STRATIFIED SAMPLING MEETS MACHINE LEARNING

KEVIN LANG
KONSTANTIN SHMAKOV
EDO LIBERTY
Yahoo Mobile Developer Suite for your apps
Measure, monetize, advertise and improve your apps with Yahoo tools.

Sign Up
Log in to Flurry

Flurry Analytics
Get free insights from the industry’s leading mobile app analytics tool.

Yahoo App Publishing
Monetize your app with native and video ads from Yahoo, Flurry, and BrightRoll advertisers.

Yahoo App Marketing
Reach your target audience with the Gemini native and video marketplace.
Introducing the all new Flurry Analytics

Measure and analyze activity across your app portfolio to answer your hardest questions and optimize your app experience.

Sign Up  Log in to Flurry  Documentation

Flurry Grows with You

As your business grows, we are committed to supporting you at scale for free. Empower all your employees to leverage analytics for data driven decision making.

Product Features
Flurry Analytics

Apps with Flurry SDK
Flurry Analytics

Events
Examples:
- Number of event of a certain type
- Number of unique user
- Number of unique users in a specific day
- Total time spent in certain geo
- Average $ spent by age
SAMPLING

Challenges:
1. The data is very large. Computing $\sum_i q(u_i)$ exactly is too costly.
2. The function $q(\cdot)$ is user specified and completely unconstrained.

Good News:
And approximate answer is acceptable (if the error is small)

Solution:
Estimate the answer on a random subset of the records
NOTATIONS

• $q_i := q(u_i)$ for brevity

• $y := \sum_i q_i$ the exact answer for the query $q$

• $p_i$ the probability of choosing record $i$

• $S$ the set of sampled records, each chosen with probability $p_i$

• $\tilde{y} = \sum_{i \in S} q_i/p_i$ the Horvitz-Thompson estimator for $y$
PROPERTIES

- $\mathbb{E}[\tilde{y} - y] = 0$ Horvitz-Thompson estimator is unbiased

- $\sigma[\tilde{y} - y] \leq y \sqrt{1/(\zeta \cdot \text{card}(q))}$ its standard deviation isn’t large

- $\zeta = \min_i p_i$  
- $\text{card}(q) := \sum |q_i| / \max |q_i|$  

- $\Pr[|\tilde{y} - y| \geq \varepsilon y] \leq e^{-O(\varepsilon^2 \zeta \cdot \text{card}(q))}$ probability for large error is small

\[
\text{card}(q) \sim \Omega(n) \quad \rightarrow \quad |S| \sim 1/\varepsilon^2
\]

(Olken, Rotem and Hellerstein 1986, and 1990) application to databases
(Acharya, Gibbons, Poosala 2000) uniform sampling is best in the worst case
STRATIFIED SAMPLING

• Sample = 100,000 US individuals.
• Query = Republicans vs. Democrats in American Samoa?

American Samoa is 0.02% of population

Only ~20 from Samoa in the sample

Survey error is very large!

If $\text{card}(q)$ is small, $|S|$ must be large

• Sample different strata (e.g. US territories) with different probabilities.

(Neyman, Jerzy 1934)
DBLP EXAMPLE

Choosing the right strata is hard!

- 2,101,151 papers
- 1000 most populous venues
- Query example
  - title contains "learning" and # authors <= 3
  - title contains "mechanism" and year > 2004

What is the right stratification here?

- Stratifying by venue made things worse!
- Stratifying by year was better but still worse than uniform sampling.
SAMPLING, STRATIFICATION, AND DATABASES

• Design strata that minimize worst case variance on possible queries
• Linearly combine strata based on record features
• Combine stratifies and uniform sampling: Congressional Sampling
  o Acharya, Gibbons, Poosala 2000:

**Important idea:** consider past queries to the database!

• Each stratum is a set of records that agree on all queries
  o Chaudhuri, Das and Narasayya 2007: optimize for the query log

• Split to two strata, per each query. Take linear combinations
  o Joshi, Jermaine, 2008: linear combinations of stratified probabilities
OUR APPROACH

• Assume queries are drawn from a distribution $\mathbb{Q}$
• Use the query log $Q$ as a “training set” (assumed w.r.t. $\mathbb{Q}$)
• Allow each record to be sampled with a different probability $p_i$
• Minimize the Risk $\mathbb{E}[(\tilde{y} - y)^2]$  
• This translates to $\mathbb{E}_{q \sim \mathbb{Q}} \sum_{i} q_i^2 (1/p_i - 1)$
OUR APPROACH

• ERM: Minimize \( \sum_{q \in Q} \sum_{i} q_i^2 \left( \frac{1}{p_i} - 1 \right) \)

• Sampling budget \( \sum_{i} p_i c_i \leq B \) \( ( \sum_{i} c_i \ll B ) \)

