Probability, Rules and Learning

Luc De Raedt
RuleML 2014

partly based on a joint tutorial with Angelika Kimmig
Purpose of this talk

Explore the use of probability in the context of rules, in the context of RuleML

My interpretation of RuleML, it is about

rules

reasoning about objects and relationships

applications in semantic web

emerging interest in uncertainty (eg. Vojtas & Bobek)
(Wild) conjecture?

- A lot of what you (seem to) do may be reformulated in a probabilistic logic / context
- The deterministic setting is obtained as a special case
- Still many challenges ahead ...
A key question in AI:

Dealing with uncertainty
- probability theory
- graphical models
- ...

Reasoning with relational data
- logic
- databases
- programming
- ...

Learning
- parameters
- structure

Statistical relational learning, probabilistic logic learning, probabilistic programming, ...
Some SRL formalisms

- LPAD: Bruynooghe, Vennekens, Verbaeten
- Markov Logic: Domingos, Richardson
- CLP(BN): Cussens, Page, Qazi, Santos Costa
- Present
- PRMs: Friedman, Getoor, Koller, Pfeffer, Segal, Taskar
- SLPs: Cussens, Muggleton
- Prob. CLP: Eisele, Riezler
- Prob. Horn Abduction: Poole
- First KBMC approaches: Breese, Bacchus, Charniak, Glesner, Goldman, Koller, Poole, Wellmann
- 1BC(2): Flach, Lachiche
- BLs: Kersting, De Raedt
- LOHMMs: De Raedt, Kersting, Raiko
- Future
- Prob. CLP: Eisele, Riezler
- Prob. horn Abduction: Poole
- First KBMC approaches: Breese, Bacchus, Charniak, Glesner, Goldman, Koller, Poole, Wellmann
- 1BC(2): Flach, Lachiche
- BLs: Kersting, De Raedt
- LOHMMs: De Raedt, Kersting, Raiko
- Future
Overview

- Part I: Basic probabilistic Prolog framework & relation to alternative frameworks -- the rules and probabilities
- Part II: Inference (short) -- the reasoning
- Part III: Probabilistic rule learning (ProbFOIL) -- the learning
- Part IV: Dynamics & Continuous distributions for Relational Tracking (in Robotics)

Focus on ProbLog line of research at KU Leuven
PART 1: Intro to Probabilistic Prologs
Networks of Uncertain Information

- **Gene**
  - Participates in biological process
  - Codes for protein
  - Is homologous to homolog group
  - Belongs to cellular component
  - Located in locus

- **Pathway**
  - Participates in biological process

- **Locus**
  - Is related to phenotype
  - Is located in gene

- **Cellular Component**
  - Is found in protein

- **Molecular Function**
  - Has protein

- **Protein**
  - Subsumes, interacts with gene

Biomine database @ Helsinki
http://biomine.cs.helsinki.fi/
Biomine network
Biomine Network

presenilin 2
Gene
EntrezGene: 81751

Notch receptor processing
Biological Process
GO:GO:0007220
-participates_in
0.220

integral to nuclear inner
Cellular Component
GO:GO:005638
Phenetic

Causes: Mutations
All related to similar phenotype
Effects: Differentially expressed genes
27 000 cause effect pairs

Interaction network:
3063 nodes
Genes
Proteins
16794 edges
Molecular interactions
Uncertain

Goal: connect causes to effects through common subnetwork
= Find mechanism
Techniques:
DTProbLog
Approximate inference

Can we find the mechanism connecting causes to effects?

[De Maeyer et al., Molecular Biosystems 13]
Example:

Information Extraction

<table>
<thead>
<tr>
<th>instance</th>
<th>iteration</th>
<th>date learned</th>
<th>confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>kelly_andrews is a female</td>
<td>826</td>
<td>29-mar-2014</td>
<td>98.7</td>
</tr>
<tr>
<td>investment_next_year is an economic sector</td>
<td>829</td>
<td>10-apr-2014</td>
<td>95.3</td>
</tr>
<tr>
<td>shibenik is a geopolitical entity that is an organization</td>
<td>829</td>
<td>10-apr-2014</td>
<td>97.2</td>
</tr>
<tr>
<td>quality_web_design_work is a character trait</td>
<td>826</td>
<td>29-mar-2014</td>
<td>91.0</td>
</tr>
<tr>
<td>mercedes_benz_cls_by_carlsson is an automobile manufacturer</td>
<td>829</td>
<td>10-apr-2014</td>
<td>95.2</td>
</tr>
<tr>
<td>social_work is an academic program at the university rutgers_university</td>
<td>827</td>
<td>02-apr-2014</td>
<td>93.8</td>
</tr>
<tr>
<td>dante wrote the book the_divine_comedy</td>
<td>826</td>
<td>29-mar-2014</td>
<td>93.8</td>
</tr>
<tr>
<td>willie_aames was born in the city los_angeles</td>
<td>831</td>
<td>16-apr-2014</td>
<td>100.0</td>
</tr>
<tr>
<td>kitt_peak is a mountain in the state or province arizona</td>
<td>831</td>
<td>16-apr-2014</td>
<td>96.9</td>
</tr>
<tr>
<td>greenwich is a park in the city london</td>
<td>831</td>
<td>16-apr-2014</td>
<td>100.0</td>
</tr>
</tbody>
</table>

instances for many different relations

degree of certainty

NELL: http://rtw.ml.cmu.edu/rtw/
Graphs & Randomness

ProbLog, Phenetic, Prism, ICL, Probabilistic Databases, ...

- all based on a “random graph” model

Stochastic Logic Programs, ProPPR, PCFGs, ...

- based on a “random walk” model
- connected to PageRank

- not the subject of this talk!
ProbLog by example:

A bit of gambling

- toss (biased) coin & draw ball from each urn
- win if (heads and a red ball) or (two balls of same color)

Probabilistic fact: heads is true with probability 0.4 (and false with 0.6)

0.4 :: heads.

Annotated disjunction: first ball is red with probability 0.3 and blue with 0.7

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.

Annotated disjunction: second ball is red with probability 0.2, green with 0.3, and blue with 0.5

0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) <- true.

Logical rule encoding consequences

win :- heads, col(_,red).
win :- col(1,C), col(2,C).
Questions

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).

- Probability of \texttt{win}'
- Probability of \texttt{win} given \texttt{col(2,green)}?
- Most probable world where \texttt{win} is true?

\textit{MPE inference}
Possible Worlds

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).

0.4 × 0.3 × 0.3
Possible Worlds

0.4 :: heads.

0.3 :: col(1, red); 0.7 :: col(1, blue) <- true.
0.2 :: col(2, red); 0.3 :: col(2, green); 0.5 :: col(2, blue) <- true.

win :- heads, col(_, red).
win :- col(1, C), col(2, C).
All Possible Worlds

<table>
<thead>
<tr>
<th>Probability</th>
<th>World 1</th>
<th>World 2</th>
<th>World 3</th>
<th>World 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.024</td>
<td>HRR</td>
<td>RRR</td>
<td>HBR</td>
<td>BR</td>
</tr>
<tr>
<td>0.036</td>
<td>HRG</td>
<td>RRG</td>
<td>HBG</td>
<td>BG</td>
</tr>
<tr>
<td>0.060</td>
<td>HRB</td>
<td>RB</td>
<td>HBO</td>
<td>BB</td>
</tr>
<tr>
<td>0.090</td>
<td>RRB</td>
<td>BB</td>
<td>BB</td>
<td>BB</td>
</tr>
<tr>
<td>0.056</td>
<td></td>
<td>W</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>0.084</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.140</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.210</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Most likely world where \(\text{col(2, blue)}\) is false?

