TTIC 31290: Machine Learning for Algorithm Design (Fall 2025)

Avrim Blum and Dravyansh Sharma

Lecture 4: 10/09/25 Lecturer: Avrim Blum

Outline for today

- Structural approaches to Beyond-Worst-Case-Analysis
- Perturbation-Resilience
- Case study: k-clustering
- Certified algorithms
- Approximation-Stability

1 Structural approaches to BWCA

In the last class, we moved away from worst-case analysis by adding *randomness* to the instances. In structural approaches to BWCA, we instead posit a property or structure that we might expect or want instances to satisfy, and then design algorithms for instances having that property or structure. In some cases, we can then argue that these algorithms will also have interesting guarantees even on worst-case instances.

2 Perturbation-Resilience

For $\gamma > 1$, we'll say that instance I' is a γ-perturbation of instance I if you can produce I' by changing each value in I by a multiplicative factor between 1 and γ. An instance I of some optimization problem is said to be "γ-Perturbation-resilient" if all γ-perturbations I' of I have the same optimal solution (and no other). E.g., think of a clustering problem like k-center clustering. Note that the perturbation could change the cost of the solution, but it shouldn't change which solution is optimal. Our goal, then, will be to design algorithms that we can prove will find optimal solutions on γ-Perturbation-resilient instances for γ as small as possible.

A stronger goal, which is achievable for some problems, is to have a "γ-certified algorithm". This algorithm, given any instance I, produces a γ-perturbation I' of I and a solution s' such that s' is optimal for I'. So, this is like "I couldn't solve the exact problem you gave me, but here is an optimal solution to this similar problem." If instance I happens to be γ-perturbation-resilient, then s' will be optimal for I too.

3 Case study: Center-based k-clustering

Given n points $x_1, ..., x_n$ in a metric space X, the k-median problem asks to find k centers $c_1, ..., c_k$ such that if you create k clusters by assigning each x_i to its nearest c_j, the total sum (or average) distance between points and their assigned centers is minimized. The k-means problem asks the same except to minimize the sum of squared distances. The k-center problem asks the same except to minimize the maximum distance. We will call all of these objectives *center-based* because they all involve finding k centers and then assigning points to their nearest center to minimize some score that is a nondecreasing function of the distances between points and their associated centers. In the discussion below, we will assume that the centers c_j must belong to $\{x_1, ..., x_n\}$. We will also call the clusters $C_1, ..., C_k$ where C_j is the set of points x_i assigned to center c_j.

Note 1: In this formulation, one shouldn't think of k as a fixed constant like 3, since there's a trivial algorithm that runs in time polynomial in n^k. (Can you see it?) Instead, one should think of k as a parameter and we want to be polynomial in n and k.

Note 2: In this formulation, γ-perturbation resilience means that the optimal clusters $C_1, ..., C_k$ should remain the same in any γ-perturbation, but it is allowed for the centers c_j of those clusters to move around.

We say that an instance $I = \{x_1, ..., x_n\}$ satisfies γ-center-proximity if for all j, for all $x_i \in C_j$, for all $j' \neq j$, $dist(x_i, c_{j'}) > \gamma \cdot dist(x_i, c_j)$.

The γ-center-proximity property is an implication of γ-perturbation-resilience for any center-based clustering objective. We can see this by arguing the contrapositive: if you didn't have γ-center-proximity, you could blow up all pairwise distances in C_j by a factor of γ (which maintains c_j being an optimal center for C_j) but then the optimal clustering could assign x_i to $c_{j'}$ rather than to c_j.

So, what we're going to do is focus on this implication of perturbation-resilience for most of our analysis, and then get back to the specific objective function only in the last part. I'm going to describe a result for $\gamma = 3$ from Awasthi et al [ABS12]. This was improved to $\gamma = 1 + \sqrt{2}$ by Balcan and Liang [BL16] and then to $\gamma = 2$ by Angelidakis et al [AMM17]. But the factor of 3 analysis is simplest.

4 Efficient k-clustering of instances satisfying 3-center-proximity

First, let's establish some useful implications of center-proximity.