• Regularization \( \forall i \ p_i \in [\zeta, 1] \) \( ( \zeta \leq B / \sum_{i} c_i ) \)
OUR APPROACH

• Solve with Lagrange multipliers

\[
\max_{\alpha, \beta, \gamma} \left[ \frac{1}{|Q|} \sum_{q \in Q} \sum_{i} q_i^2 \left( \frac{1}{p_i} - 1 \right) - \sum_{i} \alpha_i (p_i - \zeta) \right. \\
\left. - \sum_{i} \beta_i (1 - p_i) - \gamma (B - \sum_{i} p_i c_i) \right]
\]

• By KKT conditions

\[ p_i = \zeta \quad \text{or} \quad p_i \propto \sqrt{\frac{1}{c_i} \frac{1}{|Q|} \sum_{q \in Q} q_i^2} \quad \text{or} \quad p_i = 1 \]
1: **input**: training queries $Q$, budget $B$, record costs $c_i$, regularization factor $\eta \in [0, 1]$

4: $\zeta = \eta \cdot \left( \frac{B}{\sum c_i} \right)$

5: $\forall i \quad z_i = \sqrt{\frac{1}{c_i} \frac{1}{|Q|} \sum_{q \in Q} q_i^2}$

6: Binary search for $\lambda$ satisfying $\sum_i c_i \text{CLIP}_1^1(\lambda z_i) = B$

7: **output**: $\forall i \quad p_i = \text{CLIP}_1^1(\lambda z_i)$

$$
\text{Risk}(p) \leq \text{Risk}(p^*) \left( 1 + O \left( \text{skew} \sqrt{\frac{\log(n/\delta)}{|Q|}} \right) \right)
$$

**Alg’ Risk**  **Best Risk**  **Database “badness”**  **Training Set size**
## RESULTS

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Cube 0.1</th>
<th>DBLP 0.01</th>
<th>YAM+ 0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform Sampling</td>
<td>0.664</td>
<td>0.229</td>
<td>0.104</td>
</tr>
<tr>
<td>Neyman Allocation</td>
<td>0.643</td>
<td>0.640</td>
<td>0.286</td>
</tr>
<tr>
<td>Regularized Neyman</td>
<td>0.582</td>
<td>0.228</td>
<td>0.102</td>
</tr>
<tr>
<td>ERM-η, small training set</td>
<td>0.637</td>
<td>0.222</td>
<td>0.096</td>
</tr>
<tr>
<td>ERM-ρ, small training set</td>
<td>0.623</td>
<td>0.213</td>
<td>0.092</td>
</tr>
<tr>
<td>ERM-η, large training set</td>
<td>0.233</td>
<td>0.182</td>
<td>0.064</td>
</tr>
<tr>
<td>ERM-ρ, large training set</td>
<td>0.233</td>
<td>0.179</td>
<td>0.059</td>
</tr>
</tbody>
</table>
RESULTS

DBLP Dataset

Uniform Sampling p = 1/100
5000 Training Queries
10000 Training Queries
20000 Training Queries
40000 Training Queries

Expected Error

[weaker...] Value of Regularization Parameter Eta [...stronger]
RESULTS

Cube Dataset

Uniform Sampling $p = 1/10$
- 50 Training Queries
- 100 Training Queries
- 200 Training Queries
- 800 Training Queries
- 6400 Training Queries

Expected Error vs. [weaker... Value of Regularization Parameter Eta [...stronger]}

[Graph showing expected error for different numbers of training queries for a cube dataset with uniform sampling.]
RESULTS

YAM+ Dataset

- Uniform $p = \frac{1}{100}$
- 50 Training Queries
- 100 Training Queries
- 200 Training Queries
- 400 Training Queries
- All Training Queries

Expected Error vs. Value of Regularization Parameter Eta
RESULTS

YAM+ Dataset

Expected Error vs. Numeric Cardinality of Test Query

Regularized ERM
Uniform Sampling
RESULTS

YAM+ Dataset

![Graph showing the probability distribution of average error for Uniform Sampling and Regularized ERM on the YAM+ dataset. The x-axis represents the average error, ranging from 0 to 0.4, and the y-axis represents the probability (rescaled), ranging from 0 to 0.1.]

- **Uniform Sampling**
- **Regularized ERM**

The graph compares the performance of Uniform Sampling and Regularized ERM on the YAM+ dataset, indicating that Regularized ERM has a lower average error compared to Uniform Sampling.
RESULTS

YAM+ Dataset

'Expected Error' vs 'Sampling Rate' = Budget / (Total Cost)

Uniform Sampling
Regularized ERM
RESULTS

Cube Dataset

Expected Error

Numeric Cardinality of Test Query

ERM

Uniform Sampling