# Big data computing

## Homework 2

## Academic year 2016/2017

You can discuss homeworks with other students of the course. However, you must understand well your solutions and the final writeup must be yours and written in isolation.
*Please refer to course's Web page for detailed information about collaboration policy.*

Make sure that the solutions are typewritten or clear to read.

**Hand in your solutions and keep a copy for yourself. Solutions will be posted or presented after due date. In the final exam, you will be asked to explain your solutions and/or to go over your mistakes. For the theory part you should submit a pdf. It should contain your typewritten solution or the scan of a handwritten one. In the latter case, it should be very clearly written.**

**Due date: June 5th, 11.59pm**.

**Assignment 1.** Consider a stream from some universe, without loss of generality $[n]$ (defined in the previous problem). The second frequency moment (or surprise number) is defined as $F_2 = \sum_{i=0}^{n} m_i^2$, where $m_i$ denotes the number of times item $i$ appeared in the stream. A technique to estimate $F_2$ with an $(\epsilon, \delta)$ approximation[1] is described in [1, Section 2.2], but it uses 4-wise independent families.

Instead, show how we can directly use minhashing to estimate $F_2$. In more detail:

1. Design an algorithm that uses $k$ min-hash functions to provide an unbiased estimator of $F_2$.
2. Try to compute the value of $k$ that will result in an $(\epsilon, \delta)$ approximation. To this purpose, you may draw a lot of inspiration from [1, Section 2.2], in particular the proof of Theorem 2.2.

**Assignment 2.** Assume you want to cluster $n$ points in $\boldsymbol{R}^d$ (Figure 1) show an example with $d = 2$). Assume for simplicity that you know beforehand that there are 2 clusters to separate.

i) Show how dimensionality reduction techniques can be used to fulfill this task; ii) do your best to give an explanation of why the technique you propose might work.

**Assignment 3.** Assume you are given a user-movie dataset in the form of a very large bipartite graph as the one in Figure 2, describing users' watching habits, as explained in the figure's caption. The graph is stored as a list of edges on secondary storage.

Assume there are $k$ possible genres for the movies. You want to classify users based on movies they watched, but you do not want to use expensive spectral techniques, not even approximate ones like CUR. You would rather go for something more "traditional" like $k$-means (of course, some versione suited for large datasets, such as CURE), but this purpose you need a metric space.

1. Show how you can define a metric space on the users, so that distance between two users reflects how dissimilar they are.

[1] This means that the algorithm returns an estimation $\hat{F}_2$ such that $\mathbf{P}\left(|\hat{F}_2 - F_2| > \epsilon F_2\right) \leq \delta$.

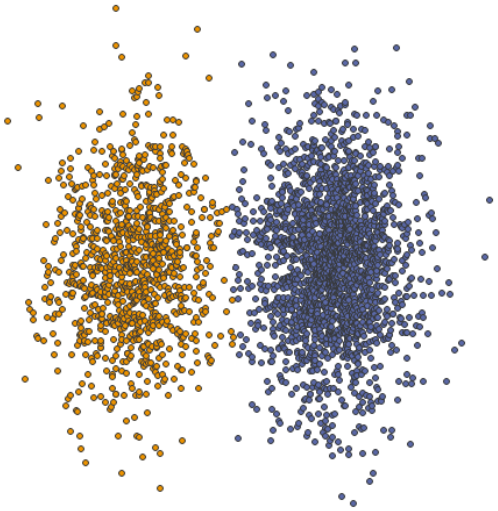

Figure 1: A cluster in two-dimension.

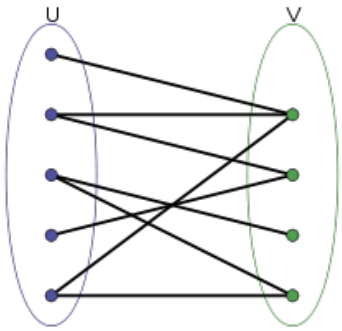


Figure 2: In the picture, set $U$ represents user, $V$ represents movies. We have an edge $(u, v)$, $u \in U$ and $v \in V$, if and only if user $u$ watched movie $v$.

2. Show how you can compute all you need to quickly (possibly with some approximation) compute pairwise distances between any two users by performing just one pass of the data.
3. Give a detailed description of the overall algorithm you are going to use. *Convincingly* discuss its computational costs (memory and time) and why it might work.

**Assignment 4.** You want to collect a sample of a large graph $G = (V, E)$ stored on secondary storage as a list of edges. Assume you have memory $\Theta(kn)$, for some sufficiently large constant $k$, with $k = |V|$. Your general strategy is the following: i) store all edge for vertices with degree at most $k$; ii) for vertices of degree higher than $k$, store a *uniform* sample of size $k$.[2]

1. Design an algorithm that will sample the graph according to the above guidelines. Give the details of the algorithm.

[2]Let's say, that for "big" vertices, it is ok if your sample is slightly smaller than $k$ edges.

2. Discuss pros and cons of the algorithm you design. E.g., how does your algorithm behave with respecting to preserving properties such as degree distribution?

Your arguments should be convincingly supported, in a quantitative way when needed.

**Assignment 5.** Use the dataset provided by the GDELT project that monitors the world's broadcast, print, and web news from nearly every corner of every country in over 100 languages and identifies the people, locations, organizations, counts, themes, sources, emotions, counts, quotes, images and events driving our global society every second of every day, creating a free open platform for computing on the entire world.

The GDELT dataset is available here: `http://data.gdeltproject.org/events/index.html`

Develop a map/reduce based solution that processes the GDELT dataset and provides the following analysis:

- top-5 countries per hour, per day
- top-5 events per hour, per day
- top-5 leaders mentioned per hour, per day
- trending leaders (fast increasing number of mentions) per hour, per day
- violent events per hour, per day
- protests per hour, per day
- violent vs non-violent actors per hour, per day
- state vs non-state actors per hour, per day
- trending violent regions (fast increasing number of events) per hour, per day.

For a quick overview of the fields in the GDELT data file format and their description:

`http://data.gdeltproject.org/documentation/GDELT-Data_Format_Codebook.pdf`

The CAMEO code reference is available here:

`http://data.gdeltproject.org/documentation/CAMEO.Manual.1.1b3.pdf`

**What to submit.** A .zip file containing the following:

1. Source code with accompanying README file about its use.
2. A brief report containing i) description of computer/cluster they used to execute the experiment; ii) what was the dataset they used (how many months they analyzed); iii) estimate on execution time (as accurate as possible); iv) brief summary on the output you got for each of the 9 questions

## References

[1] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 1996.