Similarity Query Processing for Probabilistic Sets

Ming Gao [1], Cheqing Jin [1], Wei Wang [2], Xuemin Lin [1,2], Aoying Zhou [1]*
[1]*Shanghai Key Laboratory on trustworthy computing, Software Engineering Institute, East China Normal University, Shanghai, China*
[2]*The University of New South Wales, Sydney, Australia*
omega.mgao@gmail.com, {cqjin, ayzhoug}@sei.ecnu.edu.cn
{weiw, lxue}@cse.unsw.edu.au

Motivation

- Evaluate similarity between uncertain sets
- Existing work
  - Huge model size
  - Significant similarity evaluation cost
- This paper
  - Comprehensive study for probabilistic set may have thousands of elements

Solution

- Similarities based on dynamic programming
  - Expected Similarity (ES)
  - Confidence-based Similarity (CS)
- Exact query processing based on pruning
  - Individual pruning
  - Batch pruning
- Approximate query processing based on sampling

Agenda

- Introduction
- Related work
- Problem definition and data normalization
- Exact similarity computation
- Pruning techniques
- Approximate solution
- Experiments

Introduction

- Applications
  - Personalization systems
  - Multi-label classification
- Contribution
  - Handle large p-sets efficiently
  - Similarity measure based on dynamic programming
  - Pruning techniques and approximate methods
  - Experiments upon synthetic and real datasets

Related work

- Uncertain Data Management
  - Information extraction and integration, multimedia retrieval, optical character recognition
  - MayBMS, MystiQ, Trio
- Similarity Search
  - Top-k, k-NN, reverse k-NN, range queries
- Similarity Join
  - Batch similarity queries

Related work

- Efficient processing of probabilistic set-containment queries on uncertain set-valued data.[*Inf. Sci*]
  - Same
    - Probabilistic set model, one of the similarity measure
  - Different
    - Pruning methods, approximate methods
- Probabilistic string similarity joins.[*SIGMOD 2010*]
  - Different
    - Non-neglectible correlations
    - Involving aggregated probabilities

Related work

- Set similarity join on probabilistic data.[*VLDB 2010*]

Model	Expressive Power	Exact Similarity Computation	Upper Bound Computation
Set-level [27]	Most general	$O(N^2)$	$O(N)$
Element-level [27]	Can model exclusion	$\Omega(2^n)$	$O(n^2)$ (online) or $O(n)$ (offline)
Our p-set model	A special case of Element-level model	$O(n^3)$	$O(n)$

  - Models and Similarity Evaluation
    - Set-level
    - Element-level
  - Pruning Rules
    - Jaccard Distance pruning
    - Probability upper bound pruning

Problem definition and data normalization

- Probabilistic set model

$$\mathcal{A} = \{a_i : p_{a_i} | a_i \in \mathcal{D}, \forall i \in [1, n]\}$$

- Possible world semantics

$$w(\mathcal{A}) \qquad \mathbf{Pr}[w] = \prod_{t \in w} p_t \prod_{t \notin w} (1 - p_t)$$

$$\mathcal{W}(\mathcal{A}, \mathcal{B}) = \mathcal{W}(\mathcal{A}) \times \mathcal{W}(\mathcal{B}) \qquad (w_a, w_b) \in \mathcal{W}(\mathcal{A}, \mathcal{B}) \text{ is } \mathbf{Pr}[w_a] \cdot \mathbf{Pr}[w_b]$$

- Jaccard coefficient

$$jac(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

Problem definition and data normalization

- Example
  - P-sets

$\mathcal{A}$	$\mathcal{B}$
$\{1:0.7,\ 2:1.0\}$	$\{1:1.0,\ 2:0.5,\ 3:0.8\}$

  - All the joint possible worlds

w_a	w_b	$\mathbf{Pr}[(w_a, w_b)]$	**Jaccard**
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}\}$	0.03	0
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}\}$	0.03	0.5
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.12	0
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.12	0.333
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}\}$	0.07	0.5
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}\}$	0.07	1
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.28	0.333
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.28	0.666

