Scalable Histograms on Larger Probabilistic Data

Mingwang Tang and Feifei Li

School of Computing, University of Utah

September 24, 2014

New Challenges

- Large scale data size
- Distributed data sources
- Uncertainty

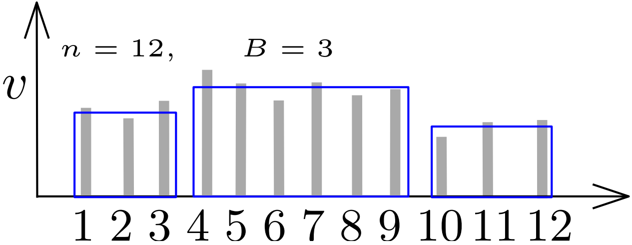
Data synopsis on large probabilistic data

- Scalable histograms on large probabilistic data

Histograms on deterministic data

- V-optimal histogram: Given a frequency vector $\overrightarrow{v} = \{v_1, \ldots, v_n\}$, where v_i is the frequency of item i in $[n]$, a space budget B, it seeks to minimize the SSE error:
-

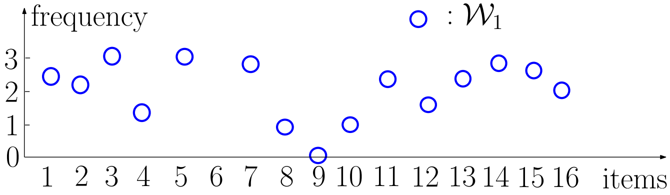
$$\min\{\sum_{k=1}^{B}\sum_{i=s_k}^{e_k}(v_i - \hat{b}_k)^2\} \tag{1}$$



- Optimal B-bucket histogram takes $O(Bn^2)$ time.

Probabilistic Database

- Probabilistic database $\mathcal{D}$ on domain $[n] = \{1, \ldots, n\}$

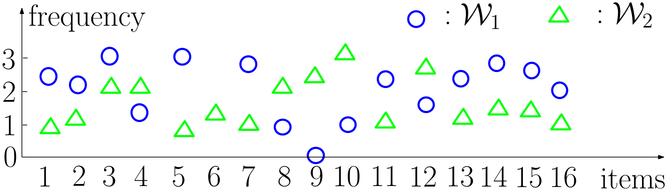


- $\mathcal{D} = \{g_1, g_2, \ldots, g_n\}$ where

$$g_i = \{(g_i(W), \Pr(W)) | W \in \mathcal{W}\} \tag{2}$$

Probabilistic Database

- Probabilistic database $\mathcal{D}$ on domain $[n] = \{1, \ldots, n\}$



- $\mathcal{D} = \{g_1, g_2, \ldots, g_n\}$ where

$$g_i = \{(g_i(W), \Pr(W)) | W \in \mathcal{W}\} \tag{2}$$

Probabilistic Data Models

$$g_i = \{(g_i(W), \Pr(W)) | W \in \mathcal{W}\}$$

Tuple Model

- Each tuple $t_j = \langle (t_{j1}, p_{j1}), \ldots, (t_{j\ell_j}, p_{j\ell_j}) \rangle$. Each t_{jk} is drawn from $[n]$ for $k \in [1, \ell_j]$.
- $1 - \sum_{k=1}^{\ell_j} p_{jk}$ specify the possibility that t_j generates no item.

t_1	$\{(1, 0.2), (3, 0.3), (7, 0.2)\}$
t_2	$\{(3, 0.3), (5, 0.1), (9, 0.4)\}$
t_3	$\{(3, 0.5), (10, 0.4), (13, 0.1)\}$
$\vdots$	$\vdots$
$t_{\vert\tau\vert}$	

Probabilistic Data Models

$$g_i = \{(g_i(W), \Pr(W)) | W \in \mathcal{W}\}$$

Tuple Model

- Each tuple $t_j = \langle (t_{j1}, p_{j1}), \ldots, (t_{j\ell_j}, p_{j\ell_j}) \rangle$. Each t_{jk} is drawn from $[n]$ for $k \in [1, \ell_j]$.
- $1 - \sum_{k=1}^{\ell_j} p_{jk}$ specify the possibility that t_j generates no item.

