ProbKB Web-Scale Probabilistic Knowledge Base

Yang Chen
yang@cise.ufl.edu

Computer and Information Science and Engineering
University of Florida

Feb 22, 2012

Outline

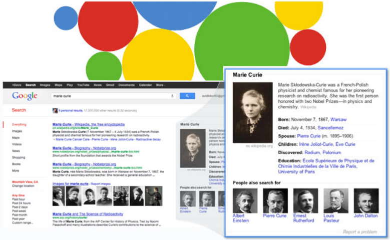
Knowledge bases–Introduction

- A *knowledge base* [ND10] 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.

Outline

Examples

Google Knowledge Graph [Goo12]



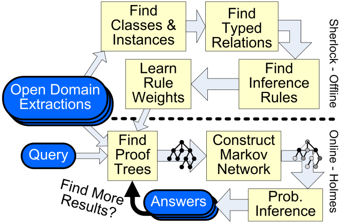
Demo

Related Work

SHERLOCK-HOLMES

The SHERLOCK-HOLMES [Sch11] is an open information extraction system consisting of two components:

- SHERLOCK [SEWD10], which learns inference rules offline, and
- HOLMES [SEW08], which uses inference rules to answer queries online.

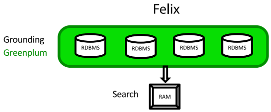


Related Work

TUFFY-FELIX

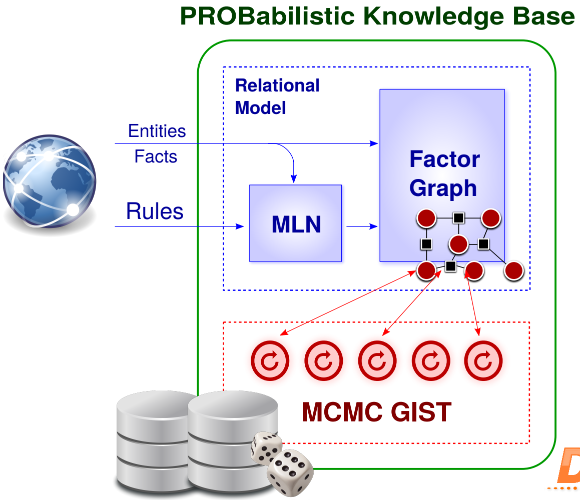
The TUFFY-FELIX [NRDS11, NZRS11] system is an Markov logic network [RD06] implementation that does large-scale probabilistic inference using an RDBMS.

- 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



Architecture

Contributions

- A relation model for extracted entities, facts, and rules.
- Efficient grounding via at most a few relational operators.
- Parallel MCMC inference implemented as `GIST` operations.
- Incremental inference: saving computation by focusing on least convergent variables.

Outline

Markov Logic–Probabilistic Inference Framework

A *Markov logic network* (MLN) [RD06] 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)

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

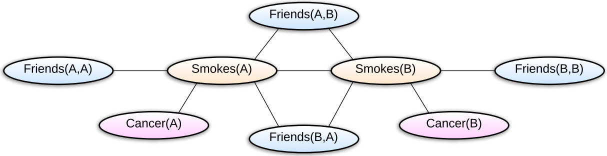


Figure: Grounded Markov network.

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.

Key Challenges

- Time-consuming, especially if the numbers of rules and entites are large.
- Grounded network has an intractably large size, making inference tasks slow.

Grounding

Scaling to the Web

- First-order Horn clauses:
  - Avoids the need to enumerate ground atoms.
  - Stored as *first-class* citizen in RDBMS, grounding expressed as a few `Join` s.
  - Easier to learn than general first-order clauses.
- Leveraging RDBMS query optimization techniques and possibly MPP frameworks (e.g. Greenplum).
- Ontology (typing): reducing the number of possible groundings and improving accuracy.