Problem definition and data normalization

- Expected Similarity (ES)

$$ES(\mathcal{A},\mathcal{B}) = \sum_{(w_a,w_b)\in\mathcal{W}(\mathcal{A},\mathcal{B})} sim(w_a,w_b)\cdot \mathbf{Pr}\left[(w_a,w_b)\right]$$
$$= \sum_{w_a\in\mathcal{W}(\mathcal{A})\wedge w_b\in\mathcal{W}(\mathcal{B})} sim(w_a,w_b)\cdot \mathbf{Pr}\left[w_a\right]\cdot \mathbf{Pr}\left[w_b\right]$$

- Confidence-based Similarity (CS)

$$CS(\mathcal{A},\mathcal{B},minconf) = \max\{\, x \mid \mathbf{CPr}(x,\mathcal{A},\mathcal{B}) \geq minconf \,\}$$

  – conditioned cumulative probability CPr(x, A, B)

$$\mathbf{CPr}(x,\mathcal{A},\mathcal{B}) = \sum_{(w_a,w_b)\in\mathcal{W}(\mathcal{A},\mathcal{B})\wedge sim(w_a,w_b)\geq x} \mathbf{Pr}\left[(w_a,w_b)\right]$$

Problem definition and data normalization

- Example

$\mathcal{A}$	$\mathcal{B}$
$\{1:0.7,\ 2:1.0\}$	$\{1:1.0,\ 2:0.5,\ 3:0.8\}$

w_a	w_b	$\mathbf{Pr}\left[(w_a, w_b)\right]$	**Jaccard**
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}\}$	0.03	0
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}\}$	0.03	0.5
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.12	0
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.12	0.333
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}\}$	0.07	0.5
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}\}$	0.07	1
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.28	0.333
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.28	0.666

	Jaccard
$ES(\mathcal{A}, \mathcal{B})$	0.44
$CS(\mathcal{A}, \mathcal{B}, minconf = 0.3)$	0.666
$CS(\mathcal{A}, \mathcal{B}, minconf = 0.5)$	0.333

Problem definition and data normalization

- Normalization of two p-sets

$$\mathcal{A} = \{\, c_1 : p^{\mathcal{A}}_{c_1}, \ldots, c_k : p^{\mathcal{A}}_{c_k},\ d_1 : p_{d_1}, \ldots, d_{n-k} : p_{d_{n-k}} \,\}$$
$$\mathcal{B} = \{\, c_1 : p^{\mathcal{B}}_{c_1}, \ldots, c_k : p^{\mathcal{B}}_{c_k},\ d_{n-k+1} : p_{d_{n-k+1}}, \ldots$$
$$d_{n+m-2k} : p_{d_{n+m-2k}} \,\}$$

$\mathcal{A}$	$\mathcal{B}$
$\{1:0.7,\ 2:1.0\}$	$\{1:1.0,\ 2:0.5,\ 3:0.8\}$

- Size and expected size

w_a	w_b	$\mathbf{Pr}\,[(w_a, w_b)]$	**Jaccard**
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}\}$	0.03	0
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}\}$	0.03	0.5
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.12	0
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.12	0.333
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}\}$	0.07	0.5
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}\}$	0.07	1
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.28	0.333
$\{2^{\mathcal{A}}, 1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}, 2^{\mathcal{B}}, 3^{\mathcal{B}}\}$	0.28	0.666

Exact similarity computation

- Equivalent classes

$$H[i,j] = \sum_{(w_a,w_b)\in\mathcal{W}(\mathcal{A},\mathcal{B})\wedge|w_a\cap w_b|=i\wedge|w_a\cup w_b|=j} \mathbf{Pr}[w_a]\cdot\mathbf{Pr}[w_b]$$