- 0.024
  - H R R W
- 0.036
  - H R R W
- 0.056
  - H B R W
- 0.084
  - B R

- 0.036
  - H R G W
- 0.054
  - R G
- 0.084
  - H B G
- 0.126
  - B G

- 0.060
  - H R B W
- 0.090
  - R B
- 0.140
  - H B B W
- 0.210
  - B B W
\[ P(\text{win}) = \sum = 0.562 \]
\[ P(\text{win}|\text{col}(2,\text{green})) = \frac{\sum}{\sum} \]

\[ = P(\text{win} \land \text{col}(2,\text{green})) / P(\text{col}(2,\text{green})) \]

Conditional Probability

\[
\begin{array}{cccc}
0.024 & 0.036 & 0.056 & 0.084 \\
H & R & R & H & R & R & R & B & R \\
W & W & W & W & W & W & W & W & W \\
\hline
0.036 & 0.054 & 0.084 & 0.126 \\
H & R & G & H & B & G \\
W & W & W & W & W & W \\
\hline
0.060 & 0.090 & 0.140 & 0.210 \\
H & R & B & H & B \\
W & W & W & W & W \\
\end{array}
\]
Distribution Semantics
(with probabilistic facts)

\[ P(Q) = \sum_{F \cup R \models Q} \prod_{f \in F} p(f) \prod_{f \not\in F} (1 - p(f)) \]

query

sum over possible worlds
where \( Q \) is true

subset of probabilistic facts

Prolog rules

probability of possible world

[Sato, ICLP 95]
weight(skis, 6).
weight(boots, 4).
weight(helmet, 3).
weight(gloves, 2).

P::pack(Item) :-
    weight(Item, Weight),
    P is 1.0/Weight.

excess(Limit) :- ...

not excess(10).
pack(helmet) v pack(boots).

constraints as FOL formulas treat as evidence
Alternative view: CP-Logic

\[
\begin{align*}
\text{throws}(john). & \quad 0.5::\text{throws}(mary). \\
0.8::\text{break} & \leftarrow \text{throws}(mary). \\
0.6::\text{break} & \leftarrow \text{throws}(john).
\end{align*}
\]

\[
P(\text{break}) = 0.6 \times 0.5 \times 0.8 + 0.6 \times 0.5 \times 0.2 + 0.6 \times 0.5 + 0.4 \times 0.5 \times 0.8
\]
CP-logic [Vennekens et al.]

E.g., “throwing a rock at a glass **breaks** it with probability **0.3** and **misses** it with probability **0.7**”

\[(\text{Broken}(G):0.3) \lor (\text{Miss } 0.7) \leftarrow \text{ThrowAt}(G)\].

Note that the actual non-deterministic event (“rock flying at glass”) is implicit.

Slides CP-logic courtesy Joost Vennekens
Semantics

\[ I \models \text{ThrowAt}(G) \]

\[ (\text{Broken}(G) \ 0.3) \lor (\text{Miss} 0.7) \]

\[ \Leftarrow \text{ThrowAt}(G) \]

Probability tree is an execution model of theory iff:
- Each tree-transition matches causal law
- The tree cannot be extended

Each execution model defines the same probability distribution over final states

Slides CP-logic courtesy Joost Vennekens
Distributional Clauses (DC)

- Discrete- and continuous-valued random variables

**random variable** with **Gaussian distribution**

\[
\text{length}(\text{Obj}) \sim \text{gaussian}(6.0, 0.45) :- \text{type}(\text{Obj}, \text{glass}).
\]

**stackable**(O\text{Bot}, O\text{Top}) :-

\[
\begin{align*}
\approx \text{length}(\text{OBot}) & \geq \approx \text{length}(\text{OTop}), \\
\approx \text{width}(\text{OBot}) & \geq \approx \text{width}(\text{OTop}).
\end{align*}
\]

**comparing** values of **random variables**

\[
\text{ontype}(\text{Obj}, \text{plate}) \sim \text{finite}([0 : \text{glass}, 0.0024 : \text{cup}, \\
0 : \text{pitcher}, 0.8676 : \text{plate}, \\
0.0284 : \text{bowl}, 0 : \text{serving}, \\
0.1016 : \text{none}])
\]

\[\quad :- \text{obj}(\text{Obj}), \text{on}(\text{Obj}, \text{O2}), \text{type}(\text{O2}, \text{plate}).\]

**random variable** with **discrete distribution**

\[\text{Gutmann et al, TPLP 11; Nitti et al, IROS 13}\]
Probabilistic Logic Programming

Distribution Semantics [Sato, ICLP 95]: probabilistic choices + logic program → distribution over possible worlds

OVERVIEW paper [Kimmig, De Raedt, Arxiv]

e.g., PRISM, ICL, ProbLog, LPADs, CP-logic, ...

- multi-valued switches
- probabilistic alternatives
- probabilistic facts
- annotated disjunctions
- causal-probabilistic laws
Probabilistic databases

programming versus database query language
different types of queries

<table>
<thead>
<tr>
<th>Company</th>
<th>Product</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>sony</td>
<td>walkman</td>
<td>0.96</td>
</tr>
<tr>
<td>microsoft</td>
<td>mac_os_x</td>
<td>0.96</td>
</tr>
<tr>
<td>ibm</td>
<td>personal_computer</td>
<td>0.96</td>
</tr>
<tr>
<td>microsoft</td>
<td>mac_os</td>
<td>0.9</td>
</tr>
<tr>
<td>adobe</td>
<td>adobe_indesign</td>
<td>0.9</td>
</tr>
<tr>
<td>adobe</td>
<td>adobe_dreamweaver</td>
<td>0.87</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Company</th>
<th>City</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>microsoft</td>
<td>redmond</td>
<td>1.00</td>
</tr>
<tr>
<td>ibm</td>
<td>san_jose</td>
<td>0.99</td>
</tr>
<tr>
<td>emirates_airlines</td>
<td>dubai</td>
<td>0.93</td>
</tr>
<tr>
<td>honda</td>
<td>torrance</td>
<td>0.93</td>
</tr>
<tr>
<td>horizon</td>
<td>seattle</td>
<td>0.93</td>
</tr>
<tr>
<td>egyptair</td>
<td>cairo</td>
<td>0.93</td>
</tr>
<tr>
<td>adobe</td>
<td>san_jose</td>
<td>0.93</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

```
select x.Product, x.Company
from ProducesProduct x, HeadquarteredIn y
where x.Company=y.Company and
y.City='san_jose'
```

[Example from Suciu et al 2011]
Probabilistic Programs

• Distributional clauses / PLP similar in spirit
  • to e.g. BLOG, ... but embedded in existing logic and programming language
  • to e.g. Church but use of logic instead of functional programming ...

• natural possible world semantics and link with prob. databases.

• somewhat harder to do meta-programming
Suppose we have two constants: Anna (A) and Bob (B)

1.5 \( \forall x \ Smokes(x) \Rightarrow Cancer(x) \)
1.1 \( \forall x, y \ Friends(x, y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y)) \)
Markov Logic

Key differences

- programming language
- soft constraints
- Pro(b)log uses least-fix point semantics
  - can express transitive closure of relation
  - this cannot be expressed in FOL (and Markov Logic), requires second order logic
- p(X,Y) :- p(X,Z), p(Z,Y).
Take away message

Key insight

• Taisuke Sato, Distribution Semantics and David Poole

Upgrading logic / rules

• unify basic notions in logic and in probability theory
• ground atoms become random variables
• retain the rules and the logic
PART II: Inference
Inference in PLP

- As in Prolog and logic programming
  - proof-based
- As in Answer Set Programming
  - model-based
- As in Probabilistic Programming
  - sampling
Inference

