ProbKB Large-Scale Probabilistic Knowledge Base

Yang Chen
yang@cise.ufl.edu

Computer and Information Science and Engineering
University of Florida

Aug 29, 2012

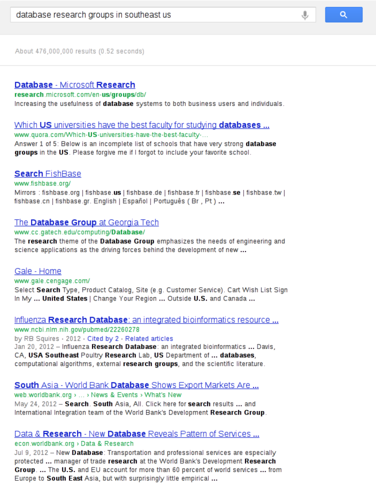
Data Science Research

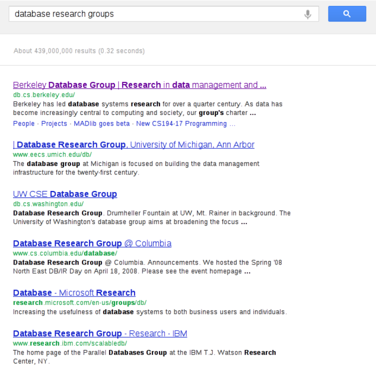
Outline

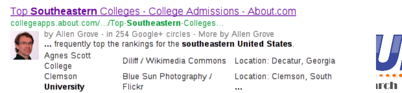
Knowledge bases–Introduction

- A *knowledge base* [3] is a special kind of database for knowledge management. A knowledge base provides a means for information to be collected, organized, shared, searched and utilized.
- A knowledge base helps machines understand humans, languages, and the world.

Knowledge bases–Introduction





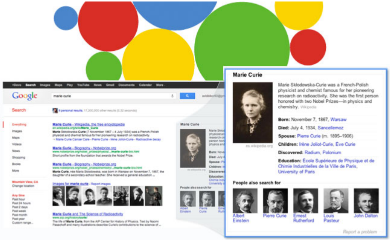


Outline

DSR @ UF
Data Science Research

Examples

Google Knowledge Graph [2]



Demo

DSR @ UF
Data Science Research

Examples

NELL

NELL [1] is a research project that attempts to create a computer system that learns over time to read the web.

Recently-Learned Facts twitter Refresh

instance	iteration	date learned	confidence
albert_iii is a monarch	623	13-aug-2012	99.9
kaseberg_creek is a river	620	01-aug-2012	99.1
senior_e_commerce_intelligence_consultant is a profession	622	09-aug-2012	98.4
deonna_anderson is a journalist	620	01-aug-2012	97.3
rose_city_comic_con is a convention	620	01-aug-2012	98.6
cancidas is a drug worked on by merck	622	09-aug-2012	93.8
softball is a sport with fans in the country new_zealand___wellington	623	13-aug-2012	100.0
mammals is an animal that eats berries	622	09-aug-2012	100.0
wccu is a TV affiliate of the network fox	625	21-aug-2012	100.0
cats is an animal that can develop respiratory_infection	622	09-aug-2012	100.0

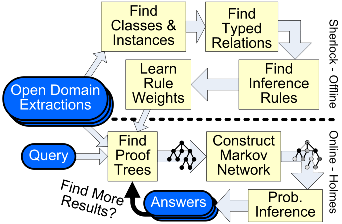
Note that NELL produces uncertain results. This is typical among automatic extraction systems and is our major motivation to develop a large-scale probabilistic knowledge base.

Examples

Sherlock-Holmes

The Sherlock-Holmes [7] is an open information extraction system consisting of two components:

- Sherlock [9], which learns inference rules offline, and
- Holmes [8], which uses inference rules to answer queries online.

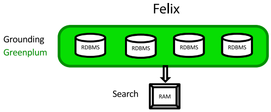


Examples

Tuffy-Felix

The Tuffy-Felix [5, 4] system is an Markov logic network [6] implementation that does large-scale probabilistic inference using an RDBMS.

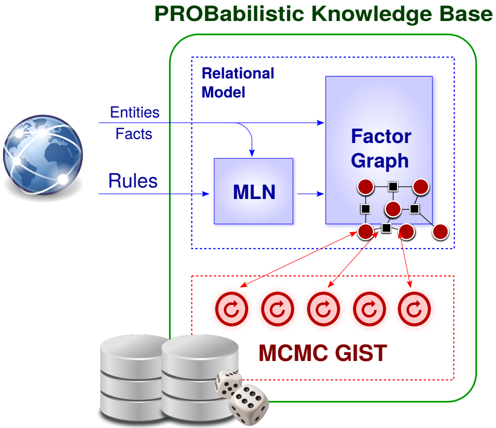
- A bottom-up approach to grounding using an RDBMS.
- A hybrid in-database grounding and in-memory inference architecture.
- Novel partitioning, loading, and parallel algorithms.
- Task decomposition to achieve web-scale.



