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 τ
- Output
  - all pairs (u,v), $u,v \in V$, such that $sim(u,v) \geq \tau$

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 **...**

[0.6, 0, 0, 0.5, 0.12, 0, 0, ...]

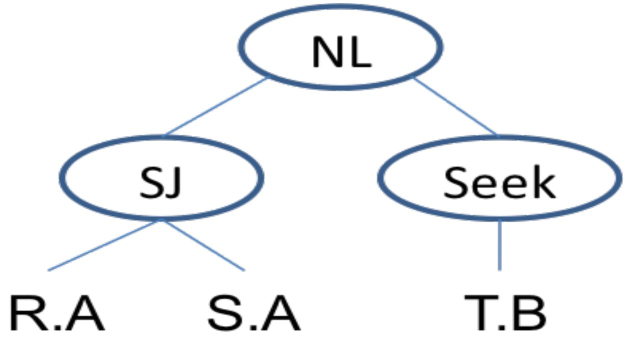
NASA's Last **Space Shuttle** Crew Takes Manhattan This Week
Space.com - Clara Moskowitz - Aug 15, 2011
NEW YORK — Move over Muppets, the astronauts are coming to town. NASA's final **space shuttle** crew will visit the Big Apple this week for a series of public events to share their experiences of flying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

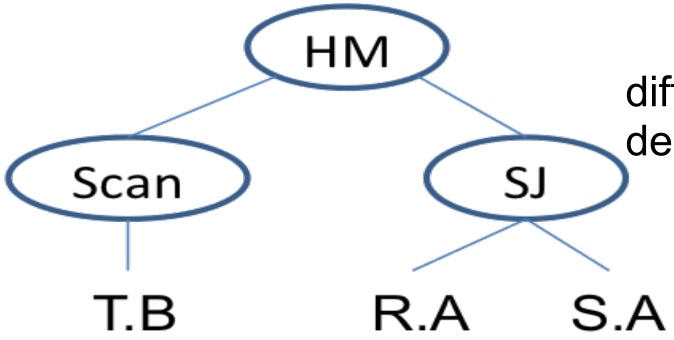
[0.2, 0.1, 0, 0.4, 0.3, 0.2, 0, ...]

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





different opt plans
depending on SJ size

Problem Statement

Input

- a collection of vectors V
- threshold τ on a similarity measure *sim*

Output

- number of pairs (u, v) such that $sim(u,v) \geq \tau$, $u,v \in V$, $u \neq v$

- focus on cosine similarity: $cos(u,v) = u \cdot v \,/\, \| u \| \; \| v \|$

Challenges

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

τ	0.1	0.3	0.5	0.7	0.9
join size	105B	267M	11M	103K	42K
selectivity	33%	.085%	.0086%	.000064%	.000013%