Given:
- program
- queries
- evidence

Find:
- logical reasoning
- data structure
- probabilistic inference

1. using proofs
2. using models

knowledge compilation

Marginal probabilities
Conditional probabilities
MPE state
Proofs in ProbLog

\[
\begin{align*}
\text{influences}(\text{bob}, \text{carl}) & \quad \& \quad \text{influences}(\text{ann}, \text{bob}) \quad \& \quad \text{stress}(\text{ann}) \\
0.8 & \cdot 0.6 & \cdot 0.8 = 0.096 \\
0.4 & \cdot 0.2 = 0.08 \\
\text{proofs overlap!} \\
\text{cannot sum probabilities} \\
\text{(disjoint-sum-problem)}
\end{align*}
\]
Disjoint-Sum-Problem

possible worlds

<table>
<thead>
<tr>
<th>possible world</th>
<th>probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>infl(bob, carl) &amp; infl(ann, bob) &amp; st(ann) &amp; !st(bob)</td>
<td>0.0576</td>
</tr>
<tr>
<td>infl(bob, carl) &amp; infl(ann, bob) &amp; st(ann) &amp; st(bob)</td>
<td>0.0384</td>
</tr>
<tr>
<td>infl(bob, carl) &amp; !infl(ann, bob) &amp; st(ann) &amp; st(bob)</td>
<td>0.0256</td>
</tr>
<tr>
<td>infl(bob, carl) &amp; infl(ann, bob) &amp; !st(ann) &amp; st(bob)</td>
<td>0.0096</td>
</tr>
<tr>
<td>infl(bob, carl) &amp; !infl(ann, bob) &amp; !st(ann) &amp; st(bob)</td>
<td>0.0064</td>
</tr>
<tr>
<td>influences(bob, carl) &amp; stress(bob)</td>
<td></td>
</tr>
</tbody>
</table>

\[ \sum = 0.1376 \]

sum of proof probabilities: 0.096 + 0.08 = 0.1760

solution: knowledge compilation
Binary Decision Diagrams

\[ \text{influences}(\text{bob}, \text{carl}) \land \text{influences}(\text{ann}, \text{bob}) \land \text{stress}(\text{ann}) \land \neg \text{stress}(\text{bob}) \]
Binary Decision Diagrams

\[ \text{smokes}(c) = \text{i}(b, c) \land \text{s}(b) \lor \text{i}(b, c) \land \text{i}(a, b) \land \text{s}(a) \]

\[
\begin{align*}
0.8 \times 0.0 + 0.2 \times 0.688 &= 0.1376 \\
0.6 \times 0.48 + 0.4 \times 1.0 &= 0.688 \\
0.4 \times 0.0 + 0.6 \times 0.8 &= 0.48 \\
0.2 \times 0.0 + 0.8 \times 1.0 &= 0.8 \\
\end{align*}
\]

\[ \text{probability of } \text{smokes}(c) ? \]

\[ \text{influences}(\text{bob, carl})? \]
\[ \text{stress}(\text{bob})? \]
\[ \text{influences}(\text{ann, bob})? \]
\[ \text{stress}(\text{ann})? \]

\[ \text{smokes}(c) = \text{i}(b, c) \land \text{s}(b) \lor \text{i}(b, c) \land \text{i}(a, b) \land \text{s}(a) \]
Initial Approach
(ProbLog1 & others)

Find all proofs of query

Binary Decision Diagram (BDD)

Calculate marginal by dynamic programming

heads(1)
heads(2) & heads(3)

\[ \text{win} \]

0.4::heads(1).
0.7::heads(2).
0.5::heads(3).
win :- heads(1).
win :- heads(2),heads(3).

0.7

\[ \text{heads(2)} \]

0.3

\[ \text{heads(1)} \]

0.5

\[ \text{heads(3)} \]

\[ \text{false} \]

\[ \text{true} \]

\[ \text{h(1)} \]

\[ \text{h(2)} \]

\[ \text{h(3)} \]

\[ \text{P(win)} = \text{probability of reaching 1-leaf} \]

41  [De Raedt et al, IJCAI 07; Kimmig et al, TPLP 11]
Answering Questions

**Given:**
- program
- queries
- evidence

**Find:**
- marginal probabilities
- conditional probabilities
- MPE state

1. using proofs
2. using models

**Diagram:**
- logical reasoning
- data structure
- probabilistic inference
Current Approach
(ProbLog2)

Find relevant ground program for queries & evidence

Weighted CNF

use weighted model counting / satisfiability

0.4::heads(1).
0.7::heads(2).
0.5::heads(3).
win :- heads(1).
win :- heads(2), heads(3).

win :- heads(1).
win :- heads(2), heads(3).

win ↔ h(1) ∨ (h(2) ∧ h(3))

may require loop-breaking

(¬win ∨ h(1) ∨ h(2))
∧ (¬win ∨ h(1) ∨ h(3))
∧ (win ∨ ¬h(1))
∧ (win ∨ ¬h(2) ∨ ¬h(3))

h(1) → 0.4  h(2) → 0.7  h(3) → 0.5
¬h(1) → 0.6  ¬h(2) → 0.3  ¬h(3) → 0.5

use standard tool

[Fierens et al, TPLP 14]
ProbLog → CNF

?- smokes(carl).

smokes(X) :- stress(X).
smokes(X) :-
    influences(Y,X),
    smokes(Y).

• Find relevant ground rules by backward reasoning

smokes(carl) :- influences(bob,carl),smokes(bob).
smokes(bob) :- stress(bob).
smokes(bob) :- influences(ann,bob),smokes(ann).
smokes(ann) :- stress(ann).

• Convert to propositional logic formula

\[ sm(c) \leftrightarrow (i(b,c) \land sm(b)) \]
\[ \land sm(b) \leftrightarrow (st(b) \lor (i(a,b) \land sm(a))) \]
\[ \land sm(a) \leftrightarrow st(a) \]

may require loop-breaking

• Rewrite in CNF (as usual)
Weighted Model Counting

A propositional formula in conjunctive normal form (CNF) is given by a ProbLog program & query:

\[ P(Q) = \sum_{F \cup R \models Q} \prod_{f \in F} p(f) \prod_{f \notin F} (1 - p(f)) \]

The Weighted Model Counting (WMC) is:

\[ WMC(\phi) = \sum_{I_V \models \phi} \prod_{l \in I_V} w(l) \]

Interpretations (truth value assignments) of propositional variables:

- For literal \( p \cdot f \), the weight is:
  \[ w(f) = p \]
  \[ w(\neg f) = 1 - p \]
Introduction.

Probabilistic logic programs are logic programs in which some of the facts are annotated with probabilities.

ProbLog is a tool that allows you to intuitively build programs that do not only encode complex interactions between a large sets of heterogenous components but also the inherent uncertainties that are present in real-life situations.

The engine tackles several tasks such as computing the marginals given evidence and learning from (partial) interpretations. ProbLog is a suite of efficient algorithms for various inference tasks. It is based on a conversion of the program and the queries and evidence to a weighted Boolean formula. This allows us to reduce the inference tasks to well-studied tasks such as weighted model counting, which can be solved using state-of-the-art methods known from the graphical model and knowledge compilation literature.

The Language. Probabilistic Logic Programming.

