Machine Learning for Data Science (CS 4786)

Lecture 12-13: Clustering, Single-link algorithm, K-means Algorithm

The text in black outlines main ideas to retain from the lecture. The text in blue give a deeper understanding of how we "derive" or get to the algorithm or method. The text in red are mathematical details for those who are interested. But is not crucial for understanding the basic workings of the method.

1 Representing Data: Feature Vectors

For most of this course we shall assume that data provided to us as feature vectors. That is, each data point is represented as a d-dimensional vector (d numbers). Specifically we assume data points are provided as $\mathbf{x}_1, \ldots, \mathbf{x}_n$ where each $\mathbf{x}_t \in \mathbb{R}^d$ is a d-dimensional vector. The process of converting given data representation into feature vectors that capture relevant information about the data provided is known as feature extraction. Below are two very naive examples of feature vectors extracted from images and text.

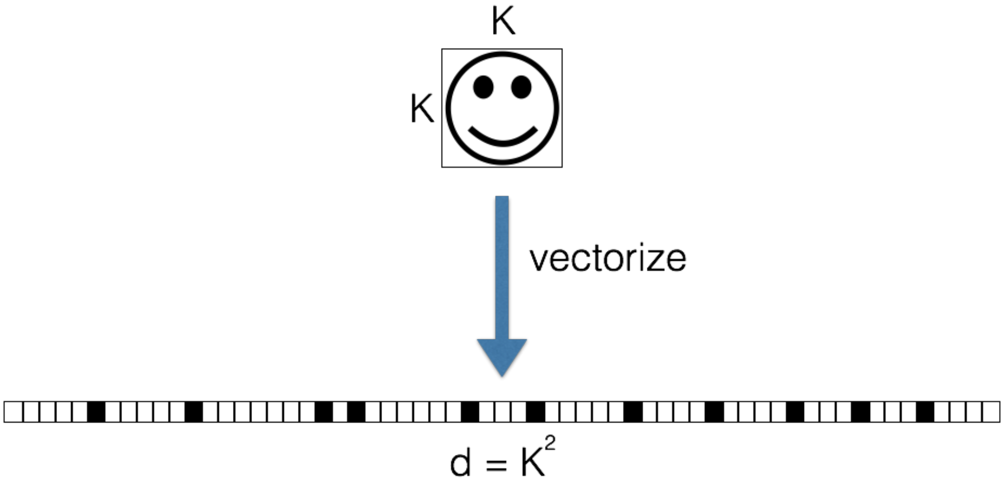
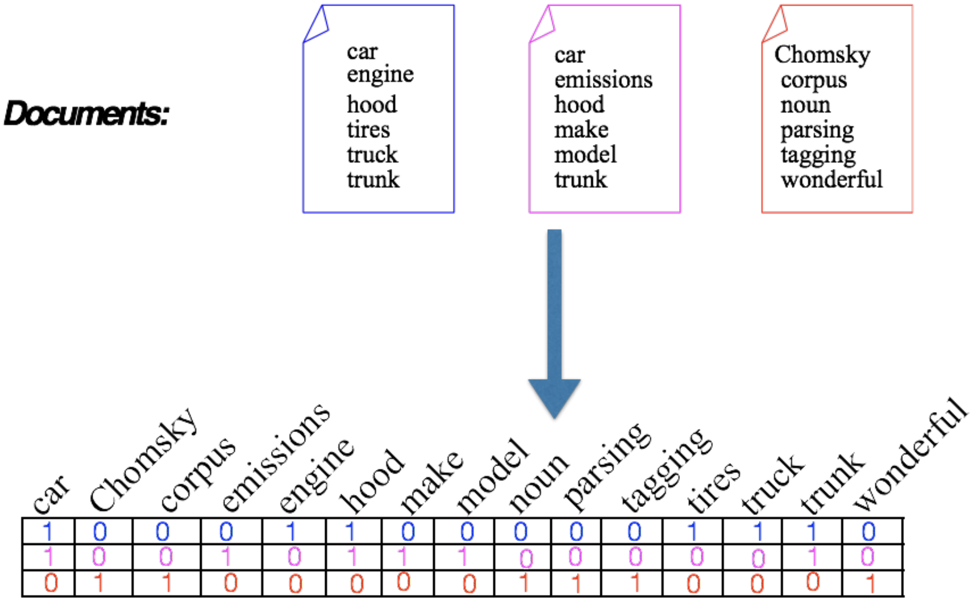


Figure 1: Image as vector

In the above, an image that is of size $K \times K$ is converted to a vector of size K^2 by simply having each entry of the vector corresponding to a pixel in the original image with value of that entry being represented by intensity or color of the pixel. (for a color image the vector will be of size $3K^2$) so that each of RGB values have a corresponding entry in the feature vector.

