CS264: Homework #4

Due by midnight on Thursday, February 9, 2017

Instructions:

(1) Form a group of 1-3 students. You should turn in only one write-up for your entire group. See the course site for submission instructions.

(2) Please type your solutions if possible and feel free to use the LaTeX template provided on the course home page.

(3) All students should complete all of the exercises. Students taking the course pass-fail do not need to do any problems. **For students taking the course for a letter grade, we'll grade the Problems out of a total of 40 points.** Any points you receive in excess of 40 will be treated as extra-credit points.

(4) Write convincingly but not excessively. Exercise solutions rarely need to be more than 1-2 paragraphs. Problem solutions rarely need to be more than a half-page (per part), and can often be shorter.

(5) You may refer to your course notes, and to the textbooks and research papers listed on the course Web page *only*. You cannot refer to textbooks, handouts, or research papers that are not listed on the course home page. (Exception: feel free to use your undergraduate algorithms textbook.) Cite any sources that you use, and make sure that all your words are your own.

(6) If you discuss solution approaches with anyone outside of your team, you must list their names on the front page of your write-up.

(7) Exercises are worth 5 points each. Problem parts are labeled with point values.

(8) No late assignments will be accepted.

Lecture 7 Exercises

Exercise 19

Let $(\hat{\mathbf{x}}, \hat{\mathbf{d}})$ be an optimal solution to the linear programming relaxation of the Min Multiway Cut problem described in lecture, and recall the randomized rounding algorithm described in the lecture.

Multiway Cut Rounding Algorithm

1. Initially, all vertices are unassigned.
2. While there is at least one unassigned vertex:
   (a) Pick a threshold $r \in (0, 1)$ uniformly at random.
   (b) Pick a group $i \in \{1, 2, \ldots, k\}$ uniformly at random.
   (c) For every unassigned vertex v, if $\hat{d}_v^i \geq r$, assign vertex v to group i.

For an edge $e = (u, v)$, call an iteration *relevant for* e if at least one of the vertices u, v is assigned in that iteration. Prove that, conditioned on an iteration being the first one relevant for the edge $e = (u, v)$, the probability that the chosen coordinate in this iteration is $i \in \{1, 2, \ldots, k\}$ equals

$$\frac{\max\{\hat{d}_u^i, \hat{d}_v^i\}}{\sum_{j=1}^k \max\{\hat{d}_u^j, \hat{d}_v^j\}}.$$

Exercise 20

Continuing the previous exercise, prove that the probability that edge $e = (u, v)$ is cut is at most

$$\frac{\sum_{i=1}^k \left(\max\{\hat{d}_u^i, \hat{d}_v^i\} - \min\{\hat{d}_u^i, \hat{d}_v^i\}\right)}{\sum_{j=1}^k \max\{\hat{d}_u^j, \hat{d}_v^j\}}.$$

Conclude that edge e is cut with probability at most

$$\frac{2\hat{x}_e}{1 + \hat{x}_e}$$

and, hence, is cut with probability at most $2\hat{x}_e$ and is not cut with probability at least $\frac{1}{2}(1 - \hat{x}_e)$.

Lecture 8 Exercises

Exercise 21

Recall the linear programming relaxation of the Maximum Cut problem described in lecture:

$$\max \sum_{(i,j)\in E} w_{ij} x_{ij} \tag{1}$$

subject to:

$$x_{ij} \in [0, 1] \qquad \text{for all } i, j \in V \tag{2}$$

$$x_{ij} + x_{jk} + x_{ik} \leq 2 \qquad \text{for all } i, j, k \in V \tag{3}$$

$$x_{ik} \leq x_{ij} + x_{jk} \qquad \text{for all } i, j, k \in V. \tag{4}$$

Assume that the following lemma holds (see Problems 12 and 13 for a proof).

