Random Walk Inference and Learning in A Large Scale Knowledge Base

Ni Lao, Tom Mitchell, William W. Cohen
Carnegie Mellon University
2011.7.28
Outline

• Motivation
  – Inference in Knowledge-Bases
  – The NELL project
  – Random Walk Inference

• Approach
  – Path Ranking Algorithm (Recap)
  – Data-Driven Path Finding
  – Efficient Random Walk (Recap)
  – Low-Variance Sampling

• Results
  – Cross Validation
  – Mechanical Turk Evaluation
Large Scale Knowledge-Bases

• Human knowledge is being transformed into structured data at a fast speed, e.g.
  
  – KnowItAll (Univ. Washington)
    • 0.5B facts extracted from 0.1B web pages
  
  – DBpedia (Univ. Leipzig)
    • 3.5M entities 0.7B facts extracted from wikipedia
  
  – YAGO (Max-Planck Institute)
    • 2M entities  20M facts extracted from Wikipedia and wordNet
  
  – FreeBase
    • 20M entities 0.3B links, integrated from different data sources and human judgments
  
  – NELL (Carnegie Mellon Univ.)
    • 0.85M facts extracted from 0.5B webpages
The Need for Robust and Efficient Inference

• Knowledge is potentially useful in many tasks
  – Support information retrieval/recommendation
  – Bootstrap information extraction/integration

• Challenges
  – **Robustness**: extracted knowledge is incomplete and noisy
  – **Scalability**: the size of knowledge base can be very large
The NELL Case Study

• **Never-Ending Language Learning:**
  – “a never-ending learning system that operates 24 hours per day, for years, to continuously improve its ability to read (extract structured facts from) the web” (Carlson et al., 2010)

  – Closed domain, semi-supervised extraction
  – Combines multiple strategies: morphological patterns, textual context, html patterns, logical inference

  – Example beliefs

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>cityInState</td>
<td>(troy, Michigan)</td>
</tr>
<tr>
<td>musicArtistGenre</td>
<td>(Nirvana, Grunge)</td>
</tr>
<tr>
<td>tvStationInCity</td>
<td>(WLS-TV, Chicago)</td>
</tr>
<tr>
<td>sportUsesEquip</td>
<td>(soccer, balls)</td>
</tr>
<tr>
<td>athleteInLeague</td>
<td>(Dan Fouts, NFL)</td>
</tr>
<tr>
<td>starredIn</td>
<td>(Will Smith, Seven Pounds)</td>
</tr>
<tr>
<td>productType</td>
<td>(Acrobat Reader, FILE)</td>
</tr>
<tr>
<td>athletePlaysSport</td>
<td>(scott shields, baseball)</td>
</tr>
<tr>
<td>cityInCountry</td>
<td>(Dublin Airport, Ireland)</td>
</tr>
</tbody>
</table>
A Link Prediction Task

• We consider 48 relations for which NELL database has more than 100 instances

• We create two link prediction tasks for each relation
  – AthletePlaysInLeague(HinesWard, ?)
  – AthletePlaysInLeague(?, NFL)

• The actual nodes y known to satisfy R(x; ?) are treated as labeled positive examples, and all other nodes are treated as negative examples
First Order Inductive Learner

- FOIL (Quinlan and Cameron-Jones, 1993) is a learning algorithm similar to decision trees, but in relational domains

- NELL implements two assumptions for efficient learning (N-FOIL)
  - The predicates are functional -- e.g. an athlete plays in at most one league
  - Only find clauses that correspond to bounded-length paths of binary relations -- relational pathfinding (Richards & Mooney, 1992)
First Order Inductive Learner

• Efficiency
  – Horn clauses can be very costly to evaluate
  – E.g. it takes days to train N-FOIL on the NELL data

• Robustness
  – FOIL can only combine rules with disjunctions, therefore cannot leverage low accuracy rules
  – E.g. rules for teamPlaysSports

\[
\begin{align*}
C \quad \text{teamAlsoKnownAs} & \rightarrow C \\
C \quad \text{teamPlaysSport} & \rightarrow C \\
C \quad \text{teamHomeStadium} & \rightarrow C \\
C \quad \text{stadiumHomeToSport} & \rightarrow C \\
C \quad \text{teamMember} & \rightarrow C \\
C \quad \text{athletePlaysSport} & \rightarrow C \\
C \quad \text{teamPlaysAgainstTeam} & \rightarrow C \\
C \quad \text{teamPlaysSport} & \rightarrow C
\end{align*}
\]