Above is an example of bag-of-words feature representation of documents where each document is represented by a vector whose dimensionality is that of number of words in the lexicon (number



car	Chomsky	corpus	emissions	engine	hood	make	model	noun	parsing	tagging	tires	truck	trunk	wonderful
1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
1	0	0	1	0	1	1	1	0	0	0	0	0	1	0
0	1	1	0	0	0	0	0	1	1	1	0	0	0	1

Figure 2: Bag of words feature for documents

of words in the english dictionary for english documents for example). Each entry of the vector for a given document, corresponds to one word in the lexicon and the value for that entry is the number of times the word appears in the given document.

2 Clustering

Clustering corresponds to grouping points so that points that are similar are in the same group and dissimilar points are in different groups. To formalize this, a K-ary clustering of points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is partitioning the n points into K groups. We will use two alternative notations to represent a K-ary clustering of points. First is by a K-partition of the n points given by $C_1, \ldots, C_K$ disjoint subsets of $\{1, \ldots, n\}$ such that $\bigcup_{k=1}^{K} C_k = \{1, \ldots, n\}$. The alternative representation is as cluster assignment function c that maps each data point to one of clusters 1 to K. That is $c(\mathbf{x}_t) = j$ implies that according to clustering c, point x_t belongs to cluster j. We will further use the representation $n_j = |C_j|$ the number of points in cluster j.

2.1 Various Clustering Objectives

We shall assume that given two points $\mathbf{x}_t$ and $\mathbf{x}_s$, dissimilarity$(\mathbf{x}_t, \mathbf{x}_s)$ is a function that measures dissimilarity between points $\mathbf{x}_t$ and $\mathbf{x}_s$. As an example, think of dissimilarity$(\mathbf{x}_t, \mathbf{x}_s)$ as the distance or squared distance between the points. The farther they are the more dissimilar they are. Below are a few possible clustering objectives, that is objectives our clustering assignments will optimize for.

1. Minimize within-cluster scatter

$$M_1 = \sum_{j=1}^{K} \sum_{\mathbf{x}_s, \mathbf{x}_t \in C_j} \text{dissimilarity}(\mathbf{x}_s, \mathbf{x}_t)$$

2. Maximize between-cluster scatter

$$M_2 = \sum_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(\mathbf{x}_s, \mathbf{x}_t)$$

3. Maximize smallest between-cluster distance

$$M_3 = \min_{\mathbf{x}_s, \mathbf{x}_t : c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \text{dissimilarity}(\mathbf{x}_s, \mathbf{x}_t)$$

4. Minimize largest within-cluster distance

$$M_4 = \max_{j \in [K]} \max_{\mathbf{x}_s, \mathbf{x}_t \in C_j} \text{dissimilarity}(\mathbf{x}_s, \mathbf{x}_t)$$

5. Minimize total within-cluster average scatter

$$M_5 = \sum_{j=1}^{K} \sum_{\mathbf{x}_s \in C_j} \text{dissimilarity}(\mathbf{x}_s, C_j) = \sum_{j=1}^{K} \sum_{\mathbf{x}_s \in C_j} \frac{1}{n_j} \sum_{\mathbf{x}_t \in C_j} \text{dissimilarity}(\mathbf{x}_s, \mathbf{x}_t)$$

6. Minimize total within-cluster variance: $\mathbf{r}_j = \frac{1}{n_j} \sum_{\mathbf{x} \in C_j} \mathbf{x}$

$$M_6 = \sum_{j=1}^{K} \sum_{\mathbf{x}_t \in C_j} \text{dissimilarity}(\mathbf{x}_t, \mathbf{r}_j)$$

M_2 Versus M_3 The objective M_2 aims at maximizing total between cluster scatter/dissimilarity and M_3 aims at maximizing minimum between cluster scatter/dissimilarity. These two objectives are both equally valid ones which we shall demonstrate now.