Outline

Architecture

- Extracted entities, facts, and rules stored as relational model.
- Efficient grounding via a few relational operators.
- Parallel MCMC inference implemented as GIST operations.
- Incremental inference: save computation by focusing on least convergent variables.



Outline

Data Science Research

Markov Logic–Probabilistic Inference Framework

A *Markov logic network* (MLN) [6] is a set of formulae with weights. Together with a finite set of constants $C = \{c_1, \ldots, c_{|C|}\}$, it defines a Markov network.

Weight	First-Order Logic
0.7	Fr(x,y)$\wedge$Fr(y,z)$\rightarrow$Fr(x,z)
1.5	Sm(x)$\rightarrow$Ca(x)
1.1	Fr(x,y)$\wedge$Sm(x)$\rightarrow$Sm(y)

Table: Example Markov logic network.

A set of constants (entities, or objects)

$$C = \{A, B\}.$$

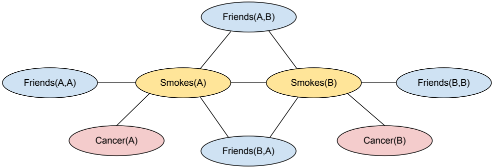


Figure: Grounded Markov network.

Markov Logic Inference

Grounding

Grounding is the process of substituting constants into MLN clauses.

The result of grounding is a *factor graph* (or *Markov network*) from which we can infer marginal probabilities for individual facts.

Challenges

- Memory-inefficient.
- Large graphical models are hard to do inference.

DSR @ UF
Data Science Research

Markov Logic Inference

Solutions to Large-Scale Grounding

- Using an RDBMS: leveraging mature query optimization techniques and possibly MPP frameworks (e.g. Greenplum).
- Lazy inference [10]: only ground *active clauses*.
- Ontology (typing): reducing the number of possible groundings.
- First-order Horn clauses: easy to learn and do inference.

Results

We can ground the whole Sherlock-Holmes dataset the first two rounds in 10 minutes using PostgreSQL, while the state-of-the-art implementation MLN (Tuffy [5]) crashes during its grouding phase.

#relations	10,672
#rules	31,000
#constants	1.1M
#evidence	250,000
#queries	10,672
Tuffy	**Crash**
ProbKB	10 min[1]

Table: Dataset statistics and performance.

[1] First two rounds.

Markov Logic Inference

Inference–Computing Probabilities

The Markov random field defines a probability distribution on all nodes in it:

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(\mathbf{x})\right),$$

where n_i is the number of ground clauses that satisfy clause i in the MLN and w_i is the weight of that clause. Z is the normalization constant, also called the *partition function*.

Markov Logic Inference

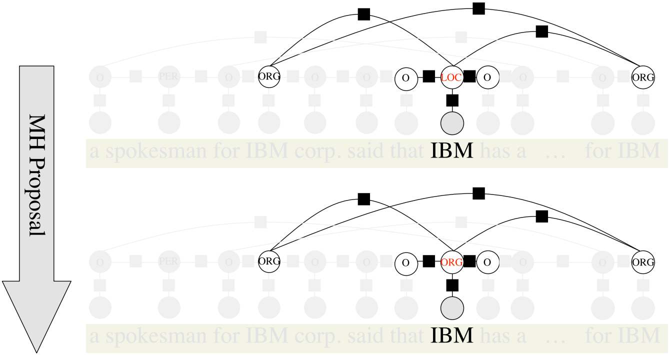
Metropolis-Hastings (MH)

- Exact inference: intractable due to Z.
- MCMC-MH: efficient since Z cancels out.
- MCMC-MH: only changed factors need to be considered [11]:

$$\frac{\pi(\mathbf{x})}{\pi(\mathbf{x}')} = \frac{\frac{1}{Z}\prod_i \phi_i(\mathbf{x}_i)}{\frac{1}{Z}\prod_i \phi_i(\mathbf{x}'_i)} = \frac{\prod_{i \text{ having } x_{(k)}} \phi_i(x_{(k)}, \mathbf{x})}{\prod_{i \text{ having } x_{(k)}} \phi_i(x'_{(k)}, \mathbf{x})}.$$

DSR @ UF
Data Science Research

MCMC-MH Efficiency



Outline

GraphLab

Data-Parallel	Graph-Parallel
Map Reduce	**GraphLab** (Carnegie Mellon)

Map Reduce

- Feature Extraction
- Cross Validation
- Computing Sufficient Statistics

GraphLab

Graphical Models
Gibbs Sampling
Belief Propagation
Variational Opt.

