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

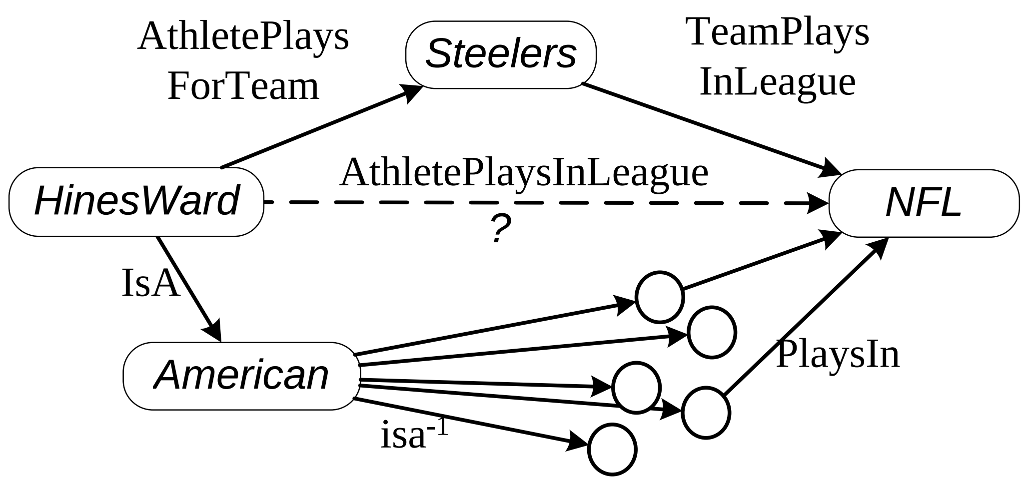
Large Scale Knowledge-Bases

- 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

Predicate	**Instance**
cityInState	(troy, Michigan)
musicArtistGenre	(Nirvana, Grunge)
tvStationInCity	(WLS-TV, Chicago)
sportUsesEquip	(soccer, balls)
athleteInLeague	(Dan Fouts, NFL)
starredIn	(Will Smith, Seven Pounds)
productType	(Acrobat Reader, FILE)
athletePlaysSport	(scott shields, baseball)
cityInCountry	(Dublin Airport, Ireland)

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 take 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

High accuracy
but low recall

$$c \xrightarrow{teamAlsoKnownAs} c \xrightarrow{teamPlaysSport} c$$

$$c \xrightarrow{teamHomeStadium} c \xrightarrow{stadiumHomeToSport} c$$

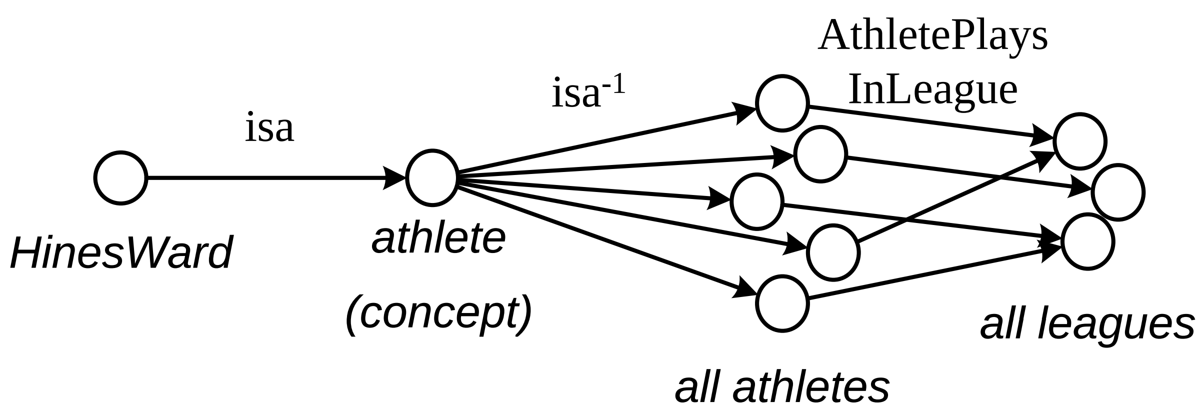
$$c \xrightarrow{teamMember} c \xrightarrow{athletePlaysSport} c$$

$$c \xrightarrow{teamPlaysAgainstTeam} c \xrightarrow{teamPlaysSport} c$$

Random Walk Inference

- Consider a low precision/high recall Horn clause
  - isa(x, c) ^ isa(x',c)^ AthletePlaysInLeague(x', y) ➔ AthletePlaysInLeague(x; y)

