Big data computing - 2019/2020

Homework 3

Due date: Sunday, January 19th, 2020, 11.59pm

Assignment 1

The intent behind this assignment is showing how you can apply sketching techniques normally conceived for streaming to other settings.

Given an undirected graph $G = (V, E)$ (which we henceforth assume to be *connected*), the individual neighbourhood $IN(v, h)$ of a vertex $v \in V$ at distance h is the number of vertices in G whose shortest path distance from v is *at most* h hops.

Reference [1] below and Section 10.8 of the Massive Datasets book show how Flajolet and Martin's sketches can be used to estimate $IN(v, h)$ for all vertices at the same time. Go over Ref. [1] up to Section 3.1 (you can read the entire paper if you like, but that won't be necessary) and/or Section 10.8 of the Massive Datsets book and be sure you get the idea. Then:

Show how the very same algorithm can be used to estimate (actually, in most cases exactly) the diameter of the graph. This is an easy assignment, there is no need for sophisticated math. All you need to do is i) understand how the same algorithm can be used to estimate diameter, ii) describe how you are doing this, iii) provide an explanation of why this might work in practice.

[1] Christopher R. Palmer, Phillip B. Gibbons, Christos Faloutsos. ANF: A Fast and Scalable Tool for Data Mining in Massive Graphs. Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge and Data Mining, 2002. Available here

Assignment 2

Assume you have a stream of keys (e.g. the destination IP addresses of packets observed at a router). At every point of the stream, you want to maintain a sample of size s, such that each *distinct* item/key observed so far appears in the sample with the same probabilty s/n, where n is the number of *distinct* items observed so far. E.g., in the following example, assume $s = 2$. We provide two possible states of the stream after respectively 5 and 10 items have been observed *overall*:

```
3 4 3 3 2 --> 3 distinct keys, each should appear with prob. 2/3
........
3 4 3 3 2 2 5 7 5 6 --> 6 distinct keys, each should appear with prob. 2/6
```

Show how you can use min-hashing to achieve this. Assume for simplicity you can efficiently sample from an ideal family. In particular, you should i) show the algorithm and ii) prove that, after $n \geq s$ distinct items have been observed, the probability of having one of the distinct items observed so far in the sample is $\simeq s/n$ and the approximation becomes increasingly better as $n >> a$.

Note: at some point you might need $\left(1 - \frac{1}{x}\right)^y \simeq 1 - \frac{y}{x}$ for $y < x$ and $x > 2$.

Assignment 3

In this assignment, we investigate the effect of random projections on angles (more precisely, inner products). Assume $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. Suppose now that we project $\mathbf{x}, \mathbf{y}$ onto a low dimensional space with m dimensions, by taking $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \frac{1}{\sqrt{m}} S(\mathbf{x}, \mathbf{y})$, where notation should be clear from context and where S has shape $m \times d$ and its generic entry S_{ij} satisfies:

$$S_{ij} = \begin{cases} 1, \text{with prob. } \frac{1}{2} \\ -1, \text{with prob. } \frac{1}{2} \end{cases}$$

and where all the S_{ij}'s are *statistically independent* (see [2] for further details).

Prove that $\mathbb{E}[\hat{\mathbf{x}}^T \hat{\mathbf{y}}] = \mathbf{x}^T \mathbf{y}$, where expectation is taken with respect to the randomness of S. Note that you should provide a complete and formal proof.

[2] Achlioptas, Dimitris. "Database-friendly random projections: Johnson-Lindenstrauss with binary coins." *Journal of computer and System Sciences* 66.4 (2003): 671-687.

Assignment 4

Consider a stream of integers belonging to the set $[n] = \{0, \ldots, n-1\}$. Design a streaming algorithm that at any point of the stream, (approximately) maintains the *sum of the distinct items observed so far*. The algorithm should use memory $O(log_2 n \cdot f(n))$, where $f(n)$ denotes the space requirements of a distinct counting algorithm over a universe of n items (e.g., an FM sketch).

Hint: each integer in the stream can be expressed as a binary string of $m = \lceil log_2 \rceil$ bits in the obvious way. How is the sum of the *distinct* integers in the stream expressed (exactly) using such a binary representation?