Mergeable Summaries

Graham Cormode
graham@research.att.com

Pankaj Agarwal (Duke)
Zengfeng Huang (HKUST)
Jeff Philips (Utah)
Zheiwei Wei (HKUST)
Ke Yi (HKUST)
Summaries

- Summaries allow approximate computations:
  - Euclidean distance (Johnson-Lindenstrauss lemma)
  - Vector Inner-product, Matrix product (sketches)
  - Distinct items, Distinct Sampling (Flajolet-Martin onwards)
  - Frequent Items (Misra-Gries onwards)
  - Compressed sensing
  - Subset-sums (samples)
Mergeability

♦ Ideally, summaries are algebraic: associative, commutative
  – Allows arbitrary computation trees  
    (see also synopsis diffusion [Nath+04], MUD model) 
  – Distribution “just works”, whatever the architecture

♦ Summaries should have bounded size
  – Ideally, independent of base data size 
  – Or sublinear in base data (logarithmic, square root) 
  – Should not depend linearly on number of merges 
  – Rule out “trivial” solution of keeping union of input
Approximation Motivation

♦ Why use approximate when data storage is cheap?
  – Parallelize computation: partition and summarize data
    ■ Consider holistic aggregates, e.g. median finding
  – Faster computation (only work with summaries, not full data)
    ■ Less marshalling, load balancing needed
  – Implicit in some tools
    ■ E.g. Google Sawzall for data analysis requires mergability
  – Allows computation on data sets too big for memory/disk
    ■ When your data is “too big to file”

Mergeable Summaries
Models of Summary Construction

♦ Offline computation: e.g. sort data, take percentiles
♦ Streaming: summary merged with one new item each step
♦ One-way merge: each summary merges into at most one
  – Single level hierarchy merge structure
  – Caterpillar graph of merges
♦ Equal-size merges: can only merge summaries of same arity
♦ Full mergeability (algebraic): allow arbitrary merging schemes
  – Our main interest
Merging: sketches

♦ **Example:** most sketches (random projections) fully mergeable

♦ **Count-Min sketch of vector** $x[1..U]$:
  - Creates a small summary as an array of $w \times d$ in size
  - Use $d$ hash functions $h$ to map vector entries to $[1..w]$
  - Estimate $x[i] = \min_j CM[h_j(i), j]$
  - Error $2|x|_1/w$ with probability $1 - \frac{1}{2^d}$

♦ Trivially mergeable: $CM(x + y) = CM(x) + CM(y)$
Merging: sketches

♦ **Consequence** of sketch mergability:
  – Full mergability of quantiles, heavy hitters, F0, F2, dot product...
  – Easy, widely implemented, used in practice

♦ **Limitations** of sketch mergeability:
  – Probabilistic guarantees
  – May require discrete domain (ints, not reals or strings)
  – Some bounds are logarithmic in domain size
Deterministic Summaries for Heavy Hitters

- **Misra-Gries (MG) algorithm** finds up to $k$ items that occur more than $1/k$ fraction of the time in a stream [MG82]

- Keep $k$ different candidates in hand. For each item in stream:
  - If item is monitored, increase its counter
  - Else, if $< k$ items monitored, add new item with count 1
  - Else, decrease all counts by 1
Streaming MG analysis

- $N =$ total weight of input
- $M =$ sum of counters in data structure
- Error in any estimated count at most $(N-M)/(k+1)$
  - Estimated count a lower bound on true count
  - Each decrement spread over $(k+1)$ items: 1 new one and $k$ in MG
  - Equivalent to deleting $(k+1)$ distinct items from stream
  - At most $(N-M)/(k+1)$ decrement operations
  - Hence, can have “deleted” $(N-M)/(k+1)$ copies of any item
  - So estimated counts have at most this much error
Merging two MG Summaries

♦ Merging alg:
  - Merge two sets of $k$ counters in the obvious way
  - Take the $(k+1)$th largest counter $= C_{k+1}$, and subtract from all
  - Delete non-positive counters
  - Sum of remaining (at most $k$) counters is $M_{12}$

