CS 388R: Randomized Algorithms, Fall 2023 8-28-23

Lecture 3: Faster min-cut and estimating a biased coin

Prof. Eric Price *Scribe: Alekhya Kuchimanchi, Kristin Sheridan*

1 Overview

In the first lecture, we discussed Karger's min cut algorithm, which finds a global min cut with probability at least $1-\delta$ in $O(n^4 \log \frac{1}{\delta})$ time. This lecture, we will see a modification of this algorithm that works in $\tilde{O}(n^2 \log \frac{1}{\delta})$ time and finds a min cut with probability at least $1-\delta$.

In the second lecture, we discussed applications of tail bounds and saw that if we make $n \geq O(\frac{p}{\epsilon^2} \cdot \log \frac{1}{\delta})$ tosses of a biased coin that lands on heads with probability p and output $\hat{p}$, the number of heads divided by n, we will have $|\hat{p} - p| \leq \epsilon$ with probability a least $1-\delta$. In this lecture, we will show how to modify this technique such that we do not need n to depend on p. In particular, this allows us to get around needing a bound on p in order to know when to stop.

2 A faster min cut algorithm

Review First, recall Karger's min cut algorithm.

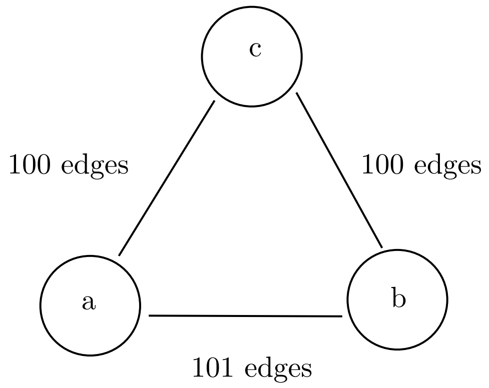
Algorithm 1 Karger's min cut algorithm

```
Input: An unweighted graph $G = (V, E)$, probability $\delta \in (0, 1)$
Output: A global min cut of $G$, with probability at least $1 - \delta$
  for $i = 1$ to $n - 2$ do
      Pick a random edge and contract it
  Return pre-images of the two remaining vertices as a cut
```

Note that when we contract an edge, we retain *all* resulting edges between the resulting two vertices (so we actually obtain a multi-graph as we begin contracting). As previously discussed, the above algorithm runs in $O(n^2)$ time and succeeds with probability at least $\frac{2}{n^2}$. Thus, with $O(n^2 \log \frac{1}{\delta})$ repetitions, we can obtain a min cut with probability $1-\delta$ and in total we take $O(n^4 \log \frac{1}{\delta})$ time.

Intuition Note that in our previous analysis, the highest failure probability came from the final steps. For example, consider the following multi-graph. To obtain a min cut on this multi-graph, we must merge a and b, but the probability of merging b and c or a and c instead is $200/301$, which is almost $2/3$. Thus, even if we got lucky early on in our random choices and we made only perfect choices up to this point, there is high probability we fail at this point. In particular, in our analysis we showed that the probability of succeeding at the ith round given we succeeded at all previous rounds is at least $(1 - \frac{2}{n-(i-1)})$, which is small when i is small but gets quite large as i gets large.

Thus, we would like to modify Karger's algorithm so that we can retain the work from the first steps while re-running the last steps multiple times.



New algorithm The modified version of Karger's algorithm is the following recursive algorithm.

```
Algorithm 2 Modified Karger's min cut algorithm
Input: An unweighted graph $G = (V, E)$ of $n$ nodes
Output: A global min cut of $G$, with probability at least $1/2$
  If $n = 2$, return the only cut
  for $t = 1$ to $2$ do
      $G' \leftarrow G$
      for $i = 1$ to $(1 - \frac{1}{\sqrt{2}})n$ do
          Modify by $G'$ by picking a random edge and contracting it
      $G_t \leftarrow$ Algorithm 2$(G')$
  Return the pre-image of the vertices in $G_1$ or $G_2$, whichever produces the smaller cut
```

Runtime of Algorithm 2 If $T(n)$ is the runtime of the algorithm on a graph of n nodes, we have

$$T(n) = T\left(\frac{n}{\sqrt{2}}\right) + O(n^2).$$

We take $O(n^2)$ steps in the upper layer of the recursion, as there are $O(n)$ steps, each at a cost of $O(n)$, and the number of nodes at the next layer is $n-(1-\frac{1}{\sqrt{2}})n = \frac{n}{\sqrt{2}}$. Using the Master Theorem, this implies an overall runtime of $T(n) = O(n^2 \log n)$. (Note that the same amount of work is done at each level of the recursion tree, so we can multiply by the number of layers.)

