CS 395T	Large-Scale Data Mining	Fall 2001

Lecture 6, 7 & 8

Lecturer: Inderjit Dhillon Date: Oct. 4, 9 & 16, 2001

Keywords: *PCA, Clustering, Hierarchical agglomerative clustering, k-means*

1 Principal Components Analysis (PCA)

Let $p(\mathbf{x})$ be the multivariate normal density function in d dimensions, where μ is the d-component mean vector and $\mathbf{\Sigma}$ is the $d \times d$ covariance matrix, i.e.,

$$\begin{aligned} p(\mathbf{x}) &= \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}, \\ \text{where } \mu &= E[\mathbf{x}], \\ \mathbf{\Sigma} &= E[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T] \end{aligned}$$

and $|\mathbf{\Sigma}|$ is the determinant of $\mathbf{\Sigma}$. Given data points $\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n$ from this distribution, the sample mean and the sample covariance matrix are

$$\begin{aligned} \hat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i, \\ \hat{\mathbf{\Sigma}} &= \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i-\mu)(\mathbf{x}_i-\mu)^T. \end{aligned}$$

The principal components are the leading eigenvectors of the sample covariance matrix. PCA projects the vectors $\mathbf{x}_1 \dots \mathbf{x}_n$ onto the principal components. Note that the eigenvectors of $\hat{\mathbf{\Sigma}}$ are the left singular vectors of the matrix

$$\frac{1}{\sqrt{n-1}}[\mathbf{x}_1-\mu, \mathbf{x}_2-\mu, \dots, \mathbf{x}_n-\mu].$$

Thus PCA can be obtained from the SVD of *mean centered data.* The mean centering is the important difference between PCA and SVD and can yield qualitatively different results for data sets where the mean is not equal to 0, as shown in figure 1.

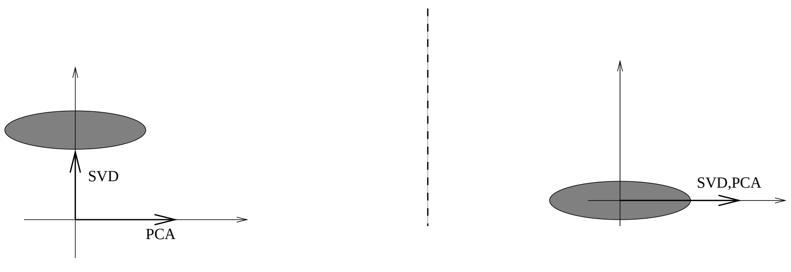

Figure 1: The leading singular vector may not be in the same direction as the principal component

2 Clustering Considerations

Clustering is the grouping together of similar objects. Usually, the clustering problem is posed as an optimization problem in which a given objective function needs to be optimized. A good objective function is critical for coming up with a good solution and must match the user's subjective criterion. Coming up with a good clustering objective function requires an understanding of the nature of the different possible objective functions and also the type of clusters produced by each of them. Invariances of the data can also help in choosing an objective function.

For example, the sum of squares criterion is invariant to translations and rotations, however the *Manhattan metric* as shown in figure 2 is not.

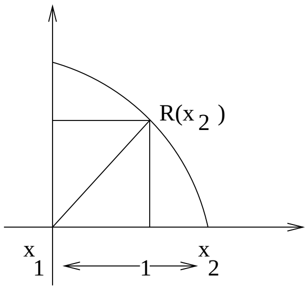

Figure 2: The manhattan metric is not invariant to rotations.

Another decision that one must make, is whether the input data can be scaled or normalized.

While trying to locate an extremum of the objective function, there is a tradeoff between time and accuracy since finding the global extremum of most clustering objectives is NP-hard. For example, the sum-of-squares criterion in (1) is NP-hard to solve even for $k = 2$.

3 Hierarchical Agglomerative Clustering

Hierarchical clustering involves constructing a tree to capture the clustering of objects. It can either be *agglomerative* (bottom up) or *divisive* (top down).

The basic idea in *Hierarchical Agglomerative Clustering* is to start out with n clusters, i.e., each sample is a separate cluster. At every step two clusters are merged till the required number of clusters is obtained:

Let $c = n$ and $C_i = \{\mathbf{x}_i\}, i = 1 \ldots n$.
While $c \geq k$,
  Find the "nearest" pair of distinct clusters C_i and C_j.
  Merge C_i and C_j, and delete C_j.
  $c \leftarrow c - 1$.
end while.

