CS-E4600 — Homework #1

fall 2018

Due: Fri, Sep 28, midnight

Return your answers as a PDF file via `mycourses.aalto.fi`

Discussing the problems with your colleagues is allowed. However, you should write the answers by yourself. If you discuss with others, give proper acknowledgments. Looking for the answers in the internet is discouraged. You should at least make a serious effort to solve a problem by yourself before looking online. If you do, however, give proper acknowledgments.

Remember that there is a budget of 5 late days, which you can use in any way you want for the three homeworks and the programming assignment. Weekend days count as late days.

Typed solutions are strongly encouraged, especially if your hand writing is messy.

"I do not know" answers receive 15% of the max score for each problem. Partial credit will be given for partial solutions, but long off-topic discussion that leads nowhere may receive 0 points. Overall, think before you write, and try to give concise and crisp answers.

Problem 1 [10 points]

Consider the cosine similarity function $\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| ||\mathbf{y}||}$ defined for vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, and the induced distance function

$$d_{\cos}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\cos(\mathbf{x}, \mathbf{y}) + 1}{2}$$

Prove or disprove: $d_{\cos}$ is a metric.

Problem 2 [10 points]

Let $d : X \times X \to \mathbb{R}_+$ be a metric on X.

Question 2.1 Show that $D(x, y) = \frac{d(x,y)}{d(x,y)+1}$ is also a metric on X. Is D still a metric if we change 1 to an arbitrary value $k > 0$?

Question 2.2 What is the role of k? How can the choice of k affect the new metric?

Question 2.3 What is a possible application of the new metric?

Problem 3 [20 points]

In class we defined the Hausdorff distance for comparing sets of points.

In more detail, let (X, d) be a metric space. Given two subsets $A, B \subseteq X$ we define the *Hausdorff* distance between A and B as

$$d_H(A, B) = \max_{x \in A} \min_{y \in B} d(x, y).$$

As already discussed in class, one potential problem with d_H is that it is not symmetric. To overcome this problem we "symmetrize" Hausdorff by defining

$$D_H(A, B) = \max\{d_H(A, B), d_H(B, A)\}.$$

Prove or disprove: D_H is a metric.

Problem 4 (Walking on a graph) [25 points]

A graph $G = (V, E)$ consists of a set of vertices V and a set of edges E. An edge $e = \{u, v\}$ is an unordered pair of vertices, and we say that e is incident to vertices u and v. A walk W on a graph $G = (V, E)$ is defined as a sequence of vertices $W = \{v_0, v_1, \ldots, v_k\}$ so that $\{v_{i-1}, v_i\} \in E$ for all $i = 1, \ldots, k$. The vertices v_0 and v_k are called *source* and *destination* of the walk, respectively.

Question 4.1 Given a graph $G = (V, E)$ and two walks W and Z on G, propose a distance function to compare the walks W and Z. You should try to design your distance function so that (i) it is intuitive, and (ii) it safisfies the metric properties.

Question 4.2 Discuss the intuition of the distance function you proposed.

Question 4.3 Is your distance function a metric? Prove or disprove your claim.

Note: You will receive full points even if your distance function is not a metric, as long as your claim and proof are correct.

Question 4.4 Provide an algorithm to compute the distance function you proposed. (i) Argue that your algorithm is correct. (ii) What is the complexity of your algorithm?

Question 4.5 We now want to compute the similarity of the walks of two robots that navigate on the 2-d space. Propose an appropriate distance function. You may want to reuse some of the ideas developed in this problem, or you may want to propose a completely different distance function. In either case, provide a justification of your answer.

Problem 5 [10 points]

Consider a set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of n vectors of dimension d, i.e., $\mathbf{x}_i \in \mathbb{R}^d$. We want to compute the pair of furthest vectors in X according to the L_∞ distance. Provide such an algorithm that runs in time $\mathcal{O}(nd)$. Argue about the correctness of your algorithm.

Problem 6 (Distance lower bounding) [25 points]

We are given a dataset $D = \{x_1, \ldots, x_n\}$ of n objects. Each object $x_i \in D$ belongs in a space X. We assume that distances of objects in X are measured with a distance function $d : X \times X \to \mathbb{R}$.

Given a query object q we want to find the object x^* in the dataset D that is the closest to q. In other words we want to find

$$x^* = \arg\min_{x \in D} d(x, q).$$

As a simple example, if D is a set of genomic sequences (strings), given a new sequence q we want to find the sequence in D that is the closest to q.

A simple algorithm to find x^* is the following.

Algorithm NEAREST
Input: dataset $D = \{x_1, \ldots, x_n\}$, distance function d, query point q
Output: object $x^* \in X$ that is the closest to q
1. $dmin \leftarrow d(x_1, q)$
2. $x^* \leftarrow x_1$
3. **for** $i = 2, ..., n$
4. $\quad dtmp \leftarrow d(x_i, q)$
5. $\quad$ **if** $(dtmp < dmin)$
6. $\quad\quad dmin \leftarrow dtmp$
7. $\quad\quad x^* \leftarrow x_i$
8. **return** x^*

Assume that computing one instance of a distance evaluation requires time T. Then the running time of the algorithm NEAREST is $\mathcal{O}(nT)$.

Issues arise when computing the distance function d is expensive. In this case, the algorithm NEAREST may become quite slow.

For instance, in the example of genomic sequences discussed above, if the distance function is the *string edit distance*, then one distance computation requires time that is proportional to the product of the length of the two sequences. For large sequences this is prohibitively expensive.

One way to address this computational challenge is to come up with an alternative distance function d_ℓ, which is much faster to compute than d, and it is a *lower bound* of d. In particular, we want to find a distance function d_ℓ so that

$$d_\ell(x, y) \leq d(x, y) \text{ for all } x, y \in X.$$

Question 6.1. Explain how a distance function d_ℓ that is fast to compute and is a lower bound of d can be used to speed up the task of finding the nearest object in X for a given query object q.

Question 6.2. Provide pseudocode for a variant of algorithm NEAREST that uses a lower bound distance d_ℓ.

Question 6.3. While it is easy to come up with trivial lower bound distance functions that are very fast to compute (e.g., $d_\ell(x, y) = 0$ for all $x, y \in X$) we want to find lower bound distance functions that are *as large as possible*. Explain why.

Question 6.4. Propose lower-bound distance functions for the string edit distance.