COS 424: Interacting with Data

Lecturer: David Blei — Lecture # 8
Scribe: Ana Pop, Hanjun Kim — February 28, 2008

1 Announcements

Homework 2 will come out soon. It will be very long, so start early, please.

2 Introduction

In the previous lectures, we talked about supervised methods. There, we learn from given data and then apply this knowledge on a new data point, where we predict the category of the new data point. We will now look at unsupervised learning methods where it is not clear in what categories, if any, the data will fall into.

3 k-means Clustering

3.1 Clustering

Clustering is used to segment data into groups of similar items. It is useful for automatically organizing data, finding some hidden structure in the data, and a compression form (for example, representing 3000 data points together in a bin called 1).

Some examples of when this would be useful are in predicting buying patterns of customers, finding patterns/groups of genes to learn the structure of the data on a higher level, grouping MySpace users according to different interests. Google would use clustering to group search results for "jaguar" into "car", "animal", and "OS" categories.

3.2 Clustering Set-Up

Clustering data such as email, gene expression profiles, or purchase histories, are represented as $\mathcal{D} = \{\mathbf{x_1}, ..., \mathbf{x_N}\}$. Since the data is p-dimensional, we represent it as $\mathbf{x_n} = \{x_{n,1}, ..., x_{n,p}\}$. The distance function is $d(\mathbf{x_n}, \mathbf{x_m})$ between two data points. The k groups we want to divide our data into are $\{z_1, ..., z_N\}$ where $x \in \{1, ..., K\}$. So we want to assign a label to each data point x. Note that we could assign them randomly, but we want the assignment to be meaningful.

3.3 K-Means on Example Data

Consider the following data in Figure 1. A good distance function to use is the squared Euclidean distance $d(\mathbf{x_n}, \mathbf{x_m}) = \sum_{i=1}^{p} (x_{n,i} - x_{m,i})^2 = ||x_n - x_m||^2$. But now we want to segment the data into k groups. We choose $k = 4$ for now because finding k is complicated.

Consider the intuitive steps that the k-means algorithm would take, as shown in Figure 2. We begin with $k = 4$ randomly placed initial means. We recompute the centers of the data points. Then we reassign the data points to new means. The detailed algorithm is as follows:

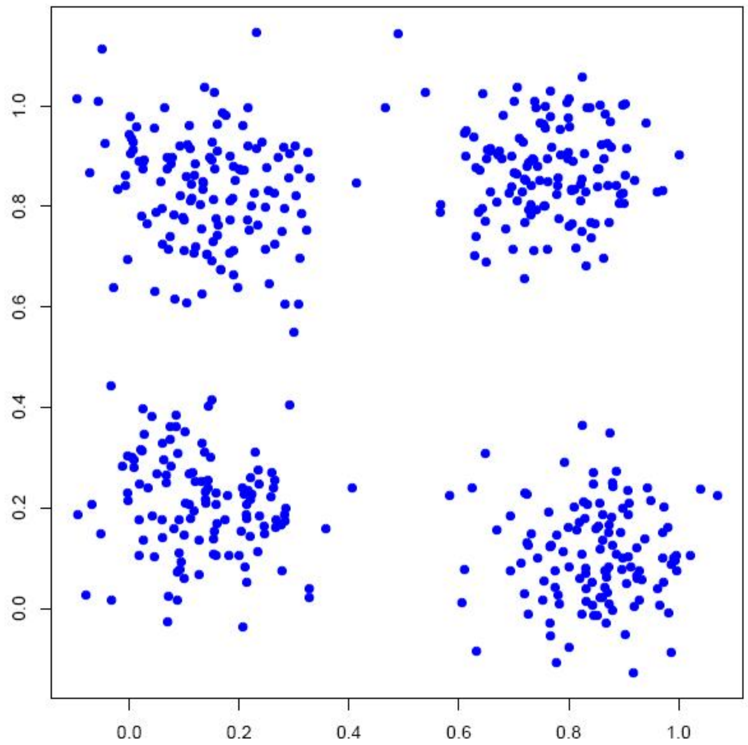

Figure 1: 500 2-dimensional data points $\mathbf{x_n} = \langle \mathbf{x_{n,1}}, \mathbf{x_{n,2}} \rangle$