Lemma 1 *Consider an instance of the* Maximum Cut *problem, where F^* denotes the edges cut in some optimal solution, and $\hat{\mathbf{x}}$ denotes a feasible solution to the linear programming relaxation* (1)–(4)*. There exists a scaling parameter $\sigma > 0$ and randomized algorithm $\mathcal{A}$ that outputs a random partition $C = (A, B)$ such that:*

(i) for every edge $e = (i, j) \notin F^*$,

$$\mathbf{Pr}[e \text{ cut by } C] \geq \sigma \cdot \frac{\hat{x}_{ij}}{\alpha},$$

where $\alpha = \Theta(\log n)$ (and $n = |V|$);

(ii) for every edge $e = (i, j) \in F^*$,

$$\mathbf{Pr}[e \text{ not cut by } C] \leq \sigma \cdot (1 - \hat{x}_{ij});$$

(iii) $\mathcal{A}$ is deterministic (i.e., always outputs the same partition) if and only if $\hat{\mathbf{x}}$ is integral.

Use this lemma to prove the following theorem.

Theorem 2 *For α as in Lemma 1, in all α-perturbation-stable instances of the* Maximum Cut *problem, every optimal solution to the linear program* (1)–(4) *is integral.*

[Hint: follow closely the argument in Lecture #7 for exact recovery in 4-perturbation-stable instances of Min Multiway Cut.]

Exercise 22

Fix an instance $G = (V, E, w)$ of the Maximum Cut problem, a cut $C = (A, B)$ of G, and a feasible solution $\hat{\mathbf{x}}$ to the linear program in (1)–(4). Define $\hat{\mathbf{y}}$ by

$$\hat{y}_{ij} = \begin{cases} \hat{x}_{ij} & \text{if } i,j \text{ are on the same side of } C; \\ 1 - \hat{x}_{ij} & \text{if } i,j \text{ are on different sides of } C. \end{cases}$$

Prove that $\hat{\mathbf{y}}$ satisfies the triangle inequality:

$$\hat{y}_{ik} \leq \hat{y}_{ij} + \hat{y}_{jk}$$

for every $i, j, k \in V$.

Exercise 23

Say that an algorithm $\mathcal{A}$ achieves *certifiable exact recovery* for a set $\mathcal{I}$ of instances if: (i) for every instance in $\mathcal{I}$, $\mathcal{A}$ computes an optimal solution; and (ii) if A does not compute an optimal solution, then it "knows" that the instance it was given is not in $\mathcal{I}$ (i.e., can correctly declare this at its termination). Does the single-link++ algorithm for 2-stable k-median instances (Lecture #6) achieve certifiable exact recovery? What about the linear programming solution for 4-stable Min Multiway Cut instances (Lecture #7)? What about the LP and SDP solutions for γ-stable Maximum Cut instances, with $\gamma = \Theta(\log n)$ and $\Theta(\sqrt{\log n} \log \log n)$, respectively (Lecture #8)?

Problems

Problem 11

Recall the Min Multiway Cut linear programming relaxation from Lecture #7. By an *(α, β)-good rounding algorithm*, we mean a randomized algorithm that takes as input a feasible solution $(\hat{\mathbf{x}}, \hat{\mathbf{d}})$ to the LP relaxation, and outputs a random multiway cut $C = (S_1, \ldots, S_k)$ such that, for every edge $e \in E$:

$$\mathbf{Pr}[e \text{ is cut by } C] \leq \alpha \cdot \hat{x}_e; \tag{5}$$

and

$$\mathbf{Pr}[e \text{ is not cut by } C] \geq \frac{1}{\beta} \cdot (1 - \hat{x}_e). \tag{6}$$

The argument in lecture shows that, if an (α, β)-good rounding algorithm exists, then the optimal solution to the LP relaxation is integral for every $\alpha\beta$-perturbation-stable instance.[1] Exercises 19 and 20 show that the rounding algorithm given at the end of Lecture #7 is a $(2, 2)$-good rounding algorithm.

[1] Actually, we also needed a third property, that the rounding algorithm is deterministic if and only if $(\hat{\mathbf{x}}, \hat{\mathbf{d}})$ is integral. We won't worry about this property in this problem.

(a) (5 points) Prove that a rounding algorithm that satisfies (5) is a randomized α-approximation algorithm for the MIN MULTIWAY CUT problem, meaning that on every instance, given an optimal solution $(\hat{\mathbf{x}}, \hat{d})$ to the LP relaxation, it outputs a (random) feasible solution with expected cost at most α times the cost of an optimal solution.