t_1	$\{(1, 0.2), (3, 0.3), (7, 0.2)\}$
t_2	$\{(3, 0.3), (5, 0.1), (9, 0.4)\}$
t_3	$\{(3, 0.5), (10, 0.4), (13, 0.1)\}$
$\vdots$	$\vdots$
$t_{\lvert\tau\rvert}$	

Probabilistic Data Models

$g_i = \{(g_i(W), \Pr(W)) | W \in \mathcal{W}\}$

Value Model

- Each tuple $t_j = \langle\, j : f_j = ((f_{j1}, p_{j1}), \ldots, (f_{j\ell_j}, p_{j\ell_j}))\,\rangle$, j is drawn from $[n]$.
- $\Pr(f_j = 0) = 1 - \sum_{k=1}^{\ell_j} p_{jk}$

t_1	$\{< 1, (50, 0.2), (7, 0.1), (14, 0.2) >\}$
t_2	$\{< 2, (6, 0.4), (7, 0.3), (15, 0.3) >\}$
t_3	$\{< 3, (10, 0.3), (15, 0.2), (20, 0.5) >\}$
$\vdots$	$\vdots$
t_n	

Probabilistic Data Models

$$g_i = \{(g_i(W), \Pr(W)) | W \in \mathcal{W}\}$$

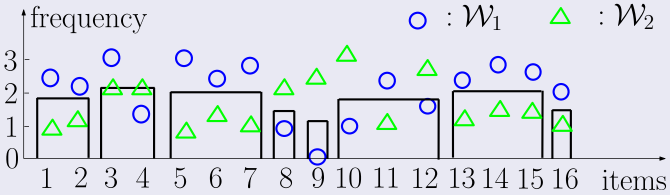
Value Model

- Each tuple $t_j = \langle\ j : f_j = ((f_{j1}, p_{j1}), \ldots, (f_{j\ell_j}, p_{j\ell_j}))\ \rangle$, j is drawn from $[n]$.
- $\Pr(f_j = 0) = 1 - \sum_{k=1}^{\ell_j} p_{jk}$

t_1	$\{< 1, (50, 0.2), (7, 0.1), (14, 0.2) >\}$
t_2	$\{< 2, (6, 0.4), (7, 0.3), (15, 0.3) >\}$
t_3	$\boxed{\{< 3, (10, 0.3), (15, 0.2), (20, 0.5) >\}}$
$\vdots$	$\vdots$
t_n	

Histograms on Probabilistic data

Possible world semantic



- g_i: frequency of item i becomes random variable across possible worlds

Expectation based histogram

$$\mathcal{H}(n, B) = \min\{\mathbf{E}_{\mathcal{W}}\left[\sum_{k=1}^{B}\sum_{j=s_k}^{e_k}(g_j - \widehat{b}_k)^2\right]\}.$$

- [ICDE09] G. Cormode et al., Histograms and wavelets on probabilistic data, ICDE 2009
- [VLDB09] G. Cormode et al., Probabilistic histograms for probabilistic data, VLDB 2009

Efficient computation of bucket error

- The optimal B bucket histogram takes $O(Bn^2)$ time.
- [TKDE10] shows that the minimal error of a bucket $b = (s, e, \widehat{b})$ is:

$$SSE(b, \widehat{b}) = \sum_{i=s}^{e} \mathbf{E}_{\mathcal{W}}[g_i^2] - \frac{1}{e - s + 1} \mathbf{E}_{\mathcal{W}}[\sum_{i=s}^{e} g_i]^2. \qquad (3)$$

by setting $\widehat{b} = \frac{1}{e-s+1} \mathbf{E}_{\mathcal{W}} \left[\sum_{i=s}^{e} g_i \right]$.

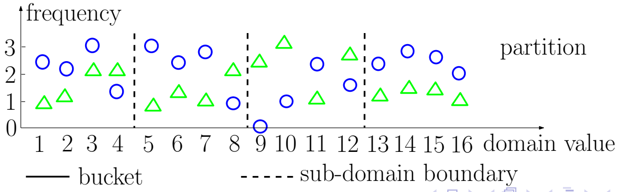
- Based on two precomputed arrays (A, B), $SSE(b, \widehat{b})$ can be computed in constant time.