High accuracy but low recall
Random Walk Inference

- Consider a low precision/high recall Horn clause
  - $\text{isa}(x, c) \land \text{isa}(x', c) \land \text{AthletePlaysInLeague}(x', y) \Rightarrow \text{AthletePlaysInLeague}(x; y)$

- A Path Constrained Random Walk following the above edge type
  sequence generates a distribution over all leagues

- $\text{Prob}(\text{HinesWard} \Rightarrow y)$ can be treated as a relational feature for
  predicting $\text{AthletePlaysInLeague}(\text{HinesWard}; y)$

All mentions of "EMNLP 2011, Edinburgh, Scotland, UK" should be removed from the natural text.
Comparison

• Inductive logic programming (e.g. FOIL)
  – Brittle facing uncertainty

• Statistical relational learning (e.g. Markov logic networks, Relational Bayesian Networks)
  – Inference is costly when the domain contains many nodes
  – Inference is needed at each iteration of optimization

• Random walk inference
  – Decouples feature generation and learning (propositionalization)
    • No inference needed during optimization

    – Sampling schemes for efficient random walks
      • Trains in minutes as opposed to days for N-FOIL

    – Low precision/high recall rules as features with fractional values
      • Doubles precision at rank 100 compared with N-FOIL
Outline

• Motivation
  – Inference in Knowledge-Bases
  – The NELL project
  – Random Walk Inference

• Approach
  – Path Ranking Algorithm (Recap)
  – Data-Driven Path Finding
  – Efficient Random Walk (Recap)
  – Low-Variance Sampling

• Results
  – Cross Validation
  – Mechanical Turk Evaluation
Path Ranking Algorithm (PRA) 
(Lao & Cohen, ECML 2010)

- A relation path $P=(R_1, ..., R_n)$ is a sequence of relations
- A PRA model scores a source-target node pair by a linear function of their path features

$$score(s,t) = \sum_{P \in \mathcal{P}} f_P(s,t) \theta_P$$

- $\mathcal{P}$ is the set of all relation paths with length $\leq L$
- $f_P(s,t) = \text{Prob}(s \rightarrow t; P)$

- Training
  - For a relation $R$ and a set of node pairs $\{(s_i, t_i)\}$,
  - we construct a training dataset $D = \{(x_i, y_i)\}$, where
  - $x_i$ is a vector of all the path features for $(s_i, t_i)$, and
  - $y_i$ indicates whether $R(s_i, t_i)$ is true or not
  - $\theta$ is estimated using L1,L2-regularized logistic regression
Data-Driven Path Finding

• Impractical to enumerate all possible paths even for small length \( l \)
  – Require any path to instantiate in at least \( \alpha \) portion of the training queries, i.e. \( f_p(s,t) \neq 0 \) for any \( t \)
  – Require any path to reach at least one target node in the training set

• Discover paths by a depth first search
  – Starts from a set of training queries, expand a node if the instantiation constraint is satisfied
Data-Driven Path Finding

• Dramatically reduce the number of paths

Table 1: Number of paths in PRA models of maximum path length 3 and 4. Averaged over 96 tasks.

<table>
<thead>
<tr>
<th></th>
<th>$\ell=3$</th>
<th>$\ell=4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>all paths up to length $\ell$</td>
<td>15,376</td>
<td>1,906,624</td>
</tr>
<tr>
<td>+query support $\geq \alpha = 0.01$</td>
<td>522</td>
<td>5016</td>
</tr>
<tr>
<td>+ever reach a target entity</td>
<td>136</td>
<td>792</td>
</tr>
<tr>
<td>+$L_1$ regularization</td>
<td>63</td>
<td>271</td>
</tr>
</tbody>
</table>
Efficient Inference

(Lao & Cohen, KDD 2010)

• Exact calculation of random walk distributions results in non-zero probabilities for many internal nodes in the graph

• but computation should be focused on the few target nodes which we care about
Efficient Inference

(Lao & Cohen, KDD 2010)

• Sampling approach
  – A few random walkers (or particles) are enough to distinguish good target nodes from bad ones
Low-Variance Sampling

• Sampling walkers/particles independently introduces variances to the result distributions

• Low-Variance Sampling (LVS)(Thrun et al., 2005) generates $M$ correlated samples, by drawing a single number $r$ from $(0, M^{-1})$

samples correspond to $M^{-1} + kr$, $k=0..M-1$

(Detail)
Low Variance Sampling

• In our evaluation
  – LVS can slightly improve prediction for both fingerprinting and particle filtering
Outline