- Example

w_a	w_b	$\mathbf{Pr}[(w_a,w_b)]$	i	j	**Jaccard**
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}\}$	0.03	0	2	0
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}},3^{\mathcal{B}}\}$	0.12	0	3	0
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}},2^{\mathcal{B}},3^{\mathcal{B}}\}$	0.12	1	3	0.333
$\{2^{\mathcal{A}},1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}},3^{\mathcal{B}}\}$	0.28	1	3	0.333
$\{2^{\mathcal{A}}\}$	$\{1^{\mathcal{B}},2^{\mathcal{B}}\}$	0.03	1	2	0.5
$\{2^{\mathcal{A}},1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}}\}$	0.07	1	2	0.5
$\{2^{\mathcal{A}},1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}},2^{\mathcal{B}},3^{\mathcal{B}}\}$	0.28	2	3	0.666
$\{2^{\mathcal{A}},1^{\mathcal{A}}\}$	$\{1^{\mathcal{B}},2^{\mathcal{B}}\}$	0.07	2	2	1

$H[i,j]$

	$j=0$	$j=1$	$j=2$	$j=3$
$i=0$	0	0	0.03	0.12
$i=1$		0	0.1	0.4
$i=2$			0.07	0.28

Exact similarity computation

- Calculate ES

$$ES = \sum_{i=1}^{k} \sum_{j=i}^{m+n-k} H[i,j] \cdot (i/j)$$

- Calculate CS

Algorithm 1: Calculate CS from $H[i,j]$

Input: $H[i,j]$, $minconf$
Data: $heap$ is a max-heap on the similarity values.
for $i = 1$ **to** k **do** $heap.push(1.0, i, i)$;
$CPr \leftarrow 0$; $sim \leftarrow 0$;
while $heap$.empty = **false do**
  $(sim, i, j) \leftarrow heap$.pop;
  $CPr \leftarrow CPr + H[i,j]$;
  if $CPr \geq minconf$ **then break**;
  if $j < m + n - k$ **then** $heap$.push$(\frac{i}{j+1}, i, j+1)$;
return sim

$H[i,j]$

	$j=0$	$j=1$	$j=2$	$j=3$
$i=0$	0	0	0.03	0.12
$i=1$		0	0.1	0.4
$i=2$			0.07	0.28

Exact similarity computation

- Computing H
  - Common element

$$H^l[i,j] = H^{l-1}[i,j](1-p_l^{\mathcal{A}})(1-p_l^{\mathcal{B}}) + H^{l-1}[i,j-1](p_l^{\mathcal{A}}(1-p_l^{\mathcal{B}}) + (1-p_l^{\mathcal{A}})p_l^{\mathcal{B}}) + H^{l-1}[i-1,j-1]p_l^{\mathcal{A}}p_l^{\mathcal{B}}$$

  - Distinct element

$$H^l[i,j] = H^{l-1}[i,j](1-p_l) + H^{l-1}[i,j-1]p_l$$

  - Time complexity $O(n^3)$
  - Space complexity $O(n^2)$

$H[i,j]$

	$j=0$	$j=1$	$j=2$	$j=3$
$i=0$	0	0	0.03	0.12
$i=1$		0	0.1	0.4
$i=2$			0.07	0.28

Pruning Techniques

Algorithm 2: Answer Queries with Pruning ($\mathcal{Q}$, $\{\mathcal{O}_i\}$, τ, $minconf$)

$C \leftarrow$ candidates that survive the batch pruning (c.f., Sec. V-D);
foreach *p-set in C* **do**
    $pruned \leftarrow$ **false**;
    if *the query type is ESQ* **then**
        $ub \leftarrow$ calcESUpperBound($\mathcal{Q}$, $\mathcal{O}_i$) (c.f., Sec. V-B);
        if $ub < \tau$ **then** $pruned \leftarrow$ **true**
    if *the query type is CSQ* **then**
        $ub \leftarrow$ calcCSUpperBound($\mathcal{Q}$, $\mathcal{O}_i$, τ) (c.f., Sec. V-C);
        if $ub < minconf$ **then** $pruned \leftarrow$ **true**
    if $pruned =$ **false** **then**
        $sim \leftarrow$ the similarity value between $\mathcal{Q}$ and $\mathcal{O}_i$;
        if $sim \geq \tau$ **then**
            output $\mathcal{O}_i$;

