Probability, Rules and Learning

Luc De Raedt

RuleML 2014

partly based on a joint tutorial with Angelika Kimmig

DTAI

MACHINE LEARNING

KU LEUVEN

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 ...

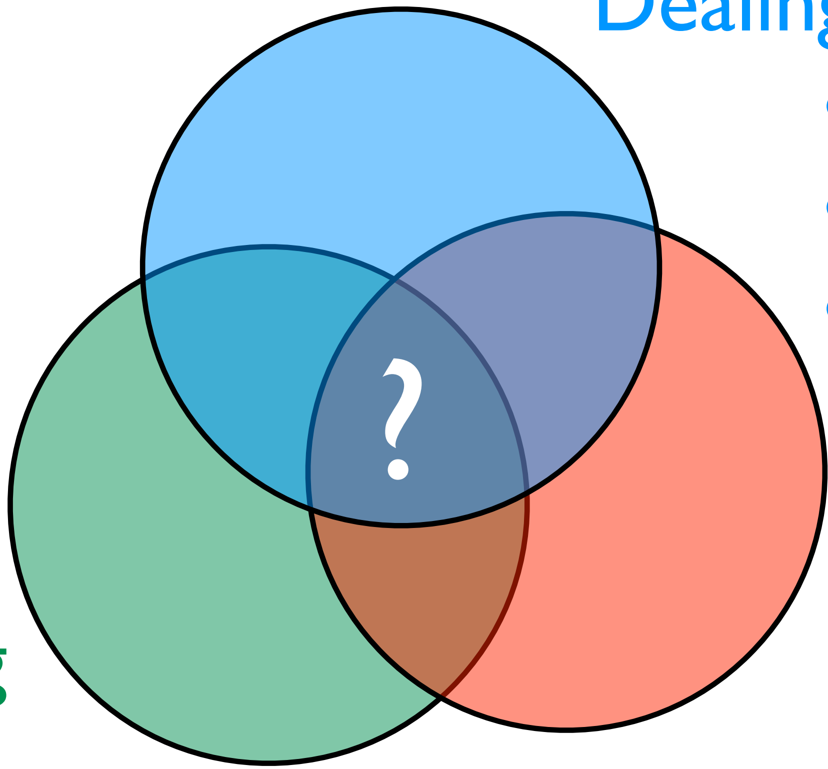
A key question in AI:

Dealing with uncertainty

- probability theory
- graphical models
- ...

Reasoning with relational data

- logic
- databases
- programming
- ...



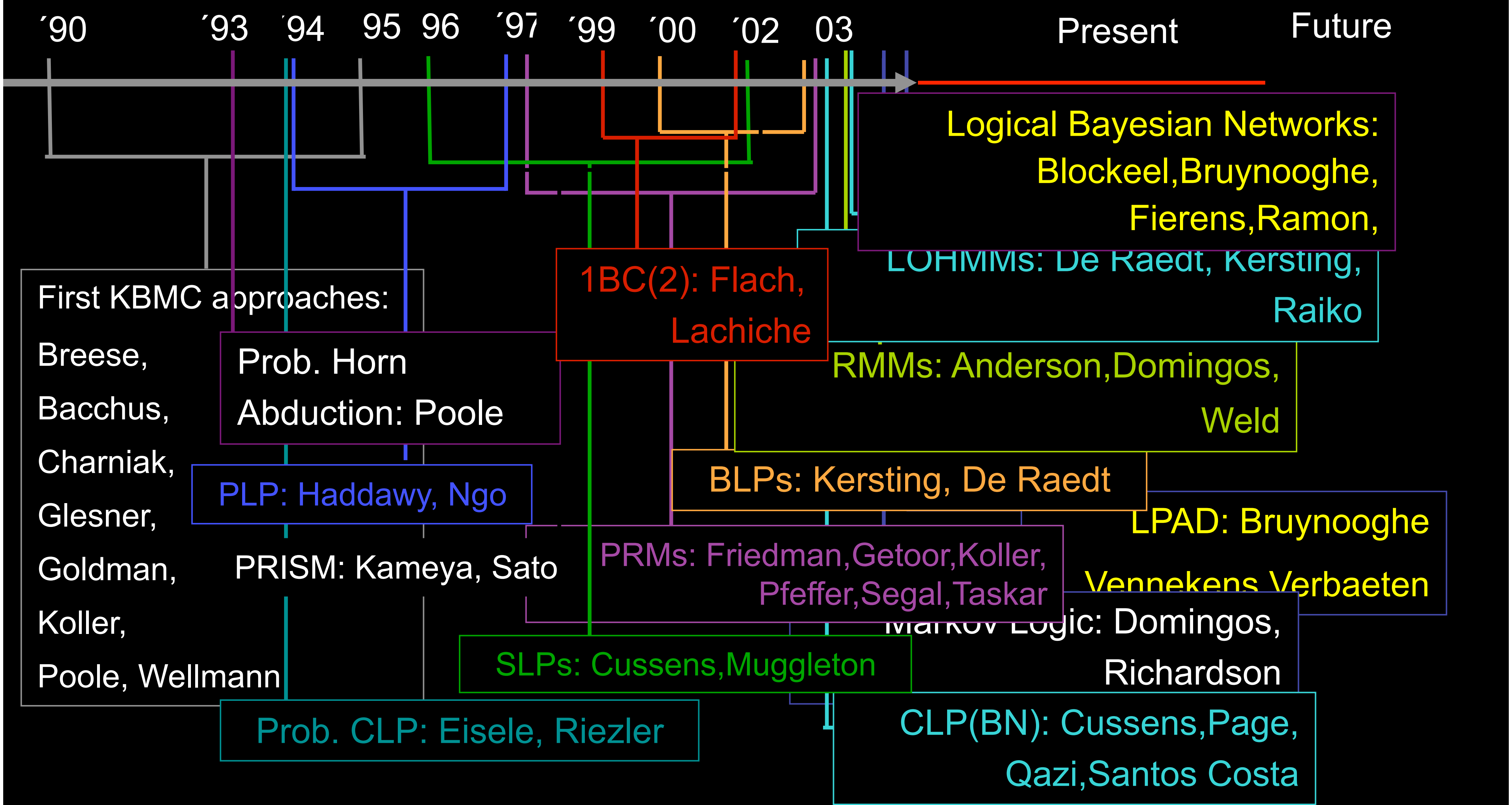
Learning

- parameters
- structure

Statistical relational learning, probabilistic logic learning, probabilistic programming, ...

Some SRL formalisms

[names in alphabetical order]



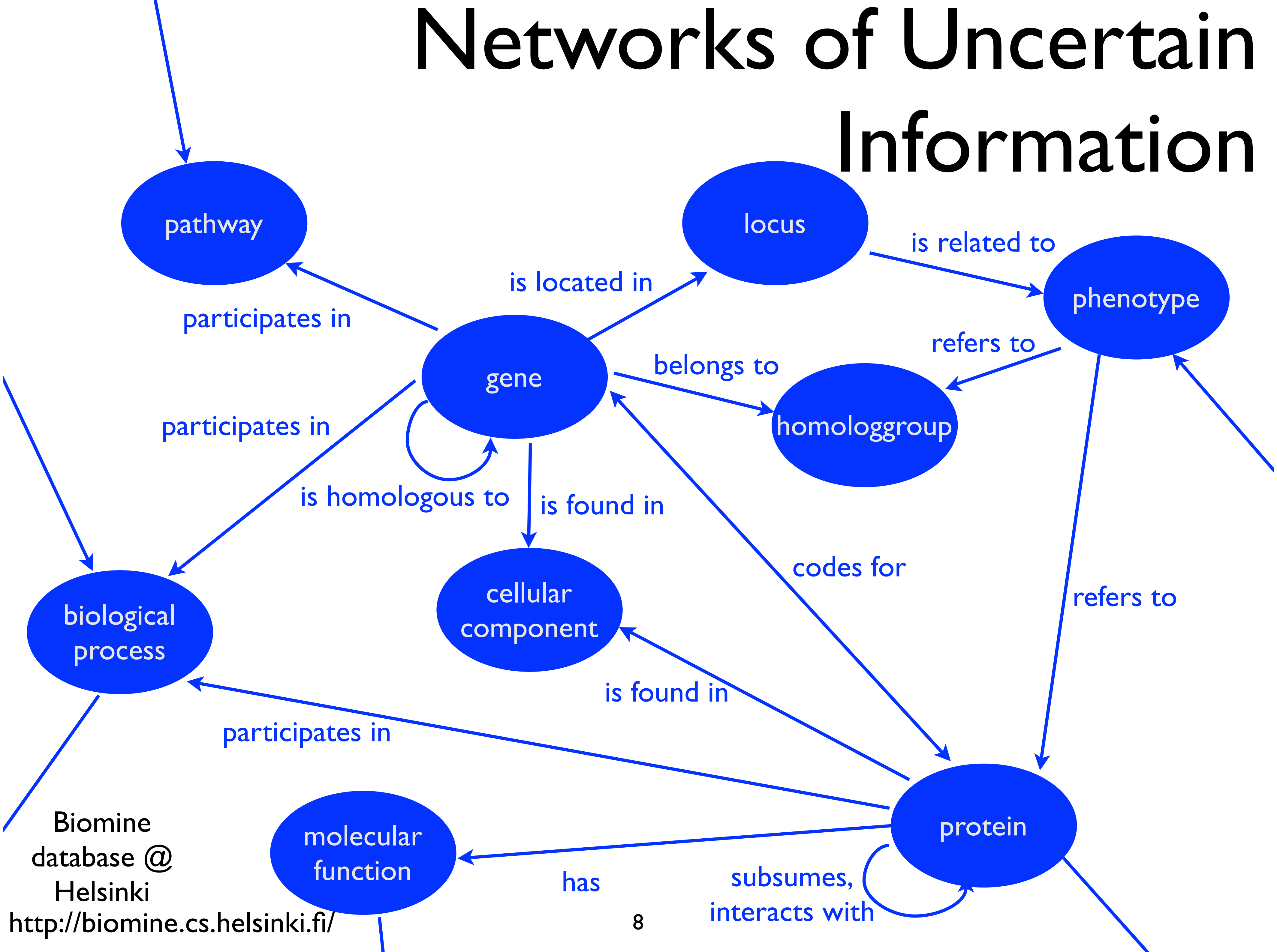
Overview

- 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)

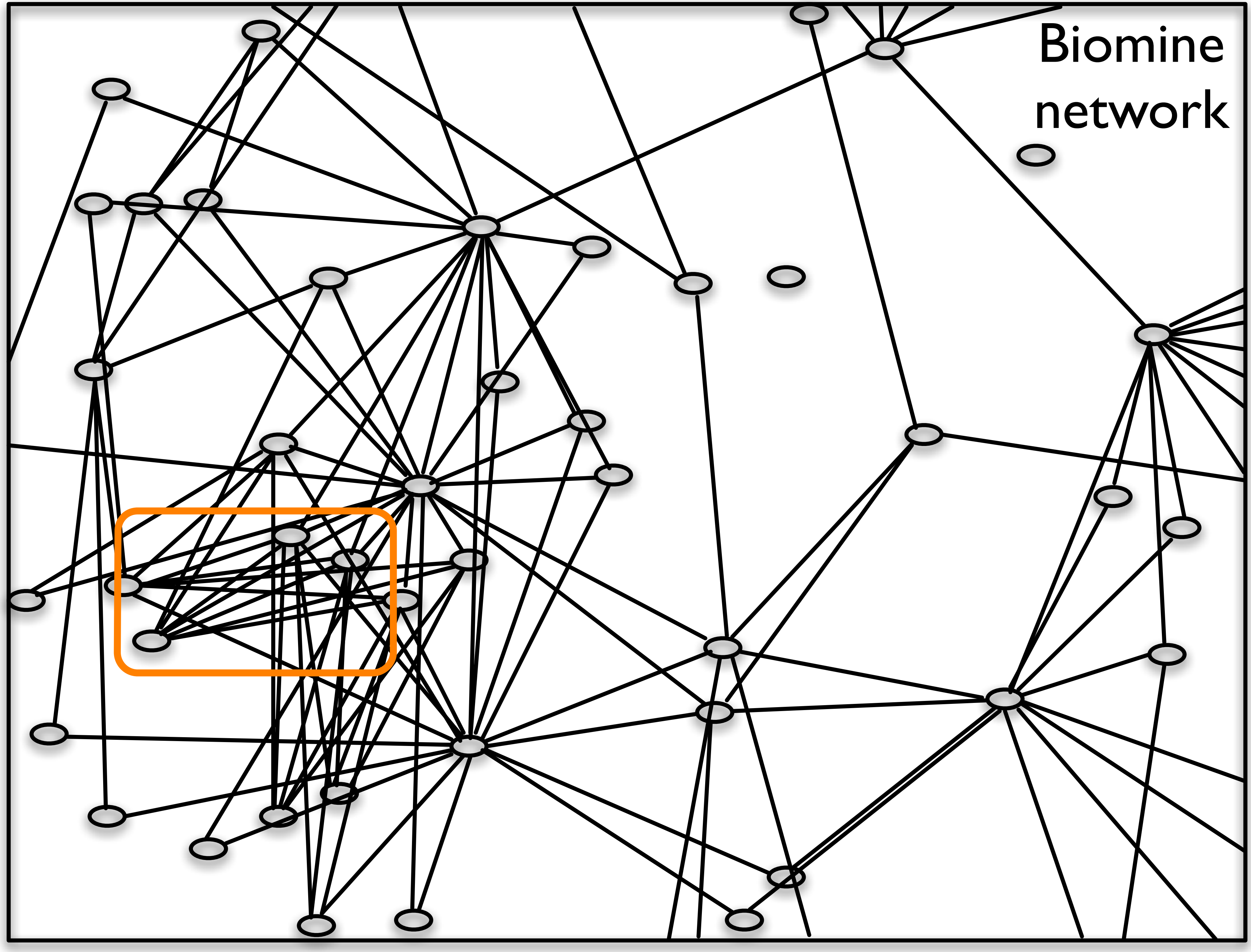
- *Focus on ProbLog line of research at KU Leuven*

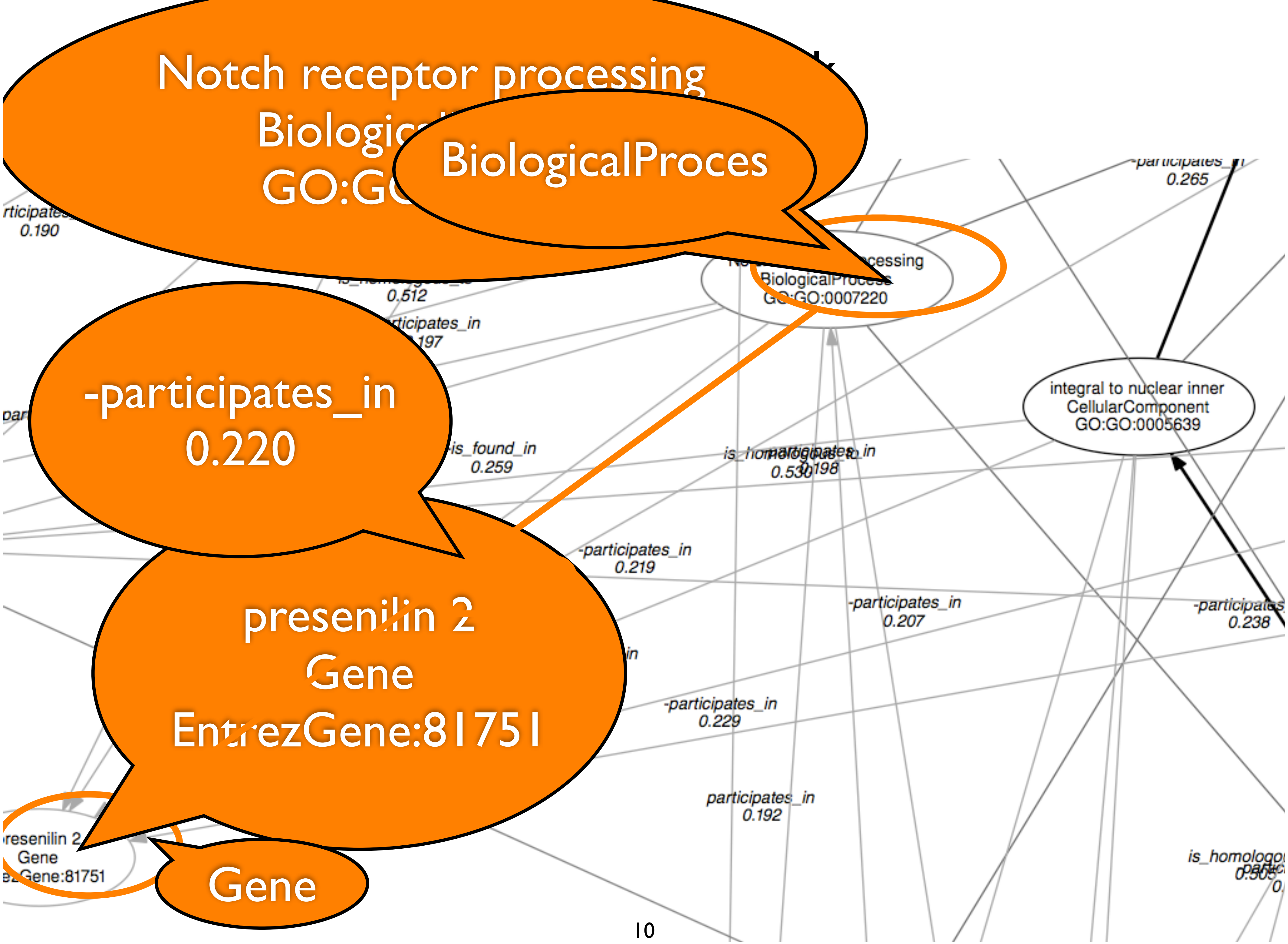
PART 1: Intro to Probabilistic Prologs

Networks of Uncertain Information



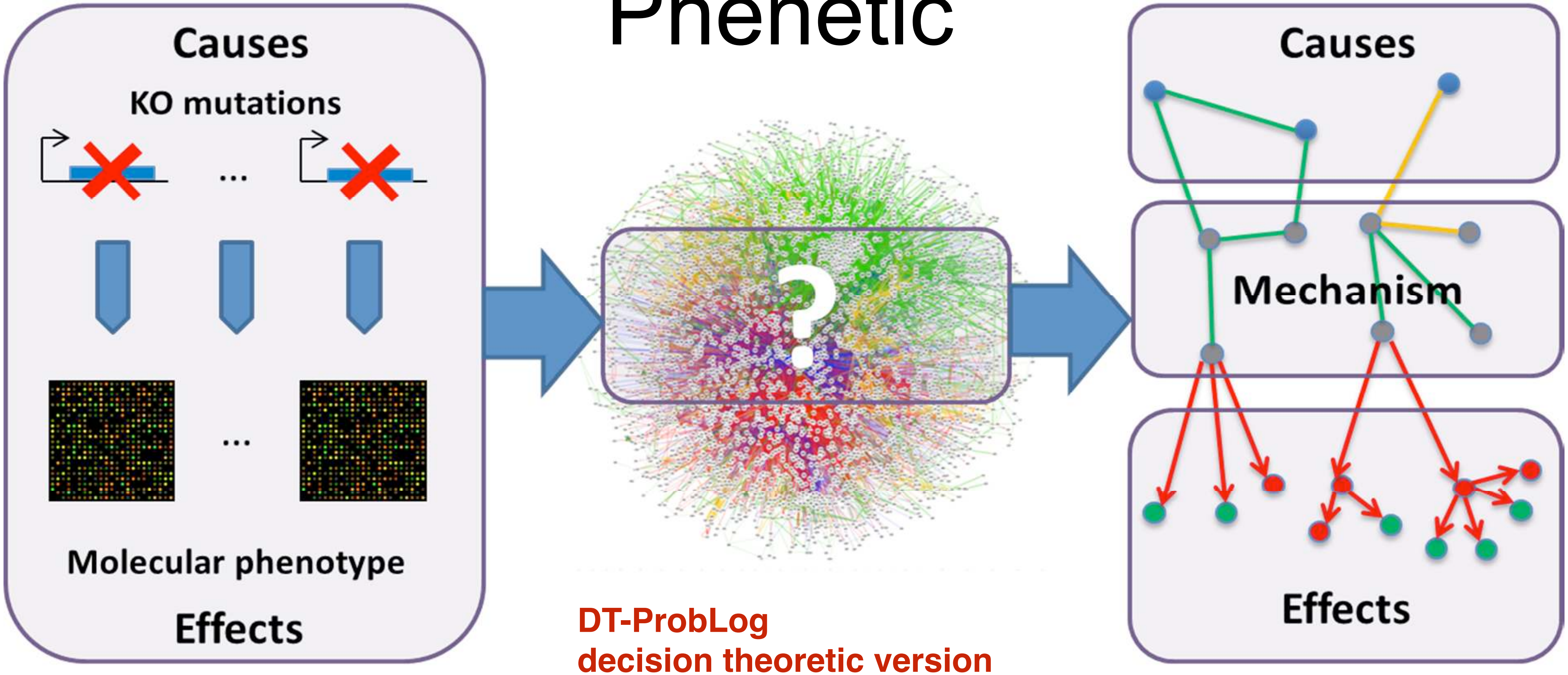
Biomine database @ Helsinki
http://biomine.cs.helsinki.fi/


Biomine
network



Notch receptor processing
BiologicalProces
-participates_in
0.220
presenilin 2
Gene
EntrezGene:81751
Gene
BiologicalProcess
GO:GO:0007220
integral to nuclear inner
CellularComponent
GO:GO:0005639
-participates_in
0.265
0.190
0.512
0.197
-is_found_in
0.259
0.530
0.198
-participates_in
0.219
-participates_in
0.207
-participates_in
0.238
-participates_in
0.229
participates_in
0.192
Gene

