COT 6936: Topics in Algorithms

Giri Narasimhan
ECS 254A / EC 2443; Phone: x3748
giri@cs.fiu.edu
http://www.cs.fiu.edu/~giri/teach/COT6936_S10.html
https://online.cis.fiu.edu/portal/course/view.php?id=427

Massive Data Sets

- Examples of large persistent data sets
  - WalMart Transaction data (1 PB?)
  - Sloan Digital Sky Survey (100 TB)
  - Web (over a Trillion pages; over 1 PB of text)
  - CERN (expected to produce ~40 TB/sec)
- Large data sets with time-sensitive data
  - Financial tickers data
  - Credit Card usage traffic
  - Network Traffic: Telecom & ISP traffic
  - Sensor data

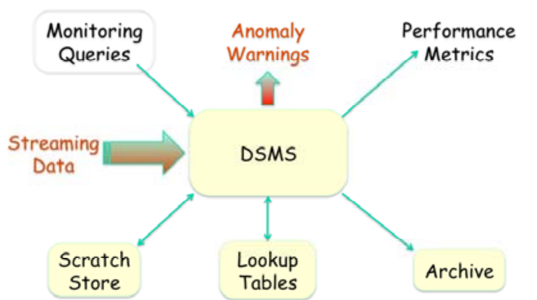
Important Issues for Stream Algorithms

- Key parameters
  - Amount of memory available; window size
  - Per-item processing time; # of Passes on data
  - Tolerance to error
- What is needed?
  - Summarizations, synposes, sketches
  - Randomization and sampling
  - Pattern Discovery
  - Anomaly Detection
  - Clustering and Classifications

Streaming Model of Computation

- N = # of items seen so far, window size
  - amount of memory available
- ε = error tolerance
- Memory usage = poly(1/ ε , log N)
- Query Time = poly(1/ ε , log N)

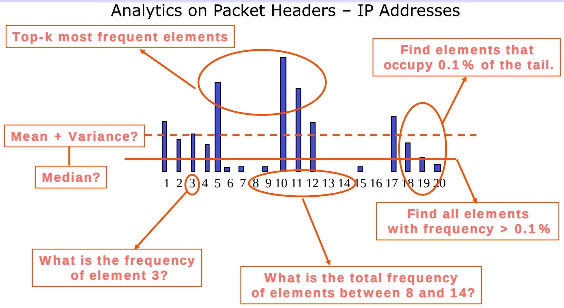
Network Monitoring System



Based on slide by R. Motwani, 2005

Frequency Related Problems

Analytics on Packet Headers – IP Addresses



How many elements have non-zero frequency?

Based on slide by R. Motwani, 2005

Warm up Problems

- Given stream of values, find mean.
  - Easy.
  - Maintain sum of all values and number of items
- Given stream, find standard deviation.
  - Not so hard
- Given stream of bits and window size N, count number of 1s in window
  - Space and time?
    - Naïve: Store the window: requires N bits
    - Can you do better?

Problem: Finding Missing Labels

- Packets arrive in random order, each is labeled from set {1,….,n}.
- Assume that one packet is missing.
- Find the label of the missing packet.

- Bit vector of length n
  - Space O(n)
- Maintain sum of labels and subtract from N
  - Space O(log n)

Problem: Finding Missing Numbers

- Same as problem 1, but there may be up to k missing numbers.
- Instead of sum of numbers, we maintain k different functions of the numbers seen.
  - Decoding is not so easy
    - Needs factoring polynomials
  - Randomized algorithms
    - $O(k^2 \log n)$
    - $O(k \log k \log n)$

Problem: Find number of unique items

- Simple hashing scheme to do counting
  - Space = O(m)
  - Time = O(1) per item in stream

Problem: Find fraction of rare items

- Rarity $r[t] = |\{j | c_t[j] = 1\}| / u$
  - Number of items in stream that are rare (i.e., appear only once)

Problem: Counting

- Given a stream of bits, at every time instant, maintain count of number of 1s in last N elements
  - Deterministic algorithms
    - Θ(N) bits of memory to answer in O(1) time [Why?]

Problem: Counting

- How well can you approximate with o(N) memory? [Datar et al. SIAM J C 2002]
  - Use histogram techniques
    - Build time-based histograms in which every bucket represents a contiguous time interval
    - Idea: Use uniform buckets
    - Problem: 1s may not be distributed uniformly
    - Solution: Use non-uniform buckets
  - Results
    - $O((1/\varepsilon)\log^2 N)$ bits
    - $(1+\varepsilon)$-approximate count in $O(1)$ time

$\Omega((1/\varepsilon)\log^2(N\varepsilon))$

Other problems

- COUNTING: Given a stream of bits, at every time instant, maintain count of number of 1s in last N elements
- SUM: Given a stream of positive integers in range [0..R], at every time instant, maintain sum of last N elements

Clustering

- K-Means
  - Constant-factor approximation, O(nk log k) time, O(k) space, single pass [Charikar et al. 1997]
- K-Medians
  - Constant-factor approximation, O(nk log k) time, $O(n^{\varepsilon})$ space, single pass [Guha et al. 2002]