$$\mathbf{E}[|\mathcal{A}|], \text{ is } \sum_{w \in \mathcal{W}(\mathcal{A})} |w| \cdot \mathbf{Pr}[w] = \sum_{l=1}^{n} p_l^{\mathcal{A}}$$

$$\mathbf{E}[|\mathcal{A} \cap \mathcal{B}|] \quad \text{is} \sum_{(w_a, w_b) \in \mathcal{W}(\mathcal{A}, \mathcal{B})} |w_a \cap w_b| \cdot \mathbf{Pr}[(w_a, w_b)] = \sum_{l=1}^{k} p_l^{\mathcal{A}} \cdot p_l^{\mathcal{B}}$$

$$\mathbf{E}[|\mathcal{A} \cup \mathcal{B}|] \quad \text{is} \quad \mathbf{E}[|\mathcal{A}|] + \mathbf{E}[|\mathcal{B}|] - \mathbf{E}[|\mathcal{A} \cap \mathcal{B}|] = \sum_{l=1}^{k} (p_l^{\mathcal{A}} + p_l^{\mathcal{B}} - p_l^{\mathcal{A}} \cdot p_l^{\mathcal{B}}) + \sum_{l=k+1}^{n+m-k} p_l$$

Pruning Techniques

- Pruning Rule for ESQ

$$\mathbf{E}\left[X/Y\right] < UB_1(\mathbf{E}\left[X\right], \mathbf{E}\left[Y\right])$$

$$UB_1(u,v) = \min_{\exp(-u/3)\leq\epsilon\leq 1}\left(2\epsilon + \frac{u+\sqrt{-3u\ln\epsilon}}{v-\sqrt{-2v\ln\epsilon}}\right)$$

$$UB_1(\mathbf{E}\left[|\mathcal{Q}\cap\mathcal{O}|\right], \mathbf{E}\left[|\mathcal{Q}\cup\mathcal{O}|\right]) \leq \tau$$

- Pruning Rule for CSQ

$$\mathbf{Pr}\left[X \geq \alpha Y\right] < UB_2(\mathbf{E}\left[X\right], \mathbf{E}\left[Y\right], \alpha)$$

$$UB_2(u,v,\alpha) = \min_{u\leq\xi\leq\min(\alpha v, 2u)}\left(e^{\frac{-(\alpha v-\xi)^2}{2\alpha^2 v}} + e^{\frac{-(\xi-u)^2}{3u}}\right)$$

$$\mathbf{E}\left[|\mathcal{Q}\cap\mathcal{O}|\right] \leq \tau\cdot\mathbf{E}\left[|\mathcal{Q}\cup\mathcal{O}|\right] \qquad UB_2(\mathbf{E}\left[|\mathcal{Q}\cap\mathcal{O}|\right], \mathbf{E}\left[|\mathcal{Q}\cup\mathcal{O}|\right], \tau) \leq minconf$$

Pruning Techniques

- Batch Pruning
  - Discard many p-sets in the database without even evaluating their similarity upper bounds
    - Index all the p-sets in the database by their expected sizes
    - Compute a lower bound S_L and an upper bound S_U of the expected size for the appropriate query type
    - Only consider p-sets in the database whose expected sizes fall within $[S_L, S_U]$