Phenetic



DT-ProbLog
decision theoretic version

- 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

Recently-Learned Facts twitter

Refresh

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

instances for many different relations

degree of certainty

NELL: http://rtw.ml.cmu.edu/rtw/

Graphs & Randomness

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

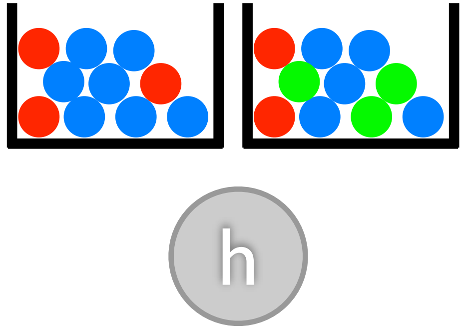
- all based on a “random graph” model

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)

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

probabilistic choices

```
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.
```

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

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

logical rule encoding background knowledge

consequences

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).
```

marginal probability

- Probability of `win`?

conditional probability

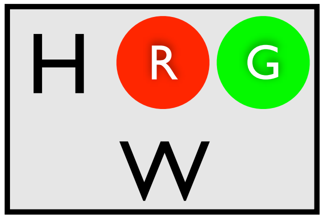
- Probability of `win` given `col(2,green)`?
- Most probable world where `win` is true?

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).
```

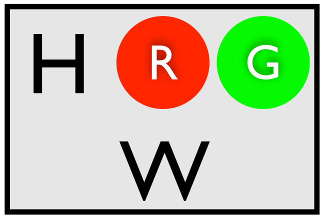
$0.4 \times 0.3 \times 0.3$



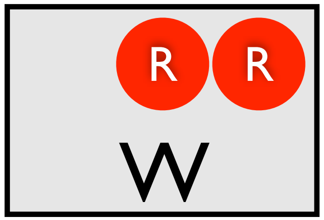
Possible Worlds

```
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).
```

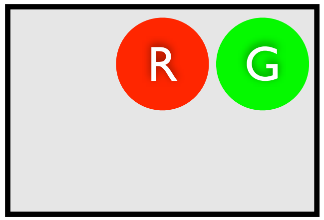
$0.4 \times 0.3 \times 0.3$



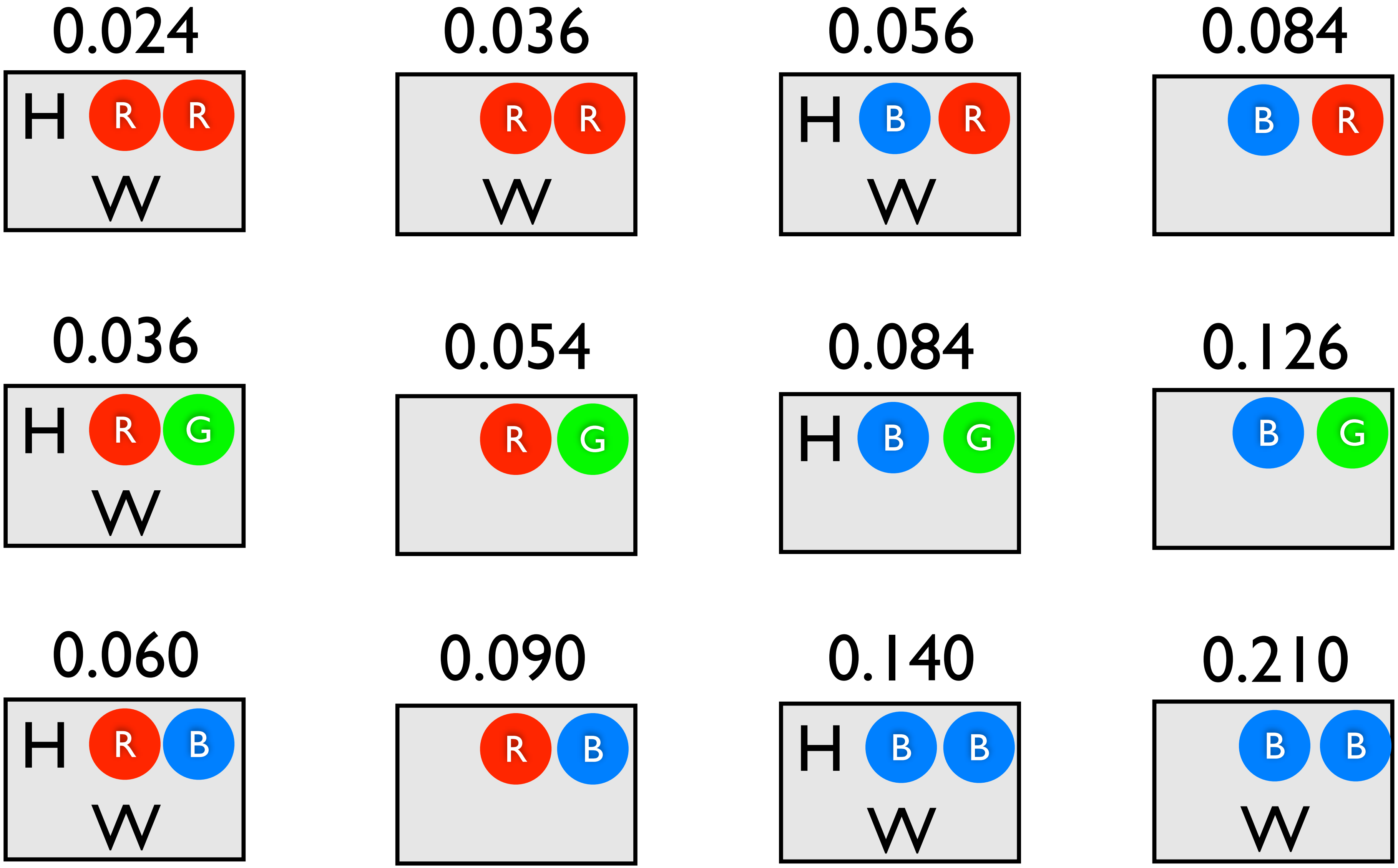
$(1-0.4)\times 0.3 \times 0.2$



$(1-0.4)\times 0.3 \times 0.3$

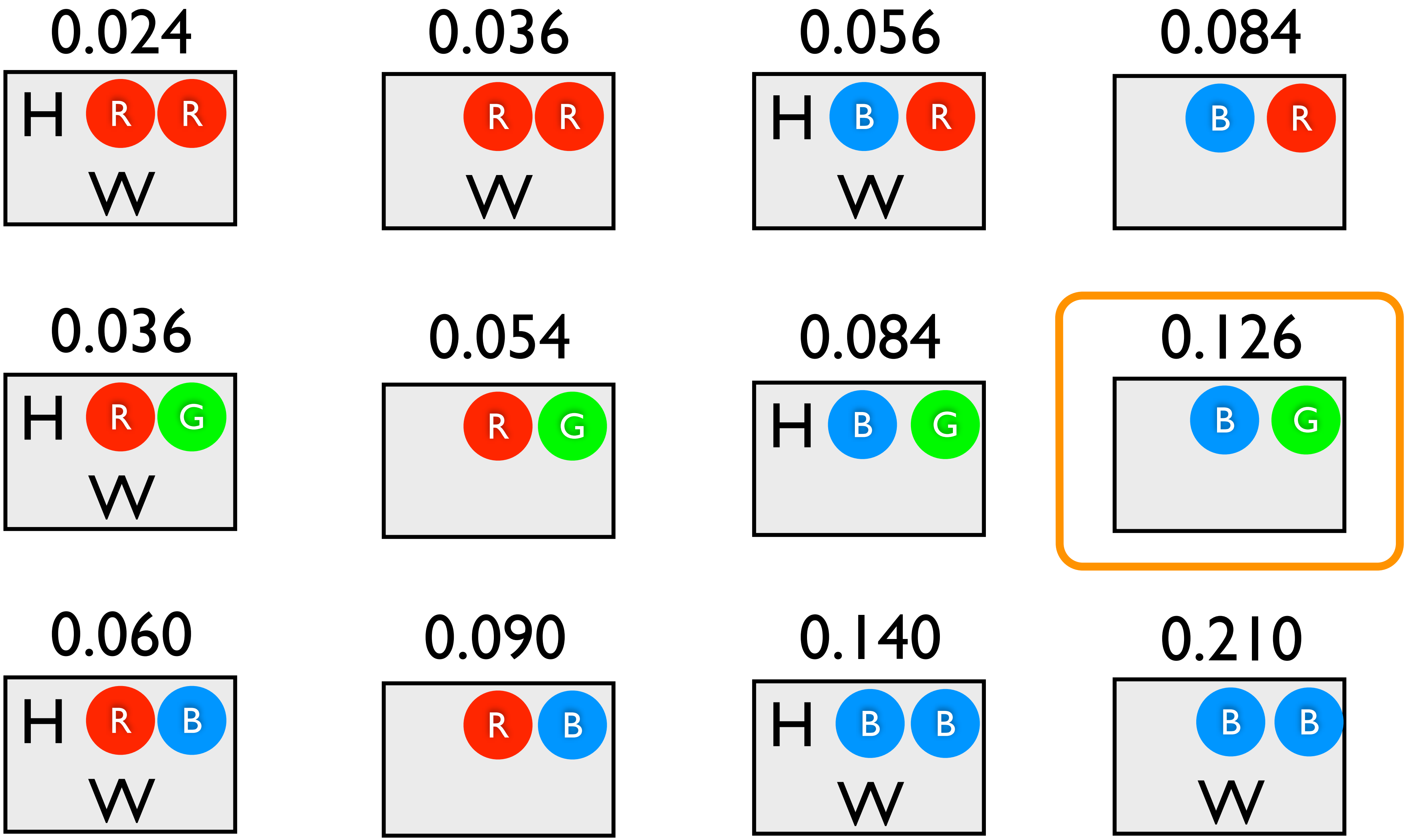


All Possible Worlds



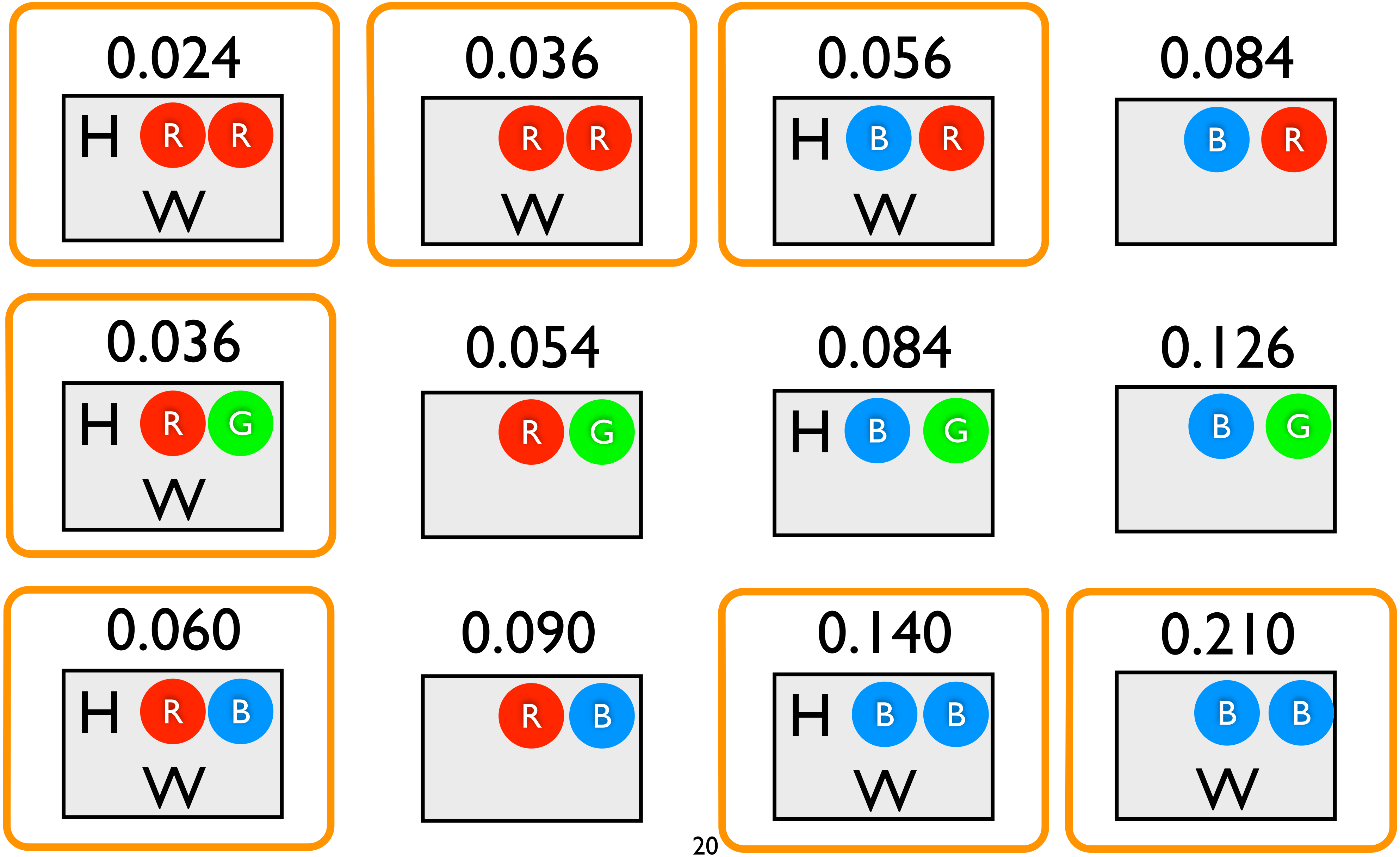
Most likely world where `col(2,blue)` is false?

MPE Inference



$$P(\underline{win}) = ?\sum = 0.562$$

Marginal Probability

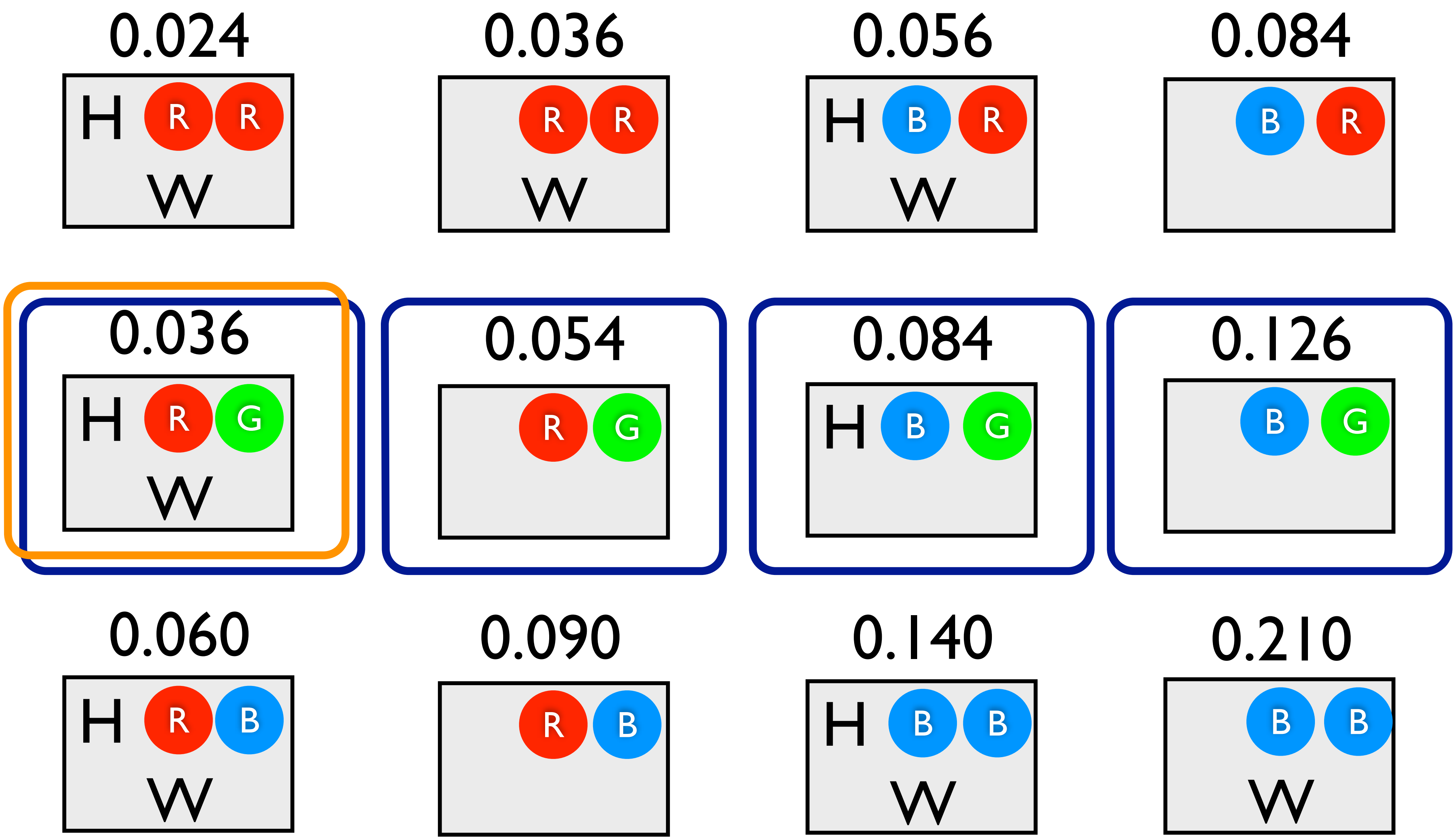


$P(win|col(2,green)) = ?/\Sigma$

$= P(win \wedge col(2,green)) / P(col(2,green))$

$= 0.036/0.3 = 0.12$

Conditional Probability



Distribution Semantics

(with probabilistic facts)

[Sato, ICLP 95]

query

sum over possible worlds where Q is true

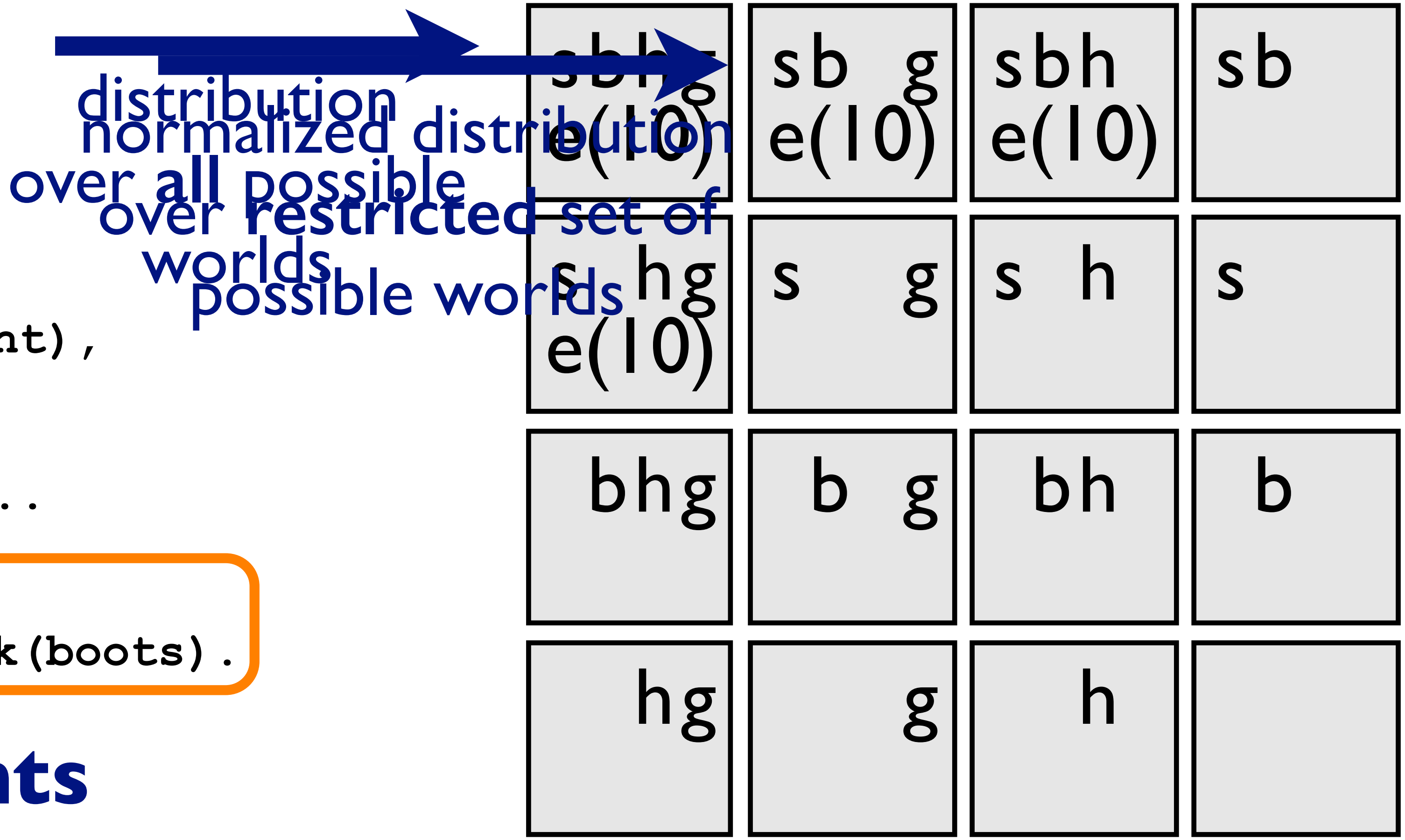
$$P(Q) = \sum_{F \cup R \models Q} \underbrace{\prod_{f \in F} p(f) \prod_{f \notin F} 1 - p(f)}_{\text{probability of possible world}}$$

subset of probabilistic facts

Prolog rules

cProbLog: constraints on possible worlds

```
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

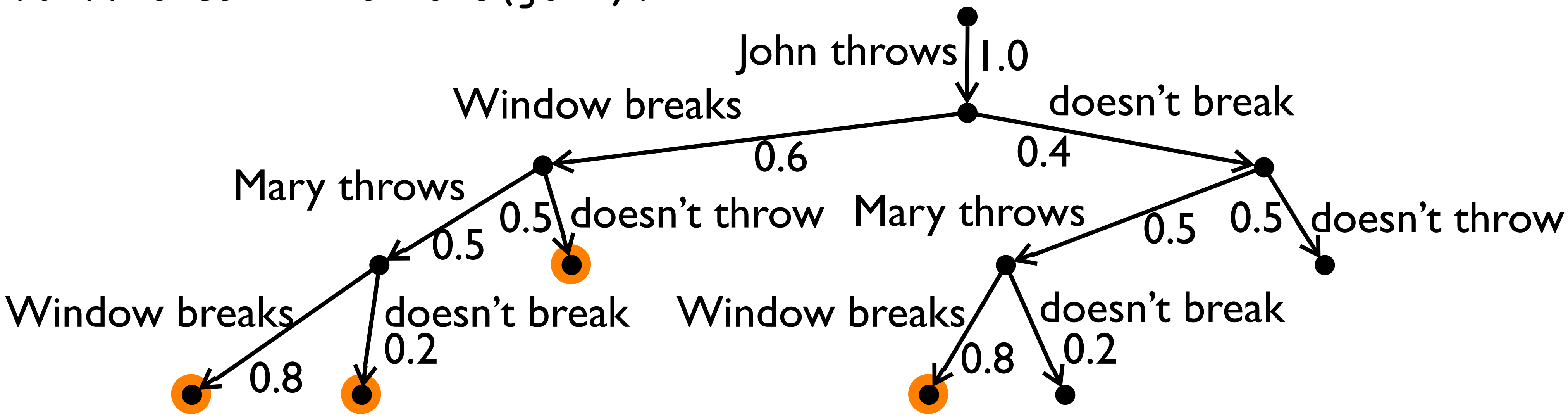
[Fierens et al, PP 12; Shterionov et al]

[Vennekens et al, ICLP 04]

Alternative view: CP-Logic

```
throws(john).
0.5::throws(mary).