ProbLog makes it easy to express complex, probabilistic models.

\[
\begin{align*}
0.3::\text{stress}(X) & : \text{person}(X), \\
0.2::\text{influences}(X,Y) & : \text{person}(X), \text{person}(Y).
\end{align*}
\]
Take-away message

• Inference is hard (#P-complete, WMC)
• Focus of a lot of research
• A lot of progress with usable implementations, but many challenges remain (like lifted inference)
Part III: Rule learning
Information Extraction in NELL

instances for many different relations

<table>
<thead>
<tr>
<th>instance</th>
<th>iteration</th>
<th>date learned</th>
<th>confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>kelly_andrews is a female</td>
<td>826</td>
<td>29-mar-2014</td>
<td>98.7</td>
</tr>
<tr>
<td>investment_next_year is an economic sector</td>
<td>829</td>
<td>10-apr-2014</td>
<td>95.3</td>
</tr>
<tr>
<td>shibenik is a geopolitical entity that is an organization</td>
<td>829</td>
<td>10-apr-2014</td>
<td>97.2</td>
</tr>
<tr>
<td>quality_web_design_work is a character trait</td>
<td>826</td>
<td>29-mar-2014</td>
<td>91.0</td>
</tr>
<tr>
<td>mercedes_benz_cls_by_carlsson is an automobile manufacturer</td>
<td>829</td>
<td>10-apr-2014</td>
<td>95.2</td>
</tr>
<tr>
<td>social_work is an academic program at the university rutgers_university</td>
<td>827</td>
<td>02-apr-2014</td>
<td>93.8</td>
</tr>
<tr>
<td>dante wrote the book the_divine_comedy</td>
<td>826</td>
<td>29-mar-2014</td>
<td>93.8</td>
</tr>
<tr>
<td>willie_aames was born in the city los_angeles</td>
<td>831</td>
<td>16-apr-2014</td>
<td>100.0</td>
</tr>
<tr>
<td>kitt_peak is a mountain in the state or province arizona</td>
<td>831</td>
<td>16-apr-2014</td>
<td>96.9</td>
</tr>
<tr>
<td>greenwich is a park in the city london</td>
<td>831</td>
<td>16-apr-2014</td>
<td>100.0</td>
</tr>
</tbody>
</table>

NELL: http://rtw.ml.cmu.edu/rtw/
Rule learning in NELL (I)

- Original approach
  - Make probabilistic data deterministic
  - run classic rule-learner (variant of FOIL)
  - re-introduce probabilities on learned rules and predict
Rule learning in NELL (2)

- Newer Page Rank Based Approach (Cohen et al. CIKM, Arxiv) -- ProPPR
  - Change the underlying model, from random graph / database to random walk one;
  - No longer “degree of belief” assigned to facts;
  - more like stochastic logic programs
  - Learn rules / parameters
Probabilistic Rule Learning

- Learn the rules directly in a PLP setting
- Generalize relational learning and inductive logic programming directly towards probabilistic setting
- Traditional rule learning/ILP as a special case
- Apply to probabilistic databases like NELL
Quinlan’s Playtennis

<table>
<thead>
<tr>
<th>ex</th>
<th>outlook</th>
<th>temperature</th>
<th>humidity</th>
<th>wind</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>t</td>
<td>t</td>
<td>f</td>
<td>f</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>t</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>-</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Our Windsurfing Example

<table>
<thead>
<tr>
<th>ex</th>
<th>pop</th>
<th>windok</th>
<th>sunshine</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7</td>
<td>0.5</td>
<td>0.7</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
<td>0.7</td>
<td>0.6</td>
<td>0.85</td>
</tr>
<tr>
<td>3</td>
<td>0.4</td>
<td>0.3</td>
<td>0.4</td>
<td>0.45</td>
</tr>
<tr>
<td>4</td>
<td>0.3</td>
<td>0.7</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*pop = Probability of Precipitation*
Differences

• Observations (features) are uncertain
• Class is uncertain as well

• This type of data occurs naturally in applications in
  • image / video analysis
  • text processing and the web
  • life sciences (e.g., Muggleton et al. MLJ 09)
  • probabilistic databases
Rule learning

In the logical setting

\[
\text{playtennis} :\!-\! \text{outlook}=\text{ok}, \text{wind}=\text{ok}
\]

\[
\text{playtennis} :\!-\! \text{outlook}=\text{ok}, \text{humidity}=\text{ok}
\]

In the probabilistic case

\[
\text{surfing} :\!-\! \neg \text{pop}, \text{wind}\text{ok}
\]

\[
\text{surfing} :\!-\! \neg \text{pop}, \text{sunshine}
\]

both a declarative and a probabilistic interpretation
Computing Probabilities

Consider the rules

surfing :- not pop, windok
surfing :- not pop, sunshine

The example 0.2::pop, 0.7::windok, 0.6::sunshine. Then

\[ P(\text{surfing}) = P( \text{not pop and windok}) \text{ or } \text{not pop and sunshine}) \]

= \[ P( \text{not pop and windok}) \text{ or } \text{not pop and sunshine and not windok}) \]

= 0.8 \times 0.7 + 0.8 \times 0.6 \times 0.3 \quad \text{disjoint sum problem}
In ProbLog (I)

Basic Setting

surfing(X) :- not pop(X), windok(X)

surfing(X) :- not pop(X) and sunshine(X)

0.2::pop(e1). 0.7::windok(e1). 0.6::sunshine(e1).

?-P(surfing(e1)).

gives 0.8 x 0.7 + 0.8 x 0.6 x 0.3 = P(B U H |= e)
In ProbLog (2)

Extended Setting

\begin{align*}
P_1 &:: \text{surfing}(X) :- \text{not pop}(X) \text{ and windok}(X). \\
P_2 &:: \text{surfing}(X) :- \text{not pop}(X) \text{ and sunshine}(X).
\end{align*}

\begin{align*}
0.2::\text{pop}(e1). &\quad 0.7::\text{windok}(e1). &\quad 0.6::\text{sunshine}(e1).
\end{align*}

?-P(\text{surfing}(e1)).

gives

\[0.8 \times 0.7 \times p_1 + 0.8 \times 0.6 \times 0.3 \times p_2 = P(B \cup H \models e)\]
Inductive Probabilistic Logic Programs

Given

a set of example facts \( e \in E \) together with the probability \( p \) that they hold

a background theory \( B \) in ProbLog

a hypothesis space \( L \) (a set of clauses)

Find

\[
\arg \min_H \text{loss}(H, B, E) = \arg \min_H \sum_{e_i \in E} \left| P_s(B \cup H \models e_i) - p_i \right|
\]
Observations

Propositional versus first order

- traditional rule learning = propositional
- inductive logic programming = first order

Deterministic case

- all probabilities 0 or 1
- traditional rule learning / ILP as special case
Analysis

1) the true positive part $tp_i = \min(p_i, p_{H,i})$,

2) the true negative part $tn_i = \min(n_i, n_{H,i})$,

3) the false positive part $fp_i = \max(0, n_i - tn_i)$, and

4) the false negative part $fn_i = \max(0, p_i - tp_i)$.

62

Inducing Probabilistic Logic Programs from Probabilistic Examples

Fig. 1: The true and false positive and negative part of a single example (left) and the probabilistic contingency table (right).

Fig. 2: The true and false positive and negative part of an entire dataset for the probabilistic case (left), and for the deterministic case (right).

The different notions are graphically displayed in Figure 2, in which the x-axis contains the examples and the y-axis their probability and all the examples are ordered according to increasing target probability. The areas then denote the respective rates. The deterministic case is illustrated in Figure 2 (right), which shows that in this case the examples take on 1/0 values. Figure 2 (left) illustrates this for the probabilistic case. From this picture, it may be clear that the notions of $TP$, $TN$, $FP$ and $FN$ correspond to the usual notions of true/false positive/negative rates from the literature in classification, yielding a probabilistic contingency table as shown in Figure 1 (right). Because the $TP$ and $FP$ rates form the basis for ROC space and PN-space, the traditional ROC analysis (as described in, for instance, [16]), used in rule learning can be applied to the probabilistic rule learning setting that we study in this paper and can be interpreted in a similar way as in traditional rule learning. Therefore, ROC analysis techniques, the analysis of heuristics and measures such as AUC essentially carry over to the probabilistic case.

4.2 Calculating...
Analysis

current hypothesis

target hypothesis

p(FN)

p(TP)

examples

p(TN)

current hypothesis

target hypothesis

p(FP)

examples

probability
Contingency Table

Fig. 1: The true and false positive and negative part of a single example (left) and the probabilistic contingency table (right).

Fig. 2: The true and false positive and negative part of an entire dataset for the probabilistic case (left), and for the deterministic case (right).

