8 Clustering

8.1 Some Clustering Examples

Clustering comes up in many contexts. For example, one might want to cluster journal articles into clusters of articles on related topics. In doing this, one first represents a document by a vector. This can be done using the vector space model introduced in Chapter 2. Each document is represented as a vector with one component for each term giving the frequency of the term in the document. Alternatively, a document may be represented by a vector whose components correspond to documents in the collection and the i^{th} vector's j^{th} component is a 0 or 1 depending on whether the i^{th} document referenced the j^{th} document. Once one has represented the documents as vectors, the problem becomes one of clustering vectors.

Another context where clustering is important is the study of the evolution and growth of communities in social networks. Here one constructs a graph where nodes represent individuals and there is an edge from one node to another if the person corresponding to the first node sent an email or instant message to the person corresponding to the second node. A community is defined as a set of nodes where the frequency of messages within the set is higher than what one would expect if the set of nodes in the community were a random set. Clustering partitions the set of nodes of the graph into sets of nodes where the sets consist of nodes that send more messages to one another than one would expect by chance. Note that clustering generally asks for a strict partition into subsets although in reality in this case for instance, a node may well belong to several communities.

In these clustering problems, one defines either a similarity measure between pairs of objects or a distance measure (a notion of dissimilarity). One measure of similarity between two vectors $\mathbf{a}$ and $\mathbf{b}$ is the cosine of the angle between them:

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T\mathbf{b}}{|\mathbf{a}|\,|\mathbf{b}|}.$$

To get a distance measure, subtract the cosine similarity from one.

$$\text{dist}(\mathbf{a}, \mathbf{b}) = 1 - \cos(\mathbf{a}, \mathbf{b})$$

Another distance measure is the Euclidean distance. There is an obvious relationship between cosine similarity and Euclidean distance. If $\mathbf{a}$ and $\mathbf{b}$ are unit vectors, then

$$|\mathbf{a} - \mathbf{b}|^2 = (\mathbf{a} - \mathbf{b})^T(\mathbf{a} - \mathbf{b}) = |\mathbf{a}|^2 + |\mathbf{b}|^2 - 2\mathbf{a}^T\mathbf{b} = 2\left(1 - \cos\left(\mathbf{a}, \mathbf{b}\right)\right).$$

In determining the distance function to use, it is useful to know something about the origin of the data. In problems where we have to cluster nodes of a graph, we may represent each node as a a vector, namely, as the row of the adjacency matrix corresponding to the node. One notion of dissimilarity here is the square of the Euclidean distance. For 0-1

vectors, this measure is just the number of "uncommon" 1's, whereas, the dot product is the number of common 1's. In many situations one has a stochastic model of how the data was generated. An example is customer behavior. Suppose there are d products and n customers. A reasonable assumption is that each customer generates from a probability distribution, the basket of goods he or she buys. A basket specifies the amount of each good bought. One hypothesis is that there are only k types of customers, $k << n$. Each customer type is characterized by a probability density used by all customers of that type to generate their baskets of goods. The densities may all be Gaussians with different centers and covariance matrices. We are not given the probability densities, only the basket bought by each customer, which is observable. Our task is to cluster the customers into the k types. We may identify the customer with his or her basket which is a vector. One way to formulate the problem mathematically is via a clustering criterion which we then optimize. Some potential criteria are to partition the customers into k clusters so as to minimize

1. the sum of distances between all pairs of customers in the same cluster,

2. the sum of distances of all customers to their "cluster center" (any point in space may be designated as the cluster center), or

3. minimize the sum of squared distances to the cluster center.

The last criterion is called the *k-means* criterion and is widely used. A variant called the *k-median* criterion minimizes the sum of distances (not squared) to the cluster center**.** Another possibility, called the *k-center* criterion, is to minimize the maximum distance of any point to its cluster center.

