Scalable Histograms on Large Probabilistic Data

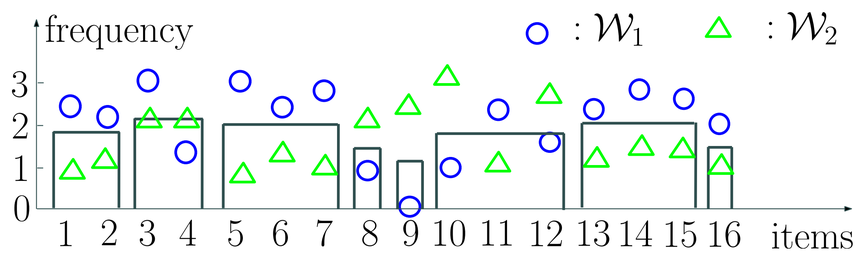
Mingwang Tang & Feifei Li
School of Computing, University of Utah
tang,lifeifei@cs.utah.edu

Abstract

Histogram construction is a fundamental problem in data management, and a good histogram supports numerous mining operations. Recent work has extended histograms to probabilistic data. However, constructing histograms for probabilistic data can be extremely expensive, and existing studies suffer from limited scalability. This work designs novel approximation methods to construct scalable histograms on probabilistic data. We show that our methods provide constant approximations compared to the optimal histograms produced by the state-of-the-art in the worst case. We also extend our methods to parallel and distributed settings so that they can run gracefully in a cluster of commodity machines. We introduced novel synopses to reduce communication cost when running our methods in such settings. Extensive experiments on large real data sets have demonstrated the superb scalability and efficiency achieved by our methods, when compared to the state-of-the-art methods. They also achieved excellent approximation quality in practice.

Problem Formulation

- Histograms on probabilistic data



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

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

$$\mathcal{H}(i,j) = \min_{1 \le \ell < i} \mathcal{H}(\ell, j-1) + \min_{\widehat{b}} (\ell+1, i, \widehat{b})$$

- Efficient computation of bucket error

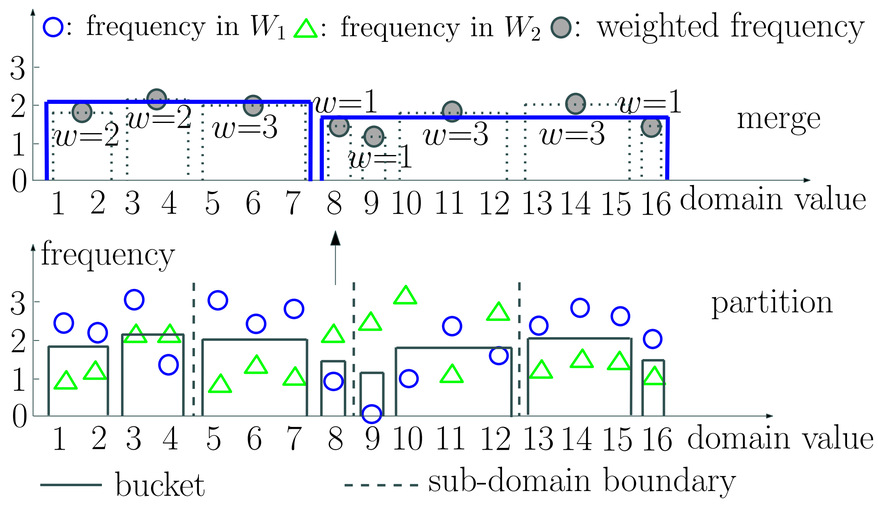
$$SSE(s,e,\widehat{b}) = \sum_{i=s}^{e} \mathbf{E}_{\mathcal{W}}[g_i^2] - \frac{1}{e-s+1} \mathbf{E}_{\mathcal{W}}[\sum_{i=s}^{e} g_i]^2. \quad (2)$$

$$A[e] = \sum_{i=1}^{e} \mathbf{E}_{\mathcal{W}}[g_i^2] = \sum_{i=1}^{e} (\mathbf{Var}_{\mathcal{W}}[g_i] + \mathbf{E}_{\mathcal{W}}[g_i]^2) \quad B[e] = \sum_{i=1}^{e} \mathbf{E}_{\mathcal{W}}[g_i] \quad (3)$$

Approximate Histograms

PMERGE method

- 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 takes $O(N + Bn^2/m + B^3m^2)$ time.
- PMERGE produces a $\sqrt{10}$ approximation ratio with respect to the ℓ_2 distance.



