Theory of Data Streams

S. Muthu Muthukrishnan
Familiar Puzzle: Missing Number

- $A$ shows $B$ numbers $1, \ldots, n$ but in a permuted order and leaves out one of the numbers.
- $B$ has to determine the missing number.

- **Key:** $B$ has only $O(\log n)$ bits.
Familiar Puzzle: Missing Number

- **A** shows **B** numbers $1, \ldots, n$ but in a permuted order and leaves out one of the numbers.

- **B** has to determine the **missing number**.

- **Key:** **B** has only $O(\log n)$ bits.

- **Solution:** **B** maintains the running sum $s$ of numbers seen. Missing number is $\frac{n(n+1)}{2} - s$. 
A New Puzzle: One Word Median

- A sees items $i_1, i_2, \ldots$ arrive in a stream.
- A has to maintain the median $m_j$ of the items $i_1, \ldots, i_j$.
- Key: A is allowed to store only one word of memory (of $\log n$ bits).
A New Puzzle: One Word Median

- \( A \) sees items \( i_1, i_2, \ldots \) arrive in a stream.
- \( A \) has to maintain the median \( m_j \) of the items \( i_1, \ldots, i_j \).
- Key: \( A \) is allowed to store only one word of memory (of \( \log n \) bits).
- Each \( i_j \) generated independently and randomly from some unknown distribution \( \mathcal{D} \) over integers \([1, n]\).
A New Puzzle: One Word Median

- $A$ sees items $i_1, i_2, \ldots$ arrive in a stream.
- $A$ has to maintain the median $m_j$ of the items $i_1, \ldots, i_j$.
- **Key:** $A$ is allowed to store only one word of memory (of $\log n$ bits).
- Each $i_j$ generated independently and randomly from some unknown distribution $\mathcal{D}$ over integers $[1, n]$.

**Solution.** Maintain $\mu_j$.
- If $i_{j+1} > \mu_j$, $\mu_{j+1} \leftarrow \mu_j + 1$.
- If $i_{j+1} < \mu_j$, $\mu_{j+1} \leftarrow \mu_j - 1$. 
This Talk

- Two basic primitives and applications with data stream algorithms.
This Talk

- Two basic primitives and applications with data stream algorithms.
  - Count-Min sketch, applications to compressed sensing
  - $L_0$ sampling, applications to graph problems
- Some topics:
  - $L_2$ sketches and applications
  - Nonstreaming applications of streaming results
  - Distributed streaming
  - Pan-privacy
  - Cryptography
A Basic Problem: Indexing

- Imagine a virtual array $F[1 \cdots n]
- Updates: $F[i] + +, F[i] --$
- Assume $F[i] \geq 0$ at all times
- Query: $F[i] = ?$
- **Key:** Use $o(n)$ space, may be $O(\log n)$ space
Count-Min Sketch

- For each update $F[i] + +$,
  - for each $j = 1, \ldots, \log(1/\delta)$, update $cm[h_j(i)] + +$.
  - Estimate $\tilde{F}(i) = \min_{j=1,\ldots,\log(1/\delta)} cm[h_j(i)]$. 

\[
\begin{array}{ccc}
  \log(1/\delta) & +1 & cm \ array \\
  2 & h_2(i) & h_1(i) \\
  1 & F[i] + + & e/\epsilon \\
\end{array}
\]
Count-Min Sketch Analysis

- \( F[i] \leq \tilde{F}[i] \). With probability at least \( 1 - \delta \),
  \[ \tilde{F}[i] \leq F[i] + \epsilon \sum_{j \neq i} F[j]. \]
Count-Min Sketch Analysis

- $F[i] \leq \tilde{F}[i]$. With probability at least $1 - \delta$,
  \[
  \tilde{F}[i] \leq F[i] + \epsilon \sum_{j \neq i} F[j].
  \]