1. Initialization
   (a) Data is $\mathbf{x_{1:N}}$
   (b) Randomly pick initial cluster means $\mathbf{m_{1:k}}$
2. Repeat
   (a) Assign each data point to its closest mean,
   $$\mathbf{z_n} = \text{arg min}_{i \in 1\ldots k} d(\mathbf{x_n}, \mathbf{m_i})$$
   (b) Compute average distance between all coordinates assigned to the new cluster and the cluster mean,
   $$\mathbf{m_k} = \frac{1}{N_k} \sum_{\mathbf{n:z_n=k}} \mathbf{x_n}$$
3. Until assignments $\mathbf{z_{1:N}}$ do not change

3.4 Objective Function

We measure how well the algorithm is doing using the sum of the squared distances of each point to its respective mean, $F(\mathbf{z_{1:N}}, \mathbf{m_{1:k}}) = \frac{1}{2}\sum_{n=1}^{N} ||\mathbf{x_n} - \mathbf{m_{Z_n}}||^2$. Remember that $\mathbf{x_n}$ is a data point and $\mathbf{m_{z_n}}$ is the mean for that data point. The objective function for our example is shown in Figure 3. We find convergence by looking at the relative change between successive rounds and stop when we have reached what we deem a negligible difference.

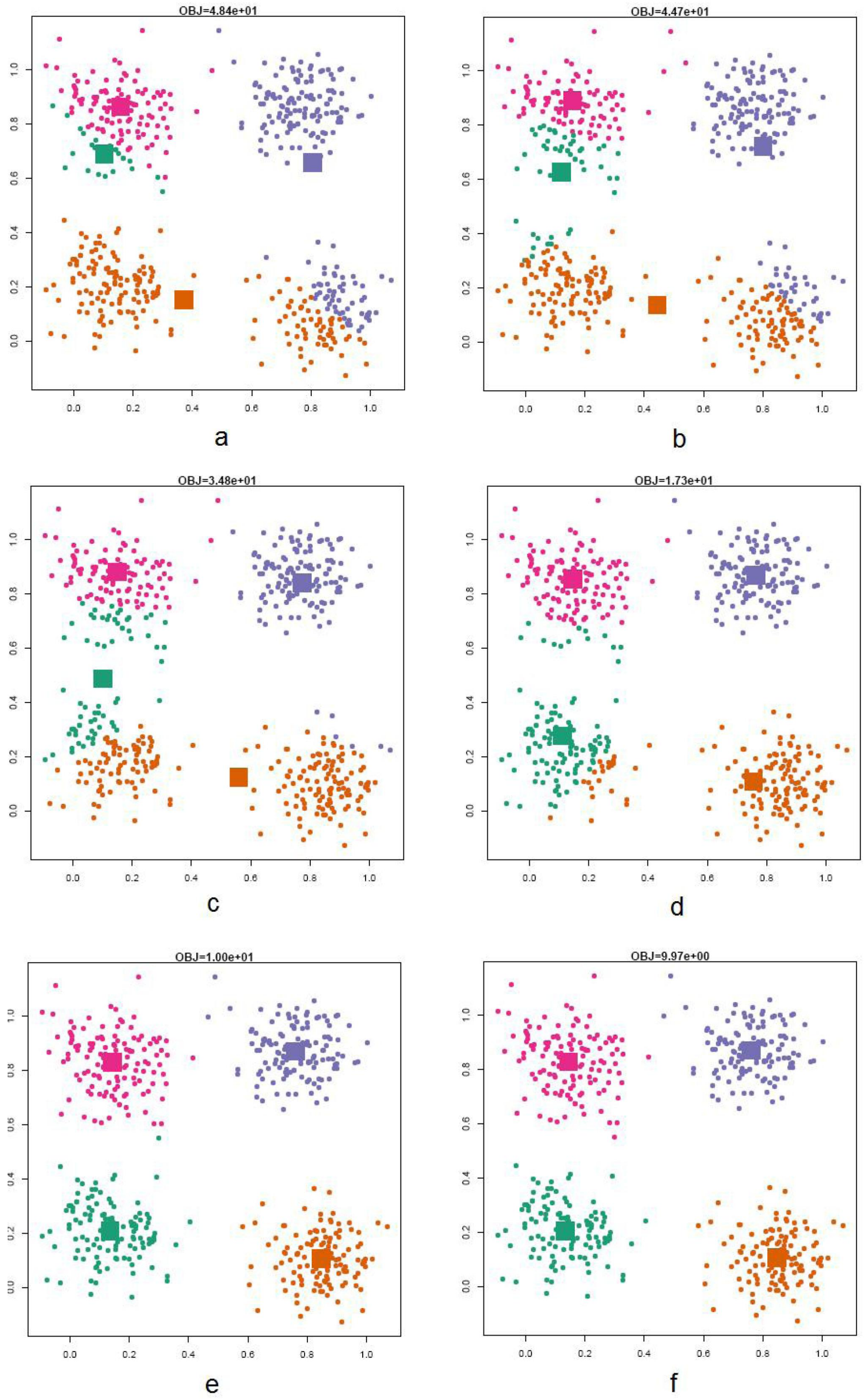

Figure 2: Progression of the k-means clustering algorithm on data from Figure 1

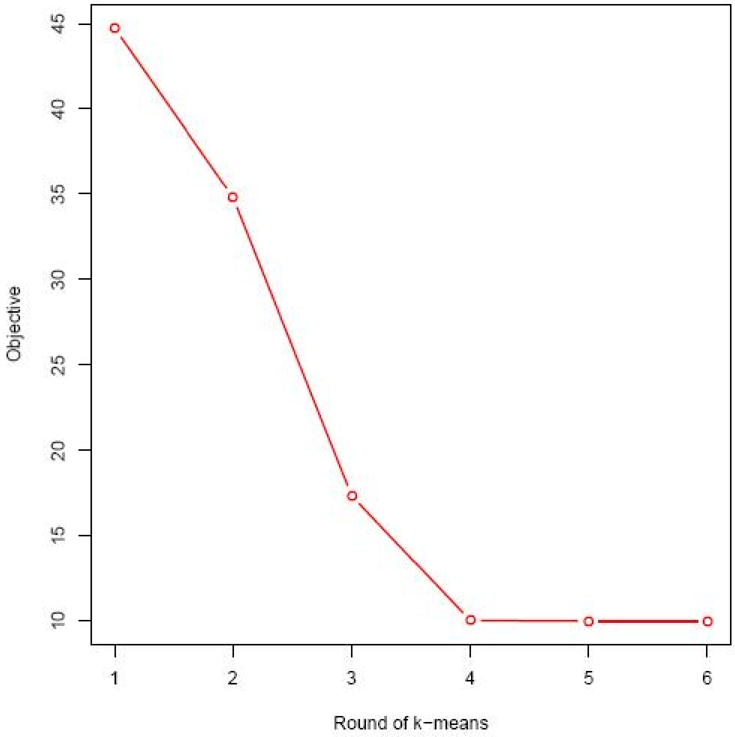

Figure 3: Objective function for the clusters in Figure 2

3.5 Coordinate Descent

k-means is a coordinate descent algorithm. First, it assigns each point to its closest mean while keeping the means fixed, thus minimizing F (from the previous section) with respect to $\mathbf{z_{1:N}}$. Second, it computes the new means of every cluster while keeping the assignments fixed, thus minimizing F with respect to $\mathbf{m_{1:k}}$. Note that since k-means attempts to minimize both these quantities, it is not a convex function, so it does not have a global minimum. It does, however, have local minimums, so it is essential to run the algorithm multiple times.