Success probability of Algorithm 2 Let $p(n)$ be the probability that Algorithm 2 succeeds on an input graph of n nodes. Then we have

$$\begin{aligned} p(n) &= 1 - (\text{probability one branch fails})^2 \\ &= 1 - (1 - \text{probability one branch succeeds})^2 \\ &\geq 1 - (1 - \frac{1}{2}p(\frac{n}{\sqrt{2}}))^2, \end{aligned}$$

where (probability one branch fails) is the probability that G_t does not produce a min cut for some choice of t. The first equality is because both cuts must not be min in order for the algorithm to fail. The second is because a branch succeeds if and only if it does not fail. Finally, we need only show that the probability one branch succeeds is at least $\frac{1}{2}p(\frac{n}{\sqrt{2}})$ in order to obtain the above inequality. We proceed via a similar method to that we saw in the previous lecture, but with early termination. In particular, as discussed last time, the probability that we succeed in round i given we succeeded in all previous rounds is at least $1 - \frac{2}{n-(i-1)} = \frac{n-(i+1)}{n-(i-1)}$. We can write the probability of success as follows, where A_i is the event that we find success on round i (i.e. the probability that we select an edge to collapse such that both end points are in the same side of the optimum min cut, for some optimum min cut we select). From the analysis last time, we have $\mathbb{P}[A_i | \cap_{j<i} A_j] \geq \frac{n-(i+1)}{n-(i-1)}$

$$\begin{aligned} \mathbb{P}[\text{a branch succeeds}] &= \mathbb{P}[A_1] \cdot \mathbb{P}[A_2|A_1] \cdot \mathbb{P}[A_3|A_2 \cap A_1] \cdots \\ &\quad \cdots \mathbb{P}[A_{(1-\frac{2}{\sqrt{2}})n} | \cap_{j<(1-\frac{2}{\sqrt{2}})n} A_j] \cdot \mathbb{P}[\text{recursive call succeeds} | \cap_{j\leq(1-\frac{2}{\sqrt{2}})n} A_j] \\ &\geq \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdots \frac{n-(n-\frac{n}{\sqrt{2}}+1)}{n-(n-\frac{n}{\sqrt{2}}-1)} \cdot p\left(\frac{n}{\sqrt{2}}\right) \\ &= \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdots \frac{\frac{n}{\sqrt{2}}-1}{\frac{n}{\sqrt{2}}+1} \cdot p\left(\frac{n}{\sqrt{2}}\right) \\ &= \frac{1}{n} \cdot \frac{1}{n-1} \cdot \left(\frac{n}{\sqrt{2}}-2\right) \cdot \left(\frac{n}{\sqrt{2}}-1\right) \cdot p\left(\frac{n}{\sqrt{2}}\right) \\ &\geq \frac{\left(\frac{n}{\sqrt{2}}-2\right)^2}{n^2} \cdot p\left(\frac{n}{\sqrt{2}}\right) \\ &\approx \frac{1}{2} \cdot p\left(\frac{n}{\sqrt{2}}\right) \end{aligned}$$

where the last inequality holds when $n \geq 16/\sqrt{2}$ (i.e. for sufficiently large n).

Consider solving the recursion

$$\begin{aligned} p(n) &\geq 1 - (1 - \frac{1}{4}p(n/\sqrt{2}))^2 \\ &= \frac{1}{2} \cdot p\left(\frac{n}{\sqrt{2}}\right) - \frac{1}{16} \cdot p\left(\frac{n}{\sqrt{2}}\right)^2. \end{aligned}$$

Let $X_i := p(\sqrt{2}^i)$. Note that if $i \geq 1/2$, then $X_i \geq 1/i$. In particular, we get that $\frac{1}{i+1} \geq \frac{4i-1}{4i^2}$ if and only if $4i^2 \leq (4i-1)(i+1)$. At the lowest level (base case), $i = 2$ and at the highest level $i = \log_{\sqrt{2}} n$, so overall we get that the probability $p(n) \geq \frac{1}{\log_{\sqrt{2}} n} = \frac{1}{\Theta(\log n)}$.

