Final exam for CS-E4600

Wed, Dec 14, 2016
09:00–12:00

READ CAREFULLY

This is an **open book** exam. It is allowed to use any textbook, printed material, or personal notes brought in the room. Using any electronic device is **not** allowed.

There are **3** problems. Each problem receives an equal number of points.

To get full points you should answer **any 2 out of the 3** problems. **Indicate clearly** in the first page of your paper for which 2 problems you wish to be graded.

A PhD student, Polina Rozenshtein, will come in class around 10am to answer clarifying questions about the problems.

Problem 1 [Clustering aggregation]

Consider a set $X = \{x_1, \ldots, x_n\}$ of n data objects. A *clustering* C is a partitioning of the set X into disjoint clusters. To represent a clustering we use the notation $C = \{(id_1, X_1), \ldots, (id_k, X_k)\}$, where id_i are cluster ids, and X_i are pairwise-disjoint subsets of X that cover X.

For example, given a set of objects $X = \{a, b, c, d, e\}$, the clustering $C = \{(1, \{a, b\}), (2, \{c\}), (3, \{d, e\})\}$ denotes that objects a and b are assigned to the 1st cluster, object c is assigned to the 2st cluster, and objects d and e are assigned to the 3rd cluster.

Given two different clusterings C_1 and C_2 over a set of objects X, we want to *compare* C_1 and C_2. Thus, we want to design a *distance function* d over the space of clusterings of a set of objects.

Question 1.1. Propose a distance function $d(C_1, C_2)$ for comparing two clusterings over a set of objects X.

Your distance function should be robust to naming conventions of the cluster ids and to permutations in object representation. For example, given three clusterings

$$C_1 = \{(1, \{a, b\}), (2, \{c\}), (3, \{d, e\})\}$$

$$C_2 = \{(1, \{c\}), (2, \{e, d\}), (3, \{a, b\})\}$$

$$C_3 = \{(red, \{d, e\}), (blue, \{c\}), (green, \{b, a\})\}$$

your distance function should recognize that the clusterings are in fact identical, and it should give

$$d(C_1, C_2) = d(C_1, C_3) = d(C_2, C_3) = 0.$$

More importantly, your distance function should be a metric. Thus, in addition to describing your proposed distance function, you should also prove that it is a metric.

Question 1.2. Given a set of n objects $X = \{x_1, \ldots, x_n\}$, provide an example of two clusterings C_1 and C_2 over X, for which the distance $d(C_1, C_2)$ is as large as possible.

In other words, you are asked to provide an example of two clusterings C_1 and C_2, which are as far as possible (under the distance function d that you proposed in Questsion 1.1.).

Question 1.3. Consider the problem of *clustering aggregation* or *clustering ensemble*: Given a set of n objects $X = \{x_1, \ldots, x_n\}$, and m different clusterings $C_1, \ldots, C_m$ over X, we want to find a clustering C^* that agrees with the given m clusterings as much as possible. Here C^* is any clustering in the space of possible clusterings of X.

To formulate this problem mathematically, we ask to find a clustering C^* that minimizes the objective

$$\sum_{i=1}^{m} d(C_i, C^*).$$

Propose an algorithm for this clustering aggregation problem. Justify your algorithm.

Hint: Even if you do not answer Questions 1.1. and 1.2., you can still approach Question 1.3. by assuming that you have at your disposal a distance function d between clusterings that satisfies the metric properties.

Problem 2 [Data streams]

We are processing a stream of numbers $S = s_1, s_2 \ldots, s_n$, where n is potentially very large. We apply the following algorithm:

Algorithm SIMPLECOUNTER

```
m ← 0
c ← 0
for i ← 1,...,n
    if (c = 0)
        m ← s_i
        c ← 1
    else if (m = s_i)
        c ← c + 1
    else
        c ← c − 1
return m
```

Question 2.1. Describe the algorithm SIMPLECOUNTER in words (no more than 5 lines).

Question 2.2. Prove that if there is a number x in the stream S that has majority (i.e., x appears at least $\frac{n}{2}$ times in the stream) then SIMPLECOUNTER will return that number x.

Question 2.3. Discuss the relevance of SIMPLECOUNTER as a streaming algorithm.

Problem 3 [Graph mining]

We have an undirected and unweighted graph $G = (V, E)$. In the programming project we have seen different methods for compute graph statistics, among of which the *diameter* of the graph. Recall that the diameter of a graph is defined to be

$$\Delta = \max_{x,y \in V} d(x, y),$$

where $d(x, y)$ is the shortest-path distance between nodes x and y.

We consider the following algorithm:

Algorithm SIMPLEFURTHEST
start with any node $r \in V$
find $z \in V$ to be the furthest node from r
$D \leftarrow d(r, z)$
return D

Question 3.1. What is the running time of SIMPLEFURTHEST?

Question 3.2. Show that SIMPLEFURTHEST is a factor-2 approximation algorithm. In other words, show that if Δ is the true graph diameter and D is the value returned by SIMPLEFURTHEST, then

$$D \geq \frac{1}{2}\Delta.$$

Question 3.3. Discuss how good is the performance (running time) of SIMPLEFURTHEST in practice.

Discuss how good is the quality (approximation guarantee) of SIMPLEFURTHEST in practice.