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



- Prob(HinesWard → y) can be treated as a relational feature for predicting AthletePlaysInLeague(HinesWard; y)

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

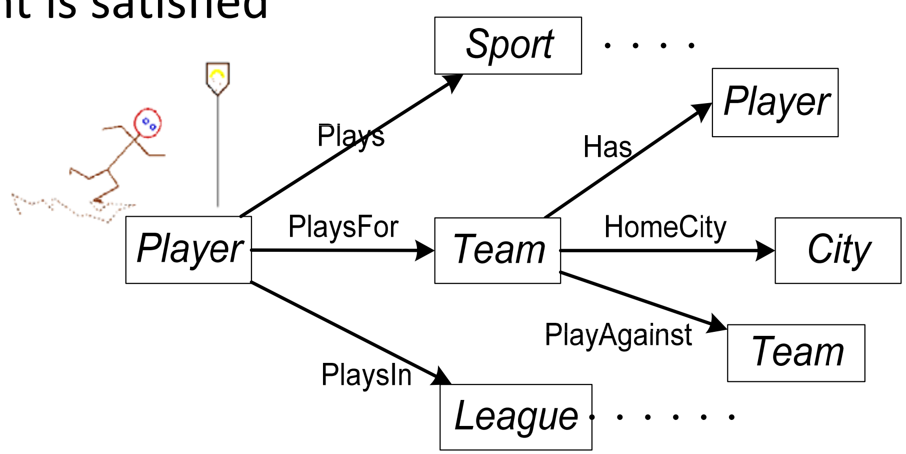
$$score(s,t) = \sum_{P \in \mathbf{P}} f_P(s,t)\theta_P$$

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

- 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
  - θ 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 instantiat e in at least α 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

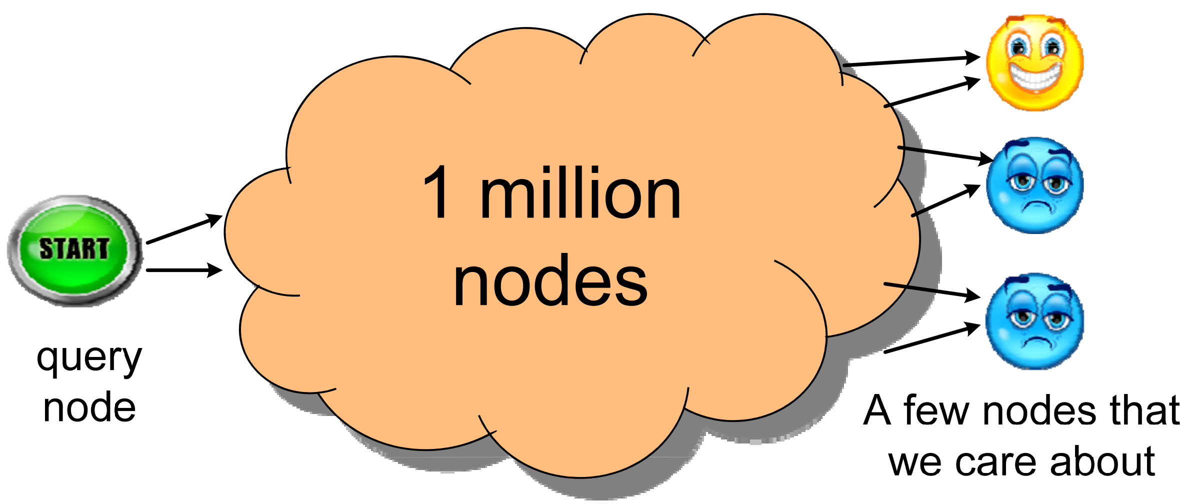
Table 1: Number of paths in PRA models of maximum path length 3 and 4. Averaged over 96 tasks.

	ℓ=3	ℓ=4
all paths up to length ℓ	$15,376$	$1,906,624$
+query support$\geq \alpha = 0.01$	522	5016
+ever reach a target entity	136	792
+L_1 regularization	63	271

Efficient Inference

(Lao & Cohen, KDD 2010)