♦ This alg gives **full mergeability**:
  - Merge subtracts at least $(k+1)C_{k+1}$ from counter sums
  - So $(k+1)C_{k+1} \leq (M_1 + M_2 - M_{12})$
  - By induction, error is
    $$[((N_1-M_1) + (N_2-M_2) + (M_1+M_2-M_{12}))/ (k+1) = ((N_1+N_2) - M_{12})/ (k+1)$$
    (prior error) (from merge) (as claimed)
Other heavy hitter summaries

♦ The “SpaceSaving” (SS) summary also keeps \( k \) counters \(^{[MAA05]}\)
  – If stream item not in summary, overwrite item with least count
  – SS seems to perform better in practice than MG

♦ Surprising observation: SS is actually isomorphic to MG!
  – An SS summary with \( k+1 \) counters has same info as MG with \( k \)
  – SS outputs an upper bound on count, which tends to be tighter than the MG lower bound

♦ Isomorphism is proved inductively
  – Show every update maintains the isomorphism

♦ Immediate corollary: SS is fully mergeable
  – Just merge as if it were an MG structure
Quantiles (order statistics)

- Quantiles generalize median:
  - Exact answer: $\text{CDF}^{-1}(\phi)$ for $0 < \phi < 1$
  - Approximate version: tolerate answer in $\text{CDF}^{-1}(\phi - \varepsilon)\ldots\text{CDF}^{-1}(\phi + \varepsilon)$
  - Quantile summaries solve dual problem: estimate $\text{CDF}(x) \pm \varepsilon$

- **Hoeffding bound**: sample of size $O(1/\varepsilon^2 \log 1/\delta)$ suffices

- Fully mergeable samples of size $s$ via “Min-wise sampling”:
  - Pick a random “tag” for samples in $[0\ldots1]$
  - Merge two samples: keep the $s$ items with smallest tags
  - Tags of $O(\log N)$ bits suffice whp
    - Can draw tie-breaking bits when needed
One-way mergeable quantiles

Easy result: one-way mergeability in $O(1/\epsilon \log (\epsilon n))$

- Assume a streaming summary (e.g. [Greenwald Khanna 01])
- Extract an approximate CDF $F$ from the summary
- Generate corresponding distribution $f$ over $n$ items
- Feed $f$ to summary, error is bounded
- Limitation: repeatedly extracting/inserting causes error to grow
Equal-weight merging quantiles

♦ A classic result (Munro-Paterson ’78):
  – **Input**: two summaries of equal size $k$
  – **Base case**: fill summary with $k$ input items
  – Merge, sort summaries to get size $2k$
  – Take every other element

♦ **Deterministic bound**:
  – Error grows proportional to height of merge tree
  – Implies $O(1/\varepsilon \log^2 n)$ sized summaries (for $n$ known upfront)

♦ **Randomized twist**:
  – Randomly pick whether to take odd or even elements
Equal-sized merge analysis: absolute error

- Consider any interval $I$ over sample $S$ from a single merge
- Estimate $2|I \cap S|$ has absolute error at most 1
  - $|I \cap D|$ is even: $2|I \cap S| = |I \cap X|$ (no error)
  - $|I \cap D|$ is odd: $2|I \cap S| - |I \cap X| = \pm 1$
  - Error is zero in expectation (unbiased)
- Analyze total error after multiple merges inductively
  - Binary tree of merges
Equal-sized merge analysis: error at each level

- Consider $j$’th merge at level $i$ of $L^{(i-1)}$, $R^{(i-1)}$ to $S^{(i)}$
  - Estimate is $2^i | I \cap S^{(i)} |$
  - Error introduced by replacing $L$, $R$ with $S$ is
    \[ X_{i,j} = 2^i | I \cap S^i | - (2^{i-1} | I \cap (L^{(i-1)} \cup R^{(i-1)}) |) \]
    (new estimate) \quad (old estimate)
  - Absolute error $|X_{i,j}| \leq 2^{i-1}$ by previous argument

- Bound total error over all $m$ merges by summing errors:
  - $M = \sum_{i,j} X_{i,j} = \sum_{1 \leq i \leq m} \sum_{1 \leq j \leq 2^{m-i}} X_{i,j}$
  - Analyze sum of unbiased bounded variables via Chernoff bound
Equal-sized merge analysis: Chernoff bound