• Motivation
  – Inference in Knowledge-Bases
  – The NELL project
  – Random Walk Inference

• Approach
  – Path Ranking Algorithm (Recap)
  – Data-Driven Path Finding
  – Efficient Random Walk (Recap)
  – Low-Variance Sampling

• Results
  – Cross Validation
  – Mechanical Turk Evaluation
Parameter Tuning

• Cross Validation on Training Queries
  – Supervised training can improve retrieval quality (RWR)
  – Path structure can produce further improvement (PRA)

---

Table 3: Compare PRA with RWR models. MRRs and training times are averaged over 96 tasks.

<table>
<thead>
<tr>
<th></th>
<th>( l=2 )</th>
<th>( l=3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR Training</td>
<td>MRR Training</td>
</tr>
<tr>
<td>RWR(no train)</td>
<td>0.271(^\dagger)</td>
<td>0.456(^\dagger)</td>
</tr>
<tr>
<td>RWR</td>
<td>0.280</td>
<td>0.471</td>
</tr>
<tr>
<td>PRA</td>
<td>0.307(^\dagger)</td>
<td>0.516(^\dagger)</td>
</tr>
<tr>
<td></td>
<td>3.7s</td>
<td>9.2s</td>
</tr>
<tr>
<td></td>
<td>5.7s</td>
<td>15.4s</td>
</tr>
</tbody>
</table>

RWR: Random Walk with Restart (personalized page rank)
\(^\dagger\)Paired t-test give p-values 7\( \times 10^{-3} \), 9\( \times 10^{-4} \), 9\( \times 10^{-8} \), 4\( \times 10^{-4} \)
Example Paths

**athletePlaysSport**

\[
\begin{align*}
C & \xrightarrow{isa} C & \xrightarrow{isa^{-1}} C & \xrightarrow{\text{athletePlaysSport}} C \\
C & \xrightarrow{\text{athletePlaysInLeague}} C & \xrightarrow{\text{superpartOfOrganization}} C & \xrightarrow{\text{teamPlaysSport}} C
\end{align*}
\]

**teamHomeStadium**

\[
\begin{align*}
C & \xrightarrow{\text{teamPlaysInCity}} C & \xrightarrow{\text{cityStadiums}} C \\
C & \xrightarrow{\text{teamMember}} C & \xrightarrow{\text{athletePlaysForTeam}} C & \xrightarrow{\text{teamHomeStadium}} C
\end{align*}
\]

Synonyms of the query team
Evaluation by Mechanical Turk

• There are many test queries per predicate
  – All entities of a predicate’s domain/range, e.g.
    • WorksFor(person, organization)
  – On average 7,000 test queries for each functional predicate, and 13,000 for each non-functional predicate

• Sampled evaluation
  – We only evaluate the top ranked result for each query
  – We sort the queries for each predicate according to the scores of their top ranked results, and then evaluate precisions at top 10, 100 and 1000 queries

• Each belief is voted by 5 workers
  – Workers are given assertions like “Hines Ward plays for the team Steelers”, as well as Google search links for each entity
Evaluation by Mechanical Turk

- On 8 functional predicates where N-FOIL can successfully learn
  - PRA is comparable to N-FOIL for p@10, but has significantly better p@100

- On randomly sampled 8 non-functional (one to many mapping) predicates
  - Slightly lower accuracy than functional predicates

<table>
<thead>
<tr>
<th>Task</th>
<th>#Rules</th>
<th>N-FOIL p@10</th>
<th>p@100</th>
<th>#Paths</th>
<th>PRA p@10</th>
<th>p@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Predicates</td>
<td>2.1(+37)</td>
<td><strong>0.76</strong></td>
<td><strong>0.380</strong></td>
<td>43</td>
<td><strong>0.79</strong></td>
<td>0.668</td>
</tr>
<tr>
<td>Non-functional Predicates</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>92</td>
<td><strong>0.65</strong></td>
<td>0.620</td>
</tr>
</tbody>
</table>

PRA: Path Ranking Algorithm
Conclusion

• Random walk inference
  – Generate path features for link prediction tasks
  – Use sampling schemes for efficient inference
  – Use low precision rules as fractional valued features

• Future work (in model expressiveness)
  – Efficiently discover long paths
  – Discover lexicalized paths (contains constant nodes)
  – Generalize relation paths to trees/networks

• Thank you! Questions?