Semi-Supervised Learning
Label Propagation
CoEM

Collaborative Filtering
Tensor Factorization

Data-Mining
PageRank
Triangle Counting

GIST on Datapath

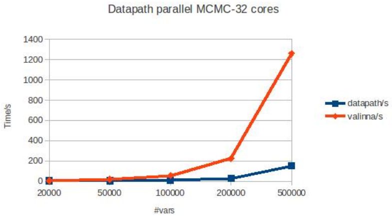
GraphLab Preliminary Results

#samples	10	100	200	500	State-of-the-art
IE	0.2s	2s	4s	10.2s	25.216s
ER	90.8s	181.5s	373s	>600s	225s
RC	5.2s	52.7s	111.3s	297.8s	**Crashed**
SHERLOCK-600	1.2s	12.6s	28.3s	65.1s	55min

Table: GraphLab-based parallel inference vs the state-of-the-art.

Datapath Preliminary Results

Figure: Inference over simulated factor graphs



Incremental MCMC

Goal

- Evolve over time.
- Learn from past samples.
- Integrate new evidence.

Incremental MCMC

The incremental maintenance algorithm adopts the Query-aware MCMC [12] technique:

- Assumption: not much new information is added to the knowledge base each time.
- Newly extracted knowledge serves as the query node.
- Maintaining samples for both old and new nodes.

Incremental Maintenance of MCMC Samples

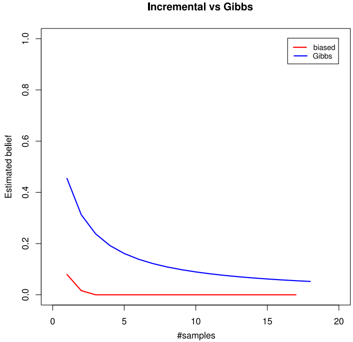
The incremental MCMC proposal function T employs the following steps:

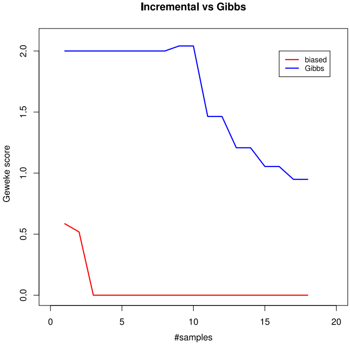
1. Sample the variable index space according to some distribution p reflecting influence of new nodes and recent sample behaviors.
2. Sample the selected variable according to some distribution q over that variable's domain, leaving all other variables unchanged.

We adjust the distribution p so that it focuses on newly added variables.

DSR @ UF
Data Science Research

MCMC Maintenance Results





Note: This algorithm is still under development.

Questions?

Thank you!

Outline

DSR @ UF
Data Science Research

References I

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr, and T.M. Mitchell.
Toward an architecture for never-ending language learning.
In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1306–1313, 2010.

Google.
Introducing the knowledge graph: things, not strings, 2012.

A. Nath and P. Domingos.
Efficient belief propagation for utility maximization and repeated inference.
In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

DSR @ UF
Data Science Research

References II

F. Niu, C. Zhang, C. Ré, and J. Shavlik.
Felix: Scaling Inference for Markov Logic with an Operator-based Approach.
ArXiv e-prints, August 2011.

Feng Niu, Christopher Ré, AnHai Doan, and Jude W. Shavlik.
Tuffy: Scaling up statistical inference in markov logic networks using an rdbms.
PVLDB, 4(6):373–384, 2011.

M. Richardson and P. Domingos.
Markov logic networks.
Machine learning, 62(1):107–136, 2006.

References III

S. Schoenmackers.
Inference Over the Web.
PhD thesis, University of Washington, 2011.

S. Schoenmackers, O. Etzioni, and D.S. Weld.
Scaling textual inference to the web.
In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–88. Association for Computational Linguistics, 2008.

S. Schoenmackers, O. Etzioni, D.S. Weld, and J. Davis.
Learning first-order horn clauses from web text.
In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098. Association for Computational Linguistics, 2010.

References IV

P. Singla and P. Domingos.
Memory-efficient inference in relational domains.
In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 21, page 488. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

M. Wick, A. McCallum, and G. Miklau.
Scalable probabilistic databases with factor graphs and mcmc.
Proceedings of the VLDB Endowment, 3(1-2):794–804, 2010.

Michael L Wick and Andrew McCallum.
Queryaware mcmc.
In *proceedings of the 25th Conference on Neural Information Processing Systems (NIPS)*, pages 2564–2572, 2011.