Similarity Join Size Estimation using Locality Sensitive Hashing

Hongrae Lee, Google Inc
Raymond Ng, University of British Columbia
Kyuseok Shim, Seoul National University
Highly Similar, but not Identical, Data

NASA’s Last **Space Shuttle** Crew Takes **Manhattan** This Week
Space.com - Clara Moskowitz - Aug 15, 2011
NEW YORK — Move over Muppets, the astronauts are coming to town. NASA’s final **space shuttle** crew will visit the Big Apple this week for a series of public events to share their experiences of flying on the ...

---

NASA’s last **space shuttle** crew heads to **Manhattan**
Digitaltrends.com - Aug 16, 2011
The four-person crew of NASA’s **space shuttle** Atlantis will travel to New York City (and Sesame Street) this week. Mere weeks after NASA closed up shop on the **space shuttle** program, the crew of the final **space shuttle** mission are headed to the Big Apple ...
Introduction

- Finding all pairs of similar objects is an important operation in many applications
  - Near duplicate detection
    - Identifying spams/plagiarism [HZ'03]
  - Web search
    - Search quality, result diversification, storage [FMN'03, CGM'03, H'06]
  - Data integration/record linkage [BMCW+'03]
  - Community mining [SSB'05], collaborative filtering [BMS'07]
Similarity Join

- Similarity Join is proposed as a general framework for such operations
- Input
  - a collection of objects (vectors) \( V \)
  - similarity measure \( sim \)
  - similarity threshold \( \tau \)
- Output
  - all pairs \((u, v), u, v \in V\), such that \( sim(u, v) \geq \tau \)

Example:

- NASA's last space shuttle crew heads to Manhattan
  - Digitaltrends.com - 4 hours ago
  - The four-person crew of NASA's space shuttle Atlantis will travel to New York City (and Sesame Street) this week. Mere weeks after NASA closed up shop on the space shuttle program, the crew of the final space shuttle mission are headed to the Big Apple ...

- NASA's Last Space Shuttle Crew Takes Manhattan This Week
  - Space.com - Clara Moskowitz - Aug 15, 2011
  - NEW YORK — Move over Muppets, the astronauts are coming to town. NASA's final space shuttle crew will visit the Big Apple this week for a series of public events to share their experiences of flying on the ...
Estimation of Similarity Join Size

- Similarity Join in RDBMs
  - Approximate text processing is being integrated into commercial database systems
  - Similarity Join as a primitive operator [CGK'06]
  - Data cleaning as a repetitive operation [FFM'05]

- Efficient and accurate estimation of Similarity Join size is crucial in query optimization
  - Poor size estimation can result in sub-optimal plans

![Diagram](chart.png)
Problem Statement

Input
- a collection of vectors $V$
- threshold $\tau$ on a similarity measure $\text{sim}$

Output
- number of pairs $(u, v)$ such that $\text{sim}(u,v) \geq \tau$, $u,v \in V$, $u \neq v$
- focus on cosine similarity: $\text{cos}(u,v) = u \cdot v / \| u \| \| v \|$
Challenges

- Join selectivity changes dramatically depending on the threshold: reliable estimates can be hard

<table>
<thead>
<tr>
<th>$\tau$</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>join size</td>
<td>105B</td>
<td>267M</td>
<td>11M</td>
<td>103K</td>
<td>42K</td>
</tr>
<tr>
<td>selectivity</td>
<td>33%</td>
<td>0.085%</td>
<td>0.0086%</td>
<td>0.000064%</td>
<td>0.000013%</td>
</tr>
</tbody>
</table>

- Estimation based on value frequency (as in equi-join) doesn't work in similarity joins

Value Frequency
<table>
<thead>
<tr>
<th>Value</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$10 \times 20 = 200$
Overview

1. LSH-U
   - Estimation based on bucket counts and assumptions on data distribution
   - Very efficient, but sensitive to parameters and data distribution

2. LSH-SS
   - Estimation based on sampling
   - Reliable estimates throughout the threshold range
Outline

● Introduction

● **Locality Sensitive Hashing**
  ● LSH-U: Estimation based on LSH function analysis
  ● LSH-SS: Stratified Sampling based on LSH

● Experiments

● Conclusions
Locality Sensitive Hashing (LSH) [IM '98]