- [TKDE10] G. Cormode et al., Histograms and wavelets on probabilistic data, TKDE 2010

PMERGE Method

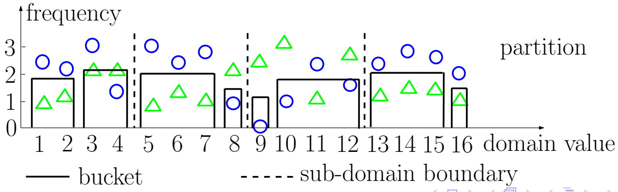
PMERGE method based on partition and merge principle

- Partition phase: partition the domain n into m sub-domain of equal size and compute the local optimal B buckets for each sub-domain.
- Merge phase: merge mB input buckets from the partition phase into B buckets.



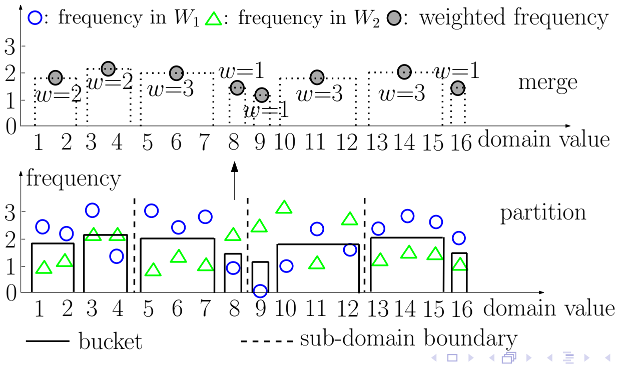
PMERGE method based on partition and merge principle

- Partition phase: partition the domain n into m sub-domain of equal size and compute the local optimal B buckets for each sub-domain.
- Merge phase: merge mB input buckets from the partition phase into B buckets.



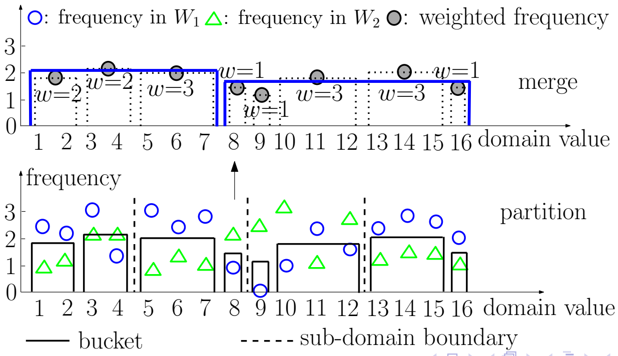
PMERGE method based on partition and merge principle

- Partition phase: partition the domain n into m sub-domain of equal size and compute the local optimal B buckets for each sub-domain.
- Merge phase: merge mB input buckets from the partition phase into B buckets.



PMERGE method based on partition and merge principle

- Partition phase: partition the domain n into m sub-domain of equal size and compute the local optimal B buckets for each sub-domain.
- Merge phase: merge mB input buckets from the partition phase into B buckets.

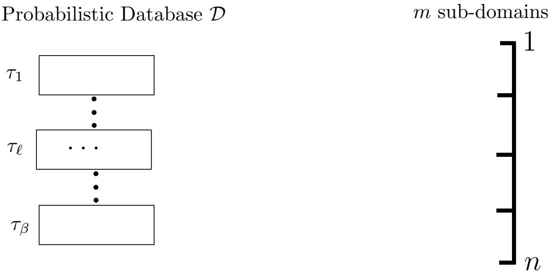


Recursive Merging Method

- PMERGE method:
  - Approximation quality: PMERGE produces a $\sqrt{10}$ approximation in $O(N + Bn^2/m + B^3m^2)$ time.
- Recursive merging (RPMERGE):
  - Partition $[n]$ into m^ℓ subdomains, producing Bm^ℓ.
  - Using ℓ iterations and each iteration reduce the domain size by a factor of m.
  - Takes $O(N + B\frac{n^2}{m^\ell} + B^3\sum_{i=1}^{\ell} m^{(i+1)})$ time and the RPMERGE method gives a $10^{\frac{\ell}{2}}$ approximation of the optimal B-buckets histogram found by OPTHIST.
- In practice, PMERGE and RPMERGE always provide close to optimal approximation quality as shown in our experiments.