0.8 :: break <- throws(mary).
0.6 :: break <- throws(john).
```

probabilistic causal laws



$$P(break)=0.6\times0.5\times0.8+0.6\times0.5\times0.2+0.6\times0.5+0.4\times0.5\times0.8$$

CP-logic [Vennekens et al.]

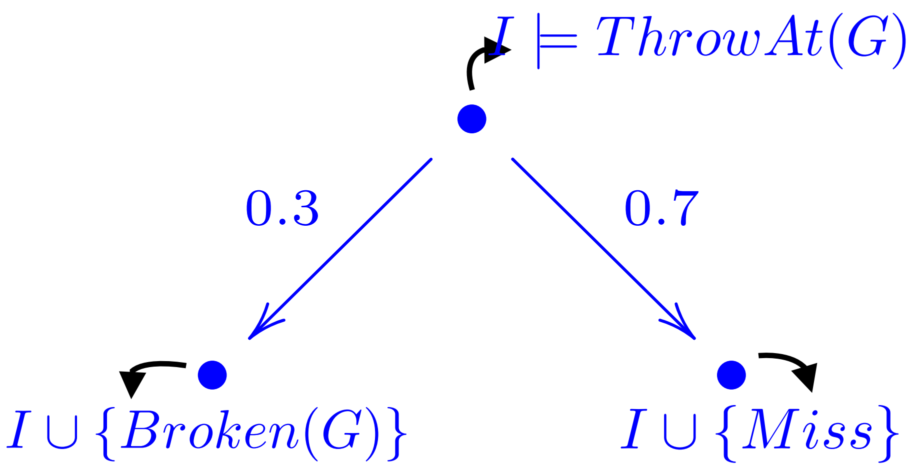
E.g., "throwing a rock at a glass breaks it with probability 0.3 and misses it with probability 0.7"

$$(Broken(G){:}0.3) \vee (Miss\ 0.7) \leftarrow ThrowAt(G).$$

ote that the actual non-deterministic event ("rock flying at glass") is implicit

Slides CP-logic courtesy Joost Vennekens

Semantics



(Broken(G) 0.3) ∨ (Miss 0.7)
← 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 <u>same</u> 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

```
length(Obj) ~ gaussian(6.0,0.45) :- type(Obj,glass).
stackable(OBot,OTop) :-
      ≃length(OBot) ≥ ≃length(OTop),
      ≃width(OBot) ≥ ≃width(OTop).
ontype(Obj,plate) ~ finite([0 : glass, 0.0024 : cup,
                            0 : pitcher, 0.8676 : plate,
                            0.0284 : bowl, 0 : serving,
                            0.1016 : none])
                  :- obj(Obj), on(Obj,O2), type(O2,plate).
```

comparing values of random variables

random variable with discrete distribution

[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

ProducesProduct

Company	Product	P
sony	walkman	0.96
microsoft	mac_os_x	0.96
ibm	personal_computer	0.96
microsoft	mac_os	0.9
adobe	adobe_indesign	0.9
adobe	adobe_dreamweaver	0.87
...	...	...

HeadquarteredIn

Company	City	P
microsoft	redmond	1.00
ibm	san_jose	0.99
emirates_airlines	dubai	0.93
honda	torrance	0.93
horizon	seattle	0.93
egyptair	cairo	0.93
adobe	san_jose	0.93
...	...	...

```
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]

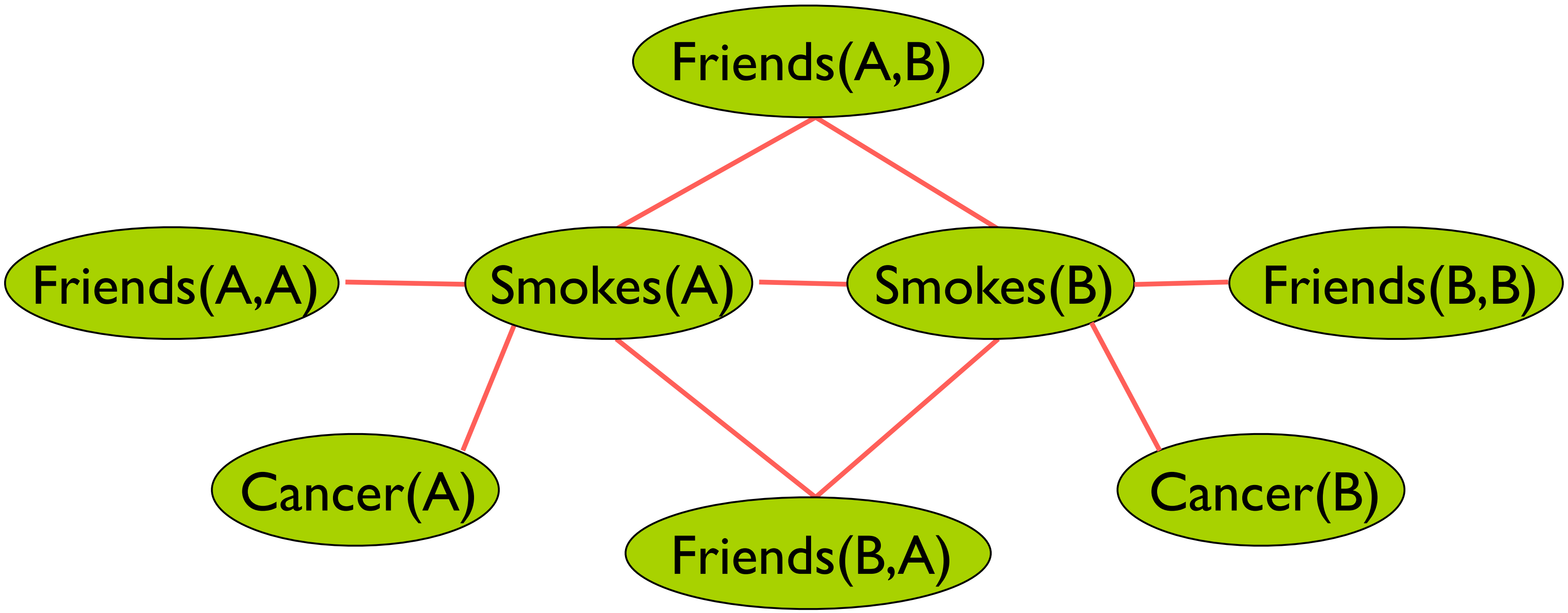
Probabilistic Programs

- 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

Markov Logic

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x,y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Suppose we have two constants: **Anna** (A) and **Bob** (B)



slides by Pedro Domingos

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

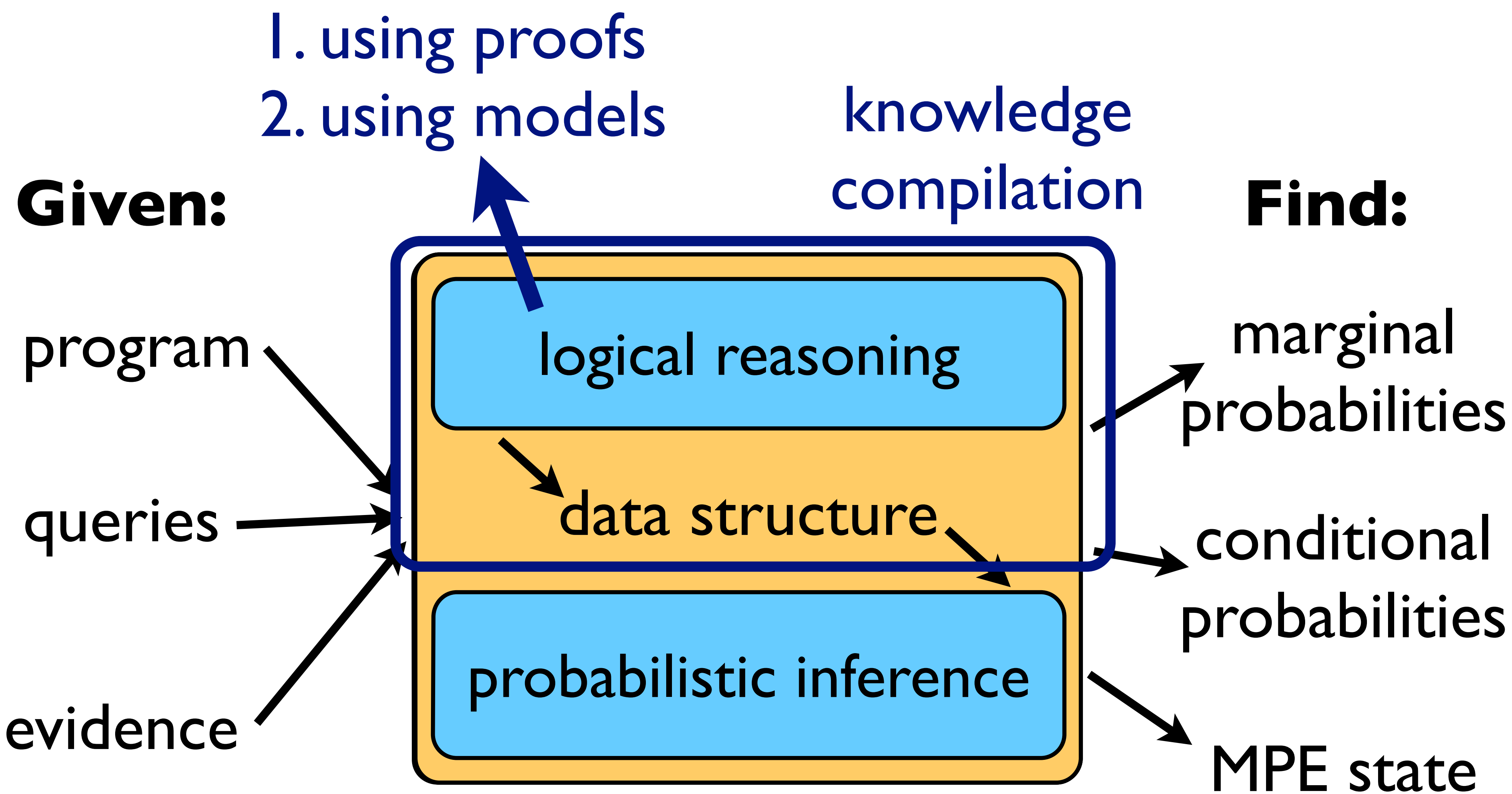
- unify basic notions in logic and in probability theory
- ground atoms become random variables
- retain the rules and the logic

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


1. using proofs
2. using models
knowledge
compilation
Given:
Find:
program
queries
evidence
logical reasoning
data structure
probabilistic inference
marginal
probabilities
conditional
probabilities
MPE state

Proofs in ProbLog

```
0.8::stress(ann).
0.4::stress(bob).
0.6::influences(ann,bob).
0.2::influences(bob,carl).

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

?- smokes(carl).

?- stress(carl).

?- influences(Y,carl),smokes(Y).

Y=bob

?- smokes(bob).

influences(bob,carl)
& stress(bob)

?- stress(bob)

0.2×0.4
= 0.08

?- influences(Y1,bob),smokes(Y1).

Y1=ann

?- smokes(ann).

?- stress(ann).

?- influences(Y2,ann),smokes(Y2).

influences(bob,carl)
& influences(ann,bob)
& stress(ann)

0.2×0.6×0.8
= 0.096

proofs overlap!
cannot sum probabilities
(disjoint-sum-problem)

Disjoint-Sum-Problem

possible worlds

solution: knowledge compilation

```
infl(bob,carl) &   infl(ann,bob) &   st(ann) & \+st(bob)     0.0576
infl(bob,carl) &   infl(ann,bob) &   st(ann) &   st(bob)     0.0384
infl(bob,carl) & \+infl(ann,bob) &   st(ann) &   st(bob)     0.0256
infl(bob,carl) &   infl(ann,bob) & \+st(ann) &   st(bob)     0.0096
infl(bob,carl) & \+infl(ann,bob) & \+st(ann) &   st(bob)     0.0064
...        influences(bob,carl) & stress(bob)
```

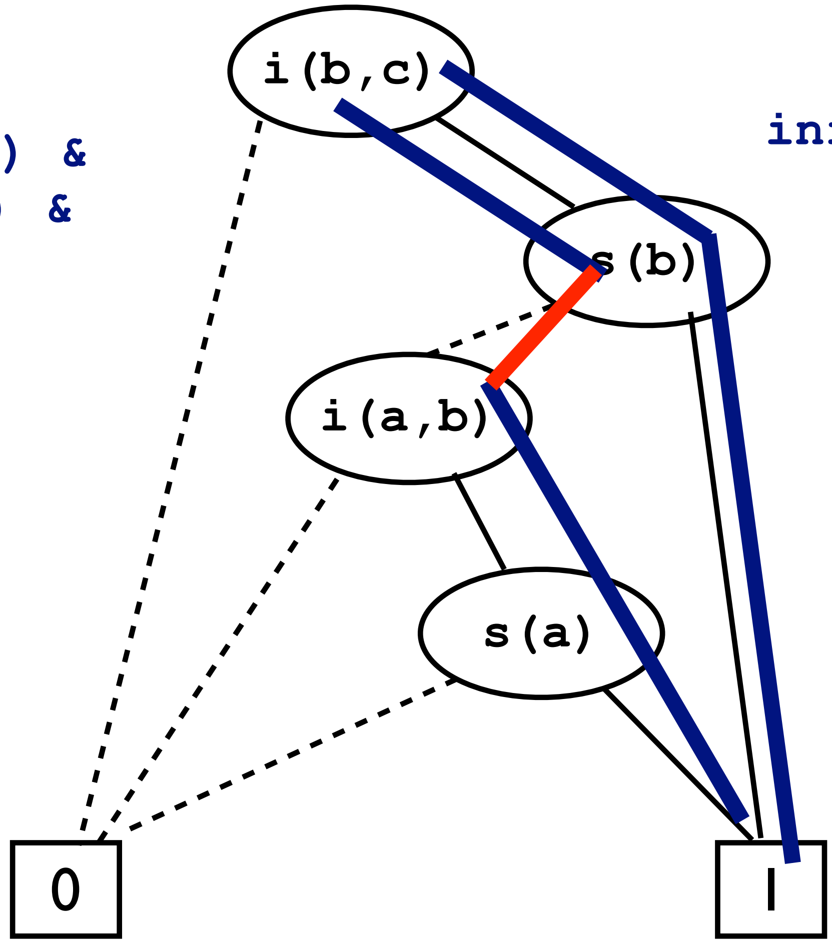
$\sum = 0.1376$

sum of proof probabilities: 0.096+0.08 = 0.1760

Binary Decision Diagrams

[Bryant 86]

influences(bob,carl) &
influences(ann,bob) &
stress(ann)
& not stress(bob)



influences(bob,carl) &
stress(bob)

Binary Decision Diagrams

0.8×0.0+0.2×0.688=0.1376

influences(bob,carl)?

0.6×0.48+0.4×1.0=0.688

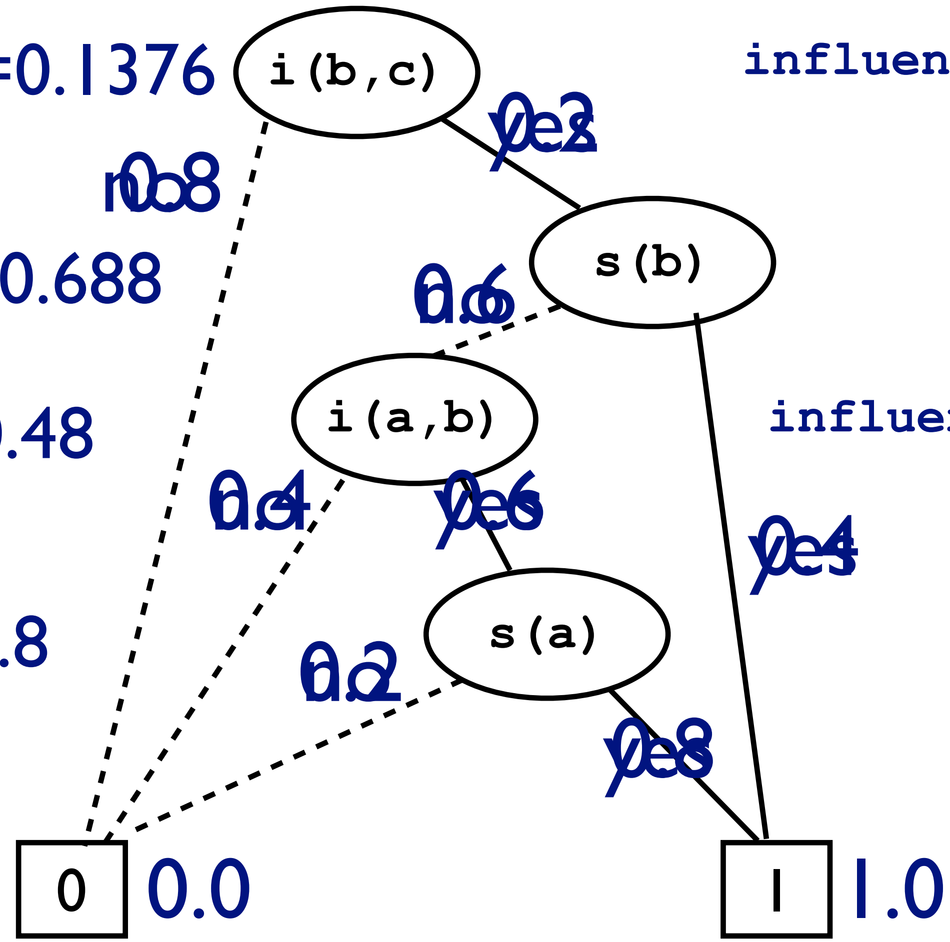
stress(bob)?

0.4×0.0+0.6×0.8=0.48

influences(ann,bob)?

0.2×0.0+0.8×1.0=0.8

stress(ann)?



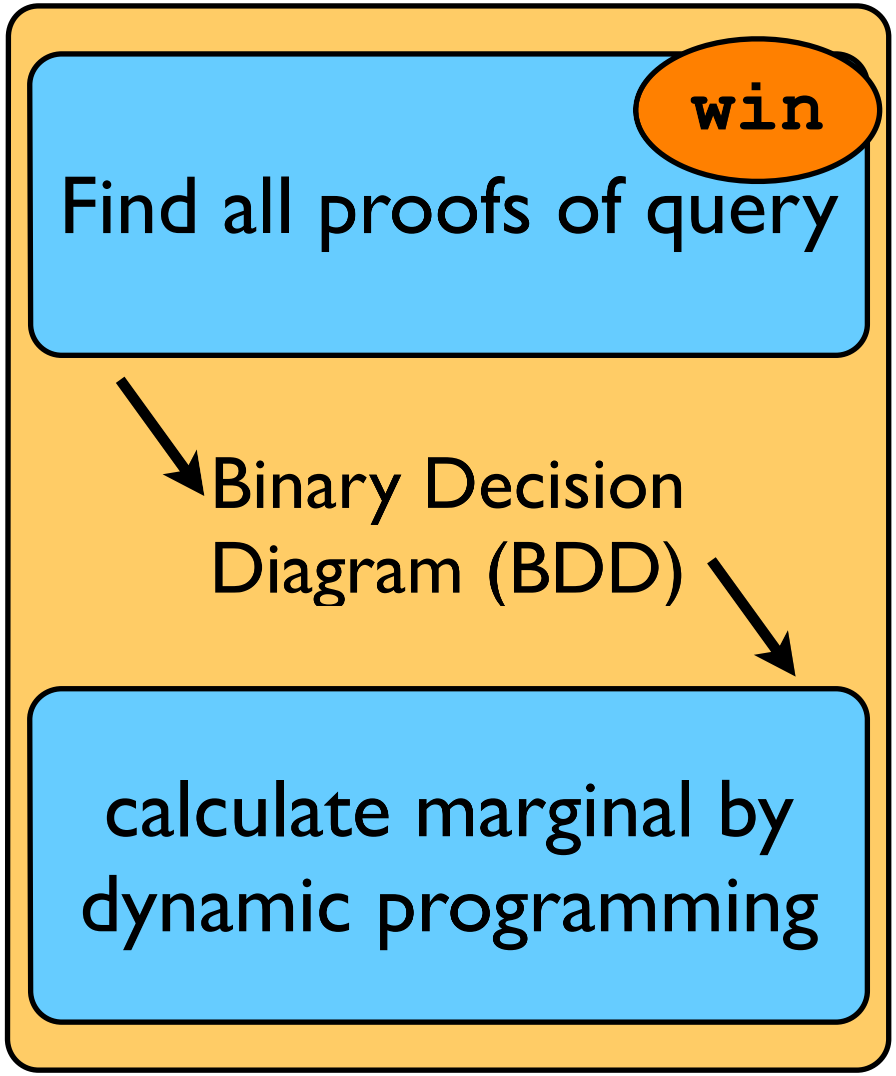
probability of `smokes(c)`?

`smokes(c) = i(b,c)&s(b) v i(b,c)&i(a,b)&s(a)`

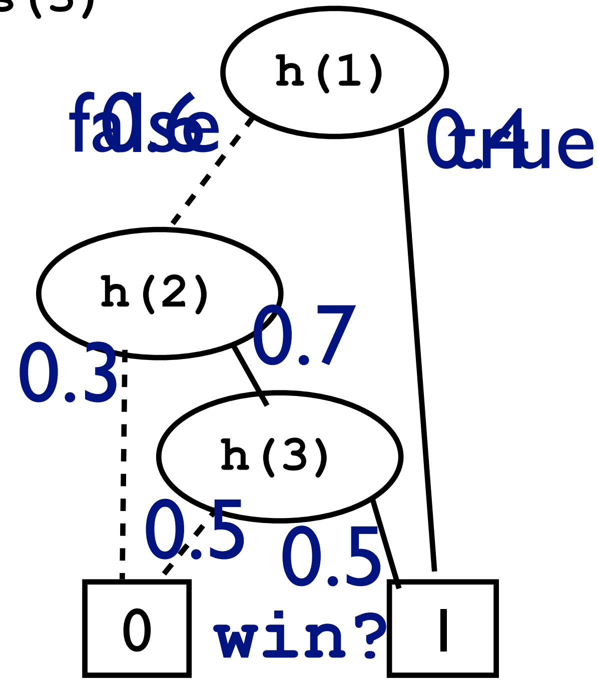
Initial Approach

(ProbLog1 & others)

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



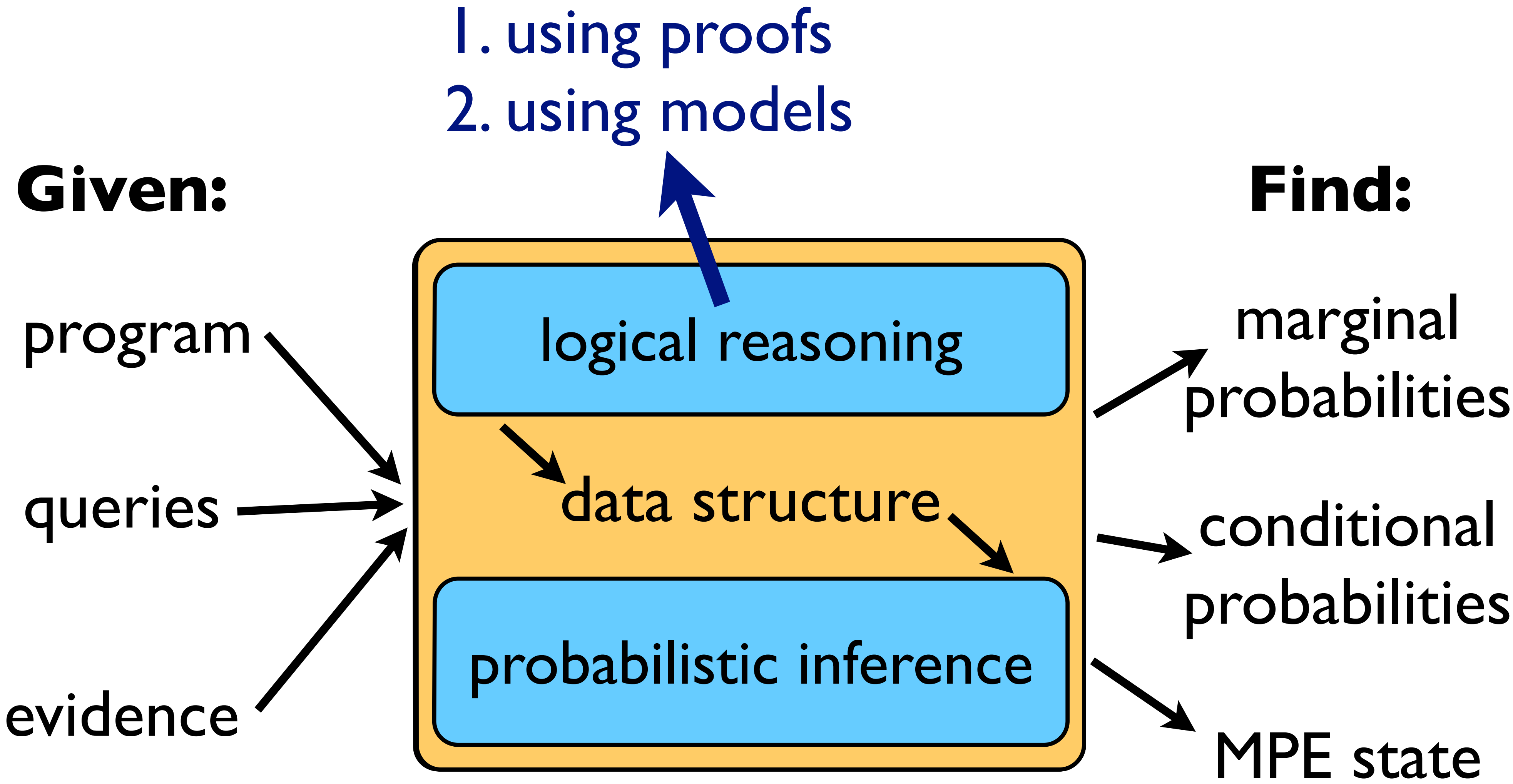

