Algorithmic Data Analysis

Esther Galbrun
Autumn 2020
Q3.1: Streaming paradigm

Name the main constraints encountered when mining streams

a) Data points are processed as they come or lost for ever

b) The number of distinct values is so large that counting them or computing basic statistics is challenging

c) At peak periods the system has to shed part of the load

d) The data comes from a non-stationnary distribution
Q3.2: Algorithm families

Name these two families of algorithms, which are naturally suited to the streaming setting:

a) always have an answer ready, which gets better as they learn

b) do not require access to the entire training data at once
Q3.3: Inequalities

Piece the inequalities together

\[ P(X \leq (1 - c)E[X]) \leq e^{-E[X]c^2/2} \quad \text{Chebychev} \]
\[ P(X \geq c) \leq E[X]/c \quad \text{Chernoff} \]
\[ P(|X - E[X]| \geq c) \leq \text{var}[X]/c^2 \quad \text{Hoeffding} \]
\[ P(E[X] - X \geq c) \leq e^{-2c^2 \sum r_i^2} \quad \text{Markov} \]

\( X \) is a random variable
\( X \) takes only nonnegative values
\( X = \sum Y_i \quad Y_i \sim B(1, p_i) \) (Bernoulli)
\( Y_i \) has range of bounded size \( r_i \)
\( Y_i \)'s are independent random variables

\( c \in [0, 1] \quad c \) is a constant
\( c \geq 0 \quad c \) satisfies \( E[X] \leq c \)
This is a bound on both tails
This is an upper-tail bound
This is a lower-tail bound
There is a similar upper-tail bound
There is a similar lower-tail bound
Q3.4: Bloom-filters

Consider a Bloom filter of size $m$ storing $n$ distinct values. Assume that the number of hash functions $k$ is increased.

What can you say about the probability of false positives?

What can you say about the probability of false negatives?
## Q3.5: Tools for purposes

Associate tools and purposes

<table>
<thead>
<tr>
<th>Tools</th>
<th>Purposes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alon–Matias–Szegedy sketch</td>
<td>Count distinct values</td>
</tr>
<tr>
<td>Approximate counting</td>
<td>Estimate zeroth-order moment</td>
</tr>
<tr>
<td>Bloom filters</td>
<td>Estimate first-order moment</td>
</tr>
<tr>
<td>Count-min sketch</td>
<td>Estimate second-order moment</td>
</tr>
<tr>
<td>Flajolet–Martin algorithm</td>
<td>Identify frequent items</td>
</tr>
<tr>
<td>Hoeffding trees</td>
<td>Test set-membership</td>
</tr>
<tr>
<td>Lossy counting algorithm</td>
<td>Train a classifier</td>
</tr>
<tr>
<td>Reservoir sampling</td>
<td></td>
</tr>
</tbody>
</table>
Which of these methods use hash functions as an ingredient?

- Alon–Matias–Szegedy sketch
- Approximate counting
- Bloom filters
- Count-min sketch
- Flajolet–Martin algorithm
- Hoeffding trees
- Lossy counting algorithm
- Reservoir sampling
What is the mean-median trick?

a) A method to estimate the mean of a stream of data by combining the medians of multiple samples

b) A method to obtain a robust estimate of a random variable from a collection of weaker estimates