at step } n-2 \mid \text{it didn't fail before } n-2) \cdots \Pr(\text{not fail at step } 1) \\
&= \prod_{i=1}^{n-2} \Pr(\text{not fail at step } i \mid \text{didn't fail before } i) \\
&= \prod_{i=1}^{n-2} \left(1 - \Pr(\text{fail at step } i \mid \text{didn't fail before } i)\right) \\
&\geq \prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) \\
&= \prod_{i=1}^{n-2} \left(\frac{n-i-1}{n-i+1}\right) \\
&= \frac{\cancel{n-2}}{n} \frac{\cancel{n-3}}{n-1} \frac{\cancel{n-4}}{\cancel{n-2}} \cdots \frac{\cancel{3}}{\cancel{5}} \frac{2}{\cancel{4}} \frac{1}{\cancel{3}} = \frac{2}{n(n-1)} = \frac{1}{\binom{n}{2}}.
\end{aligned}$$

Since the probability of not failing, is the probability to sample $\{S, V \setminus S\}$, we have concluded the proof. $\square$

Finally, it's very easy to prove that there are no more than $\binom{n}{2}$ minimum cuts. By contradiction, assume that there were more than $\binom{n}{2}$ minimum cuts. Each one of them will have a probability of $\frac{1}{\binom{n}{2}}$ to be the output of Karger's algorithm. This means, that the total probability of Karger's to output one of the minimum cuts is

$$\begin{aligned}\Pr(\text{sample a min. cut}) &= \sum_{i=1}^{\text{num. of min. cuts}>\binom{n}{2}} \Pr(\text{the } i\text{-th min. cut is selected}) \\ &\geq \sum_{i=1}^{\text{num. of min. cuts}>\binom{n}{2}} \frac{1}{\binom{n}{2}} > 1\end{aligned}$$

Where in the second line we used lemma 1. Of course having a probability to be more than 1 is a contradiction. This implies that the total number of minimum cuts cannot be more than $\binom{n}{2}$.

2. The proof of this part follows a very similar strategy to the one seen before. First we need and important lemma.

Lemma 2 (Bound number of edges)**.** *Let $G = (V, E)$ be a graph (or multigraph) where, $n = \#V$ and the minimum $k-$cut has size M. Then*

$$\#E \geq \frac{M}{1 - \left(1 - \frac{k-1}{n}\right)\left(1 - \frac{k-1}{n-1}\right)}.$$

Proof. □

Now, we propose to use the Karger's algorithm but stopping after $n - k$ iterations. The outputs will be a graph with k nodes where each node correspond to a subset of V. The k subsets will represent the selected k-cut. Now we can prove the key theorem

Theorem 1 (Karger's efficency for k-cuts)**.** *Given a graph $G = (V, E)$ where $\#V = n$. Let*

$$C = \left\{A_1, A_2, \ldots, A_{k-1}, V \setminus \left(\bigcup_{i=1}^{k-1} A_i\right)\right\}, \text{ where } A_i \cap A_j = \emptyset \ \forall i \neq j$$

be a minimum k-cut. The probability that the early-stopped (after $n - k$ iterations) Karger's algorithm outputs C is

$$\Pr\left(\textit{Krager's outputs } C\right) \geq c\frac{(k-1)^{2(k-1)}}{(ne)^{2k-2}}$$

Proof. Let $\tilde{E}$ be the set of edges that cut C. Let $M = \#\tilde{E}$. We say that the algorithm fails if at some point it picks an edge from $\tilde{E}$. If it does so, the algorithm will not output C. If it does not, then it must output C. We divide the proof in two parts:

Failing at step i: We now look at the probability of failing at step i given that it has not failed before. Let $G_i = (V_i, E_i)$ the multigraph output of the previous steps. Note that the minimum cut of such graph must be M. Furthermore, note that $\#V_i = n - i + 1$ (at step 1 $V_1 = V$, and then we take out one node per iteration).

Thanks to lemma 2 we are guaranteed that

$$\#E_i \geq \frac{M}{1 - \left(1 - \frac{k-1}{\#V_i}\right)\left(1 - \frac{k-1}{\#V_i - 1}\right)} = \frac{M}{1 - \left(1 - \frac{k-1}{n-i+1}\right)\left(1 - \frac{k-1}{n-i}\right)}.$$

Let $\hat{e}$ be the edge sampled at the i-th step. Since we sample uniformly at random from E_i, we get that

$$\Pr\left(\hat{e} \in \tilde{E}\right) \leq \frac{\#\tilde{E}}{\#E_i} = \frac{M}{\frac{M}{1-\left(1-\frac{k-1}{n-i+1}\right)\left(1-\frac{k-1}{n-i}\right)}} = 1 - \left(1 - \frac{k-1}{n-i+1}\right)\left(1 - \frac{k-1}{n-i}\right).$$