- A hash function, $h$, is *locality sensitive*, if for any vectors $u$ and $v$,
  - $P(h(u) = h(v)) = \text{sim}(u,v)$ [C '02]

- Many similarity search related applications, e.g. KNN search
Indexing Vectors using LSH

● LSH Table
  ○ Concatenates $k$ independent LSH functions: defines a hash table
    ■ $g(v) = (h_1(v),...,h_k(v))$, $P(g(u) = g(v)) = \text{sim}^k(u,v)$
  ○ Group similar objects together into buckets

$h: V \rightarrow \{0,1\}$
$h_1(u) = 1$
$h_2(u) = 0$
$h_3(u) = 0$
$h_4(u) = 1$
$h_5(u) = 0$
g(u) = 10010
Outline

- Introduction
- Locality Sensitive Hashing
- **LSH-U: Estimation based on LSH function analysis**
- LSH-SS: Stratified Sampling based on LSH
- Experiments
- Conclusions
Basic Definition

- Assume an LSH table and a threshold $\tau$
- $N$: # pairs
- $B(u)$: $u$'s bucket
- Consider a random pair $(u,v)$ and define events as follows:
  - $H$: $B(u) = B(v)$, High (expected) similarity
  - $L$: $B(u) \neq B(v)$, Low (expected) similarity
  - $T$: $\text{sim}(u,v) \geq \tau$, True pair
  - $F$: $\text{sim}(u,v) < \tau$, False pair
- e.g.
  - $N_H$: # pairs in the same bucket
  - $N_T$: # true pairs
  - $P(T|H)$: the probability that a random pair from a bucket is a true pair
LSH-U (1/2)

- Observation: a pair of vectors from a bucket is either a true pair or a false pair
  - \( N_H = N_T \cdot P(H|T) + N_F \cdot P(H|F) \)
  - \( N_H \): from bucket counts (# records at each bucket), \( N_T (= J) \): join size, \( P(H|T), P(H|F) \): from data, \( N_F \): # tot pairs - \( N_T \)

- LSH-U: an estimator based on the above equation
  - Assumes actual data distribution (\( P(H|T), P(H|F) \)) follows LSH
  - e.g. \( k = 1 \) (See the paper for the general form of the estimator),
    - \( J = N_T = (2 - \tau)N_H - \tau N_L \), \( N_H, N_L \) can be computed from bucket counts

Data distribution assumed by LSH-U when \( k = 1 \)
An estimation with only bucket counts and an assumption on the data distribution
  ○ No sampling
  ○ Analogous to traditional equi-join size estimation using histograms with uniformity assumptions
  ○ Sensitive to LSH parameters and data distribution
Outline

● Introduction
● Locality Sensitive Hashing
● LSH-U: Estimation based on LSH function analysis
● **LSH-SS: Stratified Sampling based on LSH**
● Experiments
● Conclusions
Stratified Sampling Using LSH

- Our observation: an LSH table implicitly partitions data into two strata
  1. Pairs in the same bucket
  2. Pairs that are not in the same bucket
    ○ Pairs in the same bucket are likely to be more similar
- Key intuition to overcome the difficulty of sampling at high thresholds
  ○ Even at high thresholds, it is relatively easy to sample a true pair from pairs in the same bucket

| $\tau$ | $P(T)$  | $P(T|H)$ |
|-------|---------|---------|
| 0.1   | 0.082   | 0.31    |
| 0.3   | 0.0024  | 0.054   |
| 0.5   | 0.000034| 0.049   |
| 0.7   | 0.0000039| 0.045  |
| 0.9   | 0.00000091| 0.040  |

DBLP
$T$: $\text{sim}(u,v) \geq \tau$
$H$: $u,v$ in the same bucket
LSH-SS: Stratified Sampling

- Define two strata of pairs of vectors
  - \( S_H : \{(u,v) : u,v \in V, B(u) = B(v)\} \)
  - \( S_L : \{(u,v) : u,v \in V, B(u) \neq B(v)\} \)
- \( J = J_H + J_L \)
  - \( J_H = |\{(u,v) \in S_H : \text{sim}(u,v) \geq \tau\}| \)
  - \( J_L = |\{(u,v) \in S_L : \text{sim}(u,v) \geq \tau\}| \)
- Our estimator
  - \( J_{\text{SS}} = J_H + J_L \)
Sampling from $S_H$ and $S_L$