Different definitions of "nearest", i.e., of distances between clusters, lead to different clustering algorithms. Examples of distances between clusters are,

$$d_{min}(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{x}' \in C_j} \|\mathbf{x} - \mathbf{x}'\|,$$

$$\begin{aligned} d_{max}(C_i, C_j) &= \max_{\mathbf{x}\in C_i, \mathbf{x}'\in C_j} \|\mathbf{x}-\mathbf{x}'\|, \\ d_{avg}(C_i, C_j) &= \frac{1}{n_i n_j} \sum_{\mathbf{x}\in C_i} \sum_{\mathbf{x}'\in C_j} \|\mathbf{x}-\mathbf{x}'\|, \\ d_{mean}(C_i, C_j) &= \|\mathbf{m}_i - \mathbf{m}_j\|, \end{aligned}$$

where $\mathbf{m}_i$ is the mean of cluster C_i and $n_i = |C_i|$. The clusters produced by these different metrics can be very different. Using the d_{min} metric tends to favor elongated clusters, it is also referred to as a *single-linkage* algorithm. For $k = 1$, this is equivalent to *Kruskal's* minimum spanning tree algorithm. The complexity of this algorithm is $O(n^2)$.

Use of d_{max} leads to the farthest neighbor algorithm. With this criterion the formation of elongated clusters is discouraged. If the diameter of a cluster is defined as the largest distance between points in the cluster, d_{max} is the diameter of their union. The complexity of this algorithm is also $O(n^2)$.

Both the nearest neighbor and farthest neighbor clustering algorithms are greedy and satisfy local objective criteria. Instead, we can also use a global objective function, such as the one used in k-means clustering:

$$\mathcal{J} = \sum_{i=1}^{k} \sum_{\mathbf{x}\in C_i} \|\mathbf{x} - \mathbf{m}_i\|^2. \tag{1}$$

An agglomerative clustering algorithm that tries to minimize this objective function at each step is described below.

3.1 Stepwise-Optimal Hierarchical Clustering

Let $c = n$ and $C_i = \{\mathbf{x}_i\}$, $i = 1 \dots n$.
While $c \geq k$
    Find the pair of distance clusters, say C_i and C_j whose merger increases the objective function as little as possible.
    Merge C_i and C_j, and delete C_j.
    $c \leftarrow c - 1$.
endwhile.

Let $J_i = \sum_{\mathbf{x}\in C_i} \|\mathbf{x} - \mathbf{m}_i\|^2$ and $J_k = \sum_{\mathbf{x}\in C_k} \|\mathbf{x} - \mathbf{m}_k\|^2$. Merging C_i and C_k results in the new mean $\hat{\mathbf{m}}_i$:

$$\hat{\mathbf{m}}_i = \frac{n_i \mathbf{m}_i + n_k \mathbf{m}_k}{n_i + n_k} = \mathbf{m}_i + \frac{n_k(\mathbf{m}_k - \mathbf{m}_i)}{n_i + n_k} = \mathbf{m}_k + \frac{n_i(\mathbf{m}_i - \mathbf{m}_k)}{n_i + n_k}.$$

Thus the increased objective function value (see (1)) is:

$$\begin{aligned} & \sum_{\mathbf{x}\in C_i} \|\mathbf{x} - \hat{\mathbf{m}}_i\|^2 + \sum_{\mathbf{x}\in C_k} \|\mathbf{x} - \hat{\mathbf{m}}_k\|^2 \\ = & \sum_{\mathbf{x}\in C_i} \|\mathbf{x} - \mathbf{m}_i - \frac{n_k(\mathbf{m}_k - \mathbf{m}_i)}{n_i + n_k}\|^2 + \sum_{\mathbf{x}\in C_k} \|\mathbf{x} - \mathbf{m}_k - \frac{n_i(\mathbf{m}_i - \mathbf{m}_k)}{n_i + n_k}\|^2 \\ = & J_i + \frac{n_k^2 n_i}{(n_i + n_k)^2}\|\mathbf{m}_k - \mathbf{m}_i\|^2 + J_k + \frac{n_i^2 n_k}{(n_i + n_k)^2}\|\mathbf{m}_i - \mathbf{m}_k\|^2 \quad \text{(because } \textstyle\sum_{\mathbf{x}\in C_i} \mathbf{x} - \mathbf{m}_i = 0) \\ = & (J_i + J_k) + \frac{n_i n_k}{n_i + n_k}\|\mathbf{m}_i - \mathbf{m}_k\|^2. \end{aligned}$$