Pruning Techniques

- Batch Pruning
  - How to decide S_L and S_U
  - Batch Pruning for ESQ

$$\mathbf{E}\left[|\mathcal{Q} \cap \mathcal{O}|\right] \leq \min(\mathbf{E}\left[|\mathcal{Q}|\right], \mathbf{E}\left[|\mathcal{O}|\right])$$

$$\mathbf{E}\left[|\mathcal{Q} \cup \mathcal{O}|\right] \geq \max(\mathbf{E}\left[|\mathcal{Q}|\right], \mathbf{E}\left[|\mathcal{O}|\right])$$

$$x + \sqrt{-3x \ln \epsilon^*} = (\tau - 2\epsilon^*)(\mathbf{E}\left[|\mathcal{Q}|\right] - \sqrt{-2\mathbf{E}\left[|\mathcal{Q}|\right] \ln \epsilon^*})$$

$$x - \sqrt{-2x \ln \epsilon^*} = \left(\mathbf{E}\left[|\mathcal{Q}|\right] + \sqrt{-3\mathbf{E}\left[|\mathcal{Q}|\right] \ln \epsilon^*}\right) / (\tau - 2\epsilon^*)$$

  - Batch Pruning for CSQ

$$\exp\left(\frac{-(\xi_1^* - x)^2}{3x}\right) = minconf/2$$

$$\exp\left(\frac{-(\tau \cdot x - \xi_2^*)^2}{2\tau^2 \cdot x}\right) = minconf/2$$

Approximate solution

- Sampling-based method
  - Approximate algorithm for ES

$$\lceil (\ln \tfrac{2}{\delta})/(2\epsilon^2) \rceil \qquad \mathbf{Pr}\left[\left|\widehat{ES} - ES\right| \le \epsilon\right] \ge 1 - \delta$$

  - Approximate algorithm for CS

$$G = 24 \cdot \lceil \ln \tfrac{1}{\delta} \rceil,\ M = \lceil 2\epsilon^{-2} \rceil \qquad \mathbf{Pr}\left[CS^{-} \le \widehat{CS} \le CS^{+}\right] \ge 1 - \delta$$

  - O(n)

Experiments

- Implementation
  - Java
  - Intel Pentium IV 2.8GHz CPU
  - 4GB memory
- Synthetic datasets
  - SYN*a*-U
    - a uniform distribution within the range of [v, 0.9] with a default v value of 0.2.
  - SYN*a*-G
    - a Gaussian distribution N(u, o) capped to the range of (0, 1]. By default, u = 0.8 and o = 0.2.

Experiments

Dataset	DB Size	p-set Min/Max/Avg Size
pDBLP	5,000	27 / 708 / 204.9
pDeli	44,876	50 / 293,214 / 453.2

- Real-world datasets
  - pDBLP
    - a fairly simple yet effective method based on topical terms used in authors' DBLP entries
  - pDeli
    - the social bookmarking dataset which was crawled from the Del.icio.us web site during 2006 and 2007
  - Sigmoid function

$$p(e) = \frac{2}{1+\exp(-c(e))} - 1$$

Experiments

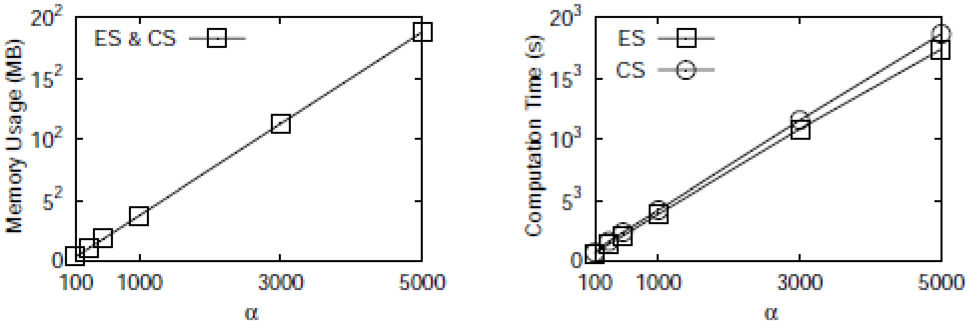
- Default parameters

are: $minconf = 0.5$ (for CS), $\tau = 0.5$, $\alpha = 1000$, $\gamma = 10\%$, $\epsilon = 0.06$, $\delta = 0.06$, $\upsilon = 0.2$, $\sigma = 0.2$, and $\mu = 0.8$.

