Space-Efficient Algorithms for Maintaining Multi-Dimensional Histograms

Sudipto Guha
Piotr Indyk
Nick Koudas
Nitin Thaper

November 4, 2001

Overview of Talk

- Problem Statement
- Applications
- Algorithms
- Experimental results

Problem Statement

Maintain a histogram for a dynamic, multi-dimensional data distribution.

Definitions

- The *frequency distribution* is a function $D : \{1 \ldots n\}^\ell \to \{1 \ldots M\}$
  - D can be updated,i.e., $D(p) = D(p) + \Delta$
- Histogram H for distribution D is a function $H : \{1 \ldots n\}^\ell \to \{1 \ldots M\}$
  - Defined by hyperrectangles $S_1 \ldots S_k$ and values $v_1 \ldots v_k$.
  - For point $p \in \{1 \ldots n\}^\ell$, $H(p)$ represents an estimate to $D(p)$.

Goal

Efficiently maintain *good* approximation H of D (*wrt* L_2 *norm*)

- small memory
- low running time

Motivation

Underlying database:

- Query estimation and optimization
- Approximate query answering

No underlying database:

- Analysis and visualization of networking data

Previous work

- Initial work on one-dimensional histograms [Poosala'97,Kooi'80,JagadishKM+'98]

- Dynamic histograms in one dimension [GibbonsMP'97,MattiasVW'98,GilbertKMS'01]

- Histograms in multiple dimensions [MuralikrishnaD'88,PoosalaI'97,GunopulosKTD'00]

- Self-tuning histograms [BrunoGC'01(STholes)]

Our work deals with dynamic, multi-dimensional histograms with a practical focus.

Classes of histograms

- *Tiling histograms*: the hyperrectangles form a tiling of $\{1 \ldots n\}^\ell$ For any p we have $H(p) = v_i$, where $p \in S_i$

- *Priority histograms*: the hyperrectangles can overlap. For any p we have $H(p) = v_i$ where i is the largest index such that $p \in S_i$; if none exists, $H(p) = 0$.

- *Additive histograms*: the hyperrectangles can overlap. For any p we have
  $H(p) = \sum_{i:p \in S_i} v_i$
  (the value of $H(p)$ is 0 if there is no S_i containing p).

Observation: The distribution D and histogram H can be viewed as vectors in an N-dimensional space (where $N = n^{\ell}$)

Thus we can use the concept of *sketches*.

Informally, a *sketch* is a projection of the original high-dimensional vector onto a low-dimensional space.

The following theorem explains why this is feasible.

Theorem 1 (Johnson-Lindenstrauss) *Consider a random linear mapping* $A : \Re^N \to \Re^d$*, such that each entry of the matrix* A *is chosen independently from a certain distribution. If* $d = O(\log(1/P)/\epsilon^2)$*, then the mapping* A *has the property that for any fixed* $x \in \Re^n$ *we have* $\|x\|_2 \le \|Ax\|_2 \le (1+\epsilon)\|x\|_2$ *with probability at least* $1 - P$*.*

Nice properties of sketches

- Exponentially small space
- Fast updates (an update requires $O(d)$ time)
- Composable (useful for combining sketches from multiples sources)

Extracting histogram from sketch

Algorithm:

- Maintain sketch AD of distribution D
- Find H s.t. $\|AD - AH\|_2$ minimized

Problem: Large space of histograms to choose from

Idea: Choose buckets one by one greedily !

Algorithm GREEDY:

Inputs:

Distribution $D : \{1 \ldots n\}^\ell \to \{1 \ldots M\}$

Matrix A *chosen according to Theorem 1*

Output:

Priority Histogram H *with* B *buckets*

Initialization:

Generate Sketch AD *of* D

Set $N = n^\ell$

Initialize the histogram H *to empty*

For $i = 1$ to $B = k \ln(NM^2)$

For all hyperrectangles $S \subset \{1 \ldots n\}^\ell$

(1)Create the histogram $H_S[x]$ obtained by adding the rectangle S to H and setting its value to the indeterminate variable x

(2)Transform $H_S[x]$ to its vector representation $\bar{H}[x]$. Compute the sketch $A\bar{H}_S[x]$; note that the sketch is a linear function in x

(3)Define $C_S(x) = \|A\bar{H}_S[x] - AD\|_2^2$; observe that $C_S(x)$ is quadratic function of x