Example where M_2 is better than M_3: (outliers are bad for M_3)
Consider the set of points placed in 2D as the ones depicted in the figure below:

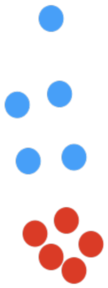


Figure 3: M_2 better than M_3

In the figure above, notice that maximizing M_3 would lead to the outlier blue point being its own cluster and all other points being one cluster. However, maximizing M_2 would lead to the clustering shown in above figure. While this might seem a bit toyish, the example can be made more real by noticing that if we take two gaussians with equal variance and have them separated, then, as we increase number of points sampled in each cluster will ensure that we have an outlier whose distance to other points in larger than separation. In fact, if we consider higher dimensional cases,

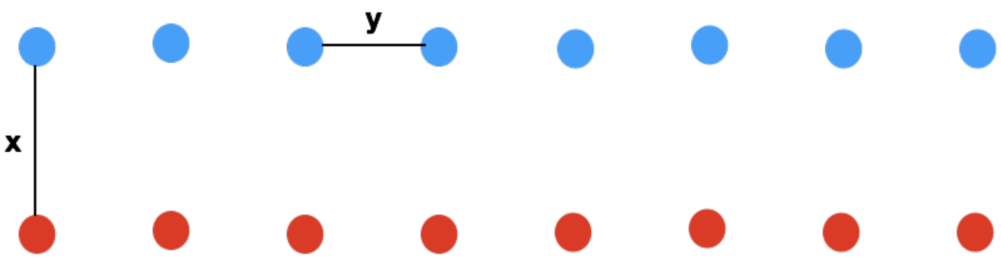


Figure 4: M_2 better than M_3

this difference is even more pronounced. However, as long as number of points in the two clusters are roughly the same, M_2 will correctly cluster the two gaussians.
Example where M_3 is better than M_2: (long connected clusters are bad for M_2)
Consider the set up of points in the figure below: Say x is the distance between the two rows and y is the distance between two columns and assume $x > y$. Then it is easy to check that first, maximizing objective M_3 gives the clustering indicated in the figure above. This is because, distance between columns is smaller then distance between the two rows and so M_3 would prefer making each row it own cluster. However, depending on number of points and how large x is relative to y, maximizing M_2 leads to one cluster be the left half of the points and other right half as depicted in figure below.

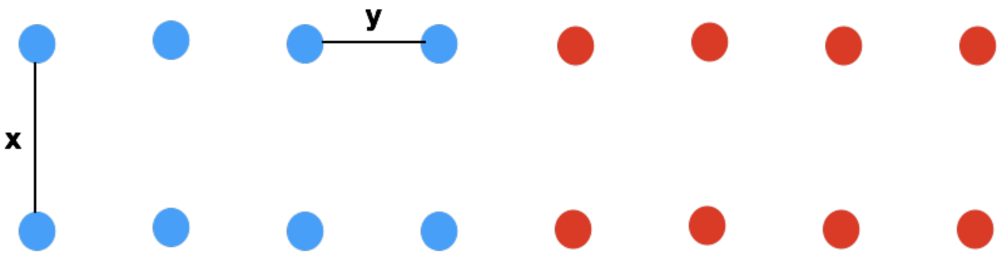


Figure 5: M_2 better than M_3

As an example, in the figure below if x is 2 times y, optimizing M_2 would lead to the clustering shown above.

Minimizing $M_1 \equiv$ **Maximizing** M_2 Why is this?
For any given set of points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, irrespective of clustering assignments we use, the quantity $\sum_{t,s:t\neq s} \text{dissimilarity}(\mathbf{x}_t, \mathbf{x}_s)$ is a constant. But note that

$$M_1 + M_2 = \sum_{t,s:t\neq s} \text{dissimilarity}(\mathbf{x}_t, \mathbf{x}_s)$$

Hence Minimizing M_1 is same as minimizing $\sum_{t,s:t\neq s} \text{dissimilarity}(\mathbf{x}_t, \mathbf{x}_s) - M_2$ which is same as maximizing M_2.

3 Single-Link Clustering Algorithm

In general minimizing or maximizing given objectives might not be computationally efficiently possible. However now we review the objective of maximizing M_3 and see that this objective can be achieved by a simple algorithm called the single-link clustering objective described below.

- Initialize n clusters with each point $\mathbf{x}_t$ to its own cluster
- Until there are only K clusters, do

  1. Find closest two clusters and merge them into one cluster
  2. Update between cluster distances (called proximity matrix)

Theorem 1. *Single link clustering algorithm maximizes objective M_3.*

Proof. We will prove this by contradiction. First for simplicity we shall assume that distances between all pairs of points are unique so we don't worry about tie breaking. Now, just as in the animation depicted in lecture slides, on every round, we look at distance between clusters and merge the two clusters that are closest. Here distance between two clusters is given by minimum distance between pairs of points where one point is in first cluster and other is in the second. Now, each time we merge two points, let us draw an edge between the closest pair of points, where first point is in first merged cluster and second is in the second merged cluster. We will refer to these points as merged points. A cluster will consist of points, all connected by edges corresponding to merged points and further, this graph will be a tree.

