LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Database Systems Group • Prof. Dr. Thomas Seidl

Exercise 6: Clustering

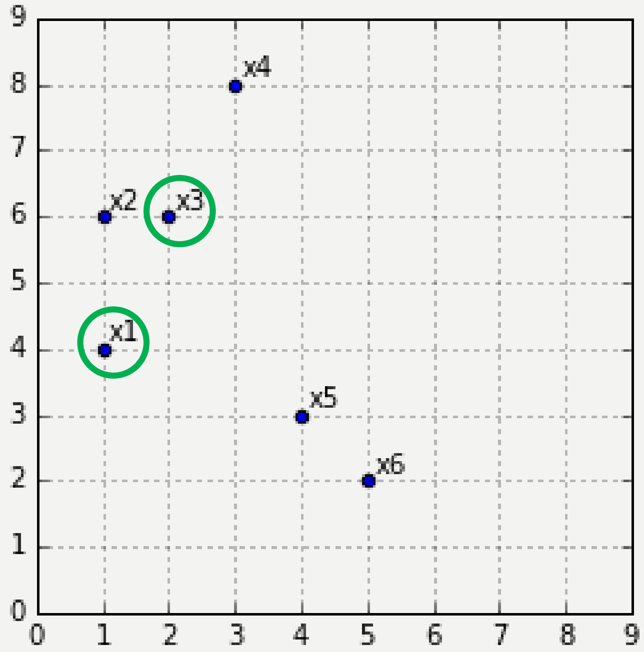
Knowledge Discovery in Databases I

SS 2016

Exercise 6-1 (a): K-Medoid (PAM)

Consider the following 2-dimensional data set.

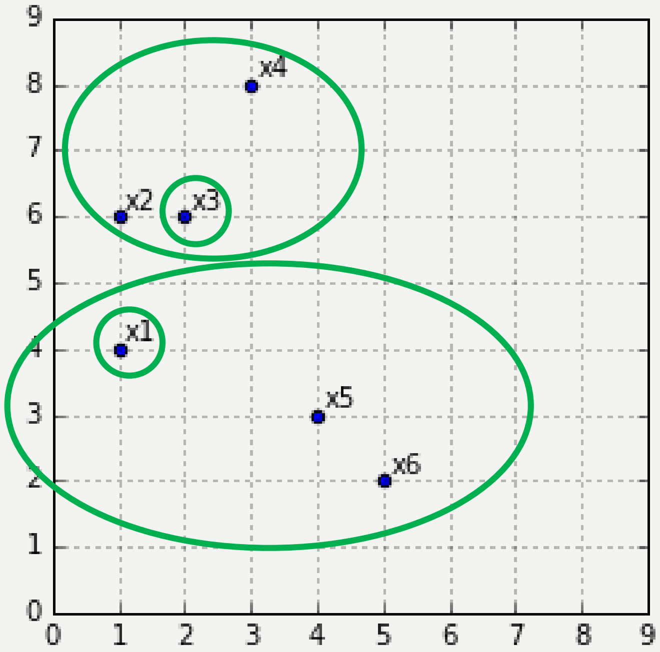
a) Perform the first loop of the PAM algorithm (k=2) using the Euclidian distance. Select x_1 and x_3 as initial medoids and compute the resulting medoids and clusters.



Exercise 6-1 (a): K-Medoid (PAM)

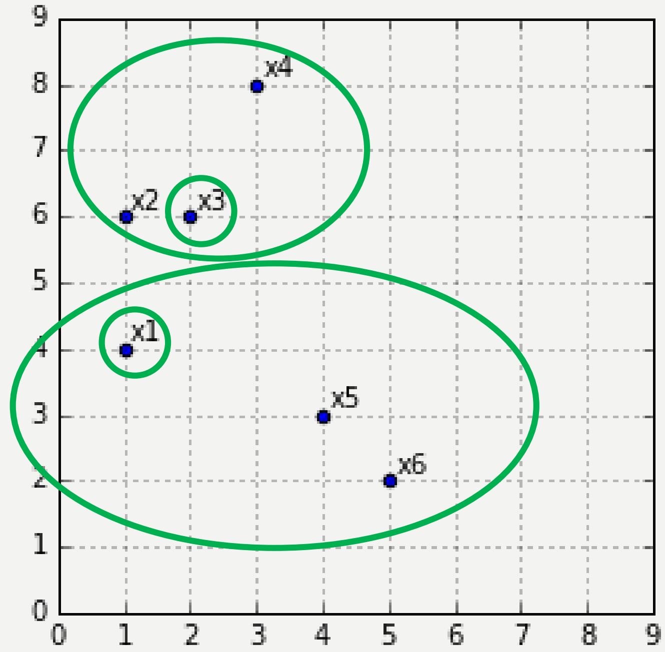
- Compute compactness of the initial clustering:

$$TD = d(x_3, x_2) + d(x_3, x_4) + d(x_1, x_5) + d(x_1, x_6)$$
$$= 1 + 3\sqrt{5} + \sqrt{10}$$



Exercise 6-1 (a): K-Medoid (PAM)

