Big data computing - 2020/2021

Homework 3

- **Due date: January 10th, 2021, 11.59pm**

Assignment 1

(*Topic: locality sensistive hashing*) Consider Locality Sensitive Hashing for the estimation of Jaccard similarity between sets. Assume you use $m = r \cdot b$ independent permutations (hash functions). Assume further that you use the *banding technique* with b bands of r rows each. Work out the probability that the signatures of two sets S_1 and S_2 with Jaccard similarity J are identical in at least one band. *Introduce whatever notation you consider necessary*.

Assignment 2

(*Topic: locality sensitive hashing, approximate k-nearest neighbours*) Consider points in a metric space (e.g., assume you are interested in cosine similarity/distance). Assume that, to this purpose, you are representing a set S of points in $\mathbb{R}^d$ using locality sensitive hashing (in particular, assume you are using the banding technique with bucketing). Write an *efficient* algorithm (sublinear with respect to the number of points) that, given a *new* point $\mathbf{p} \in \mathbb{R}^d$, returns the k best candidates to be (approximately) the k nearest neighbours of $\mathbf{p}$. You should be rigorous and clearly describe the algorithm. *To this purpose, introduce whatever notation you consider necessary*.

Assignment 3

(*Topic: dimensionality reduction*) Assume we have a vector $\mathbf{x} \in \mathbb{R}^d$ and consider $F(\mathbf{x}) = \mathbf{s}^T\mathbf{x}$, where $\mathbf{s}$ is a d-dimensional *random* vector with entries drawn *uniformly and independently* from $\{-1, 1\}$. **1**. What is the value of $\mathbb{E}[F(\mathbf{x})^2]$? **2.** Give a formal proof of your claim. *Introduce whatever notation you consider necessary*.

Assignment 4

(*Topic: Streaming and sampling*) Assume a stream S of items (e.g., tweets or news feeds that are captured in real time). Define the *Reservoir sampling* algorithm to maintain, at any point in time, a uniform sample of k items from stream S. *Introduce whatever notation you consider necessary*.

Assignment 5

(*Topic: streaming and sampling*) Assume a stream S of items from a discrete universe U (we can assume without loss of generality that $U = [n] = \{0, \ldots, n-1\}$). Note that the same item can appear multiple times in S. Design an algorithm to keep a sample of 1 item from S such that, at any point of the stream, each of the *distinct* items observed so far has the same probability of being the sampled item. You should convince me that your scheme is going to work.

Example. Assume you have observed the following elements so far: $2, 1, 2, 4, 1, 9$. After observing the 6-th element in the stream, we observed 4 distinct items (namely, 1, 2, 4 and 9). So, at this point of the stream S, each of these distinct items should have the same probability $1/4$ of being the sampled item. Next, assume the 7-th element in S is item 8. Thus, after the 7-th item has been observed we have a total of 5 distinct items so far. The probability that each of these items is the sampled one after the 7-th element has been observed should be 0.2.

Hint: hash functions might help.

Introduce whatever notation you consider necessary.