Hence among all pairs C_i and C_k, chose the pair such that:

$$d_{opt}(C_i, C_j) = \sqrt{\frac{n_i n_k}{n_i + n_k}} \|\mathbf{m}_i - \mathbf{m}_k\|$$

is minimized.

4 k-Means Clustering (Divisive Clustering)

The goal is to produce k clusters such that the objective function in (1) is minimized.

1. Start with some partitioning of the data into k clusters: $C_1^{(0)}, C_2^{(0)}, \ldots, C_k^{(0)}$. Set iteration count $t \leftarrow 0$.
2. Update the mean of each cluster based on the current data partitioning:

$$\mathbf{m}_i^{(t)} = \frac{1}{n_i^{(t)}} \sum_{\mathbf{x} \in C_i^{(t)}} \mathbf{x} \tag{2}$$

3. Update partitions based on the new cluster means as follows. For each $\mathbf{x}$, find the mean vector $\mathbf{m}_i^{(t)}$ nearest to it. If the closest mean vector is $\mathbf{m}_j^{(t)}$, then put $\mathbf{x}$ in $C_j^{(t+1)}$. Thus:

$$C_j^{(t+1)} = \{\mathbf{x} : \|\mathbf{x} - \mathbf{m}_j^{(t)}\| \leq \|\mathbf{x} - \mathbf{m}_i^{(t)}\|, i = 1, 2, \ldots, k\} \tag{3}$$

   Note that ties (where $\mathbf{x}$ is equidistant to two $\mathbf{m}_i^{(t)}$) can be broken arbitrarily.

4. If not "converged", $t \leftarrow t + 1$ and go to step 2. Otherwise, stop.

4.1 Properties of k-means

Theorem: The k-means objective function value never increases

$$\begin{aligned}
J^{(t)} &= \sum_i \sum_{\mathbf{x} \epsilon C_i^{(t)}} \|\mathbf{x} - \mathbf{m}_i^{(t)}\|^2 \\
\text{Let } \|\mathbf{x} - \mathbf{m}_j^{(t)}\| &= \min_l \|\mathbf{x} - \mathbf{m}_l^{(t)}\|. \\
\text{Then } J^{(t)} &\geq \sum_i \sum_{\mathbf{x} \epsilon C_i^{(t)}} \|\mathbf{x} - \mathbf{m}_j^{(t)}\|^2 \\
&\geq \sum_j \sum_{\mathbf{x} \epsilon C_i^{(t+1)}} \|\mathbf{x} - \mathbf{m}_j^{(t+1)}\|^2 \\
&= J^{(t+1)}.
\end{aligned}$$

The k-means objective function and the algorithm results in the data points being partitioned by hyperplanes. For example consider the case where k = 2.

$$\mathbf{x} \in C_1 \quad \text{iff} \quad \|\mathbf{x} - \mathbf{m}_1\|^2 \leq \|\mathbf{x} - \mathbf{m}_2\|^2$$

The locus of points equidistant from $\mathbf{m}_1$ and $\mathbf{m}_2$ (i.e., the line perpendicular to and passing through the mid point of the line joining $\mathbf{m}_1$ and $\mathbf{m}_2$) partitions the data points in such a fashion.