Recursive merging method

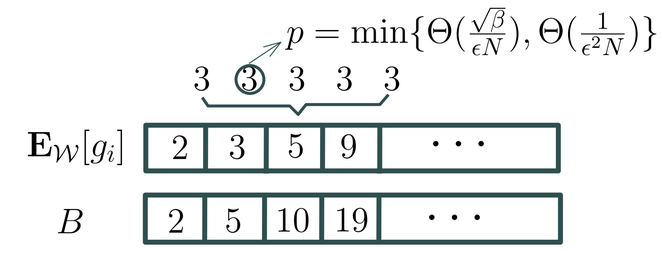
- Partition $[n]$ into m^ℓ subdomains.
- Each merge step in current iteration merge mB buckets into B buckets of next iteration (weighted V-optimal histogram idea).
- ℓ iterations in the merge phase.
- Using $O(N + B\frac{n^2}{m^\ell} + B^3 \sum_{i=1}^{\ell} m^{(i+1)})$ time, the RPMERGE method produces a $10^{\frac{\ell}{2}}$ approximation ratio.

Distributed and parallel PMERGE

- Split the database $\mathcal{D}$ into β chunks $\{\tau_1, \ldots, \tau_\beta\}$ for both models.
- Partition domain $[n]$ into m subdomains.
- Computing A_k, B_k arrays in the partition phase
- Communication cost
  - 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

- Quantile sampling on value model using $O(\min\{m\sqrt{\beta}/\epsilon, m/\epsilon^2\})$ bytes.



- AMS sketches on binary decomposition of domain on tuple model

$$A_k[j] = \sum_{\ell=1}^{\beta} \sum_{i=s_k}^{j} \mathbf{Var}_{\mathcal{W},\ell}[g_i] + \sum_{i=s_k}^{j} (\sum_{\ell=1}^{\beta} \mathbf{E}_{\mathcal{W},\ell}[g_i])^2$$

$$B_k[j] = \sum_{\ell=1}^{\beta} \sum_{i=s_k}^{j} \mathbf{E}_{\mathcal{W},\ell}[g_i]. \quad (4)$$

Experiment

Datasets

- 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: $n = 100k$ ($n = 600k$), $B = 400$ and $\ell = 2$.

Results

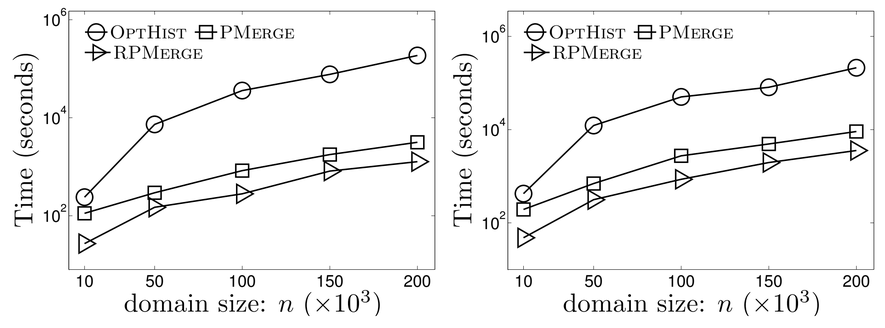


Figure 1: Running time on tuple model and value model

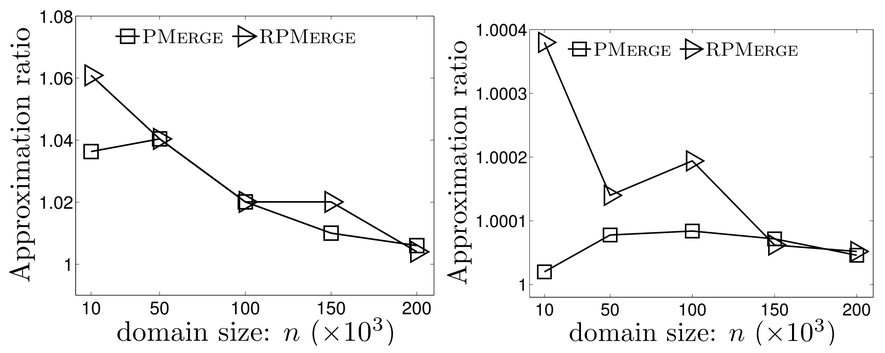


Figure 2: Approximation ratio on tuple model and value model

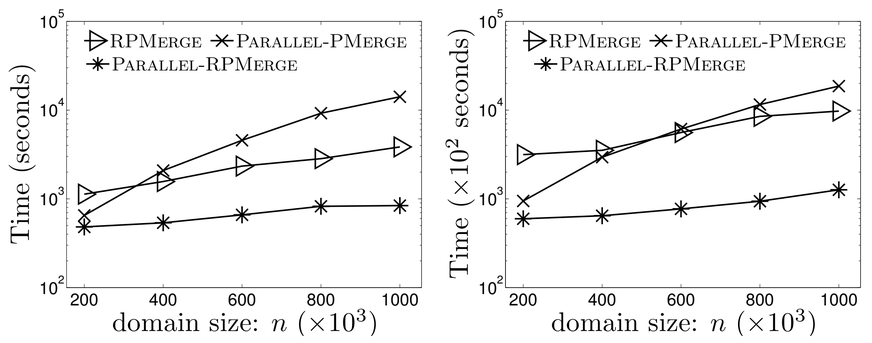


Figure 3: Running time on large datasets

Conclusions

- 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.