(4)Compute x which minimizes $C_S(x)$ and denote it by x_S

Let S be the rectangle with smallest value of $C_S(x_S)$

Add S to H with value x_S

Performance of Algorithm GREEDY

Theorem 2 *Let D be the distribution and let H^* be the tiling k-histogram which minimizes $\|D - H^*\|_2^2$. If the sketching procedure preserves the distances exactly, then the priority histogram H reported by* greedy *satisfies $\|D - H\|_2^2 \leq \|D - H^*\|_2^2$.*

Theorem can be modified accordingly when the distances are preserved only approximately

Proof Idea:

- The initial square cost of H is at most NM^2

- If we added all rectangles from H^* to H with appropriate values, the error of H would be reduced from $\|D - H\|_2^2$ to $\|D - H^*\|_2^2$. Thus, one of the rectangles must reduce the error by at least $1/k \cdot (\|D - H\|_2^2 - \|D - H^*\|_2^2)$.

- Therefore, if we add the best rectangle S to H with the best value (forming H_S), we have that

  $$\|D - H_S\|_2^2 - \|D - H^*\|_2^2 \leq (1 - 1/k)(\|D - H\|_2^2 - \|D - H^*\|_2^2)$$

- After i stages we have
  $$\|D - H\|_2^2 - \|D - H^*\|_2^2 \leq (1 - 1/k)^i NM^2$$

  .

Time complexity

- Outer loop runs B times
- Number of rectangles = $n^{2\ell}$
- Sketch computation takes $O(n^{\ell} dB)$ time*

Overall time complexity $O(n^{3\ell} dB^2)$

$O(n^6 dB^2)$ for the two-dimensional case

$O(n^4 dB^2)$ after improvement

*can be improved to $O(dB)$ but requires more working space

Heuristics to improve running time

- Restrict search to rectangles whose sides are powers of 2, $O(n^2 \log^2 ndB^2)$

- Only search squares, $O(n^2 \log ndB^2)$

- Use additive instead of priority histograms, $O(n^2 \log ndB)$

- Instead of picking *the best* bucket in each iteration, pick the first bucket that reduces the error by α/k, $O(n^2 \log nd)$