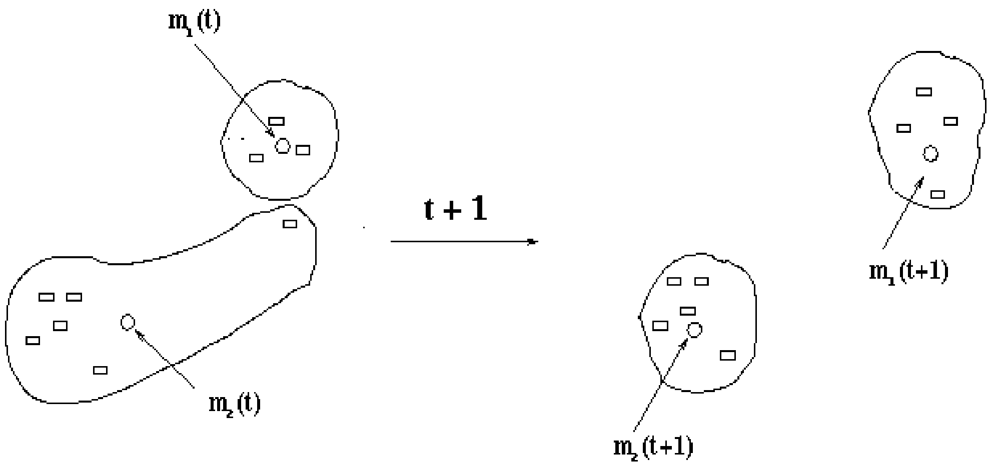

Figure 3: Iteration *t+1*

In d-dimensional space, this locus is the hyperplane normal to $\mathbf{m}_1$ - $\mathbf{m}_2$ passing midway through the line joining $\mathbf{m}_1$ - $\mathbf{m}_2$:

$$\begin{aligned} (\mathbf{x}-\mathbf{m}_1)^T(\mathbf{x}-\mathbf{m}_1) &= (\mathbf{x}-\mathbf{m}_2)^T(\mathbf{x}-\mathbf{m}_2), \\ \Rightarrow\ 2(\mathbf{m}_1-\mathbf{m}_2)^T x + (\mathbf{m}_2^T\mathbf{m}_2 - \mathbf{m}_1^T\mathbf{m}_1) &= 0. \end{aligned}$$

In general, when there are more than 2 clusters, k-means partitions the data set with a collection of hyperplanes. The resulting partitions correspond to *Voronoi regions.*

Duality of the k-means objective function

The k-means objective function can be thought of as $Trace(S_W)$ where

$$S_W = \sum_i \sum_{\mathbf{x}\epsilon C_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^T .$$

The matrix S_W is called the within-class scatter matrix.

Note: The trace of positive semi-definite matrices defines a norm on such matrices. Also $Trace(\sum \mathbf{x}\mathbf{x}^T) = \sum \mathbf{x}^T\mathbf{x}$.

The total covariance matrix is given by

$$S_T = \sum_{\mathbf{x}} (\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^T,$$

where $\mathbf{m}$ is the mean of the entire data set. S_T is also known as the total scatter matrix. It is possible to prove that $S_T = S_W + S_B$ where S_B is the between-class scatter matrix defined as follows:

$$S_B = \sum_i n_i(\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^T.$$

Since $Trace(S_T) = Trace(S_W) + Trace(S_B)$, and $Trace(S_T)$ is fixed we can view the k-means objective function as

$$\begin{aligned}\max\ Trace(S_W) &= \max \sum_i n_i \|\mathbf{m}_i - \mathbf{m}\|^2 \\ &= \max \frac{1}{2n} \sum_i \sum_j n_i n_j \|\mathbf{m}_i - \mathbf{m}_j\|^2.\end{aligned}$$

i.e., k-means can be thought of as maximizing the average pairwise squared distance between the mean vectors.

