Summarizing and Mining *Skewed* Data Streams

Graham Cormode
cormode@bell-labs.com

S. Muthukrishnan
muthu@cs.rutgers.edu

Data Streams

Many large sources of data are generated as streams of updates:

- IP Network traffic data
- Text: email/IM/SMS/weblogs
- Scientific/monitoring data

Must analyze this data which is high speed (tens of thousands to millions of updates/second) and massive (gigabytes to terabytes per day)

Data Stream Analysis

Analysis of data streams consists of two parts:

- Summarization
  - Fast memory is much smaller than data size, so need a (guaranteed) concise synopsis
  - Data is distributed, so need to combine synopses
- Mining
  - Extract information about streams from synopsis
  - Examples: Heavy hitters/frequent items, changes/difference, clustering/trending, etc.

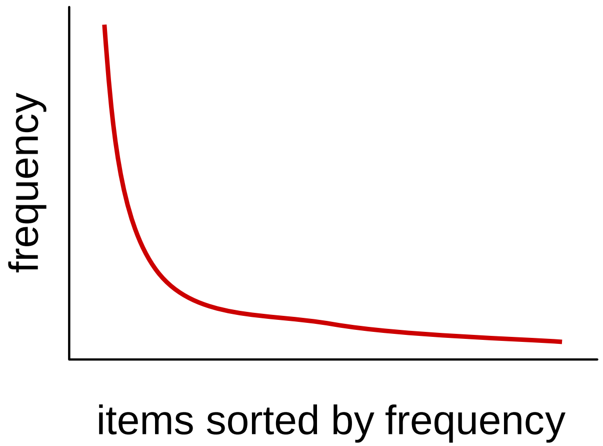
Skew In Data

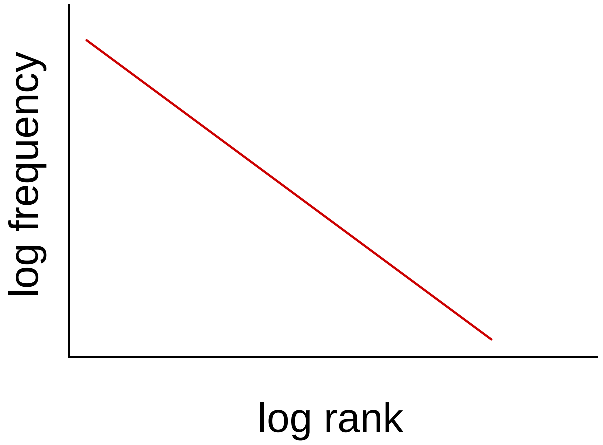
Data is rarely uniform in practice, typically skewed

Few items are frequent, then a long tail of infrequent items

Such skew is prevalent in network data, word frequency, paper citations, city sizes, etc.

One concept, many names: Zipf distribution, Pareto distribution, Power-laws, multifractals, etc.





Zipf Distribution

Items drawn from a universe of size U

Draw N items, frequency of i'th most frequency is

$$f_i \approx Ni^{-z}$$

Proportionality constant depends on U, z, not N

z indicates skewness:

- $z = 0$: Uniform distribution
- $z < 0.5$: light skew/no skew
- $0.5 \leq z < 1$: moderate skew
- $1 \leq z$: (highly) skewed

} most real data in this range

Typical Skews

Data Source	Zipf skewness, z
Web page popularity	0.7 — 0.8
FTP Transmission size	0.9 — 1.1
Word use in English text	1.1 — 1.3
Depth of website exploration	1.4 — 1.6

Our contributions

A simple synopsis used to approximately answer:

- Point queries (PQ) — given item i, return how many times i occurred in the stream, f_i
- Second Frequency moment (F_2) — compute sum of squares of frequencies of all items

The basis of many mining tasks: histograms, anomaly detection, quantiles, heavy hitters

Asymptotic improvement over prior methods:
for error bound ε, space is $o(1/\varepsilon)$ for $z>1$
previously, cost was $O(1/\varepsilon^2)$ for F_2, $O(1/\varepsilon)$ for PQ

