University of Southern Denmark
IMADA

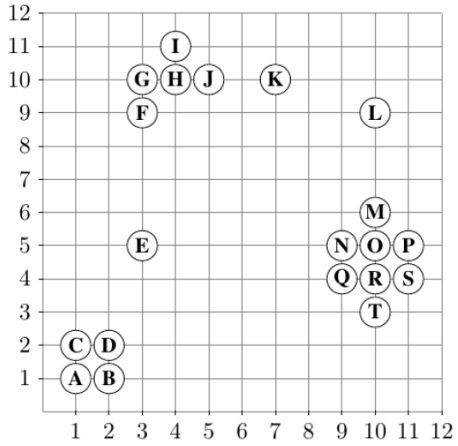
DM566: Data Mining and Machine Learning

Spring term 2022

Exercise 8

Exercise 8-1 Density Estimation

Given the following dataset:



Estimate the density around each point in the dataset, using the discrete kernel:

$$\hat{f}(x) = \frac{k}{nV_k(x)}$$

based on the Manhattan distance (L_1)

1. with fixed $k = 2$
2. with fixed $k = 4$
3. with fixed volume based on radius $\varepsilon = 1$
4. with fixed volume based on radius $\varepsilon = 2$

Explain what your choices are in computing the density, regarding

1. Including or excluding the point itself
2. Ties in the neighbourhood

Note that using the Manhattan distance results in estimators that slightly differ from those discussed in the lecture. What do you observe?

Suggested solution:
In the density estimate, it makes more sense to include the point itself, as it contributes to the density at that specific spot. We have $n = 20$ objects. $V_k(x)$ is the area of k-distance radius around x with respect to Manhattan distance. The volume for Manhattan distance ranges with radius r are given by $2 \cdot r^2$.

We calculate the density estimates for $k = 2$ the following way:
We set the k in the formula to 3, as we include the point itself, as it contributes to the density in that specific spot. We set $n = 20$. As $k = 2$, we need to look for the distance to the 2nd nearest neighbour, when calculating $V_2(x)$. We then want to find the volume for the Manhattan distance range. For example with point E, we see that the 2nd nearest neighbour has distance 4 to E. The volume for the distance range is then $2 \cdot 4^2 = 32$. Inserting this in to the formula, we calculate $\frac{3}{20 \cdot 32} = 0.0046875$.

For $k = 4$ we follow the same method as with $k = 2$.

We calculate the estimates for radius $\varepsilon = 1$ in the following way:
We still know that $n = 20$. We calculate $V_k(x)$ with the fixed value $\varepsilon = 1$, such that $V_k(x) = 2 \cdot \varepsilon^2 = 2$. We set k to be the amount of points within the range of radius $\varepsilon = 1$, including the point itself. For example with point A, we see that within the range of radius $\varepsilon = 1$, there are 3 points, including A. Inserting this in to the formula, we calculate $\frac{3}{20 \cdot 2} = 0.075$.

For $\varepsilon = 2$ we follow the same method as with $\varepsilon = 1$.

See the calculations on the next page.

