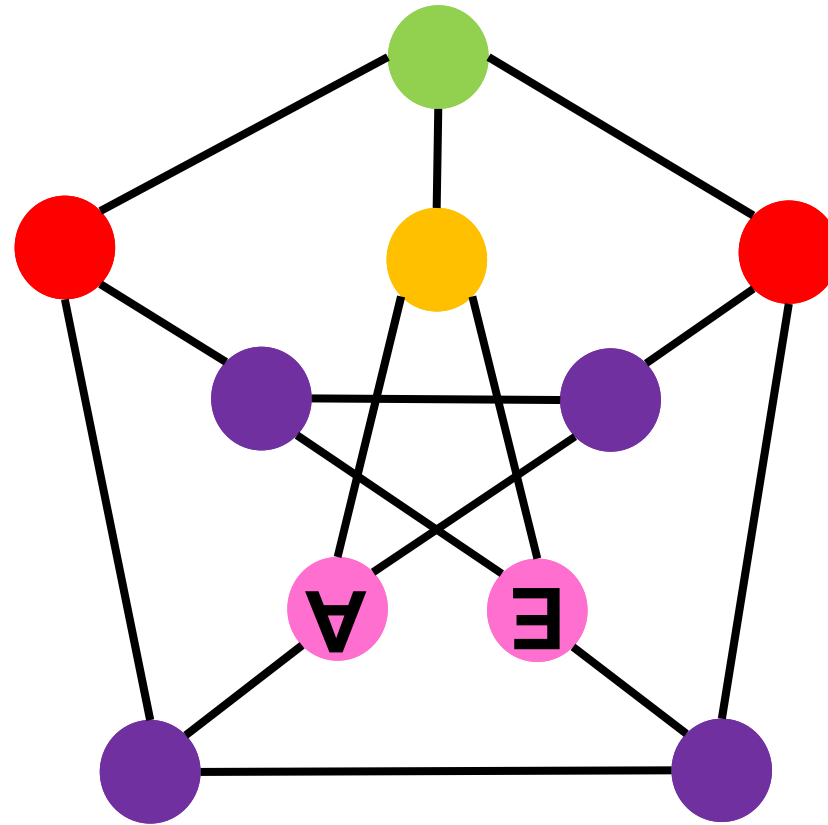




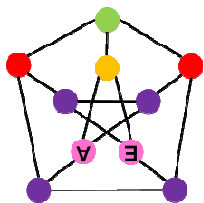
Kristian Kersting



# Statistical Relational Learning/AI\*

\* Many thanks to Luc De Raedt, Rodrigo de Salvo Braz, Pedro Domingos, Kurt Driessens, Oren Etzioni, Lise Getoor, Daphne Koller, Stanley Kok, Brian Milch, Sriraam Natarajan, Avi Pfeffer, Stuart Russell, Scott Sanner, Jude Shavlik, and many others for making their slides publically available.

# Rorschach Test



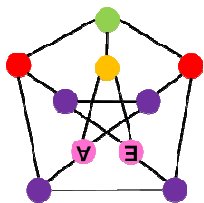
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



# Etzioni's Rorschach Test for Computer Scientists



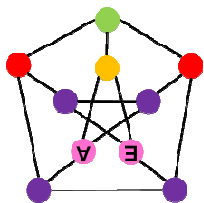
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



# Moore's Law?



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010

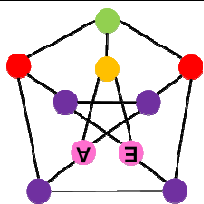


**Fraunhofer**





# Storage Capacity?



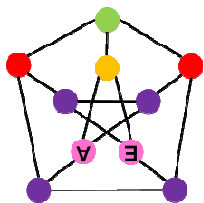
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**

**IAIS universität bonn**  
RHEINISCHE FRIEDRICH-WILHELMS-  
UNIVERSITÄT

# Number of Facebook Users?



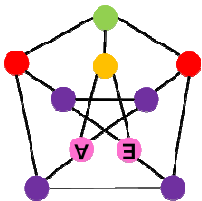
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



# Number of Scientific Publications?



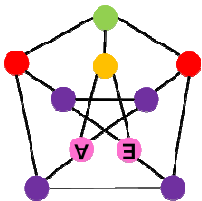
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**

**IAIS universität bonn**  
RHEINISCHE FRIEDRICH-WILHELMS-  
UNIVERSITÄT

# Number of Web Pages?



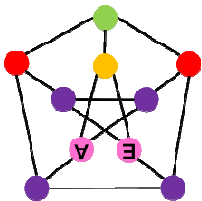
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



# Number of Actions?



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



# Computing 2020: Science in an Exponential World



“The amount of scientific data is doubling every year”

[Szalay, Gray; *Nature* 440, 413-414 (23 March 2006) ]

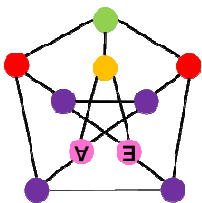
How to deal with millions of images ?

How to deal with millions of inter-related research papers ?

How to accumulate general knowledge automatically from the Web ?

How to deal with billions of shared users' perceptions stored at massive scale ?

How to realize the vision of social search?



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



# Machine Learning in an Exponential World

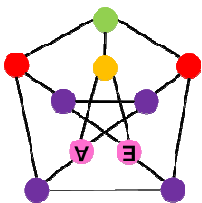
ML = Structured Data + Model + Reasoning



- Real world is structured in terms of objects and relations
  - Relational knowledge can reveal additional correlations between variables of interest . Abstraction allows one to compactly model general knowledge and to move to complex inference

[Fergus et al. PAMI 30(11) 2008; Halevy et al., IEEE Intelligent Systems, 24 2009]

- Most effort has gone into the modeling part
- How much can the data itself help us to solve a problem?



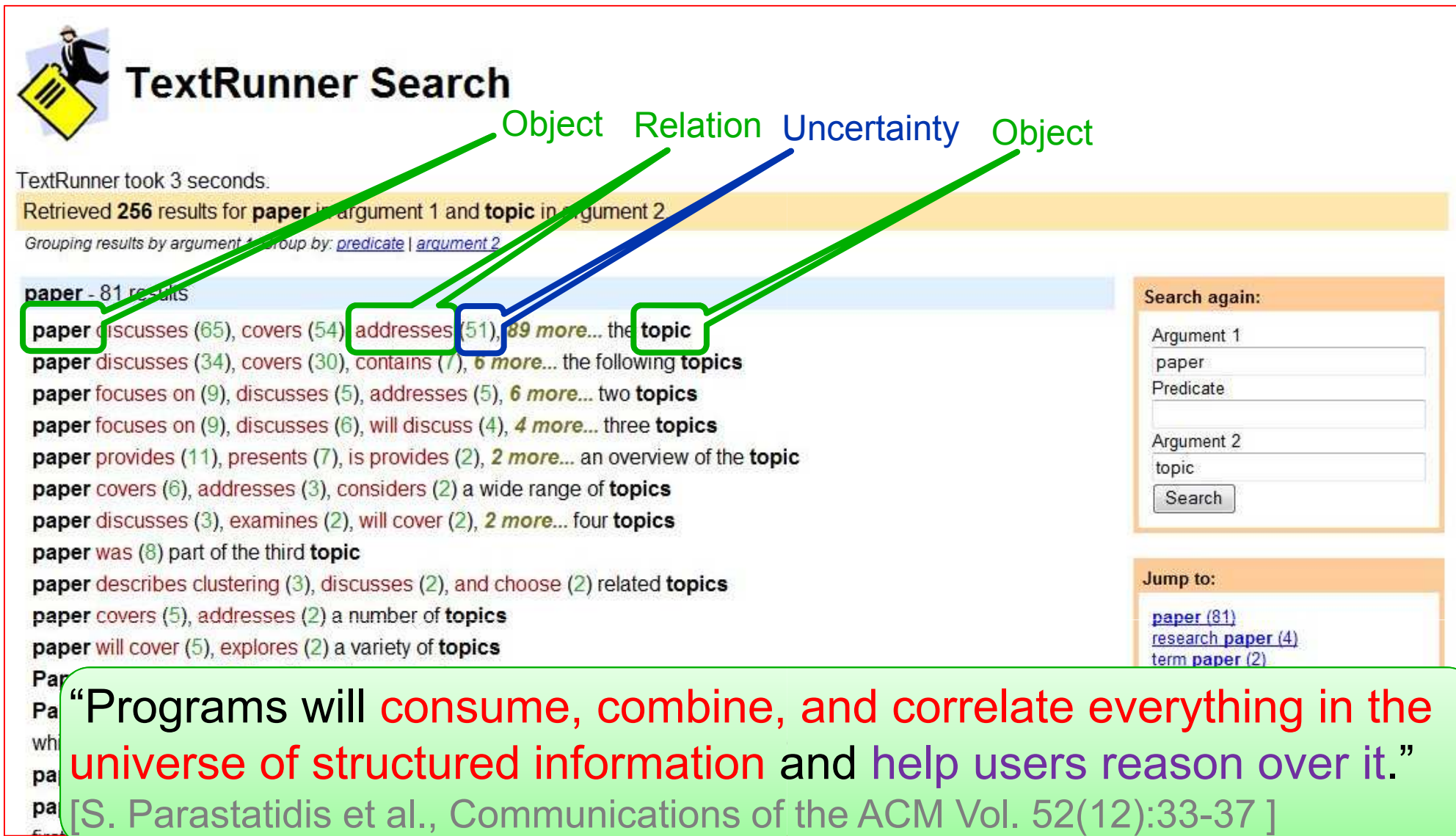
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



Fraunhofer

IAIS universität bonn  
RHEINISCHE FRIEDRICH-WILHELMS-  
UNIVERSITÄT





**TextRunner Search**

TextRunner took 3 seconds.  
Retrieved **256** results for **paper** in argument 1 and **topic** in argument 2.  
Grouping results by argument 1. Group by: [predicate](#) | [argument 2](#)

**paper** - 81 results

**paper** discusses (65), covers (54), **addresses** (51), 89 more... the **topic**

**paper** discusses (34), covers (30), contains (7), 6 more... the following **topics**

**paper** focuses on (9), discusses (5), addresses (5), 6 more... two **topics**

**paper** focuses on (9), discusses (6), will discuss (4), 4 more... three **topics**

**paper** provides (11), presents (7), is provides (2), 2 more... an overview of the **topic**

**paper** covers (6), addresses (3), considers (2) a wide range of **topics**

**paper** discusses (3), examines (2), will cover (2), 2 more... four **topics**

**paper** was (8) part of the third **topic**

**paper** describes clustering (3), discusses (2), and choose (2) related **topics**

**paper** covers (5), addresses (2) a number of **topics**

**paper** will cover (5), explores (2) a variety of **topics**

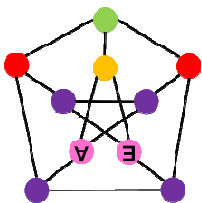
Search again:

Argument 1  
paper  
Predicate  
  
Argument 2  
topic  
Search

Jump to:

[paper \(81\)](#)  
[research paper \(4\)](#)  
[term paper \(2\)](#)

“Programs will consume, combine, and correlate everything in the universe of structured information and help users reason over it.”  
[S. Parastatidis et al., Communications of the ACM Vol. 52(12):33-37]



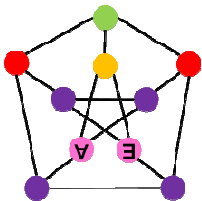


# So, the Real World is **Complex** and **Uncertain**

- Information overload
- Incomplete and contradictory information
- Many sources and modalities
- Variable number of objects and relations among them
- Rapid change



**How can computer systems handle these ?**



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



# AI and ML: State-of-the-Art



## Learning

MLSS10 Sunehag, Bartlett, ...

Decision trees, Optimization, SVMs, ...



## Logic

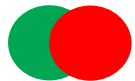
Resolution, WalkSat, Prolog, description logics, ...



## Probability

MLSS10 Sanner, Bonilla, ...

Bayesian networks, Markov networks, Gaussian Processes...



## Logic + Learning

Inductive Logic Programming (ILP)



## Learning + Probability

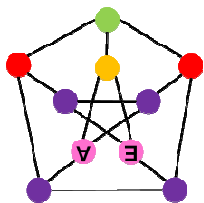
MLSS10 Uther, Johnson, Buntine, Warmuth, ...

EM, Dynamic Programming, Active Learning, ...



## Logic + Probability

Nilsson, Halpern, Bacchus, KBMC, ICL, ...