Point Estimation

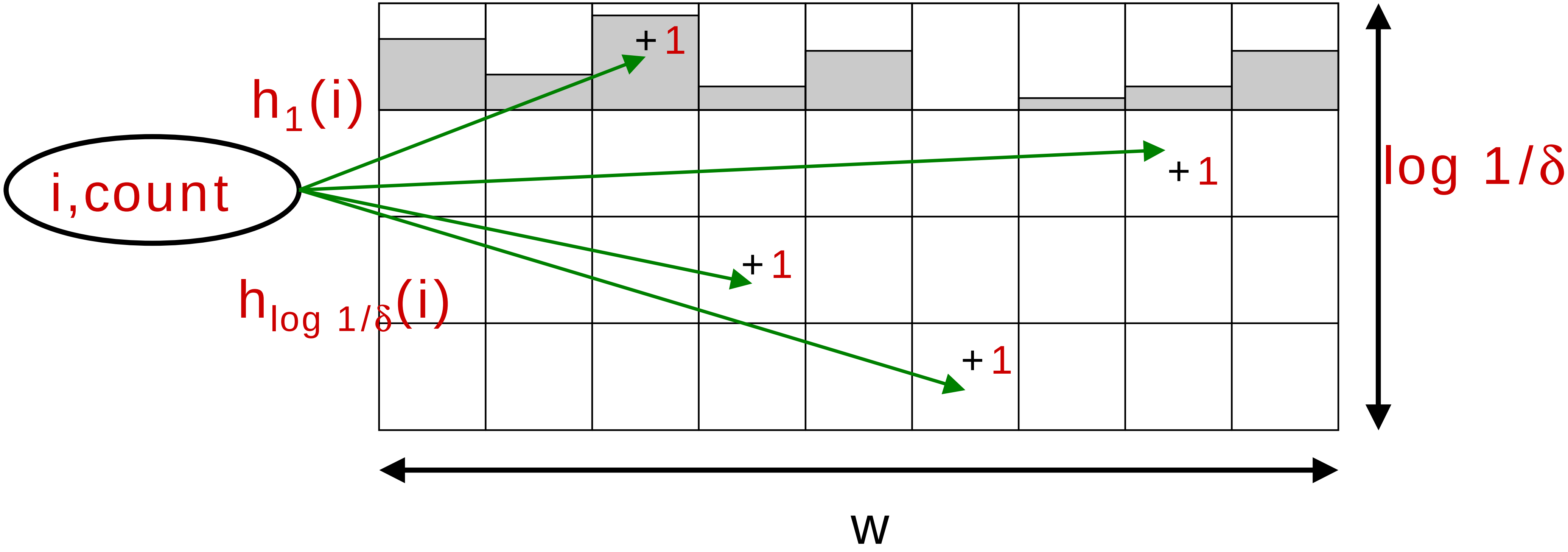
Use the CM Sketch structure, introduced in [CM04] to answer point queries with error $<\varepsilon N$ with probability at least $1-\delta$

Tighter analysis here for skewed data, plus new analysis for F_2.

Ingredients:

- –Universal hash fns $h_1..h_{\log 1/\delta}$ {items}$\rightarrow$ {1..w}
- –Array of counters $CM[1..w, 1..\log 1/\delta]$

Update Algorithm



Count-Min Sketch

Analysis for Point Queries

Split error into:

- Collisions with w/3 largest items
- Collisions with the remaining items

With constant probability (2/3), no large items collide with the queried point.

Expected error $$= \frac{1}{w} \sum_{x=w/3+1}^{U} f_x \leq \frac{N}{w}\left(\frac{w}{3}\right)^{1-z} \leq \frac{\varepsilon N}{3}$$

Applying Zipf tail bounds and setting w = $3\varepsilon^{-1/z}$.

Markov Inequality: Pr[error > εN] < 1/3.

Take Min of estimates: Pr[error > εN] < $3^{-\log 1/\delta}$ < δ

Application to top-k items

Can find f_i with (1±ε) relative error for i<k (ie, the top-k most frequent items).