The chosen criterion can affect the results. To illustrate, suppose we have data generated according to a equal weight mixture of k spherical Gaussian densities centered at $\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, \ldots, \boldsymbol{\mu_k}$, each with variance 1 in every direction. Then the density of the mixture is

$$F(\mathbf{x}) = \text{Prob}[\mathbf{x}] = \frac{1}{k}\frac{1}{(2\pi)^{d/2}}\sum_{t=1}^{k} e^{-|\mathbf{x}-\boldsymbol{\mu_t}|^2}.$$

Denote by $\boldsymbol{\mu}(\mathbf{x})$ the center nearest to $\mathbf{x}$. Since the exponential function falls off fast, we have the approximation

$$F(\mathbf{x}) \approx \frac{1}{k}\frac{1}{(2\pi)^{d/2}} e^{-|\mathbf{x}-\boldsymbol{\mu}(\mathbf{x})|^2}.$$

So, given a sample of points $\mathbf{x^{(1)}}, \mathbf{x^{(2)}}, \ldots \mathbf{x^{(n)}}$ drawn according to the mixture, the likelyhood of a particular $\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, \ldots, \boldsymbol{\mu_k}$, namely, the (posterior) probability of generating the sample if $\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, \ldots, \boldsymbol{\mu_k}$ were in fact the centers, is approximately

$$\frac{1}{k^n}\frac{1}{(2\pi)^{nd/2}}\prod_{i=1}^{n} e^{-|\mathbf{x^{(i)}}-\boldsymbol{\mu}(\mathbf{x^{(i)}})|^2} = ce^{-\sum_{i=1}^{n}|\mathbf{x^{(i)}}-\boldsymbol{\mu}(\mathbf{x^{(i)}})|^2}.$$

So, minimizing the sum of squared distances to cluster centers finds the maximum likelyhood $\boldsymbol{\mu_1, \mu_2, \ldots, \mu_k}$ and this suggests the criterion : sum of distance squared to the cluster centers.

On the other hand, if the generating process had an exponential probability distribution, with the probability law

$$Prob\left[(x_1, x_2, \ldots, x_d)\right] = \frac{1}{2}\prod_{i=1}^{d} e^{-|x_i - \mu_i|} = \frac{1}{2}e^{-\sum_{i=1}^{d}|x_i-\mu_i|} = \frac{1}{2}e^{-|\mathbf{x}-\boldsymbol{\mu}|_1},$$

one would use the L_1 norm (not the L_2 or the square of the L_1) since the probability density decreases as the L_1 distance from the center. The intuition here is that the distance used to cluster data should be related to the actual distribution of the data.

The choice of whether to use a distance measure and cluster together points that are close or use a similarity measure and cluster together points with high similarity and what particular distance or similarity measure to use can be crucial to the application. However, there is not much theory on these choices; they are determined by empirical domain-specific knowledge. One general observation is worth making. Using distance squared instead of distance, "favors" outliers more since the square function magnifies large values, which means a small number of outliers may make a clustering look bad. On the other hand, distance squared has some mathematical advantages; see for example Corollary 8.3 which asserts that with the distance squared criterion, the centroid is the correct cluster center. The widely used k-means criterion is based on sum of squared distances.

In the formulations we have discussed so far, we have one number (eg. sum of distances squared to the cluster center) as the "measure of goodness" of a clustering and we try to optimize that number (to find the best clustering according to the measure). This approach does not always yield desired results, since often, it is hard to optimize exactly (most clustering problems are NP-hard). Often, there are polynomial time algorithms to find an approximately optimal solution. But such a solution may be far from the optimal (desired) clustering. We will in section (8.4) how to formalize some realistic conditions under which an approximate optimal solution indeed gives us a desired clustering as well. But first we see some simple algorithms for getting a good clustering according to some natural measures.

8.2 A Simple Greedy Algorithm for k-clustering

There are many algorithms for clustering high-dimensional data. We start with a simple one. Suppose we use the k-center criterion. The k-center criterion partitions the points into k clusters so as to minimize the maximum distance of any point to its cluster center. Call the maximum distance of any point to its cluster center the radius of the clustering. There is a k-clustering of radius r if and only if there are k spheres, each of radius r which together cover all the points. Below, we give a simple algorithm to find k spheres covering a set of points. The lemma following shows that this algorithm needs to

use a radius that is "off by a factor of at most two" from the optimal k-center solution.

The Greedy k-clustering Algorithm

> Pick any data point to be the first cluster center. At time t, for $t = 2, 3, \ldots, k$, pick any data point that is not within distance r of an existing cluster center; make it the t^{th} cluster center.