	$k=2$	$k=4$	$\varepsilon=1$	$\varepsilon=2$
A	$\frac{3}{20\cdot 2}=0.075$	$\frac{5}{20\cdot 72}=0.003472$	$\frac{3}{20\cdot 2}=0.075$	$\frac{4}{20\cdot 8}=0.025$
B	$\frac{3}{20\cdot 2}=0.075$	$\frac{5}{20\cdot 50}=0.005$	$\frac{3}{20\cdot 2}=0.075$	$\frac{4}{20\cdot 8}=0.025$
C	$\frac{3}{20\cdot 2}=0.075$	$\frac{5}{20\cdot 50}=0.005$	$\frac{3}{20\cdot 2}=0.075$	$\frac{4}{20\cdot 8}=0.025$
D	$\frac{3}{20\cdot 2}=0.075$	$\frac{5}{20\cdot 32}=0.0078125$	$\frac{3}{20\cdot 2}=0.075$	$\frac{4}{20\cdot 8}=0.025$
E	$\frac{3}{20\cdot 32}=0.0046875$	$\frac{6}{20\cdot 50}=0.006$	$\frac{1}{20\cdot 2}=0.025$	$\frac{1}{20\cdot 8}=0.00625$
F	$\frac{3}{20\cdot 8}=0.01875$	$\frac{5}{20\cdot 18}=0.013889$	$\frac{2}{20\cdot 2}=0.05$	$\frac{3}{20\cdot 8}=0.01875$
G	$\frac{3}{20\cdot 2}=0.075$	$\frac{5}{20\cdot 8}=0.03125$	$\frac{3}{20\cdot 2}=0.075$	$\frac{5}{20\cdot 8}=0.03125$
H	$\frac{4}{20\cdot 2}=0.1$	$\frac{5}{20\cdot 8}=0.03125$	$\frac{4}{20\cdot 2}=0.1$	$\frac{5}{20\cdot 8}=0.03125$
I	$\frac{4}{20\cdot 8}=0.025$	$\frac{5}{20\cdot 18}=0.013889$	$\frac{2}{20\cdot 2}=0.05$	$\frac{4}{20\cdot 8}=0.025$
J	$\frac{5}{20\cdot 8}=0.03125$	$\frac{5}{20\cdot 8}=0.03125$	$\frac{2}{20\cdot 2}=0.05$	$\frac{5}{20\cdot 8}=0.03125$
K	$\frac{3}{20\cdot 18}=0.0083$	$\frac{6}{20\cdot 32}=0.009375$	$\frac{1}{20\cdot 2}=0.025$	$\frac{2}{20\cdot 8}=0.0125$
L	$\frac{4}{20\cdot 32}=0.00625$	$\frac{7}{20\cdot 50}=0.007$	$\frac{1}{20\cdot 2}=0.025$	$\frac{1}{20\cdot 8}=0.00625$
M	$\frac{5}{20\cdot 8}=0.03125$	$\frac{5}{20\cdot 8}=0.03125$	$\frac{2}{20\cdot 2}=0.05$	$\frac{5}{20\cdot 8}=0.03125$
N	$\frac{3}{20\cdot 2}=0.075$	$\frac{6}{20\cdot 8}=0.0375$	$\frac{3}{20\cdot 2}=0.075$	$\frac{6}{20\cdot 8}=0.0375$
O	$\frac{5}{20\cdot 2}=0.125$	$\frac{5}{20\cdot 2}=0.125$	$\frac{5}{20\cdot 2}=0.125$	$\frac{8}{20\cdot 8}=0.05$
P	$\frac{3}{20\cdot 2}=0.075$	$\frac{6}{20\cdot 8}=0.0375$	$\frac{3}{20\cdot 2}=0.075$	$\frac{6}{20\cdot 8}=0.0375$
Q	$\frac{3}{20\cdot 2}=0.075$	$\frac{6}{20\cdot 8}=0.0375$	$\frac{3}{20\cdot 2}=0.075$	$\frac{6}{20\cdot 8}=0.0375$
R	$\frac{5}{20\cdot 2}=0.125$	$\frac{5}{20\cdot 2}=0.125$	$\frac{5}{20\cdot 2}=0.125$	$\frac{8}{20\cdot 8}=0.05$
S	$\frac{3}{20\cdot 2}=0.075$	$\frac{6}{20\cdot 8}=0.0375$	$\frac{3}{20\cdot 2}=0.075$	$\frac{6}{20\cdot 8}=0.0375$
T	$\frac{5}{20\cdot 8}=0.03125$	$\frac{5}{20\cdot 8}=0.03125$	$\frac{2}{20\cdot 2}=0.05$	$\frac{5}{20\cdot 8}=0.03125$

Some notable observations are:

- A larger value of k will differentiate between A, B, C and D, whereas smaller values will not.
- As expected, O and R have the highest density estimation among all k values and ε values.
- With the lowest values ($k=2$ and $\varepsilon=1$) H has the second highest density estimation. With the higher values ($k=4$ and $\varepsilon=2$), the points N, P, Q and S have higher density estimations than H.

Exercise 8-2 Properties of DBSCAN
Discussing the following questions on DBSCAN:

1. for $minPts = 2$, what about border points?

   Suggested solution:
   Recall the differences between core- and border points. The core points, as the name suggests, lie usually within the interior of a cluster. A border point has fewer than MinPts within its ε-neighborhood, but it lies in the neighborhood of another core point.
   There cannot be any border points for $minPts = 2$. Any point belonging to a cluster must be a core point, as it needs at least one other point in the neighborhood to connect!

2. The result of DBSCAN is deterministic for core and noise points, but not for border points.

   Suggested solution:
   If a border point is density-reachable from two clusters, it is order- and implementation-dependent to which cluster it will belong.
   However, if a border point is assigned to all clusters simultaneously that can reach it, the result is deterministic again.

3. A cluster can contain less than minPts objects.

   Suggested solution:
   Yes, if border points are reassigned to some other cluster. See example 5 in the lecture. If E is assigned to the green cluster, the red cluster has only 4 elements, although $minPts = 5$.

