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

April 11, 2013

Outline

Motivation

- Similarity query processing is a fundamental and active research area in database community.
- Multi-label classification, e.g., a document can belong to multiple topics, and so on.
- The existing work on set similarity query processing upon probabilistic sets is still rare. Moreover, such work is hard to be scaled to large probabilistic sets (p-sets) due to both high time and space complexities.

Contributions

1. We define two types of similarity measures to capture different characteristics of the similarity between two p-sets;
2. We give efficient dynamic programming-based algorithm to compute these similarities;
3. We design novel individual and batch pruning techniques to speed up the query processing;
4. We conduct comprehensive experiments upon both synthetic and real datasets and demonstrate the efficiency and the effectiveness of the proposed approaches.

Preliminary

Probabilistic set model

We model a probabilistic set $\mathcal{A}$ in a domain $\mathcal{D}$ as

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

where $\forall i \neq j$, $a_i \neq a_j$.

Possible world model

The possible world space $\mathcal{W}(\mathcal{A})$ of $\mathcal{A}$ is the power set of $\mathcal{A}$, where each possible world $w(\mathcal{A}) \in \mathcal{W}(\mathcal{A})$ has a probability, which can be computed as

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

Similarity Metrics

1. Expected Similarity, *ES*

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

where *sim* can be Jaccard, Dice or Cosine.

2. Confidence-based Similarity, *CS*
   - Conditioned cumulative probability

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

   - Let $minconf \in [0,1]$ be a user-defined minimum confidence threshold. *CS* between $\mathcal{A}$ and $\mathcal{B}$ is defined as: $CS(\mathcal{A},\mathcal{B},minconf) = \max\{\, x \mid \mathbf{CPr}(x,\mathcal{A},\mathcal{B}) \geq minconf \,\}$.

Running Example

Table: Possible Worlds and Similarities

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

w_a	w_b	$\mathbf{Pr}[(w_a, w_b)]$	**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

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

Pre-processing

Normalized of Two P-sets

$$\mathcal{A} = \{ c_1 : p^{\mathcal{A}}_{c_1}, \ldots, c_k : p^{\mathcal{A}}_{c_k}, \; d_1 : p_{d_1}, \ldots, d_{n-k} : p_{d_{n-k}} \}$$

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

Computing $H[i,j]$

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

Dynamic Programming

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

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

Assuming $n \geq m \geq k$, the time complexity of computing $H^{m+n-k}[i,j]$, i.e., $H[i,j]$, $(0 \leq i \leq k$ and $0 \leq j \leq m+n-k)$ can be shown to be $\Theta(kn^2)$ or $O(n^3)$, and the space complexity is $\Theta(kn)$, or $O(n^2)$.

Exact Solution for *ES*

Table: $H[i,j]$

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

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

For *ES*, we simply compute the weighted average of all the entries in H. For example,

$$ES = (0+0+0.03+0.12)\times 0 + 0\times\frac{1}{1} + 0.1\times\frac{1}{2}$$
$$+ 0.4\times\frac{1}{3} + 0.07\times\frac{2}{2} + 0.28\times\frac{2}{3} = 0.44.$$

Exact Solution for *CS*

Let *minconf* be 0.5. We calculate *CS* by accessing entries of $H[i,j]$ by the decreasing order of the similarity values and stop whenever the answer is found.

Table: $H[i,j]$

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

Thus, $\mathbf{CPr}(1,\mathcal{A},\mathcal{B}) = 0.07$.

Exact Solution for *CS*

Let *minconf* be 0.5. We calculate *CS* by accessing entries of $H[i,j]$ by the decreasing order of the similarity values and stop whenever the answer is found.

Table: $H[i,j]$

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

Thus, $\mathbf{CPr}(1,\mathcal{A},\mathcal{B}) = 0.07$.

Exact Solution for *CS*

Let *minconf* be 0.5. We calculate *CS* by accessing entries of $H[i,j]$ by the decreasing order of the similarity values and stop whenever the answer is found.

Table: $H[i,j]$

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

Similarly, $\mathbf{CPr}(\frac{2}{3}, \mathcal{A}, \mathcal{B}) = 0.35$.