Lemma 8.1 *If there is a k-clustering of radius $\frac{r}{2}$, then the above algorithm finds a k-clustering with radius at most r.*

Proof: Suppose for contradiction that the algorithm using radius r fails to find a k-clustering. This means that after the algorithm chooses k centers, there is still at least one data point that is not in any sphere of radius r around a picked center. This is the only possible mode of failure. But then there are $k + 1$ data points, with each pair more than distance r apart. Clearly, no two such points can belong to the same cluster in any k-clustering of radius $\frac{r}{2}$ contradicting the hypothesis. ∎

There are in general two variations of the clustering problem for each of the criteria. We could require that each cluster center be a data point or allow a cluster center to be any point in space. If we require each cluster center to be a data point, the problem can be solved in time $\binom{n}{k}$ times a polynomial in the length of the data. First, exhaustively enumerate all sets of k data points as the possible sets of k cluster centers, then associate each point to its nearest center and select the best clustering. No such naive enumeration procedure is available when cluster centers can be any point in space. But, for the k-means problem, Corollary 8.3 shows that once we have identified the data points that belong to a cluster, the best choice of cluster center is the centroid. Note that the centroid might not be a data point.

8.3 Lloyd's Algorithm for k-means Clustering

In k-means criterion, a set $A = \{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}\}$ of n points in d-dimensions is partitioned into k-clusters, $S_1, S_2, \ldots, S_k$, so as to minimize the sum of squared distances of each point to its cluster center. That is, A is partitioned into clusters, $S_1, S_2, \ldots, S_k$, and a center is assigned to each cluster so as to minimize

$$d\left(S_1, S_2, \ldots, S_k\right) = \sum_{j=1}^{k} \sum_{\mathbf{a_i} \in S_j} \left(\mathbf{c_j} - \mathbf{a_i}\right)^2$$

where $\mathbf{c_j}$ is the center of cluster j.

Suppose we have already determined the clustering or the partitioning into $S_1, S_2, \ldots, S_k$. What are the best centers for the clusters? The following lemma shows that the answer is the centroids, the coordinatewise means, of the clusters.

Lemma 8.2 *Let $\{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}\}$ be a set of points. The sum of the squared distances of the $\mathbf{a_i}$ to any point $\mathbf{x}$ equals the sum of the squared distances to the centroid plus the number of points times the squared distance from the point $\mathbf{x}$ to the centroid. That is,*

$$\sum_i |\mathbf{a_i} - \mathbf{x}|^2 = \sum_i |\mathbf{a_i} - \mathbf{c}|^2 + n\,|\mathbf{c} - \mathbf{x}|^2$$

where $\mathbf{c}$ is the centroid of the set of points.

Proof:

$$\begin{aligned}\sum_i |\mathbf{a_i} - \mathbf{x}|^2 &= \sum_i |\mathbf{a_i} - \mathbf{c} + \mathbf{c} - \mathbf{x}|^2 \\ &= \sum_i |\mathbf{a_i} - \mathbf{c}|^2 + 2(\mathbf{c} - \mathbf{x}) \cdot \sum_i (\mathbf{a_i} - \mathbf{c}) + n\,|\mathbf{c} - \mathbf{x}|^2\end{aligned}$$

Since $\mathbf{c}$ is the centroid, $\sum_i (\mathbf{a_i} - \mathbf{c}) = 0$. Thus, $\sum_i |\mathbf{a_i} - \mathbf{x}|^2 = \sum_i |\mathbf{a_i} - \mathbf{c}|^2 + n\,\|\mathbf{c} - \mathbf{x}\|^2$

■

A corollary of Lemma 8.2 is that the centroid minimizes the sum of squared distances since the second term, $n\,\|\mathbf{c} - \mathbf{x}\|^2$, is always positive.