DBLP 800K

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

Equi-join

R

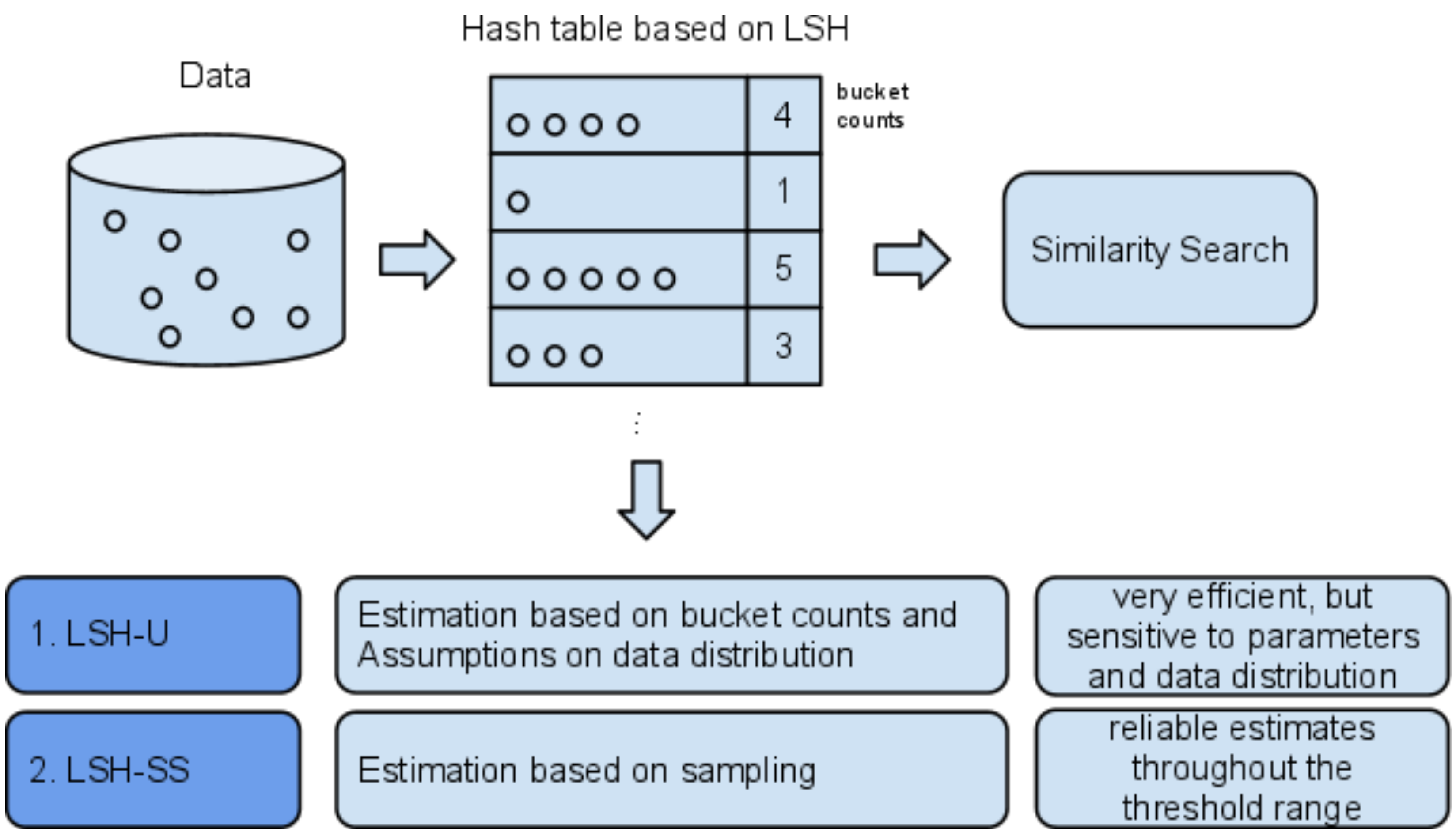
Value	Frequency
1	5
2	10
...	...

S

Value	Frequency
2	20
3	20
...	...

