Beyond Worst Case Analysis **Fall 2018**

Lecture 9: Balcan-Blum-Gupta Stability

Scribe: James Hulett 11/26/2018

9.1 Introduction

In the last meeting, we discussed one definition of stability, due to Bilu and Linial, which considered adversarial perturbations of the input. Today, we discuss a somewhat different definition, due to [BBG]:

Definition 9.1. *Let $c > 1$ and $\epsilon > 0$ be fixed, and let Φ be the objective function of some minimization problem. We say that an instance of that minimization problem is (c, ϵ)-stable if all possible solutions $\mathcal{S}$ satisfying $\Phi(\mathcal{S}) \leq c \cdot \Phi(\mathcal{S}^*)$ also satisfy dist$(\mathcal{S}, \mathcal{S}^*) \leq \epsilon$, where $\mathcal{S}^*$ is the optimal solution.*

In order for this definition to make sense, we need some sense of the "distance" between two solutions to an instance of a problem. For this meeting, we will apply this definition to k-clustering problems, so our notion of distance will simply be the fraction of points classified differently by the two clusterings. However, we do have to be a bit careful here. Suppose that we tried to define the distance between two clusters $\mathcal{C} = (C_1, ..., C_k)$ and $\tilde{\mathcal{C}} = (\tilde{C}_1, ..., \tilde{C}_k)$ as

$$\text{dist}(\mathcal{C}, \tilde{\mathcal{C}}) = \frac{1}{n} \sum_{i=1}^{k} |C_i - \tilde{C}_i|$$

If $\tilde{\mathcal{C}}$ simply permuted the names of the clusters in $\mathcal{C}$, this definition would say that the distance between these two clusterings is 1, even though intuitively we would like to say that they are the same clustering, and so should have a distance of zero.[1] To avoid this, when we define the distance between two clusterings, we allow for arbitrary renamings of the clusters, and take whichever naming gives us the lowest distance:

$$\text{dist}(\mathcal{C}, \tilde{\mathcal{C}}) = \min_{\sigma} \frac{1}{n} \sum_{i=1}^{k} |C_i - \tilde{C}_{\sigma(i)}$$

Here, the minimum is taken over all permutations of $[k]$.

[1] Indeed, with this naïve definition, it would be impossible to get (c, ϵ) stability for any $\epsilon < 1$ and $k > 1$, as we could always permute the names on the optimal clustering in order to get a clustering of the same value but at a distance of one away.

9.2 Problem Statement

As we did with Bilu-Linial in the last meeting, we will apply this new definition of stability to the k-medians problem. In order to make the definition of the algorithm easier, we are going to make some assumptions about our inputs, listed below:

Assumption 9.2. The input is $(1+\alpha, \epsilon)$-stable for some $\alpha, \epsilon > 0$.

Assumption 9.3. The values of α and ϵ in assumption 9.2 are known.

Assumption 9.4. The value $\text{OPT} := \Phi(\mathcal{C}^*)$ is known.

Assumption 9.5. Every cluster in the optimal clustering $\mathcal{C}^*$ has size at least $2\epsilon n(1+\frac{5}{\alpha})+2$.

Assumption 9.2 is, of course, key to our analysis, and cannot easily be removed. The given notes show a way to remove both assumptions 9.3 and 9.4, but actually does still use the values of α and ϵ even in the improved algorithm. Thus, here we will only show that you can remove assumption 9.4. Finally, [BBG] show that you can remove assumption 9.5, though it may lead to more points being misclassified. For the sake of time, we will not go in depth into how to remove this last assumption.

9.3 The Algorithm

9.3.1 The Good, The Bad, and The Ugly

Before stating the algorithm, it is helpful to classify some of the points we are asked to cluster based on how easy we expect it to be to put them in the right cluster. We call a point "good" if it is *close* to the median of its optimal cluster (at a distance of no more that $\text{OPT} \cdot \frac{\alpha}{5\epsilon n}$) and is *well separated* from the next closest center (the difference between the distance to its optimal median and the next closest one is greater than $\text{OPT} \cdot \frac{\alpha}{\epsilon n}$); if a point is not good, we call it "bad".

Lemma 9.6. *The number of bad points is at most* $b := \epsilon n(1+\frac{5}{\alpha})$.

Proof. First, we note that no more than $\frac{5\epsilon n}{\alpha}$ points can fail to be close to their respective centers. If this were not the case, the contribution to the objective value of the optimum solution for just those points would be larger than OPT, meaning that the overall objective value would have to be larger than OPT, as no point can be at a negative distance from its median. This is a contradiction, since OPT is defined to be the objective value of the optimal clustering.

Next, we claim that fewer than ϵn points will fail to be well separated. If this were not the case, we could move ϵn of those points to their next closest cluster. By assumption 9.5, we move fewer than half the points in each cluster, meaning that the σ used in calculating the distance between the optimal clustering and this new clustering will just be the identity. Hence, the distance between this new clustering and the optimum would be ϵ. But since change in the objective value would be at most $\text{OPT} \cdot \alpha$, meaning that this clustering would violate assumption 9.2. □

9.3.2 Phase 1

[BBG]'s algorithm proceeds in two stages. The first gets us an initial clustering that is guaranteed to correctly cluster all of the good points. By assumptions 9.3 and 9.4, we can define the value $\tau := \frac{2}{5} \cdot \text{OPT} \cdot \frac{\alpha}{\epsilon n}$ and create a graph $G = (X, E)$ where the vertices are the n points to be clustered and $(x, y) \in E$ if and only if $d(x, y) \leq \tau$. This graph is a bit too connected to be useful to us, so we trim it back a bit, creating a new graph $H = (X, F)$ where $F \subseteq E$ and we keep edge (x, y) if and only if x and y have at least b common neighbors in G.[2] To get our clustering, we simply take the k largest connected components in H as our clusters, and assign any points not in any of those components arbitrarily.