Applying similar analysis and tail bounds gives:

$$\frac{Nk^{1-z}}{w} = \frac{\varepsilon N k^{-z}}{2}$$

and so w = O(k/ε) for any z>1.

Improves the $O(k/\varepsilon^2)$ bound due to [CCFC02]

We only require z>1, do not need value of z.

Second Frequency Moment

Second Frequency Moment, $F_2 = \sum_i f_i^2$

Two techniques to make estimate from CM sketch:

- CM^+: $\min_j \sum_{k=1}^{w} CM[j,k]^2$
  — min of F_2 of rows in sketch
- CM^-: $\text{median}_j \sum_{k=1}^{w/2} (CM[j,2k] - CM[j,2k-1])^2$
  — median of F_2 of differences of adjacent entries in the sketch

We compare bounds for both methods.

CM$^+$ Analysis

With constant probability, the largest w$^{1/2}$ items all fall in different buckets. For z>1:

$$\begin{aligned}
\mathsf{E}(X_j) &= \sum_{i=1}^{U} f_i^2 + \sum_{i=1}^{U} \sum_{j=1, j\neq i}^{U} f_i f_j \Pr[h(i) = h(j)] - F_2 \\
&\leq F_2 + \frac{1}{w}(\sum_{i=1}^{U} \sum_{j=1, j\neq i}^{U} f_i f_j - \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} f_i f_j) - F_2 \\
&\leq \frac{1}{w}(2 \sum_{i=1}^{m} f_i \sum_{j=m+1}^{U} f_j + (\sum_{i=m+1}^{U} f_i)^2) \\
&\leq \frac{2}{w}(\sum_{i=1}^{U} f_i \sum_{j=m+1}^{U} f_j) \leq \frac{2N^2 c_z m^{1-z}}{w(z-1)} \leq \frac{2F_2 c_z (2z-1)}{c_z^2 (z-1)} w^{\frac{-(1+z)}{2}}
\end{aligned}$$

CM+ Analysis

Simplifying, we set the expected error = ½εF_2.

This gives w = $O(\varepsilon^{-2/(1+z)})$.

Applying Markov inequality shows error is at most εF_2 with constant probability.

Taking the minimum of the log 1/δ repetitions reduces failure probability to δ.

Total space cost = $O(\varepsilon^{-2/(1+z)} \log 1/\delta)$, provided z>1

CM^- Analysis

For $z>1/2$, again constant probability that the largest $w^{1/2}$ items all fall in different buckets.

We show that:

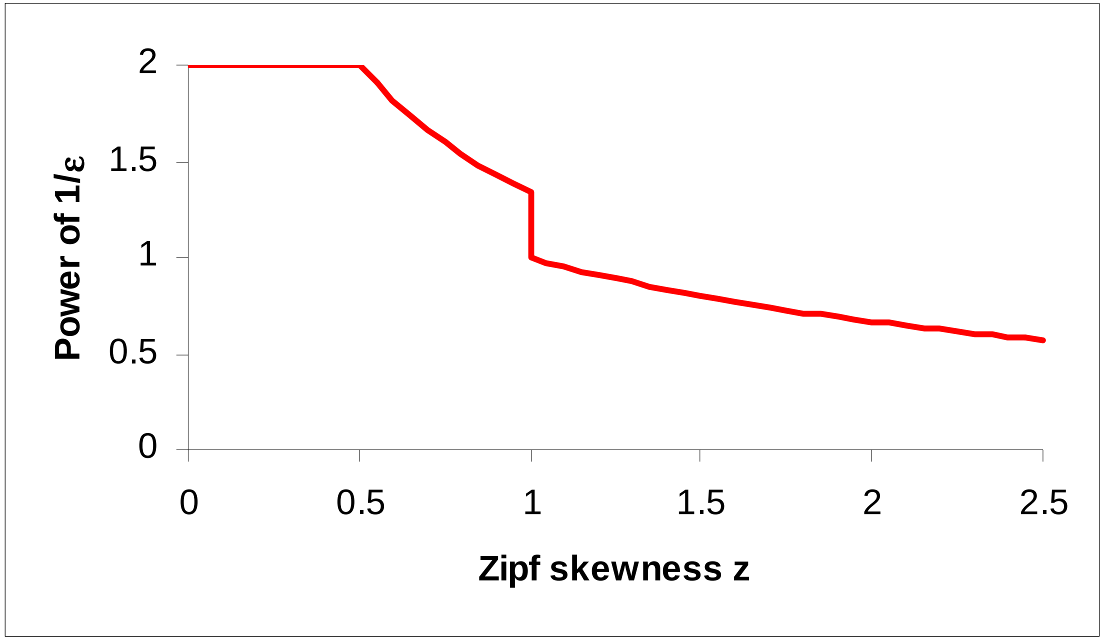
- Expectation of each CM^- estimate is F_2
- Variance $\leq 8F_2^2\ w^{-(1-2z)/2}$