10 X 20=200

Overview

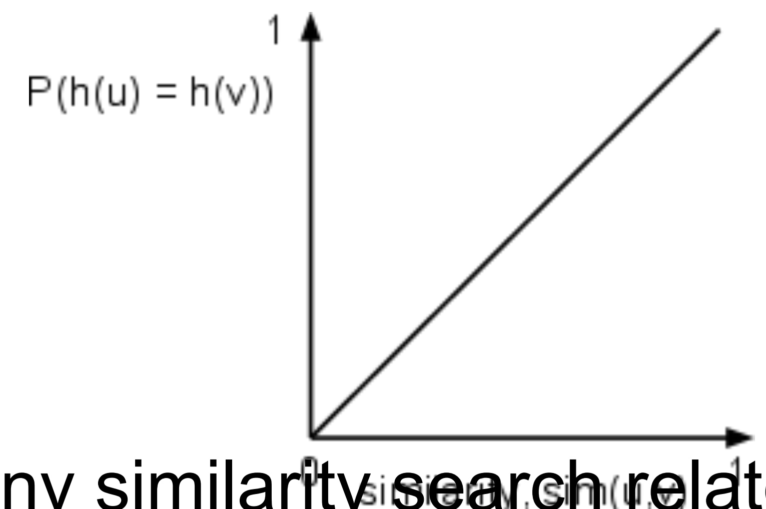

Hash table based on LSH
Data
bucket counts
4
1
5
3
Similarity Search
1. LSH-U
Estimation based on bucket counts and Assumptions on data distribution
very efficient, but sensitive to parameters and data distribution
2. LSH-SS
Estimation based on sampling
reliable estimates throughout the threshold range

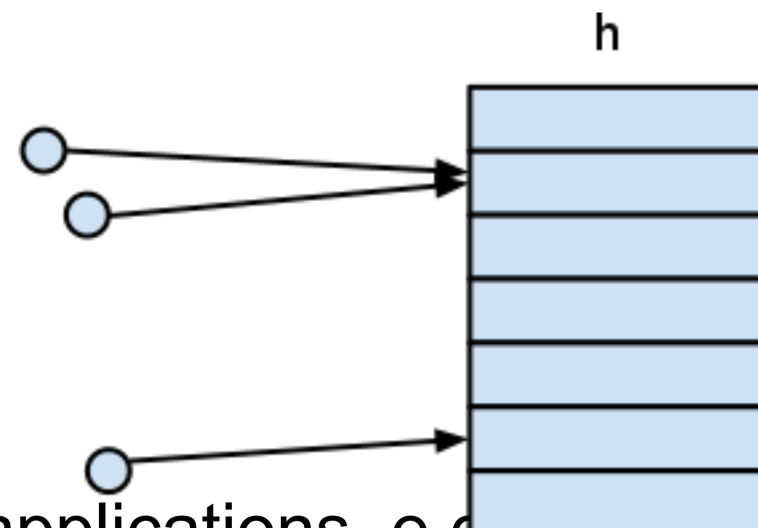
Outline

- 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*)) = sim(*u*,*v*) [C '02]