Now, say c was the clustering returned by single link clustering and let c' be the clustering that maximizes our objective M_3. If the two clusterings didn't match then there has to be points $\mathbf{x}_t$ and $\mathbf{x}_s$ such that, $c(\mathbf{x}_t) = c(\mathbf{x}_s)$ but $c'(\mathbf{x}_t) \neq c'(\mathbf{x}_s)$. The key observation for the proof is that distances of merged points thus far, under single link algorithm are smaller than minimum between cluster distances under the single link algorithm. That is,

$$\min_{t,s:c(\mathbf{x}_t)\neq c(\mathbf{x}_s)} d(\mathbf{x}_t, \mathbf{x}_s) > \text{Distance of merged points so far}$$

This is because we merge closest two clusters at every step to form new clusters.

Now, since the two points $\mathbf{x}_t$ and $\mathbf{x}_s$ are in the same cluster according to single link algorithm, there should be a path between $\mathbf{x}_t$ and $\mathbf{x}_s$ in the tree consisting of merged edges. This path is depicted in the Figure 6.

Now consider the distance between cluster with id $c'(\mathbf{x}_t)$ and points in cluster with id $c'(\mathbf{x}_s)$ under clustering c'. This distance is smaller than distance of at least one of the merged distances. Specifically in the figure this is depicted by edge ab. However, by our observation, all merged distances are smaller than minimum inter-cluster distance according to c. This shows that minimum inter cluster distance according c' is smaller than minimum inter-cluster distance according to c. Thus single link clustering has a larger value for M_3 than c' which would be a contradiction since c' is said to be the optimal solution. Hence $c = c'$.

□

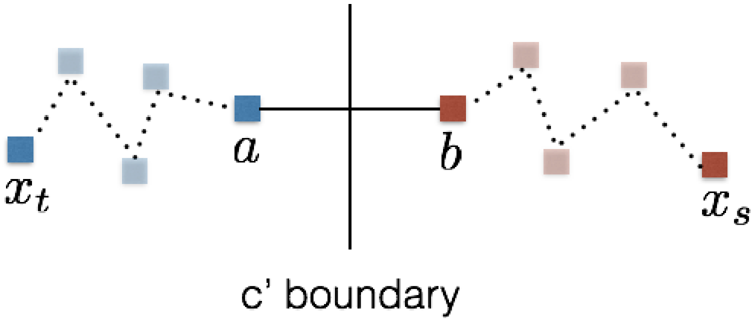


Figure 6: $\mathbf{x}_t$ and $\mathbf{x}_s$ are in same cluster according single link but not c'

4 K-means Algorithm

The second algorithm we consider is the k-means algorithm which aims at minimizing objective M_6 and equivalently M_6. The algorithm is as stated below.

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point
  $$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \|\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1}\|^2$$
  2. For each $j \in [K]$, set new representative as
  $$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t$$
  3. $m \leftarrow m + 1$

4.1 M_5 Versus M_6 for euclidean distance squared

Theorem 2. *When* dissimilarity$(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{y})^2 = \|\mathbf{x} - \mathbf{y}\|^2$, *then* $M_5 \equiv M_6$.

Proof. We shall show first that objectives M_5 and M_6 are equivalent when dissimilarity is measured by euclidean distance squared.

