CS264: Homework #4

Due by midnight on Wednesday, October 22, 2014

Instructions:

(1) Form a group of 1-3 students. You should turn in only one write-up for your entire group.

(2) Turn in your solutions at `http://rishig.scripts.mit.edu/cs264-bwca/submit-paper.html`. You can contact the TA (Rishi) at `bwca-staff@lists.stanford.edu`. Please type your solutions if possible and feel free to use the LaTeX template provided on the course home page.

(3) Students taking the course pass-fail should complete 5 of the Exercises and can skip the Problems. **Students taking the course for a letter grade should complete 8 of the Exercises. We'll grade the Problems out of a total of 40 points; if you earn more than 40 points on the Problems, the extra points will be treated as extra credit.**

(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. 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 26

Recall the bad example showing that single-link clustering fails. There are three sets of M points each (cities), and a fourth set with 10 points (a village). Within one of these sets, all points are co-located (with distance zero from each other). The first and second cities are connected by an edge of cost 1, the second and third cities by an edge of cost 10, and the third city to the village by an edge of cost 2. Prove that for every $\gamma \geq 1$, one can choose M sufficiently large so that this example is γ-stable.

Exercise 27

Show that the single-link++ algorithm can be implemented in polynomial time. Specifically, prove that, given the hierarchical cluster tree H and a value of $k \geq 1$, the k-pruning with the minimum k-median objective function value (under the best choice of centers) can be computed in polynomial time.

[Hint: dynamic programming.]

Exercise 28

In lecture we mentioned that the single-link++ algorithm is not guaranteed to recover the optimal solution in γ-stable k-median instances with $\gamma < 3$. Explain why the following example demonstrates this: consider points 1,2,3,4,5 arranged in a line, with $d(1,2) = 2$, $d(2,3) = 1$, $d(3,4) = 2 - \epsilon$, and $d(4,5) = 1$.

[Hint: don't forget to argue that the instance is $(3 - \epsilon)$-stable.]

Exercise 29

For a metric space (X, d) and choice S of k centers, the *k-means* objective function is

$$\sum_{x \in X} \left(\min_{c \in S} d(c, x) \right)^2 .$$

What changes to the algorithm and/or analysis of single-link++ are needed to guarantee the recovery of the optimal solution in γ-stable k-means instances, where $\gamma \geq 3$?

Exercise 30

Prove that a k-median instance that is $(c, 0)$-stable in the sense of Lecture #6 ("approximation stability") is also c-stable in the sense of Lecture #7 ("perturbation stability"). Prove that the converse is false.

[Remark: the same arguments apply to the "ϵ versions" of the definitions. In this sense, perturbation stability is a strictly weaker restriction on instances than approximation stability, so positive results are harder to come by.]

Lecture 8 Exercises

Exercise 31

Let $(\hat{x}, \hat{d})$ be an optimal solution to the linear programming relaxation of the s-t minimum cut problem described in lecture. Let $B(r) = \{v \in V : d(v) \leq r\}$ denote the ball of radius r around s. Prove that if $r \in (0, 1)$ is chosen uniformly at random, then for every edge $e \in E$, the probability that e has exactly one endpoint in $B(r)$ is exactly $\hat{x}_e$.

[Hint: prove that the optimality of $(\hat{x}, \hat{d})$ implies that $\hat{x}_e = |\hat{d}(u) - \hat{d}(v)|$ for every edge $e = (u, v)$.]

Exercise 32

Recall the linear programming relaxation of the multiway cut problem discussed in lecture. Since the multiway cut problem is NP-hard and linear programs can be solved in polynomial time, we know that the optimal linear programming solution will not generally correspond to a multiway cut (assuming $P \neq NP$). The point of this exercise is to give an explicit demonstration of this fact.

Consider a graph with 3 terminals t_1, t_2, t_3 and three other vertices u, v, w. The vertices u, v, w are connected in a triangle of unit-cost edges. Terminals are connected to two of these vertices (e.g., t_i to u, v, t_2 to v, w, and t_3 to u, v) via edges of cost 2. Prove that every multiway cut has total cost at least 8, while the optimal objective function value of the linear programming relaxation is strictly less.