Setting Var $= \varepsilon^2\ F_2^2$ and applying Chebyshev bound gives constant probability of $< \varepsilon F_2$ error.

Taking the median amplifies this to δ probability

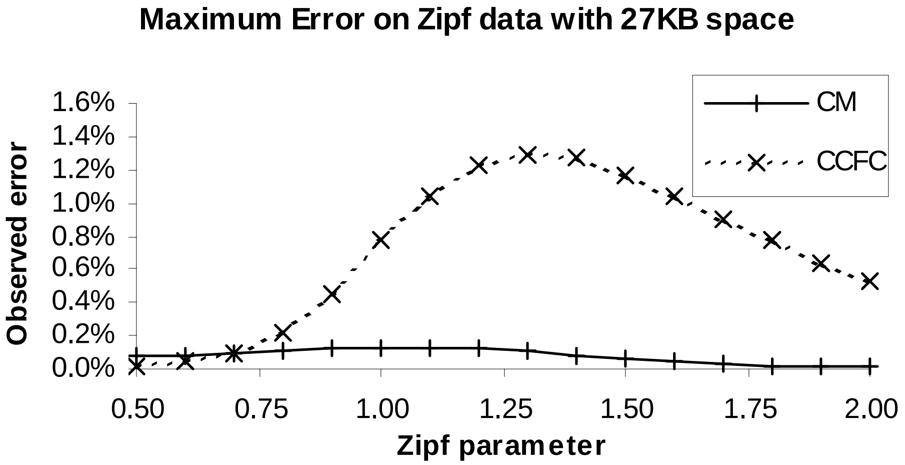
Total cost space $= O(\varepsilon^{-4/(1+2z)} \log 1/\delta)$, if $z>½$

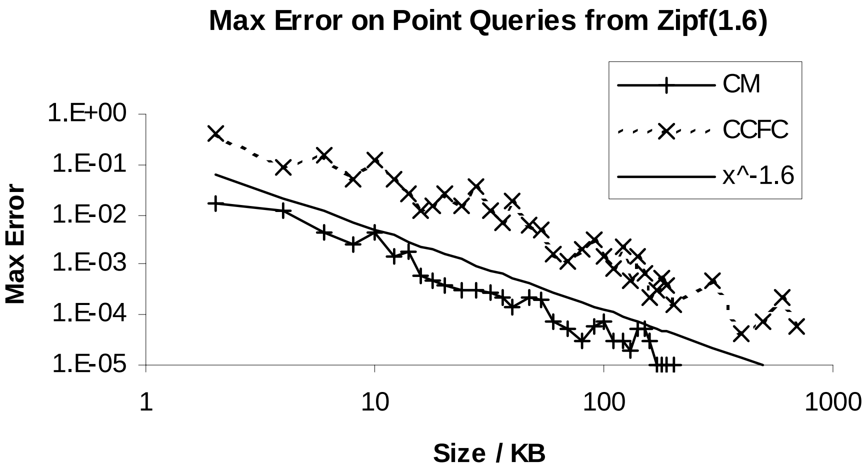
F_2 Estimation Summary



Skewness	Space Cost	Method
$z \leq ½$	$(1/\varepsilon)^2$	CM^-
$½ < z \leq 1$	$(1/\varepsilon)^{4/(1+2z)}$	CM^-
$1 < z$	$(1/\varepsilon)^{2/1+z}$	CM^+