# (First-order) Logic handles Complexity

E.g., rules of chess (which is a tiny problem):

1 page in first-order logic,

~100000 pages in propositional logic,

[illegible]

## Explicit enumeration

- Many types of entities
- Relations between them
- Arbitrary knowledge

# Logic

true/false

## 5<sup>th</sup> C B.C.



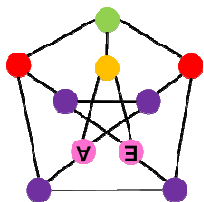
## 19<sup>th</sup> C



# atomic

# propositional

## first-order/relational

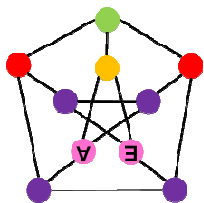
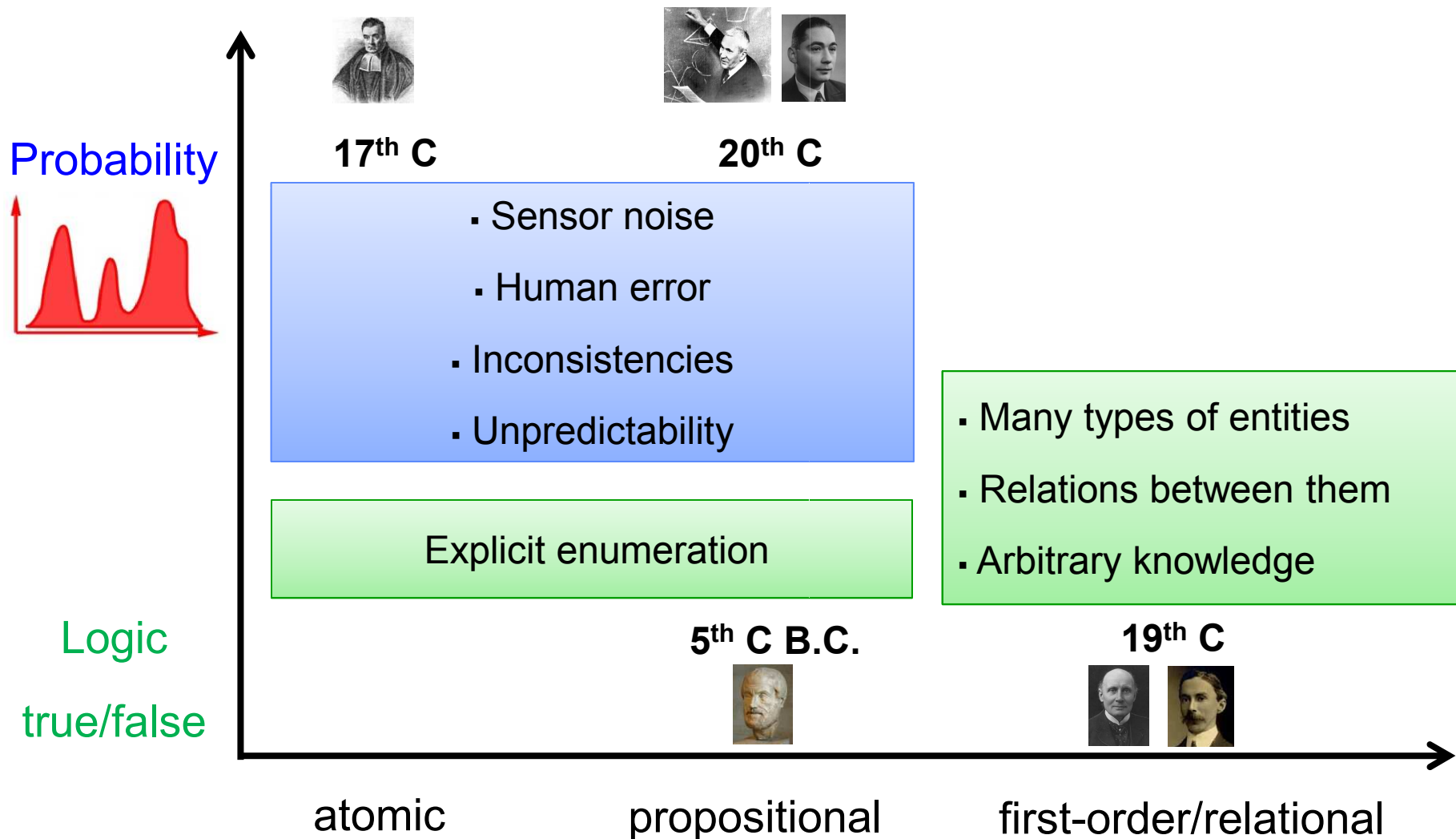


K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010

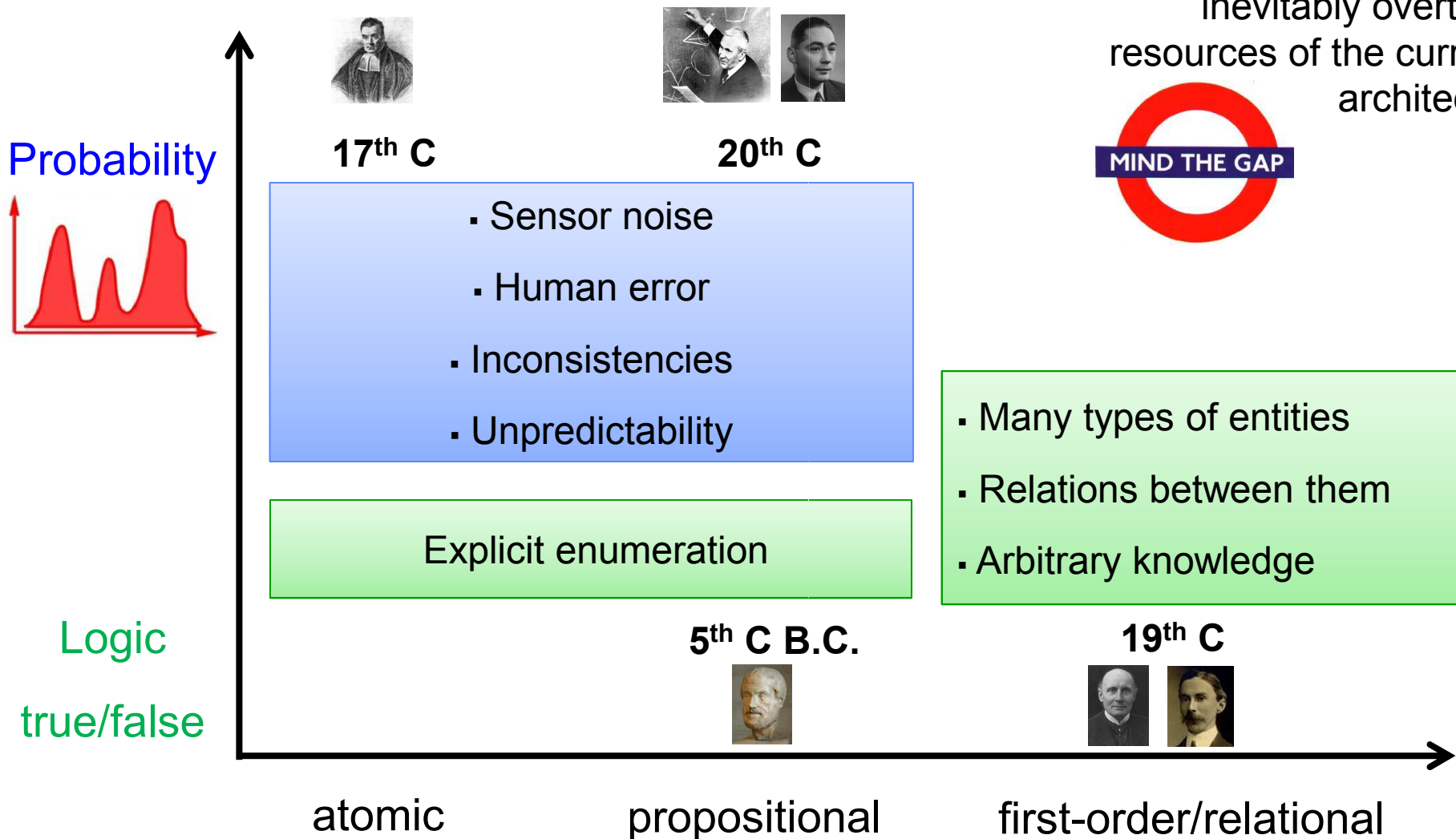

**Fraunhofer**

**fer**  
IAIS universität**bonn**  
RHEINISCHE FRIEDRICH-WILHELMS-  
UNIVERSITÄT

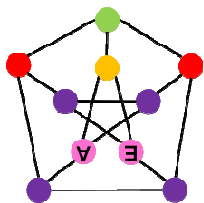
# Probability handles **Uncertainty**



# Will Traditional AI Scale ?

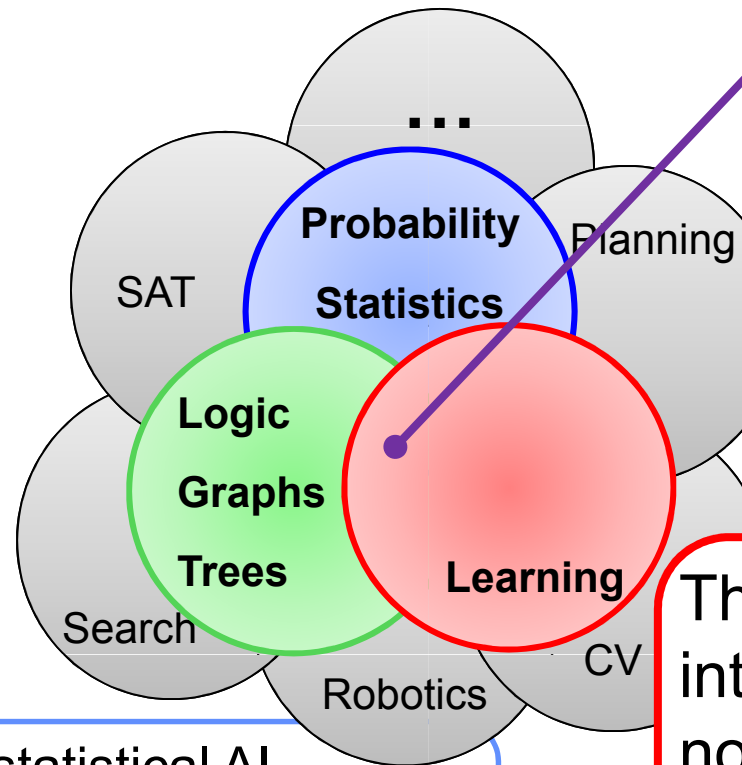


“Scaling up the environment will inevitably overtax the resources of the current AI architecture.”



# Statistical Relational Learning / AI (StarAI\*)

Let's **deal** with **uncertainty**, **objects**, and **relations** jointly



- Natural domain modeling: objects, properties, relations
- Compact, natural models
- Properties of entities can depend on properties of related entities
- Generalization over a variety of situations

... unifies logical and statistical AI,  
... solid formal foundations,  
... is of interest to many communities.

The study and design of intelligent agents that act in noisy worlds composed of objects and relations among the objects

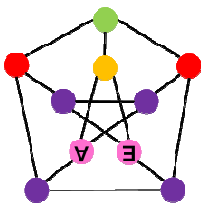
(\*)First StarAI workshop at AAAI10;co-chaired with S. Russell, L. Kaelbling, A.Halevy, S. Natarajan, and L. Milhalkova



---

# Today we can ...

- ... learn probabilistic relational models automatically from millions of inter-related objects
- ... generate optimal plans and learn to act optimally in uncertain environments involving millions of objects and relations among them
- ... perform lifted probabilistic inference avoiding explicit state enumeration by manipulating first-order state representations directly
- ... exploit shared factors to speed up message-passing algorithms for relational inference but also for classical propositional inference such as solving SAT problems



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**

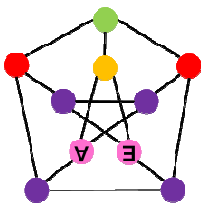


# Today's Menu

- Amuse-Gueule - Why SRL/StarAI?
- Appetizer – **More Motivation**
- Fish Course - **SR Frameworks**
  - Logic, MLNs, ProbLog, and Co
- Salad Course - **SR Learning**
  - (Vanilla) SRL, Boosting, Reinforcement Learning, Relational Topic Models
- Main Course - **Lifted Probabilistic inference**
  - Lifted Variable Elimination, Lifted Belief Propagation
- Dessert - **Conclusions and Outlook**

## Disclaimer

- Not a complete survey of SRL/StarAI or of foundational areas
- Focus is overview and providing some insights
- Assumes basic background in logic, probability and statistics, etc.
- Please ask questions



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**

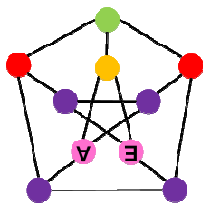




---

# Let's consider a simple example: Reviewing Papers

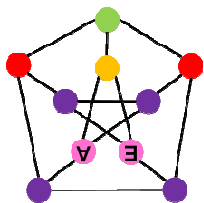
- The grade of a paper at a conference depends on the paper's quality and the difficulty of the conference.
  - Good papers may get A's at easy conferences
  - Good papers may get D's at top conference
  - Weak papers may get B's at good conferences
  - ...