- $X_{i,j}$ is the expected contribution of $F[j]$ to the bucket containing $i$, for any $h$.
  \[
  E(X_{i,j}) = \frac{\epsilon}{e} \sum_{j \neq i} F[j].
  \]
Count-Min Sketch Analysis

- $F[i] \leq \tilde{F}[i]$. With probability at least $1 - \delta$,

$$\tilde{F}[i] \leq F[i] + \varepsilon \sum_{j \neq i} F[j].$$

- $X_{i,j}$ is the expected contribution of $F[j]$ to the bucket containing $i$, for any $h$.

$$E(X_{i,j}) = \frac{\varepsilon}{e} \sum_{j \neq i} F[j].$$

- Consider $\Pr(\tilde{F}[i] > F[i] + \varepsilon \sum_{j \neq i} F[j])$:

$$\Pr() = \Pr(\forall j, F[i] + X_{i,j} > F[i] + \varepsilon \sum_{j \neq i} F[j])$$

$$= \Pr(\forall j, X_{i,j} \geq eE(X_{i,j})) < e^{-\log(1/\delta)} = \delta$$
Count-Min Sketch

- Claim: $F[i] \leq \tilde{F}[i]$. With probability at least $1 - \delta$,

  $$\tilde{F}[i] \leq F[i] + \varepsilon \sum_{j \neq i} F[j]$$

- Space used is $O(\frac{1}{\varepsilon} \log \frac{1}{\delta})$.

- Time per update is $O(\log \frac{1}{\delta})$. Indep of $n$.

Improve Count-Min Sketch?

- **Index Problem:**
  - ALICE has \( n \) long bitstring and sends messages to BOB who wishes to compute the \( i \)th bit.
  - Needs \( \Omega(n) \) bits of communication.

- **Reduction of estimating** \( F[i] \) **in data stream model.**
  - \( I[1 \ldots 1/(2\varepsilon)] \) such that
    - \( I[i] = 1 \Rightarrow F[i] = 2 \)
    - \( I[i] = 0 \Rightarrow F[i] = 0; F[0] \leftarrow F[0] + 2 \)
  - Observe that \( \|F\| = \sum_i F[i] = 1/\varepsilon \)
Improve Count-Min Sketch?

Index Problem:
- ALICE has $n$ long bitstring and sends messages to BOB who wishes to compute the $i$th bit.
- Needs $\Omega(n)$ bits of communication.

Reduction of estimating $F[i]$ in data stream model.
- $I[1 \cdots 1/(2\varepsilon)]$ such that
  - $I[i] = 1 \rightarrow F[i] = 2$
  - $I[i] = 0 \rightarrow F[i] = 0; F[0] \leftarrow F[0] + 2$
- Observe that $||F|| = \sum_i F[i] = 1/\varepsilon$

Estimating $F[i] \leq \tilde{F}[i] \leq F[i] + \varepsilon||F||$ implies,
- $I[i] = 0 \rightarrow F[i] = 0 \rightarrow 0 \leq \tilde{F}[i] \leq 1$
- $I[i] = 1 \rightarrow F[i] = 2 \rightarrow 2 \leq \tilde{F}[i] \leq 3$
and reveals $I[i]$.
- Therefore, $\Omega(1/\varepsilon)$ space lower bound for index problem.
Count-Min Sketch, The Challenges

- Not all projections, dimensionality reduction are the same:
  - All prior work $\Omega(1/\varepsilon^2)$ space, via Johnson-Lindenstrauss
Count-Min Sketch, The Challenges

- Not all projections, dimensionality reduction are the same:
  - All prior work $\Omega(1/\varepsilon^2)$ space, via Johnson-Lindenstrauss
- Not all hashing algorithms are the same:
  - Pairwise independence
Count-Min Sketch, The Challenges

- Not all projections, dimensionality reduction are the same:
  - All prior work $\Omega(1/\varepsilon^2)$ space, via Johnson-Lindenstrauss
- Not all hashing algorithms are the same:
  - Pairwise independence
- Not all approximations are sampling.
  - Recovering $F[i]$ to $\pm 0.1|F|$ accuracy will retrieve each item precisely.