3.6 Compressing Images

Take the application of compressing a picture. In this case, we want to replace pixels (coordinates) with some color assignments (means), effectively using k-means to compress the image. The progression of the coloring of the image is shown in Figure 4 for different k values. In this particular application, the objective function tells us how distorted a picture is compared to the original one. Notice that the picture becomes less distorted the more clusters we use.

3.7 K-Medoids

So far we have only used Euclidean distance as a distance measure. However, when we have discrete multivariate data, or data that should not be clustered in circles, or data that is on different scales, Euclidean distance is not appropriate. Instead, we use the k-medoids

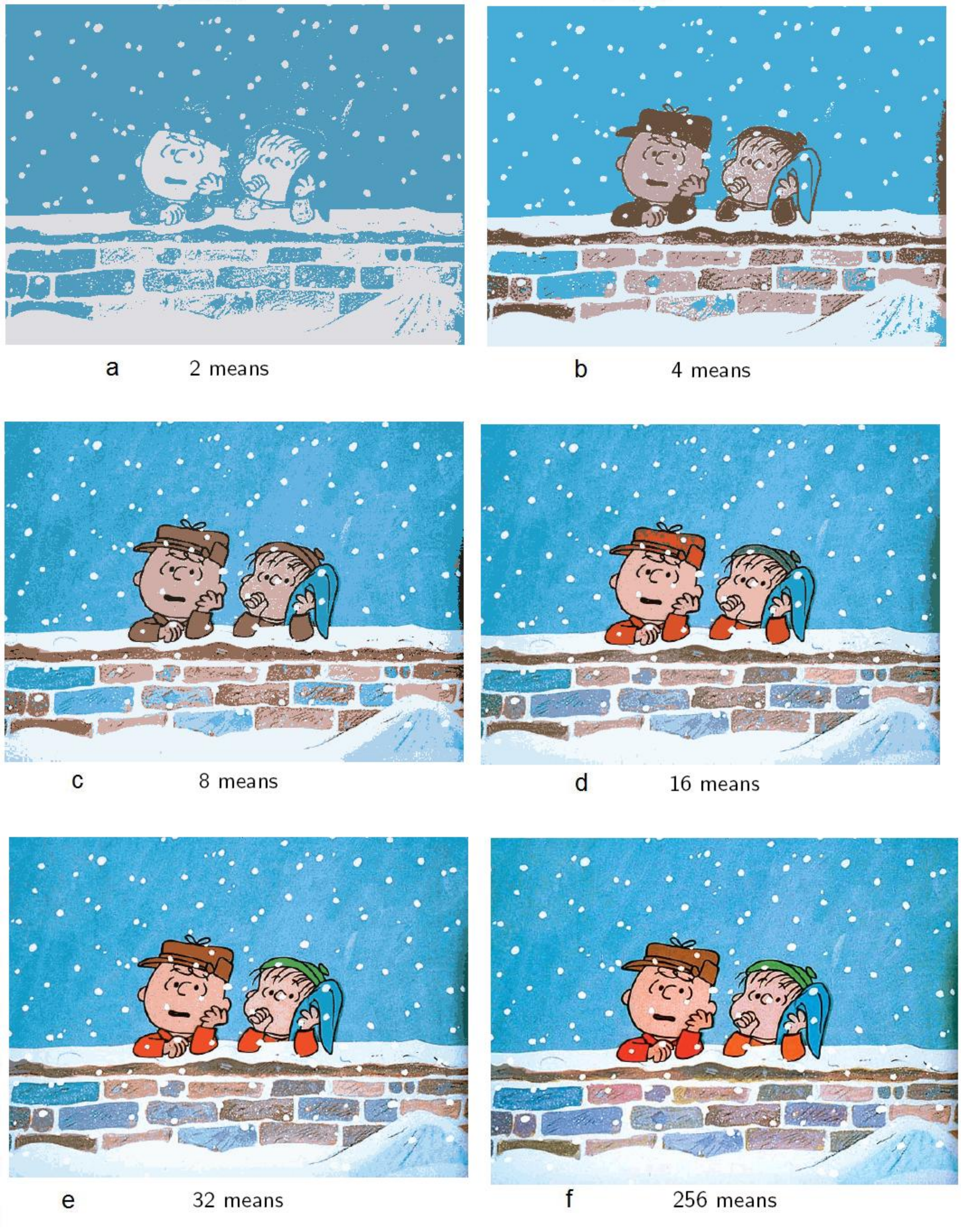

Figure 4: Using k-means to compress an image with $k = 2, 4, 8, 16, 32, 256$

algorithm, which does not require us to know the means, only distances between data points. The k-medoids algorithm is as follows.

1. Initialization
   (a) Data is $\mathbf{x_{1:N}}$
   (b) Pick initial cluster identities $\mathbf{m_{1:k}}$
2. Repeat
   (a) Assign each data point to its closest center,
   $$\mathbf{z_n} = \textsf{arg min}_{i \in 1 \ldots k} d(\mathbf{x_n}, \mathbf{m_i})$$
   (b) Find a data point in a cluster that is closest to the other data points in the cluster,
   $$\mathbf{i_k} = \textsf{arg min}_{\mathbf{n:z_n=k}} \sum_{\mathbf{m:z_m=k}} d(\mathbf{x_n}, \mathbf{x_m})$$
   (c) New cluster centers are set to the closest data points, $\mathbf{m_k} = \mathbf{x_{i_k}}$