- Measures
  - Memory Usage
  - Computation Time
  - Query Time, Pruning time
  - Candidate size, result size
  - Pruning rate
  - Average precision

Experiments

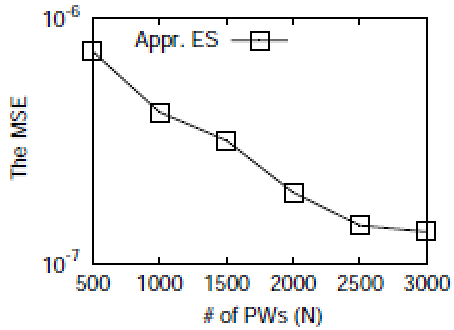
- Computing Similarities Exactly



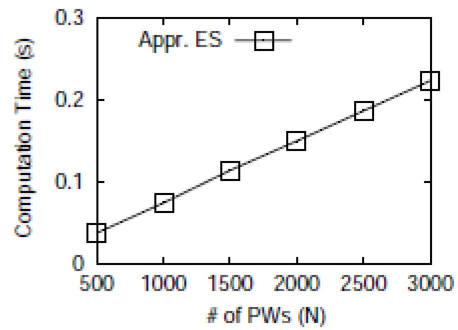
(a) Space consumption (b) Computation time

Experiments

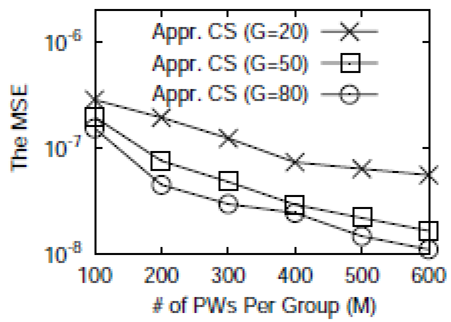
- Computing Similarities Approximately



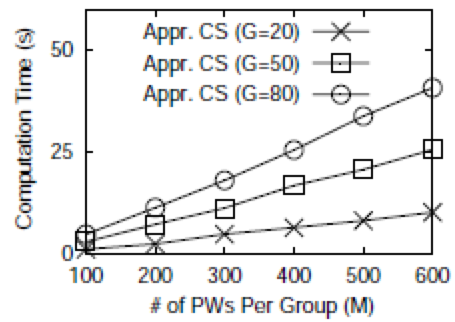
(a) *ES*, MSE



(b) *ES*, Computation time



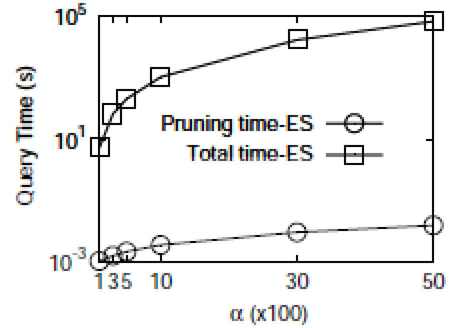
(c) *CS*, MSE



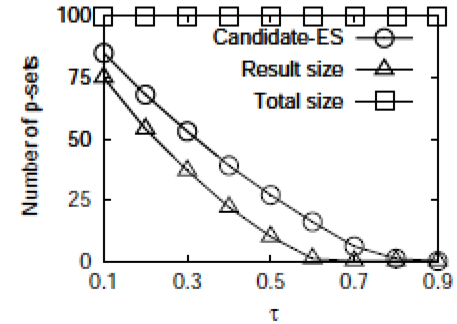
(d) *CS*, Computation time