Experimental evaluation

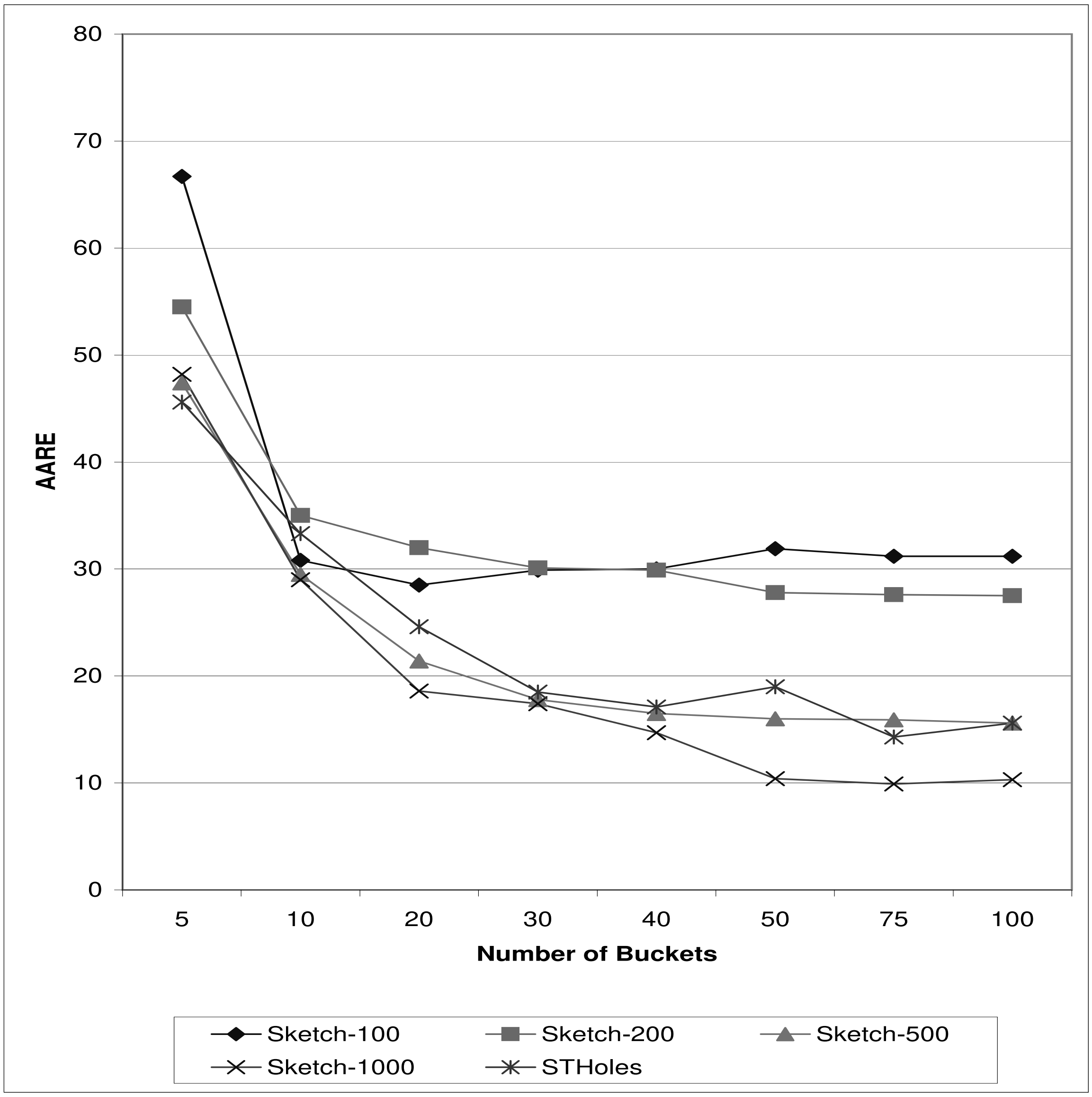
- Accuracy

- Running time

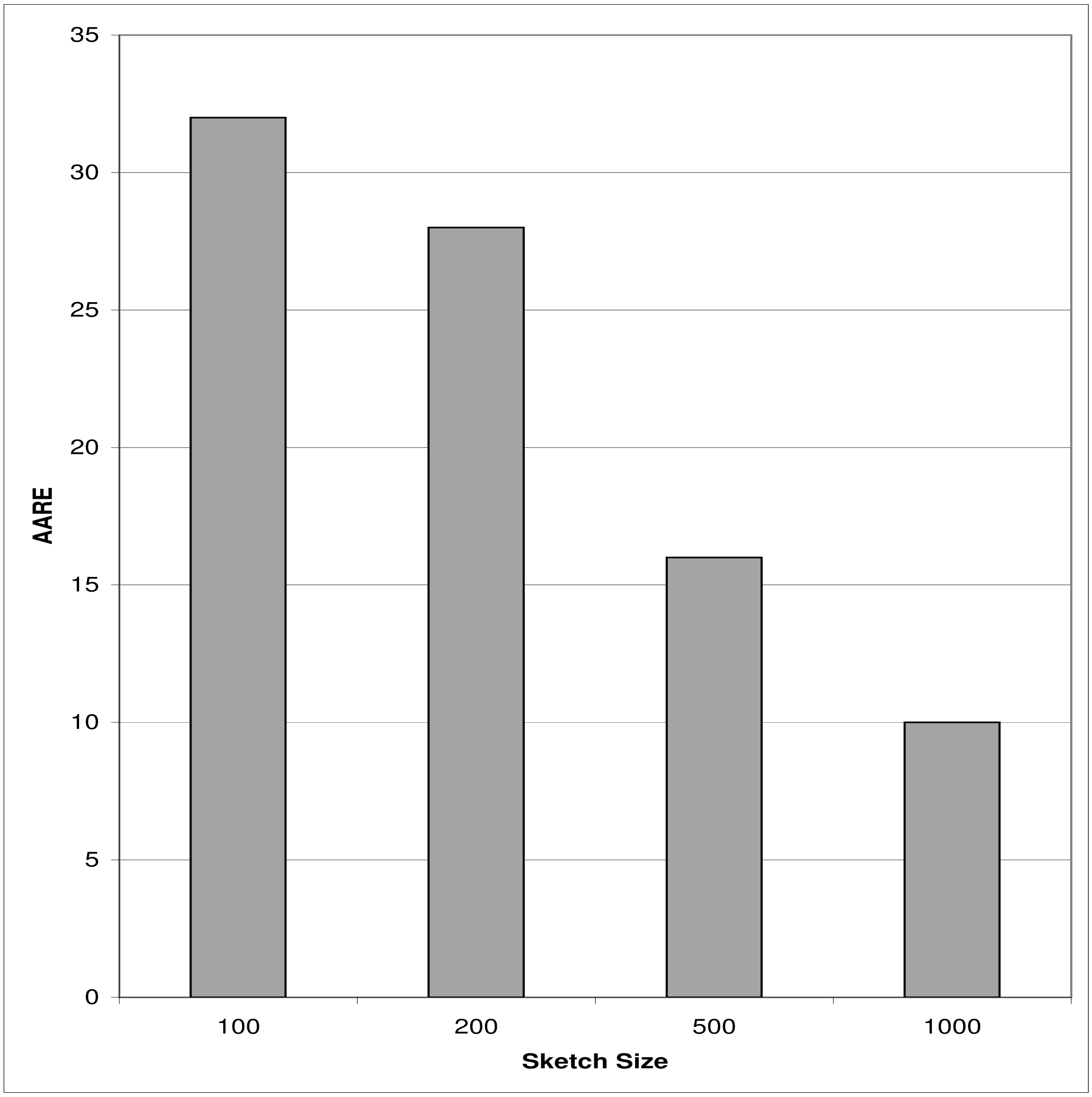
Comparison with the *STholes* algorithm

Used real data sets generated from network traffic