Proof that $S_T = S_W + S_B$

$$\begin{aligned}\text{Define}\quad S_T(\mathbf{a}) &= \sum_{\mathbf{x}} (\mathbf{x}-\mathbf{a})(\mathbf{x}-\mathbf{a})^T, \qquad \text{where } \mathbf{a} \text{ is any vector.} \\ &= \sum_i \sum_{\mathbf{x}\epsilon C_i} (\mathbf{x}-\mathbf{a}+\mathbf{m}_i-\mathbf{m}_i)(\mathbf{x}-\mathbf{a}+\mathbf{m}_i-\mathbf{m}_i)^T \\ &= \sum_i \sum_{\mathbf{x}\epsilon C_i} ((\mathbf{x}-\mathbf{m}_i)+(\mathbf{m}_i-\mathbf{a}))((\mathbf{x}-\mathbf{m}_i)^T+(\mathbf{m}_i-\mathbf{a})^T) \\ &= \sum_i \sum_{\mathbf{x}\epsilon C_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^T + \sum_i \sum_{\mathbf{x}\epsilon C_i} (\mathbf{m}_i-\mathbf{a})(\mathbf{x}-\mathbf{m}_i)^T \\ &\quad + \sum_i \sum_{\mathbf{x}\epsilon C_i} (\mathbf{m}_i-\mathbf{a})(\mathbf{x}-\mathbf{m}_i)^T + \sum_i \sum_{\mathbf{x}\epsilon C_i} (\mathbf{m}_i-\mathbf{a})(\mathbf{m}_i-\mathbf{a})^T \\ S_T(\mathbf{a}) &= \sum_i \sum_{\mathbf{x}\epsilon C_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^T + \sum_i n_i (\mathbf{m}_i-\mathbf{a})(\mathbf{m}_i-\mathbf{a})^T, \\ &\quad \text{since the second and third terms in the previous equation are zero.} \\ &= S_W + S_B(\mathbf{a})\end{aligned}$$

The above equation is true for all vectors $\mathbf{a}$; in particular for $\mathbf{a} = \mathbf{m}$, which proves the result.

We can also show that

$$\begin{aligned} S_T &= \frac{1}{2n} \sum_i \sum_j (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T, \\ S_W &= \sum_i \frac{1}{2n_i} \sum_{\mathbf{x}\in C_i} \sum_{\mathbf{y}\in C_i} (\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^T, \\ \text{and}\quad S_B &= \frac{1}{2n} \sum_i \sum_j n_i n_j (\mathbf{m}_i - \mathbf{m}_j)(\mathbf{m}_i - \mathbf{m}_j)^T. \end{aligned}$$

Since $Trace(S_W) =$ k-means objective function, k-means minimizes the average pairwise sum-of-squares of distances between points within clusters.

Proof that $S_T = \frac{1}{2n} \sum_i \sum_j (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$

$$\begin{aligned} S_T &= \sum_{\mathbf{x}} (\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^T \\ &= \sum_{\mathbf{x}} \mathbf{x}\mathbf{x}^T - \sum_{\mathbf{x}} (\mathbf{m}\mathbf{x}^T + \mathbf{x}\mathbf{m}^T) + n\mathbf{m}\mathbf{m}^T = \sum_{\mathbf{x}} \mathbf{x}\mathbf{x}^T - n\mathbf{m}\mathbf{m}^T \end{aligned}$$

$$
\begin{aligned}
\text{Pairwise scatter} \quad S_P &= \sum_i \sum_j (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \\
&= \sum_i \sum_j (\mathbf{x}_i\mathbf{x}_i^T - \mathbf{x}_j\mathbf{x}_i^T - \mathbf{x}_i\mathbf{x}_j^T + \mathbf{x}_j\mathbf{x}_j^T) \\
&= 2n \sum_{\mathbf{x}} \mathbf{x}\mathbf{x}^T - \sum_j \mathbf{x}_j \left(\sum_i \mathbf{x}_i{}^T \right) - \sum_i \mathbf{x}_i \left(\sum_j \mathbf{x}_j^T \right) \\
&= 2n \sum_{\mathbf{x}} \mathbf{x}\mathbf{x}^T - \sum_j n\mathbf{x}_j\mathbf{m}^T - \sum_i n\mathbf{x}_i\mathbf{m}^T \\
&= 2n \sum_{\mathbf{x}} \mathbf{x}\mathbf{x}^T - 2n^2\mathbf{m}\mathbf{m}^T = 2nS_T \\
\Rightarrow \quad S_T &= \frac{1}{2n} \cdot S_P
\end{aligned}
$$

5 Demo on different Clustering techniques

In this section, some of the results of various agglomerative clustering algorithms and k-means clustering algorithms on 2-dimensional toy data sets are shown to highlight the properties of different clustering algorithms.

Figures 4 through 6 show the results of three agglomerative clustering algorithms on three data sets from Duda and Hart [1, p. 231]. For all these data sets, the clusters are obvious when data points are examined "by eye". Note how the nearest neighbor clustering technique tends to make elongated chain like clusters while the farthest neighbor and mean clustering tend to form spherical clusters.