# Propositional Logic

- **Good papers get A's at easy conferences**
  - $\text{good}(p1) \wedge \text{conference}(c1, \text{easy}) \Rightarrow \text{grade}(p1, c1, a)$
  - $\text{good}(p2) \wedge \text{conference}(c1, \text{easy}) \Rightarrow \text{grade}(p2, c1, a)$
  - $\text{good}(p3) \wedge \text{conference}(c3, \text{easy}) \Rightarrow \text{grade}(p3, c3, a)$

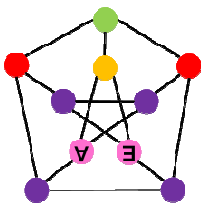
- Number of statements explodes with the number of papers and conferences
- No generalities, thus no generalization



# First Order Logic

- The grade of a paper at a conference depends on the paper's quality and the difficulty of the conference.
  - **Good papers get A's at easy conferences**
- $\forall \mathbf{P}, \mathbf{C} \text{ [good}(\mathbf{P}) \wedge \text{conference}(\mathbf{C}, \text{easy}) \Rightarrow \text{grade}(\mathbf{P}, \mathbf{C}, \mathbf{a}) \text{ ]}$

- Many 'all universals' are (almost) false
  - Even good papers can get either A, B, C
- True universals are rarely useful

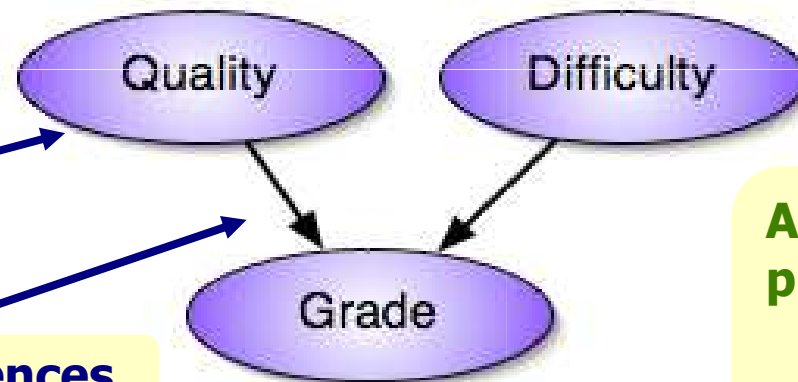


# Modeling the Uncertainty Explicitly

## Bayesian Networks: Directed Acyclic Graphs

Random Variables

Direct Influences

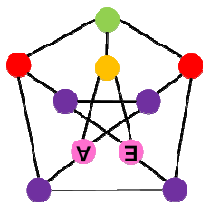


Associate a conditional probability distribution

$P(X_i | \text{pa}(X_i))$   
to each node

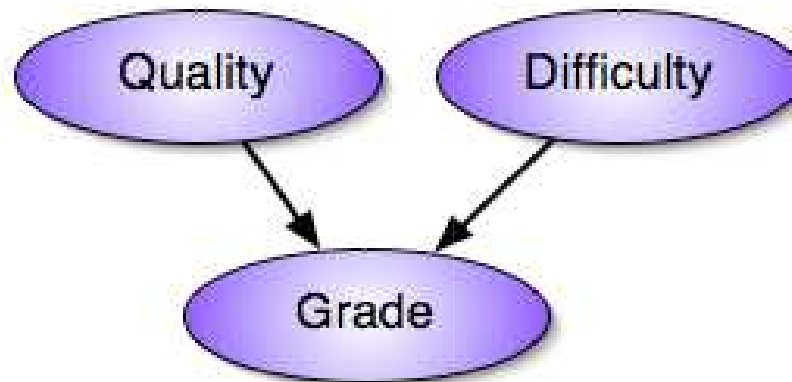
Compact representation of the joint probability distribution

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{pa}(X_i))$$



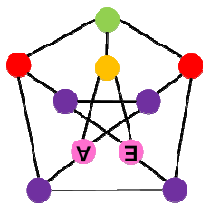
# (Reviewing) Bayesian Network ...

P(Qual)		
low	middle	high
0.3	0.5	0.2



P(Diff)		
low	middle	high
0.2	0.3	0.5

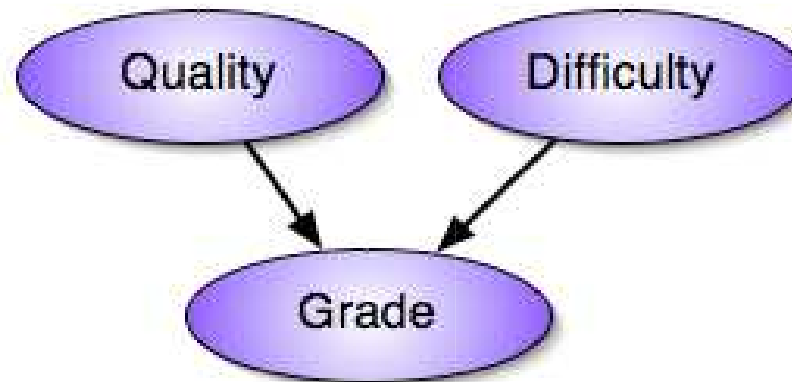
Qual	Diff	P(Grade)		
		c	b	a
low	low	0.2	0.5	0.3
low	middle	0.1	0.7	0.2
...				



## (Reviewing) Bayesian Network ...

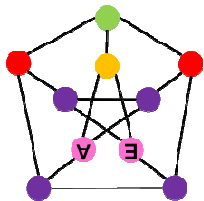
$$P(Qual = low, Diff = middle, Grade = a) = 0.3 \cdot 0.3 \cdot 0.2 = 0.018$$

P(Qual)		
low	middle	high
0.3	0.5	0.2



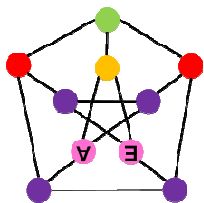
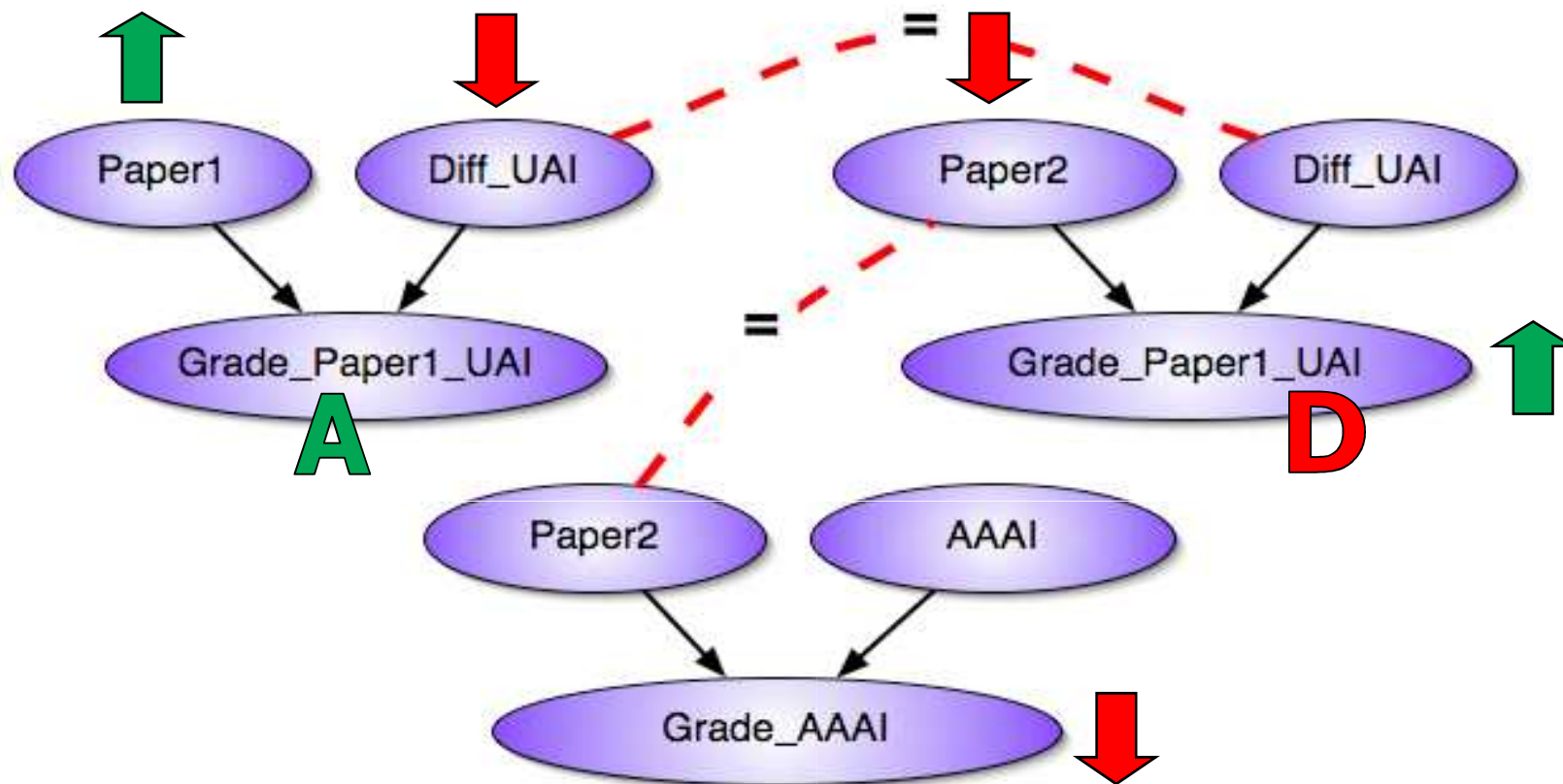
P(Diff)		
low	middle	high
0.2	0.3	0.5

Qual	Diff	P(Grade)		
		c	b	a
low	low	0.2	0.5	0.3
low	middle	0.1	0.7	0.2
...				



The real world, however, has **inter-related objects**

These 'instance' are not independent !



---

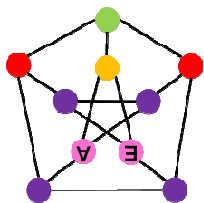
# Information Extraction

Parag Singla and Pedro Domingos, “Memory-Efficient Inference in Relational Domains” (AAAI-06).

Singla, P., & Domingos, P. (2006). Memory-efficient inference in relational domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

H. Poon & P. Domingos, Sound and Efficient Inference with Probabilistic and Deterministic Dependencies”, in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficient inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**





# Information Extraction

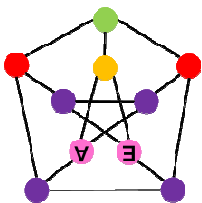


Parag Singla and Pedro Domingos, “Memory-Efficient Inference in Relational Domains” (AAAI-06).

Singla, P., & Domingos, P. (2006). Memory-efficient inference in relational domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

H. Poon & P. Domingos, Sound and Efficient Inference with Probabilistic and Deterministic Dependencies”, in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficient inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



Author

Title

Paper

Venue

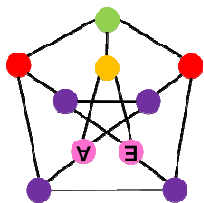
## Segmentation

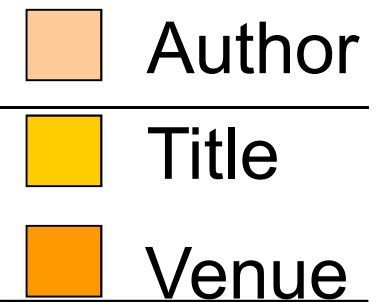
Parag Singla and Pedro Domingos, “Memory-Efficient Inference in Relational Domains” (AAAI-06).

Singla, P., & Domingos, P. (2006). Memory-efficient inference in relational domains. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (pp. 500-505). Boston, MA: AAAI Press.

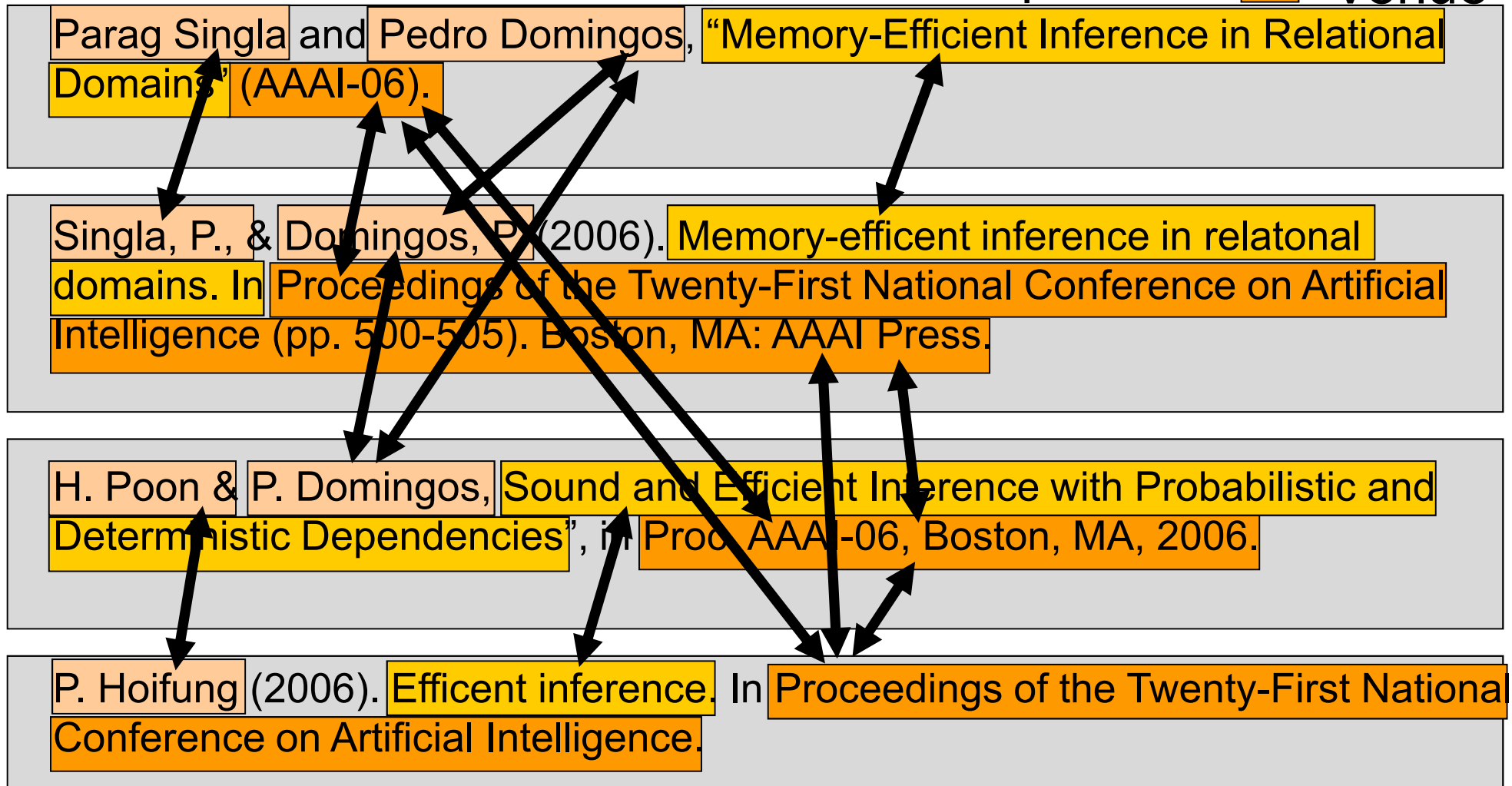
H. Poon & P. Domingos, “Sound and Efficient Inference with Probabilistic and Deterministic Dependencies”, in Proc. AAAI-06, Boston, MA, 2006.