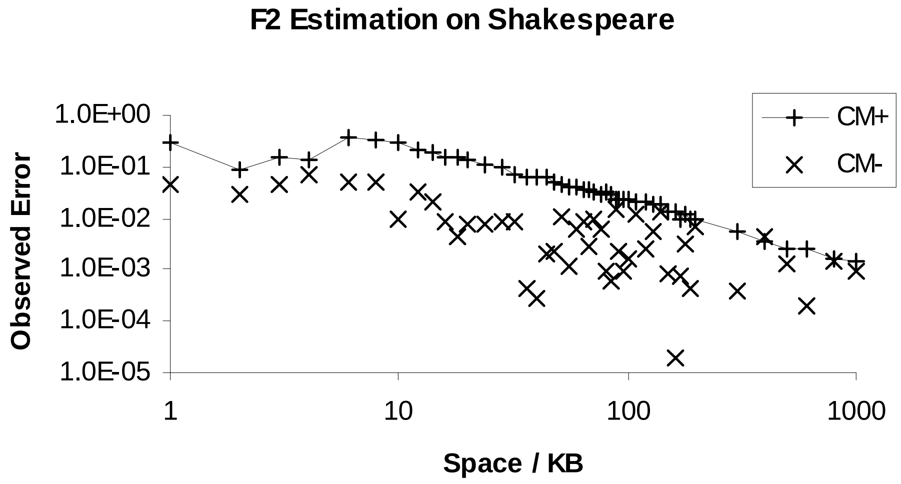
Experiments: Point Queries

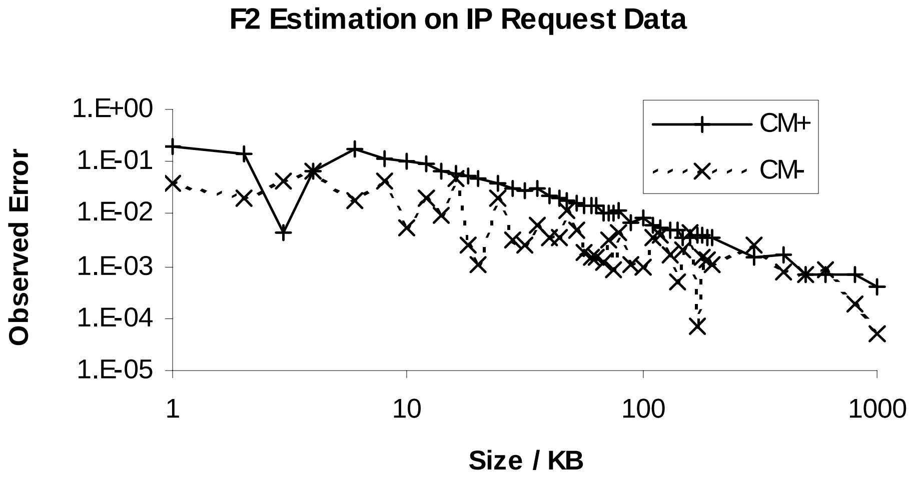




- On synthetic data, significantly outperforms worst error from comparable method [CCFC02]
- Error decays as space increases, as predicted

Experiments: F_2 Estimation





- Experiments on complete works of Shakespeare (5MB, z≈1.2) and IP traffic data (20MB, z≈1.3)
- CM^- seems to do better in practice on real data.

Experiments: Timing

Easily process 2-3million new items / second on standard desktop PC.

Queries are also fast

- point queries ≈ 1μs
- F_2 queries ≈ 100μs

Alternative methods are at least 40-50% slower.

Conclusions

By taking account of the skew inherent in most realistic data sources, can considerably improve results for summarizing and mining tasks.

Similar analysis is of interest for other mining tasks, eg. inner product / join size estimation.

Other structured domains: hierarchical domains, graph data etc.