```
heads(1)
heads(2) & heads(3)
```



P(**`win`**) = probability of reaching 1-leaf

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

Answering Questions



Current Approach
(ProbLog2)

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

Find relevant ground program for queries & evidence

win

Weighted CNF

use weighted model counting / satisfiability

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

$$\text{win} \leftrightarrow h(1) \lor (h(2) \land h(3))$$

may require loop-breaking

$$(\neg\text{win} \lor h(1) \lor h(2)) \land (\neg\text{win} \lor h(1) \lor h(3)) \land (\text{win} \lor \neg h(1)) \land (\text{win} \lor \neg h(2) \lor \neg h(3))$$

use standard tool

$h(1) \rightarrow 0.4$	$h(2) \rightarrow 0.7$	$h(3) \rightarrow 0.5$
$\neg h(1) \rightarrow 0.6$	$\neg h(2) \rightarrow 0.3$	$\neg h(3) \rightarrow 0.5$

[Fierens et al, TPLP 14]

ProbLog → CNF

```
0.8::stress(ann).
0.4::stress(bob).
0.6::influences(ann,bob).
0.2::influences(bob,carl).

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

```
?- smokes(carl).
```

- 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

may require loop-breaking

$$\mathrm{sm}(c) \leftrightarrow (\mathrm{i}(b,c) \wedge \mathrm{sm}(b))$$
$$\wedge\ \mathrm{sm}(b) \leftrightarrow (\mathrm{st}(b) \vee (\mathrm{i}(a,b) \wedge \mathrm{sm}(a)))$$
$$\wedge\ \mathrm{sm}(a) \leftrightarrow \mathrm{st}(a)$$

- Rewrite in CNF (as usual)

Weighte

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

propositional formula in conjunctive normal form (CNF)

given by ProbLog program & query

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

interpretations (truth value assignments) of propositional variables

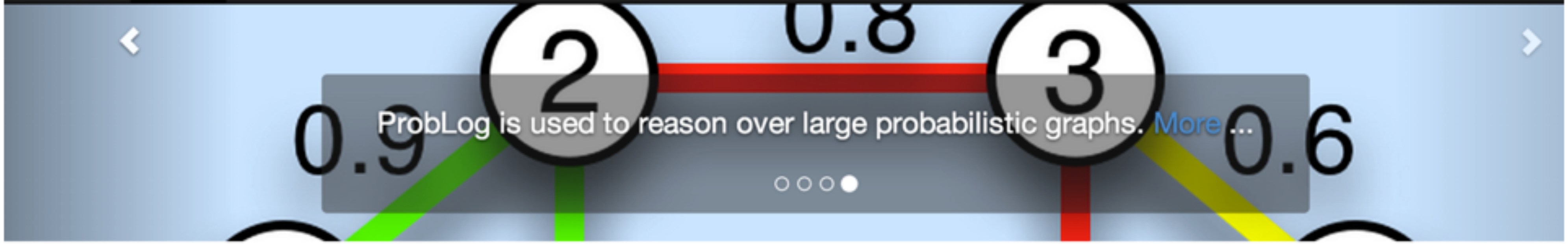
possible worlds

weight of literal

for p::f,
w(f) = p
w(not f) = 1−p

dtai.cs.kuleuven.be/problog/

ProbLog | Home | Download | Publications | Help | People | Interactive



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.

```
0.3::stress(X) :- person(X).
0.2::influences(X,Y) :- person(X), person(Y).
```

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

Recently-Learned Facts twitter

Refresh

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

instances for many different relations

degree of certainty

NELL: http://rtw.ml.cmu.edu/rtw/

Rule learning in NELL (1)

- 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

ex	outlook ok	temperature ok	humidity ok	wind ok		class
1	t	t	f	f	-	+
2	f	t	f	t	-	+
3	t	f	f	f	-	-
4	f	f	t	f	-	-
...						
...						

Our Windsurfing Example

ex	pop	windok	sunshine	class
1	0,7	0,5	0,7	0,9
2	0,6	0,7	0,6	0,85
3	0,4	0,3	0,4	0,45
4	0,3	0,7	0,2	0,3
...				
...				

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

playtennis :- outlook=ok, wind=ok

playtennis :- outlook=ok, humidity=ok

In the probabilistic case

surfing :- not pop, windok

surfing :- not pop, 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(surfing) = P((not pop and windok) or (not pop and sunshine))

= P((not pop and windok) or (not pop and sunshine and not windok)

= 0.8 x 0.7 + 0.8 x 0.6 x 0.3 disjoint sum problem

In ProbLog (1)

Basic Setting

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

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

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

?-P(surfing(e1)). e

gives $0.8 \times 0.7 + 0.8 \times 0.6 \times 0.3 = P(B \cup H \models e)$

In ProbLog (2)

Extended Setting

p1:: surfing(X) :- not pop(X) and windok(X).

H

p2:: surfing(X) :- not pop(X) and sunshine(X).

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

e

?-P(surfing(e1)).

gives 0.8 x 0.7 x p1 + 0.8 x 0.6 x 0.3 x p2 = P(B U H |= e)

Inductive Probabilistic Logic Programs

Given

a set of example facts e ∈ E together with the probability p that they hold

a background theory B in ProbLog

a hypothesis space L (a set of clauses)

Find

$$\arg\min_H loss(H, B, E) = \arg\min_H \sum_{e_i \in E} |P_s(B \cup H \models e) - p_i|$$

Observations

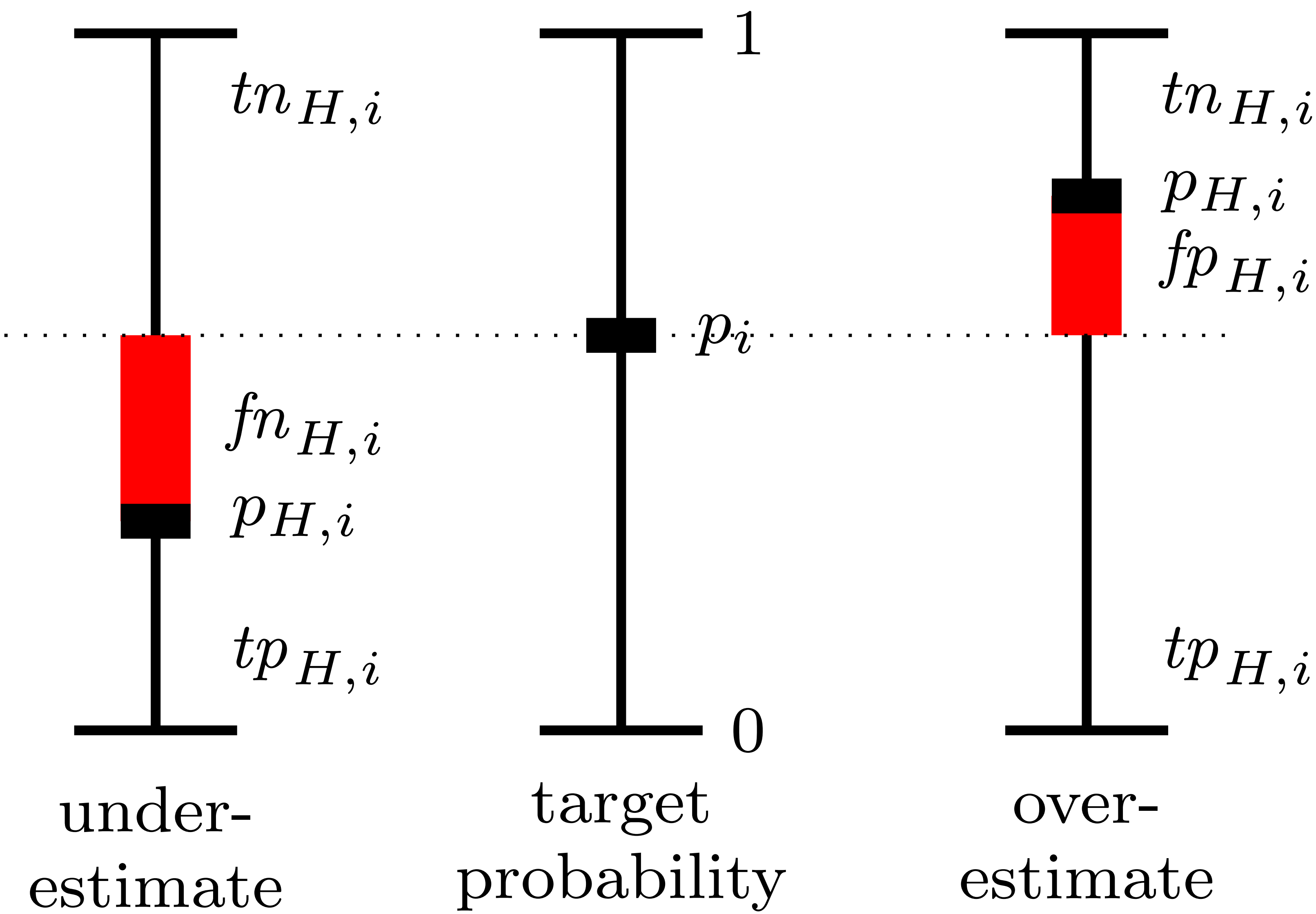
Propositional versus first order

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

Deterministic case

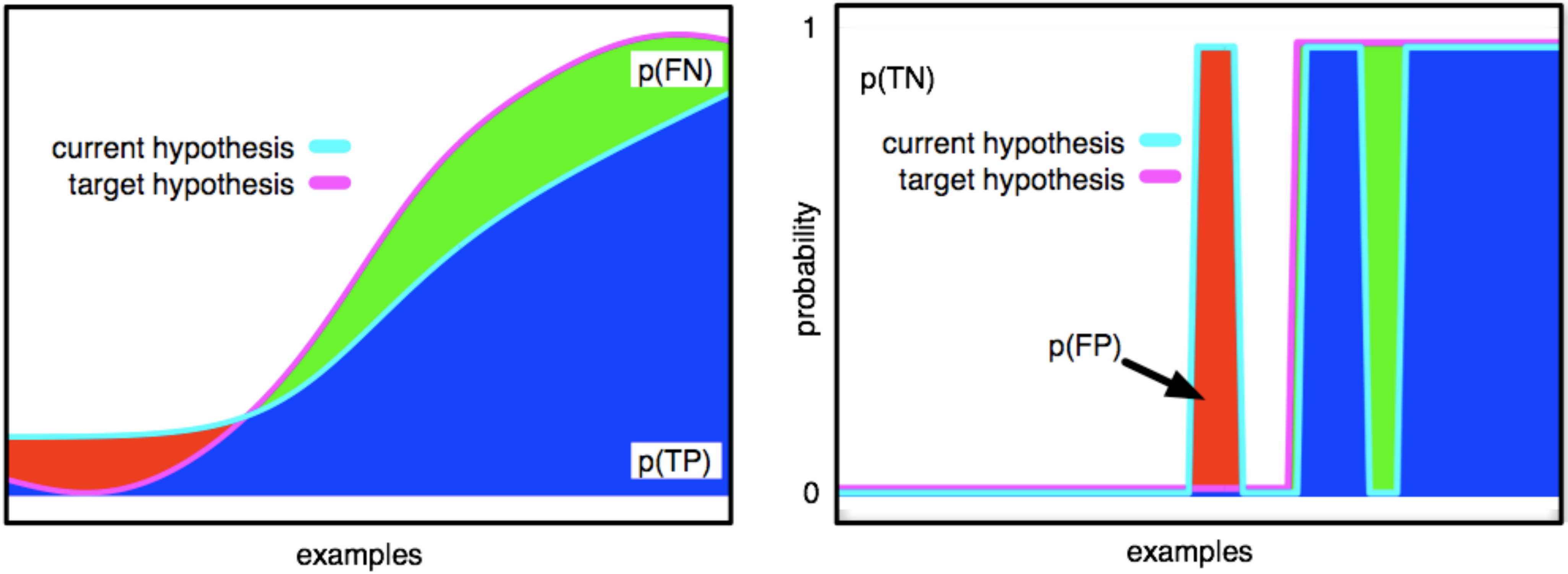
- all probabilities 0 or 1
- traditional rule learning / ILP as special case

Analysis

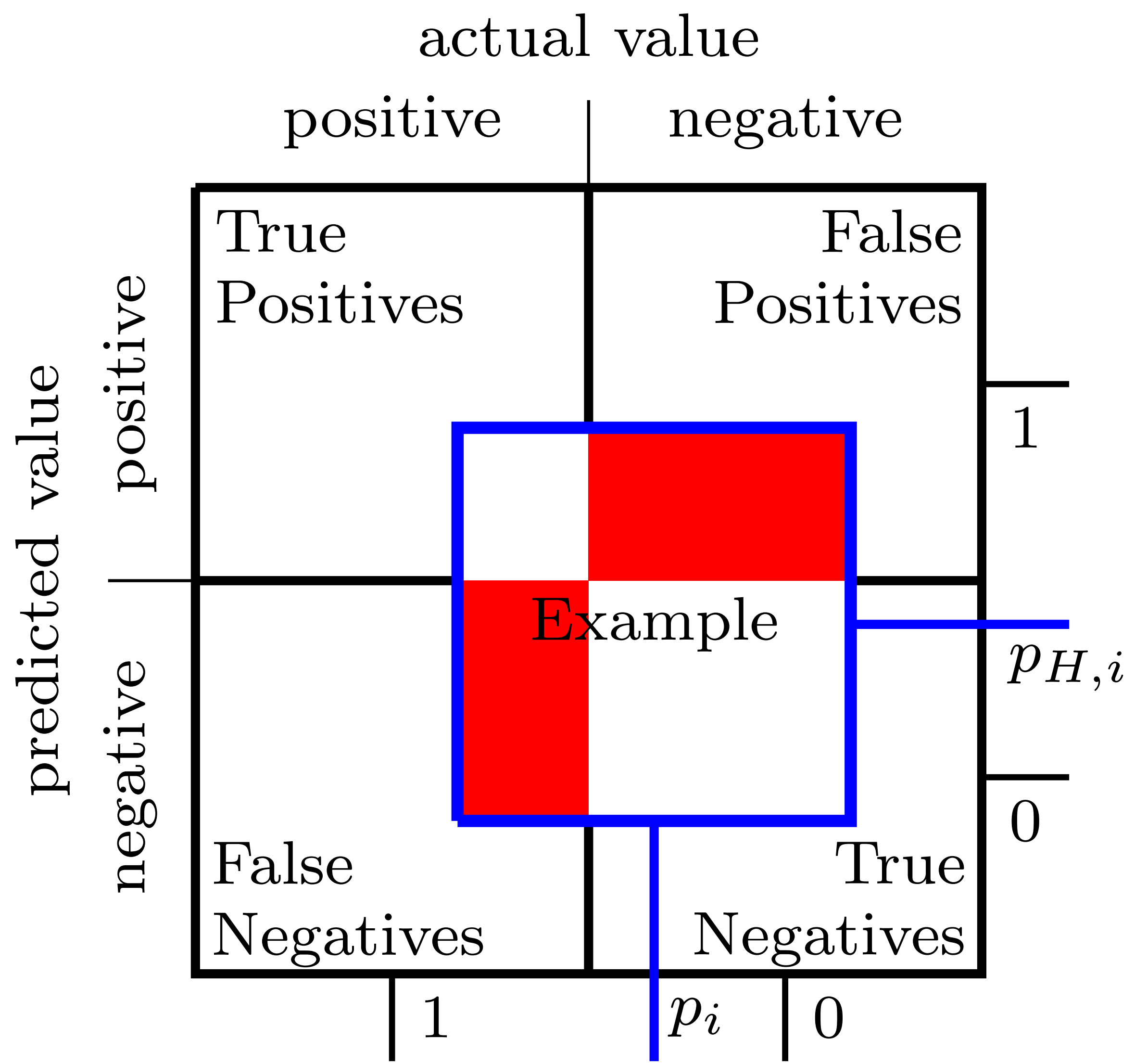


1
0
$tn_{H,i}$
$fn_{H,i}$
$p_{H,i}$
$tp_{H,i}$
p_i
$tn_{H,i}$
$p_{H,i}$
$fp_{H,i}$
$tp_{H,i}$
under-
estimate
target
probability
over-
estimate

Analysis


p(FN)
current hypothesis
target hypothesis
p(TP)
examples
1
p(TN)
current hypothesis
target hypothesis
probability
p(FP)
0
examples