Figures 7 through 9 show the results of k-means clustering algorithm on three data sets. Each figure shows four different runs of k-means with different initializations. Note how sensitive k-means is to these initializations. It generally converges to a local minima that is "near" to the initial partitioning.

Thus, for good clustering the key is to first figure out the characteristics of your data sets and then apply the clustering technique that is appropriate for the data set. There is no universal clustering algorithm that is best for every data set.

References

[1] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

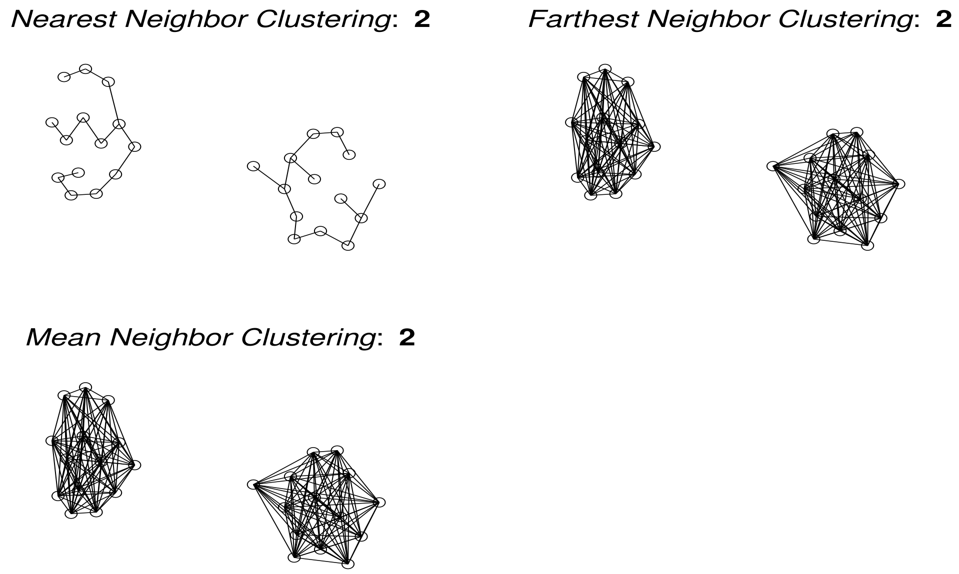

Figure 4: Data set 1

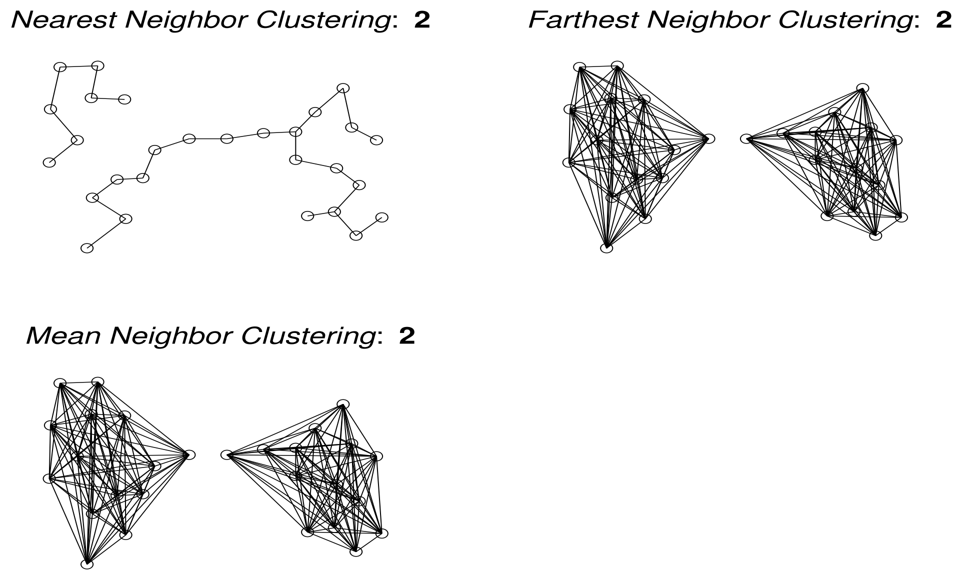

Figure 5: Data set 2: Nearest neighbor tends to make chain like clusters, while farthest neighbor and mean form spherical clusters