Thus, we can repeat $O(\log n \log \frac{1}{\delta})$ times and take the best result, and we will get a min cut with probability $1 - \delta$. Total runtime is $O(n^2 \log n) \cdot O(\log n \log \frac{1}{\delta}) = \tilde{O}(n^2 \log \frac{1}{\delta})$. (Recall that $\tilde{O}$ mean we are hiding poly-log factors.)

3 Estimating a biased coin

Consider a biased coin that is heads with unknown probability p. Last class, we considered methods to approximate p. In particular, let X_i be 1 if coin toss i is heads and 0 otherwise. Then we let $X = \sum_{i=1}^n X_i$ and output $\hat{p} := \sum_{i=1}^n X_i/n$ (the empirical average). We saw that there exists a constant c such that for any $\epsilon, \delta > 0$ if $n \geq c \cdot \frac{p}{\epsilon^2} \log \frac{1}{\delta}$, then with probability at least $1 - \delta$, $|p - \hat{p}| \leq \epsilon$.

However, a major issue with the above analysis is that the number of coin tosses we need to examine depends on p. If we know nothing about p, we won't know when to stop. In this lecture, we will see how to modify this method so that we can decide when to stop without knowing anything about p.

Modified algorithm: Toss the biased coin and let X_i be 1 if and only if the result is heads on round i. Let n be the number of coins tossed so far. If $\sum_{i=1}^n X_i \geq 2c \cdot \frac{1}{\epsilon^2} \log \frac{2}{\delta}$, stop and output $\sum_{i=1}^n X_i/n$.

Note that we are now stopping when the number of heads seen exceeds a particular value, rather than when the number of coins tossed exceeds a specific value. We will analyze the number of samples we expect to take and the probability we are within a given range.

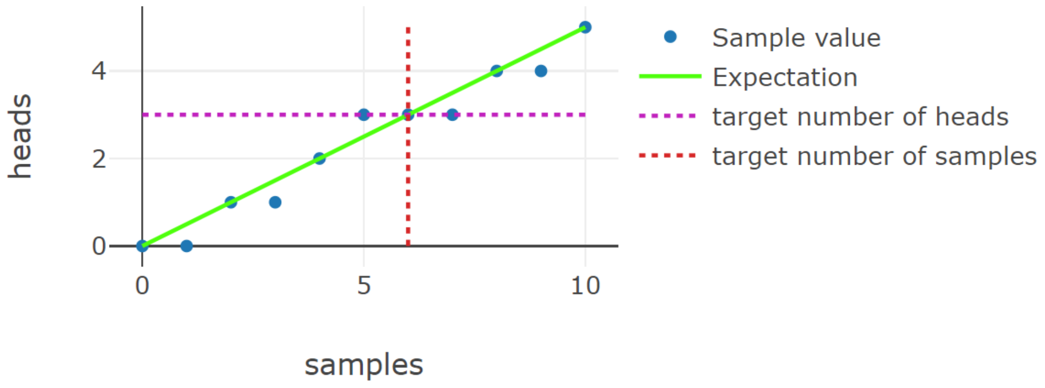
Number of samples: If we toss coins until we see k heads, we expect to toss k/p coins (Pascal RV/extension of geometric RV). Thus, we expect to stop after $4 \cdot \frac{1}{p\epsilon^2} \log \frac{2}{\delta}$ samples.

Correctness: We now analyze the probability that we output the correct answer. In particular, we would like to show that with probability at least $1 - \delta$, $\hat{p} \in [(1-\epsilon)p, (1+\epsilon)p]$.

Analysis from last lecture: Consider $\mathbb{P}[|\sum_{i=1}^n X_i - pn| \geq \epsilon pn]$ for any choice of n coin tosses. In particular, by applying Chernoff bounds we get the following

$$\begin{aligned}\mathbb{P}[|\sum_{i=1}^n X_i - pn| \geq \epsilon pn] &\leq 2e^{-\epsilon^2 pn/3} \\ &\leq \delta,\end{aligned}$$

where the first inequality is by standard Chernoff bounds and last inequality holds when $n \geq \frac{3}{\epsilon^2 p} \log \frac{2}{\delta}$. We can view the whole process as a function that randomly goes up by 1 or stays the same (with probability p and $1-p$, respectively). We wanted to know the probability that the function value is close to its expectation at the time when we first cross a horizontal line, as in the following figure. Now, we want to show that the function value is likely close to its expectation at the time when we first cross particular vertical line.