P. Hoifung (2006). Efficient inference. In Proceedings of the Twenty-First National Conference on Artificial Intelligence.

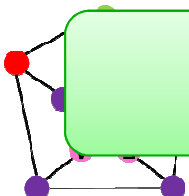




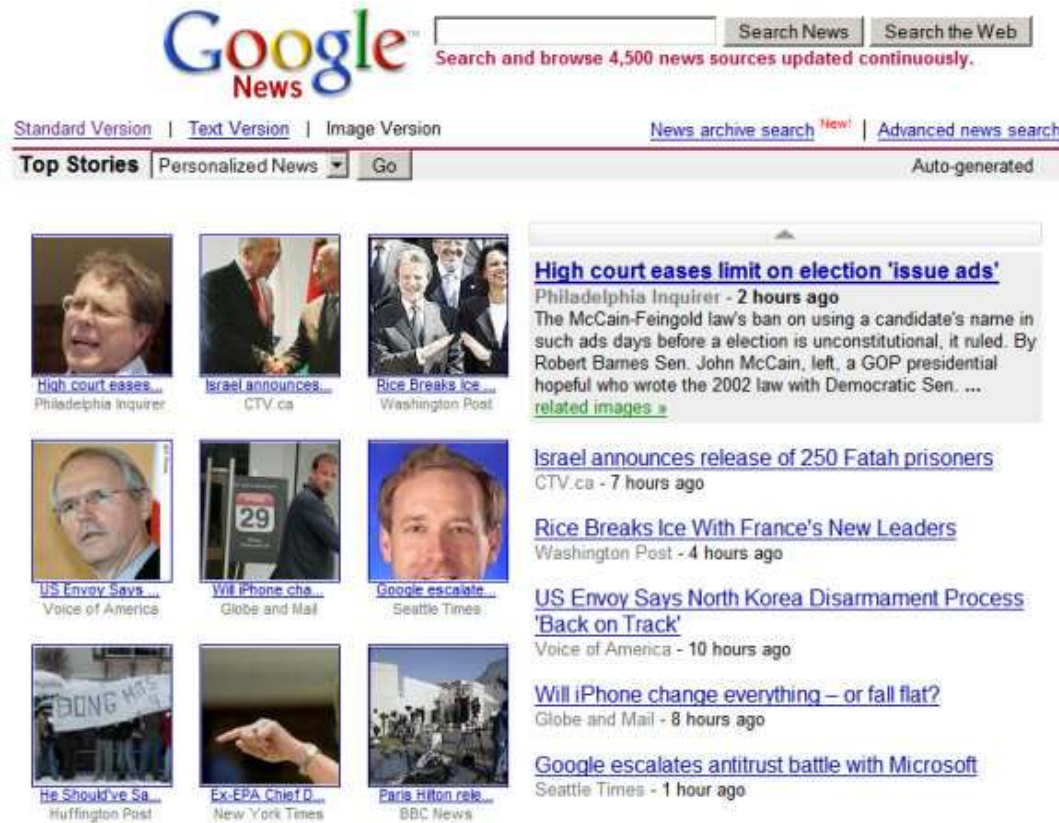
# Entity Resolution



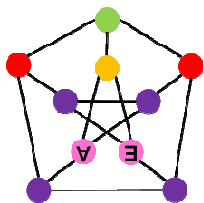
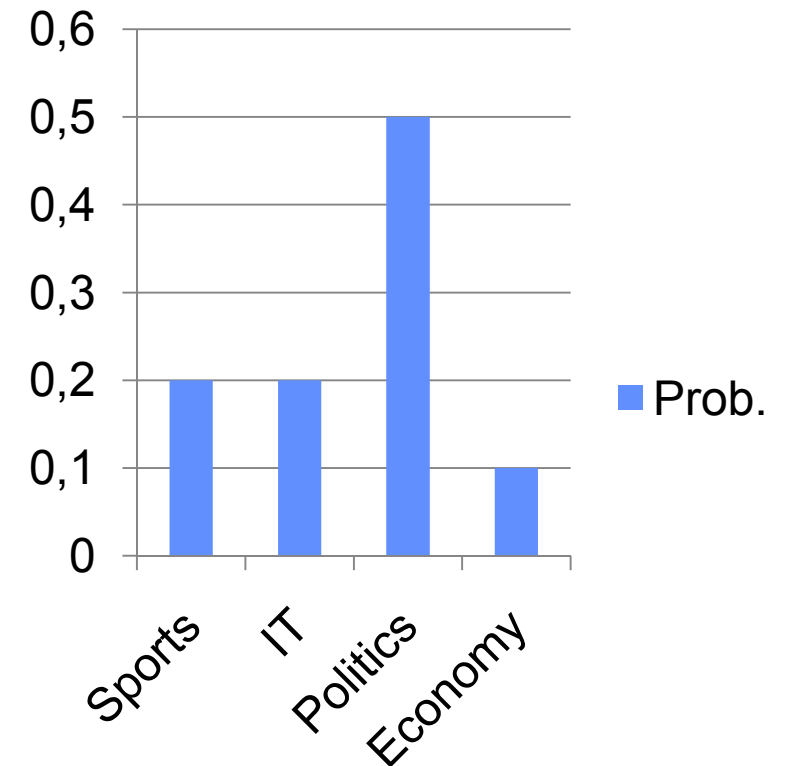
**Again, 'instance' are not independent !**



# Topic Models



Prob.



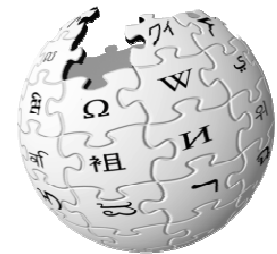
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



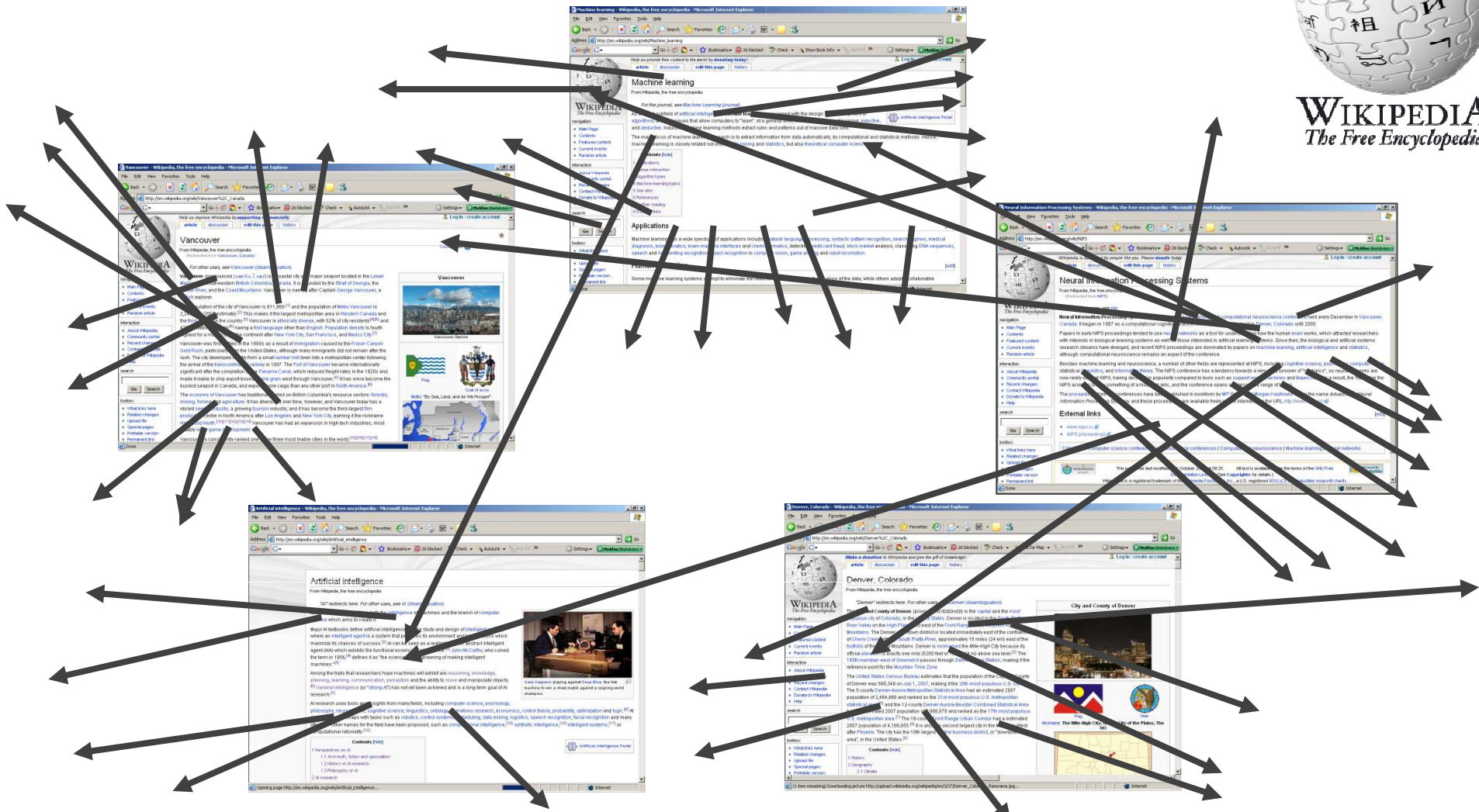
Fraunhofer



# Wikipedia



WIKIPEDIA  
The Free Encyclopedia



Again, 'instance' are not independent !





## TextRunner Search

TextRunner took 3 seconds.

Retrieved **256** results for **paper** in argument 1 and **topic** in argument 2.

Grouping results by argument 1 group by: [predicate](#) | [argument 2](#)

paper - 81 results

paper discusses (65), covers (54), addresses (51), 89 more... the topic  
 paper discusses (34), covers (30), contains (7), 6 more... the following to  
 paper focuses on (9), discusses (5), addresses (5), 6 more... two topics  
 paper focuses on (9), discusses (6), will discuss (4), 4 more... three topics  
 paper provides (11), presents (7), is provides (2), 2 more... an overview of  
 paper covers (6), addresses (3), considers (2) a wide range of topics  
 paper discusses (3), examines (2), will cover (2), 2 more... four topics  
 paper was (8) part of the third topic  
 paper describes clustering (3), discusses (2), and choose (2) related topics  
 paper covers (5), addresses (2) a number of topics  
 paper will cover (5), explores (2) a variety of topics

Object Relation Uncertainty Object

No complex inference (yet) !

**TextRunner:** (Turing, born in, London)

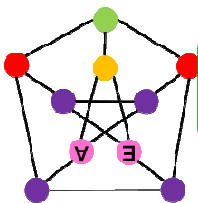
+ **WordNet:** (London, part of, England)

+ **Rule:** 'born in' is transitive thru 'part of'

**Conclusion:** (Turing, born in, **England**)

"Programs will consume, combine, and correlate everything in the universe of structured information and help users reason over it."

[S. Parastatidis et al., Communications of the ACM Vol. 52(12):33-37]



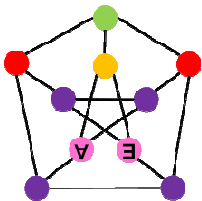
And again, 'instance' are not independent !

# Relations are everywhere ...

- Hyperlinks in web pages
- References in scientific publications
- Social networks
- Ontologies
- ...

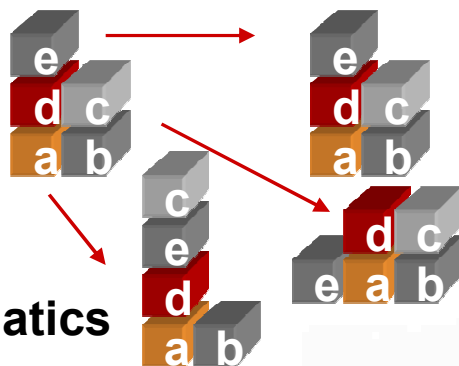
**and connectivity is important**

- PageRank

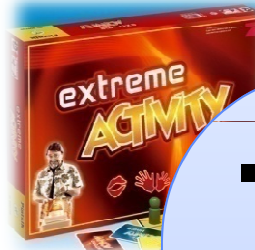


# Objects + Relations + Uncertainty are everywhere

Planning



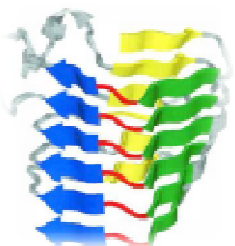
Activity  
Recognition



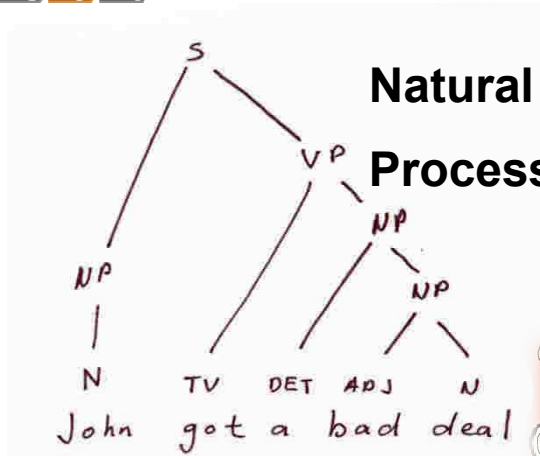
Social Networks



BioInformatics



Natural Language  
Processing

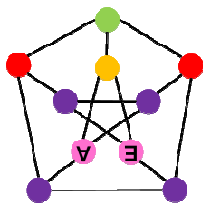


Robotics



Data Cleaning

- Web data (**web**)
- Biological data (**bio**)
- Social Network Analysis (**soc**)
- Bibliographic data (**cite**)
- Epidemiological data (**epi**)
- Communication data (**comm**)
- Customer networks (**cust**)
- Collaborative filtering problems (**cf**)
- Trust networks (**trust**)
- ...