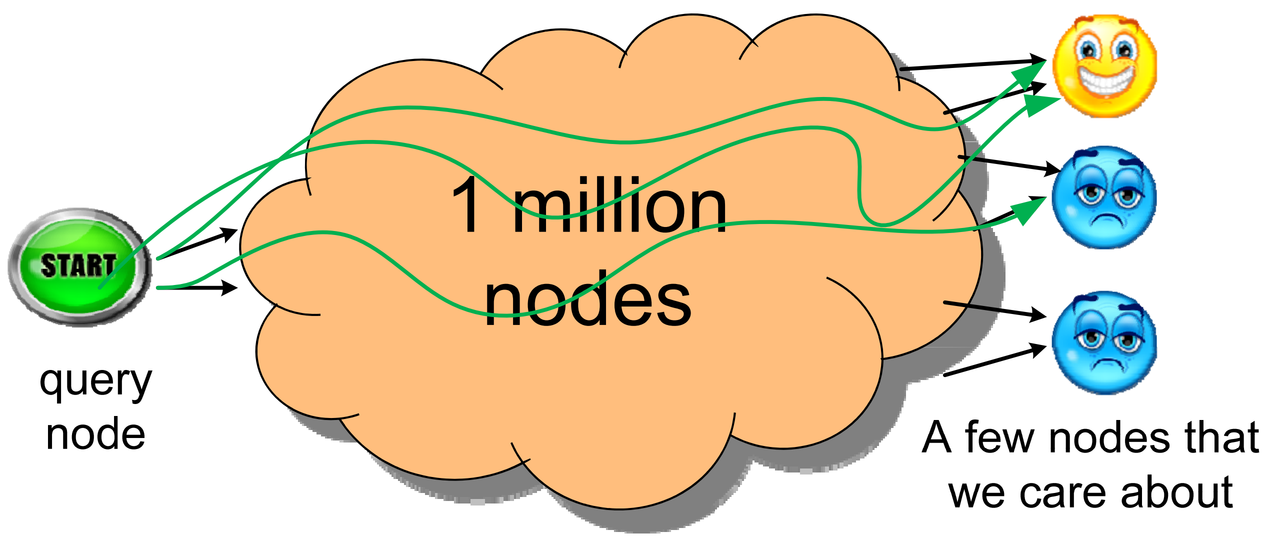
- Exact calculation of random walk distributions results in non-zero probabilities for many internal nodes in the graph
- but computation should be focused on the few target nodes which we care about



Efficient Inference

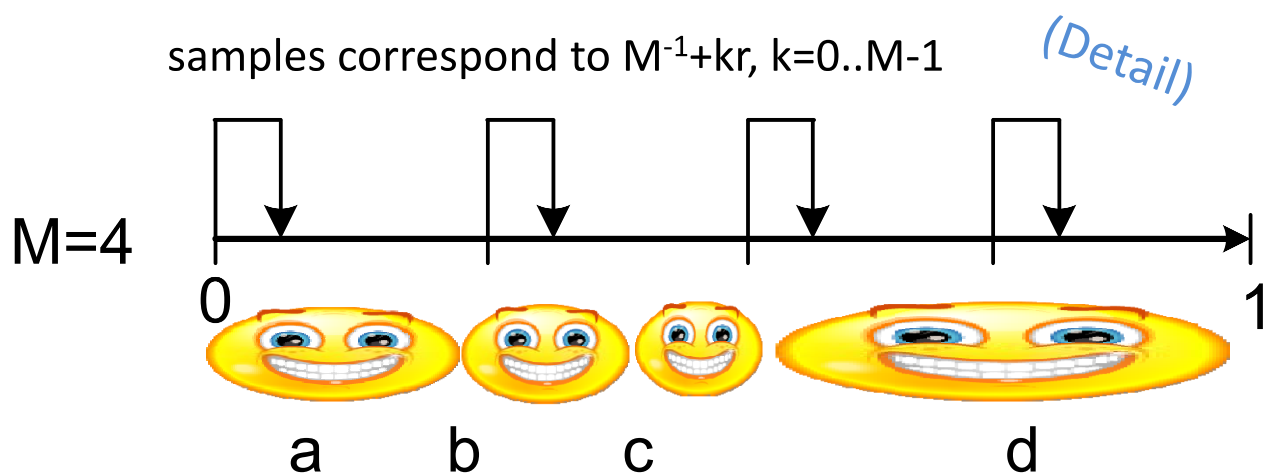
(Lao & Cohen, KDD 2010)