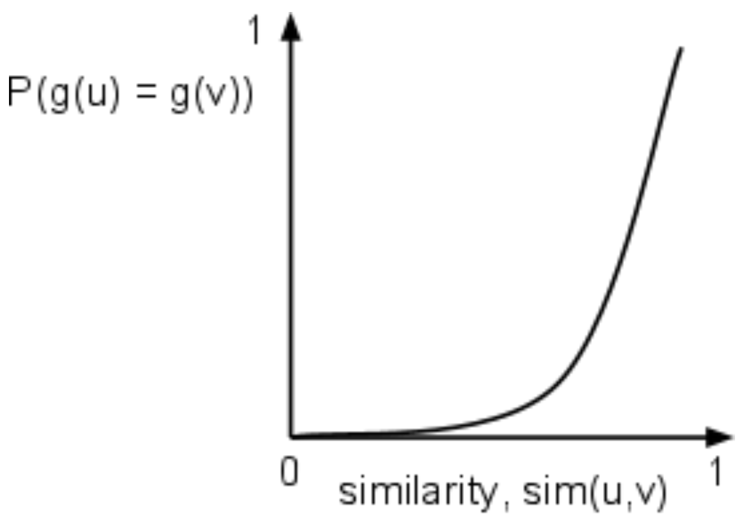


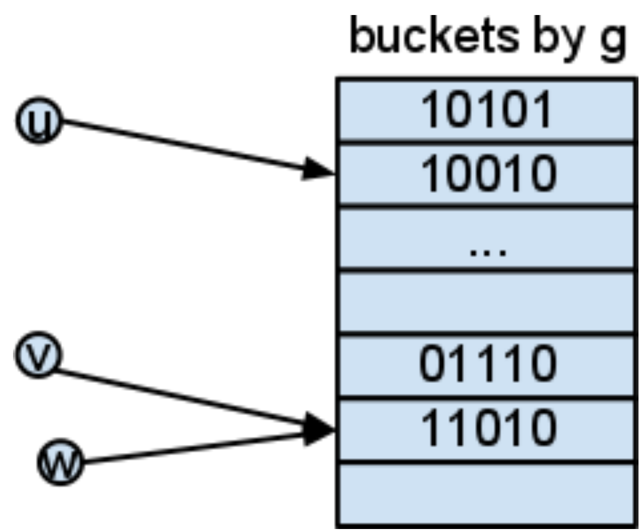


- Many similarity search related applications, e.g. kNN search

Indexing Vectors using LSH

- LSH Table
  - Concatenates *k* independent LSH functions: defines a hash table
    - $g(v) = (h_1(\mathrm{v}),...,h_k(\mathrm{v}))$, $\mathrm{P}(g(u) = g(v)) = sim^k(u,v)$





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

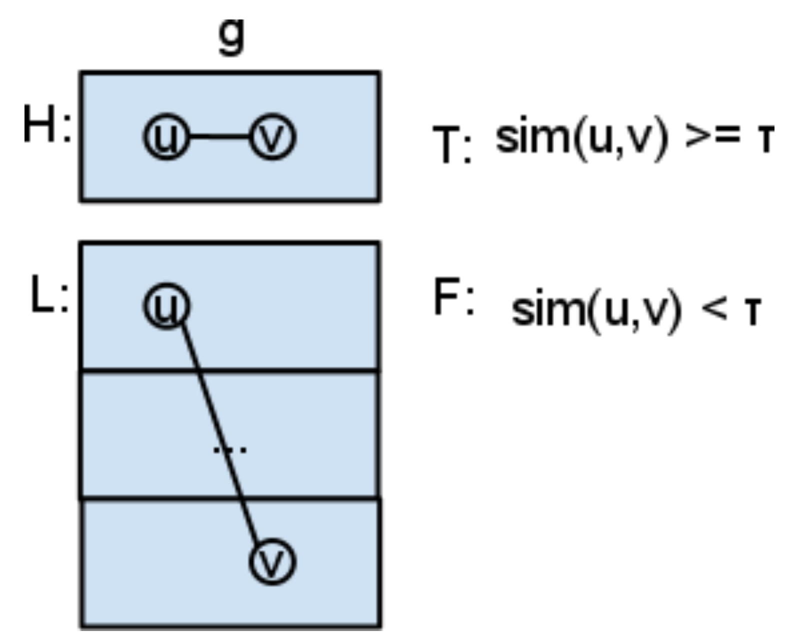
  - Group similar objects together into buckets

Outline

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

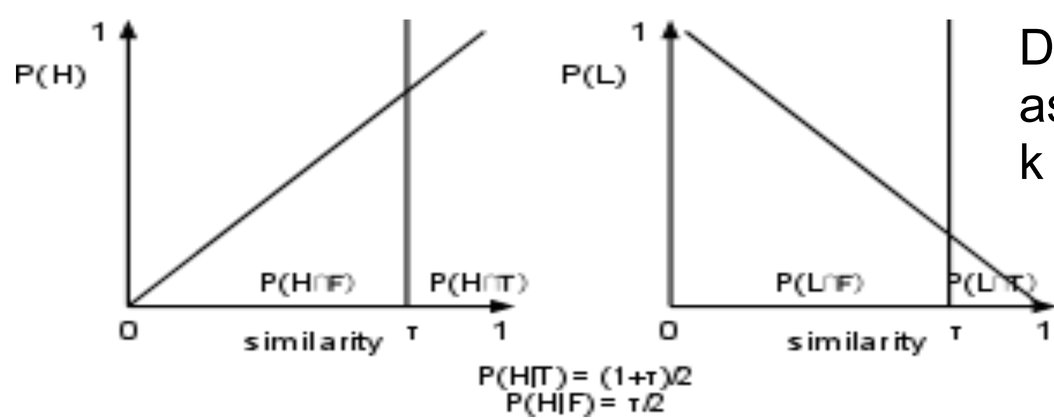
Basic Definition

- Assume an LSH table and a threshold τ
- 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: $sim(u,v) \geq \tau$, True pair
  - F: $sim(u,v) < \tau$, 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
  - $N_H = N_{T*}P(H|T) + N_{F*}P(H|F)$
  - 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