$$\begin{aligned}
M_5 &= \sum_{j=1}^{K} \sum_{\mathbf{x}_s \in C_j} \text{dissimilarity}(\mathbf{x}_s, C_j) \\
&= \sum_{j=1}^{K} \sum_{\mathbf{x}_s \in C_j} \frac{1}{n_j} \sum_{\mathbf{x}_t \in C_j} \text{dissimilarity}(\mathbf{x}_s, \mathbf{x}_t) \\
&= \sum_{j=1}^{K} \sum_{\mathbf{x}_s \in C_j} \frac{1}{n_j} \sum_{\mathbf{x}_t \in C_j} \|\mathbf{x}_s - \mathbf{x}_t\|^2 \\
&= \sum_{j=1}^{K} \sum_{\mathbf{x}_s \in C_j} \frac{1}{n_j} \sum_{\mathbf{x}_t \in C_j} \|\mathbf{x}_s - \mathbf{r}_j + \mathbf{r}_j - \mathbf{x}_t\|^2 \\
&= \sum_{j=1}^{K} \frac{1}{n_j} \sum_{\mathbf{x}_s \in C_j} \sum_{\mathbf{x}_t \in C_j} \left(\|\mathbf{x}_s - \mathbf{r}_j\|^2 + \|\mathbf{r}_j - \mathbf{x}_t\|^2 + 2(\mathbf{x}_t - r_j)^\top (r_j - \mathbf{x}_s)\right) \\
&= \sum_{j=1}^{K} \frac{1}{n_j} \left(n_j \sum_{\mathbf{x}_s \in C_j} \|\mathbf{x}_s - \mathbf{r}_j\|^2 + n_j \sum_{\mathbf{x}_t \in C_j} \|r_j - \mathbf{x}_t\|^2 + 2 \sum_{\mathbf{x}_s \in C_j} \sum_{\mathbf{x}_t \in C_j} (\mathbf{x}_t - r_j)^\top (\mathbf{r}_j - \mathbf{x}_s) \right) \\
&= \sum_{j=1}^{K} \left(\sum_{\mathbf{x}_s \in C_j} \|\mathbf{x}_s - \mathbf{r}_j\|^2 + \sum_{\mathbf{x}_t \in C_j} \|\mathbf{r}_j - \mathbf{x}_t\|^2 + 2 \sum_{\mathbf{x}_s \in C_j} (\frac{1}{n_j} \sum_{\mathbf{x}_t \in C_j} \mathbf{x}_t - r_j)^\top (\mathbf{r}_j - \mathbf{x}_s) \right) \\
&= \sum_{j=1}^{K} \left(\sum_{\mathbf{x}_s \in C_j} \|\mathbf{x}_s - \mathbf{r}_j\|^2 + \sum_{\mathbf{x}_t \in C_j} \|r_j - \mathbf{x}_t\|^2 + 2 \sum_{\mathbf{x}_s \in C_j} (\mathbf{r}_j - \mathbf{r}_j)^\top (\mathbf{r}_j - \mathbf{x}_s) \right) \\
&= \sum_{j=1}^{K} \left(\sum_{\mathbf{x}_s \in C_j} \|\mathbf{x}_s - \mathbf{r}_j\|^2 + \sum_{\mathbf{x}_t \in C_j} \|\mathbf{r}_j - \mathbf{x}_t\|^2 \right) \\
&= 2 \sum_{j=1}^{K} \sum_{\mathbf{x}_s \in C_j} \|\mathbf{x}_s - \mathbf{r}_j\|^2 \\
&= 2M_6
\end{aligned}$$

□

4.2 K-means algorithm minimizes objective M_6

It turns out that the k-means algorithm is grared towards minimizing objective M_6. To see this, we shall rewrite the objective M_6 as follows:

$$M_6 = \sum_{j=1}^{K} \sum_{s \in C_j} \left\| \mathbf{x}_s - \frac{1}{n_j} \sum_{t \in C_j} \mathbf{x}_t \right\|^2 = \min_{\mathbf{r}_1, \ldots, \mathbf{r}_K} \sum_{j=1}^{K} \sum_{s \in C_j} \|\mathbf{x}_s - \mathbf{r}_j\|^2$$

Lemma 3. *For any vector* $\mathbf{x}$*, we have that,*

$$\frac{1}{n_j} \sum_{s \in C_j} \|\mathbf{x}_s - \mathbf{x}\|^2 = \frac{1}{n_j} \sum_{s \in C_j} \left\| \mathbf{x}_s - \frac{1}{n_j} \sum_{t \in C_j} \mathbf{x}_t \right\|^2 + \left\| \mathbf{x} - \frac{1}{n_j} \sum_{t \in C_j} \mathbf{x}_t \right\|^2$$

Proof.

