Scalable Histograms on Larger Probabilistic Data

Mingwang Tang and Feifei Li

September 24, 2014
Introduction

New Challenges

- Large scale data size
- Distributed data sources
- Uncertainty

Data synopsis on large probabilistic data

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

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

Optimal B-bucket histogram takes $O(Bn^2)$ time.
Probabilistic database $\mathcal{D}$ on domain $[n] = \{1, \ldots, n\}$

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

$$g_i = \{(g_i(W), \Pr(W))| W \in \mathcal{W}\} \quad (2)$$
Probabilistic database $D$ on domain $[n] = \{1, \ldots, n\}$

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

$$g_i = \{(g_i(W), \Pr(W)) | W \in \mathcal{W}\}$$  \hspace{1cm} (2)
\[ g_i = \{(g_i(W), \Pr(W)) | W \in \mathcal{W}\} \]

### 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.

| \( t_1 \)     | \( \{(1, 0.2), (3, 0.3), (7, 0.2)\} \) |
| \( t_2 \)     | \( \{(3, 0.3), (5, 0.1), (9, 0.4)\} \) |
| \( t_3 \)     | \( \{(3, 0.5), (10, 0.4), (13, 0.1)\} \) |
| \( \cdots \)  | \( \cdots \) |
| \( t_{|\tau|} \) | \( \cdots \) |
\[ g_i = \{(g_i(W), \Pr(W)) | W \in W\} \]

**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.

<table>
<thead>
<tr>
<th>( t_1 )</th>
<th>( {(1, 0.2), (3, 0.3), (7, 0.2)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_2 )</td>
<td>( {(3, 0.3), (5, 0.1), (9, 0.4)} )</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>( {(3, 0.5), (10, 0.4), (13, 0.1)} )</td>
</tr>
</tbody>
</table>
| \( 
\)             | \( \cdot \)                               |
| \( \vdots \)      | \( \cdot \)                               |
| \( t_{|\tau|} \)  | \( \cdot \)                               |
\[ g_i = \{(g_i(W), \Pr(W)) \mid W \in \mathcal{W}\} \]

**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 \([n]\).
- \( \Pr(f_j = 0) = 1 - \sum_{k=1}^{\ell_j} p_{jk} \)

| \( t_1 \) | \( \{ < 1, (50, 0.2), (7, 0.1), (14, 0.2) > \} \) |
| \( t_2 \) | \( \{ < 2, (6, 0.4), (7, 0.3), (15, 0.3) > \} \) |
| \( t_3 \) | \( \{ < 3, (10, 0.3), (15, 0.2), (20, 0.5) > \} \) |
| \( \cdots \) | \( \cdots \) |
| \( t_n \) | \( \cdots \) |
\[ g_i = \{(g_i(W), \Pr(W)) | W \in \mathcal{W}\} \]

**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 \([n]\).
- \( \Pr(f_j = 0) = 1 - \sum_{k=1}^{\ell_j} p_{jk} \)

| \( t_j \) | \{ <1, (50, 0.2), (7, 0.1), (14, 0.2) > \} | \{ <2, (6, 0.4), (7, 0.3), (15, 0.3) > \} | \{ <3, (10, 0.3), (15, 0.2), (20, 0.5) > \} | \( \cdots \) | \( \cdots \) |
|---|---|---|---|---|
| \( t_1 \) | \{ <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) > \} | \| \| |
| \( \cdots \) | \| \| |
| \( t_n \) | \| \| |
Histograms on Probabilistic data

Possible world semantic

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

Expectation based histogram

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

- [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
The optimal $B$ bucket histogram takes $O(Bn^2)$ time.

[TKDE10] shows that the minimal error of a bucket $b = (s, e, \hat{b})$ is:

$$SSE(b, \hat{b}) = \sum_{i=s}^{e} E_{\mathcal{W}}[g_i^2] - \frac{1}{e - s + 1} E_{\mathcal{W}}[\sum_{i=s}^{e} g_i]^2. \quad (3)$$

by setting $\hat{b} = \frac{1}{e - s + 1} E_{\mathcal{W}} [\sum_{i=s}^{e} g_i]$.

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

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

![Diagram showing partition and bucket distribution](attachment://diagram.png)
**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**

**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.

![Diagram showing partition and merge process](image)
**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.
**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 $m^\ell$ subdomains, producing $Bm^\ell$.
  - Using $\ell$ 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.
• Partition phase in the distributed environment

Probabilistic Database $\mathcal{D}$

$\tau_1$

$\tau_\ell$

$\tau_\beta$

$m$ sub-domains

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.
- Partition phase in the distributed environment

\[ h(i) = \left\lceil \frac{i}{\left\lceil \frac{n}{m} \right\rceil} \right\rceil \]

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.
Partition phase in the distributed environment

Probabilistic Database $\mathcal{D}$

\[ h(i) = \left\lceil \frac{i}{\lceil n/m \rceil} \right\rceil \]

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

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

Estimate $A_k, B_k$ arrays using quantile sampling

$$E[f_i] \quad 2 \quad 3 \quad 5 \quad 9 \quad \cdots$$

item: 1 2 3 4 ...
**Sampling** $A, B$ arrays in the partition phase

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

**Estimate** $A_k, B_k$ arrays using quantile sampling

<table>
<thead>
<tr>
<th>$E[f_i]$</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>9</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>item:</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>...</td>
</tr>
</tbody>
</table>
Sampling $A$, $B$ arrays in the partition phase

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

Estimate $A_k$, $B_k$ arrays using quantile sampling

$E[f_i]$:

\[
\begin{array}{cccccc}
3 & 3 & 3 & 3 & 3 & 3 \\
\end{array}
\]

item: 1 2 3 4 ...
Sampling $A, B$ arrays in the partition phase

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

Estimate $A_k, B_k$ arrays using quantile sampling

\[ p = \min\{\Theta(\frac{\sqrt{\beta}}{\epsilon N}), \Theta(\frac{1}{\epsilon^2 N})\} \]

\[
\begin{array}{cccc}
3 & 3 & 3 & 3 \\
\end{array}
\]

\[
\begin{array}{cccc}
E[f_i] & 2 & 3 & 5 & 9 & \cdots \\
\end{array}
\]

item: 1 2 3 4 ...
Tuple model $A, B$ arrays

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

(a) binary decomposition

\[ F_2 = M_k'' \]

(b) local Q-AMS

AMS

\[ E_{W,\ell}[g_{\alpha_k,1}] \]

\[ E_{W,\ell}[g_{\alpha_k,\frac{1}{\epsilon}-1}] \]
1 Optimal $B$-buckets Histograms

2 Approximate Histograms

3 Pmerge Based on Sampling

4 Experiments
Generate tuple model and the value model dataset using the client id field of 1998 WorldCup dataset and atmospheric measurements from the SAMOS project.

The default experimental parameters:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>number of buckets</td>
<td>400</td>
</tr>
<tr>
<td>$n$</td>
<td>domain size</td>
<td>100k (600k)</td>
</tr>
<tr>
<td>$\ell$</td>
<td>depth of recursions</td>
<td>2</td>
</tr>
</tbody>
</table>
Running time:

- $n$: domain size

**Figure: Tuple Model**

**Figure: Value Model**
Approximation Ratio:

- $n$: domain size

Figure: Tuple Model

Figure: Value Model
Running time on large scale probabilistic data

- $n$: domain size

**Figure:** Tuple Model

**Figure:** Value Model
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