Accuracy

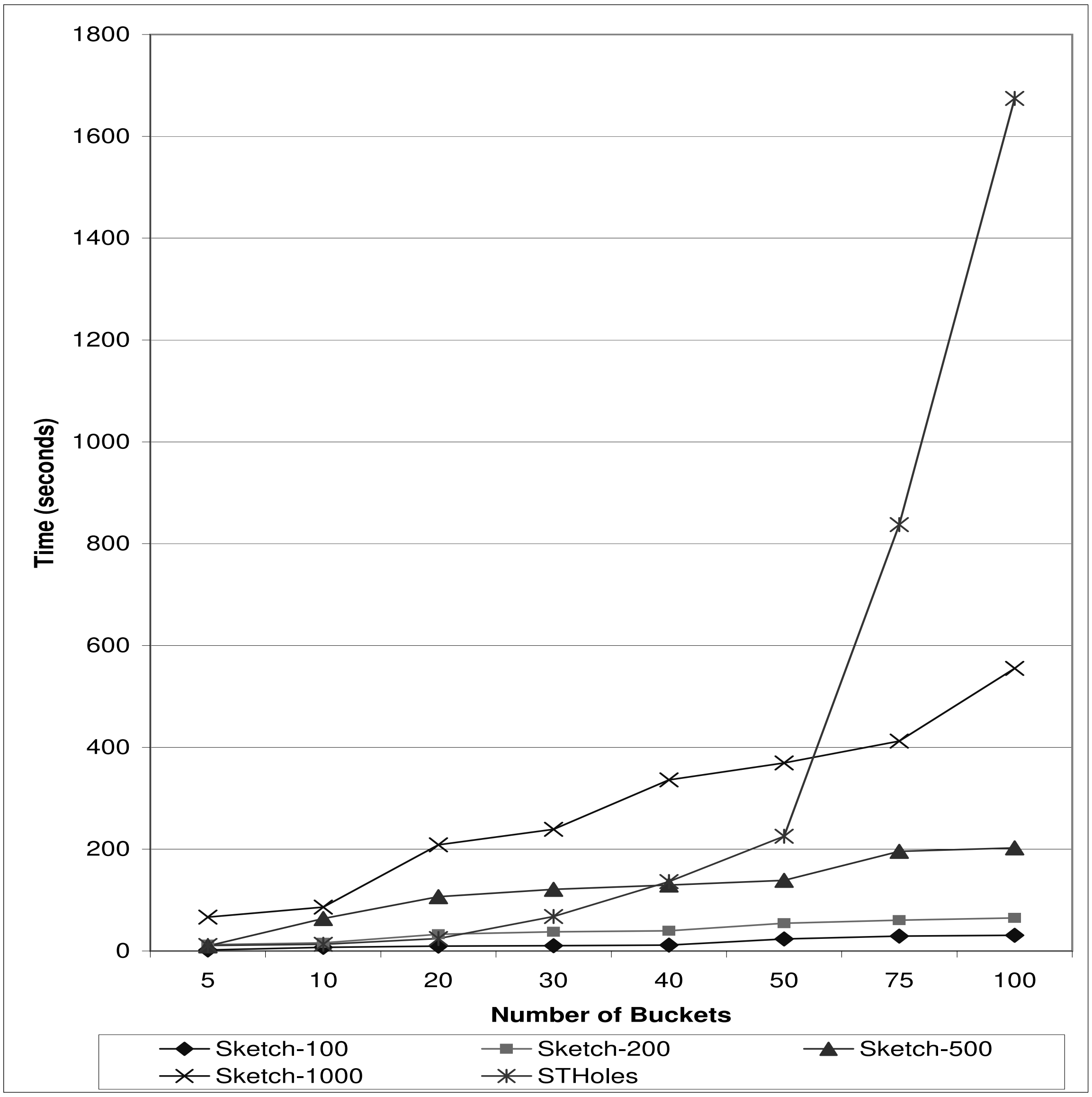


Error versus number of buckets for various sketch sizes

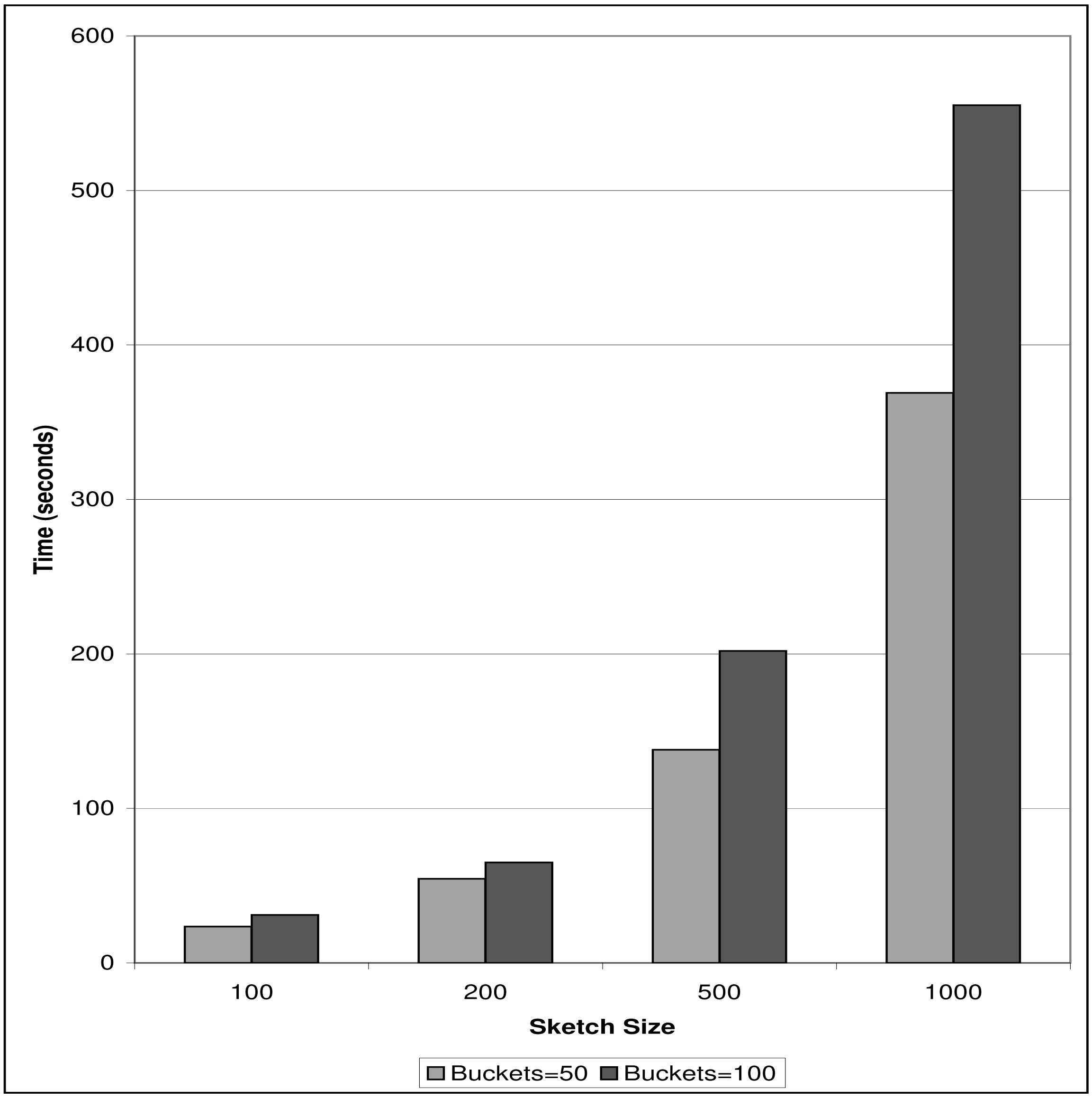


Error versus sketch size for 50 buckets

Running time

