LSH Ensemble: Internet-Scale Domain Search

Erkang (Eric) Zhu+, Fatemah Nargesian+, Ken Q. Pu*, Renée J. Miller+ 

+ University of Toronto * University of Ontario Institute of Technology
Alice, a data scientist, wishes to understand the factors affecting companies’ funding to the universities.
**Scenario 1:** she has the two tables at hand.

**Solution 1:** join the two tables on the Industry Partners and Company columns.

<table>
<thead>
<tr>
<th>Industry Partners</th>
<th>Province</th>
<th>Grant Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>Ontario</td>
<td>...</td>
</tr>
<tr>
<td>Imperial Oil Ltd</td>
<td>Alberta</td>
<td>...</td>
</tr>
<tr>
<td>Hydro-Quebec</td>
<td>Quebec</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Company</th>
<th>CRA Tax ID</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>C0112</td>
<td>...</td>
</tr>
<tr>
<td>Imperial Oil Ltd</td>
<td>C1234</td>
<td>...</td>
</tr>
<tr>
<td>IBM Canada Ltd</td>
<td>C5678</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
**Scenario 2:** she has only one table, and a **repository** of MANY tables - manual inspection is not preferred.

**Solution 2:** requires a search engine for relevant datasets.

<table>
<thead>
<tr>
<th>Industry Partners</th>
<th>Province</th>
<th>Grant Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>Ontario</td>
<td>...</td>
</tr>
<tr>
<td>Imperial Oil Ltd</td>
<td>Alberta</td>
<td>...</td>
</tr>
<tr>
<td>Hydro-Quebec</td>
<td>Quebec</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
An Internet of Data
Examples of Open Data (Early 2016)

Number of Datasets by Country

- Singapore: 11,992
- UK: 26,153
- USA: 191,695
- Canada: 244,885
Open Data v.s. Web Data

Singapore
UK
USA
Canada

WDC Web Table 2015
(English Relational Subset)

11,992
26,153
191,695
244,885

50,820,165
What is a reasonable objective for searching datasets?
Original Table

Open Data & Web Data

Alice
We would like to maximize the amount of **retained values** in the original table.
Join columns should have **high overlap** in values.
Domain Search

**Domain:** a set of values in a dataset (e.g. a column in a table)

**Domain Search:** given a query domain \( Q \) and threshold \( t^* \), find all domains \( X \) such that \( \text{Containment}(Q, X) > t^* \).

\[
\text{Containment}(Q, X) = \frac{|Q \cap X|}{|Q|}
\]
Approximate Domain Search

We borrow an insight from Approximate Nearest-Neighbour Search.

\[ P(x_1) > P(x_2) \]
MinHash LSH [Broder 97, Indyk 98] is a search index for Jaccard.

$$\text{Jaccard}(Q, X) = \frac{|Q \cap X|}{|Q \cup X|}$$

- Can be tuned for Jaccard threshold
- Constant space requirement for all domain sizes
- Biased against large domains
Asymmetric MinHash LSH is a search index for containment [Shrivastava 15].

\[
Jaccard(Q, X_{padded}) = \frac{\text{Containment}(Q, X)}{|Q| + 1 - \text{Containment}(Q, X)}
\]

\[
\text{Containment}(Q, X) \propto Jaccard(Q, X_{padded})
\]

MinHash LSH is used to index the padded domains.

Existing Solution
Asymmetric MinHash LSH is a search index for containment [Shrivastava 15].

- Difficult to tune for containment threshold
- Padding reduces MinHash accuracy, especially when domain sizes have a skewed distribution

Domains

Padding values

Largest domain with size $u$
Research Gap: we need an index for containment search that maintains high accuracy on skewed domain size distribution and is tunable for containment threshold.
LSH Ensemble

(Our Contribution)
Can we use MinHash LSH for containment search without padding?
\[ \text{Jaccard}(Q, X) = \frac{\text{Containment}(Q, X)}{\frac{|X|}{|Q|} + 1 - \text{Containment}(Q, X)} \]
Query Domain: $Q$
Containment Threshold: $t^*$

$\exists s^* \Rightarrow \text{Domains } X \text{ s.t. } \text{Containment}(Q,X) \geq t^*$

$$s^* = \frac{t^*}{\frac{|X|}{|Q|} + 1 - t^*}$$

$s^*$: Jaccard threshold

MinHash LSH

Repository of Domains
Query Domain: \( Q \)
Containment Threshold: \( t^* \)

\[
s^* = \frac{X}{|Q|} + 1 - t^*
\]

Domain size \( |X| \) is not a constant!