Corollary 8.3 *Let $\{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}\}$ be a set of points. The sum of squared distances of the $\mathbf{a_i}$ to a point $\mathbf{x}$ is minimized when $\mathbf{x}$ is the centroid, namely $\mathbf{x} = \frac{1}{n}\sum_i \mathbf{a_i}$.*

Another expression for the sum of squared distances of a set of n points to their centroid is the sum of all pairwise distances squared divided by n. First, a simple notational issue. For a set of points $\{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}\}$, $\sum_{i=1}^{n}\sum_{j=i+1}^{n} |\mathbf{a_i} - \mathbf{a_j}|^2$ counts the quantity $|\mathbf{a_i} - \mathbf{a_j}|^2$ once for each ordered pair (i, j), $j > i$. However, $\sum_{i,j} |\mathbf{a_i} - \mathbf{a_j}|^2$ counts each $|\mathbf{a_i} - \mathbf{a_j}|^2$ twice, so the later sum is twice the first sum.

Lemma 8.4 *Let $\{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}\}$ be a set of points. The sum of the squared distances between all pairs of points equals the number of points times the sum of the squared distances of the points to the centroid of the points. That is, $\sum_i \sum_{j>i} |\mathbf{a_i} - \mathbf{a_j}|^2 = n \sum_i |\mathbf{a_i} - \mathbf{c}|^2$ where $\mathbf{c}$ is the centroid of the set of points.*

Proof: Lemma 8.2 states that for every $\mathbf{x}$,

$$\sum_i |\mathbf{a_i} - \mathbf{x}|^2 = \sum_i |\mathbf{a_i} - \mathbf{c}|^2 + n\,|\mathbf{c} - \mathbf{x}|^2 .$$

Letting $\mathbf{x}$ range over all $\mathbf{a_j}$ and summing the n equations yields

$$\sum_{i,j} |\mathbf{a_i} - \mathbf{a_j}|^2 = n \sum_i |\mathbf{a_i} - \mathbf{c}|^2 + n \sum_j |\mathbf{c} - \mathbf{a_j}|^2$$
$$= 2n \sum_i |\mathbf{a_i} - \mathbf{c}|^2.$$

Observing that

$$\sum_{i,j} |\mathbf{a_i} - \mathbf{a_j}|^2 = 2 \sum_i \sum_{j>i} |\mathbf{a_i} - \mathbf{a_j}|^2$$

yields the result that

$$\sum_i \sum_{j>i} |\mathbf{a_i} - \mathbf{a_j}|^2 = n \sum_i |\mathbf{a_i} - \mathbf{c}|^2 .$$

■

The k-means clustering algorithm

A natural algorithm for k-means clustering is given below. There are two unspecified aspects of the algorithm. One is the set of starting centers and the other is the stopping condition.

k-means algorithm

Start with k centers.
Cluster each point with the center nearest to it.
Find the centroid of each cluster and replace the set of old centers with the centroids.
Repeat the above two steps until the centers converge (according to some criterion).

The k-means algorithm always converges but possibly to a local minimum. To show convergence, we argue that the cost of the clustering, the sum of the squares of the distances of each point to its cluster center, always improves. Each iteration consists of two steps. First, consider the step which finds the centroid of each cluster and replaces the old centers with the new centers. By Corollary 8.3, this step improves the sum of internal cluster distances squared. The other step reclusters by assigning each point to its nearest cluster center, which also improves the internal cluster distances.

8.4 Meaningful Clustering via Singular Value Decomposition

Optimizing a criterion such as $k-$means is often not an end in itself. It is a means to finding a good (meaningful) clustering. How do we define a meaningful clustering? Here is a possible answer: an optimal clustering is meaningful if it is unique, in the sense that any other nearly optimal clustering agrees with it on most data points. We will formalize this below. But the bad news is that we will soon see that this is too much to ask for and many common data sets do not admit such a clustering. Luckily though, the discussion

will lead us to a weaker requirement which has the twin properties of being met by many data sets as well as admitting an efficient (SVD-based) algorithm to find the clustering. We start with some notation.

We denote by n the number of (data) points to be clustered; they are listed as the rows A_i of a $n \times d$ matrix A. A clustering (partition) of the data points is represented by a *matrix of cluster centers* C which is also a $n \times d$ matrix; the i th row of C is the center of the cluster that A_i belongs to. So C has only k distinct rows. We refer to A as *the data* and to C as *the clustering.*

Definition: The *cost* of the clustering C is the sum of distances squared to the cluster centers; so we have

$$\text{cost}(A, C) = ||A - C||_F^2.$$

The *mean squared distance* (MSD) of a clustering C is just $\text{cost}(A, C)/n$.