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



Low-Variance Sampling

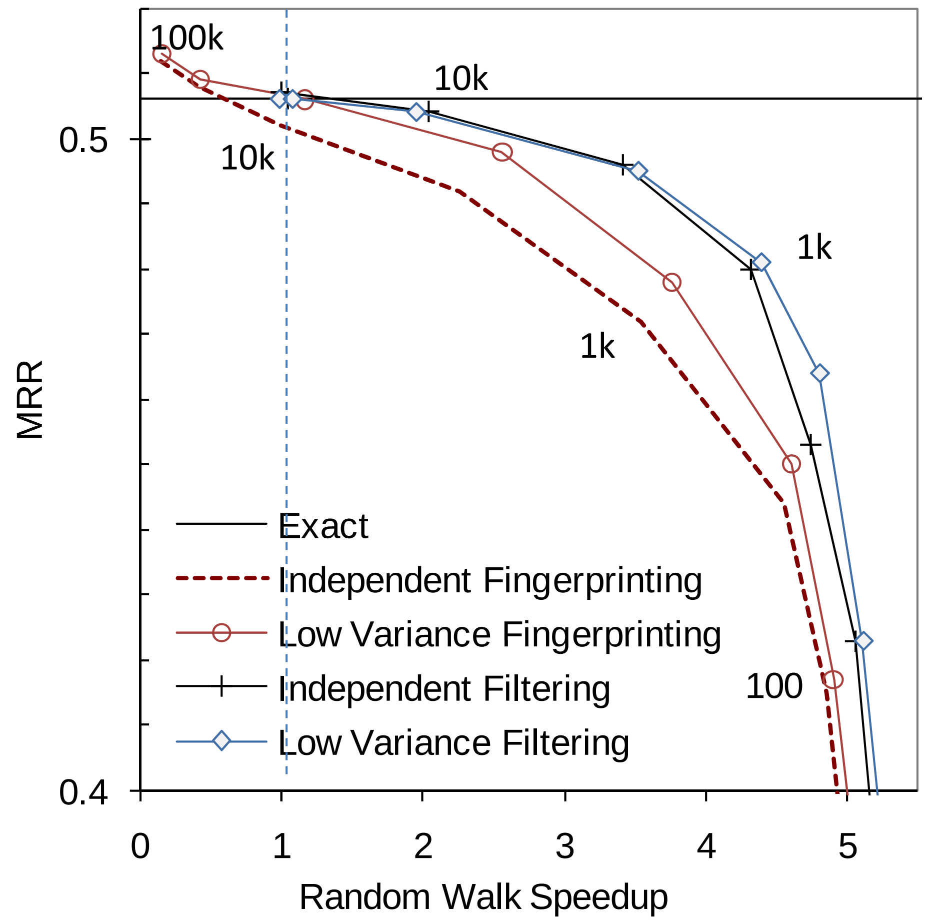
- Sampling walkers/particles independently introduces variances to the result distributions
- Low-Variance Sampling (LVS)(Thrun et al., 2005) generates M correlated samples, by drawing a single number r from $(0,M^{-1})$



Low Variance Sampling

- In our evaluation
  - LVS can slightly improve prediction for both finger printing and particle filtering

Averaged over 96 tasks



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.

	l=2		l=3	
	MRR	**Training**	**MRR**	**Training**
RWR(no train)	0.271		0.456	
RWR	0.280†	3.7s	0.471†	9.2s
PRA	0.307†	5.7s	0.516†	15.4s

RWR: Random Walk with Restart (personalized page rank)

†Paired t-test give p-values $7\text{x}10^{-3}$, $9\text{x}10^{-4}$, $9\text{x}10^{-8}$, $4\text{x}10^{-4}$

Example Paths

athletePlaysSport

$$c \xrightarrow{\text{isa}} c \xrightarrow{\text{isa}^{-1}} c \xrightarrow{\text{athletePlaysSport}} c$$

$$c \xrightarrow{\text{athletePlaysInLeague}} c \xrightarrow{\text{superpartOfOrganization}} c \xrightarrow{\text{teamPlaysSport}} c$$

teamHomeStadium

$$c \xrightarrow{\text{teamPlaysInCity}} c \xrightarrow{\text{cityStadiums}} c$$

$$c \xrightarrow{\text{teamMember}} c \xrightarrow{\text{athletePlaysForTeam}} c \xrightarrow{\text{teamHomeStadium}} c$$

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

		N-FOIL			PRA	
Task	**#Rules**	**p@10**	**p@100**	**#Paths**	**p@10**	**p@100**
Functional Predicates	2.1(+37)	**0.76**	**0.380**	43	**0.79**	**0.668**
Non-functional Predicates	----	----	----	92	**0.65**	**0.620**

PRA: Path Ranking Algorithm

Conclusion

- Random walk inference
  - Generate path features for link prediction tasks
  - Use sampling schemes for efficient inference
  - User 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?