Khoury College of Computer Sciences
Northeastern University
CS 7870: Seminar in Theoretical Computer Science

Fall 2023
19 September 2023
Scribe: Rajmohan Rajaraman

Lecture 4 Outline:

- Perturbation-stability
- Single-link clustering
- A refinement of single-link clustering is optimal for certain perturbation-stable instances

1 Introduction

In the previous lecture, we considered a local search algorithm for k-median and showed that it can be used to obtain a $(5+\varepsilon)$-approximate algorithm in polynomial time, for any constant $\varepsilon > 0$. An exercise in PS1 asks you to prove, via an example, that this is essentially tight, so a better than 5-approximation is not achievable using the particular local search approach.

The k-median problem for metric spaces is not only NP-hard, but is also APX-hard, meaning that there is an absolute constant $c > 1$ such that there is no polynomial time c-approximation for the problem, unless P = NP. The best hardness of approximation that is currently known is a factor of $1 + 2/e$ [GK99]. While this provides some comfort for using the local search algorithm, there is still a gap in our understanding. Several questions come to mind.

- Are there variants of local search that do better than 5-approximate?
  - Yes! In fact, a natural variant where instead of swapping a pair of medians, we consider p swaps, for an integer constant $p \geq 1$, is $(3 + 2/p + \varepsilon)$-approximate for any $\varepsilon > 0$. And this bound is tight. Note that by choosing p to be a sufficiently large constant, we can achieve a $(3+\varepsilon)$-approximation for any $\varepsilon > 0$.
- What is the best approximation algorithm achievable in polynomial time by *any* algorithm?
  - As one can imagine, this is an active area of research. The current best approximation ratio for metric k-median is $2.675 + \varepsilon$, for arbitrary $\varepsilon > 0$ [BPR$^+$17]. The algorithm is based on a careful rounding of a linear programming relaxation, using an approach referred to as dependent rounding.
- The approximation algorithms that have best-known approximation ratios tend to be sophisticated and are often not implemented in practice. On the other hand, local search and other simple approaches are often used, and appear to work well. Are there special kinds of metric spaces for which local search or other simple clustering approaches can be shown have much better approximation ratios than can be shown in the worst-case?
  - Recent results have shown that local search yields a $(1+\varepsilon)$-approximation for metric k-median weighted planar graphs and for k-means in metrics of bounded doubling dimension (thus, including Euclidean metrics of bounded dimension).

- An alternative approach to considering special cases has been that of defining notions of stability. We will consider one such notion—perturbation-stability—in today's lecture.

2 Perturbation-stability

In a bid to derive more informative analyses of algorithms, beyond the worst-case, several notions of stability and perturbation have been introduced. We will study a natural and general model, called *smoothed analysis*, later in the course. Here, we consider a model specific to metric spaces. Informally, a perturbation-stable instance is one where an optimal solution remains optimal even if the instance is perturbed a bit.

Definition 1. For any parameter $\gamma \geq 1$, a γ-perturbation of a metric space (V, d) is a metric space (V, d') such that for all (u, v), we have

$$d(u, v)/\gamma \leq d'(u, v) \leq d(u, v).$$

Definition 2. For a parameter $\gamma \geq 1$, a clustering instance (V, d) is γ-perturbation stable if there is a set $\mathcal{C}^*$ of clusters such that $\mathcal{C}^*$ is the unique optimal clustering solution for every γ-perturbation (V, d') of (V, d).

A few remarks are in order. First, note that the γ-perturbation stable condition requires that the optimal clustering solution for the given k-median instance is unique. And the same solution continues to be optimal even for γ-perturbations of the underlying metric. It does not require that the centers associated with all the clusters also stay the same; the centers and the associated costs may change with perturbations, the latter especially since the metric is being perturbed.

One way to think about γ-perturbation stable instances is the following: we are given a metric space for which there is a single unique clustering that remains uniquely optimal even when we perturb distances up to γ factors; can we find such a clustering efficiently?

Before proceeding with this question, it is natural to ask whether γ-perturbation stable instances even exist! With some exploration, you will find that situations where the "right" clustering seems clear (points within a cluster are close to one another while points in different clusters are not that close) are indeed γ-perturbation stable. It is, however, not easy to tell whether an instance is γ-perturbation stable (for one, the definition concerns optimal k-median solutions, which are hard to find in the first place). Nevertheless, such a notion of perturbation stability has proved useful for various combinatorial optimization problems and has been used to understand and evaluate practical algorithms. An interesting exercise would be to study the perturbation stability of real-world instances, experimentally or analytically.