[Hint: linearity of expectation.]

(b) (10 points) Consider the following modified rounding algorithm.

Multiway Cut Rounding Algorithm #2

1. Initially, all vertices are unassigned.
2. Choose a random permutation $\pi : \{1, 2, \ldots, k\} \to \{1, 2, \ldots, k\}$.
3. Pick a threshold $r \in (0, 1)$ uniformly at random.
4. For $i = 1, 2, \ldots, k-1$:
   (a) For every unassigned vertex v, if $\hat{d}_v^{\pi(i)} \geq r$, assign vertex v to group $\pi(i)$.
5. Assign all still-unassigned vertices to the group $\pi(k)$.

Prove that this rounding algorithm satisfies (5) with $\alpha = \frac{3}{2}$. (And so by (a) is a $\frac{3}{2}$-approximation algorithm.)

[Hints: fix $e = (u, v)$ and first suppose that $\hat{d}_u^i = \hat{d}_v^i$ in all but two coordinates, say h and j. It should not be difficult to verify (5) with $\alpha = 2$. To improve this to $\alpha = \frac{3}{2}$, note that for some values of r, e will be cut for exactly one of the two possible relative orders of h and j in π.]

(c) (10 points) For which values of β does this rounding algorithm satisfy (6)?

Problem 12

A *metric space* is a set X equipped with a distance function $d : X \times X \to \mathbb{R}^+$ such that $d(x, x) = 0$ for every $x \in X$, $d(x, y) = d(y, x)$ for every $x, y \in X$, and $d(x, y) \leq d(x, z) + d(z, y)$ for every $x, y, z \in X$. A restricted class of metric spaces (X, d), called ℓ_1 *metrics*, are those that can be represented as a subset of some real space $\mathbb{R}^k$ with the ℓ_1 norm. That is, there should be a mapping $\sigma : X \to \mathbb{R}^k$ for some $k \in \mathbb{N}$ such that $d(x, y) = \sum_{i=1}^k |\sigma(x)_i - \sigma(y)_i|$ for every $x, y \in X$.

An *embedding* of one metric space (X, d) into another (X', d') is just a map $\sigma : X \to X'$. If for some $\alpha > 0$ it holds that

$$d'(\sigma(x), \sigma(y)) \in \left[\frac{1}{\alpha} \cdot d(x, y), d(x, y)\right]$$

for every $x, y \in X$, then the embedding has *distortion* α. Thus a low-distortion embedding approximately represents the metric space (X, d) as the (possibly simpler) metric space (X', d').[2]

Theorem 3 (Bourgain's Theorem) *Every n-point metric space admits an embedding into an ℓ_1 metric space with distortion $O(\log n)$.*

This problem will walk you through the proof. We first describe the (randomized) embedding, which ascribes k coordinates to each point $x \in X$. Each coordinate will have the form (i, j), where i and j are positive integers between 1 and $\Theta(\log n)$ (and so $k = \Theta(\log^2 n)$).[3] In the description below, c is a constant; you can set it as large as you want for your proof, as long as it is independent of (X, d).

[2] One simple example is dimension reduction (e.g., if you've heard of the Johnson-Lindenstrauss (JL) Lemma), where (X, d) are points in Euclidean space (with d equal to Euclidean distance), and (X', d') is again a Euclidean space, but with fewer dimensions.

[3] We omit floors and ceilings for clarity.

Bourgain's Embedding

1. For $i = 1, 2, \ldots, c \log n$:
   (a) For $j = 1, 2, \ldots, \log n$:
      i. Choose a subset $S_{ij} \subseteq X$ at random, with each point of X included independently with probability 2^{-j}.
      ii. Define the (i, j) coordinate of each point $x \in X$ as $d(x, S_{ij}) := \min_{y \in S_{ij}} d(x, y)$.

That is, the embedding σ maps x to the k-vector $(d(x, S_{11}), d(x, S_{12}), \ldots)$. It remains to analyze the distortion. The first part gives an upper bound on how much the embedding can change distances; the remaining parts give a lower bound.

(a) (4 points) Prove that, with probability 1, for every $x, y \in X$,

$$\|\sigma(x) - \sigma(y)\|_1 \leq k \cdot d(x, y),$$

where $k = c \log^2 n$ is the number of coordinates.

