CSE 4502/5717 Big Data Analytics

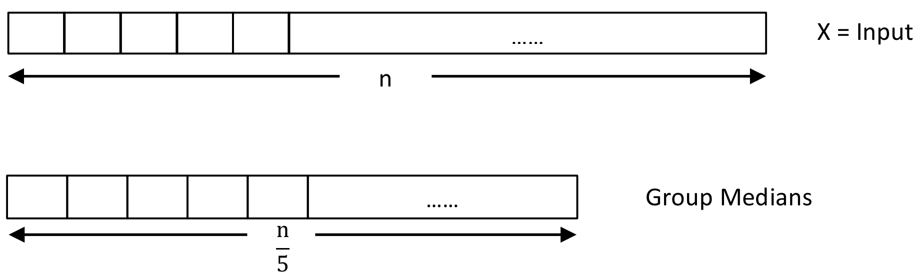
Notes taken by: Sheng Xiong

Lecture 5 – 02/05/2018

An out–of–core implementation of BFPRT algorithm:

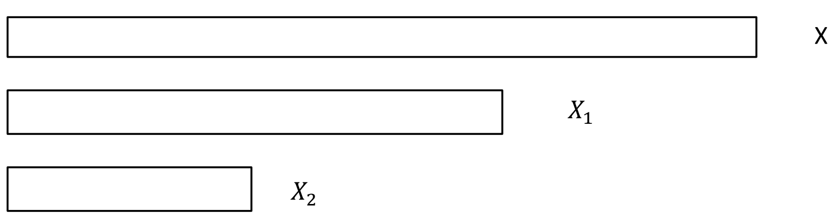
1. Bring one block at a time and group the blocks into groups of size 5 each. Then, find the median of each group and write the medians in an output buffer.

    - When we are done with one block, bring the next block.
    - When the output buffer has one block, write it in the disk.
    - Proceed similarly until all the blocks have been processed.



The number of (read) I/O operations = $\frac{n}{B}$

2. Find recursively the median of the medians.
3. Partition the input into X_1 and X_2, using M as the pivot.



The number of I/O operations = $\frac{n}{B}$

4. Do a recursively selection on X_1 or X_2 (as needed).

Analysis:

Let I(n) be the number of I/O operations needed on any input of size n and for any i.

$$I(n) = \frac{n}{B} + I(\frac{n}{5}) + \frac{n}{B} + I(\frac{7}{10}n)$$

Hypothesis: $I(n) \le \frac{C \cdot n}{B}$ for some constant C.

Proof by induction:

Base case: Easy

Induction step: Assume the hypothesis for inputs of size up to (n - 1).

We will prove it for n:

$$I(n) \le I(\frac{n}{5}) + I(\frac{7}{10}n) + 2 \cdot \frac{n}{B}$$

$$\le \frac{C \cdot n}{5B} + \frac{C \cdot 7}{10} \cdot \frac{n}{B} + 2 \cdot \frac{n}{B}$$

$$RHS = 0.9\frac{C \cdot n}{B} + 2 \cdot \frac{n}{B}$$

$$RHS \le \frac{C \cdot n}{B} \text{ if } 0.9\frac{C \cdot n}{B} + 2 \cdot \frac{n}{B} \le C \cdot n$$

$$\Rightarrow 0.1C \ge 2$$

$$\Rightarrow C \ge 20$$

$$\Rightarrow I(n) \le 20 \cdot \frac{n}{B}$$

Chernoff Bounds:

A Bernoulli trial has two outcomes: success or failure.

Assume Prob. [Success] = p

The number of successes in *n* independent Bernoulli trials is a Binomial Random Variable denoted as B(n, p).

If X = B(n, p), then:

(1) Prob. $[X > m] \le (\frac{np}{m})^m \cdot e^{-np+m}$, for any $m > np$;

(2) Prob. [X > (1 + ε)np] ≤ exp($\frac{-\varepsilon^2 np}{3}$), for any $0 < \varepsilon < 1$; and

(3) Prob. [X < (1 - ε)np] ≤ exp($\frac{-\varepsilon^2 np}{2}$), for any $0 < \varepsilon < 1$.

Example:

X = B(1000, $\frac{1}{2}$)

(1 + ε) · 500 = 600

=> $\varepsilon = \frac{1}{5}$

Prob. [X > 600] ≤ exp(- $\frac{1}{25 \cdot 3}$ · 500)

= exp(- $\frac{20}{3}$)

Markov's inequality: Prob. [X > 1.2 (500)] ≤ $\frac{1}{1.2} = \frac{5}{6}$

Let X be any sequence of n real numbers;

Let S be a random sample from X, with |S| = s;

Let q ∈ S such that Rank(q, S) =j

Note: Rank(x, X) = |{ q ∈ X : q < x}| + 1

> **Example:**
>
> X = 3, 8, 12, 9, 5, 4, 11, 35, 2
>
> Rank(5, X) = 3 + 1 = 4

Let r_j be the rank of q in X.

Then, E[r_j] = j · $\frac{n}{s}$

Lemma (Rajasekaran & Reif 1986):

Prob. [| $r_j - j \cdot \frac{n}{s}$ | > $\sqrt{3\alpha}\frac{n}{\sqrt{s}}\sqrt{\log n}$] ≤ $n^{-\alpha}$

A Randomized Algorithm (Floyd & Rivest 1975):

1. Pick a random sample S from X.
2. Identify two elements l_1 and l_2 such that

   Rank(l_1, S) = i · $\frac{s}{n} - \delta$

   Rank(l_2, S) = i · $\frac{s}{n} + \delta$, where $\delta = \sqrt{4\alpha s \log n}$

Let $n_1 = | \{ q \in X : q < l_1 \} |$

Let $n_2 = | \{ q \in X : q \leq l_2 \} |$

If the i^{th} smallest element of X is not within $[l_1, l_2]$, then start all over. Note that the i^{th} smallest element of X will be in the interval if $i > n_1$ and $i \leq n_2$.

If the number of elements of X that are in interval $[l_1, l_2]$ is "large", then start all over.

3. Scan through X to get $Y = \{ q \in X : l_1 \leq q \leq l_2 \}$.
4. Find the $(i-n_1)^{th}$ smallest element of Y and output.