LSH-U (2/2)

- 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

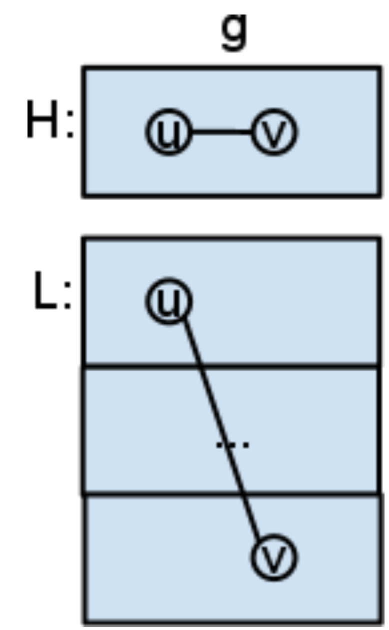
- Our observation: an LSH table implicitly partitions data into two strata
    1. Pairs in the same bucket
    2. Pairs that are not in the same bucket
    - Pairs in the same bucket are likely to be more similar
- Key intuition to overcome the difficulty of sampling at high thresholds
    - Even at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

τ	P(T)	P(T|H)
0.1	.082	.31
0.3	.00024	.054
0.5	.0000034	.049
0.7	.00000039	.045
0.9	.000000091	.040

DBLP
T: sim(u,v) >= τ
H: u,v in the same bucket

LSH-SS: Stratified Sampling

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



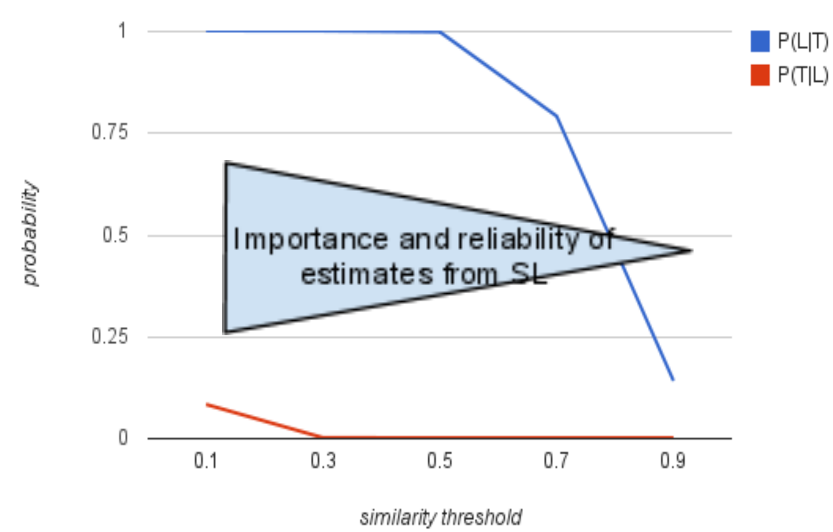
Sampling from S_H and S_L

- Sampling from S_H
  - Each bucket has a weight proportional to # 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 τ, and repeat it m_H times
  - $J_H = n_H * |S_H| / m_H$
    - # 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 τ, and repeat it m_L times
  - $J_L = n_L * |S_L| / m_L$: 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	.08	~1
0.3	.0002	~1
0.5	.00003	.997
0.7	.00000028	.79
0.9	.000000013	.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

See the paper for details

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]