# Costs and Benefits of SRL / StarAI

Relations can reveal additional correlations. Abstraction allows for generalization.

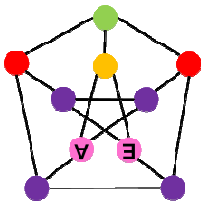
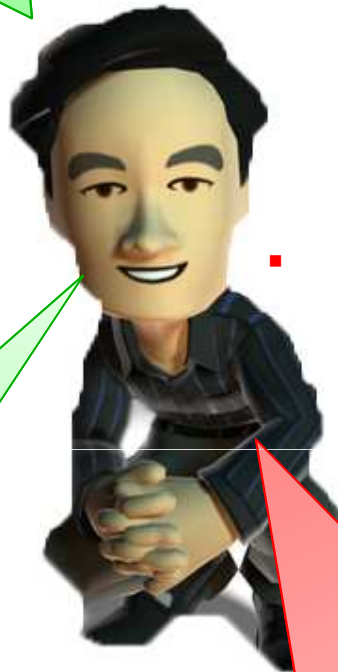
## ■ Benefits

- Better predictive accuracy
- Better understanding of domains
- Growth path for machine learning and artificial intelligence

## ■ Costs

- Learning is much harder
- Inference becomes a crucial issue
- Greater complexity for user

SRL/StarAI techniques have the potential to lay the foundations of next generation AI systems



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School  
ANU, Canberra, Australia

**Yes, SRL/StarAI is challenging but knowing one of its ingredients is half the battle**

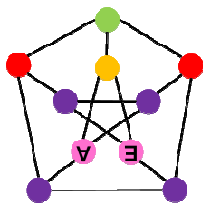
---

## So far

- The world is complex and uncertain
- Reviewing papers
- Joint segmentation and entity resolution
- Topic models

## Now

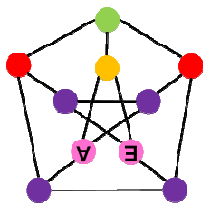
- Let's get started!
- From logic to statistical relational frameworks



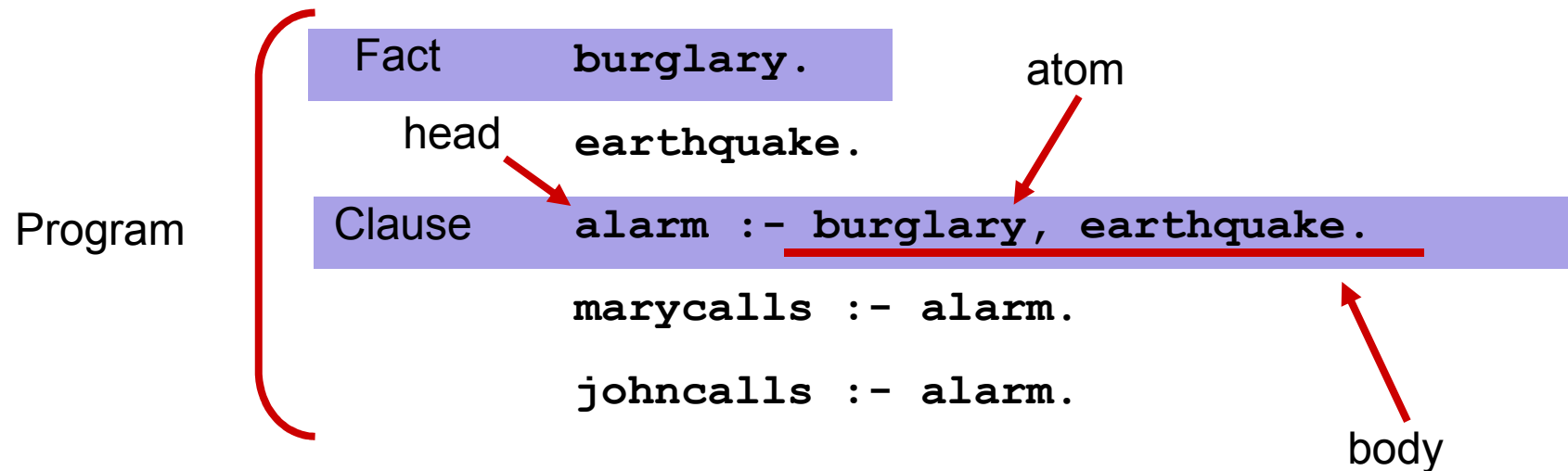
---

# Main StarAI / SRL Key Dimensions

- **Logical language**  
First-order logic, Horn clauses, frame systems
- **Probabilistic language**  
Bayesian networks, Markov networks, PCFGs
- **Type of learning**
  - Generative / Discriminative
  - Structure / Parameters
  - Knowledge-rich / Knowledge-poor
- **Type of inference**
  - MAP / Marginal
  - Full grounding / Partial grounding / Lifted



# (Propositional) LP – Some Notations



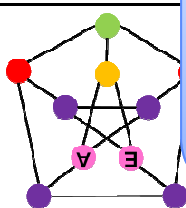
**Herbrand Base (HB)** = all atoms in the program

burglary, earthquake, alarm, marycalls, johncalls

**Clauses:** IF burglary and earthquake are true THEN alarm is true

Two closely related ways to define semantics

1. Model-theoretic
2. Proof-theoretic



# Model Theoretic: Restrictions on Possible Worlds

- Herbrand Interpretation

- Truth assignments to all elements of HB

- An interpretation is a **model** of a clause  $C \Leftrightarrow$

If the body of  $C$  holds then the head holds, too

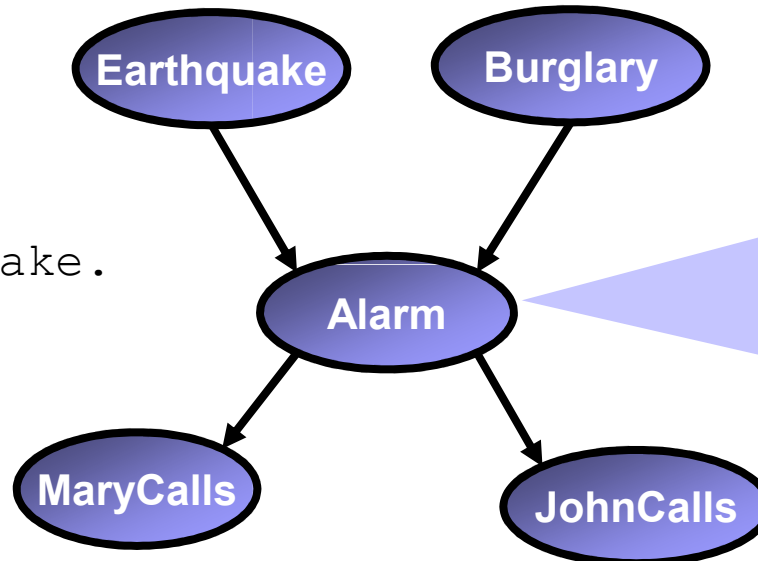
burglary.

earthquake.

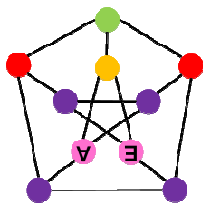
alarm :- burglary, earthquake.

marycalls :- alarm.

johncalls :- alarm.



$E$	$B$	$P(A \mid B, E)$	
$e$	$b$	0.9	0.1
$e$	$\bar{b}$	0.2	0.8
$\bar{e}$	$b$	0.9	0.1
$\bar{e}$	$\bar{b}$	0.01	0.99



# Proof Theoretic: Restrictions on Possible Derivations

- A set of clauses can be used to prove that atoms are entailed by the set of clauses.

Goal

```
:- johncalls.
```

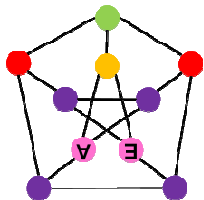
```
burglary, earthquake.
```

```
earthquake.
```

```
alarm :- burglary, earthquake.
```

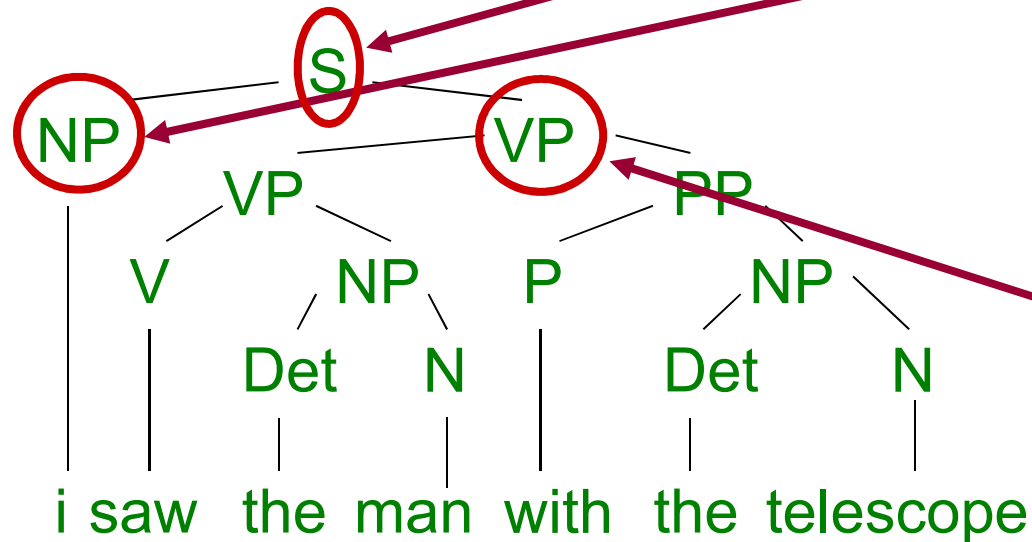
```
marycalls :- alarm.
```

```
johncalls :- alarm.
```



# Stochastic Grammars

Upgrade HMMs (regular languages) to more complex languages such as context-free languages.



$$1.0 * 1/3 * 0.5 * 0.5 * 1.0 * \dots$$

$$= 0.00231$$

## Weighted Rewrite Rules

1.0 : S → NP, VP

1/3 : NP → i

1/3 : NP → Det, N

1/3 : NP → NP, PP

1.0 : Det → the

0.5 : N → man

0.5 : N → telescope

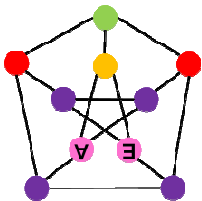
0.5 : VP → V, NP

0.5 : VP → VP, PP

1.0 : PP → P, NP

1.0 : V → saw

1.0 : P → with



# Upgrading to First-Order Logic



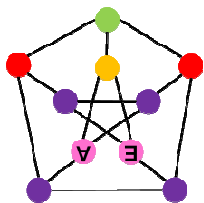
```
father(rex,fred).      mother(ann,fred).  
father(brian,doro).   mother(utta,doro).  
father(fred,henry).   mother(doro,henry).  
pchrom(rex,a).  mchrom(rex,a).  
pchrom(ann,a).  mchrom(ann,b).  
...
```

The maternal information `mchrom/2` depends on the maternal and paternal `pchrom/2` information of the mother `mother/2`:

```
mchrom(fred,a). mchrom(fred,b), ...
```

or better

```
mchrom(P,a) :- mother(M,P), pchrom(M,a), mchrom(M,a).  
mchrom(P,a) :- mother(M,P), pchrom(M,a), mchrom(M,b).  
mchrom(P,b) :- mother(M,P), pchrom(M,a), mchrom(M,b).  
...
```



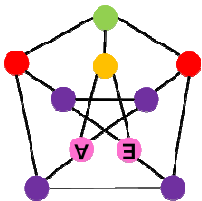


# Upgrading - continued

**Propositional Clausal Logic**  
Expressions can be true or false

head                      body

clause      `alarm :- burglary, earthquake.`



# Upgrading - continued

**Relational Clausal Logic**  
Constants and variables refer to objects

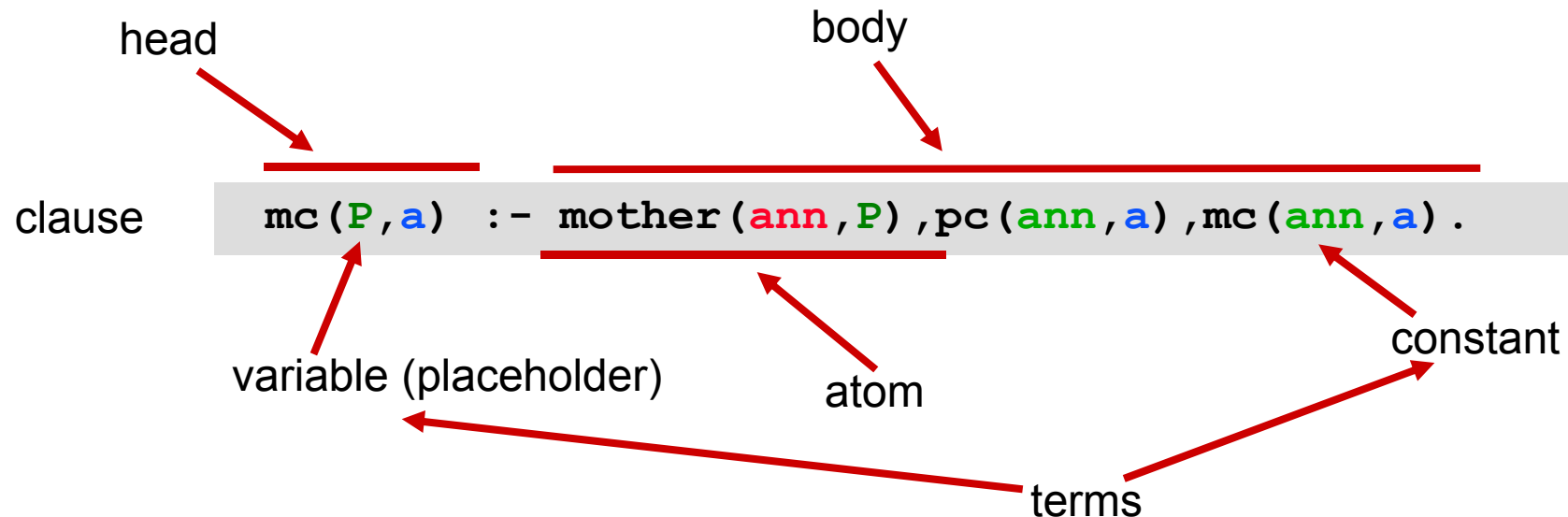
**Propositional Clausal Logic**  
Expressions can be true or false