$$
\begin{aligned}
&\frac{1}{n_j}\sum_{s\in C_j}\|\mathbf{x}_s-\mathbf{x}\|^2\\
&\qquad=\frac{1}{n_j}\sum_{s\in C_j}\left\|\mathbf{x}_s-\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t+\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t-\mathbf{x}\right\|^2\\
&\qquad=\frac{1}{n_j}\sum_{s\in C_j}\left(\left\|\mathbf{x}_s-\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t\right\|^2+\left\|\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t-\mathbf{x}\right\|^2+2\left(\mathbf{x}_s-\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t\right)^\top\left(\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t-\mathbf{x}\right)\right)\\
&\qquad=\frac{1}{n_j}\sum_{s\in C_j}\left\|\mathbf{x}_s-\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t\right\|^2+\left\|\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t-\mathbf{x}\right\|^2+2\frac{1}{n_j}\sum_{s\in C_j}\left(\mathbf{x}_s-\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t\right)^\top\left(\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t-\mathbf{x}\right)\\
&\qquad=\frac{1}{n_j}\sum_{s\in C_j}\left\|\mathbf{x}_s-\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t\right\|^2+\left\|\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t-\mathbf{x}\right\|^2+2\left(\frac{1}{n_j}\sum_{s\in C_j}\mathbf{x}_s-\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t\right)^\top\left(\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t-\mathbf{x}\right)\\
&\qquad=\frac{1}{n_j}\sum_{s\in C_j}\left\|\mathbf{x}_s-\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t\right\|^2+\left\|\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t-\mathbf{x}\right\|^2
\end{aligned}
$$

□

The above lemma easily shows us that $\mathbf{r}_j=\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t$ is the minimizer of $\sum_{s\in C_j}\|\mathbf{x}_s-\mathbf{r}_j\|^2$. This in turn shows us that as claimed earlier,

$$M_6=\sum_{j=1}^{K}\sum_{s\in C_j}\left\|\mathbf{x}_s-\frac{1}{n_j}\sum_{t\in C_j}\mathbf{x}_t\right\|^2=\min_{\mathbf{r}_1,\ldots,\mathbf{r}_K}\sum_{j=1}^{K}\sum_{s\in C_j}\|\mathbf{x}_s-\mathbf{r}_j\|^2$$

Now the key idea to see why k-means is optimizing objective M_6 is to treat k-means algorithm as a procedure that picks cluster assignments and centers $\mathbf{r}_1,\ldots,\mathbf{r}_K$ jointly to minimize with respect to cluster assignment and the K centers, the objective

$$O(c,\mathbf{r}_1,\ldots,\mathbf{r}_K)=\sum_{j=1}^{K}\sum_{s\in C_j}\|\mathbf{x}_s-\mathbf{r}_j\|^2$$

The following theorem formalizes the notion that k-means aims at minimizing the above objective:

Theorem 4. *For any iteration m of the k-means algorithm, we have that*

$$O(\hat{c}^m,\hat{\mathbf{r}}_1^m,\ldots,\hat{\mathbf{r}}_K^m)\le O(\hat{c}^{m-1},\hat{\mathbf{r}}_1^{m-1},\ldots,\hat{\mathbf{r}}_K^{m-1})$$

Proof. For any arbitrary choice of centroids $\mathbf{r}_1^{m-1},\ldots,\mathbf{r}_K$, note that by definition of $\hat{c}^m$,

$$O(\hat{c}^m,\mathbf{r}_1^{m-1},\ldots,\mathbf{r}_K^{m-1})=\min_c O(c,\mathbf{r}_1^{m-1},\ldots,\mathbf{r}_K)\le O(\hat{c}^{m-1},\mathbf{r}_1^{m-1},\ldots,\mathbf{r}_K^{m-1})$$

This is because we assign each point to centroid closest to it.
On the other hand, by Lemma 3 we have that for any cluster assignment, the means of points in each cluster is the optimal centroid and so, specifically,

$$\begin{aligned}
O(\hat{c}^m, \mathbf{r}_1^{m-1}, \ldots, \mathbf{r}_K^{m-1}) &= \sum_{j=1}^{K} \sum_{s \in C_j^m} \left\| \mathbf{x}_s - \mathbf{r}_j^{m-1} \right\|^2 \\
&\geq \min_{\mathbf{r}_1, \ldots, \mathbf{r}_K} \sum_{j=1}^{K} \sum_{s \in C_j^m} \left\| \mathbf{x}_s - \mathbf{r}_j \right\|^2 \\
&= \min_{\mathbf{r}_1, \ldots, \mathbf{r}_K} \sum_{j=1}^{K} \sum_{s \in C_j^m} \left\| \mathbf{x}_s - \frac{1}{|\hat{C}_j^m|} \sum_{t \in \hat{C}_j^m} \mathbf{x}_t \right\|^2 \\
&= O(\hat{c}^m, \mathbf{r}_1^m, \ldots, \mathbf{r}_K^m)
\end{aligned}$$

Combining the two we conclude the theorem statement.

□