University of Augsburg, Institute of Computer Science — SS 2019
Prof. Dr. P. Fischer — 14. June 2019
J. Kastner, L. Rudenko — Sheet 7

Analyzing Massive Data Sets

Exercise 1: Inverted Index (homework)

The following documents $D_1 - D_6$ and a query Q are given:

- D_1: ”the cat lies on this bench”
- D_2: ”the black cat on your desk”
- D_3: ”the red cat on my table”
- D_4: ”the red cat lies on my table”
- D_5: ”the black rabbit sits under this desk”
- D_6: ”this white rabbit lies behind your bench”
- Q: ”your rabbit sits behind this bench under our desk”

Find the documents that are more than 0.4 similar to the query using the **Jaccard Similarity**:

a) Create an **inverted index** for all documents.

b) Use the query and the created inverted index to determine the **candidates** (documents that may belong to the result).

c) Among the candidates, find the documents that have a similarity to the query of **at least 0.4** in **Jaccard Similarity**.

Exercise 2: Hierarchical Clustering (homework)

The following **words** are given:

- *able*
- *notable*
- *tabloid*
- *taboo*
- *tabby*
- *disable*
- *lovably*
- *stab*
- *ability*
- *labor*

The distance between the words is determinated with **Levenshtein** (chapter 4, slide 18), i.e., the number of edit operations to transform the word from one to the other. Initially, each word is in a cluster by itself. The clustering process can be stopped once we found **3 clusters**. To determine the distance between multipoint-clusters use (a) **single linkage** and (b) **complete linkage** method.

Exercise 3: k-means Clustering (live)

Given a set of **2-dimensional Points**.
$p_1 = (2, 12), p_2 = (3, 6), p_3 = (7, 15), p_4 = (9, 5), p_5 = (9, 17),$
$p_6 = (12, 19), p_7 = (12, 2), p_8 = (14, 1), p_9 = (14, 15), p_{10} = (16, 6).$

Please apply the **k-means clustering** algorithm until **termination** with $k = 3$ and following **centroids** for the first iteration: $c_1 = p_2$, $c_2 = p_4$ and $c_3 = p_8$.
Use the **Euclidean distance** as distance measure for the calculation of the distances between each point and the centroids.

Exercise 4: BFR (live)

In this exercise you should process the points with the BFR algorithm, gradually adding data from disk using chunks. You have two clusters (each with two points) at the beginning (see Figure 1). The points to be clustered are divided into two chunks:

- **first chunk:**
  $B(2, 8)$, $F(3, 3)$, $I(8, 9)$, $N(11, 2)$, $M(10.5, 1.5)$
- **second chunk:**
  $C(1.5, 9.5)$, $G(4, 5)$, $H(5, 5)$, $J(10, 3)$, $E(3, 2)$

A data point from a chunk can be added to the cluster (“discard set”), if it is ”close enough” to this cluster. ”Close enough” means, that the **Mahalanobis distance** between this point and the centroid of the cluster is smaller than the threshold (here **t**$=\ 2.5$). Points no fitting to any cluster may be processed with **hierarchical clustering** approach (complete linkage), forming mini-clusters (“compressed set”) and standalone items (“retained set”). To make a decision when to stop this clustering, use the relation between the diameter after adding a point and the average diameter. If the value is significantly higher (more than twice), the clustering should stop. Diameter is the maximum distance between points in a cluster. If you have no average diameter for CS (mini-clusters) (this is the case at the beginning of computation), use the average diameter for DS (“discard set”) clusters.

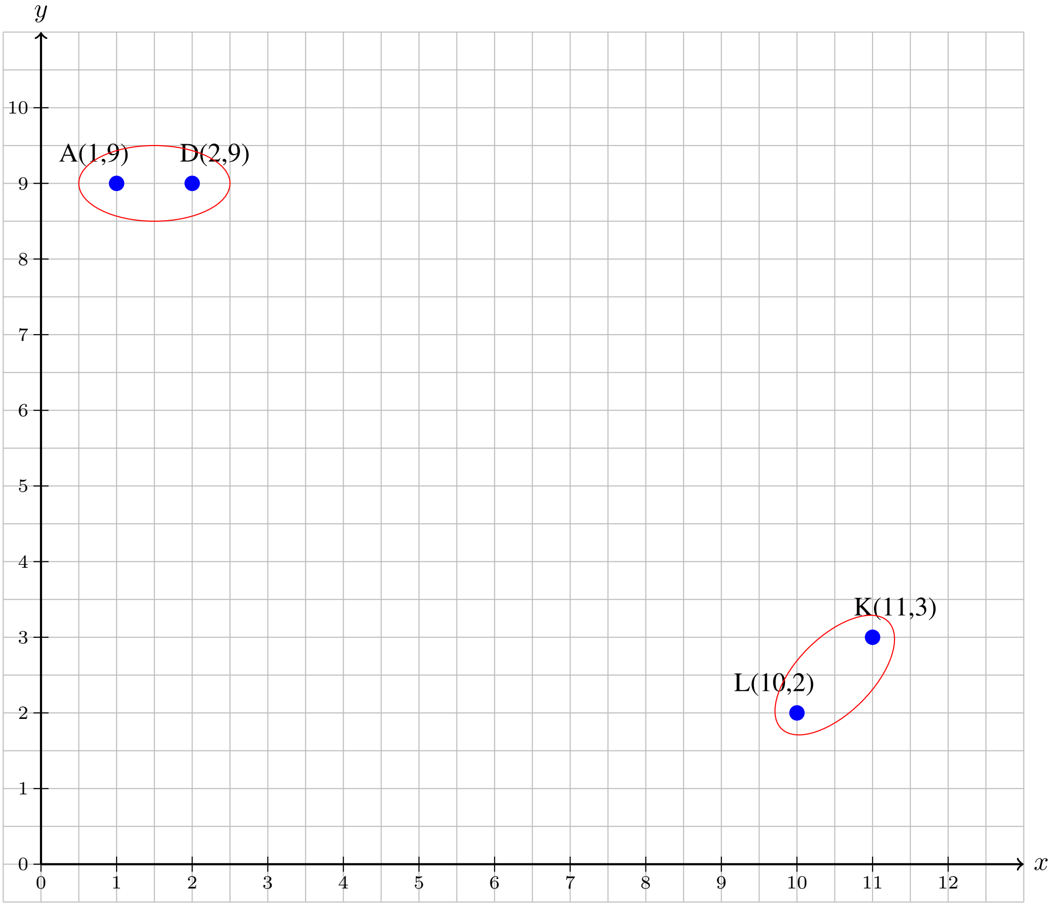

Abbildung 1: Two clusters.