Distributed and Parallel PMERGE

- Partition phase in the distributed environment

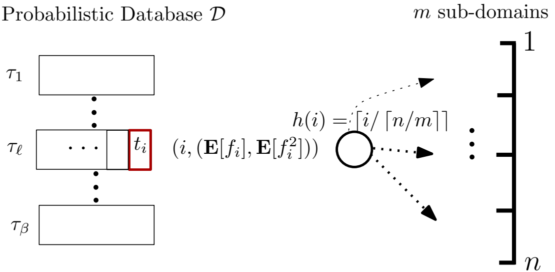


Communication cost

- Computing A_k, B_k arrays in the partition phase
  - Tuple model: $O(\beta n)$ bytes.
  - Value model: $O(n)$ bytes.
- $O(Bm)$ bytes in the merge phase for both models.

Distributed and Parallel PMERGE

- Partition phase in the distributed environment

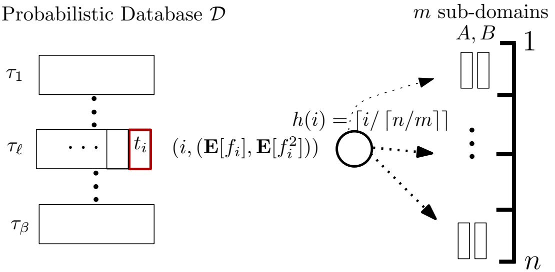


Communication cost

- Computing A_k, B_k arrays in the partition phase
  - Tuple model: $O(\beta n)$ bytes.
  - Value model: $O(n)$ bytes.
- $O(Bm)$ bytes in the merge phase for both models.

Distributed and Parallel PMERGE

- Partition phase in the distributed environment



Communication cost

- Computing A_k, B_k arrays in the partition phase
  - Tuple model: $O(\beta n)$ bytes.
  - Value model: $O(n)$ bytes.
- $O(Bm)$ bytes in the merge phase for both models.

PMERGE Based on Sampling

Sampling A, B arrays in the partition phase

$$A_k[j] = \sum_{i=1}^{i} \mathbf{E}[f_j^2], B_k[j] = \sum_{i=1}^{j} \mathbf{E}[f_i]$$

Estimate A_k, B_k arrays using quantile sampling

$E[f_i]$	2	3	5	9	$\cdots$
item:	1	2	3	4	$\ldots$

PMERGE Based on Sampling

Sampling A, B arrays in the partition phase

$$A_k[j] = \sum_{i=1}^{i} \mathbf{E}[f_j^2], B_k[j] = \sum_{i=1}^{j} \mathbf{E}[f_i]$$

Estimate A_k, B_k arrays using quantile sampling

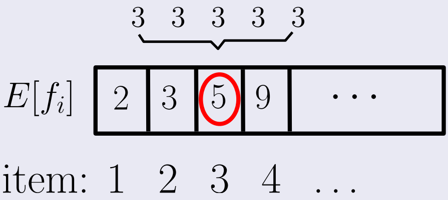
$E[f_i]$	2	3	5	9	$\cdots$
item:	1	2	3	4	$\ldots$

PMERGE Based on Sampling

Sampling A, B arrays in the partition phase

$$A_k[j] = \sum_{i=1}^{i} \mathbf{E}[f_j^2],\ B_k[j] = \sum_{i=1}^{j} \mathbf{E}[f_i]$$