3 Single-link clustering

A popular algorithm for clustering is hierarchical in nature. For a given set of n points, we start with n clusters, one for each point. We then identify two clusters that may be closer to one another (via the presence of a single suitable link between them) than with others, and merge them into one cluster and repeat this until we achieve a desired objective. An astute student of algorithms may recognize that this appears very similar to the well-known Kruskal's algorithm for minimum spanning trees. And indeed it is.

The simplest variety of single-link clustering is the following. For the given metric space (V, d), construct a complete graph G in which the weight of edge (u, v) is $d(u, v)$. Run Kruskal's algorithm until we have exactly k connected components (i.e., we have added $n - k$ edges to the MST). Return the k clusters.

How well does single-link clustering do with respect to the k-median objective? Not surprisingly, not well in the worst case, since the algorithm is not even bothering with the actual objective.

Consider a set of $k + 1$ regions in a metric space, spread over a line, with a distance of 1 units between consecutive regions, except that regions k and $k + 1$ are separated by a distance of 2 units. Suppose each of the regions 1 through k has a large number, say $L \gg 1$, of points within it very close to one another (at distance 0, for simplicity), while region $k + 1$ has one point. The optimal clustering for this instance is for the clusters to be regions 1 through $k - 1$ and one cluster merging the regions k and $k + 1$, with a total cost of 1. The above clustering approach will create k clusters, one combining say regions 1 and 2 and then one for each of regions 3 through $k + 1$, with a total cost of L. Since L can be arbitrarily large, the MST based algorithm will have an arbitrarily large approximation ratio. This holds despite the fact the instance is highly stable to perturbations.

We consider the following refined single-link clustering algorithm, due to Angelidakis, Makarychev, and Makarychev [AMM17], which we call *Kruskal+DP*.

1. Run Kruskal's algorithm to calculate a minimum spanning tree T.
2. Determine the set of $k - 1$ edges to remove from T so that the resulting set of k clusters has minimum k-median cost in the metric (V, d). This step can be accomplished in polynomial time using a suitable dynamic program.

4 Analysis for perturbation stable instances

Let us begin by identifying some conditions under which the Kruskal+DP algorithm is optimal. We say that a cluster C is split in tree T if there exist two vertices u and v in C such that the unique path in T between u and v passes through a vertex not in C.

Lemma 1. Kruskal+DP returns an optimal clustering if and only if there exists an optimal clustering such that none of the optimal clusters is split in the MST.

Proof. The proof is straightforward. Since any clustering returned by Kruskal+DP is not split in the MST, the clustering returned by the algorithm is optimal only if there exists an optimal clustering such that none of the optimal clusters is split in the MST. For the other direction, suppose there is an optimal clustering that is not split in the MST. Since Kruskal+DP returns a clustering that is optimal among all clusterings that are not split in the MST, the clustering it returns has optimal cost. □

The remainder of the proof establishes that for γ-perturbation stable instances, an optimal clustering is not split in the MST. We will establish this through a series of lemmas. First, we show that in any γ-perturbation stable instance, every point is much closer to the center of its cluster than to any other center. In the remainder of this section, we use $\mathcal{C} = \{C_i : 1 \leq i \leq k\}$ to denote the unique optimal clustering for (V, d) and let c_i denote the center of cluster C_i.

Lemma 2. If the instance is γ-perturbation stable, then for all u, if u is in the cluster of center c_i, then $d(u, c_j) > \gamma d(u, c_i)$ for $j \neq i$.

Proof. The proof is by contradiction. Suppose there exists a vertex u in cluster C_i such that $d(u, c_j) \leq \gamma d(u, c_i)$ for some center $j \neq i$. We will argue that the instance is not γ-perturbation stable. Let G denote the complete graph with egde (x, y) having weight $d(x, y)$ for $x, y \in V$. Let G' denote the graph that is identical to G except that the edge (u, c_j) has weight $d(u, c_i)$. Consider the metric (V, d'), where d' is the shortest path distance metric in G'.

By construction, (V, d') is a metric. We now argue that it is a γ-perturbation of (V, d). Consider any $x, y \in V$. If the shortest path between x and y in G' does not use the edge (u, c_j), then $d'(x, y) = d(x, y)$. Otherwise, we have

$$d'(x, y) = d'(x, u) + d'(u, c_j) + d'(c_j, y) = d(x, u) + d'(u, c_j) + d(c_j, y).$$

Since $d'(u, c_j) > d(u, c_j)/\gamma$, we have $d'(c_j, y) > d(x, y)/\gamma$. It is also easy to see that $d'(x, y) \leq d(x, y)$.