(b) (10 points) Fix two points $x, y \in X$. For $t = 0, 1, 2, \ldots$, let r_t be the smallest number r such that there are at least 2^t points within distance r of x, and also at least 2^t points within distance r of y. Let T be the first index t for which $r_t + r_{t-1} > d(x, y)$, and redefine $r_T = d(x, y) - r_{T-1}$. For $j \in \{1, 2, \ldots, T\}$, call a set S_{ij} *good (for x, y)* if either:

(i) S_{ij} includes at least one point within distance r_{j-1} of x and no points within distance strictly less than r_j of y; or

(ii) S_{ij} includes at least one point within distance r_{j-1} of y and no points within distance strictly less than r_j of x.

Prove that there is a constant c' such that, with probability at least $1 - \frac{1}{n^3}$, for every $j \in \{1, 2, \ldots, T\}$ and for at least $c' \log n$ choices of $i \in \{1, 2, \ldots, c \log n\}$, the set S_{ij} is good (for x, y).

[Hint: you probably want to make use of the Chernoff bound; see the background resources on the course Web site.]

(c) (7 points) Prove that there is a constant c'' such that, with probability at least $1 - \frac{1}{n}$, for every $x, y \in X$,

$$\|\sigma(x) - \sigma(y)\|_1 \geq c'' \log n \cdot d(x, y).$$

(d) (4 points) Using parts (a)–(c), complete the proof of Bourgain's theorem.

Problem 13

Bourgain's theorem above is phrased in terms of ℓ_1 embeddings, while in lecture we phrased it in terms of randomized partitioning algorithms. This problem provides the translation.

(a) (9 points) A metric space (X, d) is called a *cut metric* if there exists a set $S \subseteq X$ such that $d(x, y) = 0$ whenever $x, y \in S$ or $x, y \notin S$ and $d(x, y) = 1$ whenever exactly one of x, y is in S.

Prove that, for a finite set X, a metric space (X, d) is an ℓ_1 metric if and only if it is a nonnegative linear combination of cut metrics (each with point set X).

(b) (6 points) Use part (a) and the version of Bourgain's theorem in Problem 12 to prove the version of Bourgain's theorem stated in class: if (X, d) is an n-point metric space, then there exists a randomized

algorithm (outputting a partition $C = (A, B)$ of X) and a scaling parameter $\lambda > 0$ such that, for every pair $x, y \in X$,

$$\mathbf{Pr}[x, y \text{ are on different sides of } C] \in \lambda \cdot \left[\frac{d(x,y)}{\alpha}, d(x,y)\right],$$

where $\alpha = \Theta(\log n)$.

Problem 14

(15 points) The goal of this problem is to identify a fundamental barrier to the exact recovery of optimal solutions in $O(1)$-perturbation-stable instances of the Maximum Cut problem.

In an instance of the Sparsest Cut problem, the input consists of a vertex set V and two sets of (possibly overlapping) edges, E_u and E_b. Each edge $e \in E_u$ has a *capacity* u_e. Each edge $e \in E_b$ has a *demand* b_e. The *sparsity* $\phi(A, B)$ of a cut (A, B) of G (with $A, B \neq \emptyset$) is

$$\phi(A,B) = \frac{\sum_{e \in E_u \cap \delta(C)} u_e}{\sum_{e \in E_b \cap \delta(C)} b_e},$$

where $\delta(C)$ denotes the set of edges with one endpoint on either side of the cut C. The objective of the problem is to compute the smallest sparsity of any cut.[4]

In the *γ-gap version* of the Sparsest Cut problem, the goal is to distinguish between instances with minimum sparsity less than 1 and those with minimum sparsity at least γ. An algorithm for this version of the problem should report "yes" whenever the optimal sparsity is less than 1 and "no" whenever the optimal sparsity is strictly larger than γ. If the optimal sparsity is between 1 and γ, then the algorithm is regarded as correct no matter what it says. It is known that, assuming the "Unique Games Conjecture," a plausible strengthening of the $P \neq NP$ conjecture, for every constant $\gamma \geq 1$, there is no polynomial-time algorithm that solves the γ-gap version of the Sparsest Cut problem.