New analysis: Fix a target T and let n be the expected time that we reach target T - i.e. T/p. Consider values $n_{lo} := \frac{T}{p(1+\epsilon)}$ and $n_{hi} := \frac{T}{p(1-\epsilon)}$. We show that if $n < n_{lo}$, then it is unlikely that we have reached the target $X = \sum_{i=1}^{n} X_i$ value and if $n > n_{hi}$, we have probably exceeded this target value. Thus, we are most likely to cross this target threshold at some time between n_{lo} and n_{hi} and we are most likely to get a total of $pn = T$ heads sometime in this range.

If we stop at time n_{lo} or later, we approximate $\hat{p}$ as at most $T/n_{lo} = Tp(1+\epsilon)/T = p(1+\epsilon)$. If we stop at time n_{hi} or earlier, we approximate $\hat{p}$ at at least $T/n_{hi} = Tp(1-\epsilon)/T = p(1-\epsilon)$. Thus, if we successfully reach this target T for the first time at some time between n_{lo} and n_{hi} coin tosses, our estimated probability is within ϵ of the correct probability.

Now we must select the target T such that we have a high probability of this target first being reached between n_{lo} and n_{hi} tosses. In particular, we will set $\mathbb{P}[\sum_{i=1}^{n_{hi}} X_i \leq T] \leq \delta/2$ and $\mathbb{P}[\sum_{i=1}^{n_{lo}} X_i \geq T] \leq \delta/2$, which by the union bound implies that with probability at least δ we reach T heads after time n_{lo} and before n_{hi}. Note that $\mu := E[\sum_{i=1}^{n_{hi}} X_i] = pn_{hi} = T/(1-\epsilon)$, so $T = (1-\epsilon)\mu$

$$\begin{aligned}\mathbb{P}[\sum_{i=1}^{n_{hi}} X_i \leq T] &= \mathbb{P}[\sum_{i=1}^{n_{hi}} X_i \leq (1-\epsilon)\mu] \\ &\leq e^{-\epsilon^2\mu/3} \\ &= e^{-\epsilon^2 T/(3(1-\epsilon))}.\end{aligned}$$

Setting this to be at most $\delta/2$ and solving for T, we get that this holds for $T \geq \frac{3(1-\epsilon)}{\epsilon^2} \cdot \ln\frac{2}{\delta}$. Since $\epsilon > 0$, it is sufficient to pick $T \geq \frac{3}{\epsilon^2} \cdot \ln\frac{2}{\delta}$.

Similarly, we get $\mu := E[\sum_{i=1}^{n_{lo}} X_i] = pn_{hi} = T/(1+\epsilon)$, so $T = (1+\epsilon)\mu$ and since $\epsilon < 1$ (or else the problem is trivial), we get

$$\begin{aligned}\mathbb{P}[\sum_{i=1}^{n_{lo}} X_i \geq T] &= \mathbb{P}[\sum_{i=1}^{n_{lo}} X_i \geq (1+\epsilon)\mu] \\ &\leq e^{-\epsilon^2\mu/2} \\ &= e^{-\epsilon^2 T/(2(1+\epsilon))}.\end{aligned}$$

This is at most $\delta/2$ if $T \geq \frac{2(1+\epsilon)}{\epsilon^2} \cdot \ln\frac{2}{\delta}$. It is sufficient to pick $T \geq \frac{4}{\epsilon^2} \cdot \ln\frac{2}{\delta}$ by our bound on ϵ.

Thus, if we pick $T \geq \frac{4}{\epsilon^2} \cdot \ln\frac{2}{\delta}$, we get that with probability at least $1-\delta$, we reach the target threshold T at some point between $\frac{T}{p(1+\epsilon)}$ and $\frac{T}{p(1-\epsilon)}$, and thus our estimate $\hat{p} \in [(1-\epsilon)p, (1+\epsilon)p]$, as desired