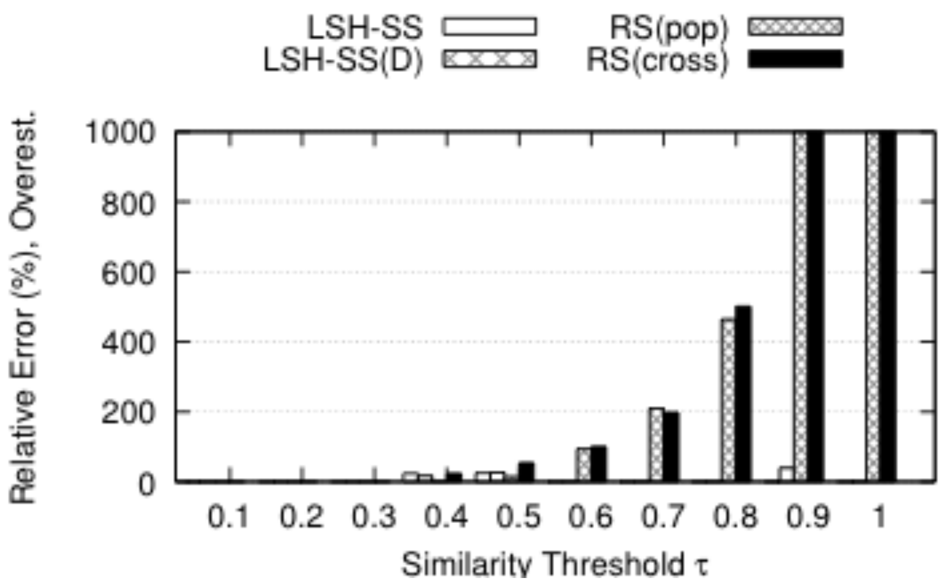
Outline

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

Experimental Evaluation

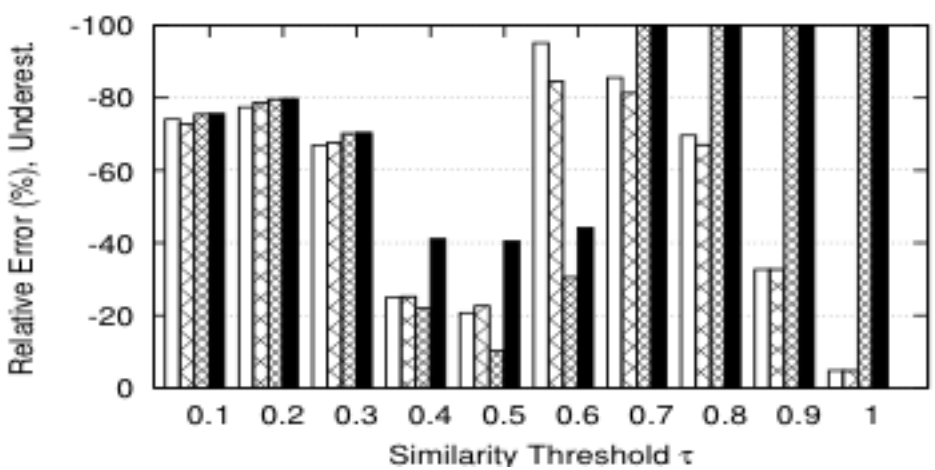
- Data set
  - DBLP: 800K
  - NYT: NY Times articles, 150K
  - PUBMED: PubMed abstracts, 400K
- Algorithms
  - LSH-SS: discard J_L when it's not reliable
  - LSH-SS(D): uses a dampened scaling-up factor
  - RS(pop): sample pairs from the whole cross product
  - RS(cross): cross sampling, sample records and consider all pairs in the sample

