AM 221: Advanced Optimization Spring 2018

Dr. Rasmus Kyng *Problem Set 10 — Due Wednesday, April. 25th at 23:59*

Instructions:

- All your solutions should be prepared in LaTeX and the PDF and .tex should be submitted to Canvas. Please submit all your files as ONE archive of filetype zip, tgz, or tar.gz.
- Name the file [your-first-name]_[your-last-name].[filetype]. For example, I would call my submission rasmus_kyng.zip.
- INCLUDE your name in the submisson pdf and any files with code.
- If the TFs cannot easily deduce your identity from your files alone, they may decide not to grade your submission.
- For each question, a well-written and correct answer will be selected a sample solution for the entire class to enjoy. If you prefer that we do not use your solutions, please indicate this clearly on the first page of your assignment.

1. Randomized Algorithm for Maximum Cut: Success with High Probability. In class, we saw an approximation algorithm for the Maximum Cut problem (see the lecture on April 11th).

MAX CUT is the following discrete optimization problem: Given an undirected graph $G = (V, E)$, find a set of vertices $S \subseteq V$ (a.k.a. a *cut*) that maximizes $C(S) := |\{u, v\} \in E : u \in S \text{ and } v \in V \setminus S|$.

Let $S^* \subseteq V$ be an optimal solution to the MAX CUT problem.

In class, we proved that the Goemans-Williamson rounding algorithm produces a *random* cut $\hat{S} \subseteq V$ s.t. $\mathbb{E}\, C(\hat{S}) \geq \alpha C(S^*)$ for a constant $\alpha > 0.5$.

(a) Show that for an non-negative random variable Z, s.t. $Z \leq K$ where $\mathbb{E}\, Z \geq K/2$ and $K \geq 1$,

$$\mathbb{P}[Z \geq \mathbb{E}\, Z - 1] \geq \frac{1}{10K}.$$

Hints:

- Use Markov's inequality on $K - Z$.

(b) Use the result you just established to prove that if we run the Goemans-Williamson rounding algorithm independently $10|E| \log(1/\delta)$ times for some $\delta \in (0, 1)$, then with probability at least $1 - \delta$, one of the cuts has value at least $\alpha C(S^*) - 1$.

2. Randomized Algorithm for Maximum Cut: A Stronger Result for Large Cuts. This exercise also deals with the MAX CUT problem (see Problem 1.). MAX CUT is the following discrete optimization problem: Given an undirected graph $G = (V, E)$, find a set of vertices $S \subseteq V$ (a.k.a. a *cut*) that maximizes $C(S) := |\{u, v\} \in E : u \in S \text{ and } v \in V \setminus S|$. Let $S^* \subseteq V$ be an optimal solution to the MAX CUT problem. Let $m = |E|$, and let us index the vertices so that $V = [n]$.

In class (see the lecture on April 11th), we learnt that algorithm exists for solving the Goemans Williamson Semi-Definite Program in polynomial time. The program, in its vector form, is stated below.

$$\begin{aligned} &\max_{\mathbf{u}_1,\ldots,\mathbf{u}_n \in \mathbb{R}^n} \sum_{\{i,j\}\in E} \frac{1}{4}||\mathbf{u}_i - \mathbf{u}_j||^2 \\ &\forall i \in [n].||\mathbf{u}_i||^2 = 1. \end{aligned} \tag{1}$$

In class, we studied a rounding algorithm based on choosing a random hyperplane with normal vector $\mathbf{g} \in \mathbb{R}^n$ and defining a graph cut $\hat{S} \subseteq V$ by $\hat{S} = \{i \in V : \mathbf{u}_i^\intercal \mathbf{g} \geq 0\}$. Recall that we can obtain uniformly random hyperplane through the origin by letting each entry $\mathbf{g}(i)$ be iid. $N(0, 1)$ distributed (standard normal distribution).

In class, we prove that $\mathbb{E}\, C(\hat{S}) \geq \alpha C(S^*)$ for a constant $\alpha > 0.5$.

Suppose the maximum cut of the graph has value $C(S^*) = (1 - \epsilon)m$ for some scalar $\epsilon \in (0, 1/2]$.

(a) Prove that the SDP always takes value at most m.

(b) Prove that $\mathbb{E}\, C(\hat{S}) \geq (1 - \sqrt{\varepsilon})m$.

Hints:

- You may find it helpful to note that $\frac{1}{\pi}\arccos(2x - 1) \geq 1 - \sqrt{x}$.
- To relate the average probability of cutting an edge to the average value of $\frac{1}{4}||\mathbf{u}_i - \mathbf{u}_j||^2$, you may find Jensen's Inequality helpful.

(c) What factor approximation to the true objective does the algorithm get (in expectation) when $\epsilon = 10^{-4}$? Is this better than 0.878?

3. Properties of Submodular Functions.

a. Recall that for a finite set N, a function $f : 2^N \to \mathbb{R}$ is defined to be *submodular* if $\forall S, T \subseteq N$, such that $S \subseteq T$ and $\forall u \in N \setminus T$, the following holds:

$$f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$$

Prove that any non-negative submodular function is also subadditive, i.e. if $f : 2^N \to \mathbb{R}_{\geq 0}$ is submodular then $f(S \cup T) \leq f(S) + f(T)$ for any $S, T \subseteq N$.

b. Prove that a function $f : 2^N \to \mathbb{R}_{\geq 0}$ is submodular if and only if for any $S \subseteq N$ the marginal contribution function f_S defined by $f_S(T) = f(S \cup T) - f(S)$ (for $T \subseteq N$) is subadditive.

Algorithm 1 Greedy Algorithm

1: Set $S = \emptyset$
2: **while** $|S| < k$ **do**
3: $\quad S \leftarrow S \cup \text{argmax}_{a \in N} f_S(a)$
4: **end while**
5: **return** S

4. Approximate Oracles. Recall the greedy algorithm for maximizing a non-negative monotone submodular function described below.

For a given $\alpha < 1$ and set $S \subseteq N$ we will say that $\tilde{f}_S$ is an α-approximate marginal oracle if for any $T \subseteq N$ we have the guarantee that $\alpha f_S(T) \leq \tilde{f}_S(T) \leq f_S(T)$, for a given constant $\alpha < 1$. Prove that a greedy algorithm like the one above which uses an α-approximate marginal oracle in every iteration in line (3) provides a $1 - 1/e^{\alpha}$ approximation to the optimal solution.