3. Until assignments $\mathbf{z_{1:N}}$ do not change

3.8 Choosing k

This is a hard problem. An intuitive way is to choose it in such a way as to end up with ”natural” clusters, but this is not very well defined. One heuristic is to use the kink in the objective function, as in Figure 6. Notice that before $k = 4$, the successive increase in k yields a large improvement over the previous k. But after $k = 4$, we do not improve so drastically anymore. This suggests that $k = 4$ is the correct value.

4 Hierarchical Clustering

4.1 Introduction

Hierarchical clustering is widely used. It builds a tree of data in order to merge similar groups of points. Thus, visualizing this tree is a good summary of the data. Its advantage over k-means is that there is no need to pick k in advance because it uses a measure of distance between groups of data points. To perform agglomerative clustering, we begin by placing every data point in its own cluster. Then we iteratively merge the closest groups, not necessarily between individual data points but between already clustered points. We repeat until we have merged all the data into a single cluster. Sample iterations from this process are shown in Figure 7.

4.2 Dendrogram

Running the algorithm results in a sequence of groupings. Each level of the tree is a segmentation of the data. It is a monotonic algorithm so the similarity between clusters decreases with each level. This is for larger iterations we begin to add points that are father and farther away. A dendrogram provides a visualization of the data and of the similarity between two merged groups.