We will say two clusterings of A *differ in s points* if s is the minimum number of data points to be reassigned to get from one clustering to the other. [Note: a clustering is specified by just a partition; the cluster centers are just the centroids of data points in a cluster.] Here is the first attempt at defining when a clustering is meaningful:

A $k-$ clustering C is *meaningful* if every $k-$clustering C' which differs from C in at least ϵn points has cost at least $(1 + \Omega(\epsilon))$ cost(C). [ϵ is a small constant.]

Claim 8.1 *If C is meaningful under this definition, and each of its clusters has $\Omega(n)$ data points in it, then for any two cluster centers μ, μ' of C,*

$$|\mu - \mu'|^2 \geq MSD(C) :$$

Proof: We will prove the claim by showing that if two cluster centers in C are too close, then we may move ϵn points from cluster to the other without increasing the cost by more than a factor of $(1 + O(\epsilon))$, thus contradicting the assumption that C is meaningful.

Let T be the cluster with cluster center μ. Project each data point $A_i \in T$, onto the line through μ, μ' and let d_i be the distance of the projected point to μ. Let T_1 be the subset of T whose projections land on the μ' side of μ and let $T_2 = T \setminus T_1$ be the points whose projections land on the other side. Since μ is the centroid of T, we have $\sum_{i \in T_1} d_i = \sum_{i \in T_2} d_i$. Since each $A_i \in T$ is closer to μ than to μ', for $i \in T_1$, we have $d_i \leq |\mu - \mu'|/2$ and so $\sum_{i \in T_1} d_i \leq |T||\mu - \mu'|/2$; hence also, $\sum_{i \in T_2} d_i \leq |T||\mu - \mu'|/2$. So,

$$\sum_{i \in T} d_i \leq |T||\mu - \mu'|.$$

Now from the assumption that $|T| \in \Omega(n)$, we have $|T| \geq 2\epsilon n$. So, the ϵn th smallest d_i is at most $\frac{|T|}{|T| - \epsilon n}|\mu - \mu'| \leq 2|\mu - \mu'|$. We can now get a new clustering C' as follows: move the ϵn A_i in T with the smallest d_i to the cluster with μ' as center. Recompute centers. The recomputatation of the centers can only reduce the cost (as we saw in Corollary (8.3)). What about the move? The move can only add cost (distance squared) in the direction of the line joining μ, μ' and this extra cost is at most $4\epsilon n|\mu - \mu'|^2$. By the assumption that C is meaningful (under the proposed definition above), we must thus have $|\mu - \mu'|^2 \geq$ MSD(C) as claimed. ∎

But we will now see that the condition that $|\mu - \mu'|^2 \geq \text{MSD}(C)$ is too strong for some common data sets. Consider two spherical Gaussians in $d-$ space, each with variance 1 in each direction. Clearly if we have data generated from a (equal weight) mixture of the two, the "correct" 2-clustering one would seek is to split them into the Gaussians they were generated from with the actual centers of the Gaussians (or a very close point) as the cluster centers. But the MSD of this clustering is approximately d. So, by the Claim, for C to be meaningful, the centers must be $\Omega(d)$ apart. It is easy to see however that if the separation is $\Omega(\sqrt{\ln n})$, the clusters are distinct : the projection of each Gaussian to the line joining their centers is a 1 dimensional Gaussian of variance 1, so the probability that any data point lies more than a tenth of the way (in the projection) to the wrong center is at most $O(1/n^2)$, so by union bound, all data points have their distance to the wrong center at least 10 times the distance to the correct center, provided, we measure distances only in the projection. Since the variance is 1 in **each direction**, the Mean Squared Distance to the cluster center in **each direction** is only $O(1)$. So, in this example, it would make more sense to require a inter-center separation of the maximum mean squared distance in any one direction to the cluster center. We will be able to achieve this in general.

Definition: Let A, C be the data and cluster centers matrices respectively. The *mean squared distance in a direction* (denoted $\text{MSDD}(A, C)$) is the maximum over all unit length vectors $\mathbf{v}$ of the mean squared distance of data points to their cluster centers in the direction $\mathbf{v}$, namely,

$$\text{MSDD}(A, C) = \frac{1}{n}\text{Max}_{\mathbf{v}:|\mathbf{v}|=1}|(A - C)\mathbf{v}|^2 = \frac{1}{n}||A - C||_2^2,$$

where, we have used the basic definition of the largest singular value to get the last expression.