Prove that, for every constant $\gamma \geq 1$, there is a polynomial-time algorithm for certifiable exact recovery (in the sense of Exercise 23) in γ-perturbation-stable Maximum Cut instances only if there is a polynomial-time algorithm for the γ-gap version of the Sparsest Cut problem.

[Hint: Given an instance $G = (V, E_u, E_b, u, b)$ of the Sparsest Cut problem, construct an instance of the Maximum Cut problem that has two copies V_1 and V_2 of V, a very heavy-weight edge between the two copies of each vertex, a copy of E_b (with weights equal to demands) inside each of V_1 and V_2, and two copies of E_u (with weights equal to capacities) going between V_1 and V_2.]

Problem 15

In the *Metric Traveling Salesman Problem (MTSP)*, the input is a complete undirected graph $G = (V, E)$, with a nonnegative cost $c_e \geq 0$ for each edge $e \in E$, and where the edge costs satisfy the triangle inequality ($c_{ij} \leq c_{ik} + c_{kj}$ for every $i, j, k \in V$). By a *TSP tour*, we mean a simple cycle that visits each vertex exactly once. The goal is to compute the TSP tour with the minimum total cost.

A *γ-perturbation* of an MTSP instance is a new set c' of edge costs with the property that $c'_e \in [\frac{1}{\gamma} c_e, c_e]$ for every $e \in E$.[5] An MTSP instance is *γ-perturbation-stable* if there is a TSP tour T such that, for every γ-perturbation of the instance, T is the unique optimal TSP tour.

[4] If we take E_b to be the complete edge set and set $b_e = 1$ for every $e \in E_b$, then the Sparsest Cut problem is more-or-less striving for a roughly balanced cut (with a constant fraction of vertices on either side) that cuts as little edge capacity as possible.

For further context, the Sparsest Cut problem can be thought of as an (integral version of) the dual to a natural network flow problem: for every $e = (i, j) \in E_b$ (simultaneously), route b_e units of flow between i and j in the network $G = (V, E_u)$ while respecting the capacities of the edges in E_u. If there is a cut with sparsity less than 1, then this problem is infeasible (why?).

[5] For this problem, and in contrast to our discussion of the metric k-median problem in Lecture #6, we allow γ-perturbations to violate the triangle inequality. This makes the perturbation-stability assumption stronger, and hence algorithmic results weaker.

(a) (5 points) Prove that for every $\gamma > 2$, there do not exist any γ-perturbation-stable MTSP instances with 4 or more vertices.

[Thus, the interesting regime to consider is values of γ between 1 and 2.]

(b) (10 points) Consider the following algorithm for constructing a tour.

Constructing a TSP Tour

1. Initialize the collection $\mathcal{C}$ with a singleton set corresponding to each vertex of G. [Invariant: $\mathcal{C}$ will consist of a collection of simple paths, with each vertex in exactly one path.]
2. While $\mathcal{C}$ contains more than one path:
   (a) Let e be the cheapest edge that connects the endpoints of two distinct paths of $\mathcal{C}$, say paths P_1 and P_2. Remove the paths P_1 and P_2 from $\mathcal{C}$, and add the path $P_1 \cup \{e\} \cup P_2$ to $\mathcal{C}$.
   [When this loop completes, $\mathcal{C}$ consists solely of one Hamiltonian path.]
3. Extend the Hamiltonian path P in $\mathcal{C}$ to a TSP tour by adding the missing edge between P's endpoints.

Prove that there is a constant $\gamma < 2$ such that this algorithm recovers the optimal tour in all γ-perturbation-stable instances of MTSP.

(c) (5 points) Prove that, for every $\gamma < \frac{5}{3}$, there exists a γ-perturbation-stable instance of MTSP such that the algorithm above will not recover the optimal tour.

[Hint: Let G be a graph consisting of a Hamiltonian cycle plus one chord (with all unit edge lengths), and consider the shortest-path metric defined by G.]

(d) (15 extra credit points) Devise a polynomial-time algorithm that, for some $\gamma < \frac{5}{3}$, recovers the optimal tour in every γ-perturbation-stable instance of MTSP.