- **Sampling from $S_H$**
  - Each bucket has a weight proportional to the number of pairs in it.
  - Perform a weighted sampling of buckets, and then select a pair in the bucket uniformly at random.
  - Test if the pair satisfies $\tau$, and repeat it $m_H$ times.
  - $J_H = n_H \times |S_H| / m_H$
  - Number of true pairs among $m_H$ samples: $n_H$

- **Sampling from $S_L$**
  - Select a pair $(u,v)$ uniformly at random.
  - Discard the pair if $B(u) = B(v)$.
  - Test if the pair satisfies $\tau$, and repeat it $m_L$ times.
  - $J_L = n_L \times |S_L| / m_L$: not reliable at high thresholds!
Challenges in Sampling from $S_L$

- Sampling probability at $S_L$, $P(T|L)$, can be very small
- At high thresholds
  - Reliable sampling is hard since $P(T|L)$ is very small
  - A majority of true pairs are in $S_H$
- At low thresholds
  - $P(T|L)$ becomes larger
  - Most of true pairs are in $S_L$

| $t$  | $P(T|L)$ | $P(L|T)$ |
|------|----------|----------|
| 0.1  | 0.08     | ~1       |
| 0.3  | 0.0002   | ~1       |
| 0.5  | 0.00003  | 0.997    |
| 0.7  | 0.00000028 | 0.79   |
| 0.9  | 0.000000013 | 0.14    |
Our Solution: Using Adaptive Sampling at $S_L$

- Adaptive Sampling [LNS'90]: based on true samples observed, it gives either
  1) An estimate with error guarantees or
  2) An upper bound on the estimate
- Sampling from $S_L$
  - In case 1), output the estimate from $S_L$
  - In case 2), discard the estimate from $S_L$ ($J_{SS} = J_H$) or scale it down ($J_{SS} = J_H + \alpha J_L$, $\alpha < 1$)
- Why is it acceptable to scale down $J_L$ in case 2)?
  - When an estimate from $S_L$ is not reliable, its contribution to $J_{SS}$ is generally small
Analysis

- We show that the proposed algorithms give reliable estimates both at high and low threshold ranges
  - Proposed sample size: each $n$ pairs at $S_H$ and $S_L$
  - Assumes $P(T|H) > \log n/n$, which is easily satisfied by known LSH schemes
Related Work

Similarity join processing
- MergeOpt [SK'04]
- PartEnum [AGK'06]
- All-pairs [BMS'07]

Join size estimation
- Adaptive sampling [LNS'90]
- Cross/index/tuple sampling [HNSS'93]
- Bi-focal sampling [GGMS'96]
- Tug-of-war [AGMS'99]

Set similarity join size estimation
- Lattice Counting [LNS'09]
Outline

- Introduction
- Locality Sensitive Hashing
- LSH-U: Estimation based on LSH function analysis
- LSH-SS: Stratified Sampling based on LSH
- **Experiments**
- Conclusions
Experimental Evaluation

● Data set
  ○ DBLP: 800K
  ○ NYT: NY Times articles, 150K
  ○ PUBMED: PubMed abstracts, 400K

● Algorithms
  ○ LSH-SS: discard $J_L$ when it's not reliable
  ○ LSH-SS(D): uses a dampened scaling-up factor
  ○ RS(pop): sample pairs from the whole cross product
  ○ RS(cross): cross sampling, sample records and consider all pairs in the sample
Relative Error in DBLP

- RS show huge overestimations at high thresholds

Overestimation

- RS show extreme underestimations at high thresholds
- That is, RS's estimation fluctuate a lot, especially at high thresholds

Underestimation
Variance in DBLP

- Variance of LSH-SS methods is generally much smaller than that of RS throughout the threshold range
Sensitivity Analysis on LSH Parameters

- LSH-S: estimation based on the LSH function analysis
- LSH-SS is generally not sensitive to LSH parameter choices

Impact of $k$ (# LSH functions) on DBLP
Conclusion

● Proposed stratified sampling algorithms using an LSH index
● Provide reliable estimates throughout the similarity threshold range
● Can be easily applied to existing LSH indices
Thank you!