10000000 items inserted
999996 items deleted
4 items left
Using Count-Min Sketch

- For each $i$, determine $\tilde{F}[i]$
- Keep the set $S$ of heavy hitters ($\tilde{F}[i] \geq 2\varepsilon \|F\|$).
  - Guaranteed that $S$ contains $i$ such that $F[i] \geq 2\varepsilon \|F\|$ and no $F[i] \leq \varepsilon \|F\|$.
  - Extra $\log n$ factor space for $n$ queries.

Problem is of database interest.
Using Count-Min Sketch

- For each $i$, determine $\tilde{F}[i]$
- Keep the set $S$ of heavy hitters ($\tilde{F}[i] \geq 2\varepsilon||F||$).
  - Guaranteed that $S$ contains $i$ such that $F[i] \geq 2\varepsilon||F||$ and no $F[i] \leq \varepsilon||F||$
  - Extra $\log n$ factor space for $n$ queries
Problem is of database interest.
- Faster recovery: In each bucket, recover majority $i$ ($F[i] > \sum_j$ same bucket as $i F[j]/2$)
Using Count-Min Sketch

- For each $i$, determine $\tilde{F}[i]$
- Keep the set $S$ of heavy hitters ($\tilde{F}[i] \geq 2\varepsilon \|F\|$).
  - Guaranteed that $S$ contains $i$ such that $F[i] \geq 2\varepsilon \|F\|$ and no $F[i] \leq \varepsilon \|F\|$.
  - Extra $\log n$ factor space for $n$ queries

Problem is of database interest.

- Faster recovery: In each bucket, recover majority $i$ ($F[i] > \sum j$ same bucket as $i F[j]/2$)
  - Takes $O(\log n)$ extra time, space
  - Gives compressed sensing in $L_1$:

\[
\|F - \tilde{F}_k\|_1 \leq \|F - F^*_k\|_1 + \varepsilon \|F\|_1
\]

Count-Min Sketch: Summary

- Solves many problems:
  - Heavy hitters, compressed sensing, inner products, ...
- Applications to other CS/EE areas:
  - NLP, ML, Password checking.
- Systems, code, hardware.
  - Gigascope, CMON, Sawzall, MillWheel, ...

Wiki: http://sites.google.com/site/countminsketch/
**L₀ Sampling**

- Imagine a virtual array $F[1 \cdots n]$
- Updates: $F[i]++$, $F[i]--$
- Assume $F[i] \geq 0$ at all times

- Query: inverse sample?
  Return $i$, $F[i] \neq 0$ with prob $\frac{1}{|\{i | F[i] \geq 0\}|}$

- **Key**: Use $o(n)$ space, may be $O(\log n)$ space

- Solutions use $O(1/\varepsilon^2)$ space.
Application of $L_0$ Sampling

- **Graph Sketch**: For node $i$, let $a_i$ be vector indexed by node pairs. $a_i[i, j] = 1$ if $j > i$ and $a_i[i, j] = -1$ if $j < i$.
- For any subset $S \subset V$, $\text{support}(\sum_{i \in S} a_i) = E(S, V - S)$
Application of $L_0$ Sampling

- **Graph Sketch**: For node $i$, let $a_i$ be vector indexed by node pairs. $a_i[i, j] = 1$ if $j > i$ and $a_i[i, j] = -1$ if $j < i$.
- For any subset $S \subset V$, $\text{support}(\sum_{i \in S} a_i) = E(S, V - S)$
- Prob: Is $G$ connected?
  - **Algorithm (Spanning Forest)**:
    - For each node, select an incident edge
    - Contract selected edges. Repeat until no edges
  - **Data structure**: $L_0$ sketch $C$ for each $a_j$.
  - Use $Ca_j$ to get incident edge. Then, run algorithm above.