4. If the dataset has n objects, DBSCAN computes always exactly n neighbourhood range queries.

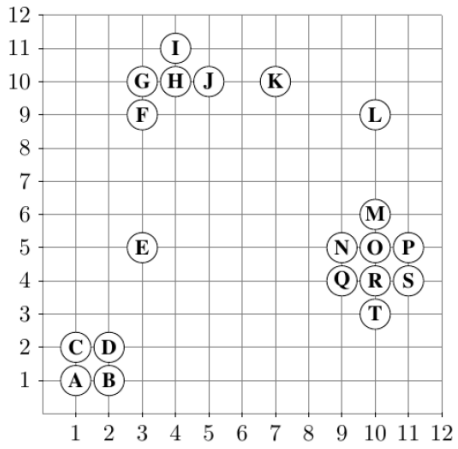
   Suggested solution:
   Correct. It is not quite obvious from the pseudo code but we have indeed 1 range query for each object; it will be labeled either before or afterwards. For labeled objects, we never have a second range query and the algorithm terminates when all objects are labeled.
   That means: a naïve implementation is in $O(n^2)$, as a single range query is in $O(n)$. An index-accelerated implementation can typically achieve $O(n \log n)$, as most index structures can answer a range query in $O(\log n)$.

5. On uniformly distributed data, DBSCAN will typically put everything in one cluster or everything in noise. k-means will typically partition the uniformly distributed data in k approximately equal-size partitions.

   Suggested solution:
   Correct. Depending on the density-threshold, DBSCAN will mark either almost all or almost no object as core point. (However, if you choose for example $\varepsilon = min_{o \in D} 10 - dist(o)$ and $minPts = 10$, you can provoke to get some core points. The choice of such unsuitable parameters will become more difficult with increasing dataset size.)
   For k-means, solutions on such datasets are (locally-) optimal if all clusters are approx. equally sized (at least if $k \cdot d << n$).
   Solutions of several runs of k-means can be very different, though, which can be a hint at questionable solutions.

Exercise 8-3 Shared Nearest Neighbours

Given the following dataset:



1. Compute the pairwise shared-nearest-neighbour-similarities SNN_5 of the objects M, N, O, P, Q, R, S, and T. Use Manhattan distance L_1 to obtain the neighbours and the size 5. The query point is a member of its neighbourhood.

   Suggested solution:

   Recall Shared Nearest Neighbours:

   $SNN_k(p, q) = |NN_k(p) \cap NN_k(q)|$

	p	$NN(p,5)$							
	M	M	N	O	P		R		
	N	M	N	O	P	Q	R		
	O	M	N	O	P		R		
Neighborhoods:	P	M	N	O	P		R	S	
	Q		N	O		Q	R	S	T
	R			O		Q	R	S	T
	S			O	P	Q	R	S	T
	T			O		Q	R	S	T

   Note: we have some ties and thus get neighborhoods larger than 5.

   Similarities based on the neighborhoods:

SNN_5	M	N	O	P	Q	R	S	T
M	5	5	5	5	3	2	3	2
N	5	6	5	5	4	3	4	3
O	5	5	5	5	3	2	3	2
P	5	5	5	6	4	3	4	3
Q	3	4	3	4	6	5	5	5
R	2	3	2	3	5	5	5	5
S	3	4	3	4	5	5	6	5
T	2	3	2	3	5	5	5	5

   The similarity measure lacks expressiveness to distinguish cases that have a complete overlap of their five neighbors (similarity of 5) and those that overlap in 5 neighbors but actually have 6 neighbors due to ties. How could we refine the similarity measure?

   We could normalize: $SNN_5(o, p) = \frac{|NN(o,5) \cap NN(p,5)|}{\max(|NN(0,5)|, |NN(p,5)|)}$

2. Give parameters ε and $minPts$ s.t. the SNN variant of DBSCAN identifies the 8 points as "dense" and connects them into a single cluster.

   Suggested solution:
   We observe that O and R have a similarity of only 2: they have each other in their 5-NN, but not other neighbors of each other. This is surprising as both are at the most dense area in Euclidean space. To connect them, we could therefore, e.g., choose a similarity-threshold $\varepsilon = 2$ and $minPts = 2$.

Clustering lab

The objective of this lab session is to familiarise yourself with clustering in practice.

1. Go to https://archive.ics.uci.edu/ml/datasets/seeds, where you can find the *seeds* dataset.

2. Preprocessing: The dataset is given in tab-separated format. Find out how to work with this data in Python, for instance using the `loadtxt` function in numpy. Which other preprocessing steps may we want to use?

3. k-means: Use a k-means implementation to analyse the data. What is a suitable choice for k? Try different variants, such as MacQueen, Lloyd, Elkan, k-means++. Do you observe any differences or tendencies in the results?

4. EM-clustering: Run EM-clustering on the seeds dataset. Compare to the results of k-means.

5. DBSCAN: Run DBSCAN on the dataset. How do you find suitable parameters for ε and $minPts$?

6. Compare the results. Which type of clustering algorithm do you think is most suitable for this dataset? Why?