Grounding

A single JOIN operation handles all rules of type

$$p(x : c_1, y : c_2) \leftarrow q(x : c_1, z : c_3), r(z : c_3, y : c_2)$$

```
SELECT DISTINCT mln.head AS head, mln.body1 AS body1,
    mln.body2 AS body2, r1.ent1 AS ent1, r1.ent2 AS
    ent2, r2.ent2 AS ent3
FROM relations r1, mln, relations r2, relations r3,
    instances i1, instances i2, instances i3
WHERE r1.pred = mln.head AND r2.pred = mln.body1 AND
    r3.pred = mln.body2
AND r1.ent1 = r2.ent1 AND r1.ent2 = r3.ent2 AND r2.ent2
    = r3.ent1
AND i1.ent = r1.ent1 AND i1.class = mln.class1
AND i2.ent = r1.ent2 AND i2.class = mln.class2
AND i3.ent = r2.ent2 AND i3.class = mln.class3
```

Grounding Results

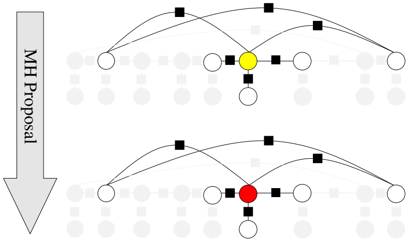
We can ground the whole Sherlock-Holmes dataset the first two rounds in 10 minutes using PostgreSQL, while the state-of-the-art implementation MLN (Tuffy [NRDS11]) crashes during its grouding phase.

#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.

Inference: MCMC-MH



Markov locality property allows for parallel computing.

Outline

GraphLab

Data-Parallel | Graph-Parallel

Map Reduce

Feature Extraction

Cross Validation

Computing Sufficient Statistics

Graphical Models
Gibbs Sampling
Belief Propagation
Variational Opt.

Semi-Supervised Learning
Label Propagation
CoEM

Collaborative Filtering
Tensor Factorization

Data-Mining
PageRank
Triangle Counting

GraphLab Execution Model

Algorithm 1 GraphLab Execution Model

Input: Data graph $G = (V, E, D)$
Input: Initial vertex set $\mathcal{T} = \{v_1, v_2, \ldots\}$
while $\mathcal{T}$ is not empty **do**
  $v \leftarrow$`RemoveNext`$(\mathcal{T})$
  $(\mathcal{T}', \mathcal{S}_v) \leftarrow f(v, \mathcal{S}_v)$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}'$
Output: Modified data graph $G = (V, E, D')$

- Vertexes schedule execution of their neighbors.
- Not applicable to general MCMC algorithms.

Datapath GIST

Generalized Iterable State Transforms (GIST)

- GIST Performs *transitions* upon a *state* until that state has converged to the desired result.
  - **Transition** MCMC Proposal function.
  - **State** Factor graph with its samples.
- A user-defined local scheduler allows general MCMC proposal implementation.
- The GIST state keeps track of the inference result.

DSR @ UF
Data Science Research

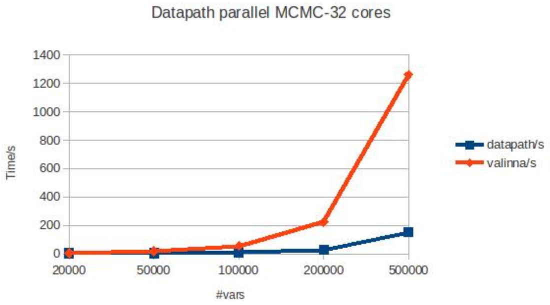
GraphLab Preliminary Results

#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



Outline

Incremental MCMC (Future Work)

A natural challenge arising in automatic knowledge-base construction is continuously incoming information.

- Expanding frontier belief propagation (EFBP) [ND10] for repeated inference;
- Query-aware MCMC [WM11] for query-spedific inference.

Following this line, we're trying to build an MCMC algorithm that:

- Focuses computation on most recently added nodes.
- Maintains previous samples to avoid repeated computation.

Incremental MCMC (Future Work)

Challenges

- How to detect variables that are mostly affected by new ones.
- The biased nature of incremental MCMC poses inbalance to the Datapath scheduler.

Conclusions

- PROBKB is a web-scale **PROB**abilistic **K**nowledge **B**ase with Markov logic network as the primary data model.
- All extractions, including entities and rules, are stored in RDBMS as relational model, allowing efficient grounding algorithms.
- In-database GIST operator allows parallel MCMC inference.
- Incremental MCMC focuses computation on most recently added variables, speeding up convergence.

Questions?

Thank you!

Outline

References I

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.

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.

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.

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

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

References III

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

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