**Substitution:** Maps variables to terms:  $\{M / ann\}$ :

`mc(P, a) :- mother(ann, P), pc(ann, a), mc(ann, a) .`

**Herbrand base:** set of ground atoms (no variables):

`{mc(fred, fred), mc(rex, fred), ...}`



# Upgrading - continued

**Full Clausal Logic**  
Functors aggregate objects

**Relational Clausal Logic**  
Constants and variables refer to objects

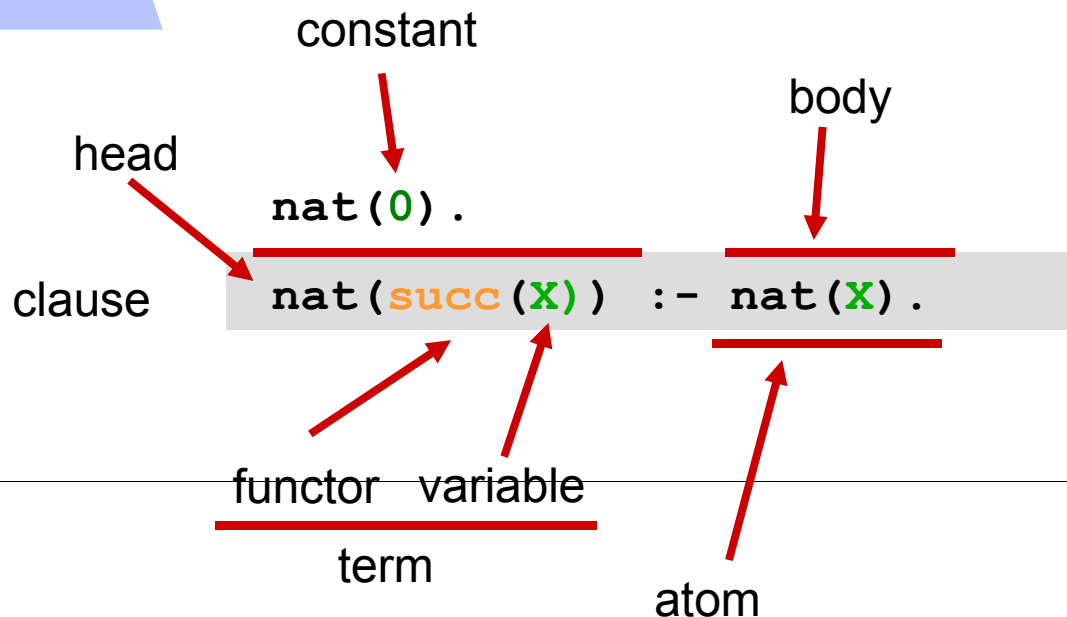
**Propositional Clausal Logic**  
Expressions can be true or false

**Substitution:** Maps variables to terms:  $\{M / ann\}$ :

▪  $mc(P, a) :- mother(ann, P), pc(ann, a), mc(ann, a) .$

**Herbrand base:** set of ground atoms (no variables):

▪  $\{mc(fred, fred), mc(rex, fred), \dots\}$



Interpretations can be infinite !

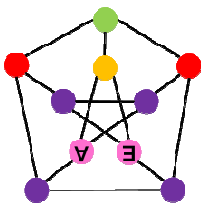
`nat(0), nat(succ(0)),`

`nat(succ(succ(0))), ...`

---

# Inference in First-Order Logic

- Traditionally done by theorem proving (e.g.: Prolog)
- Main approach within SRL: Propositionalization followed by “model checking”
  - **Propositionalization:**  
Create all ground atoms and clauses
  - **Model checking:** Inference in graphical models, weighted Satisfiability testing



# Forward Chaining

```

father(rex,fred) .      mother(ann,fred) .
father(brian,doro) .    mother(utta, doro) .
father(fred,henry) .    mother(doro,henry) .
pc(rex,a) .   mc(rex,a) .
pc(ann,a) .   mc(ann,b) .
...

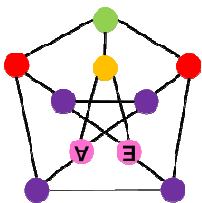
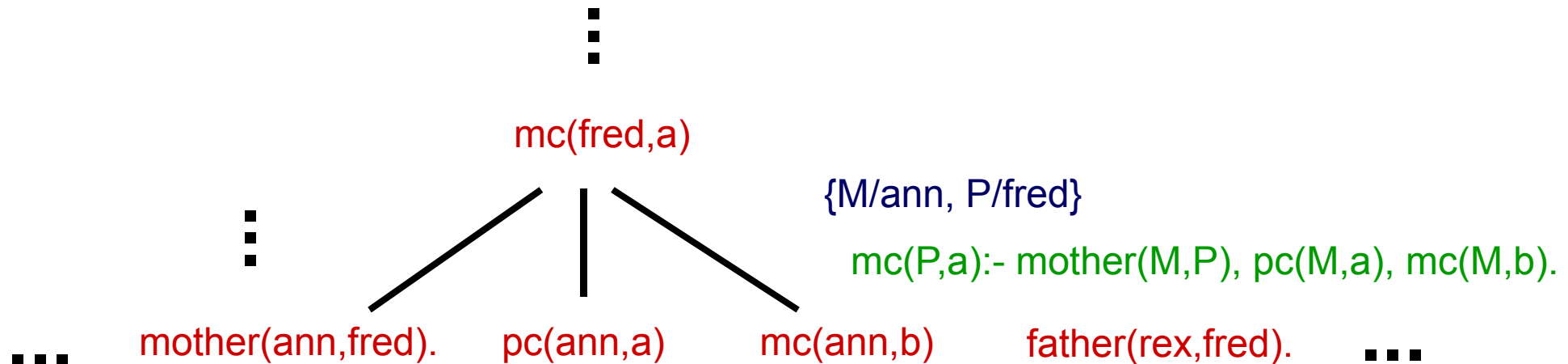
```

```

mc(P,a) :- mother(M,P), pc(M,a), mc(M,a).
mc(P,a) :- mother(M,P), pc(M,a), mc(M,b).

```

Set of derivable ground atoms = least Herbrand model



# Backward Chaining

```
father(rex,fred) .      mother(ann,fred) .
father(brian,doro) .    mother(utta, doro) .
father(fred,henry) .    mother(doro,henry) .
pc(rex,a) .   mc(rex,a) .
pc(ann,a) .   mc(ann,b) .
...
```

```
mc(P,a) :- mother(M,P), pc(M,a), mc(M,a).
mc(P,a) :- mother(M,P), pc(M,a), mc(M,b).
```

mc(fred,a)

mc(P,a) :- mother(M,P), pc(M,a), mc(M,a).

{P/fred}

mother(M,fred),pc(M,a),mc(M,a)

mother(ann,fred).

{M/ann}

pc(ann,a),mc(ann,a)

pc(ann,a).

mc(ann,a)

fail

mc(P,a) :- mother(M,P), pc(M,a), mc(M,b).

{P/fred}

mother(M,fred),pc(M,a),mc(M,b)

mother(ann,fred).

{M/ann}

pc(ann,a),mc(ann,b)

pc(ann,a).

mc(ann,b)

success

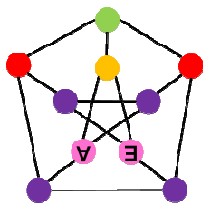
---

## So far

- Motivation
- Brief review of logic

## Now

- Let's see some actual SRL frameworks



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



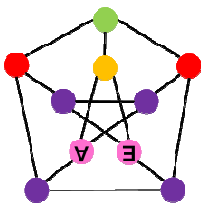
---

# Alphabetic Soup of SRL

- Knowledge-based model construction [Wellman et al., 1992]
- PRISM [Sato & Kameya 1997]
- Stochastic logic programs [Muggleton, 1996]
- Probabilistic relational models [Friedman et al., 1999]
- Bayesian logic programs [Kersting & De Raedt, 2001]
- Bayesian logic [Milch et al., 2005]
- Markov logic [Richardson & Domingos, 2006]
- Relational dependency networks [Neville & Jensen 2007]
- ProbLog [De Raedt et al., 2007]



**And many others!**



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



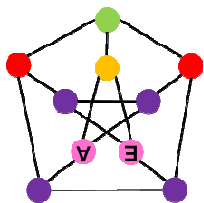
**Fraunhofer**





# Probabilistic Relational Models

- **Logical language:** Frame systems
- **Probabilistic language:** Bayes nets
  - Bayes net template for each class of objects
  - Object's attributes can depend on attributes of related objects
  - >1 related objects of same type: aggregate functions avg, min, ...
  - Only binary relations
  - No dependencies of relations on relations
- **Learning:**
  - Parameters: Closed form (EM/Gradient if missing data)
  - Structure: "Tiered" Bayesian network structure search
- **Inference:** Full grounding + Belief propagation



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010

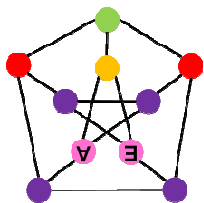


**Fraunhofer**



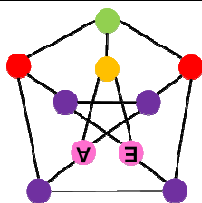
# PRISM

- **Logical language:** Prolog / Horn clauses
- **Probabilistic language:**  
Probabilistic context-free grammars
  - Attach probabilities to clauses using probability switches
  - $\sum \text{Probs. of switch} = 1$
- **Learning:** EM-BDD
- **Inference:** Do all proofs, add probabilities, use BDDs for efficient computation



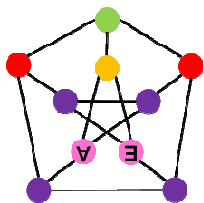
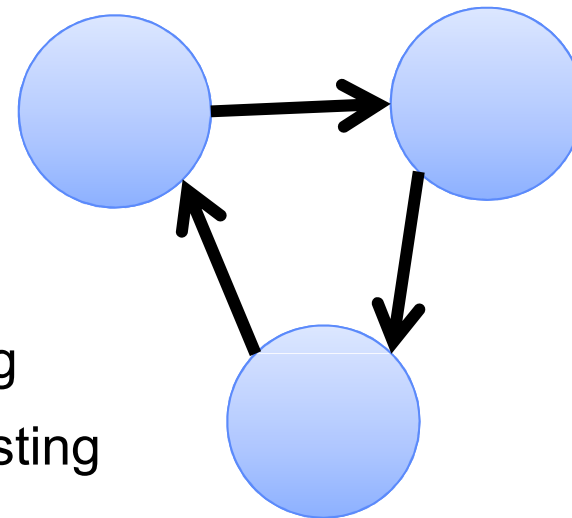
# Stochastic Logic Programs

- **Logical language:** Horn clauses
- **Probabilistic language:**  
Probabilistic context-free grammars
  - Attach probabilities to clauses
  - $\sum$  Probs. of clauses w/ same head = 1
- **Learning:** ILP + “Failure-adjusted” EM
- **Inference:** Do all proofs, add probabilities



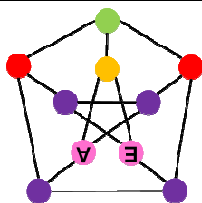
# Relational Dependency Networks

- **Logical language:** SQL queries
- **Probabilistic language:** Dependency networks
  - Conditional probability template for each predicate
  - Atoms depend on related atoms
  - >1 clause w/ head: aggregate functions
  - Cyclic dependencies
- **Learning:**
  - Parameters: EM based on Gibbs sampling
  - Structure: relational probability trees, boosting
- **Inference:** Gibbs sampling