Exact Solution for *CS*

Let *minconf* be 0.5. We calculate *CS* by accessing entries of $H[i,j]$ by the decreasing order of the similarity values and stop whenever the answer is found.

Table: $H[i,j]$

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

Similarly, $\mathbf{CPr}(\frac{1}{2}, \mathcal{A}, \mathcal{B}) = 0.45$.

Exact Solution for *CS*

Let *minconf* be 0.5. We calculate *CS* by accessing entries of $H[i,j]$ by the decreasing order of the similarity values and stop whenever the answer is found.

Table: $H[i,j]$

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

Similarly, $\mathbf{CPr}(\frac{1}{3},\mathcal{A},\mathcal{B}) = 0.85$. Since $\mathbf{CPr}(\frac{1}{3},\mathcal{A},\mathcal{B}) = 0.85 > minconf$ and $\mathbf{CPr}(\frac{1}{2},\mathcal{A},\mathcal{B}) = 0.45 < minconf$, $CS(\mathcal{A},\mathcal{B},minconf) = \frac{1}{3}$.

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

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

Preliminary

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

Batch Pruning

1. We index all the p-sets in the database by their expected sizes;
2. We compute a lower bound S_L and an upper bound S_U of the expected size for the appropriate query type;
3. We consider p-sets whose expected sizes fall within $[S_L, S_U]$.

Upper Bounds

1. Upper Bound for *ES*: query p-set $\mathcal{Q}$, a data p-set $\mathcal{O}$, the similarity threshold τ, $\mathcal{O}$ can be pruned if $UB_1(\mathbf{E}[|\mathcal{Q} \cap \mathcal{O}|], \mathbf{E}[|\mathcal{Q} \cup \mathcal{O}|]) \leq \tau$, where

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

2. Upper Bound for *CS*: given a *CS* query p-set $\mathcal{Q}$, a data p-set $\mathcal{O}$, parameters *minconf* and τ. Then $\mathcal{O}$ can be pruned if (i) $\mathbf{E}[|\mathcal{Q} \cap \mathcal{O}|] \leq \tau \cdot \mathbf{E}[|\mathcal{Q} \cup \mathcal{O}|]$, and (ii) $UB_2(\mathbf{E}[|\mathcal{Q} \cap \mathcal{O}|], \mathbf{E}[|\mathcal{Q} \cup \mathcal{O}|], \tau) \leq \mathit{minconf}$, where

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

Approximate Solutions

Sampling-based Method for *ES*

We randomly sample $\lceil (\ln \frac{2}{\delta})/(2\epsilon^2) \rceil$ joint possible worlds (where ϵ and δ refer to the error threshold and confidence parameter respectively) and use the average of the similarity in the sampled possible worlds to approximate the expected similarity *ES*.

Sampling-based Method for *CS*

For any ϵ, δ, we randomly generate $24 \cdot \lceil \ln \frac{1}{\delta} \rceil$ groups of possible worlds, and each group contains $\lceil 2\epsilon^{-2} \rceil$ pairs of possible worlds from $\mathcal{A}$ and $\mathcal{B}$. In each group, we select the $(minconf \cdot M)$-th largest similarity value into an array *sa*. Finally, we select the median value from G entries in the *sa* array.

We have the error guarantee of these two approximate computations.

Experiment Settings

Data sets

- SYN
- pDBLP
- pDeli

Measures

- Efficiency measure: memory usage; computation time; pruning time.
- Effectiveness measure: candidate size; result size; pruning rate; average precision.