We now establish that in (V, d'), the clustering $\mathcal{C}$ is not optimal. Clearly, u is closer to c_j than to c_i. Therefore, if $\{c_\ell\}$ continue to be the centers used in an optimal clustering, u will be in a different cluster than in $\mathcal{C}$. It is possible, however, even though the centers may be different, the clusters remain the same. We now claim that the distances within the clusters C_i and C_j under d' are identical to those under d, implying that if the clusters C_i and C_j are in the optimal solution, then their centers are c_i and c_j, respectively. But that would lead to a contradiction since u will move to the cluster with center c_j, yielding a new solution with lesser cost than $\mathcal{C}$.

Let x and y be two vertices in C_i. If the shortest path between x and y under d' does not use the edge (u, c_j), then this path is the same as under d, so $d(x, y) = d'(x, y)$. Otherwise, the shortest path under d' is one of the following two forms:

$$x \rightsquigarrow u \rightarrow c_j \rightsquigarrow y \qquad x \rightsquigarrow c_j \rightarrow u \rightsquigarrow y.$$

Suppose it is of the first form (the argument for the second form is similar). Consider the path $x \rightarrow u \rightarrow c_i \rightarrow y$. We know that $d(x, u) = d'(x, u)$ and $d(u, c_i) = d'(u, c_j)$. Furthermore, $d'(c_j, y) = d(c_j, y) \geq d(c_i, y)$. Therefore, we have $d(x, y) \leq d'(x, y)$, and indeed they are equal.

Now, consider x and y in C_j. Again, if the shortest path between x and y under d' does not use the edge (u, c_j), then this path is the same as under d, so $d(x, y) = d'(x, y)$. Otherwise, the shortest path under d' is one of the following two forms:

$$x \rightsquigarrow u \rightarrow c_j \rightsquigarrow y \qquad x \rightsquigarrow c_j \rightarrow u \rightsquigarrow y.$$

Suppose it is of the first form (the argument for the second form is similar). We argue that $d(x, c_j) \leq d'(x, u) + d'(u, c_j)$, which would yield $d(x, y) \leq d'(x, y)$ as desired. If not, then $d(x, c_i) \leq d(x, u) + d(u, c_i) = d'(x, u) + d'(u, c_j) < d(x, c_j)$, contradicting the fact that x is in cluster C_j in metric (V, d). □

Lemma 3. Let u be in cluster with center c_i in the optimal clustering. For any γ-perturbation instance with $\gamma \geq 2$, the distance between u and c_i is strictly less than that between u and v for any v in a different cluster than u.

Proof. Let u, v, and c_i be as given. Then, using Lemma 2 and triangle inequality, we have

$$\begin{array}{rcl} d(u,v) & \geq & d(u,c_j) - d(v,c_j) \\ & \geq & 2d(u,c_i) - d(v,c_j) \\ & \geq & 2d(u,c_i) - \dfrac{d(v,c_i)}{/}2 \\ & \geq & \dfrac{3d(u,c_i)}{2} - \dfrac{d(u,v)}{2}, \end{array}$$

implying that $d(u,v) \geq d(u,c_i)$. □

Lemma 4. For any γ-perturbation instance (V,d), the optimal clustering $\mathcal{C}^*$ is not split in the MST.

Proof. The proof is by contradiction. Suppose cluster C_i is split in the MST. This means that there exists a point $x \in C_i$ such that the path from x to c_i in the MST goes through a vertex y not in C_i. Let z denote the last vertex in the subpath from x to y that is in C_i, and let $w \notin C_i$ be the next vertex after z along this subpath. By Lemma 3, $d(z,w) > d(z,c_i)$; if we add the edge (z,c_i) and remove (z,w), we obtain a new spanning tree of lesser weight, contradicting the optimality of Kruskal's algorithm. This completes the proof of the lemma. □

Theorem 1. Kruskal+DP is optimal for γ-perturbation stable instances with $\gamma \geq 2$.

Proof. Follows directly from Lemmas 1 and 4. □

References

[AMM17] Haris Angelidakis, Konstantin Makarychev, and Yury Makarychev. Algorithms for stable and perturbation-resilient problems. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 438–451, New York, NY, USA, 2017. Association for Computing Machinery.

[BPR$^+$17] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Trans. Algorithms*, 13(2), mar 2017.

[GK99] Sudipto Guha and Samir Khuller. Greedy strikes back. *J. Algorithms*, 31(1):228–248, apr 1999.