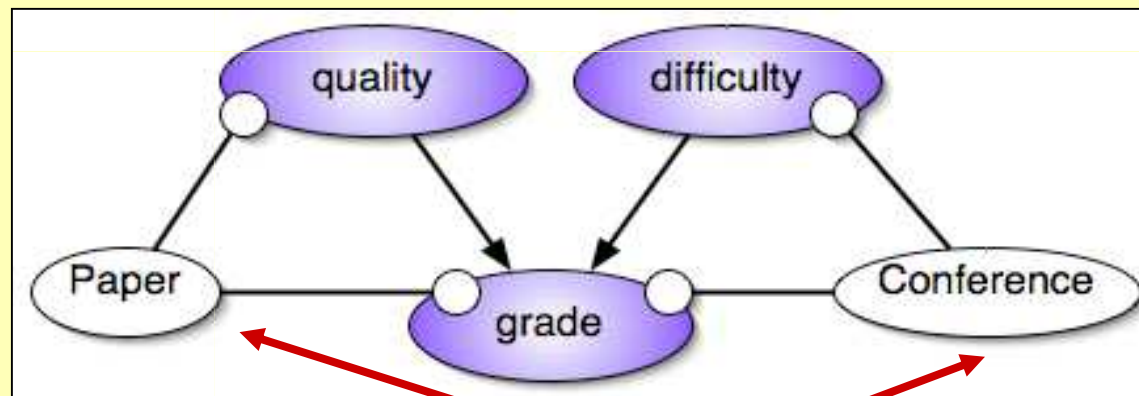
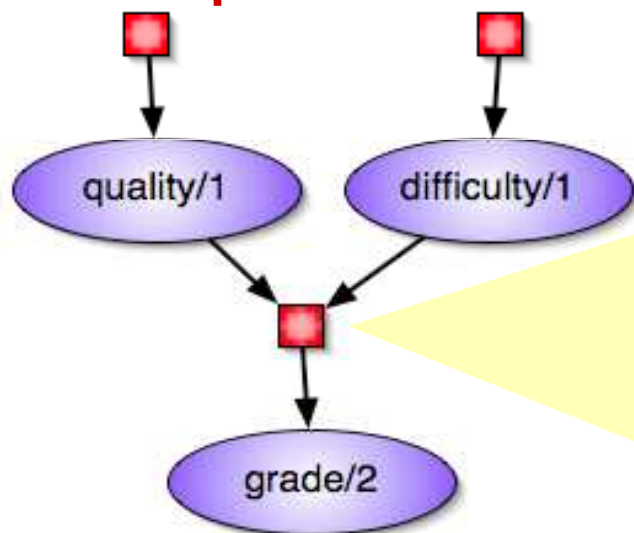
# Bayesian Logic Programs

- **Logical language:** Prolog / Horn clauses
- **Probabilistic language:** Bayesian networks
  - Ground atom  $\rightarrow$  Node
  - Head of clause  $\rightarrow$  Child node
  - Body of clause  $\rightarrow$  Parent nodes
  - $>1$  clause w/ same head  $\rightarrow$  Combining functions noise-or, ...
- **Learning:** Tight integration of ILP + EM/Gradient
- **Inference:** Grounding + Belief propagation



# (Reviewing) Bayesian Logic Programs

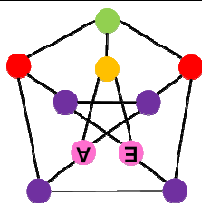
## Rule Graph



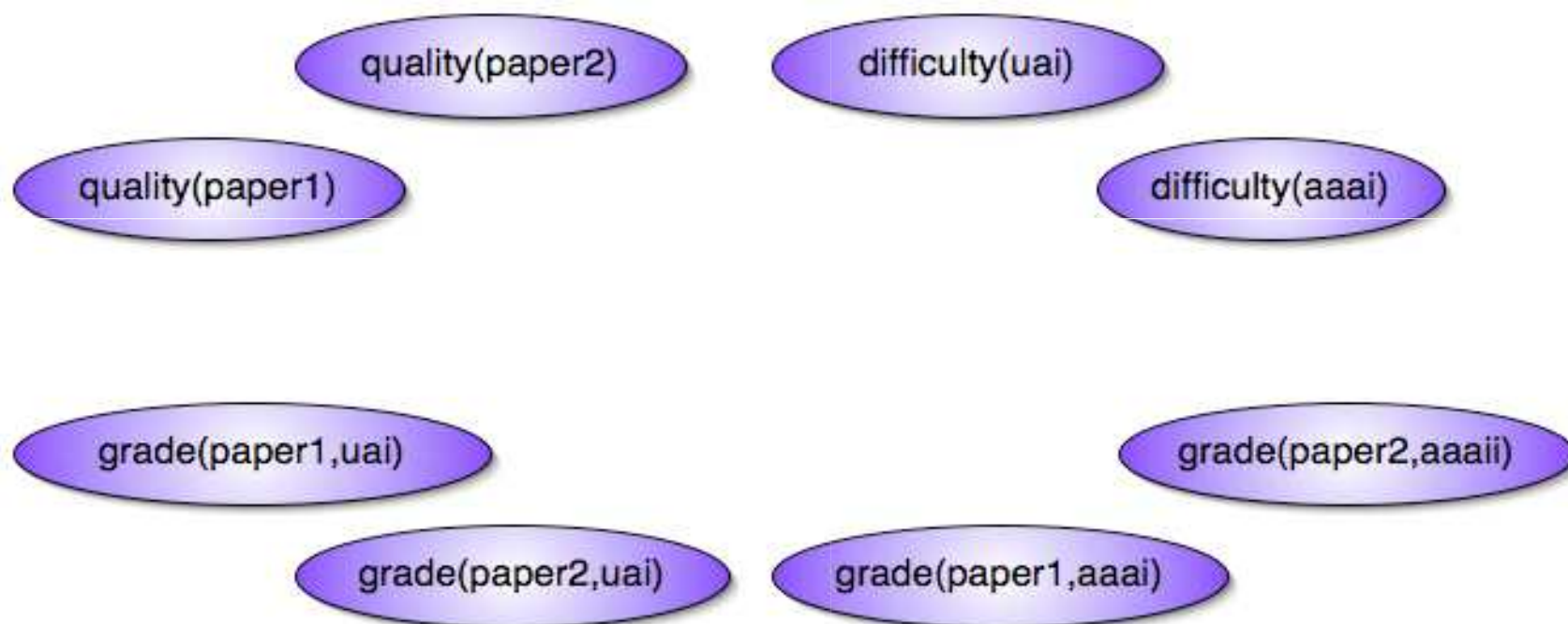
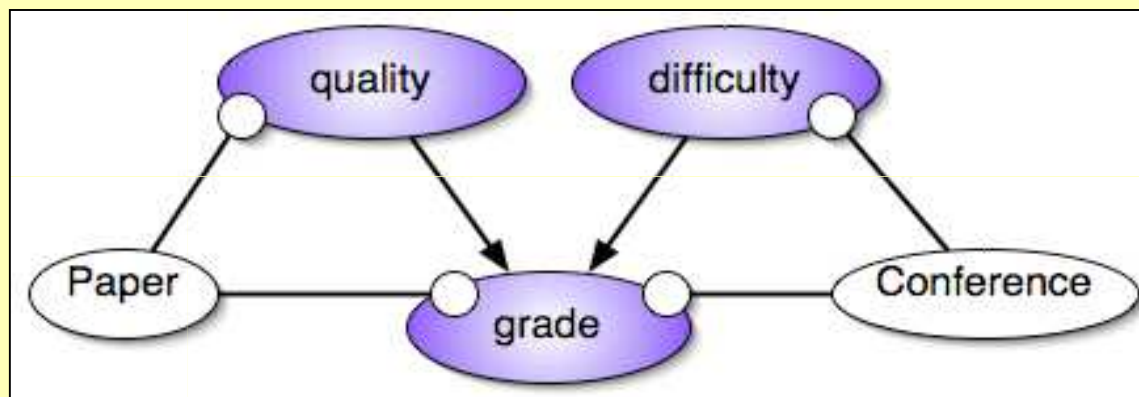
placeholder

`grade(P, C) | quality(P), difficulty(C).`

quality(Paper)	Difficulty(Conference)	P(grade(Paper,Conference))		
		c	b	a
low	low	0.2	0.5	0.3
low	middle	0.1	0.7	0.2
...				

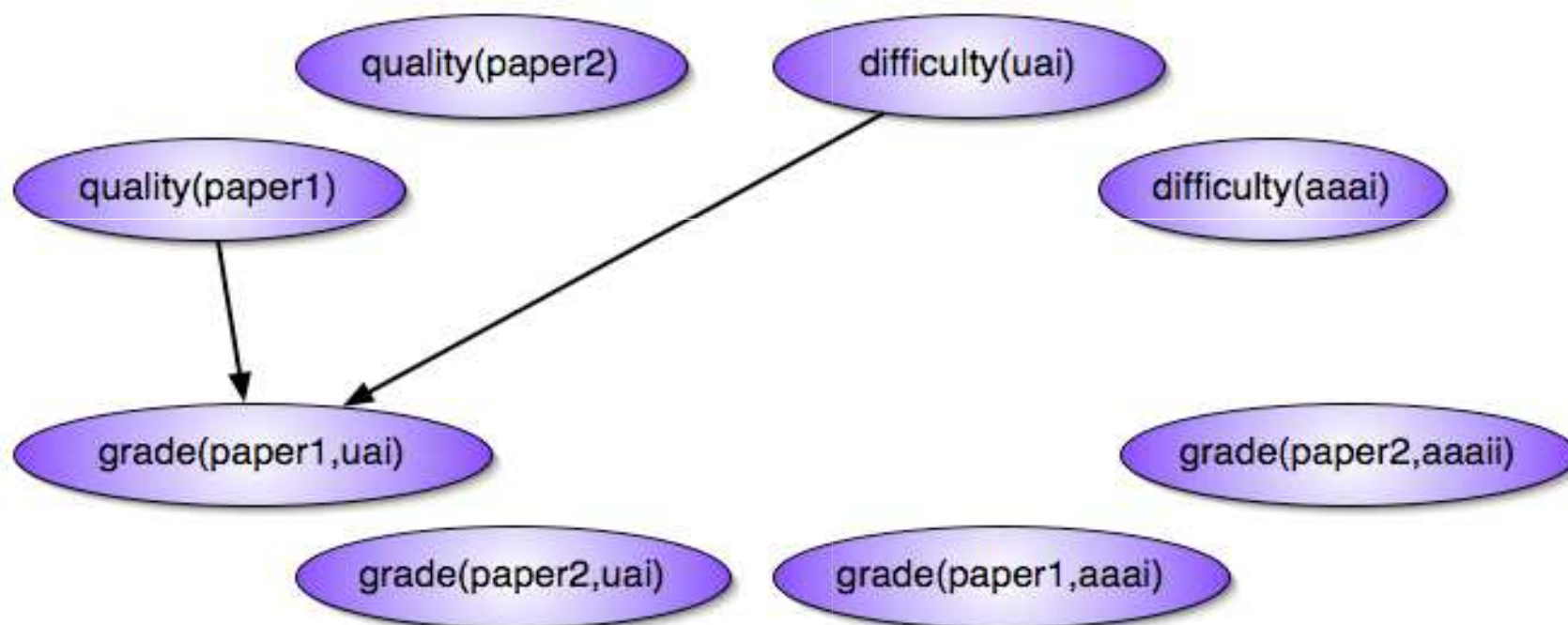
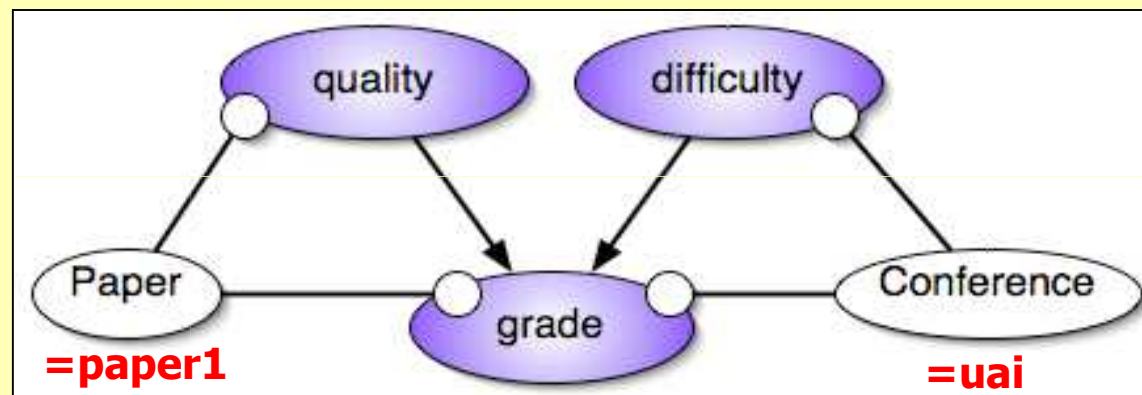