Lemma 9.7. *If x and y are good points in the same optimal cluster C_i^*, $(x, y) \in E$.*

Proof. Since all good points are close to their respective centers, we have by the triangle inequality that

$$d(x, y) \leq d(x, c_i) + d(c_i, y) \leq \left(\text{OPT} \cdot \frac{\alpha}{5\epsilon n}\right) + \left(\text{OPT} \cdot \frac{\alpha}{5\epsilon n}\right) = \tau$$

□

This lemma tells us that the good points in a given cluster will all form a clique in G. By assumption 9.5, we have that the number of points in C_i^* is at least $2b + 2$, so the number of good points is at least $b + 2$. Hence, the clique of good points will survive the creation of H, since every pair of points in that clique shares at least the other b clique vertices as neighbors. This leads nicely into our next lemma:

Lemma 9.8. *Each of the k largest connected components in H contains all the good points from exactly one optimal cluster.*

Proof. From above, we know that the clique of good points from a given cluster will survive in H, so we know that all the good points in a given cluster will end up in the same connected component. We now have to show that two different clusters can't end up in the same connected component. In order to do this, we note that no bad point can be neighbors in G with good points from two different clusters. If this was the case, the triangle inequality would tell us that the distance between those two good points would be at most $2\tau = \frac{4}{5} \cdot \text{OPT} \cdot \frac{\alpha}{\epsilon n}$. But then those points are good, meaning they are within $\frac{1}{5} \cdot \text{OPT} \cdot \frac{\alpha}{\epsilon n}$ of their cluster center, so an additional application of the triangle inequality gives us that the each good point is within $\text{OPT} \cdot \frac{\alpha}{\epsilon n}$ of the other's cluster center; this contradicts the well-separated property of the good points. This tells us that along any path in G between good points in two different clusters, there must be two adjacent vertices whose only shared neighbors are bad points. Since this leaves them with at most $b - 2 < b$ potential shared neighbors, we know that this edge does not get included in H, and so every path between two clusters gets cut.

Now that we know every cluster of good points ends up in its own connected component, we just need to show that the k largest connected components will all be of this form. But we know that there are at most b bad points, so the largest cluster not of this form can have at most b vertices in it. By contrast, assumption 9.5 tells us that there are at least $b + 2$ good points in every cluster, so the connected components of this form will all have size at least $b + 2$. Hence, the k "cluster" connected components will all be bigger than any other components, meaning that they are the k largest ones. □

This lemma tells us that our algorithm will correctly cluster all of the good points, meaning that at worst it misclassifies the $O(\epsilon n(1 + \frac{1}{\alpha}))$ bad points.

[2] This step relies on assumption 9.3, since b is defined based on α and ϵ. This is the part that the given notes overlook when they claim later on that we can eliminate that assumption.

9.3.3 Phase 2

The second phase of the algorithm is a rather clever "clean-up" phase which is guaranteed to correctly classify all the points which are well separated, even if they aren't close. The idea is that the property of being well separated will mean that a point is generally closer to the good points in its correct cluster than it is to those in any other cluster. In order to leverage this idea, we compute the median distance between each point and the points in a given cluster from phase 1, then assign a point to whichever cluster had the smallest median distance.

Lemma 9.9. *If x is a well separated point that belongs to the optimal cluster C_i^*, cluster i will have the smallest median distance to it.*

Proof. By the triangle inequality, we know that the distance between x and any good point y in cluster C_i^* is at most $d(x, c_i) + d(c_i, y)$, which is at most $d(x, c_i) + \frac{\tau}{2}$, as y is close to its center. But there are at least $b + 1$ good points in the phase 1 cluster corresponding to C_i^* at at most b other (bad) points, meaning that the median distance will also be at most $d(x, c_i) + \frac{\tau}{2}$. If we instead look at the distance to some point y in some other cluster C_j^*, we have that $d(x, y) \geq d(x, c_j) - d(c_j, y)$. The former term is at least $\frac{5}{2}\tau + d(x, c_i)$ since x is well separated, while the later term is at most $\frac{\tau}{2}$ since y is close. Hence, in this case we have that $d(x, y) \geq d(x, c_i) + 2\tau$. Again, more than half the points in the cluster corresponding to C_j^* are good points from that cluster, meaning that the median distance is at least $d(x, c_i) + 2\tau$. □

This lemma tells us that after this clean-up step, the only points that might still be misclassified are those which are not well separated. From the proof of Lemma 9.6, we see that there are at most ϵn such points.

9.3.4 Removing Assumptions

As promised, we now will briefly discuss how to remove assumption 9.4. While we used the value of OPT many times in our analysis, the only place we used it in the actual algorithm was to determine the value of τ when building our graph G. However, we note that the exact value of τ is not important–the only thing that actually matters is which edges get included in G. Thus, we need not try every possible value of τ, merely the ones that actually cause the graph G to change. Since there are only $\binom{n}{2}$ possible edges that could appear in G, we only need to try $\binom{n}{2}$ different values for τ. If we simply take the best clustering we find from any choice of τ, our result will be at least as good as if we knew the exact value of τ and ran the algorithm as initially described.

Finally, we give a brief word about assumption 9.5. As it turns out, this assumption is unnecessary for phase 1 of the algorithm, though we don't have time to discuss exactly how to remove it from the analysis. However, [BBG] were not able to come up with a way to remove this assumption from the second phase of the analysis. Thus, if we remove this assumption, we end up with potentially as many as $O(\epsilon n(1 + \frac{1}{\alpha}))$ points misclassified, as opposed to the at most ϵn we can get with that assumption.

References

[BBG] M.-F. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1068 - 1077, 2009.