Performance on SYN

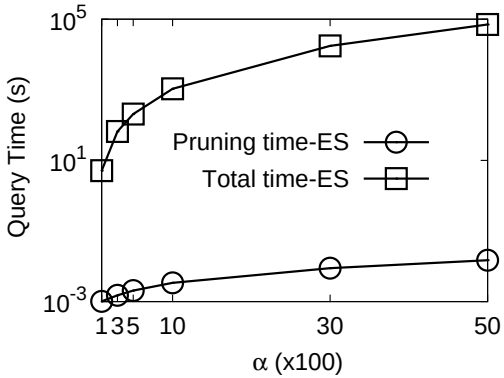

Figure: Query Time

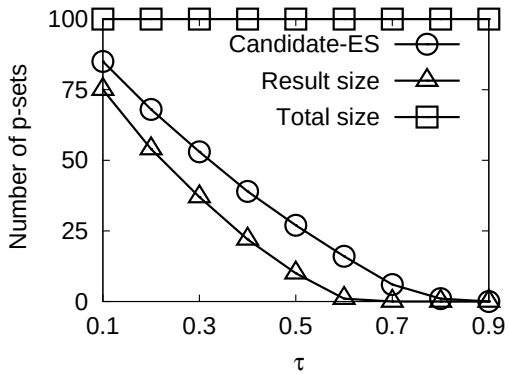

Figure: Candidate Size

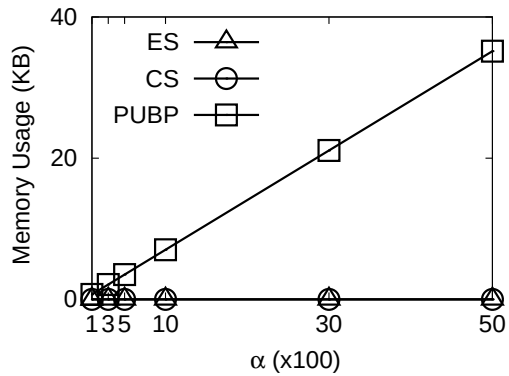

Figure: Memory Usage

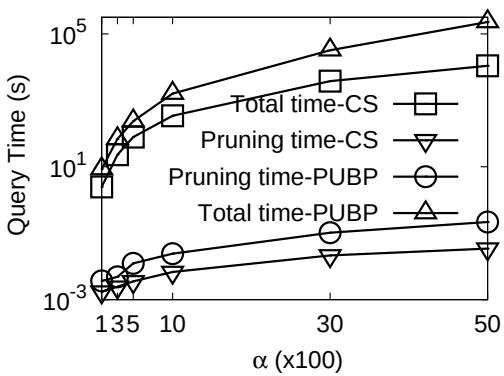

Figure: Query Time

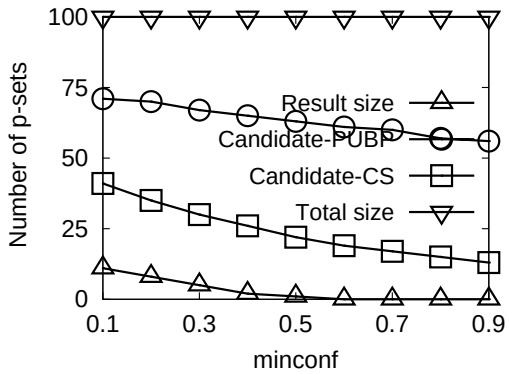

Figure: Candidate Size

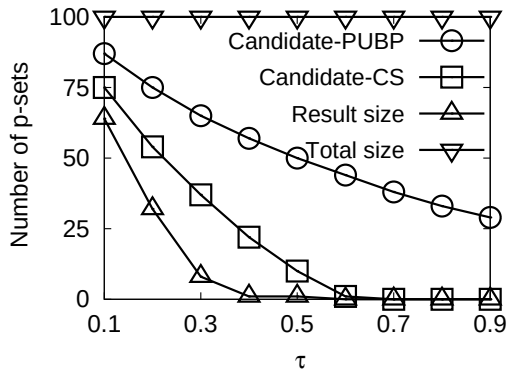

Figure: Candidate Size

Performance on pDBLP

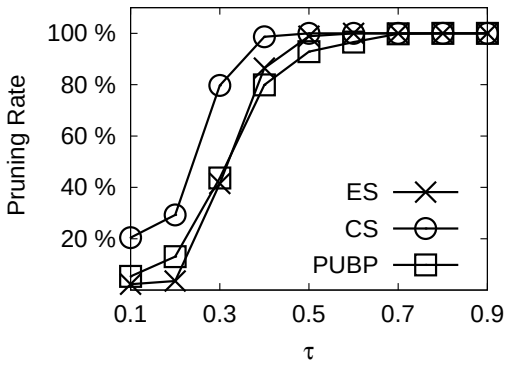

Figure: Pruning Rate

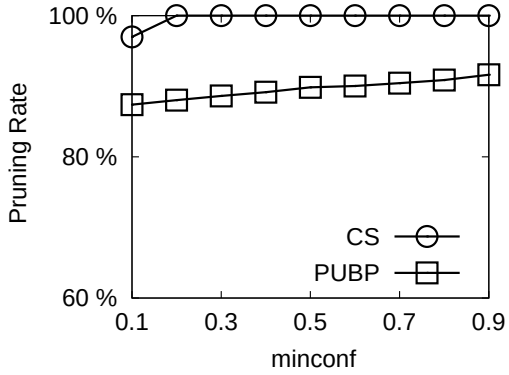

Figure: Pruning Rate

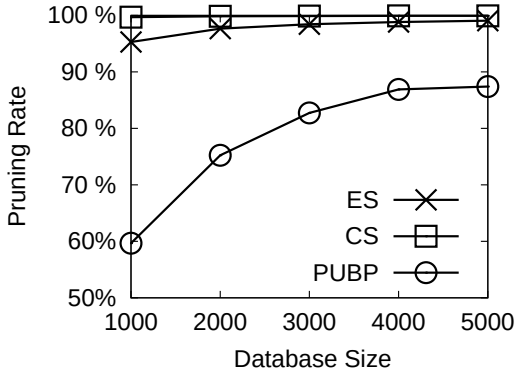