Estimate A_k, B_k arrays using quantile sampling

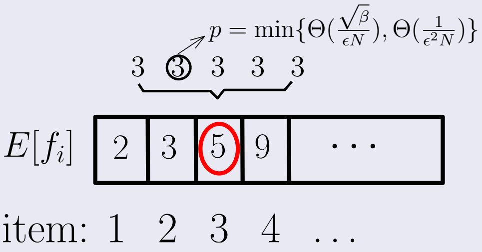


PMERGE Based on Sampling

Sampling A, B arrays in the partition phase

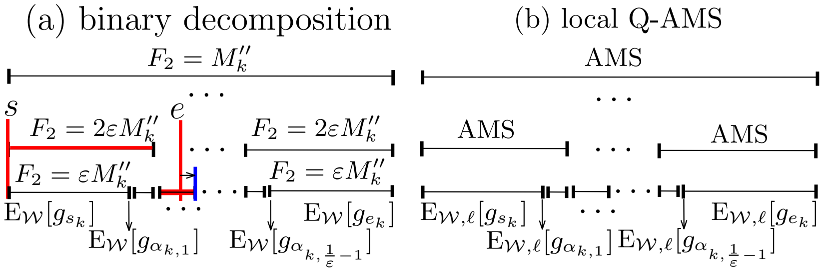
$$A_k[j] = \sum_{i=1}^{i} \mathbf{E}[f_j^2],\ B_k[j] = \sum_{i=1}^{j} \mathbf{E}[f_i]$$

Estimate A_k, B_k arrays using quantile sampling



Tuple model A, B arrays

- Estimate $F_2 = \sum_{i=s_k}^{j}(\sum_{\ell=1}^{\beta} \mathbf{E}_{\mathcal{W},\ell}[g_i])^2$ using AMS Sketch techniques and binary decomposition of domain $[s_k, e_k]$.



Outline

1. Optimal B-buckets Histograms
2. Approximate Histograms
3. PMERGE Based on Sampling
4. Experiments

Experiment setup

- Generate tuple model and the value model dataset using the client id filed of 1998 WorldCup dataset and atmospheric measurements from the SAMOS project.
- The default experimental parameters:

Symbol	Definition	Default Value
B	number of buckets	400
n	domain size	100k (600k)
ℓ	depth of recursions	2

Running time:

- n: domain size

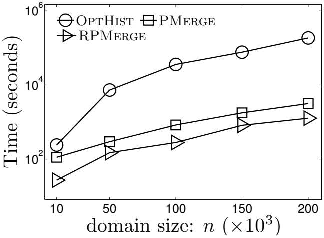


Figure: Tuple Model

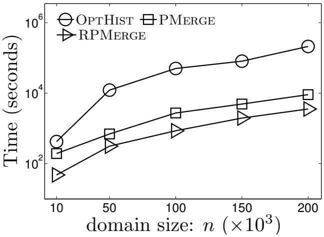


Figure: Value Model

Approximation Ratio:

- n: domain size

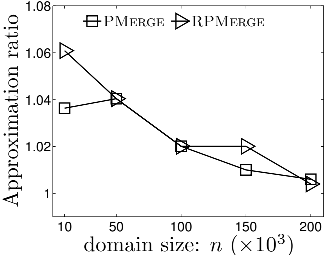


Figure: Tuple Model

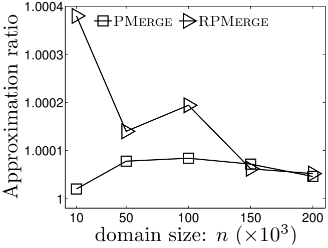


Figure: Value Model

- n: domain size

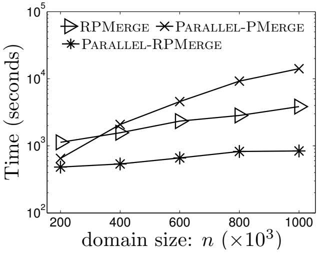


Figure: Tuple Model

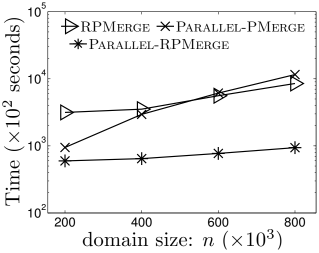


Figure: Value Model

Conclusion & Future work

- Conclusion
  - Novel approximation methods for constructing scalable histograms on large probabilistic data.
  - The quality of the approximate histograms are almost as good as the optimal histogram in practice.
  - Extended the techniques to distributed and parallel settings to further improve scalability.
- Future work
  - extend our study to probabilistic histograms with pdf bucket representatives and handle histogram of other error metrics

Thank You

Q and A