Contingency Table



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

$$
\begin{aligned}
precision &= \frac{TP}{TP + FP} \\
m\text{-}estimate &= \frac{TP + m \cdot \frac{P}{N}}{TP + FP + m} \\
recall &= \frac{TP}{TP + FN} \\
accuracy &= \frac{TP + TN}{TP + TN + FP + FN}
\end{aligned}
$$

local score

global score

Avoiding overfitting using significance test

ProbFOIL

Algorithm 1 The ProbFOIL$^+$ learning algorithm

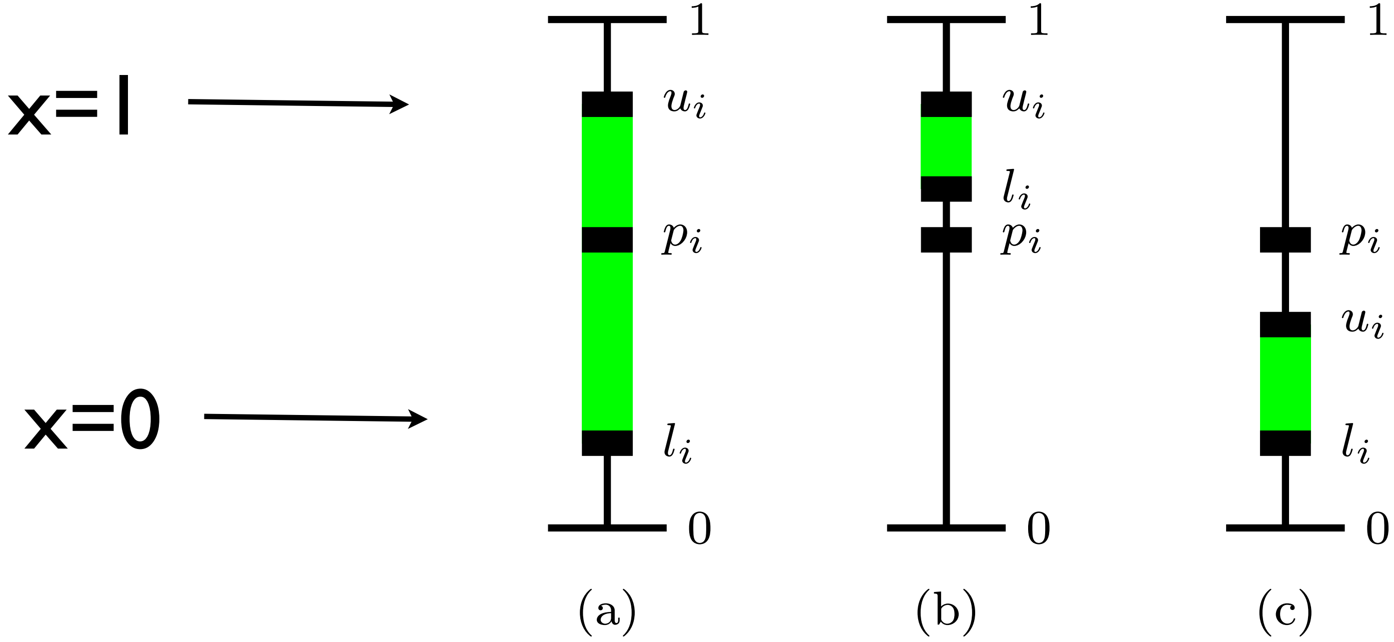
```
function ProbFOIL+(target)                                   ▷ target is the target predicate
    H := ∅
    while true do
        clause := LearnRule(H, target)
        if GlobalScore(H) < GlobalScore(H ∪ {clause}) then
            H := H ∪ {clause}
        else
            return H
function LearnRule(H, target)
    candidates := {x :: target ← true}                       ▷ Start with an empty (probabilistic) body
    bestrule := (x :: target ← true)
    while candidates ≠ ∅ do                                  ▷ Grow rule
        nextcandidates := ∅
        for all x :: target ← body ∈ candidates do
            for all literal ∈ ρ(target ← body) do            ▷ Generate all refinements
                if not RejectRefinement(H, bestrule, x :: target ← body) then    ▷ Reject unsuited refinements
                    nextcandidates := nextcandidates ∪ {x :: target ← body ∧ l}
                    if LocalScore (H, x :: target ← body ∧ literal) > LocalScore(H, bestrule) then
                        bestrule := (x :: target ← body ∧ literal)      ▷ Update best rule
        candidates := nextcandidates
    return bestrule
```

Extended rule learning

Learn rules with probability *x:: head :- body*

What changes ?

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



Extended rule learning

Express local score as a function of *x*

Compute optimal value of *x*

Implementation Optimizations

Incremental grounding

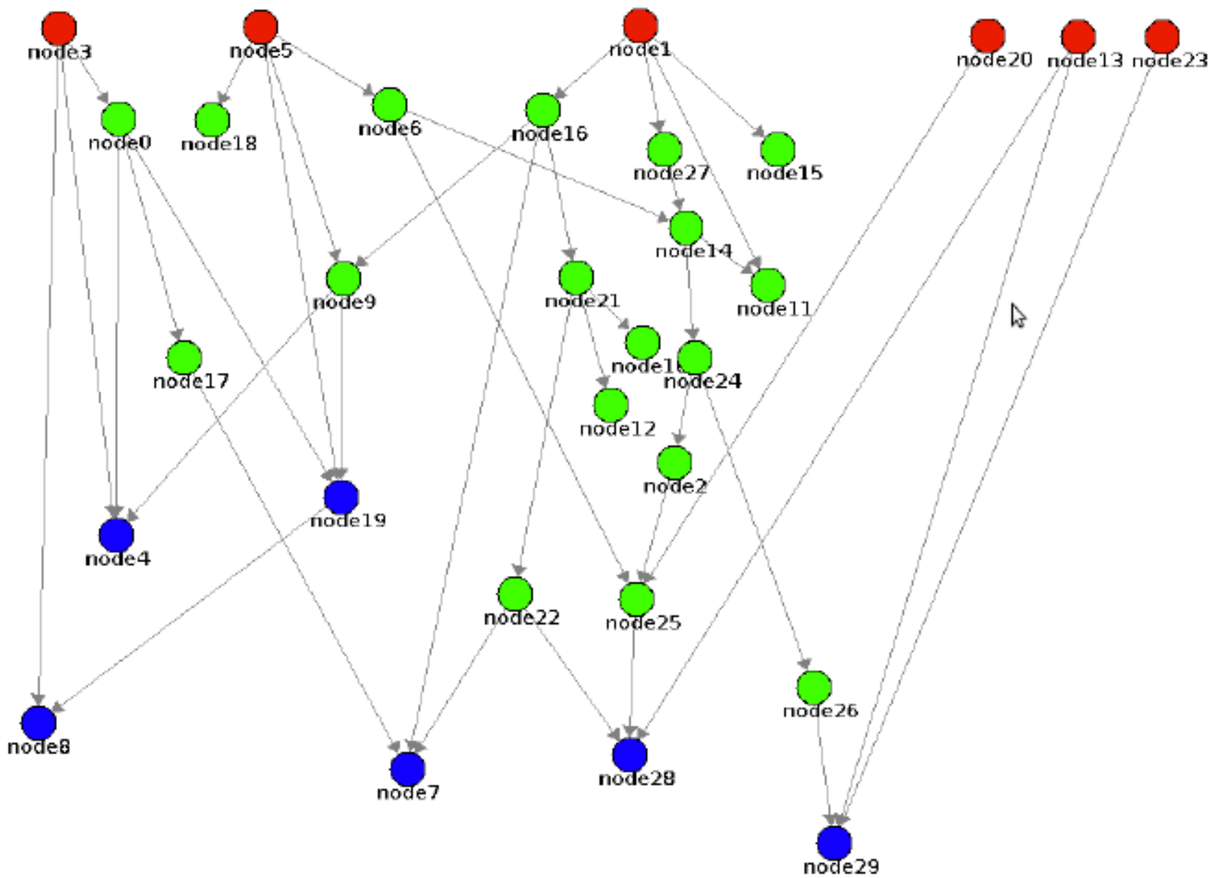
Simplified CNF conversion to ProbLog

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 $\sim$ Beta(α, β), averaged over all target attributes. Observed nodes are independent.

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

Observed Nodes Independent

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

Results

α/β	1.0000	0.1000	0.0100	0.0010	0.0001	0.00001
LinearRegression	2.6×10^{-3}	0.018	0.021	0.020	0.018	0.018
MultilayerPerceptron	4×10^{-4}	3.1×10^{-3}	5.7×10^{-3}	3.9×10^{-3}	3.3×10^{-3}	3.3×10^{-3}
M5P	7×10^{-4}	4.9×10^{-3}	6.5×10^{-3}	5.4×10^{-3}	4.4×10^{-3}	4.4×10^{-3}
M5P -R -M 4.0	5.2×10^{-3}	0.021	0.023	0.025	0.024	0.024
SMOreg	2.6×10^{-3}	0.017	0.021	0.020	0.017	0.017
ProbFOIL(1,10,0.0,rel)	0.015	0.012	1.9×10^{-3}	9.4×10^{-8}	4.2×10^{-8}	4.2×10^{-8}
ProbFOIL$^+$(1,5,0.0,rel)	3.9×10^{-3}	3.9×10^{-3}	5.3×10^{-4}	2.8×10^{-7}	4.2×10^{-8}	4.2×10^{-8}

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 $\sim$ Beta(α, β), averaged over all target attributes. Observed nodes are **dependent**. There is **partial** observability.

α/β	1.0000	0.1000	0.0100	0.0010	0.0001	0.00001
ZeroR	0.023	0.077	0.085	0.093	0.096	0.096
LinearRegression	4.8×10^{-3}	0.019	0.026	0.032	0.034	0.034
MultilayerPerceptron	1.2×10^{-3}	2.9×10^{-3}	9.7×10^{-3}	0.020	0.031	0.032
M5P	1.6×10^{-3}	6.1×10^{-3}	0.022	0.027	0.027	0.027
M5P -R -M 4.0	6.2×10^{-3}	0.022	0.033	0.040	0.040	0.040
SMOreg	4.5×10^{-3}	0.018	0.022	0.029	0.034	0.034
ProbFOIL(1,15,0.0,rel)	0.020	0.023	0.012	0.015	0.015	0.015
ProbFOIL$^+$(1,10,0.0,rel)	9.5×10^{-3}	0.011	0.011	0.013	0.013	0.013

Observed Nodes Dependent / Partial Observability

ProbFOIL better on deterministic data
Regression learners sometimes better than ProbFOIL

Information Extraction in NELL

Recently-Learned Facts twitter

Refresh

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

instances for many different relations

degree of certainty

NELL: http://rtw.ml.cmu.edu/rtw/

NELL

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

athletecoach(person,person)	18	athleteplaysforteam(person,team)	721
athleteplayssport(person,sport)	1921	teamplaysinleague(team,league)	1085
athleteplaysinleague(person,league)	872	athletealsoknownas(person,name)	17
coachesinleague(person,league)	93	coachesteam(person,team)	132
teamhomestadium(team,stadium)	198	teamplayssport(team,sport)	359
athleteplayssportsteamposition(person,position)	255	athletehomestadium(person,stadium)	187
athlete(person)	1909	attraction(stadium)	2
coach(person)	624	female(person)	2
male(person)	7	hobby(sport)	5
organization(league)	1	person(person)	2
personafrica(person)	1	personasia(person)	4
personaustralia(person)	22	personcanada(person)	1
personeurope(person)	1	personmexico(person)	108
personus(person)	6	sport(sport)	36
sportsleague(league)	18	sportsteam(team)	1330
sportsteamposition(position)	22	stadiumoreventvenue(stadium)	171

NELL

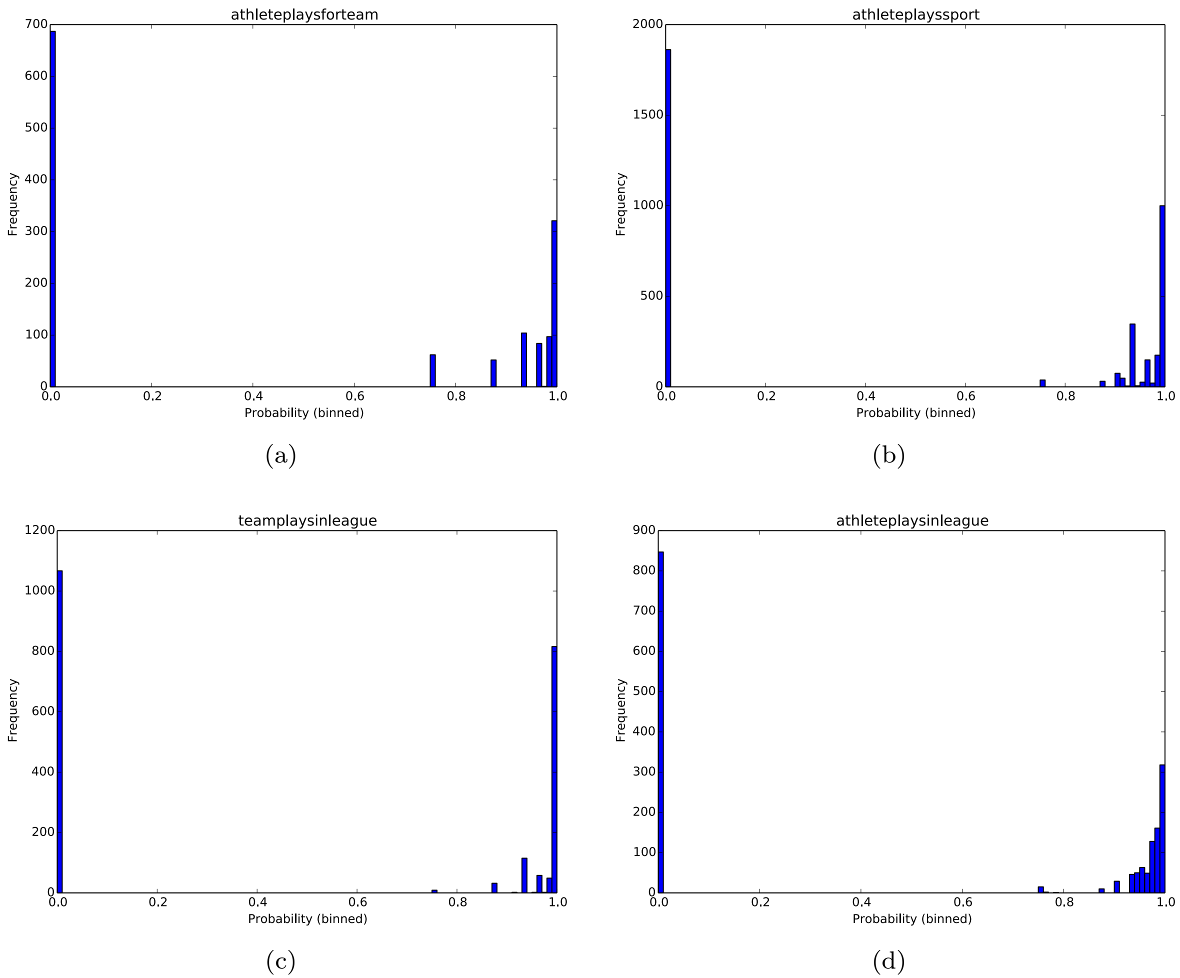


Fig. 5: Histogram of probabilities for each of the binary predicates with more then 500 facts: (a) athleteplaysforteam; (b) athleteplayssport; (c) teamplaysinleague; and, (d) athleteplaysinleague.

5.4.2 *athleteplayssport(person,sport)*

```
0.98604::athleteplayssport(A,B) ← athleteplaysforteam(V_2,V_1),
     athleteplaysinleague(A,V_3), coachesinleague(V_2,V_3),
     teamplayssport(V_1,B), athleteplaysforteam(A,V_1).
0.907::athleteplayssport(A,B) ← athleteplaysforteam(V_2,V_1), teammate(V_2,A),
     teamplayssport(V_1,B).
0.91817::athleteplayssport(A,B) ← coachesteam(A,V_1), teamplayssport(V_1,B).
```

Listing 2: Learned rules for the `athleteplayssport` predicate.

5.4.3 *teamplaysinleague(team,league)*

```
0.95848::teamplaysinleague(A,B) ← athleteplaysforteam(V_1,V_2),
     coachesinleague(V_1,B), teamplaysagainstteam(V_2,A).
0.92240::teamplaysinleague(A,B) ← athleteplaysinleague(V_1,B),
     coachesteam(V_1,V_2), teamplaysagainstteam(V_2,A),
     athleteplaysforteam(V_1,V_2), athleteledsportsteam(V_1,V_2).
1.0::teamplaysinleague(A,B) ← athleteledsportsteam(V_1,A),
     athleteplaysinleague(V_1,B).
0.99998::teamplaysinleague(A,B) ← coachesinleague(V_1,B), coachwontrophy(V_1,V_2),
     teamwontrophy(A,V_2).
1.0::teamplaysinleague(A,B) ← athleteplaysinleague(V_1,B), coachesteam(V_1,V_2),
     teamplaysagainstteam(V_2,A).
```

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

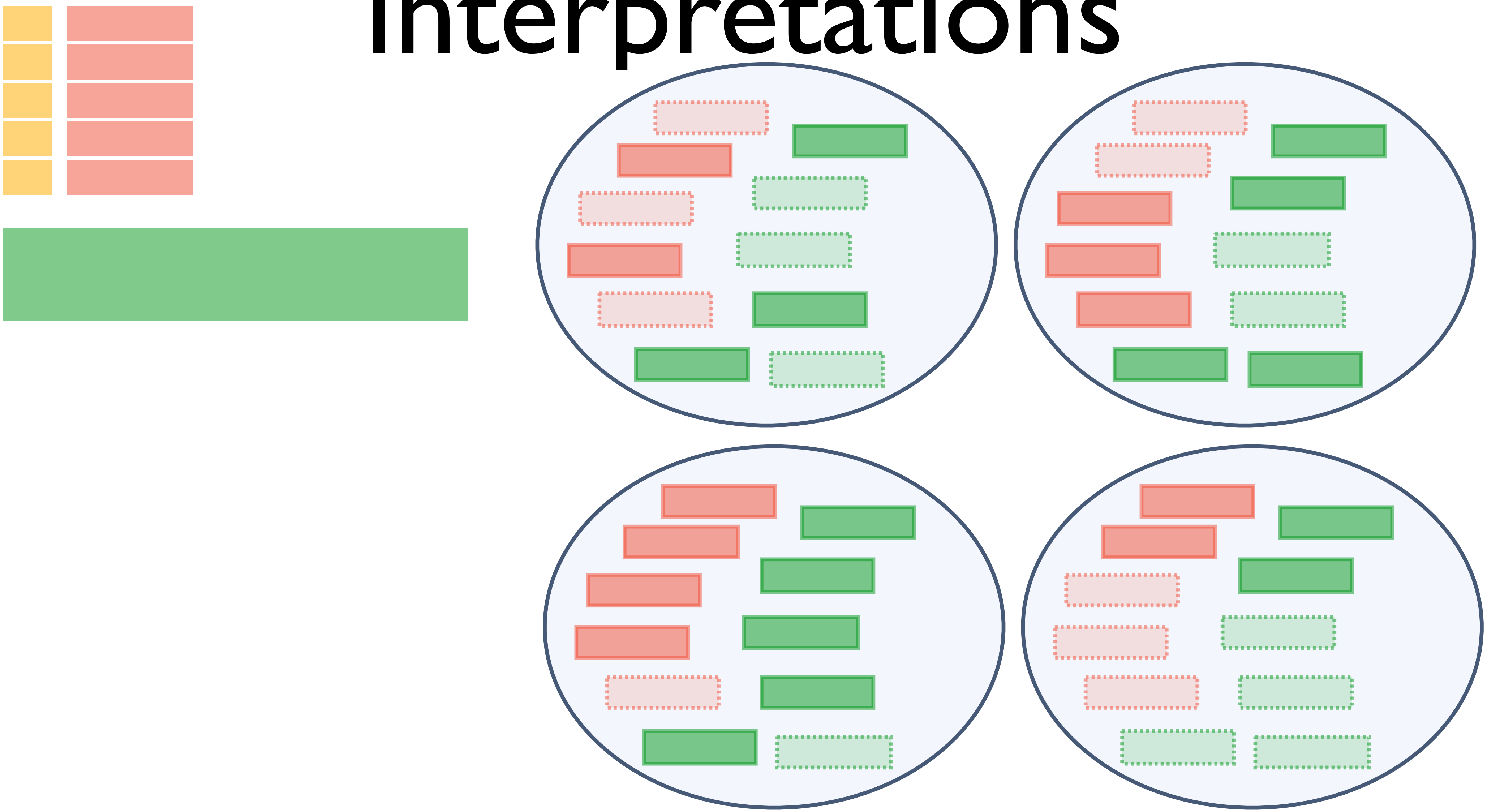
e.g., webpage classification model

for each *CLASS1*, *CLASS2* and each *WORD*

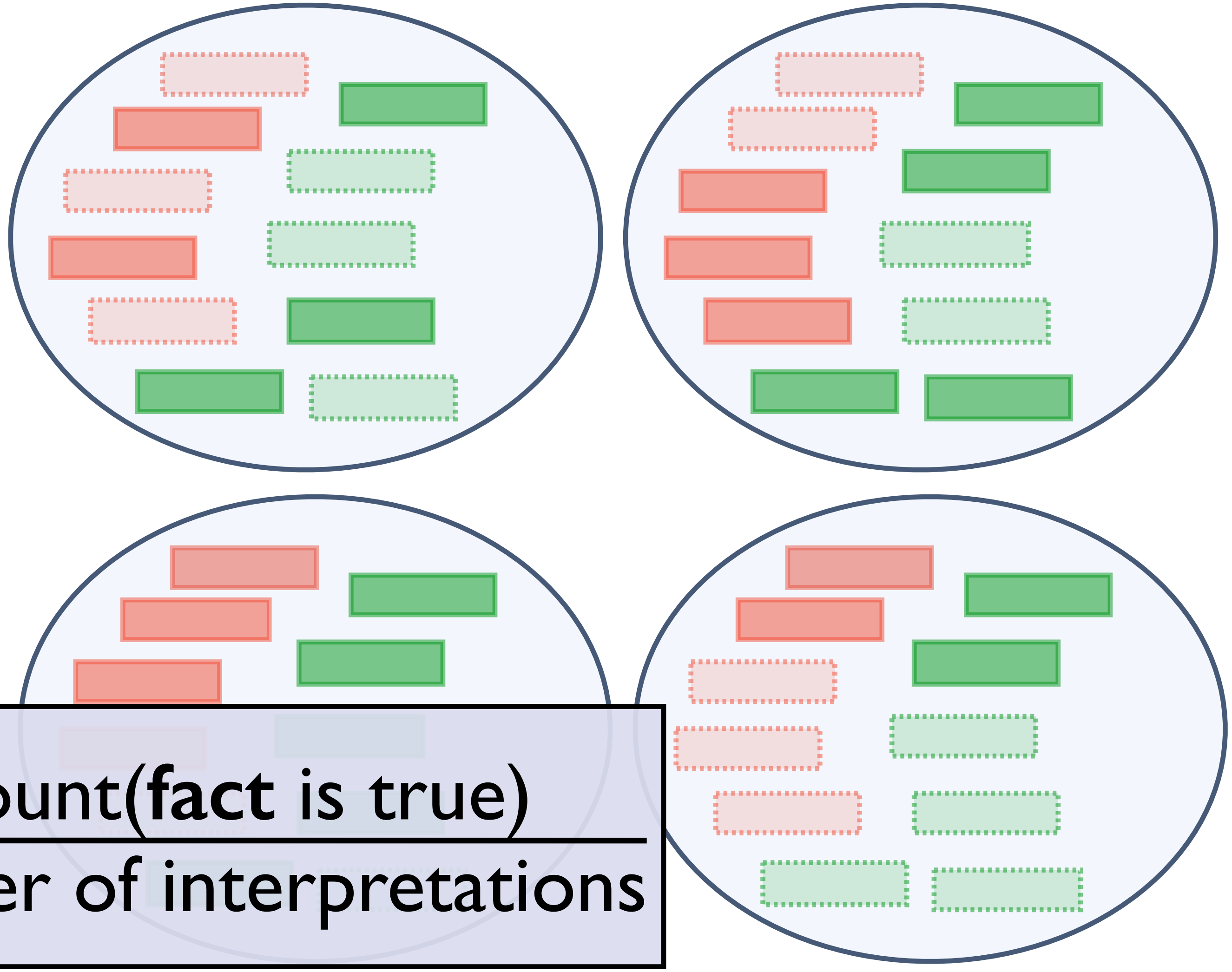
```
?? :: link_class(Source,Target,CLASS1,CLASS2).
?? :: word_class(WORD,CLASS).
```

```
class(Page,C)    :-   has_word(Page,W), word_class(W,C).

class(Page,C)          :-   links_to(OtherPage,Page),
     class(OtherPage,OtherClass),
                            link_class(OtherPage,Page,OtherClass,C).
```