Experiments

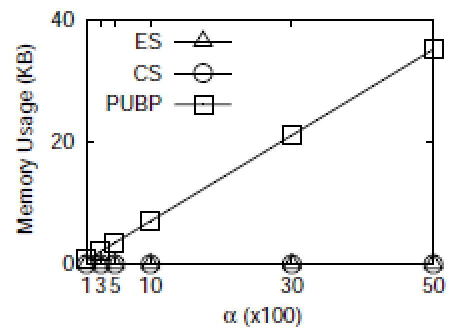
- Evaluating Pruning Efficiency on SYN



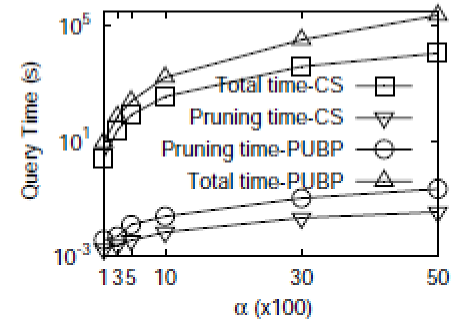
(a) *ESQ*, Query Time



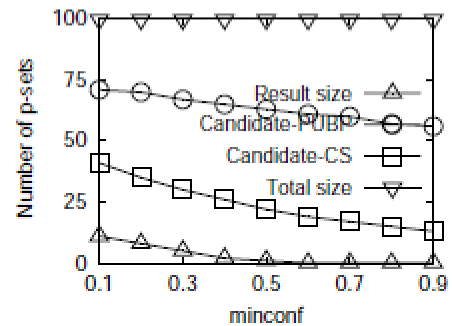
(b) *ESQ*, Candidate Size



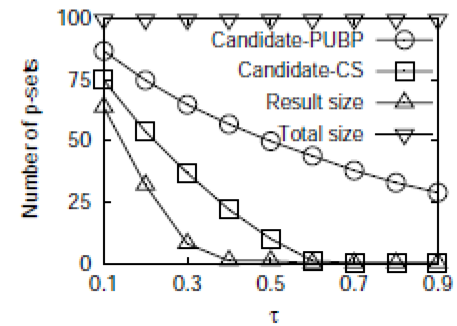
(c) *CSQ*, Memory Usage



(d) *CSQ*, Query Time



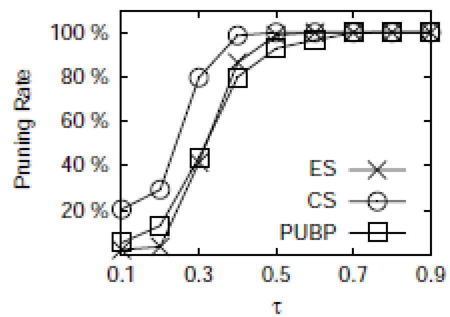
(e) *CSQ*, Candidate Size



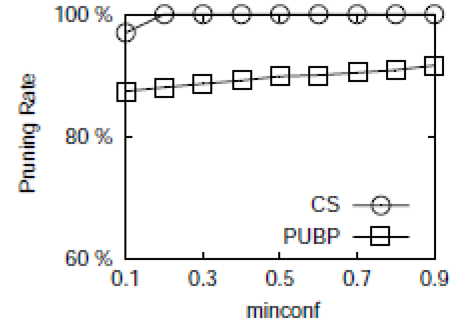
(f) *CSQ*, Candidate Size

Experiments

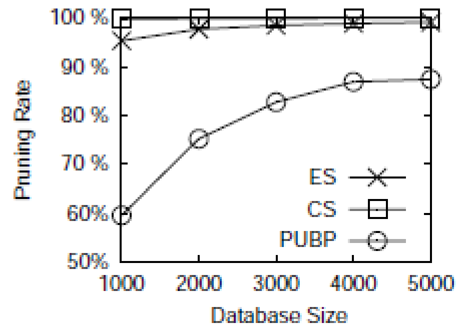
- Performance on the pDBLP Dataset



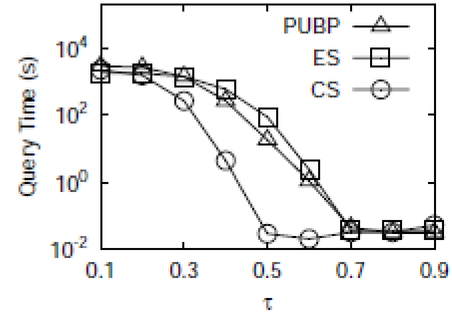
(a) Pruning Rate