The different notions are graphically displayed in Figure 2, in which the x-axis contains the examples and the y-axis their probability and all the examples are ordered according to increasing target probability. The areas then denote the respective rates. The deterministic case is illustrated in the Figure 2 (right), which shows that in this case the examples take on 1/0 values. Figure 2 (left) illustrates this for the probabilistic case. From this picture, it may be clear that the notions of TP, TN, FP and FN correspond to the usual notions of true/false positive/negative rates from the literature in classification, yielding a probabilistic contingency table as shown in Figure 1 (right). Because the TP and FP rates form the basis for ROC space and PN-space, the traditional ROC analysis (as described in, for instance, [16]), used in rule learning can be applied to the probabilistic rule learning setting that we study in this paper and can be interpreted in a similar way as in traditional rule learning. Therefore, ROC analysis techniques, the analysis of heuristics and measures such as AUC essentially carry over to the probabilistic case.

4.2 Calculating...
Rule learning

Interesting properties

• adding a rule is monotonic, this can only increase the probability of an example

• adding a condition to a rule is anti-monotonic, this can only decrease the probability of an example

• several rules may be needed to cover an example
  • use all examples all of the time (do not delete them while learning), do not forget the positives
  • disjoint sum problem
ProbFOIL

Quinlan’s well-known FOIL algorithm combined with ProbLog and probabilistic examples and background knowledge

Essentially a vanilla sequential covering algorithm with m-estimate as local score and accuracy as global score.

(But other variations based on e.g. Fuernkranz tutorial are possible ... )
Criteria

\[
\text{precision} = \frac{TP}{TP + FP}
\]
\[
\text{m-estimate} = \frac{TP + m \cdot \frac{P}{N}}{TP + FP + m}
\]
\[
\text{recall} = \frac{TP}{TP + FN}
\]
\[
\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]

Avoiding overfitting using significance test
We now present our algorithm for learning probabilistic clauses, which is a generalization of the mFOIL rule learning algorithms. The outline of the algorithm is shown as Algorithm 1. It follows the typical separate-and-conquer approach (also known as sequential covering) that is commonly used in rule learning algorithms. The outer loop of the algorithm, labeled \texttt{ProbFOIL}, starts from an empty set of clauses and repeatedly adds clauses to the hypothesis until no more improvement is observed with respect to a global scoring function. The clauses obtained by the function \texttt{LearnRule}, which greedily searches for the clause that maximizes a local scoring function.

The resulting algorithm is very much like the standard rule-learning algorithm known from the literature (cf. [16,31]).

\begin{algorithm}
\caption{The \texttt{ProbFOIL}+ learning algorithm}
\begin{algorithmic}[1]
\Function {ProbFOIL\texttt{+}}{target} \Comment{\texttt{target} is the target predicate}
\State $H := \emptyset$
\While {true}
\State clause := \texttt{LearnRule}($H$, target) \Comment{Start with an empty (probabilistic) body}
\If {$\text{GlobalScore}(H) < \text{GlobalScore}(H \cup \{\text{clause}\})$}
\State $H := H \cup \{\text{clause}\}$
\Else \Comment{Grow rule}
\State return $H$
\EndIf
\EndWhile
\EndFunction
\Function {LearnRule}{$H$, target} \Comment{Generate all refinements}
\State candidates := \{\texttt{x :: target} $\leftarrow$ \texttt{true}\} \Comment{Reject unsuited refinements}
\State bestrule := (\texttt{x :: target} $\leftarrow$ \texttt{true}) \Comment{Update best rule}
\While {candidates $\neq \emptyset$} \Comment{Generate all refinements}
\State nextcandidates := \emptyset \Comment{Generate all refinements}
\ForAll {\texttt{x :: target} $\leftarrow$ \texttt{body} $\in$ candidates} \Comment{Generate all refinements}
\ForAll {\texttt{literal} $\in$ \texttt{p(target} $\leftarrow$ \texttt{body})} \Comment{Reject unsuited refinements}
\If {\texttt{rejectRefinement}(\texttt{H}, bestrule, \texttt{x :: target} $\leftarrow$ \texttt{body})} \Comment{Reject unsuited refinements}
\State nextcandidates := nextcandidates $\cup$ \{\texttt{x :: target} $\leftarrow$ \texttt{body} $\land$ \texttt{l}\} \Comment{Generate all refinements}
\EndIf
\EndFor
\EndFor
\EndWhile
\EndFunction
\end{algorithmic}
\end{algorithm}

As the global scoring function, which determines the stopping criterion of the outer loop, we use accuracy which is defined as

$$\text{accuracy}(H) = \frac{TP + TN}{M},$$

where $M$ is the size of the dataset.
Extended rule learning

Learn rules with probability $x::\text{head} : - \text{body}$

What changes?

- value of $x$ determines prob. of coverage of example

$x=1$  

$x=0$  

Diagram:

(a)  

(b)  

(c)
Extended rule learning

Express local score as a function of $x$

Compute optimal value of $x$
Implementation
Optimizations

Incremental grounding
Simplified CNF conversion to ProbLog
Sometimes direct calculation of probabilities
Even simpler when propositional data only
Some language bias (range-restricted)
Experiments on Bayesian Net
Results

Table 1: Mean absolute error on network A with CPTs $\sim \text{Beta}(\alpha, \beta)$, averaged over all target attributes. Observed nodes are independent.

<table>
<thead>
<tr>
<th>$\alpha/\beta$</th>
<th>1.0000</th>
<th>0.1000</th>
<th>0.0100</th>
<th>0.0010</th>
<th>0.0001</th>
<th>0.00001</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZeroR</td>
<td>0.054</td>
<td>0.11</td>
<td>0.11</td>
<td>0.13</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>LinearRegression</td>
<td>$7.7 \times 10^{-3}$</td>
<td>0.027</td>
<td>0.024</td>
<td>0.025</td>
<td>0.024</td>
<td>0.024</td>
</tr>
<tr>
<td>MultilayerPerceptron</td>
<td>$1.8 \times 10^{-3}$</td>
<td>$8.5 \times 10^{-3}$</td>
<td>$6.3 \times 10^{-3}$</td>
<td>$5.8 \times 10^{-3}$</td>
<td>$5.7 \times 10^{-3}$</td>
<td>$5.7 \times 10^{-3}$</td>
</tr>
<tr>
<td>M5P</td>
<td>$1.7 \times 10^{-3}$</td>
<td>$6.7 \times 10^{-3}$</td>
<td>$4.2 \times 10^{-3}$</td>
<td>$4.2 \times 10^{-3}$</td>
<td>$4.0 \times 10^{-3}$</td>
<td>$4.0 \times 10^{-3}$</td>
</tr>
<tr>
<td>M5P -R -M 4.0</td>
<td>0.013</td>
<td>0.031</td>
<td>0.026</td>
<td>0.029</td>
<td>0.028</td>
<td>0.028</td>
</tr>
<tr>
<td>SMOreg</td>
<td>$7.7 \times 10^{-3}$</td>
<td>0.027</td>
<td>0.024</td>
<td>0.026</td>
<td>0.025</td>
<td>0.025</td>
</tr>
<tr>
<td>ProbFOIL(1,15,0.0,rel)</td>
<td>0.069</td>
<td>0.051</td>
<td>$5.9 \times 10^{-4}$</td>
<td>$1.6 \times 10^{-7}$</td>
<td>$1.6 \times 10^{-7}$</td>
<td>$1.6 \times 10^{-7}$</td>
</tr>
<tr>
<td>ProbFOIL$^+$ (1,1,0.0,rel)</td>
<td>$1.8 \times 10^{-3}$</td>
<td>$3.0 \times 10^{-3}$</td>
<td>$10.0 \times 10^{-5}$</td>
<td>$1.6 \times 10^{-7}$</td>
<td>$1.6 \times 10^{-7}$</td>
<td>$1.6 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

Observed Nodes Independent

ProbFOIL better on deterministic data
Regression learners better than BASIC ProbFOIL
Extended ProbFOIL : best of both worlds
Results