Sampling Interpretations

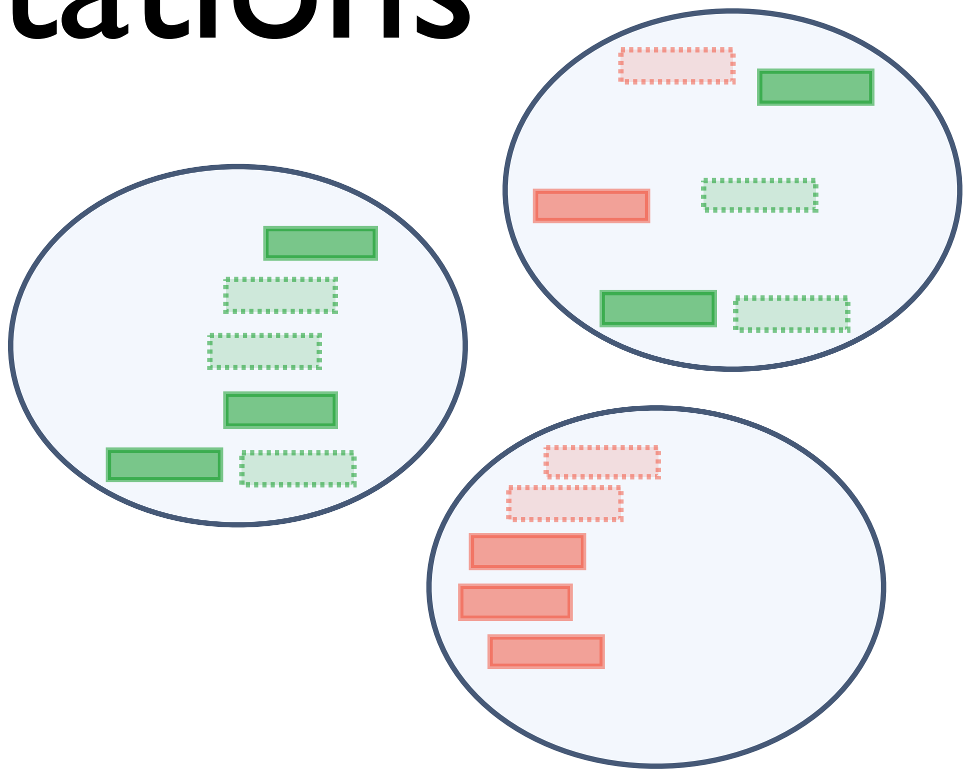


Parameter Estimation



$$p(\textbf{fact}) = \frac{\text{count}(\textbf{fact} \text{ is true})}{\text{Number of interpretations}}$$

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]

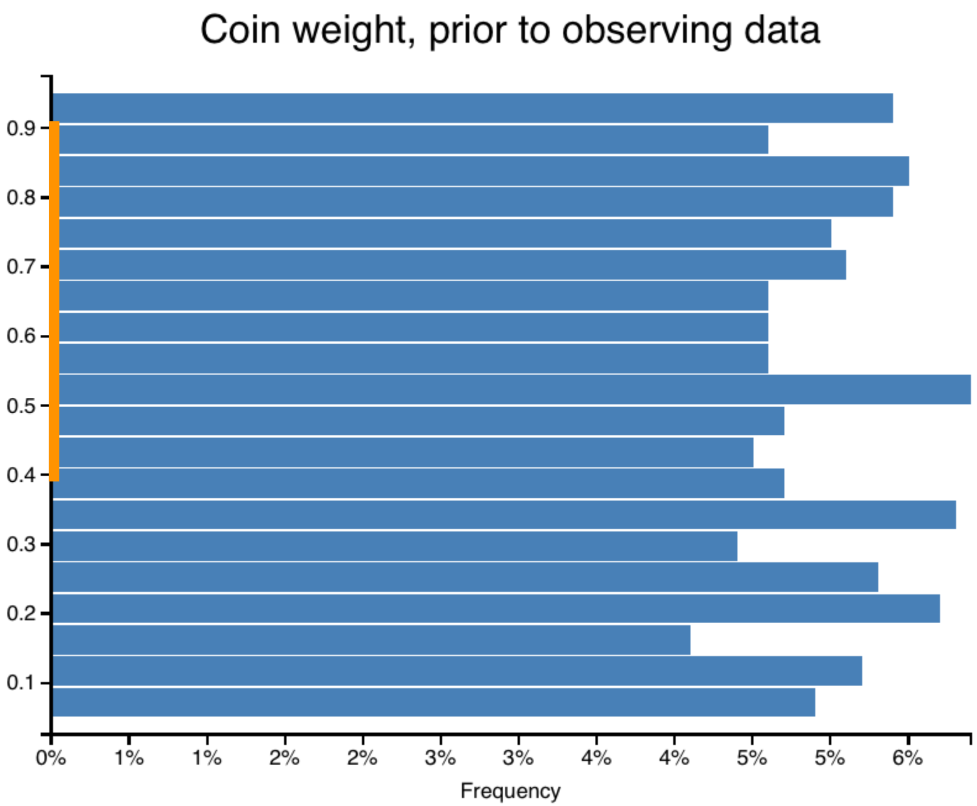
Bayesian Parameter Learning

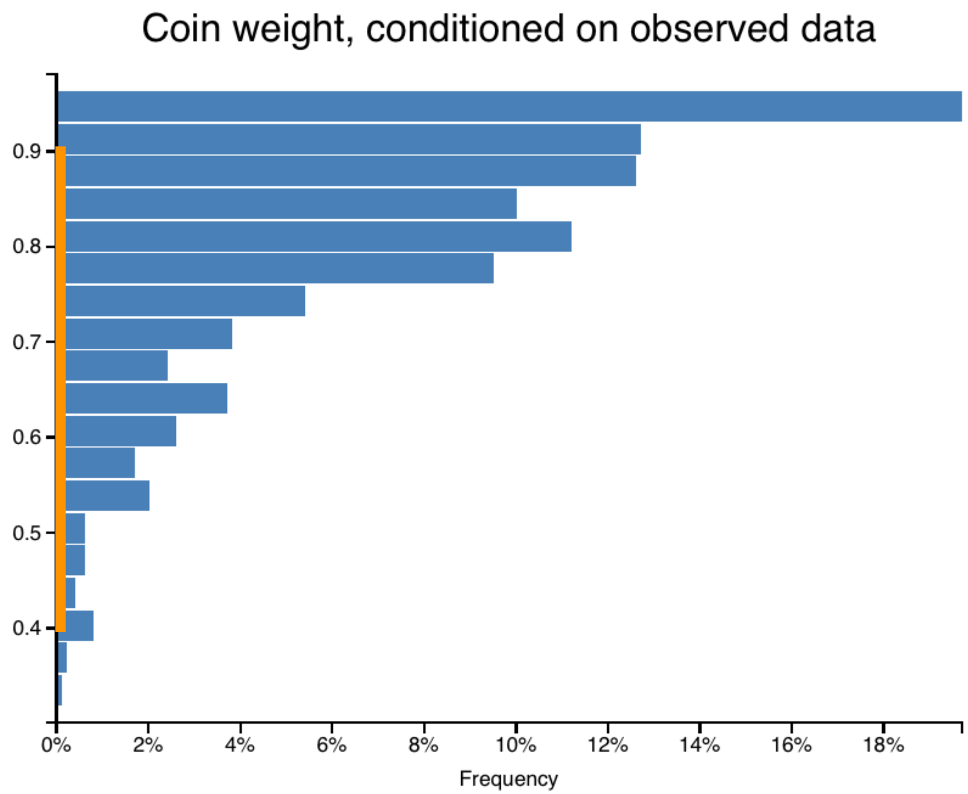
- Learning as inference (e.g., Church)
- Prior distributions for parameters
- Given data, find most likely parameter values

Example

[from probmods.org]

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

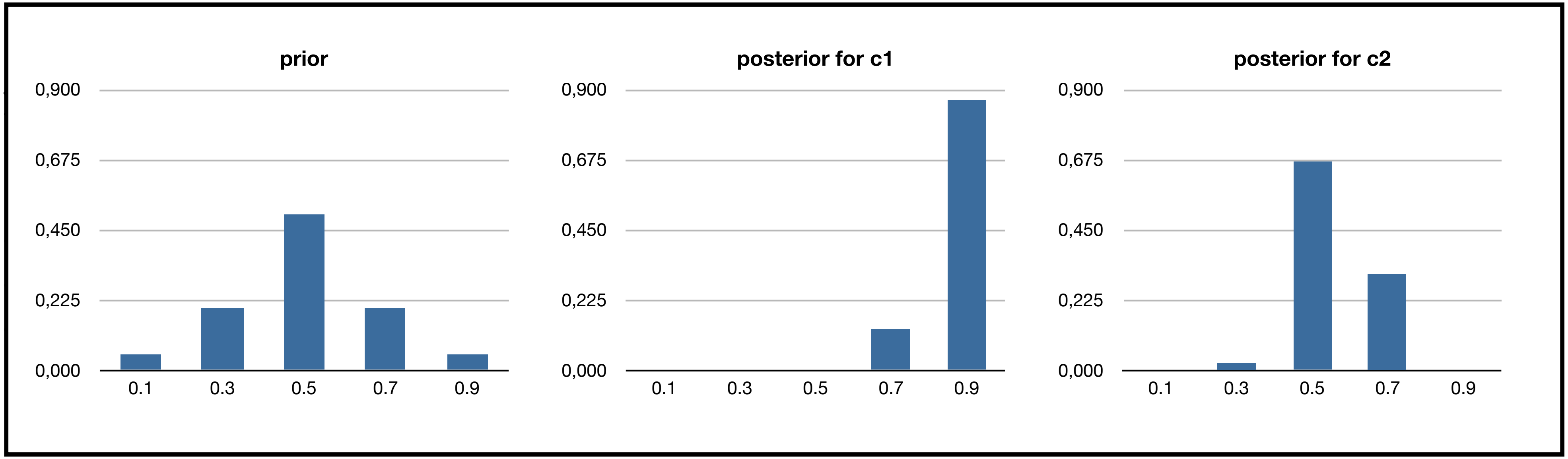




ProbLog Example

prior

```
0.05::weight(C,0.1); 0.2::weight(C,0.3); 0.5::weight(C,0.5);
                 0.2::weight(C,0.7); 0.05::weight(C,0.9) <- coin(C).
```




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

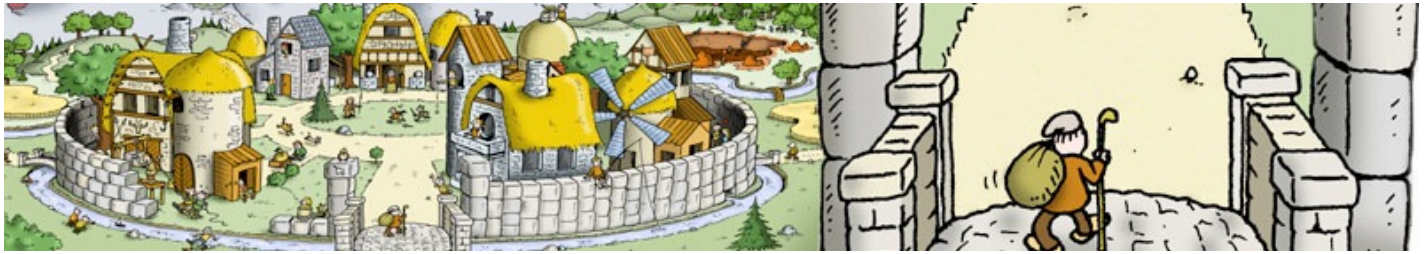
ask for posterior

```
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,h,t,t,h,t,t,h]),true).
```