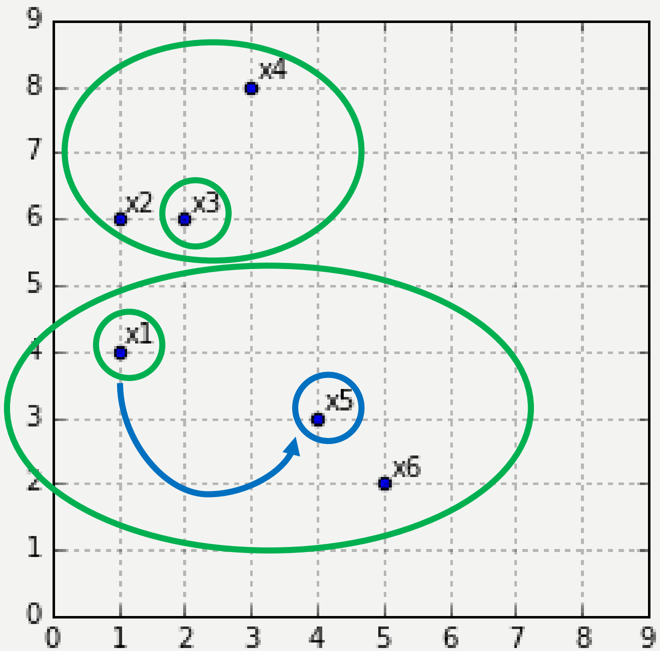
- For each pair of medoid and non-medoid (M,N): Compute compactness if swapped ($TD_{M\leftrightarrow N}$)
- Find the pair for which $TD_{M\leftrightarrow N}$ is minimal and perform the swap if $TD_{M\leftrightarrow N} < TD$

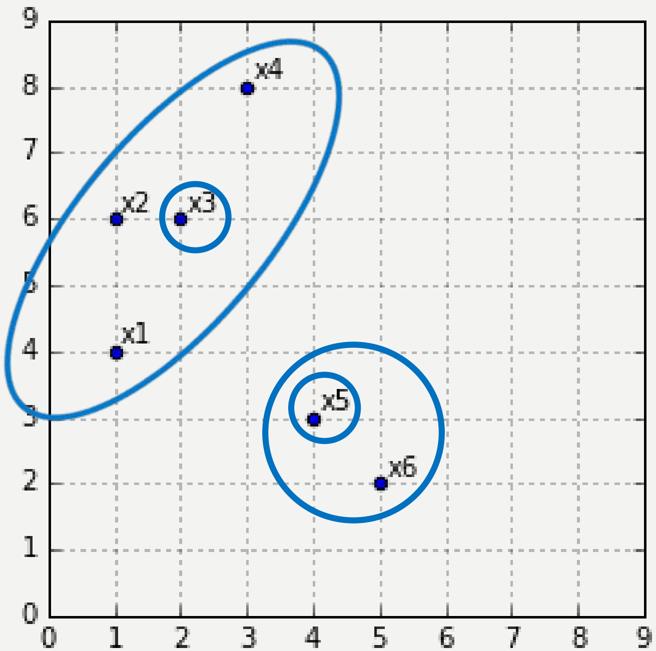


Exercise 6-1 (a): K-Medoid (PAM)

- In this case: Swap x_1 with either x_5 or x_6:

$$TD_{x_1 \leftrightarrow x_5} = TD_{x_1 \leftrightarrow x_6}$$
$$= d(x_3, x_2) + d(x_3, x_4) + d(x_3, x_1) + d(x_5, x_6)$$
$$= 1 + 2\sqrt{5} + \sqrt{2} < T$$





Exercise 6-1 (b): K-Medoid (PAM)

b) How can the clustering result

$$C_1 = \{x_1, x_5, x_6\}, C_2 = \{x_2, x_3, x_4\}$$

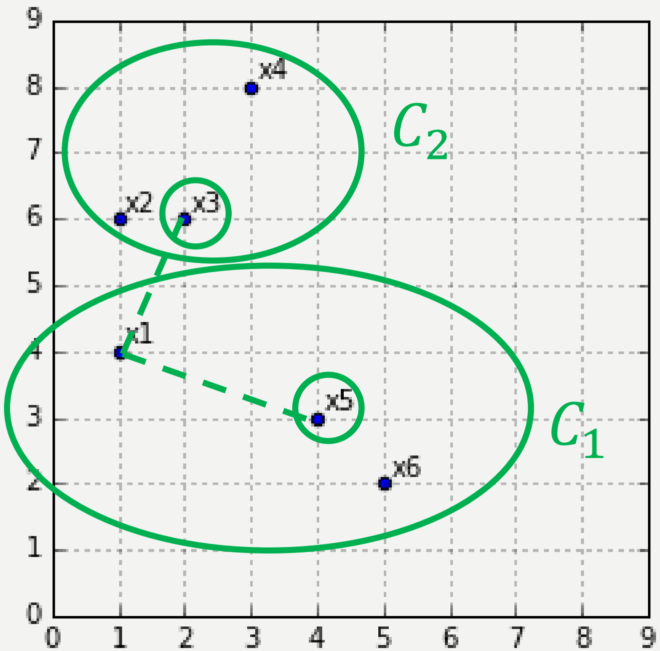
be obtained with the PAM algorithm (k=2) using the weighted Manhattan distance

$$d(x, y) = w_1 \cdot |x_1 - y_1| + w_2 \cdot |x_2 - y_2|?$$

Assume that x_1 and x_3 are the initial medoids and give values for the weights w_1 and w_2 for the first and second dimension respectively.