<table>
<thead>
<tr>
<th>( \alpha/\beta )</th>
<th>1.0000</th>
<th>0.1000</th>
<th>0.0100</th>
<th>0.0010</th>
<th>0.0001</th>
<th>0.00001</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinearRegression</td>
<td>2.6 ( \times ) 10^{-3}</td>
<td>0.018</td>
<td>0.021</td>
<td>0.020</td>
<td>0.018</td>
<td>0.018</td>
</tr>
<tr>
<td>MultilayerPerceptron</td>
<td>4 ( \times ) 10^{-4}</td>
<td>3.1 ( \times ) 10^{-3}</td>
<td>5.7 ( \times ) 10^{-3}</td>
<td>3.9 ( \times ) 10^{-3}</td>
<td>3.3 ( \times ) 10^{-3}</td>
<td>3.3 ( \times ) 10^{-3}</td>
</tr>
<tr>
<td>M5P</td>
<td>7 ( \times ) 10^{-4}</td>
<td>4.9 ( \times ) 10^{-3}</td>
<td>6.5 ( \times ) 10^{-3}</td>
<td>5.4 ( \times ) 10^{-3}</td>
<td>4.4 ( \times ) 10^{-3}</td>
<td>4.4 ( \times ) 10^{-3}</td>
</tr>
<tr>
<td>M5P -R -M 4.0</td>
<td>5.2 ( \times ) 10^{-3}</td>
<td>0.021</td>
<td>0.023</td>
<td>0.025</td>
<td>0.024</td>
<td>0.024</td>
</tr>
<tr>
<td>SMOreg</td>
<td>2.6 ( \times ) 10^{-3}</td>
<td>0.017</td>
<td>0.021</td>
<td>0.020</td>
<td>0.017</td>
<td>0.017</td>
</tr>
<tr>
<td>ProbFOIL(1,10,0.0,rel)</td>
<td>0.015</td>
<td>0.012</td>
<td>1.9 ( \times ) 10^{-3}</td>
<td>9.4 ( \times ) 10^{-8}</td>
<td>4.2 ( \times ) 10^{-8}</td>
<td>4.2 ( \times ) 10^{-8}</td>
</tr>
<tr>
<td>ProbFOIL^+(1,5,0.0,rel)</td>
<td>3.9 ( \times ) 10^{-3}</td>
<td>3.9 ( \times ) 10^{-3}</td>
<td>5.3 ( \times ) 10^{-4}</td>
<td>2.8 ( \times ) 10^{-7}</td>
<td>4.2 ( \times ) 10^{-8}</td>
<td>4.2 ( \times ) 10^{-8}</td>
</tr>
</tbody>
</table>

Observed Nodes Dependent / Full Observability

ProbFOIL better on deterministic data
Regression learners better than BASIC ProbFOIL
Extended ProbFOIL: best of both worlds
Results

Table 4: Mean absolute error on network B with CPTs $\sim \text{Beta}(\alpha, \beta)$, averaged over all target attributes. Observed nodes are dependent. There is partial observability.

<table>
<thead>
<tr>
<th>$\alpha/\beta$</th>
<th>1.0000</th>
<th>0.1000</th>
<th>0.0100</th>
<th>0.0010</th>
<th>0.0001</th>
<th>0.00001</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZeroR</td>
<td>0.023</td>
<td>0.077</td>
<td>0.085</td>
<td>0.093</td>
<td>0.096</td>
<td>0.096</td>
</tr>
<tr>
<td>LinearRegression</td>
<td>$4.8 \times 10^{-3}$</td>
<td>0.019</td>
<td>0.026</td>
<td>0.032</td>
<td>0.034</td>
<td>0.034</td>
</tr>
<tr>
<td>MultilayerPerceptron</td>
<td>$1.2 \times 10^{-3}$</td>
<td>$2.9 \times 10^{-3}$</td>
<td>$9.7 \times 10^{-3}$</td>
<td>0.020</td>
<td>0.031</td>
<td>0.032</td>
</tr>
<tr>
<td>M5P</td>
<td>$1.6 \times 10^{-3}$</td>
<td>$6.1 \times 10^{-3}$</td>
<td>0.022</td>
<td>0.027</td>
<td>0.027</td>
<td>0.027</td>
</tr>
<tr>
<td>M5P -R -M 4.0</td>
<td>$6.2 \times 10^{-3}$</td>
<td>0.022</td>
<td>0.033</td>
<td>0.040</td>
<td>0.040</td>
<td>0.040</td>
</tr>
<tr>
<td>SMOreg</td>
<td>$4.5 \times 10^{-3}$</td>
<td>0.018</td>
<td>0.022</td>
<td>0.029</td>
<td>0.034</td>
<td>0.034</td>
</tr>
<tr>
<td>ProbFOIL$(1,15,0.0,\text{rel})$</td>
<td>0.020</td>
<td>0.023</td>
<td>0.012</td>
<td>0.015</td>
<td>0.015</td>
<td>0.015</td>
</tr>
<tr>
<td>ProbFOIL$^+$$(1,10,0.0,\text{rel})$</td>
<td>$9.5 \times 10^{-3}$</td>
<td>0.011</td>
<td>0.011</td>
<td>0.013</td>
<td>0.013</td>
<td>0.013</td>
</tr>
</tbody>
</table>

Observed Nodes Dependent / Partial Observability

ProbFOIL better on deterministic data
Regression learners sometimes better than ProbFOIL
## Information Extraction in NELL

### Recently-Learned Facts

<table>
<thead>
<tr>
<th>instance</th>
<th>iteration</th>
<th>date learned</th>
<th>confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>kelly_andrews is a female</td>
<td>826</td>
<td>29-mar-2014</td>
<td>98.7</td>
</tr>
<tr>
<td>investment_next_year is an economic sector</td>
<td>829</td>
<td>10-apr-2014</td>
<td>95.3</td>
</tr>
<tr>
<td>shibenik is a geopolitical entity that is an organization</td>
<td>829</td>
<td>10-apr-2014</td>
<td>97.2</td>
</tr>
<tr>
<td>quality_web_design_work is a character trait</td>
<td>826</td>
<td>29-mar-2014</td>
<td>91.0</td>
</tr>
<tr>
<td>mercedes_benz cls by carlsson is an automobile manufacturer</td>
<td>829</td>
<td>10-apr-2014</td>
<td>95.2</td>
</tr>
<tr>
<td>social_work is an academic program at the university rutgers_university</td>
<td>827</td>
<td>02-apr-2014</td>
<td>93.8</td>
</tr>
<tr>
<td>dante wrote the book the divine comedy</td>
<td>826</td>
<td>29-mar-2014</td>
<td>93.8</td>
</tr>
<tr>
<td>willie_aames was born in the city los_angeles</td>
<td>831</td>
<td>16-apr-2014</td>
<td>100.0</td>
</tr>
<tr>
<td>kitt_peak is a mountain in the state or province arizona</td>
<td>831</td>
<td>16-apr-2014</td>
<td>96.9</td>
</tr>
<tr>
<td>greenwich is a park in the city london</td>
<td>831</td>
<td>16-apr-2014</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Instances for many different relations

Degree of certainty
In order to test probabilistic rule learning for facts extracted by NELL, we used the NELL athlete dataset, which has already been used in the context of meta-interpretive learning of higher-order dyadic Datalog [36]. This dataset contains 10130 facts. The number of facts per predicate is listed in Table 5. The unary predicates in this dataset are deterministic, whereas the binary predicates have a probability attached.