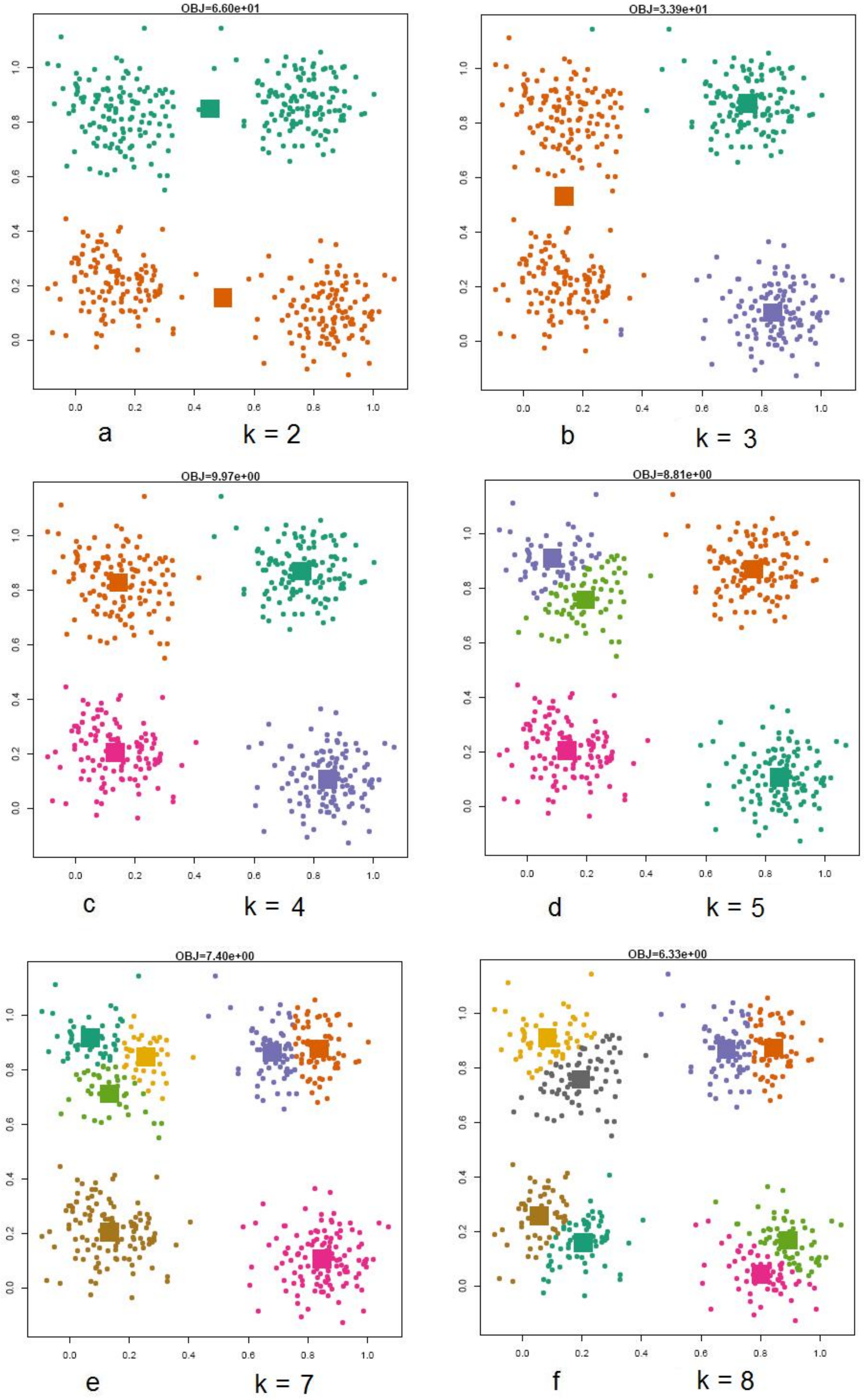

Figure 5: Different cluster means for several k values

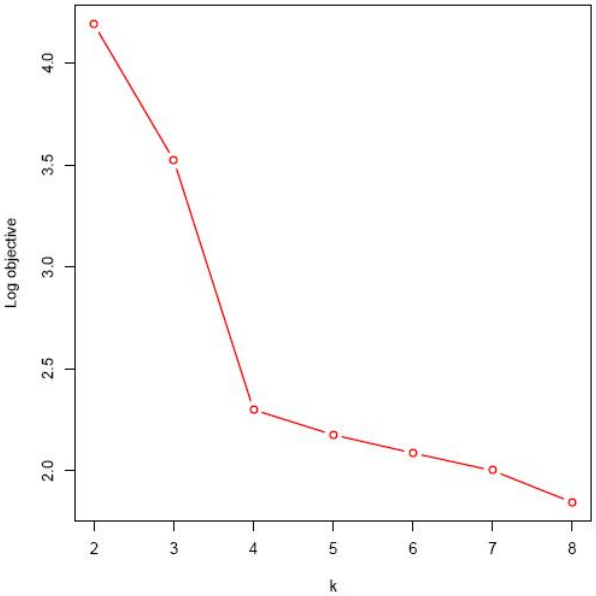

Figure 6: An example of a kink in the objective function

Remember that in k-means clustering, we chose k at the place where the change between successive iterations started to decrease. In contrast, here we begin by grouping clusters that are very similar to each other so we are most likely merging points that are supposed to be in the same cluster. To determine where different clusters exist, we choose the place where the distance starts to change most. In this way, we capture the place where we are most likely merging two large groups. As shown in Figure 8, we are slowly increasing the height until we reach the left node at height 40 and the right two nodes at height 20.

4.3 Group Similarity

There are three ways we can define inter-group similarity: single-linkage, complete-linkage, and group average.

In single-linkage, the distance between two clusters is the distance between the closest points between the two clusters, ie $d_{SL}(G,H) = \mathsf{min}_{i \in G, j \in H} d_{i,j}$. The problem with using this metric is a phenomena called "chaining", where we can observe sequences of points together, as in a chain, indicating that the two endpoints in the chain are in the same cluster when they may not necessarily be.

In complete-linkage, the distance between two clusters is the distance between the farthest points between the two clusters, ie $d_{CL}(G,H) = \mathsf{max}_{i \in G, j \in H} d_{i,j}$. The problem with using this metric is that we might not merge close groups because group outliers are far apart.

In group average, the distance between two clusters is the average distance between all pairs of points between the two clusters, ie $d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{j \in H} d_{i,j}$. This is a compromise between the previous two methods, but results might change if monotone transformations are applied.