Exercise 33

Consider a γ-stable multiway cut instance (with $\gamma > 4$), with C^* the optimal multiway cut, cutting the edges F^*. As in lecture, for another multiway cut C, cutting the edges F, define

$$\Delta_4(C) = \sum_{e \in F \setminus F^*} c_e - 4 \sum_{e \in F^* \setminus F} c_e .$$

Prove that $\Delta_4(C) > 0$ for every $C \neq C^*$.

Exercise 34

Let $(\hat{x}, \hat{d})$ be an optimal solution to the linear programming relaxation of the multiway cut problem described in lecture, and recall the randomized rounding algorithm described in the lecture. 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 $i \in \{1, 2, \ldots, \}$ equals

$$\frac{\max\{d_u^i, d_v^i\}}{\sum_{j=1}^k \max\{d_u^j, d_v^j\}}.$$

Exercise 35

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

$$\frac{\max\{d_u^i, d_v^i\} - \min\{d_u^i, d_v^i\}}{\sum_{j=1}^k \max\{d_u^j, d_v^j\}}.$$

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 $\frac{1}{2}(1 - \hat{x}_e)$.

Exercise 36

Say that an algorithm A achieves *certifiable exact recovery* for a set I of instances if: (i) for every instance in I, 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 I (i.e., can correctly declare this at its termination). Does the single-link++ algorithm for 3-stable k-median instances achieve certifiable exact recovery? What about the linear programming solution for 4-stable multiway cut instances?

Problems

Problem 11

(20 points) Recall in Lecture #6 we discussed the Balcan-Blum-Gupta (BBG) k-median algorithm. Recall also our assumption that every cluster in the optimal solution has at least $2b+2$ points, where $b = \epsilon n(1 + \frac{5}{\alpha})$. (This is in addition to the assumption that the instance is $(1 + \alpha, \epsilon)$-stable.)

Our goal in lecture was to recover an ϵ-close clustering, rather than to optimize the k-median objective per se. Show that, nevertheless, the BBG algorithm gives an $O(1)$-approximation to the k-median objective in $(1 + \alpha, \epsilon)$-stable instances that satisfy the large clusters assumption. (The constant can depend on α.)

[Hint: recall that after Step 3 of the BBG algorithm, all but the non-well-separated points are correctly classified. (We mostly skipped this in class, so read about it in the lecture notes.) Use the Triangle Inequality to charge the cost of incorrectly classified points to the cost of the optimal solution.]

Problem 12

(15 points) Recall that in the *Vertex Cover* problem, you are given an undirected graph $G = (V, E)$ where each vertex has a nonnegative weight w_v. The goal is to compute the subset S of V of minimum total weight with the property that every edge has at least one of its endpoints in S.

Call a Vertex Cover instance γ-*stable* if its optimal solution S^* remains optimal even after each vertex v is scaled by an arbitrary factor $\sigma_v \in [1, \gamma]$. Prove that in Δ-stable Vertex Cover instances, the optimal solution can be recovered in polynomial time. (Here Δ denotes the maximum degree of the graph.)

Problem 13

This problem considers γ-stable instances of metric Max Cut (similar to our notions in Lecture #8). The input is a complete undirected graph $G = (V, E, w)$ with nonnegative edge weights w that satisfy the Triangle Inequality. Recall that such an instance is $(1 + \epsilon)$-stable if the maximum cut stays the same no matter how you multiply the edge weights by factors in $[1, 1 + \epsilon]$. (Even for such scalings that yield weights that violate the Triangle Inequality.) Assume that $\epsilon > 0$ is small but constant. Let (A, B) denote the maximum cut.

(a) (3 points) Prove that for every vertex v, the total weight of its incident edges that cross (A, B) is at least $(1 + \epsilon)$ times that of those that do not.

(b) (4 points) Without loss of generality, we can scale all the edge weights so that they sum to n^2. Define the weight of a vertex as the sum of the weights of its incident edges. Prove that every vertex has weight at least n.

(c) (5 points) Construct a (polynomial-size, non-metric) graph $G' = (V', E')$ as follows. For every edge $v \in V$ with weight w_v, add $\lfloor w_v \rfloor$ vertices to V' ("copies of v"). For each $u, v \in V$, add an edge to E' between each copy of u and of v, with weight $w_{uv}/\lfloor w_u \rfloor \lfloor w_v \rfloor$. Prove that the property in part (a) continues to hold for the graph G' (perhaps with a different constant ϵ'). Prove that a maximum cut of G can be recovered from one of G'.