Table 5: Number of facts per predicate (NELL athlete dataset)

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Facts</th>
</tr>
</thead>
<tbody>
<tr>
<td>athletecoach(person, person)</td>
<td>18</td>
</tr>
<tr>
<td>athleteplaysport(person, sport)</td>
<td>1921</td>
</tr>
<tr>
<td>athleteplaysinleague(person, league)</td>
<td>872</td>
</tr>
<tr>
<td>coachesinleague(person, league)</td>
<td>93</td>
</tr>
<tr>
<td>teamhomestadium(team, stadium)</td>
<td>198</td>
</tr>
<tr>
<td>athleteplayssportsteamposition(person, position)</td>
<td>255</td>
</tr>
<tr>
<td>athlete(person)</td>
<td>1909</td>
</tr>
<tr>
<td>coach(person)</td>
<td>624</td>
</tr>
<tr>
<td>male(person)</td>
<td>7</td>
</tr>
<tr>
<td>organization(league)</td>
<td>1</td>
</tr>
<tr>
<td>personafrika(person)</td>
<td>1</td>
</tr>
<tr>
<td>personaustralia(person)</td>
<td>22</td>
</tr>
<tr>
<td>personneurope(person)</td>
<td>1</td>
</tr>
<tr>
<td>personus(person)</td>
<td>6</td>
</tr>
<tr>
<td>sportsleague(league)</td>
<td>18</td>
</tr>
<tr>
<td>sportsteamposition(position)</td>
<td>22</td>
</tr>
<tr>
<td>athleteplaysforteam(person, team)</td>
<td></td>
</tr>
<tr>
<td>teamplaysinleague(team, league)</td>
<td></td>
</tr>
<tr>
<td>athletealsoknownas(person, name)</td>
<td>17</td>
</tr>
<tr>
<td>coachesteam(person, team)</td>
<td>132</td>
</tr>
<tr>
<td>teamplayssport(team, sport)</td>
<td>359</td>
</tr>
<tr>
<td>athletemhomestadium(person, stadium)</td>
<td>187</td>
</tr>
<tr>
<td>attraction(stadium)</td>
<td>2</td>
</tr>
<tr>
<td>female(person)</td>
<td>2</td>
</tr>
<tr>
<td>hobby(sport)</td>
<td>5</td>
</tr>
<tr>
<td>person(person)</td>
<td>2</td>
</tr>
<tr>
<td>personasia(person)</td>
<td>4</td>
</tr>
<tr>
<td>personcanada(person)</td>
<td>1</td>
</tr>
<tr>
<td>personmexico(person)</td>
<td>108</td>
</tr>
<tr>
<td>sport(sport)</td>
<td>36</td>
</tr>
<tr>
<td>sportsteam(team)</td>
<td>1330</td>
</tr>
<tr>
<td>stadiumoreventvenue(stadium)</td>
<td>171</td>
</tr>
</tbody>
</table>

Table 5 also shows the types that were used for the variables in the base declarations for the predicates. As indicated in Section 4.5, this typing of the variables forms a syntactic restriction on the possible groundings and ensures that arguments are only instantiated with variables of the appropriate type. Furthermore, the LearnRule function of the ProbFOIL algorithm is based on mFOIL and allows to incorporate a number of variable constraints. To reduce the search space, we imposed that unary predicates that are added to a candidate rule during the learning process can only use variables that have already been introduced. Binary predicates can introduce at most one new variable.

6.2 Relational probabilistic rule learning

In order to illustrate relational probabilistic rule learning with ProbFOIL in the context of NELL, we will learn rules and report their respective accuracy for each binary predicate with more than 500 facts. In order to show ProbFOIL’s speed, also the runtimes are reported. Unless indicated otherwise, both the m-estimate’s m value and the beam width were set to 1. The value of p for rule significance was set to 0.9. The rules are postprocessed such that only range-restricted rules are obtained. Furthermore, to avoid a bias towards the majority class, the examples are balanced, i.e., negative examples are added to balance the number of positives.
Fig. 5: Histogram of probabilities for each of the binary predicates with more than 500 facts: (a) athleteplaysforteam; (b) athleteplayssport; (c) teamplaysinleague; and, (d) athleteplaysinleague.
5.4.2 athleteplayssport(person, sport)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.98604: athleteplayssport(A, B) ← athleteplayssport(V_2, V_1), athleteplaysinleague(A, V_3), coachesinleague(V_2, V_3), teamplayssport(V_1, B), athleteplaysforteam(A, V_1).</td>
<td>0.79391: athleteplaysforteam(A, B)</td>
</tr>
<tr>
<td>0.907: athleteplayssport(A, B) ← athleteplaysforteam(V_2, V_1), teammate(V_2, A), teamplayssport(V_1, B).</td>
<td>0.91817: athleteplayssport(A, B) ← coachesteam(A, V_1), teamplayssport(V_1, B).</td>
</tr>
</tbody>
</table>

Listing 2: Learned rules for the athleteplayssport predicate.

5.4.3 teamplaysinleague(team, league)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.95848: teamplaysinleague(A, B) ← athleteplaysforteam(V_1, V_2), coachesinleague(V_1, B), teamplayssagainstteam(V_2, A).</td>
<td>1.0: teamplaysinleague(A, B) ← athleteledsportsteam(V_1, A), athleteplaysinleague(V_1, B).</td>
</tr>
<tr>
<td>0.92240: teamplaysinleague(A, B) ← athleteplaysinleague(V_1, B), coachesteam(V_1, V_2), teamplayssagainstteam(V_2, A), athleteplayssortsteam(V_1, V_2), athleteledsportsteam(V_1, V_2).</td>
<td>0.99998: teamplaysinleague(A, B) ← coachesinleague(V_1, B), coachwontrophy(V_1, V_2), teamwontrophy(A, V_2).</td>
</tr>
<tr>
<td>1.0: teamplaysinleague(A, B) ← athleteledsportsteam(V_1, A), athleteplaysinleague(V_1, B).</td>
<td>1.0: teamplaysinleague(A, B) ← athleteledsportsteam(V_1, B), coachesteam(V_1, V_2), teamplayssagainstteam(V_2, A).</td>
</tr>
</tbody>
</table>

Listing 3: Learned rules for the teamplaysinleague predicate.
5.4.5 teamplaysagainstteam(team,team)

0.9375::teamplaysagainstteam(A,B) ← teamwontrophy(A,V_1), teamwontrophy(B,V_1).
0.58662::teamplaysagainstteam(A,B) ← athleteplaysforteam(V_1,A),
coachesteam(V_1,V_2), teamplayssport(B,V_3), teamplayssport(V_2,V_3).

Listing 5: Learned rules for the athleteplaysinleague predicate.

5.4.1 athleteplaysforteam(person,team)

0.9375::athleteplaysforteam(A,B) ← athleteledsportsteam(A,B).
0.9675::athleteplaysforteam(A,B) ← athleteledsportsteam(A,V_1),
teamplaysagainstteam(B,V_1).
0.79391::athleteplaysforteam(A,B) ← athleteplaysinleague(A,V_1),
teamplaysinleague(B,V_1).

Listing 1: Learned rules for the athleteplaysforteam predicate.
Take away message

Rule learning applies / generalizes naturally to probabilistic data and databases.
Parameter Learning

e.g., webpage classification model

for each \texttt{CLASS1, CLASS2} and each \texttt{WORD}

\begin{align*}
\text{?? :: } & \text{link\_class(Source, Target, CLASS1, CLASS2).} \\
\text{?? :: } & \text{word\_class(WORD, CLASS).}
\end{align*}

\begin{align*}
\text{class(Page, C) } & \text{ :- } \text{has\_word(Page, W), word\_class(W, C).} \\
\text{class(Page, C) } & \text{ :- } \text{links\_to(OtherPage, Page),} \\
& \text{class(OtherPage, OtherClass),} \\
& \text{link\_class(OtherPage, Page, OtherClass, C).}
\end{align*}
Sampling Interpretations
Parameter Estimation

\[ p(\text{fact}) = \frac{\text{count(\text{fact is true})}}{\text{Number of interpretations}} \]
Learning from partial interpretations

• Not all facts observed
• Soft-EM
• use expected count instead of count
• $P(Q |E)$ -- conditional queries!

