TU Dortmund 14.4.2022
Artificial Intelligence / Data Science and Data Engineering **Machine Learning for Complex Data**
Prof. Dr. Erich Schubert
Prof. Dr. Emmanuel Müller **Summer 2022**
Daniel Wilmes

Assignment 1: Distance Functions and Apriori

Due: Thursday, 21.4.2022

Problem 1-1 Distance and Similarity **1+2+2+1**

In this exercise you will familiarize yourself with distance metrics. Note, that jupyter notebooks also allows Markdown cells in which you can render Math-Equations. Thus, you can include your answers in the juypter notebook.

(a) How many distance calculations are necessary to compute the $n \times n$ distance matrix for n points? Justify your answer.

(b) Given the vertices V of a connected, undirected graph with positive edge lengths, and the distance function $d(x,y)$ that computes the shortest path between vertices x and y in this graph.
Show that d is a metric by checking that all required conditions of a metric are satisfied:

1) $d(x,y) \geq 0$ (non-negativity)
2) $d(x,y) = 0 \Leftrightarrow x = y$ (identity of indiscernibles)
3) $d(x,y) = d(y,x)$ (symmetry)
4) $d(x,y) \leq d(x,o) + d(o,y)$ (triangle inequality)

(c) Given two input vectors $x, y \in \mathbb{R}^d$ in a d-dimensional vector space, we may use the Pearson correlation coefficient or the cosine similarity to compute the similarity between the vectors.

The Pearson correlation coefficient is defined as:

$$sim_P(x,y) = \frac{\sum_{i=1}^{d}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{d}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{d}(y_i - \overline{y})^2}}.$$

The cosine similarity between two vectors is defined as:

$$\mathrm{sim}_C(x,y) = \frac{x \cdot y}{\|x\|\,\|y\|}$$

with $\cdot$ being the dot product and $\|x\|$ being the Euclidean norm of vector x.
Under which conditions are the two measures sim_P and sim_C equal and why?.

(d) What is the geometric interpretation of the cosine similarity between two points $x, y \in \mathbb{R}^d$?

Problem 1-2 Apriori algorithm **9**

Let the set of items be {A, B, C, D, E, F, G, H, I, K, L, M}.

The transactions *T* are given by the following table:

Transaction database *T*

T_ID	items bought	T_ID	items bought
1	B E G H	7	A B D G H
2	A B C E G H	8	A B D G
3	A B C E F H	9	B D F G
4	B C D E F G H L	10	C E F
5	A B E K H	11	A C E F H
6	B E F G H I K	12	A B E G

For the minimum support 30%, determine the frequent itemsets using the Apriori algorithm. Pay attention to scan the database only once for each itemset length *k* (*not* for each candidate), as done by Apriori, to reduce your effort. Give the candidate sets after the join and the pruning step, as well as the resulting set of frequent itemsets after determining their support.