(d) (6 points) Prove that in G', for every vertex u, the maximum weight of an edge incident to u is at most a constant factor times the average weight of an edge incident to u.

(e) (7 points) Give a polynomial-time approximation scheme (PTAS) for $(1 + \epsilon)$-stable metric Max Cut instances.

[Hint: use both random sampling and brute-force search.]

Problem 14

(15 points) Prove that there is a constant $\gamma > 1$ such that it is NP-hard to recover the optimal solution in γ-stable k-median instances. Prove this for the largest value of γ that you can.

[Hint: think about examples where every distance $d(x, y)$ is either 1 or 2.]

Problem 15

This problem considers the "perturbation stability" notion of Lecture #7.

(a) (8 points) Prove that for sufficiently large constants γ, the optimal clustering $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of every γ-stable k-median instance satisfies the following property:

(*) for pair $p, p' \in \mathcal{C}_i$ of points from the same cluster and every point $q \in \mathcal{C}_j$ in a different cluster, $d(p, p') < d(p, q)$. That is, every point is closer to all points of its own cluster than any points of any other clusters.

(b) (7 points) Prove that k-median instances that satisfy property (*) can be solved by a simpler algorithm than single-link++, specifically an algorithm that makes a single pass through the points in an arbitrary order. Can you get your algorithm to run in $O(n \log k)$ time (using suitable data structures)?

Problem 16

The point of this problem is to modify the algorithm from Lecture #7 to achieve exact recovery for an even wider class of instances. Throughout this problem, we consider a γ-stable k-median instance with $\gamma > 1+\sqrt{2}$, optimal clusters $\mathcal{C}_1^*, \ldots, \mathcal{C}_k^*$, and optimal centers $c_1, \ldots, c_k$.

(a) (3 points) Prove that for every cluster $\mathcal{C}_i^*$, $p \in \mathcal{C}_i^*$, and $q \notin \mathcal{C}_i^*$, $d(p, c_i) < d(p, q)$. That is, in the optimal solution, every point is closer to its center than to any other point in any other cluster.

[Hint: this is similar to arguments from lecture.]

(b) (3 points) Prove that for every cluster $\mathcal{C}_i^*$, $p \in \mathcal{C}_i^*$, and $q \notin \mathcal{C}_i^*$, $d(p, c_i) < d(q, c_i)$. That is, in the optimal solution, all points of $\mathcal{C}_i^*$ are closer to c_i than all points outside $\mathcal{C}_i^*$.

[Hint: this is similar to arguments from lecture.]

(c) (0 points) Define $r_i = \max_{p \in \mathcal{C}_i^*} d(c_i, p)$. Observe that parts (a) and (b) imply that the set of points that lie inside a ball of radius r_i around c_i is precisely $\mathcal{C}_i^*$, and each point in this set is closer to the center c_i than to any point outside the set.

(d) (0 points) For a subset S of points, let $r(S)$ denote the smallest radius r such that there exists a center $c \in S$ such that: (i) $d(c, p) \leq r(S)$ for every $p \in S$; and (ii) for every $p \in S$ and $q \notin S$, $d(p, c_i) < d(p, q)$. [Note $r(S)$ is finite, at most $\max_{p,q \in X} d(p, q)$.]

(e) (5 points) Consider a partition of X into $S_1, \ldots, S_m$ such that each set S_i is either (i) a subset of an optimal cluster $\mathcal{C}_i^*$ or (ii) the union of one or more optimal clusters. Prove that if S_a is a subset of an optimal cluster $\mathcal{C}_i^*$, then there is another subset $S_b \subseteq \mathcal{C}_i^*$ such that $r(S_a \cup S_b) \leq r_i$.

(f) (7 points) Continuing the previous part, suppose that S_a is a strict subset of $\mathcal{C}_i$ and S_b is not. Prove that $r(S_a \cup S_b) > r_i$.

[Hint: use stability and triangle inequality arguments, as in (a) and (b).]

(g) (7 points) Using (d) and (e), give a polynomial-time algorithm that achieves exact recovery in every γ-stable k-median instance with $\gamma > 1 + \sqrt{2}$. Explain why your algorithm is correct.

[Hint: stick with single-link++ algorithm, except use a different component-merging criterion than in Kruskal's algorithm.]