Exercise 6-1 (b): K-Medoid (PAM)

- For the desired clusters C_1 and C_2, the medoids would be x_5 and x_3, respectively
- Under the standard Euclidean or Manhattan Distance, x_1 would be assigned to medoid x_3 rather than x_5



- Idea: Attach a higher weight to the second dimension, such that

$$
\begin{aligned}
& d(x_1, x_3) > d(x_1, x_5) \\
\Leftrightarrow\ & w_1|1-2| + w_2|4-6| > w_1|1-4| + w_2|4-3| \\
\Leftrightarrow\ & w_1 + 2w_2 > 3w_1 + w_2 \\
\Leftrightarrow\ & w_2 > 2w_1
\end{aligned}
$$

- That is, if we set w_2 more than twice as large than w_1 (e.g. $w_1 = 0.3, w_2 = 0.7$), then PAM will return the desired clustering

Construct a low dimensional data set D together with a clustering $\{C_1, C_2\}$ computed by k-means with the following property:

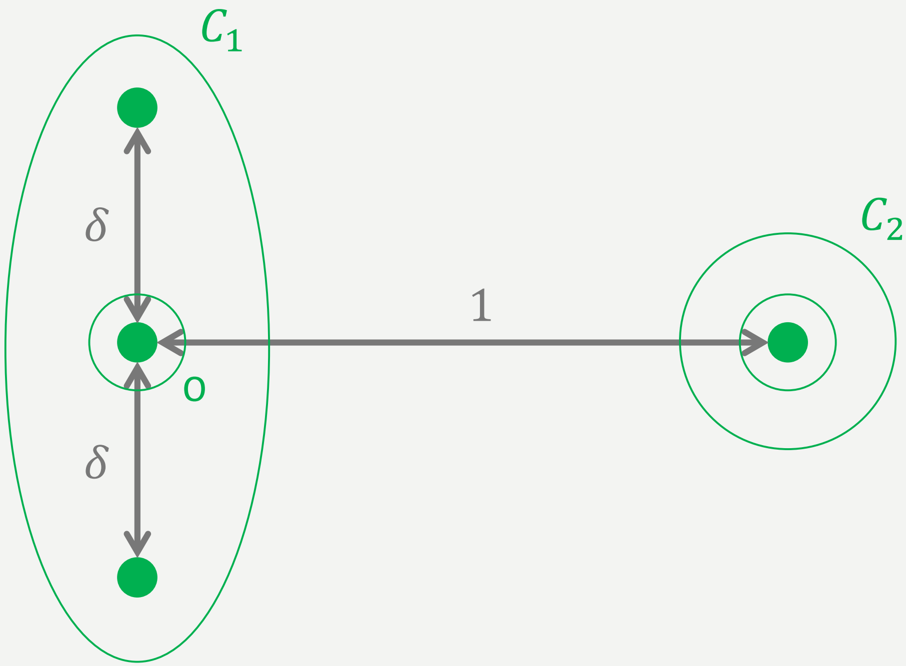
There exists an object $o \in D$ with a negative silhouette coefficient $s(o) < 0$.

Provide the means of the clusters and compute the silhouette coefficient for the corresponding object o.

Our example is based on the following ideas:

- All objects need to be closer to the centroid of their own cluster than to centroids of other clusters. Otherwise, k-means would not have terminated.
- In two dimensions, we can symmetrically expand a cluster in one dimension to make the point distances within the cluster arbitrarily large, but without changing the centroid of the cluster or getting too close to a different cluster
- We illustrate this with four points

- On the following 4-point dataset with the initial centroids as indicated, k-means would produce the clusters C_1 and C_2 (irrespective of the value of δ):



- We have:

$$a(o) = \frac{2}{3}\delta, \qquad b(o) = 1$$

- If we choose δ large enough, we can achieve that

$$a(o) > b(o)$$

- In particular, we will get

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}} = \frac{1 - \frac{2}{3}\delta}{\frac{2}{3}\delta} = \frac{3}{2\delta} - 1$$

and thus

$$s(o) \to -1 \text{ for } \delta \to \infty$$

i.e. we can make the silhouette of o arbitrarily bad.

- Note: Unfortunately, the example provided by a student in the Friday Group does *not* work. This is due to the normalization of the *a*-value (divide by the number of points in the cluster, rather than by the number of summed distances). I apologize, that I did not notice this during the exercise session. In fact, this issue was exactly the motivation for us to „blow up“ the cluster of *o*.

The solution to Exercise 6-3 will be provided as a *jupyter* notebook.