Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Thomas Seidl
Janina Sontheim, Maximilian Hünemörder

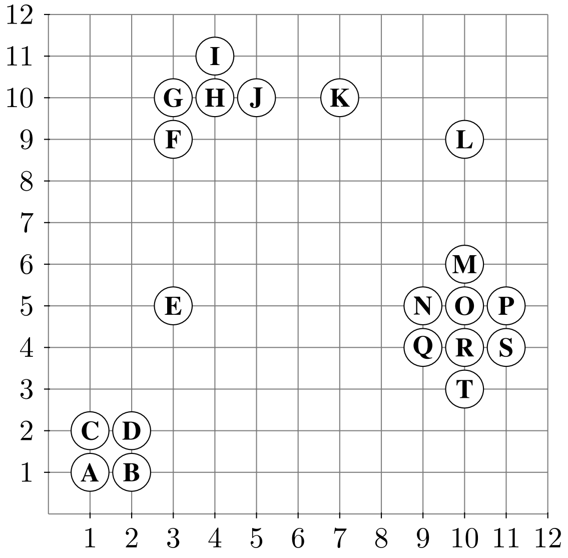
Knowledge Discovery in Databases

WS 2019/20

Exercise 7: DBSCAN, Spectral Clustering

Exercise 7-1 DBSCAN

Given the following data set:



As distance function, use Manhattan Distance:

$$L_1(x, y) = |x_1 - y_1| + |x_2 - y_2|$$

Compute DBSCAN and indicate which points are core points, border points and noise points.

Use the following parameter settings:

- Radius $\varepsilon = 1.1$ and *minPts* $= 2$
- Radius $\varepsilon = 1.1$ and *minPts* $= 3$
- Radius $\varepsilon = 1.1$ and *minPts* $= 4$
- Radius $\varepsilon = 2.1$ and *minPts* $= 4$
- Radius $\varepsilon = 4.1$ and *minPts* $= 5$
- Radius $\varepsilon = 4.1$ and *minPts* $= 4$

See tutorial slides.

Exercise 7-2 Properties of DBSCAN

Discuss the following questions/propositions about DBSCAN:

- Using *minPts* $= 2$, what happens to the border points?

  There are no border points: A border point must be a core points itself, since there must be at least one further object in its ϵ-neighborhood from which it is directly density reachable (otherwise it would not be connected to a cluster).

- The result of DBSCAN is deterministic w.r.t. the core and noise points but not w.r.t. the border points.

  If a border point is density-reachable from two clusters, it depends on the processing order and implementation, to which cluster it will be assigned.

- A cluster found by DBSCAN cannot consist of less than *minPts* points.

  Depends on the above case. It can happen that a border point will be assigned to another cluster, resulting in a cluster with less than $minPts$ points.

- If the dataset consists of n objects, DBSCAN will evaluate exactly n ϵ-range queries.

  Correct. This is not completely obvious from the pseudo-code presented in the lecture, but from each object, a single range query is executed to determine whether the object is a core object or not. If it is not a core object, it is a border object if it was discovered in a recursive call from another core object. Else, it is classified as a noise object until it is discovered from another core point, in which case it will be classified as a border object. In total, a chain of exactly one range query per object is performed.

  Therefore, a naive implementation will require $\mathcal{O}(n^2)$ time, since evaluating a range query with a sequential scan takes time $\mathcal{O}(n)$. Index-accelerated implementations typically runs in $\mathcal{O}(n \log n)$, since appropriate index structures are able to answer a range query in time $\mathcal{O}(\log n)$.

- On uniformly distributed data, DBSCAN will usually either assign all points to a single cluster or classify every point as noise. k-means on the other hand will partition the data into approximately equally sized partitions.

  Correct. Depending on the density threshold, DBSCAN will classify either all objects or no object as core objects. (By choosing e.g. $\varepsilon = \min_{o \in D}$ 10-dist(o) and *minPts* $= 10$, it can be provoked that a few single core points will be detected. However, finding such unfavorable parametrizations becomes more difficult with increasing dataset size.)
  For k-means on the other hand, solutions are (locally) optimal if all clusters are almost equally sized (at least if $k \cdot d \ll n$).
  However, solutions found in multiple k-means runs can be quite different.

Exercise 7-3 Spectral Clustering

(a) Given the dataset from Exercise 7-1, apply spectral clustering to the first ten points (i.e. A - J). When constructing the graph, make sure that each point is connected to its neighbours in an eps = 2 neighbourhood while still having at least two outgoing edges.

See tutorial slides.

(b) As shown in the lecture, spectral clustering uses the Laplacian matrix to determine its clusters. Given an arbitrary graph G and the Laplacian L for G, show that finding an indicator vector f_C that minimizes $\mathrm{f}L\mathrm{f}^T$ leads to an optimal cluster C in G, where

$$\mathrm{f}_C{}^{(i)} = \begin{cases} 1 & \text{if } v_i \in C \\ 0 & \text{else} \end{cases} \tag{1}$$

$$\begin{aligned}
fLf^T &= fDf^T - fWf^T \\
&= \sum_i d_i f_i^2 - \sum_{ij} w_{ij} f_i f_j \\
&= \frac{1}{2}\left(\sum_i (\sum_j w_{ij}) f_i^2 - 2\sum_{ij} w_{ij} f_i f_j + \sum_j (\sum_i w_{ij}) f_j^2\right) \\
&= \frac{1}{2}\left(\sum_{ij} w_{ij} f_i^2 - 2\sum_{ij} w_{ij} f_i f_j + \sum_{ij} w_{ij} f_j^2\right) \\
&= \frac{1}{2}\sum_{ij} w_{ij}(f_i^2 - 2f_i f_j + f_j^2) \\
&= \frac{1}{2}\sum_{ij} w_{ij}(f_i - f_j)^2
\end{aligned}$$