Lemma 1. *If an instance satisfies γ-center-proximity, then for any two clusters j and j', any point $x_i \in C_j$ and any $x_{i'} \in C_{j'}$, we have $dist(x_i, x_{i'}) > (\gamma - 1) dist(x_i, c_j)$.*

Proof. Suppose $dist(x_{i'}, c_{j'}) \geq dist(x_i, c_j)$. Then by triangle inequality and center proximity we get $dist(x_i, x_{i'}) \geq dist(x_{i'}, c_j) - dist(x_i, c_j) > \gamma \cdot dist(x_{i'}, c_{j'}) - dist(x_i, c_j) \geq (\gamma - 1) \cdot dist(x_i, c_j)$. Alternatively, if $dist(x_i, c_j) > dist(x_{i'}, c_{j'})$ then we get $dist(x_i, x_{i'}) \geq dist(x_i, c_{j'}) - dist(x_{i'}, c_{j'}) > \gamma \cdot dist(x_i, c_j) - dist(x_{i'}, c_{j'}) \geq (\gamma - 1) \cdot dist(x_i, c_j)$. □

So, for $\gamma = 3$ we have that every point x_i in cluster C_j is more than twice as far away from all points in a different cluster than it is to its own center. We now show that this can help us with a useful implication.

Lemma 2. *Suppose the instance satisfies 3-center-proximity. Then for any cluster C_j and any partition of C_j into two nonempty sets $A, B = C_j \setminus A$, and any other cluster $C_{j'}$, the minimum distance between A and B is strictly less than the minimum distance between C_j and $C_{j'}$.*

Before proving Lemma 2, let's see why this is useful. Consider running the "single-linkage" clustering algorithm. This algorithm starts will each node in its own cluster and then repeatedly merges the two clusters whose minimum distance is smallest (minimum distance means the distance between the closest points in the two clusters). Suppose we run this algorithm until there is only one cluster left. Notice that for any target cluster C_j, this algorithm will complete C_j into a cluster before any subset of C_j is connected to anything else. This means that if you examine the binary tree on clusters produced by this algorithm, the true target clusters $C_1, ..., C_k$ will all be nodes in this tree. They might not all be at the same level, but together they will form a pruning of this tree. Now what we can do is run Dynamic Programming on this tree, using the original objective function (like k-median or k-means). Starting at the leaves and working upward, for every $k' \leq k$, we compute the cheapest clustering using at most k' clusters we can get from the subtree rooted at that node. For any given node C, since we've already computed these results for the two children of C, the answer for some k' for C is just the minimum out of all $k'' \leq k'$ of the cost for using k'' clusters on the left and $k' - k''$ clusters on the right.

OK, so now let's complete the analysis by proving Lemma 2.

Proof of Lemma 2. Consider a cluster C_j and some partition of C_j into two nonempty sets A and B. Say the minimum distance between C_j and some other cluster $C_{j'}$ is given by some $x \in C_j$ and $x' \in C_{j'}$. We want to show that $dist(x, x')$ is strictly larger than the minimum distance between A and B. There are two cases. The easy case is that x and center c_j are on different sides of the A, B partition. Then Lemma 1 immediately gives us that $dist(x, x') > 2 \cdot dist(x, c_j) \geq mindist(A, B)$. So, suppose x and c_j are both on the same side, say A.

Let y be a point in B. By 3-center-proximity and triangle inequality and Lemma 1 (in order) we have

$$3 \cdot dist(y, c_j) < dist(y, c_{j'}) \leq dist(y, x) + dist(x, x') + dist(x', c_{j'}) < dist(y, x) + \frac{3}{2} \cdot dist(x, x').$$

Also, by triangle inequality and Lemma 1 (in order) we have

$$dist(y, x) \leq dist(y, c_j) + dist(c_j, x) < dist(y, c_j) + \frac{1}{2} \cdot dist(x, x').$$

Putting these together we have

$$3 \cdot dist(y, c_j) < dist(y, c_j) + 2 \cdot dist(x, x').$$

So, this gives us $dist(y, c_j) < dist(x, x')$ as desired. □

5 Certified Algorithms

As mentioned above, a stronger goal that is achievable for some problems is to have a "γ-certified" algorithm. This algorithm, given any instance I, produces a γ-perturbation I' of I and a solution

s' such that s' is optimal for I'. If instance I happens to be γ-perturbation-resilient, then s' will be optimal for I too. The specific bounds achievable for certified algorithms will generally be worse than for non-certified, though of course the meaning of the bound is stronger.