- Give unbiased variables $Y_j$ s.t. $|Y_j| \leq y_j$:
  \[ \Pr[\text{abs}(\sum_{1 \leq j \leq t} Y_j) > \alpha] \leq 2\exp(-2\alpha^2/\sum_{1 \leq j \leq t} (2y_j)^2) \]

- Set $\alpha = h 2^m$ for our variables:
  \[
  \begin{align*}
  &2\alpha^2/(\sum_i \sum_j (2 \max(X_{i,j})^2) \\
  &= 2(h2^m)^2 / (\sum_i 2^{m-i} \cdot 2^{2i}) \\
  &= 2h^2 2^{2m} / \sum_i 2^{m+i} \\
  &= 2h^2 / \sum_i 2^{i-m} \\
  &= 2h^2 / \sum_i 2^{-i} \\
  &\geq 2h^2
  \end{align*}
  \]

- From Chernoff bound, error probability is at most $2\exp(-2h^2)$
  - Set $h = \mathcal{O}(\log^{1/2} \delta^{-1})$ to obtain $1-\delta$ probability of success
Equal-sized merge analysis: finishing up

- Chernoff bound ensures absolute error at most $\alpha = h 2^m$
  - $m$ is number of merges $= \log(n/k)$ for summary size $k$
  - So error is at most $hn/k$
- Set size of each summary $k$ to be $O(h/\epsilon) = O(1/\epsilon \log^{1/2} 1/\delta)$
  - Guarantees give $\epsilon N$ error with probability $1-\delta$
  - Neat: naïve sampling bound gives $O(1/\epsilon^2 \log 1/\delta)$
  - Tightens randomized result of [Suri Toth Zhou 04]
Fully mergeable quantiles

- Use equal-size merging in a standard logarithmic trick:

- Merge two summaries as binary addition
- Fully mergeable quantiles, in \( O(1/\epsilon \log (\epsilon n) \log^{1/2} 1/\delta) \)
  - \( n = \) number of items summarized, not known a priori
- But can we do better?
Hybrid summary

♦ Observation: when summary has high weight, low order blocks don’t contribute much
  – Can’t ignore them entirely, might merge with many small sets

♦ Hybrid structure:
  – Keep top $O(\log 1/\varepsilon)$ levels as before
  – Also keep a “buffer” sample of (few) items
  – Merge/keep equal-size summaries, and sample rest into buffer
  – When buffer is “full”, extract points as a sample of lowest weight
Hybrid analysis (sketch)

- Keep the buffer (sample) size to $O(1/\varepsilon)$
  - Accuracy only $\sqrt{\varepsilon n}$
  - If buffer only summarizes $O(\varepsilon n)$ points, this is OK

- Analysis rather delicate:
  - Points go into/out of buffer, but always moving “up”
  - Number of “buffer promotions” is bounded
  - Similar Chernoff bound to before on probability of large error
  - Gives constant probability of accuracy in $O(1/\varepsilon \log^{1.5}(1/\varepsilon))$ space
Other Fully Mergeable Summaries

- $\epsilon$-approximations generalize quantiles for range queries in multiple dimensions
  - Generalize the “odd-even” trick to low-discrepancy colorings
  - $\epsilon$-approx for constant VC-dimension $v$ queries in $\tilde{O}(\epsilon^{-2v/(v+1)})$

- $\epsilon$-kernels in $d$-dimensional space approximately preserve the projected extent in any direction
  - $\epsilon$-kernels in $O(\epsilon^{(1-d)/2})$ for “fat” pointsets: bounded ratio between extents in any direction

- Equal-weight merging for k-median implicit from streaming
  - Implies $O(poly \: n)$ fully-mergeable summary via logarithmic trick
Open Problems

♦ Weight-based sampling over non-aggregated data
♦ Fully mergeable $\varepsilon$-kernels without assumptions
♦ More complex functions, e.g. cascaded aggregates
♦ Lower bounds for mergeable summaries
♦ Implementation studies (e.g. in Hadoop)