Figure: Pruning Rate

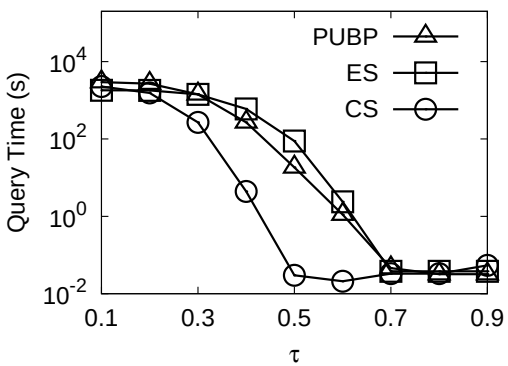

Figure: Query Time

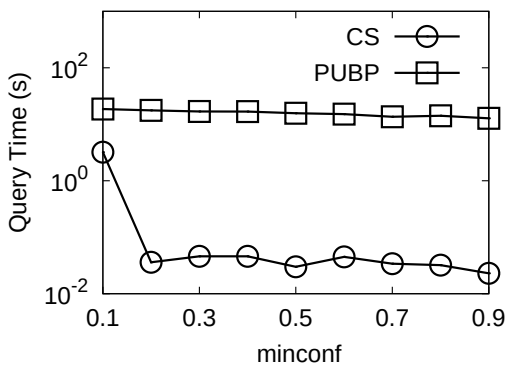

Figure: Query Time

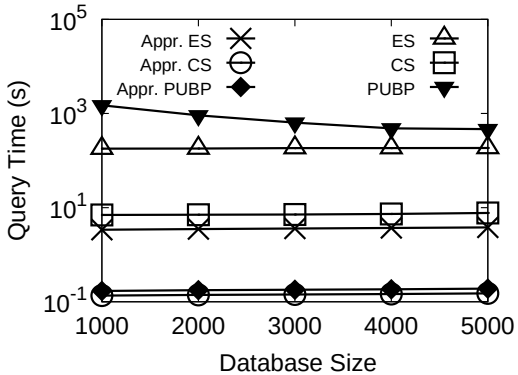

Figure: Query Time

Case Study on pDBLP

Table: Sample Query Results on pDBLP

Query Author	Top-3 Similar Authors
Hanan Samet	Thomas Seidl, Walid G. Aref, Pavel Zezula
Jeffrey D. Ullman	Leonid Libkin, Yehoshua Sagiv, Richard J. Lipton
Michael I. Jordan	Zoubin Ghahramani, Eric P. Xing, John Shawe-Taylor

As we can see, the top-3 results are indeed researchers with closely matching research interest with the query author.