Relative Error in DBLP



Overestimation

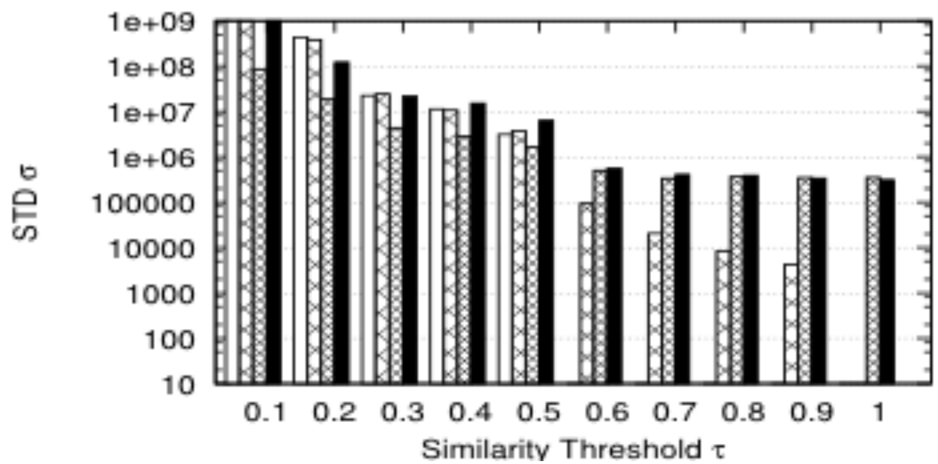
- RS show huge overestimations at high thresholds



Underestimation

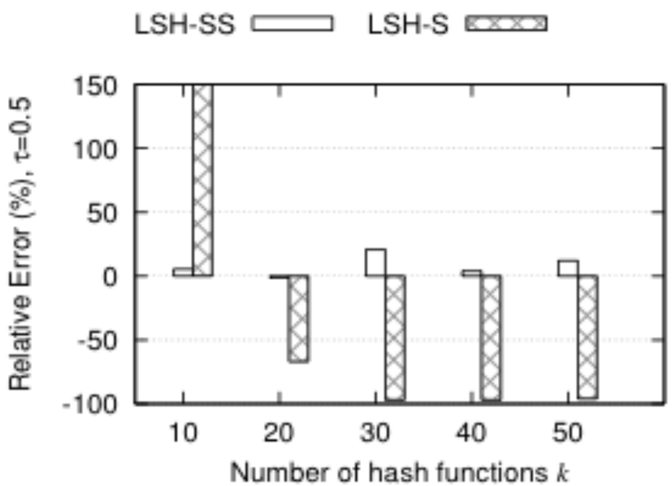
- RS show extreme underestimations at high thresholds
- That is, RS's estimation fluctuate a lot, especially at high thresholds

Variance in DBLP

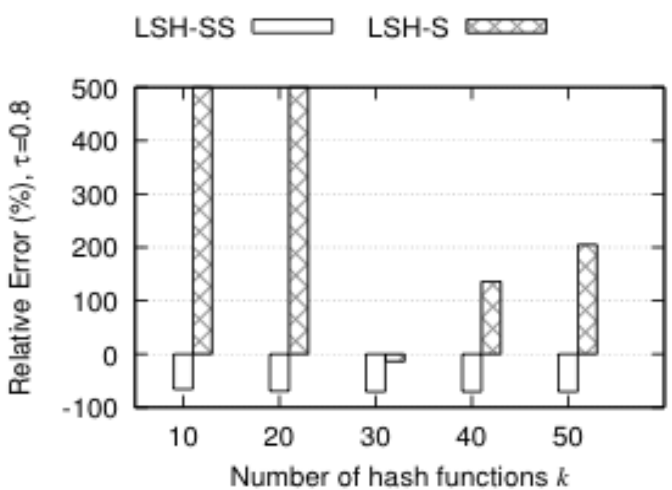


- Variance of LSH-SS methods is generally much smaller than that of RS throughout the threshold range

Sensitivity Analysis on LSH Parameters



(a) Accuracy, $\tau = 0.5$



(b) Accuracy, $\tau = 0.8$

Impact of k (# LSH functions) on DBLP

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

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!