Theorem 8.5 *Suppose there exists a $k-$clustering C of data points A with (i) $\Omega(n)$ data points per cluster and (ii) distance at least $\Omega(\sqrt{MSDD(A, C)})$ between any two cluster centers. Then any clustering returned by the following simple algorithm (which we henceforth call the SVD clustering of A) differs from C is at most ϵn points (here, the hidden constants in Ω depend on ϵ.)*

Find the SVD of A. Project data points to the space of the top k (right) singular vectors of A. Return a $2-$ approximate[8] $k-$means clustering in the projection.

Remark: Note that if we did the approximate optimization in the whole space (without projecting), we will not succeed in general. In the example of the two spherical Gaussians above, for the correct clustering C, $\text{MSDD}(A, C) = O(1)$. But if the inter-center separation is just $O(1)$, then ϵn points of the first Gaussian may be put into the second at an added cost (where cost is distance squared in the whole space) of only $O(n)$ as we argued, whereas, the cost of clustering C is $O(nd)$.

[8] A 2-approximate clustering is one which has cost at most twice the optimal cost.

Remark: The Theorem also implies a sort of uniqueness of the clustering C, namely, that any other clustering satisfying both (i) and (ii) differs from C in at most $2\epsilon n$ points, as seen from the following: The Theorem applies to the other clustering as well, since it satisfies the hypothesis. So, it also differs from the SVD clustering in at most ϵn points. Thus, C and the other clustering cannot differ in more than $2\epsilon n$ points.

The proof of the Theorem will use the following two Lemmas, which illustrate the power of SVD. The first Lemma says that for any clustering C with $\Omega(n)$ points per cluster, the SVD clustering described in the Theorem finds cluster centers fairly close to the cluster centers in C, where, close is measured in terms of MSDD(A, C). The argument will be that one candidate clustering in the SVD projection is to just use the same centers as C (projected to the space) and if the SVD clustering does not have a cluster center close to a particular cluster of C, it ends up paying too high a penalty compared to this candidate clustering to be 2-optimal.

Lemma 8.6 *Suppose A is the data matrix and C a clustering with $\Omega(n)$ points in each cluster.* [9] *Suppose C' is the SVD clustering (described in the Theorem) of A. Then for every cluster center μ of C, there is a cluster center of C' within distance $O(\sqrt{MSDD(A, C)})$.*

Proof: Let α =MSDD(A, C). Let T be the set of data points in a cluster of C and suppose for contradiction that centroid μ of T has no cluster center of C' at distance $O(\sqrt{k\alpha})$. Let $\bar{A}_i$ denote projection of A_i onto the SVD subspace; so the SVD clustering actually clusters points $\bar{A}_i$. Recall the notation that C'_i is the cluster center of C' closest to $\bar{A}_i$. The cost of a data point $A_i \in T$ in the SVD solution is

$$|\bar{A}_i - C'_i|^2 = |(\mu - C'_i) - (\mu - \bar{A}_i)|^2 \geq \frac{1}{2}|\mu - C'_i|^2 - |\mu - \bar{A}_i|^2 \geq \Omega(\alpha) - |\mu - \bar{A}_i|^2,$$

where, we have used $|a - b|^2 \geq \frac{1}{2}|a|^2 - |b|^2$ for any two vectors a and b.
Adding over all points in T, the cost of C' is at least $\Omega(n\alpha) - ||\bar{A} - C||_F^2$. Now, one way to cluster the points $\bar{A}_i$ is to just use the same cluster centers as C; the cost of this clustering is $||\bar{A} - C||_F^2$. So the optimal clustering of points $\bar{A}_i$ costs at most $||\bar{A} - C||_F^2$ and since the algorithm finds a 2-approximate clustering, the cost of the SVD clustering is at most $2||\bar{A} - C||_F^2$. So we get

$$2||\bar{A} - C||_F^2 \geq \Omega(n\alpha) - ||\bar{A} - C||_F^2 \quad \Longrightarrow \quad ||\bar{A} - C||_F^2 \geq \Omega(n\alpha).$$

We will prove that $\left\|\bar{A} - C\right\|_F^2 \leq 5k\alpha n$ in Lemma 8.7. By (i), we have $k \in O(1)$, so $||\bar{A} - C||_F^2 = O(\alpha)$. So, we get a contradiction (with suitable choice of constants under the Ω.) ∎

Note that in the Lemma below, the main inequality has Frobenius norm on the left hand side, but only operator norm on the right hand side. This makes it stronger than having either Frobenius norm on both sides or operator norm on both sides.