Domains \( X \) s.t.
\( \text{Containment}(Q,X) \geq t^* \)

Repository of Domains

MinHash LSH

\( s^* \): Jaccard threshold
\[ \frac{t^*}{u / |Q| + 1 - t^*} \leq \frac{t^*}{|X| / |Q| + 1 - t^*} \]

**u**: the upper bound of domain sizes in range \([l, u]\)
The new threshold introduces

- false positive domains
- no false negative domains

\[ S_u^* = \frac{t^*}{\frac{u}{|Q|} + 1 - t^*} \]
Query Domain: \( Q \)
Containment Threshold: \( t^* \)

\[
S_u^* = \frac{t^*}{u + 1 - t^*}
\]

\( s_u^* \): New Jaccard threshold

Domains \( X \) s.t.
\( \text{Containment}(Q,X) \geq t^* \)

Remove false positive domains

MinHash LSH

Repository of Domains
Query Domain: $Q$
Containment Threshold: $t^*$

$S_u^* = \frac{t^*}{\frac{u}{|Q|} + 1 - t^*}$

$s_u^*$: New Jaccard threshold

Domains $X$ s.t.
$\text{Containment}(Q, X) \geq t^*$

Remove false positive domains

Additional cost

MinHash LSH

Repository of Domains
The query cost can be reduced if we produce less false positive domains.

\[ T_{\text{containment}} = T_{\text{Jaccard}} + \Theta(\text{correct domains}) + \Theta(N_{l,u}^{FP}) \]

Number of false positives domains in the range \([l, u]\)
Number of domains in $[l, u]$ 

$$N_{l,u}^{FP} \leq N_{l,u} \cdot \frac{u - l + 1}{2u}$$

Tight upper bound for number of false positives in range $[l, u]$
We can reduce the range size to reduce the number of false positives domains.

Number of domains in \([l, u]\)

\[
N_{l,u}^{FP} \leq N_{l,u} \cdot \frac{u - l + 1}{2u}
\]

Tight upper bound for number of false positives in range \([l, u]\)
Sorted by domain size

[l, u)
Contiguous domain partitions

$[l_1, u_1)$  $[l_2, u_2)$  $[l_3, u_3)$  ...  $[l_n, u_n)$
Execute query in parallel

Query cost is determined by the partition with the most false positive domains
This led us to formulate an optimization problem for partitioning using the upper bound of $N_{l,u}^F$ on each partition.

$$\Pi^* = \arg\min_{\Pi} \max_{1 \leq i \leq n} M_i$$

$$M_i = N_{l_i,u_i} \cdot \frac{u - l + 1}{2u}$$

This is equivalent to finding a partitioning such that all partitions have the same $M_i$.

$$\exists \Pi^* \text{ s.t. } M_i = M_j, \forall i, j$$
So far, we have shown:

1. Partitioning improves query cost by reducing false positives, while maintaining accuracy

2. An optimal partitioning can be verified using a closed form equation

An optimal partitioning for a real-world domain size distribution?
We proved \textit{equi-depth} partitioning is optimal for domains following \textit{power-law} distribution.
One last thing: **tune** MinHash LSH for containment threshold
MinHash LSH [Indyk 98]

- **b**: number of bands
- **r**: number of concatenated hash values in each band

- **b** and **r** can be self-tuned during query time with structure proposed in LSH Forest [Bawa 05]
We derived the probability of returning with respect to containment given parameters $b$ and $r$.

Our goal is to minimize the sum of false positive and negative probabilities.

Containment threshold
Experimental Result

Compared against Asym MinHash [Shrivastava 2015] and MinHash LSH (using our Containment-to-Jaccard conversion) on accuracy, using Canadian Open Data (65,533 domains):

- LSH Ensemble consistently out-performs other techniques
- More partitions leads to higher accuracy before pruning false positives

Performance experiment used the complete 2015 WDC English Relational Web Table corpora (263 million domains):

- Mean query time around 3 seconds at 32 partitions
- More partitions leads to lower query cost
Creating more partition leads to fewer false positives, while maintaining recall.

Asymmetric MinHash LSH [Shrivastava 15] has high precision, but low recall due to padding.
Accuracy vs. Skewness

(Before Pruning)

- Skewness in domain sizes have negative impact on accuracy for all indices
- LSH Ensemble handles skewness better than others
### Query Performance

<table>
<thead>
<tr>
<th></th>
<th>Mean Query (sec)</th>
<th>Precision Before Pruning ($t^*=0.5$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MinHash LSH</td>
<td>45.13</td>
<td>0.27</td>
</tr>
<tr>
<td>LSH Ensemble (8)</td>
<td>7.55</td>
<td>0.48</td>
</tr>
<tr>
<td>LSH Ensemble (16)</td>
<td>4.26</td>
<td>0.53</td>
</tr>
<tr>
<td>LSH Ensemble (32)</td>
<td>3.12</td>
<td>0.58</td>
</tr>
</tbody>
</table>

On 263 million domains (WDC Web Table)

**Speed up is due to:**
- Fewer false positive domains to process (higher precision)
- Parallelization
Recap

LSH Ensemble

• Uses multiple MinHash LSH built on domain size partitions to approximate containment search and maintain accuracy

• Optimal partitioning (equi-depth) for power-law distributions

• Self-tunable at query time given any threshold

Thank you!

Erkang (Eric) Zhu
ekzhu@cs.toronto.edu