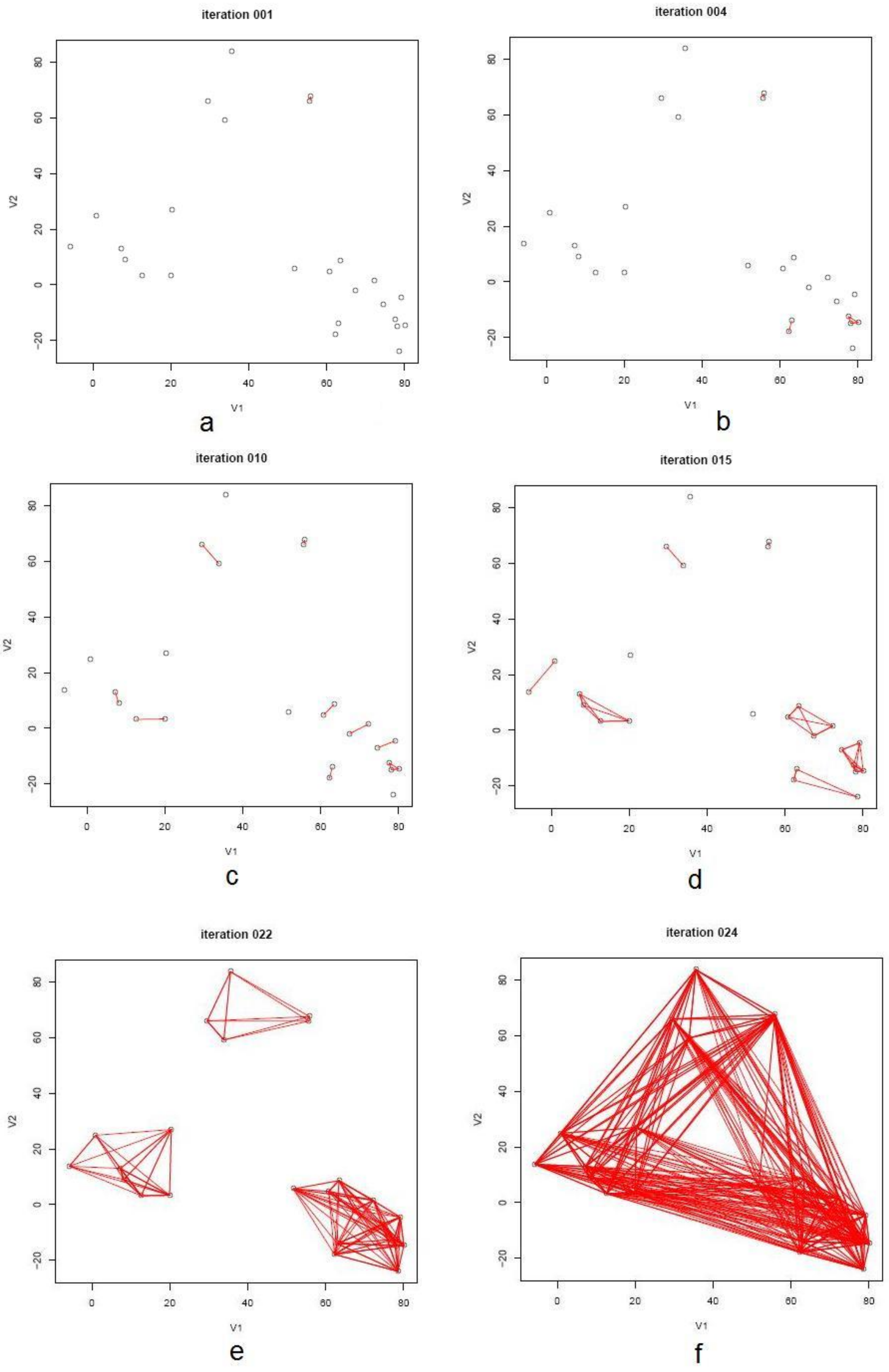

Figure 7: Agglomerative clustering for some iterations

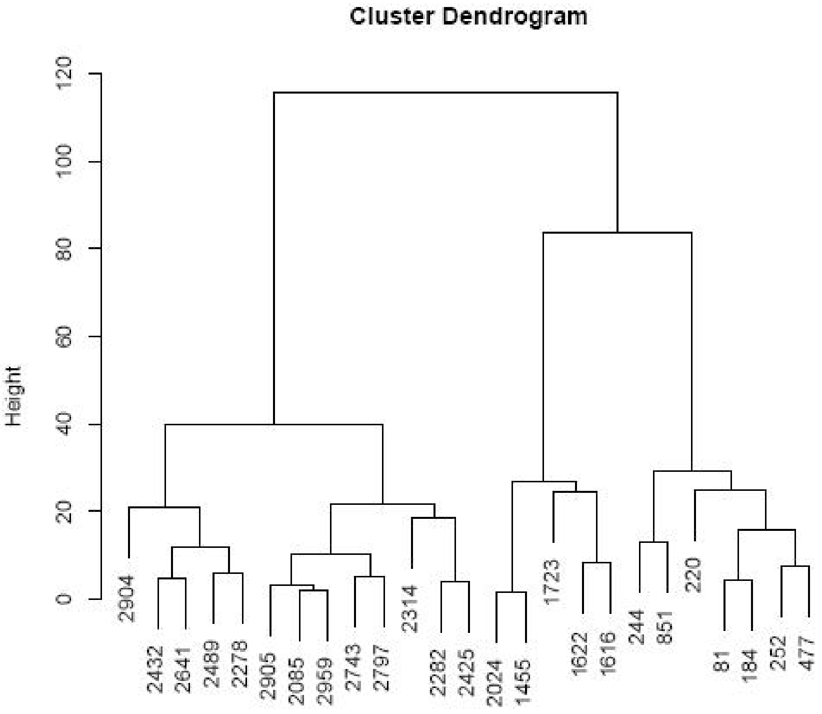

Figure 8: Dendrogram for the data in Figure 7