[Gutmann et al, ECML 11; Fierens et al, TPLP 14]
Bayesian Parameter Learning

- Learning as inference (e.g., Church)
- Prior distributions for parameters
- Given data, find most likely parameter values
Example

• Flipping a coin with unknown weight
• Prior: uniform distribution on [0, 1]
• Observation: 5x heads in a row
• Sampling from Church model:

![](Image)

**Coin weight, prior to observing data**

![Graph showing the prior distribution of coin weights.]

**Coin weight, conditioned on observed data**

![Graph showing the updated distribution of coin weights after observing 5 heads in a row.]

[from probmods.org]
ProbLog Example

query(weight(C,X)) :- coin(C),param(X).

! e evidence(data(c1,[h,h,h,h,h,h,h,h,h,h,h,h,h]),true).
evidence(data(c2,[h,t,h,h,h,h,t,t,h,t,t,h]),true).

prior

<table>
<thead>
<tr>
<th>Value</th>
<th>Prior</th>
<th>Posterior for c1</th>
<th>Posterior for c2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.3</td>
<td>0.225</td>
<td>0.225</td>
<td>0.225</td>
</tr>
<tr>
<td>0.5</td>
<td>0.450</td>
<td>0.450</td>
<td>0.450</td>
</tr>
<tr>
<td>0.7</td>
<td>0.675</td>
<td>0.675</td>
<td>0.675</td>
</tr>
<tr>
<td>0.9</td>
<td>0.900</td>
<td>0.900</td>
<td>0.900</td>
</tr>
</tbody>
</table>

ask for posterior data
Part IV: Dynamics
Dynamics: Evolving Networks

- **Travian**: A massively multiplayer real-time strategy game
  - Commercial game run by TravianGames GmbH
  - ~3,000,000 players spread over different “worlds”
  - ~25,000 players in one world

[Thon et al., MLJ 11, ECML 08]
World Dynamics

Fragment of world with

~10 alliances
~200 players
~600 cities

alliances color-coded

Can we build a model of this world?
Can we use it for playing better?

[Thon, Landwehr, De Raedt, ECML08]
World Dynamics

Fragment of world with

~10 alliances
~200 players
~600 cities

alliances color-coded

Can we build a model
of this world?
Can we use it for playing
better?

[Thon, Landwehr, De Raedt, ECML08]
World Dynamics

Fragment of world with

~10 alliances
~200 players
~600 cities

alliances color-coded

Can we build a model of this world?
Can we use it for playing better?

[Thon, Landwehr, De Raedt, ECML08]
Causal Probabilistic Time-Logic (CPT-L)

how does the world change over time?

one of the effects holds at time $T+1$

$$0.4::\text{conquest}(\text{Attacker},C); \ 0.6::\text{nil} \leftarrow \text{city}(C,\text{Owner}),\text{city}(C2,\text{Attacker}),\text{close}(C,C2).$$

if cause holds at time $T$

[Thon et al, MLJ 11]
Social Network of Chats

A Social Network Diagram for an IRC Channel

#travian

Diagram showing a network of users connected through various channels and chats, including:

- ATTenTion
- Orochimaru
- T.guest227
- sweet-lemon Secret
- Mika
- sweet-lewis
- General_Chao
- Litch
- Erica
- Carrothead
- General_Chao|class
- comx|Rockstar
- Travian20311
- Tink
Relational Tracking

- Track people or objects over time? Even if temporarily hidden?
- Recognize activities?
- Infer object properties?
Relational State Estimation over Time

**Magnetism scenario**
- object tracking
- category estimation from interactions

**Box scenario**
- object tracking even when invisible
- estimate spatial relations

[Nitti et al, IROS 13]
Speed 0x

Queries
(updated every 5 steps)

on(X,Y):
[1.0:(3,(table)),1.0:(4,(table))]
inside(X,Y):
[]
tr_inside(X,Y):
[]

Box ID=4  Cube ID=3

Particles
Magnetic scenario

- 3 object types: magnetic, ferromagnetic, nonmagnetic
- Nonmagnetic objects do not interact
- A magnet and a ferromagnetic object attract each other
- Magnetic force that depends on the distance
- If an object is held magnetic force is compensated.
Magnetic scenario

- 3 object types: magnetic, ferromagnetic, nonmagnetic

  \[ \text{type}(X)_t \sim \text{finite}([1/3:\text{magnet}, 1/3:\text{ferromagnetic}, 1/3:\text{nonmagnetic}]) \leftarrow \text{object}(X). \]

- 2 magnets attract or repulse

  \[ \text{interaction}(A, B)_t \sim \text{finite}([0.5:\text{attraction}, 0.5:\text{repulsion}]) \leftarrow \text{object}(A), \text{object}(B), A < B, \text{type}(A)_t = \text{magnet}, \text{type}(B)_t = \text{magnet}. \]

- Next position after attraction

  \[ \text{pos}(A)_{t+1} \sim \text{gaussian} (\text{middlepoint}(A, B)_t, \text{Cov}) \leftarrow \text{near}(A, B)_t, \text{not(held}(A)), \text{not(held}(B)), \text{interaction}(A, B)_t = \text{attr}, \]
  \[ c/\text{dist}(A, B)_t^2 > \text{friction}(A)_t. \]

  \[ \text{pos}(A)_{t+1} \sim \text{gaussian} (\text{pos}(A)_t, \text{Cov}) \leftarrow \text{not( attraction}(A, B) ). \]
Dynamic Distributional Clauses

Prior distribution $p(x_0)$

State transition model $p(x_t|x_{t-1},u_t)$

Measurement model $p(z_t|x_t)$

Other rules: $p(x'_t|x''_t)$
Ongoing Work

• Online parameter learning [Nitti, ICRA 2014]

• Integration with planning

• Larger Experiments

• Applications in robotics (also to learn affordances)
Take-away message

• Probabilistic rules also apply to dynamic environments
• Topic of ongoing work
• Scalability is a challenge
Learning relational affordances

Learn probabilistic model

From two object interactions
Generalize to N

Moldovan et al. ICRA 12, 13, 14
Occluded Object Search

- How to achieve a specific configuration of objects on the shelf?
- Where’s the orange mug?
- Where’s something to serve soup in?
- Models of objects and their spatial arrangement

[Moldovan et al. 14]
ProbLog for activity recognition from video

- Separation between low-level events (LLE) and high-level events (HLE)
  - LLE: walking, running, active, inactive, abrupt
  - HLE: meeting, moving, fighting, leaving_object
- Probabilistic Logic approach: Event Calculus in ProbLog (Prob-EC) to infer the high-level events from an algebra of low-level events.
- Example:

  \[
  \text{initiatedAt}(\text{fighting}(P_1,P_2) = \text{true}, T) \leftarrow \\
  \text{happensAt}(\text{abrupt}(P_1), T), \\
  \text{holdsAt}(\text{close}(P_1,P_2,44) = \text{true}, T), \\
  \text{not happensAt}(\text{inactive}(P_2), T).
  \]
Conclusions

• Probabilistic rules and logic
• Learning and inference
• Many challenges remain:
  • scalability and inference
  • users getting used to probabilities (datasets)
  • learning
• Argued that it applies to a lot of issues for RuleML?
Thanks!

http://dtai.cs.kuleuven.be/problog
• PRISM http://sato-www.cs.titech.ac.jp/prism/
• ProbLog2 http://dtai.cs.kuleuven.be/problog/
• Yap Prolog http://www.dcc.fc.up.pt/~vsc/Yap/ includes
  • ProbLog1
  • cplint https://sites.google.com/a/unife.it/ml/cplint
• CLP(BN)
• LP2
• PITA in XSB Prolog http://xsb.sourceforge.net/
• AILog2 http://artint.info/code/ailog/ailog2.html
• SLPs http://stoics.org.uk/~nicos/sware/pepl
• contdist http://www.cs.sunysb.edu/~cram/contdist/
• DC https://code.google.com/p/distributional-clauses
• WFOMC http://dtai.cs.kuleuven.be/ml/systems/wfomc
References