[9] Note that C is not assumed to satisfy condition (ii) of the Theorem.

Lemma 8.7 *Suppose A is an $n \times d$ matrix and suppose C is an $n \times d$ rank k matrix. Let $\bar{A}$ be the best rank k approximation to A found by SVD. Then, $||\bar{A} - C||_F^2 \leq 5k||A - C||_2^2$.*

Proof: Let $\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_k}$ be the top k singular vectors of A. Extend the set of the top k singular vectors to an orthonormal basis $\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_p}$ of the vector space spanned by the rows of $\bar{A}$ and C. Note that $p \leq 2k$ since $\bar{A}$ is spanned by $\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_k}$ and C is of rank at most k. Then,

$$||\bar{A} - C||_F^2 = \sum_{i=1}^{k} |(\bar{A} - C)\mathbf{u_i}|^2 + \sum_{i=k+1}^{p} |(\bar{A} - C)\mathbf{u_i}|^2.$$

Since $\{\mathbf{u_i} | \, 1 \leq i \leq k\}$ are the top k singular vectors of A and since $\bar{A}$ is the rank k approximation to A, for $1 \leq i \leq k$, $A\mathbf{u_i} = \bar{A}\mathbf{u_i}$ and thus $\left|\left(\bar{A} - C\right)\mathbf{u_i}\right|^2 = |(A - C)\,\mathbf{u_i}|^2$. For $i > k$, $\bar{A}\mathbf{u_i} = 0$, thus $\left|\left(\bar{A} - C\right)\mathbf{u_i}\right|^2 = |C\mathbf{u_i}|^2$. From this it follows that

$$\begin{aligned}\left\|\bar{A} - C\right\|_F^2 &= \sum_{i=1}^{k} |(A - C)\mathbf{u_i}|^2 + \sum_{i=k+1}^{p} |C\mathbf{u_i}|^2 \\ &\leq k||A - C||_2^2 + \sum_{i=k+1}^{p} |A\mathbf{u_i} + (C - A)\mathbf{u_i}|^2\end{aligned}$$

Using $|a + b|^2 \leq 2|a|^2 + 2|b|^2$

$$\begin{aligned}\left\|\bar{A} - C\right\|_F^2 &\leq k||A - C||_2^2 + 2\sum_{i=k+1}^{p} |A\mathbf{u_i}|^2 + 2\sum_{i=k+1}^{p} |(C - A)\mathbf{u_i}|^2 \\ &\leq k||A - C||_2^2 + 2\,(p - k - 1)\,\sigma_{k+1}^2(A) + 2\,(p - k - 1)\,||A - C||_2^2\end{aligned}$$

Using $p \leq 2k$ implies $k > p - k - 1$

$$\left\|\bar{A} - C\right\|_F^2 \leq k||A - C||_2^2 + 2k\sigma_{k+1}^2(A) + 2k||A - C||_2^2. \tag{8.1}$$

As we saw in Chapter 4, for any rank k matrix B, $||A - B||_2 \geq \sigma_{k+1}(A)$ and so $\sigma_{k+1}(A) \leq ||A - C||_2$ and plugging this in, we get the Lemma. ∎

Proof: (of Theorem (8.5)) Let $\beta = \sqrt{\text{MSDD}(A, C)}$. We will use Lemma (8.6). To every cluster center μ of C, there is a cluster center ν of the SVD clustering within distance $O(\beta)$. Also, since, C satisfies (ii) of the Theorem, the mapping from μ to ν is 1-1. C and the SVD clustering differ on a data point A_i if the following happens: A_i belongs to cluster center μ of C whose closest cluster center in C' is ν, but $\bar{A}_i$ (the projection of A_i