Observe:

$$\sum_{j \in S} Ca_j = C(\sum_{j \in S} a_j) \rightarrow e \in \text{support}(\sum_{j \in S} E(S, V - S))$$

Ahn, Guha, McGregor: Analyzing graph structure via linear measurements. SODA12.
Dynamic graph connectivity in polylogarithmic worst case time. B. Kapron, V. King, B. Mountjoy. SODA13
Topics: $L_2$ approximation

- $L_2$ estimate: $\|F\|_2 = \sum_i F[i]^2$
- $Y_j = \sum_{i=1,\ldots,w/2}(cm_F[h_j(2i - 1)] - cm_F[h_j(2i)])^2$
- Gives $\|F\|_2 \leq (1 + \varepsilon)\|F\|_2$
- Space $O(1/\varepsilon^2)$, update time is $O(\log 1/\delta)$
Topics: $L_2$ approximation

- $L_2$ estimate: $||F||_2 = \sum_i F[i]^2$
- $Y_j = \sum_{i=1,...,w/2}(cm_F[h_j(2i-1)] - cm_F[h_j(2i)])^2$
- Gives $\overline{||F||_2} \leq (1 + \varepsilon)||F||_2$
- Space $O(1/\varepsilon^2)$, update time is $O(\log 1/\delta)$

- Ex: Least squares approximation
  - Problem: Given matrix $A \in R^{n \times d}$ and a vector $b \in R^n$, find $x \in R^d$ such that $z = ||Ax - b||_2$ is minimized.
  - Result: Can find $y$ such that $||Ay - b||_2 \leq (1 + \varepsilon)z$ in $O(nd \log d)$ randomized time.
  - Solution: Consider $TA$ and $Tb$, where $T \in R^{O(d/\varepsilon \times d)}$
  - Extends to low rank matrix approx, classification,...

Low Rank Approximation and Regression in Input Sparsity Time. K. Clarkson, D. Woodruff, STOC 13
Compute the Discrete Fourier Transform of signal of size $n$

- Classical: $O(n \log n)$ time.
- Recent result: There exists a randomized $O(k \log n)$ time algorithm for $k$-sparse case.

Nearly Optimal Sparse Fourier Transform H. Hassanieh, P. Indyk, D. Katabi, E. Price. STOC 12
Topics: Distributed Learning

- Alice has data $D_A$ and Bob has $D_B$, and learn linear classifier. Minimize communication. $h^*$ is optimal.
- $E_D(h)$ is the number of points misclassified by $h$ on $D$.
- $g$ has $\varepsilon$-error if $E_D(g) - E_D(h^*) \leq \varepsilon |D|$.
- There is a $O(\log 1/\varepsilon)$ round two way communication protocol with $O(1)$ bits per round and $\varepsilon$-error.

Efficient Protocols for Distributed Classification and Optimization. H. Daume, J. Phillips, A. Saha, S. Venkatasubramanian. ALT12
Distributed Learning, Communication Complexity and Privacy. N. Balcan, A. Blum, S. Fine, Y. Mansour. ICML12
Topics: Pan Privacy

- Well known notion of differential privacy (DP). What if the internal state is breached?
- Pan-Privacy. For every two neighboring streams, at any time, internal state and final output should be DP.
- Use count-min and $L_0$ sketches to get approximate pan-private algorithms.

Topics: Streaming Cryptography

- Cryptography against polynomial time adversaries using a streaming algorithm?
- Recent result: Streaming algorithms for one-way functions and pseudorandom generators with $O(1)$ passes over two read-write tapes, under suitable assumptions.

Conclusions

- Two basic primitives and applications with data stream algorithms.
  - Count-Min sketch, applications to compressed sensing
  - $L_0$ sampling, applications to graph problems
- Some topics:
  - $L_2$ sketches and applications
  - Nonstreaming applications of streaming results
  - Distributed streaming
  - Pan-privacy
  - Cryptography

Area continues to grow tentacles