data

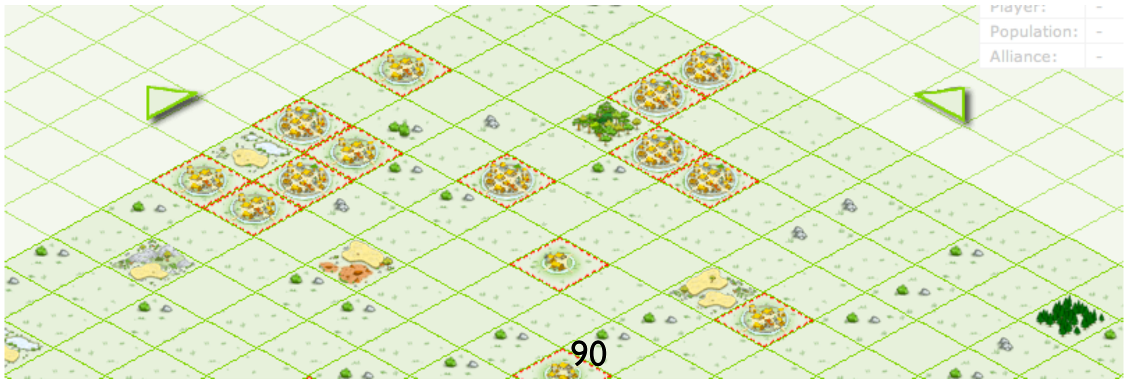
Part IV: Dynamics

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

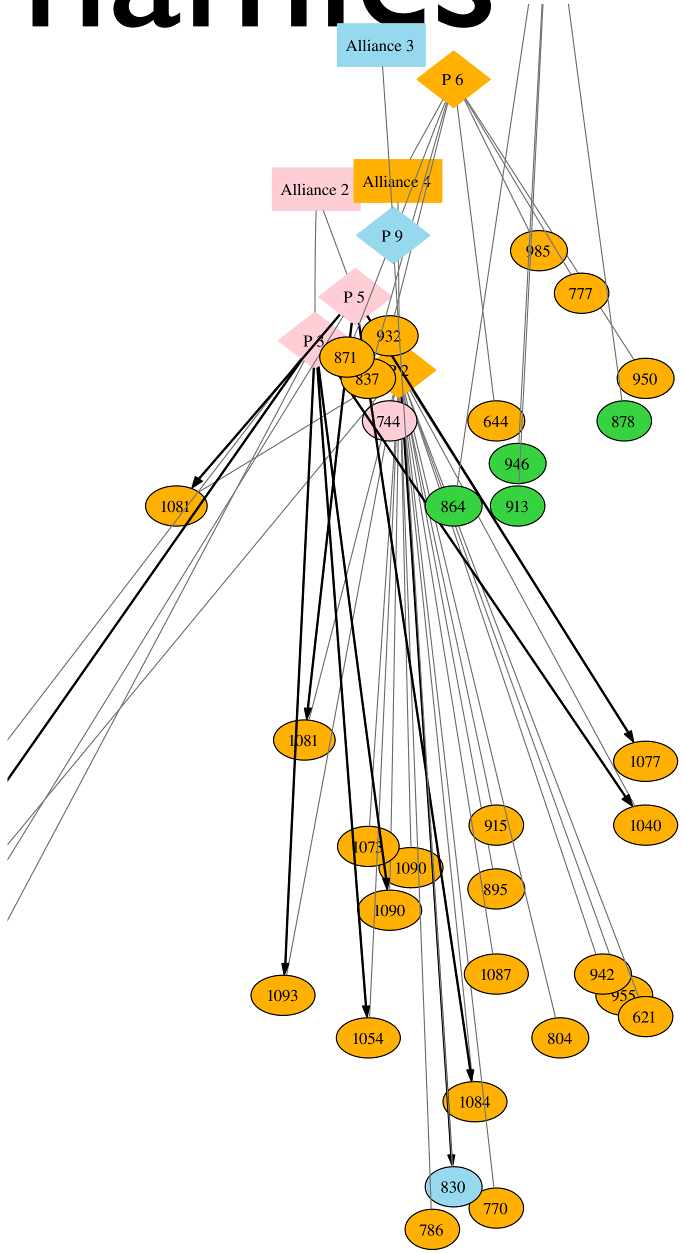
Fragment of world with

~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

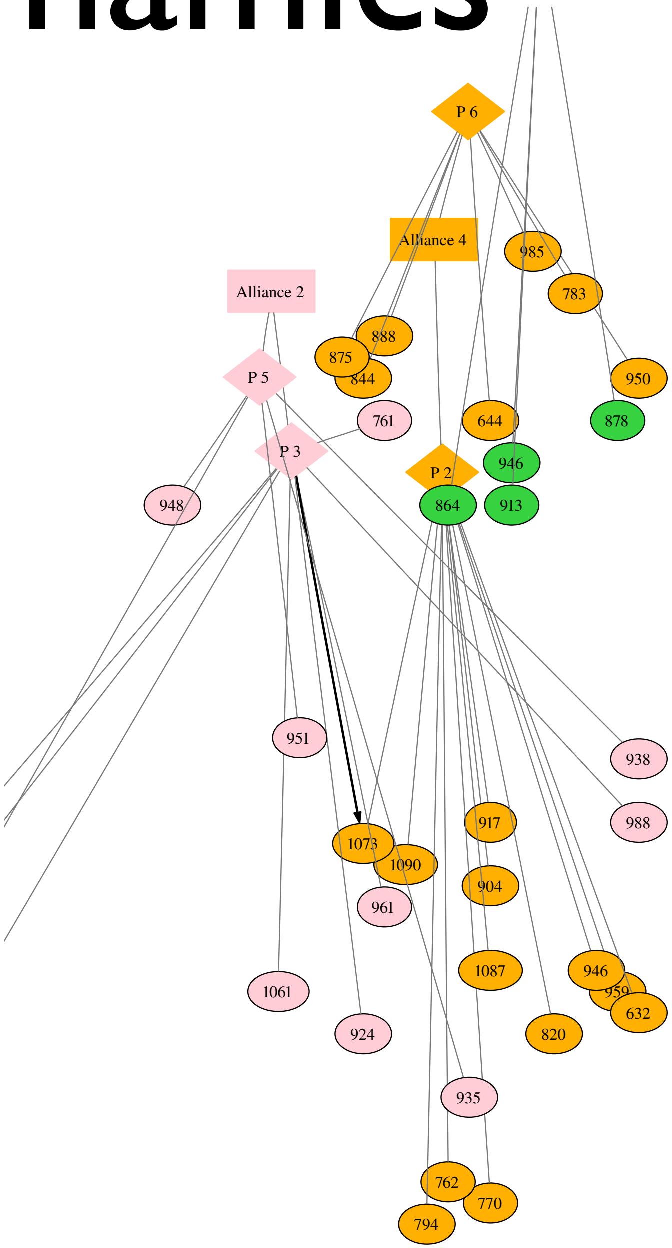
Fragment of world with

~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

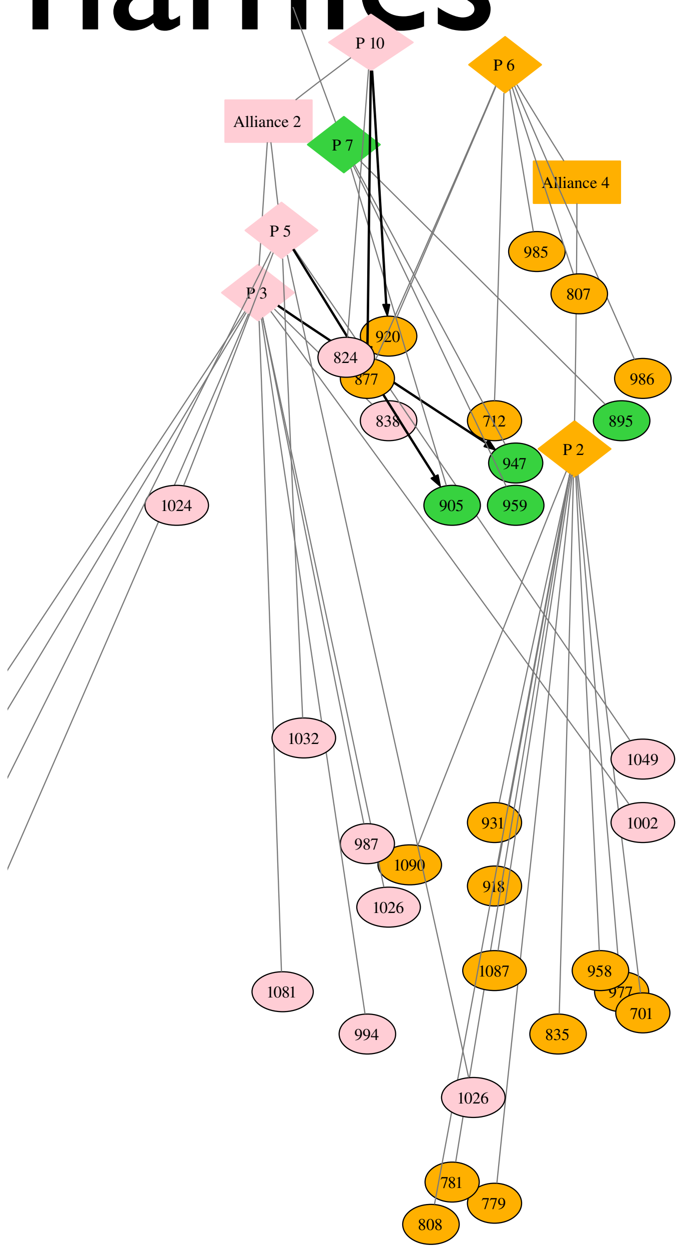
Fragment of world with

~10 alliances
~200 players
~600 cities

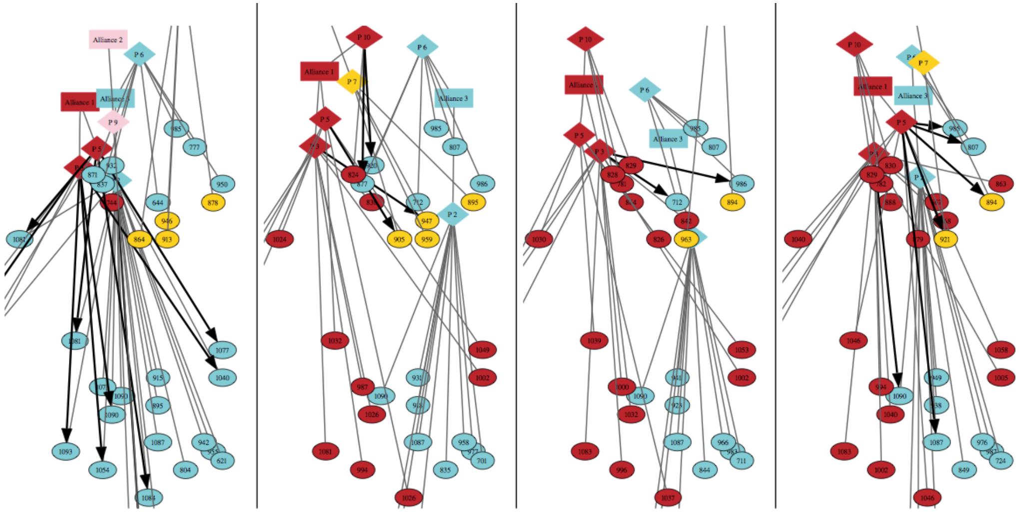
alliances color-coded

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?

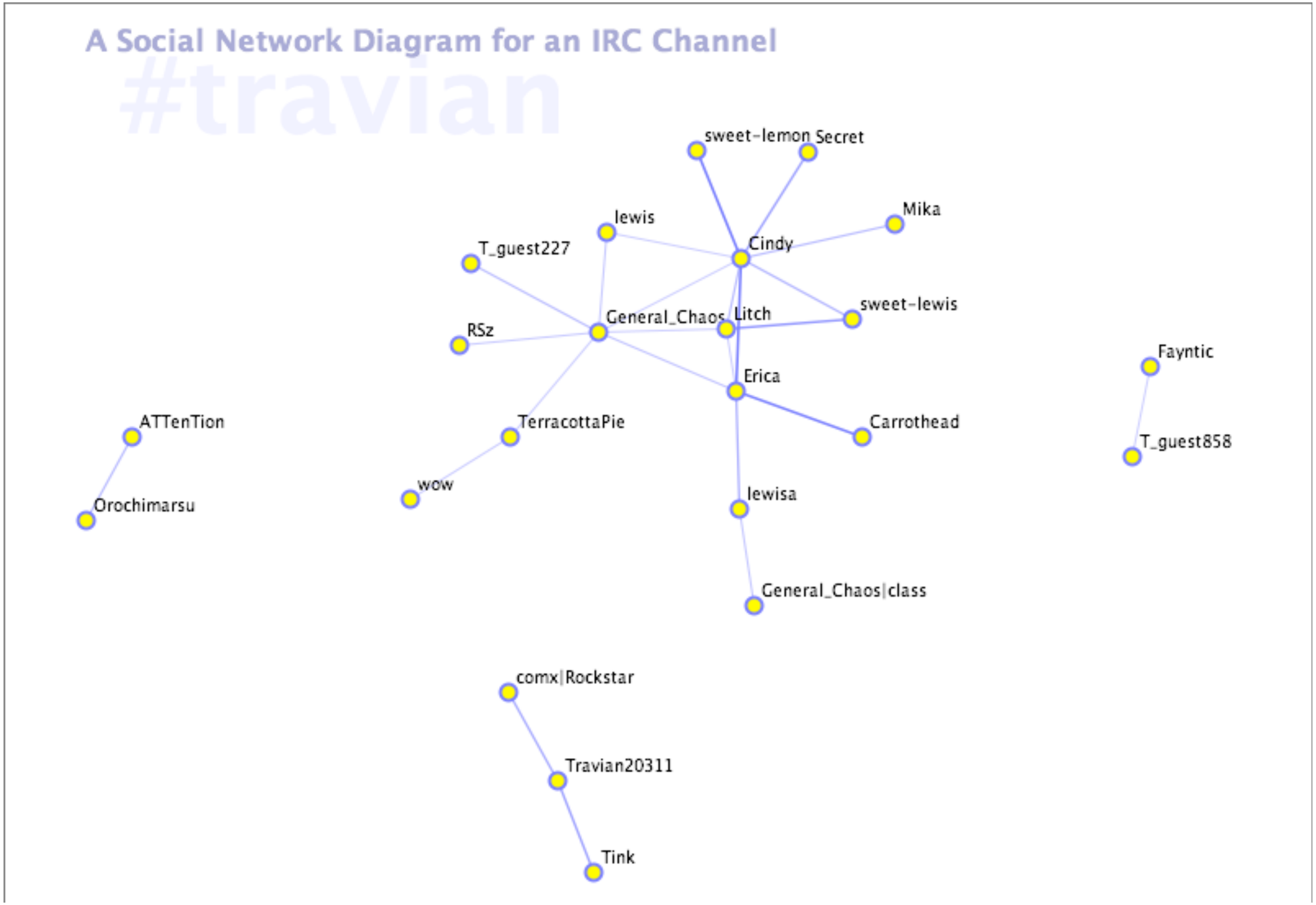
one of the **effects** holds at time T+1

```
0.4::conquest(Attacker,C); 0.6::nil <-
        city(C,Owner),city(C2,Attacker),close(C,C2).
```

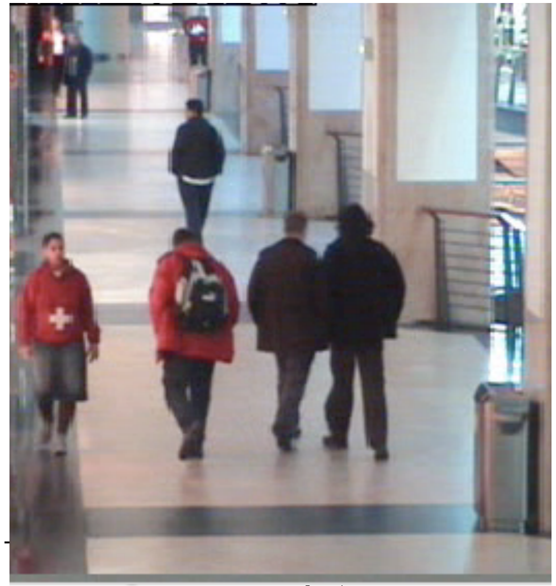
if **cause** holds at time T

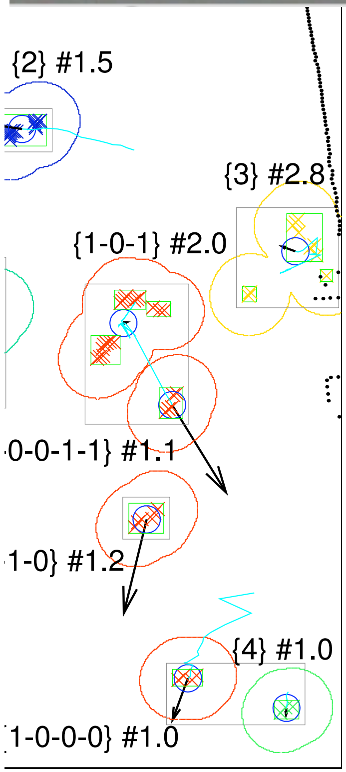
[Thon et al, MLJ 11]

Social Network of Chats

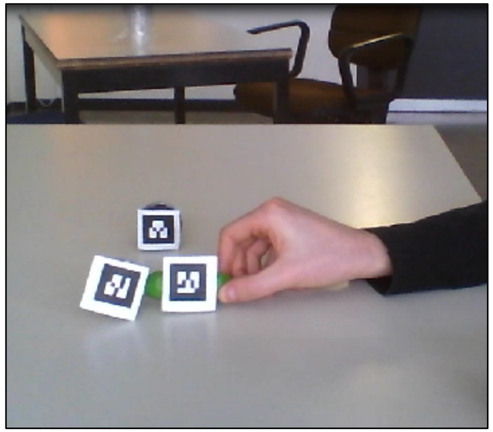


Relational Tracking





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



Relational State Estimation over Time

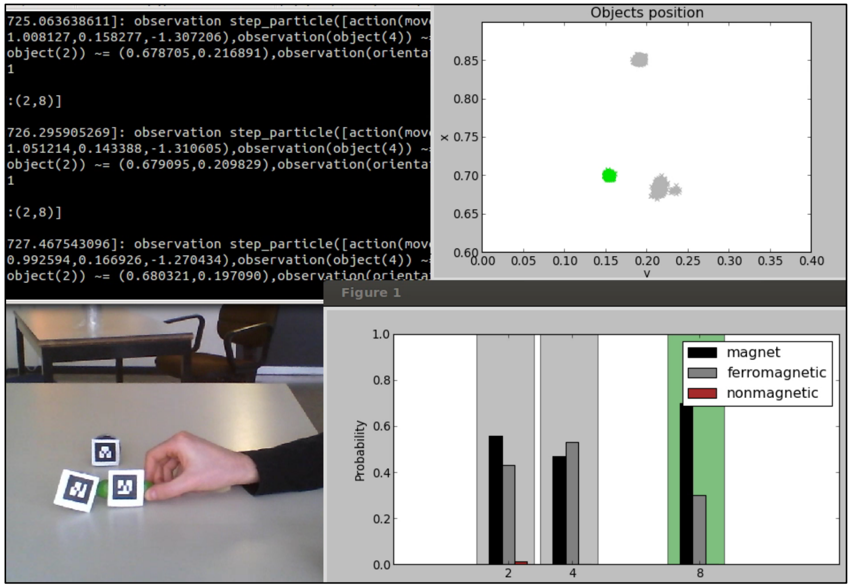
Magnetism scenario

- object tracking
- category estimation from interactions

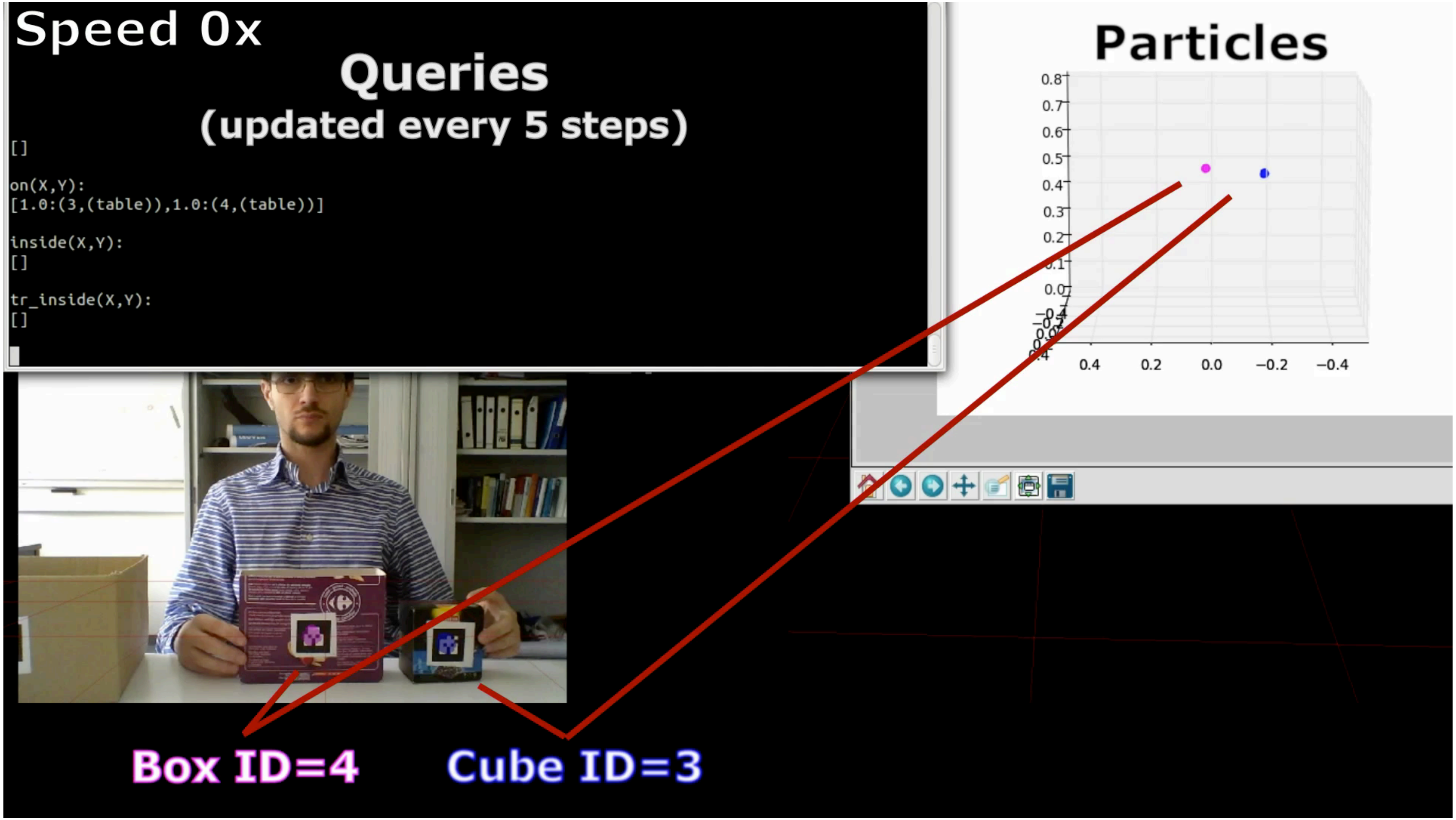
(a) cube on the box (b) cube inside the box (c) rotated box on a beige box (d) cube and box inside the beige box

Box scenario

- object tracking even when invisible
- estimate spatial relations



[Nitti et al, IROS 13]



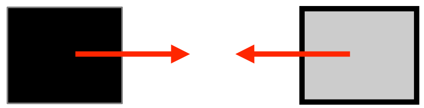
Speed 0x
Queries
(updated every 5 steps)
[]
on(X,Y):
[1.0:(3,(table)),1.0:(4,(table))]
inside(X,Y):
[]
tr_inside(X,Y):
[]
Particles
Box ID=4
Cube ID=3

IROS 13

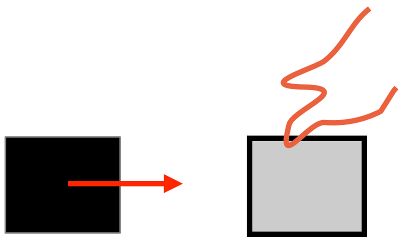
Magnetic scenario

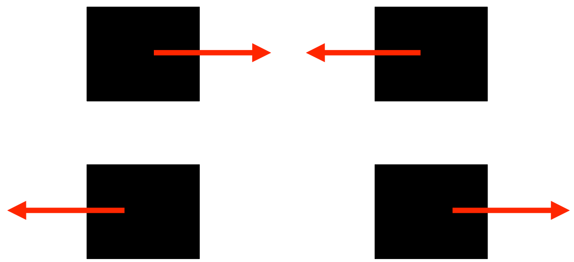
- 3 object types: magnetic, ferromagnetic, nonmagnetic

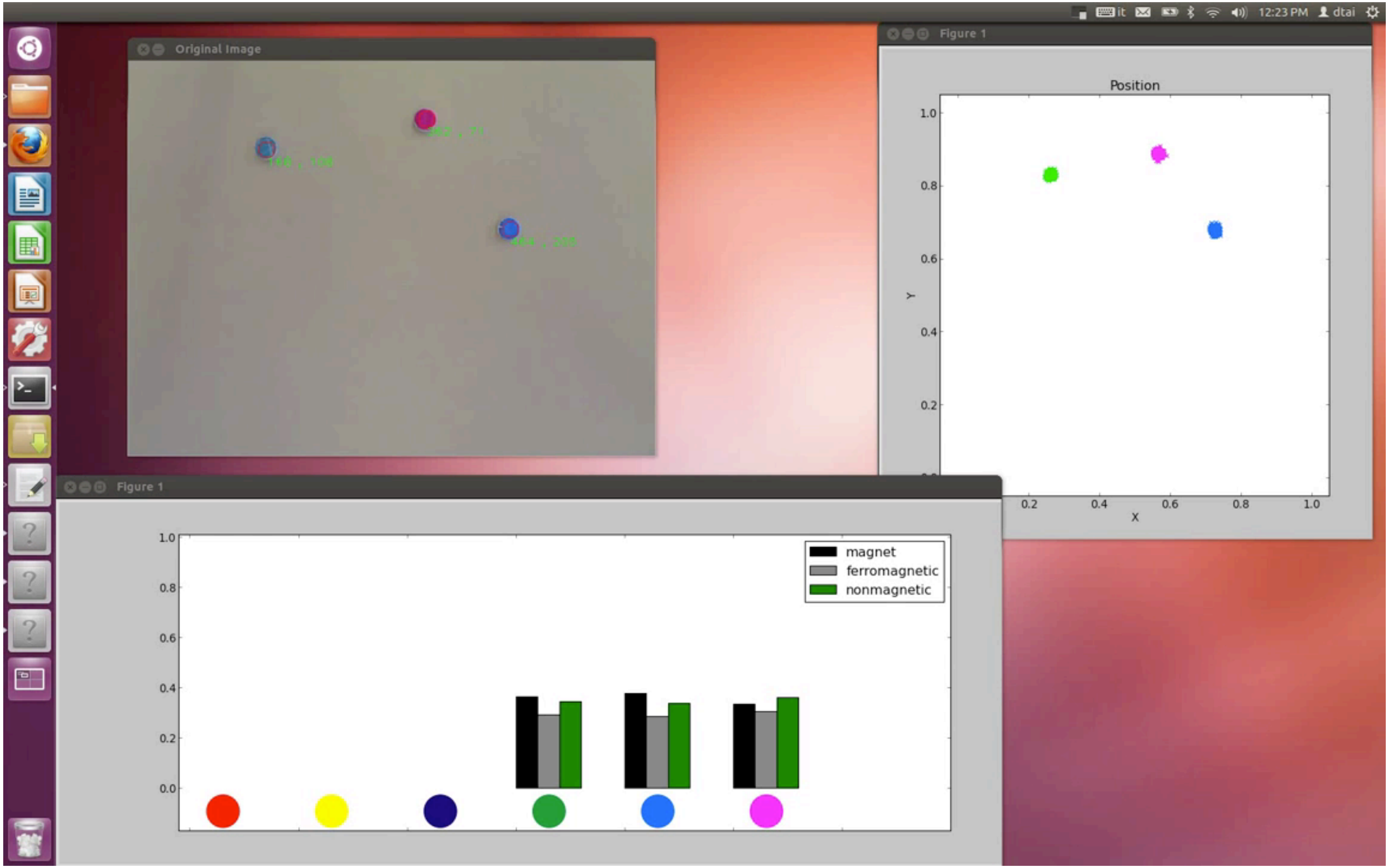
- Nonmagnetic objects do not interact
- A magnet and a ferromagnetic object attract each other



- Magnetic force that depends on the distance
- If an object is held magnetic force is compensated.