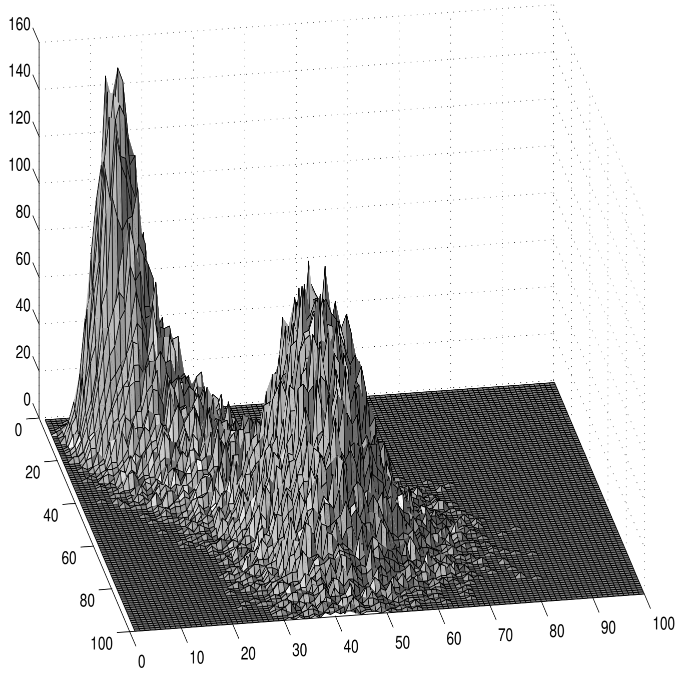


Running time versus number of buckets

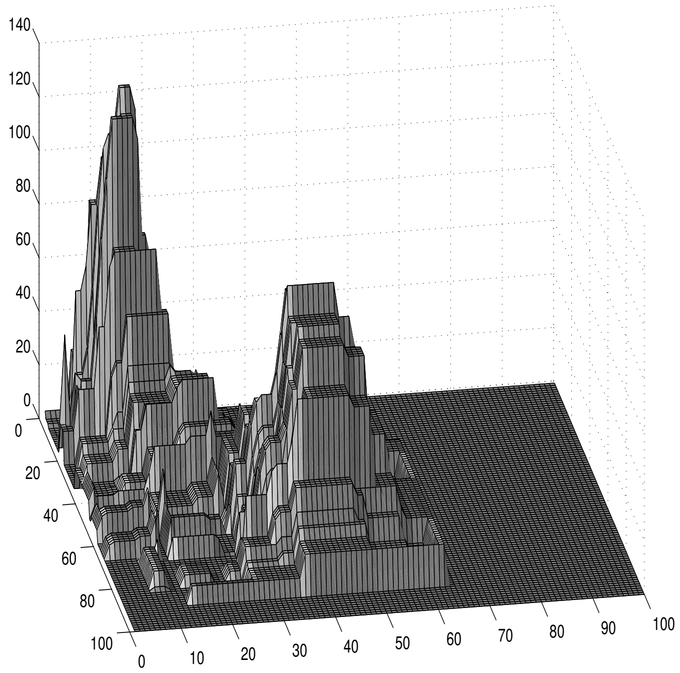


Running time for different sketch sizes

Distribution of *Traffic1*



Extracted Histogram of *Traffic1*



Conclusions

- Presented a sketch-based approach for maintaining a snapshot of the underlying multi-dimensional distribution

- Algorithms with provable guarantees

- High accuracy, efficient algorithm in practice