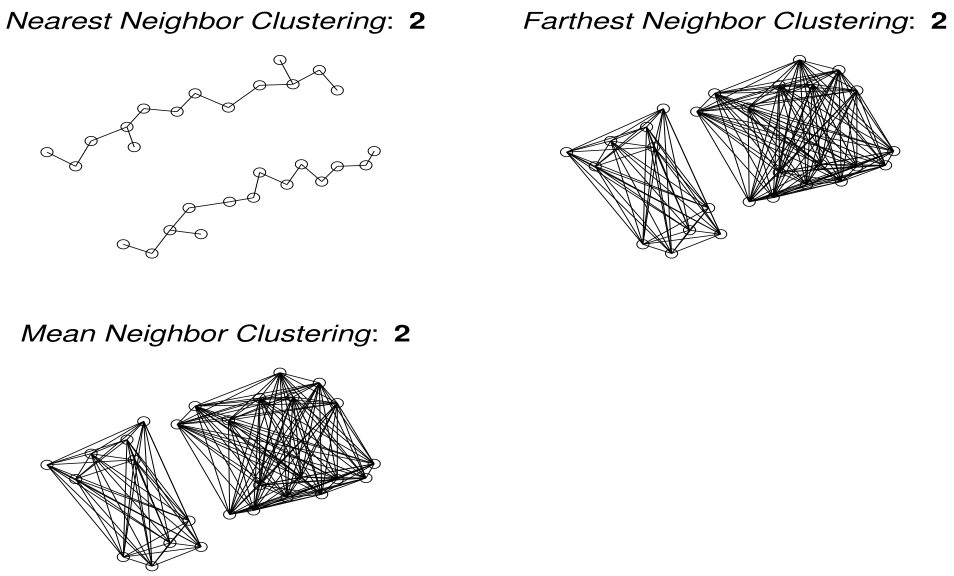

Figure 6: Data set 3

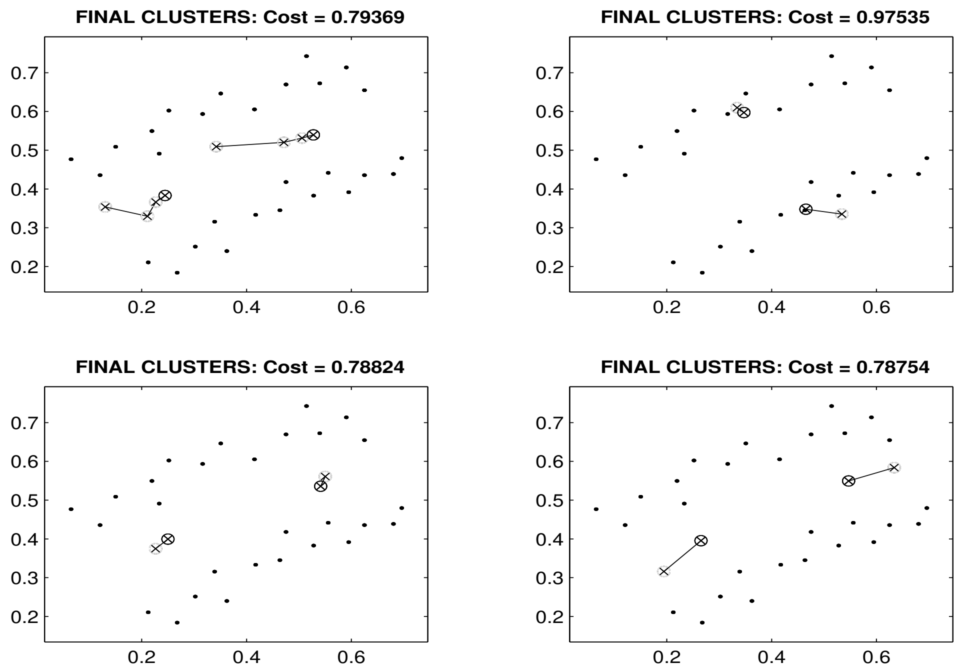

Figure 7: Data set 1: Four runs of the k-means algorithm. Two local minima are possible depending on initial clusters.