Never failing: We now look at the probability that it never fails:

$$
\begin{aligned}
\Pr(\text{never failing}) &= \Pr(\text{not fail at step } n-2 \mid \text{it didn't fail before } n-2)\cdots\Pr(\text{not fail at step } 1)\\
&= \prod_{i=1}^{n-k} \Pr(\text{not fail at step } i \mid \text{didn't fail before } i)\\
&= \prod_{i=1}^{n-k} \left(1 - \Pr(\text{fail at step } i \mid \text{didn't fail before } i)\right)\\
&\geq \prod_{i=1}^{n-k} \left(\cancel{1-1} + \left(1 - \frac{k-1}{n-i+1}\right)\left(1 - \frac{k-1}{n-i}\right)\right)\\
&= \prod_{i=1}^{n-k} \left(\frac{n-i-k+2}{n-i+1}\right)\left(\frac{n-i-k+1}{n-i}\right)\\
&= \underbrace{\prod_{i=1}^{n-k} \left(\frac{n-i-k+2}{n-i+1}\right)}_{=A}\underbrace{\prod_{i=1}^{n-k} \left(\frac{n-i-k+1}{n-i}\right)}_{=B}
\end{aligned}
$$

Now we look at A and B independently. For A we have:

$$
\begin{aligned}
A &= \prod_{i=1}^{n-k} \left(\frac{n-i-k+2}{n-i+1}\right)\\
&= \frac{n-k+1}{n}\frac{n-k}{n-1}\frac{n-k-1}{n-2}\cdots\frac{n-2k+2}{n-k+1}\cdots\frac{k-1}{2k-4}\cdots\frac{4}{k+3}\frac{3}{k+2}\frac{2}{k+1}\\
&= \frac{\cancel{n-k+1}}{n}\frac{\cancel{n-k}}{n-1}\frac{\cancel{n-k-1}}{n-2}\cdots\frac{\cancel{n-2k+2}}{\cancel{n-k+1}}\cdots\frac{\cancel{k+1}}{\cancel{2k}}\cdots\frac{4}{\cancel{k+3}}\frac{3}{\cancel{k+2}}\frac{2}{\cancel{k+1}}\\
&= \frac{k!}{\frac{n!}{(n-k+1)!}} = \frac{k(k-1)!}{\frac{n!}{(n-k+1)!}} = \frac{k}{\frac{n!}{(n-(k-1))!(k-1)!}} = \frac{k}{\binom{n}{k-1}}.
\end{aligned}
$$

For B we have:

$$
\begin{aligned}
B &= \prod_{i=1}^{n-k} \left(\frac{n-i-k+1}{n-i}\right)\\
&= \frac{n-k}{n-1}\frac{n-k-1}{n-2}\frac{n-k-2}{n-3}\cdots\frac{n-2k+1}{n-k}\cdots\frac{k}{2k-1}\cdots\frac{3}{k+2}\frac{2}{k+1}\frac{1}{k}\\
&= \frac{\cancel{n-k}}{n-1}\frac{\cancel{n-k-1}}{n-2}\frac{\cancel{n-k-2}}{n-3}\cdots\frac{\cancel{n-2k+1}}{\cancel{n-k}}\cdots\frac{\cancel{k}}{\cancel{2k-1}}\cdots\frac{3}{\cancel{k+2}}\frac{2}{\cancel{k+1}}\frac{1}{\cancel{k}}\\
&= \frac{(k-1)!}{\frac{(n-1)!}{(n-k)!}} = \frac{1}{\frac{(n-1)!}{(n-1-(k-1))!(k-1)!}} = \frac{1}{\binom{n-1}{k-1}}
\end{aligned}
$$

Finally, we can bound the probability of never failing by using a well known bound on the binomial coefficient: $\binom{n}{k} \leq \frac{n^k}{k!} \leq \left(\frac{ne}{k}\right)^k$.

$$
\begin{aligned}
\Pr(\text{never failing}) \geq AB &= \frac{k}{\binom{n}{k-1}\binom{n-1}{k-1}}\\
&\geq \frac{k}{\left(\frac{ne}{k-1}\right)^{k-1}\left(\frac{(n-1)e}{k-1}\right)^{k-1}}\\
&\geq \frac{k}{\left(\frac{ne}{k-1}\right)^{2k-2}}\\
&\geq \frac{k(k-1)^{2k-2}}{(ne)^{2k-2}}
\end{aligned}
$$

Since if the algorithm never fails, then it picks exactly C; we have concluded the proof. $\square$

Once we have theorem 1, then the result is easy. Indeed, we cannot have more than

$$\text{Number of minimum } k\text{-cuts} \leq \frac{(ne)^{2k-2}}{k(k-1)^{2k-2}}.$$

If, by contradiction we assume we had more, we get that the total probability of getting one of the minimum cuts is more than 1, which is impossible (For a similar argument more in detail look at the last part of the previous point). Now, rearranging some terms we get:

$$\begin{aligned}\text{Number of minimum } k\text{-cuts} &\leq \frac{(ne)^{2k-2}}{k(k-1)^{2k-2}} \\ &= e^{O(k)} n^{2k-2} \frac{1}{k(k-1)^{2k-2}} \leq 2^{O(k)} n^{2k-2}.\end{aligned}$$

Where $e^{O(k)} = 2^{O(k)\log_2 e} = 2^{O(k)}$, and $\frac{1}{k(k-1)^{2k-2}} \in O(1)$. This concludes the exercise. As a final note, one can notice that, $\frac{e^{2k-2}}{k(k-1)^{2k-2}}$ is bounded by a constant. A tighter bound could then be then obtained:

$$\text{Number of minimum } k\text{-cuts} \leq Cn^{2k-2} \in O(n^{2k-2})$$