Here is a result from [BW17,MM20b], for the k-median objective. I will just give the algorithm and the result, but not the proof. See [MM20a] for a detailed proof. For this algorithm, think of ρ as a small positive integer. The running time will be exponential in ρ.

ρ-local search

1. Start with an arbitrary subset S of k points in $\{x_1, ..., x_n\}$ and calculate $cost(S)$, the k-median cost of using S as the k cluster centers.

2. If we can reduce $cost(S)$ by swapping up to ρ of the points in S with other points in $\{x_1, ..., x_n\}$, then do so. Repeat until at a local optimum.

Theorem 1. *The ρ-local search algorithm is $(3 + O(1/\rho))$-certified.*

6 Approximation-Stability

Suppose I gave you a dataset to cluster, and I said that this dataset has the property that it's enough to get a 1.25 approximation to the k-median problem in order to cluster all but some ϵ fraction of the points correctly (say there is some ground-truth, like clustering images of people by who is in them). You might then read up that it's NP-hard to approximate k-median to better than $1 + 1/e$ which is greater than 1.25, and say that this is not a very useful property. But, it turns out that under this assumption, you can efficiently find a clustering that clusters all but an $O(\epsilon)$ fraction of points correctly, without solving that approximation problem. In other words, the property gives you structure that allows you to cluster directly. (And if you also have all clusters being large, then the property actually does allow you to get a 1.25-approximation quickly; in other words, those instances aren't the hard ones). And you can replace 1.25 with any constant greater than 1 (but as the constant gets smaller, the hidden term in the $O()$ gets larger). This is the notion of approximation stability. We won't discuss approximation-stability further but for more information, see [BBG09, B20].

References and Additional Resources

- [AMM17] H. Angelidakis, K. Makarychev, and Y. Makarychev, "Algorithms for Stable and Perturbation-Resilient Problems," STOC 2017. `https://home.ttic.edu/~yury/papers/bwca.pdf`

- [ABS12] P. Awasthi, Pranjal, A. Blum, and O. Sheffet, "Center-Based Clustering Under Perturbation Stability," Information Processing Letters, 112(1-2), 49–54, 2012. `https://home.ttic.edu/~avrim/Papers/clustering-IPL.pdf`

- [BBG90] M.-F. Balcan, A. Blum, and A Gupta, “Approximate Clustering Without the Approximation,” SODA 2009. Full version in JACM 2013. `https://dl.acm.org/doi/pdf/10.1145/2450142.2450144`

- [BL16] M.-F. Balcan and Y. Liang, “Clustering Under Perturbation Resilience,” SIAM Journal on Computing, 45(1), 102–155, 2016.

- [BW17] M.-F. Balcan and C. White, “Clustering Under Local Stability: Bridging the Gap Between Worst-Case and Beyond Worst-Case Analysis.” 2017. `https://arxiv.org/abs/1705.07157`

- [B20] A. Blum, “Approximation Stability and Proxy Objectives”, Chapter 6 in Beyond Worst-Case Analysis. `https://home.ttic.edu/~avrim/Papers/Chapter_6.pdf`

- [MM20a] K. Makarychev and Y. Makarychev, “Perturbation Resilience,” Chapter 5 in Beyond Worst-Case Analysis. `https://home.ttic.edu/~yury/papers/bwca.pdf`

- [MM20b] K. Makarychev and Y. Makarychev, “Certified Algorithms: Worst-Case Analysis and Beyond,” ITCS 2020. `https://par.nsf.gov/servlets/purl/10185956`