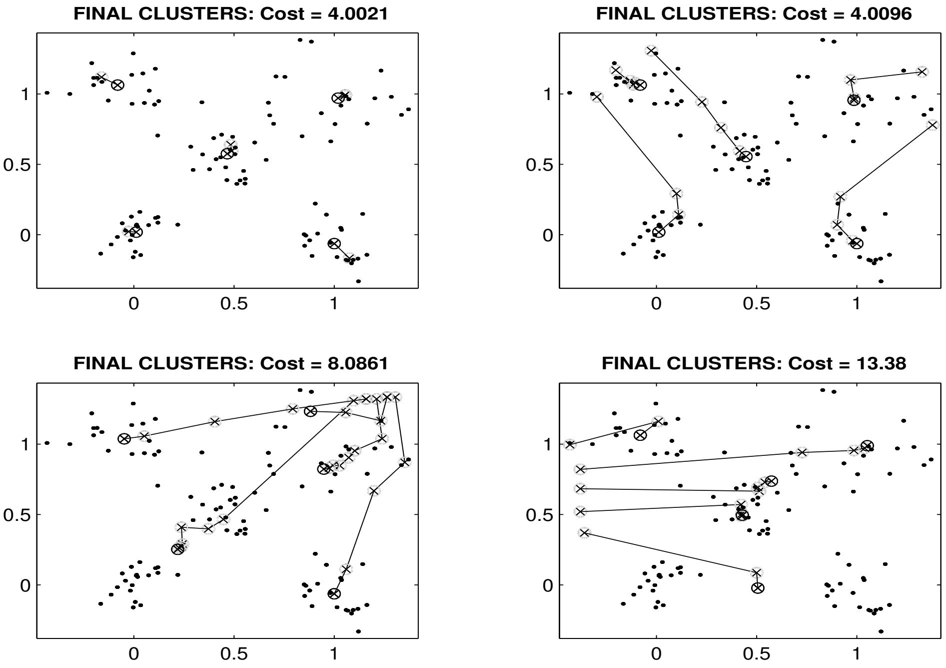

Figure 8: Data set 4: Data has 5 clusters. Different k-means results based on initialization.

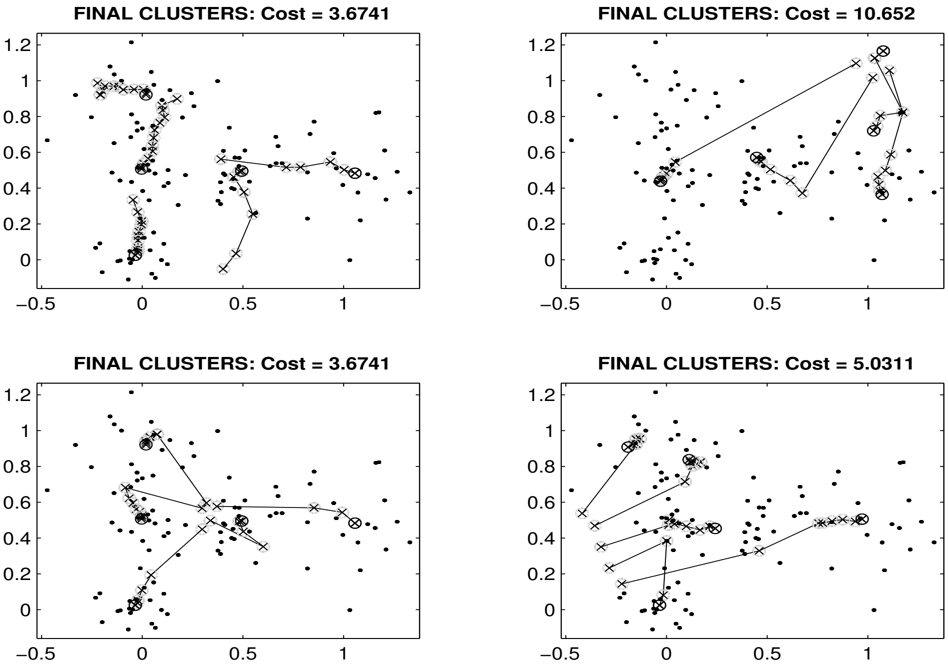

Figure 9: Data set 5: Data has 5 clusters with less obvious cluster boundaries. Again different results based on initialization.