12:23 PM
dtai
Original Image
Figure 1
Position
Y
X
magnet
ferromagnetic
nonmagnetic

Magnetic scenario

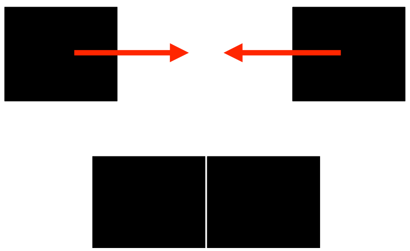
- 3 object types: magnetic, ferromagnetic, nonmagnetic

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

- 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



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

$$\text{pos}(A)_{t+1} \sim \text{gaussian}(\text{pos}(A)_t,\text{Cov}) \leftarrow \text{not}(\ \text{attraction}(A,B)\ ).$$

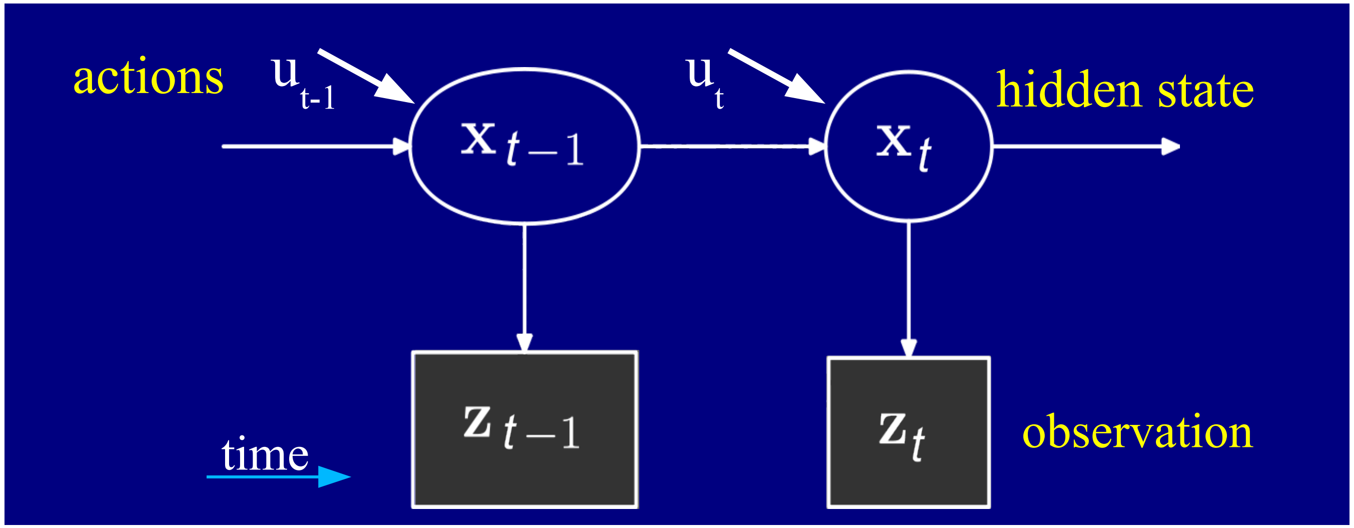
Dynamic Distributional Clauses

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

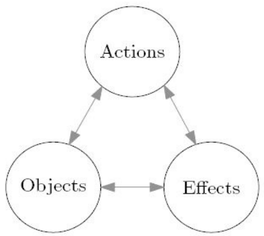
- Online parameter learning [Nitti, ICRA 2014]
- Integration with planning
- Larger Experiments
- Applications in robotics (also to learn affordances)

Take-away message

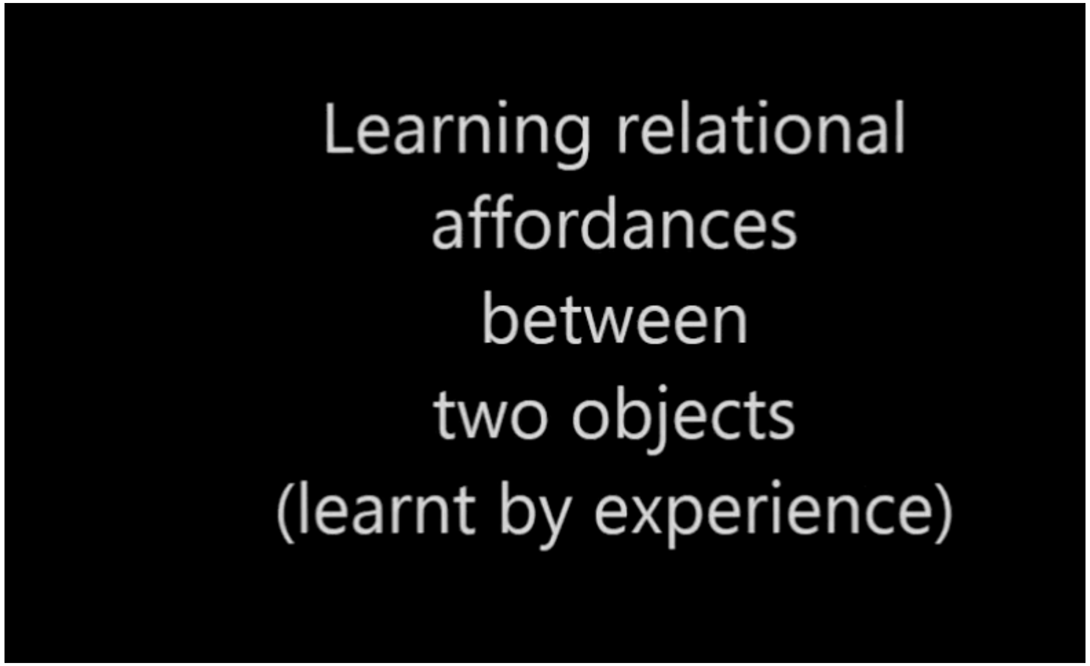
- Probabilistic rules also apply to dynamic environments
- Topic of ongoing work
- Scalability is a challenge

Learning relational affordances

Learn probabilistic model



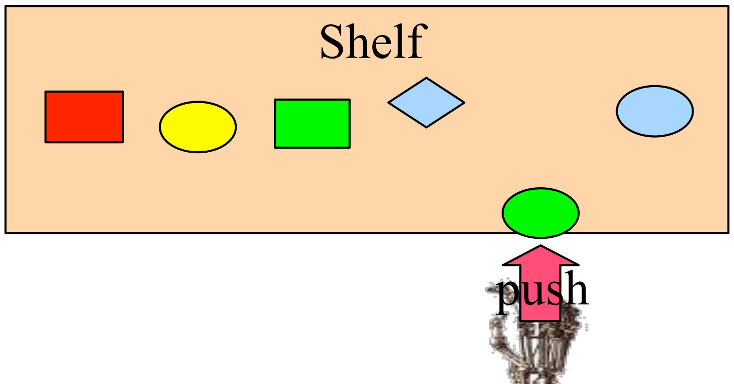
Inputs	Outputs	Function
(O, A)	E	Effect prediction
(O, E)	A	Action recognition/planning
(A, E)	O	Object recognition/selection

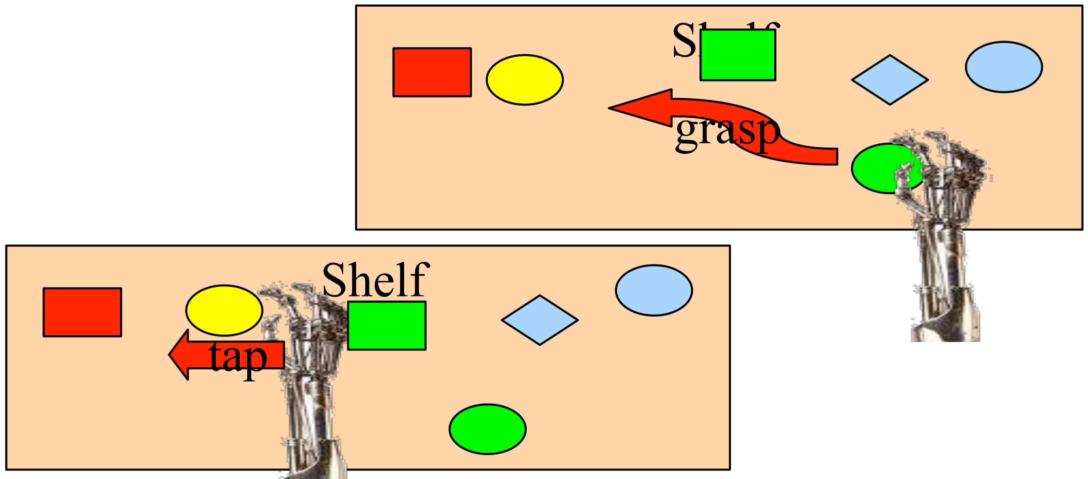


From two object interactions

Generalize to N

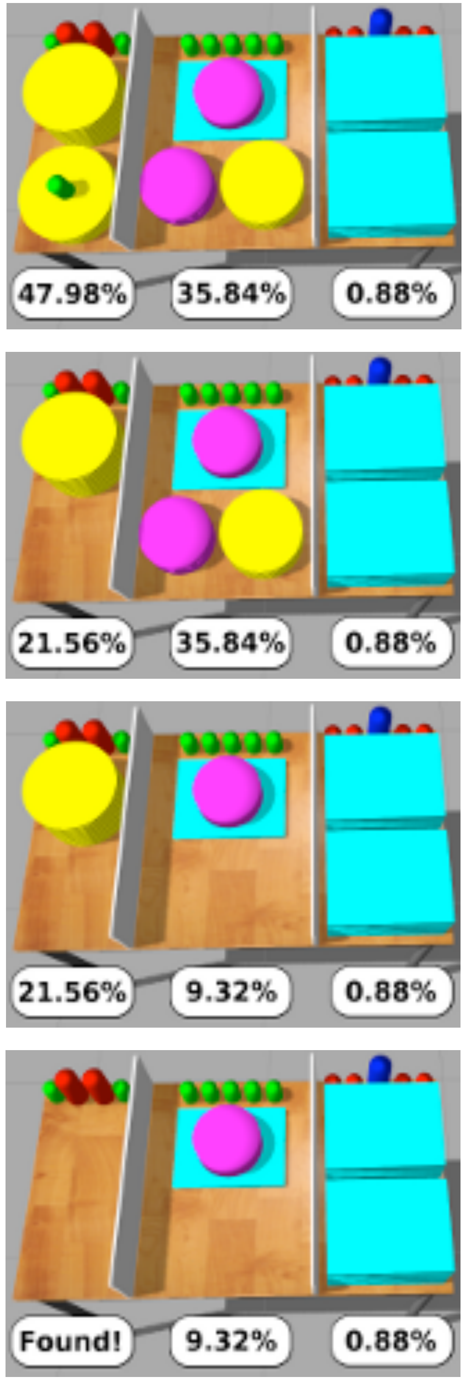
Moldovan et al. ICRA 12, 13, 14





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

CAVIAR-INRIA human activity dataset

28 videos
≈ 26.500 frames

- 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:

$$\begin{gathered} initiatedAt(fighting(P_1, P_2) = true, T) \leftarrow \\ happensAt(abrupt(P_1), T), \\ holdsAt(close(P_1, P_2, 44) = true, T), \\ not\ happensAt(inactive(P_2), T). \end{gathered}$$

[Skarlatidis et al, TPLP 13]

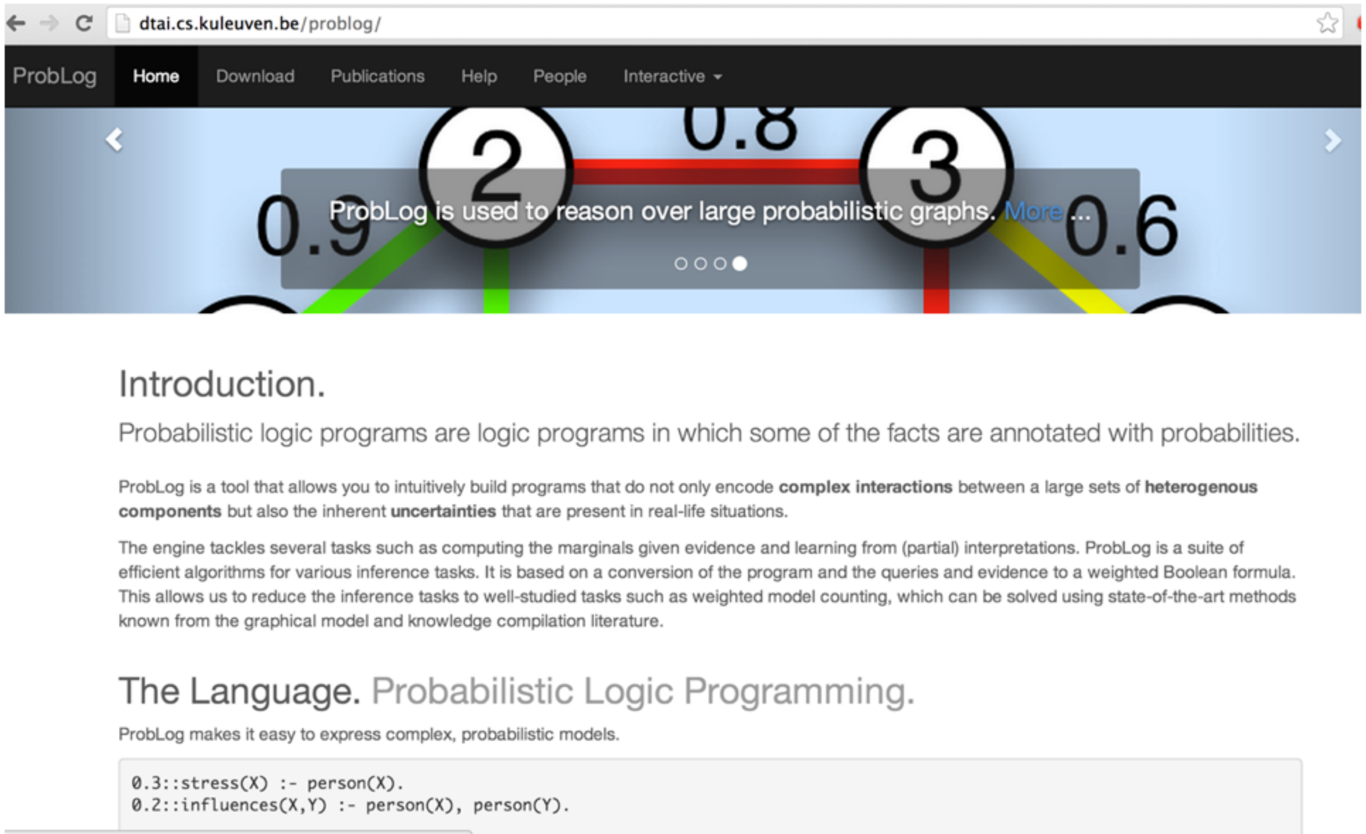
Conclusions

- 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 ?

Maurice Bruynooghe
Bart Demoen
Anton Dries
Daan Fierens
Jason Filippou
Bernd Gutmann
Manfred Jaeger
Gerda Janssens
Kristian Kersting
Angelika Kimmig
Theofrastos Mantadelis
Wannes Meert
Bogdan Moldovan
Siegfried Nijssen
Davide Nitti
Joris Renkens
Kate Revoredo
Ricardo Rocha
Vitor Santos Costa
Dimitar Shterionov
Ingo Thon
Hannu Toivonen
Guy Van den Broeck
Mathias Verbeke
Jonas Vlasselaer

Thanks !

http://dtai.cs.kuleuven.be/problog



PLP Systems

- **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

Bach SH, Broecheler M, Getoor L, O'Leary DP (2012) Scaling MPE inference for constrained continuous Markov random fields with consensus optimization. In: Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS-12)

Broecheler M, Mihalkova L, Getoor L (2010) Probabilistic similarity logic. In: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI-10)

Bryant RE (1986) Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers 35(8):677–691

Cohen SB, Simmons RJ, Smith NA (2008) Dynamic programming algorithms as products of weighted logic programs. In: Proceedings of the 24th International Conference on Logic Programming (ICLP-08)

Cussens J (2001) Parameter estimation in stochastic logic programs. Machine Learning 44(3):245–271

De Maeyer D, Renkens J, Cloots L, De Raedt L, Marchal K (2013) Phenetic: network-based interpretation of unstructured gene lists in e. coli. Molecular BioSystems 9(7):1594–1603

De Raedt L, Kimmig A (2013) Probabilistic programming concepts. CoRR abs/1312.4328

De Raedt L, Kimmig A, Toivonen H (2007) ProbLog: A probabilistic Prolog and its application in link discovery. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)

De Raedt L, Frasconi P, Kersting K, Muggleton S (eds) (2008) Probabilistic Inductive Logic Programming — Theory and Applications, Lecture Notes in Artificial Intelligence, vol 4911. Springer

Eisner J, Goldlust E, Smith N (2005) Compiling Comp Ling: Weighted dynamic programming and the Dyna language. In: Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-05)

Fierens D, Blockeel H, Bruynooghe M, Ramon J (2005) Logical Bayesian networks and their relation to other probabilistic logical models. In: Proceedings of the 15th International Conference on Inductive Logic Programming (ILP-05)

Fierens D, Van den Broeck G, Bruynooghe M, De Raedt L (2012) Constraints for probabilistic logic programming. In: Proceedings of the NIPS Probabilistic Programming Workshop

Fierens D, Van den Broeck G, Renkens J, Shterionov D, Gutmann B, Thon I, Janssens G, De Raedt L (2014) Inference and learning in probabilistic logic programs using weighted Boolean formulas. Theory and Practice of Logic Programming (TPLP) FirstView

Getoor L, Friedman N, Koller D, Pfeffer A, Taskar B (2007) Probabilistic relational models. In: Getoor L, Taskar B (eds) An Introduction to Statistical Relational Learning, MIT Press, pp 129–174

Goodman N, Mansinghka VK, Roy DM, Bonawitz K, Tenenbaum JB (2008) Church: a language for generative models. In: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI-08)

Gutmann B, Thon I, De Raedt L (2011a) Learning the parameters of probabilistic logic programs from interpretations. In: Proceedings of the 22nd European Conference on Machine Learning (ECML-11)

Gutmann B, Thon I, Kimmig A, Bruynooghe M, De Raedt L (2011b) The magic of logical inference in probabilistic programming. Theory and Practice of Logic Programming (TPLP) 11((4–5)):663–680

Huang B, Kimmig A, Getoor L, Golbeck J (2013) A flexible framework for probabilistic models of social trust. In: Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction (SBP-13)

Jaeger M (2002) Relational Bayesian networks: A survey. Linköping Electronic Articles in Computer and Information Science 7(015)

Kersting K, Raedt LD (2001) Bayesian logic programs. CoRR cs.AI/0111058

Kimmig A, Van den Broeck G, De Raedt L (2011a) An algebraic Prolog for reasoning about possible worlds. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-11)

Kimmig A, Demoen B, De Raedt L, Santos Costa V, Rocha R (2011b) On the implementation of the probabilistic logic programming language ProbLog. Theory and Practice of Logic Programming (TPLP) 11:235–262

Koller D, Pfeffer A (1998) Probabilistic frame-based systems. In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)

McCallum A, Schultz K, Singh S (2009) FACTORIE: Probabilistic programming via imperatively defined factor graphs. In: Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS-09)

Milch B, Marthi B, Russell SJ, Sontag D, Ong DL, Kolobov A (2005) Blog: Probabilistic models with unknown objects. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)

Moldovan B, De Raedt L (2014) Occluded object search by relational affordances. In: IEEE International Conference on Robotics and Automation (ICRA-14)

Moldovan B, Moreno P, van Otterlo M, Santos-Victor J, De Raedt L (2012) Learning relational affordance models for robots in multi-object manipulation tasks. In: IEEE International Conference on Robotics and Automation (ICRA-12)

Muggleton S (1996) Stochastic logic programs. In: De Raedt L (ed) Advances in Inductive Logic Programming, IOS Press, pp 254–264

Nitti D, De Laet T, De Raedt L (2013) A particle filter for hybrid relational domains. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-13)

Nitti D, De Laet T, De Raedt L (2014) Relational object tracking and learning. In: IEEE International Conference on Robotics and Automation (ICRA), June 2014

Pfeffer A (2001) IBAL: A probabilistic rational programming language. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)

Pfeffer A (2009) Figaro: An object-oriented probabilistic programming language. Tech. rep., Charles River Analytics

Poole D (2003) First-order probabilistic inference. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)

Richardson M, Domingos P (2006) Markov logic networks. Machine Learning 62(1-2):107–136

Santos Costa V, Page D, Cussens J (2008) CLP(*BN*): Constraint logic programming for probabilistic knowledge. In: De Raedt et al (2008), pp 156–188

Sato T (1995) A statistical learning method for logic programs with distribution semantics. In: Proceedings of the 12th International Conference on Logic Programming (ICLP-95)

Sato T, Kameya Y (2001) Parameter learning of logic programs for symbolic-statistical modeling. J Artif Intell Res (JAIR) 15:391–454

Sato T, Kameya Y (2008) New advances in logic-based probabilistic modeling by prism. In: Probabilistic Inductive Logic Programming, pp 118–155

Skarlatidis A, Artikis A, Filiopou J, Paliouras G (2014) A probabilistic logic programming event calculus. Theory and Practice of Logic Programming (TPLP) FirstView

Suciu D, Olteanu D, Ré C, Koch C (2011) Probabilistic Databases. Synthesis Lectures on Data Management, Morgan & Claypool Publishers

Taskar B, Abbeel P, Koller D (2002) Discriminative probabilistic models for relational data. In: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02)

Thon I, Landwehr N, De Raedt L (2008) A simple model for sequences of relational state descriptions. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD-08)

Thon I, Landwehr N, De Raedt L (2011) Stochastic relational processes: Efficient inference and applications. Machine Learning 82(2):239–272

Van den Broeck G, Thon I, van Otterlo M, De Raedt L (2010) DTProbLog: A decision-theoretic probabilistic Prolog. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10)

Van den Broeck G, Taghipour N, Meert W, Davis J, De Raedt L (2011) Lifted probabilistic inference by first-order knowledge compilation. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)

Vennekens J, Verbaeten S, Bruynooghe M (2004) Logic programs with annotated disjunctions. In: Proceedings of the 20th International Conference on Logic Programming (ICLP-04)

Vennekens J, Denecker M, Bruynooghe M (2009) CP-logic: A language of causal probabilistic events and its relation to logic programming. Theory and Practice of Logic Programming (TPLP) 9(3):245–308

Wang WY, Mazaitis K, Cohen WW (2013) Programming with personalized pagerank: a locally groundable first-order probabilistic logic. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM-13)