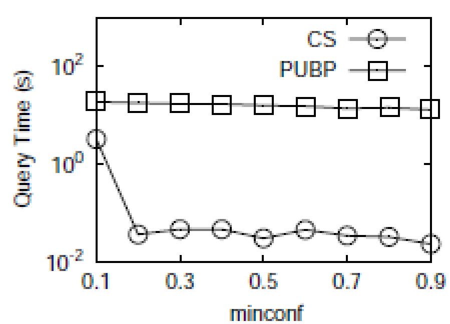
(b) Pruning Rate



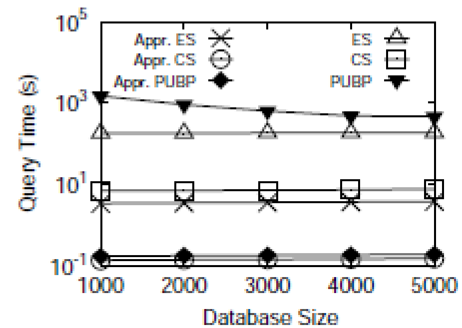
(c) Pruning Rate



(d) Query Time



(e) Query Time



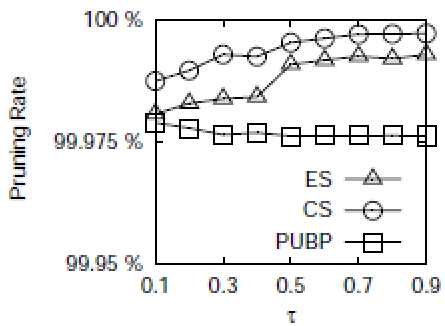
(f) Query Time

Experiments

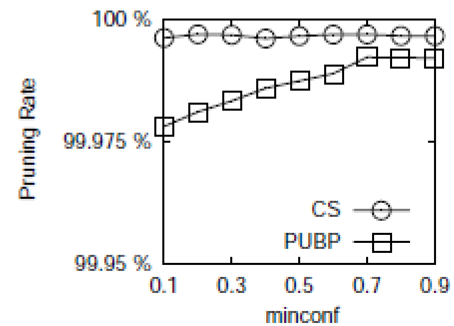
AVERAGE PRECISION@k ON PDELI

AP@k	5	10	15	20	25	30
ES	0.7000	0.6675	0.6250	0.5875	0.5825	0.5500
CS	0.7000	0.6785	0.6280	0.5825	0.575	0.5500

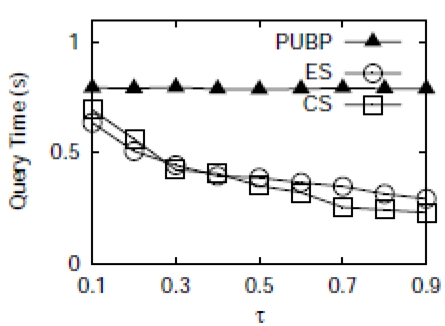
- Performance on the pDeli Dataset



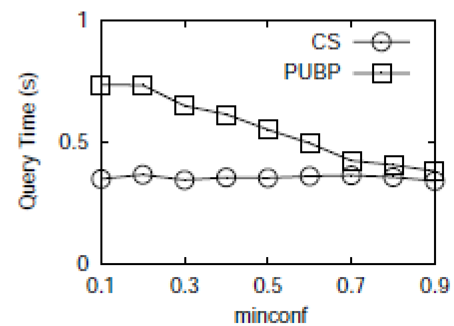
(a) Pruning Rate



(b) Pruning Rate



(c) Query Time



(d) Query Time

Thank You!