Problem

Observing that knowledge bases by human collaboration and information extraction are often incomplete, we tackle the knowledge expansion problem, i.e., inferring missing facts in knowledge bases.

Key Challenges:
- Ambiguity: Knowledge bases are often big.
- Accuracy: Extracted facts and learned rules are sometimes noisy.

Contributions

We present ProbKB, a PROBabilistic Knowledge Base system. We achieve high efficiency and quality via the following contributions:
- We design a novel relational model for probabilistic knowledge bases.
- We present a SQL-based inference algorithm that applies inference rules in batches. We use MPP databases to parallelize the inference process.
- We combine state-of-the-art quality control methods to detect and recover from errors in the inference procedure.
- We show the performance of ProbKB system through a comprehensive experimental study over real and synthetic knowledge bases.

System Overview

ProbKB Relational Model

Example KB

<table>
<thead>
<tr>
<th>I</th>
<th>R</th>
<th>x</th>
<th>C1</th>
<th>y</th>
<th>C2</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RG</td>
<td>W</td>
<td>Br</td>
<td>P</td>
<td>0.93</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RG</td>
<td>W</td>
<td>NYC</td>
<td>C</td>
<td>0.96</td>
<td></td>
</tr>
</tbody>
</table>

- \( \forall x \in W \forall y \in P \text{ (live in}(x, y) \leftarrow \text{born in}(x, y)) \) 1.40
- \( \forall x \in W \forall y \in C \text{ (grow up in}(x, y) \leftarrow \text{born in}(x, y)) \) 2.08
- \( \forall x \in W \forall y \in C \text{ (live in}(x, y) \leftarrow \text{born in}(x, y)) \) 0.74

Applying Rules in Batches

```
```

Quality Control

Error Sources
- Incorrect extractions
- Incorrect rules
- Ambiguous entities
- Propagated errors

Approaches
- Functional constraints
- Statistical rule cleaning

Efficiency

```
<table>
<thead>
<tr>
<th>Execution time/10^3</th>
<th># Inferred facts/10^6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>1.20</td>
</tr>
<tr>
<td>0.20</td>
<td>2.00</td>
</tr>
<tr>
<td>1.00</td>
<td>2.00</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>I</th>
<th>R</th>
<th>x</th>
<th>C1</th>
<th>y</th>
<th>C2</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RG</td>
<td>W</td>
<td>Br</td>
<td>P</td>
<td>0.93</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RG</td>
<td>W</td>
<td>NYC</td>
<td>C</td>
<td>0.96</td>
<td></td>
</tr>
</tbody>
</table>

```

```
PRoKB performance on KBs with varying number of rules, achieving over 300 times improvement compared to the TUFFY.

Quality

Combining functional constraints and statistical rule cleaning, we improve precision by \~60%.

Error sources distribution: Most erroneous facts are caused by ambiguous entities used as join keys and incorrect rules applied repeatedly during inference.

Related Work

- TUFFY: Scaling up Statistical Inference in Markov Logic Networks using an RDBMS
  http://hazy.cs.wisc.edu/hazy/tuffy
- OPENIE: Open Information Extraction
  http://openie.cs.washington.edu
- SHERLOCK: Learning First-Order Horn Clauses from Web Text
  http://www.cs.washington.edu/research/sherlock-hornclauses
- LEINIZ: Identifying Functional Relations in Web Text
  http://knowitall.cs.washington.edu/leiniz

Knowledge Expansion over Probabilistic Knowledge Bases

Yang Chen and Daisy Zhe Wang
{yang, daisyw}@cise.ufl.edu

# Knowledge Expansion over Probabilistic Knowledge Bases

## Problem

Observing that knowledge bases by human collaboration and information extraction are often incomplete, we tackle the knowledge expansion problem, i.e., inferring missing facts in knowledge bases.

### Key Challenges:
- **Ambiguity**: Knowledge bases are often big.
- **Accuracy**: Extracted facts and learned rules are sometimes noisy.

## Contributions

We present ProbKB, a PROBabilistic Knowledge Base system. We achieve high efficiency and quality via the following contributions:

- We design a novel relational model for probabilistic knowledge bases.
- We present a SQL-based inference algorithm that applies inference rules in batches. We use MPP databases to parallelize the inference process.
- We combine state-of-the-art quality control methods to detect and recover from errors in the inference procedure.
- We show the performance of ProbKB system through a comprehensive experimental study over real and synthetic knowledge bases.

## System Overview

### ProbKB Relational Model

**Example KB**

<table>
<thead>
<tr>
<th>I</th>
<th>R</th>
<th>x</th>
<th>C1</th>
<th>y</th>
<th>C2</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RG</td>
<td>W</td>
<td>Br</td>
<td>P</td>
<td>0.93</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RG</td>
<td>W</td>
<td>NYC</td>
<td>C</td>
<td>0.96</td>
<td></td>
</tr>
</tbody>
</table>

- \( \forall x \in W \forall y \in P \text{ (live in}(x, y) \leftarrow \text{born in}(x, y)) \) 1.40
- \( \forall x \in W \forall y \in C \text{ (grow up in}(x, y) \leftarrow \text{born in}(x, y)) \) 2.08
- \( \forall x \in W \forall y \in C \text{ (live in}(x, y) \leftarrow \text{born in}(x, y)) \) 0.74

### Applying Rules in Batches

```
```

### Quality Control

**Error Sources**

- Incorrect extractions
- Incorrect rules
- Ambiguous entities
- Propagated errors

**Approaches**

- Functional constraints
- Statistical rule cleaning

### Efficiency

```
<table>
<thead>
<tr>
<th>Execution time/10^3</th>
<th># Inferred facts/10^6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>1.20</td>
</tr>
<tr>
<td>0.20</td>
<td>2.00</td>
</tr>
<tr>
<td>1.00</td>
<td>2.00</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>I</th>
<th>R</th>
<th>x</th>
<th>C1</th>
<th>y</th>
<th>C2</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RG</td>
<td>W</td>
<td>Br</td>
<td>P</td>
<td>0.93</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RG</td>
<td>W</td>
<td>NYC</td>
<td>C</td>
<td>0.96</td>
<td></td>
</tr>
</tbody>
</table>
```

PRoKB performance on KBs with varying number of rules, achieving over 300 times improvement compared to the TUFFY.

### Quality

Combining functional constraints and statistical rule cleaning, we improve precision by \~60%.

Error sources distribution: Most erroneous facts are caused by ambiguous entities used as join keys and incorrect rules applied repeatedly during inference.

## Related Work

- **TUFFY**: Scaling up Statistical Inference in Markov Logic Networks using an RDBMS
  http://hazy.cs.wisc.edu/hazy/tuffy
- **OPENIE**: Open Information Extraction
  http://openie.cs.washington.edu
- **SHERLOCK**: Learning First-Order Horn Clauses from Web Text
  http://www.cs.washington.edu/research/sherlock-hornclauses
- **LEINIZ**: Identifying Functional Relations in Web Text
  http://knowitall.cs.washington.edu/leiniz