Performance on pDeli

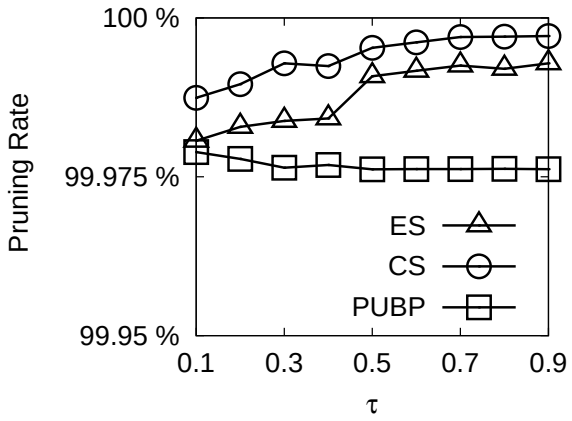

Figure: Pruning Rate

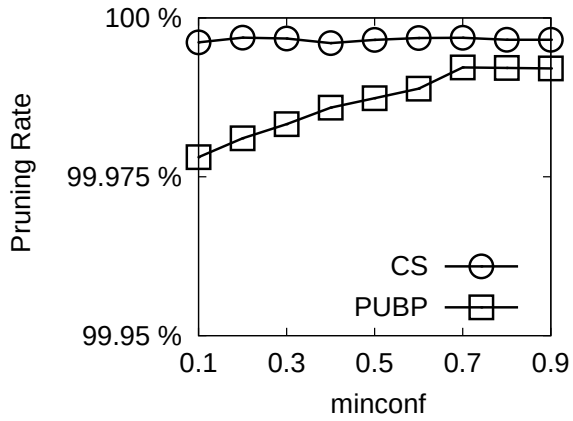

Figure: Pruning Rate

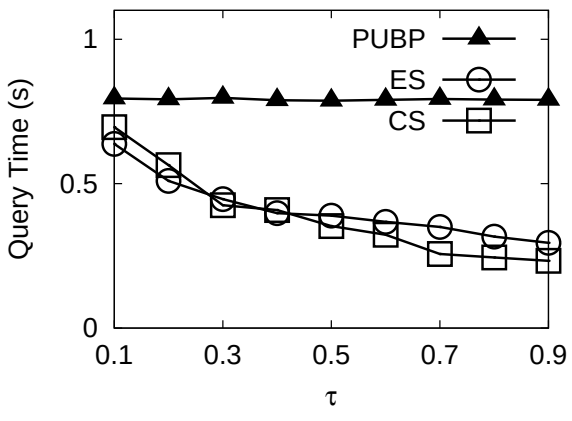

Figure: Query Time

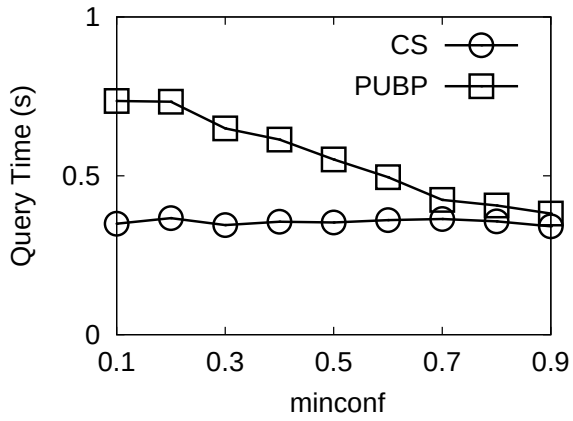

Figure: Query Time

Conclusion

1. We then study two kinds of similarity queries based on the expected and confidence-based similarity measures.
2. Both exact and approximate algorithms are developed to compute these values.
3. We develop novel pruning techniques based on upper bound estimation.
4. Both the theoretical analysis and our empirical evaluation demonstrate the effectiveness and efficiency of the proposed methods.

Acknowledgement

Q & A

Thank you for your attention!