# (Reviewing) Bayesian Logic Programs

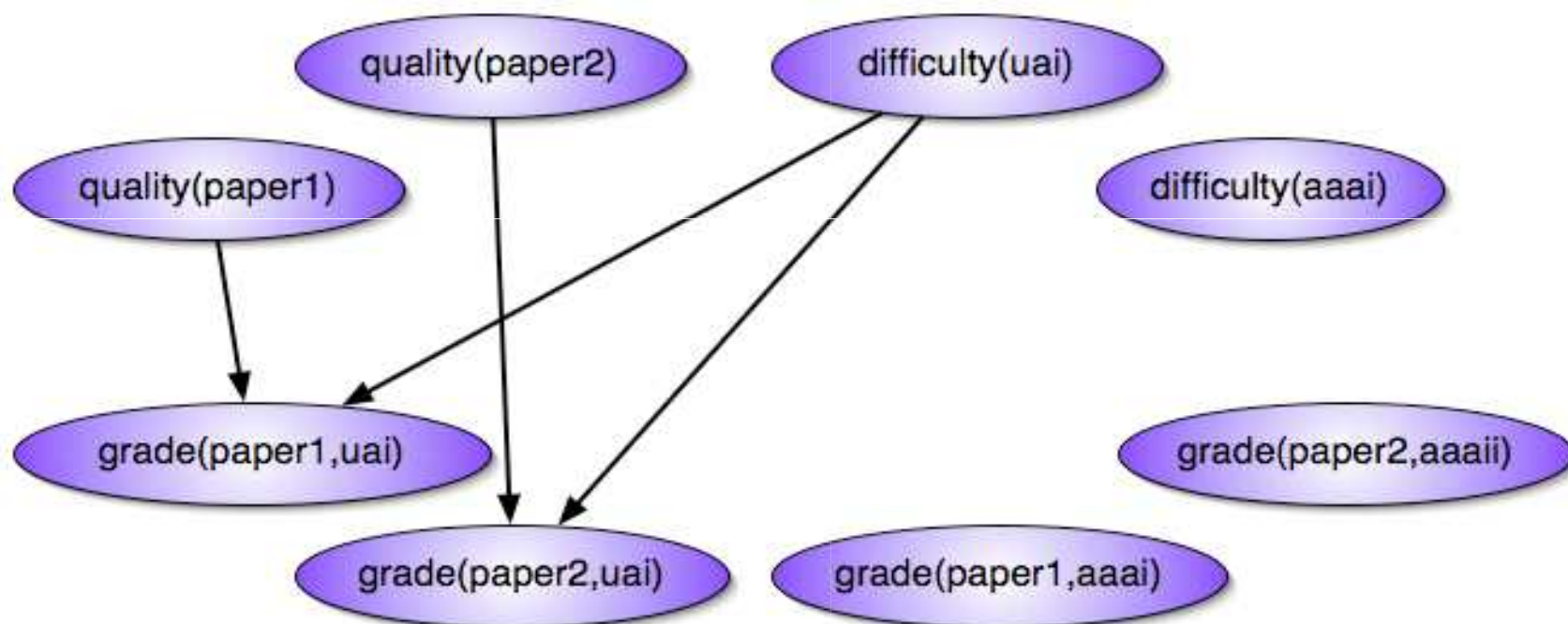
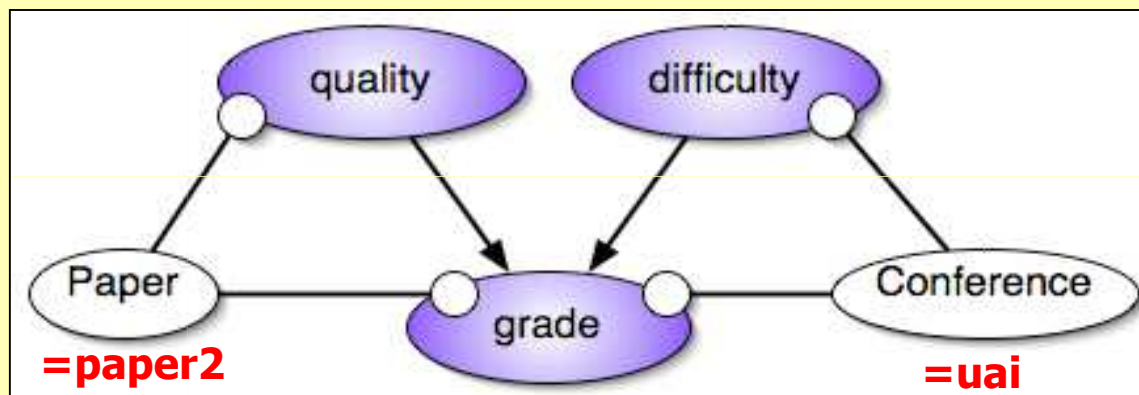




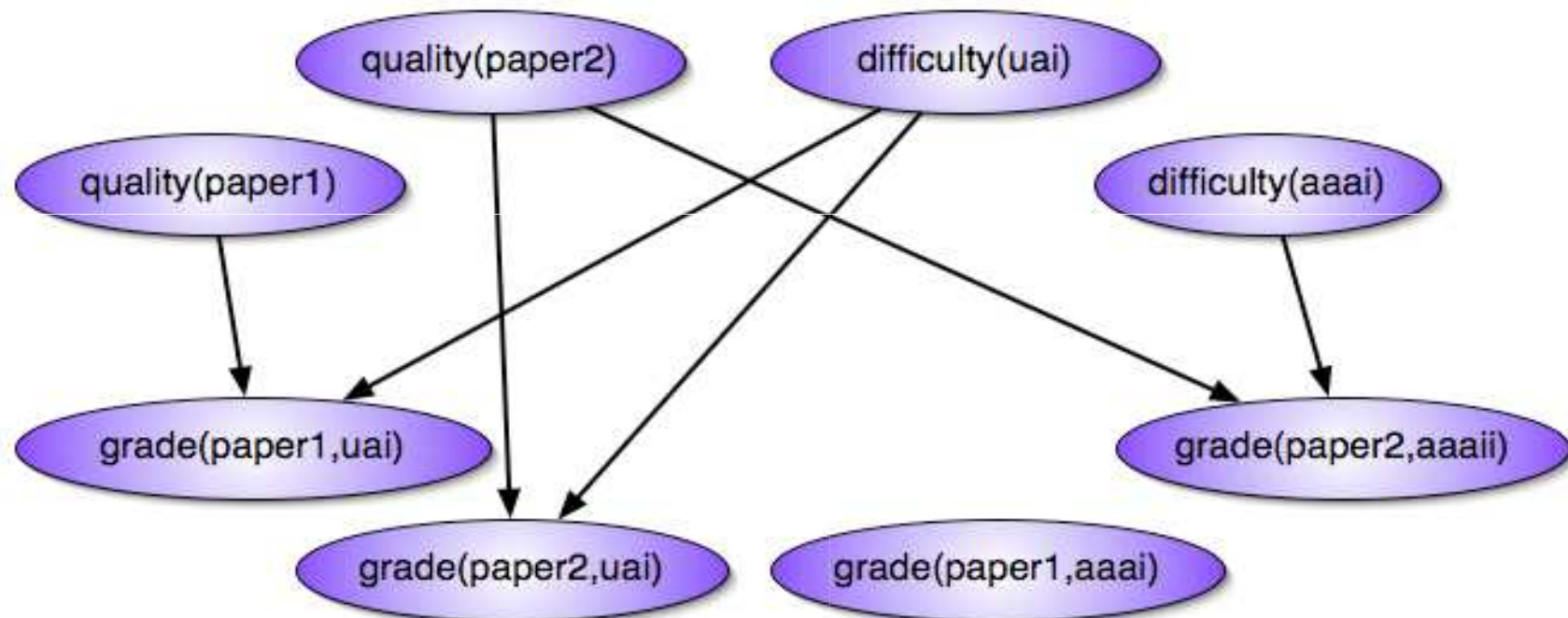
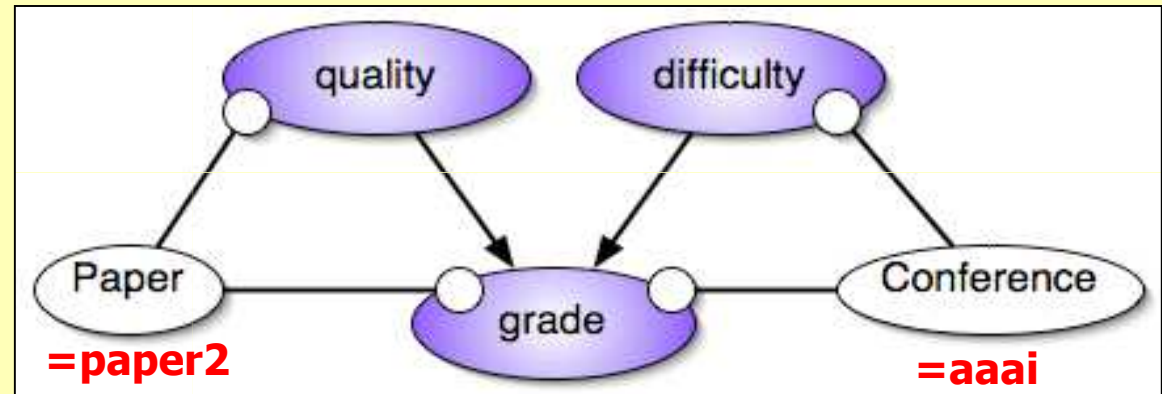
# (Reviewing) Bayesian Logic Programs



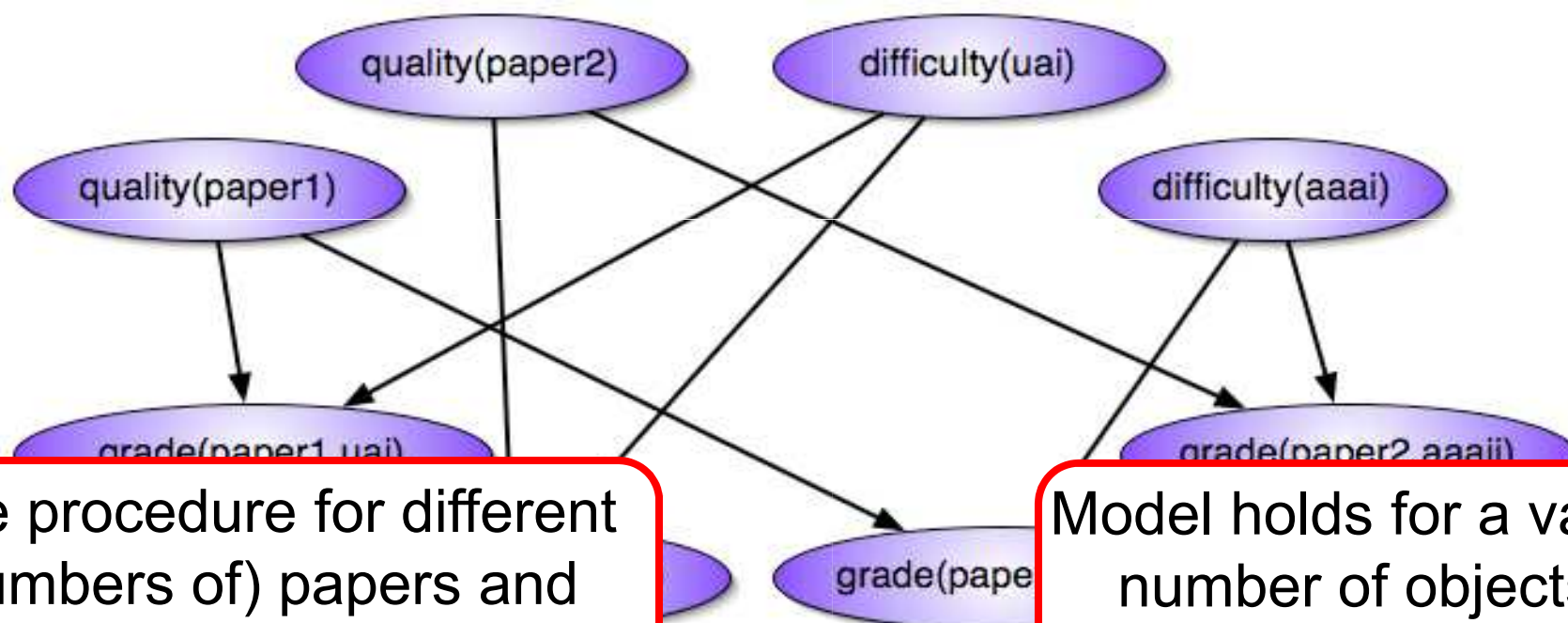
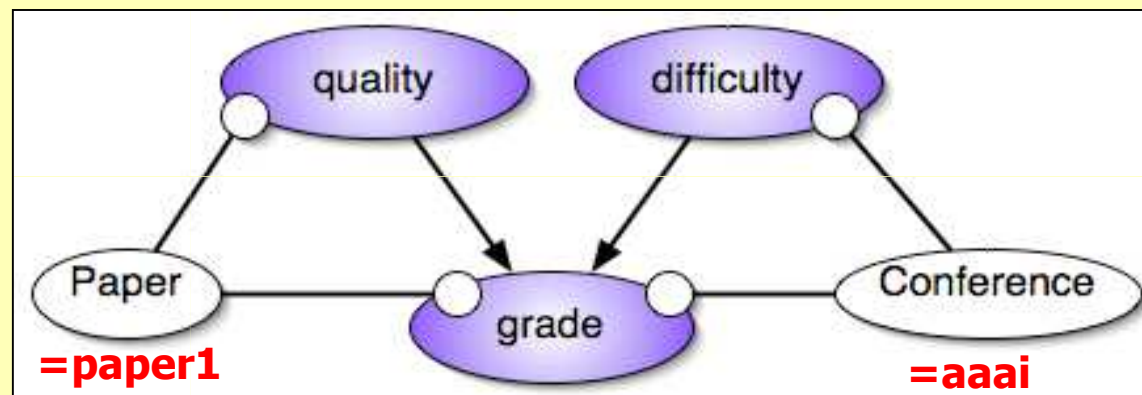
# (Reviewing) Bayesian Logic Programs



# (Reviewing) Bayesian Logic Programs



# (Reviewing) Bayesian Logic Programs

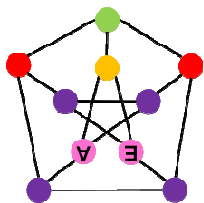


Same procedure for different (numbers of) papers and conference

Model holds for a variable number of objects and relations among objects

# Bayesian Logic

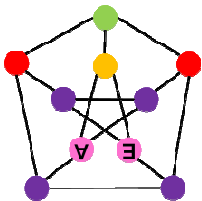
- **Logical language:** First-order logic like
- **Probabilistic language:** Bayesian networks
  - BLOG program specifies how to generate relational world
  - Parameters defined separately in Java functions
  - Allows unknown objects
  - May create Bayesian networks with directed cycles
- **Learning:** None to date
- **Inference:**
  - MCMC with user-supplied proposal distribution
  - Partial grounding





# Markov Logic

- **Logical language:** “First-order” logic
- **Probabilistic language:** Markov networks
  - **Syntax:** First-order formulas with weights
  - **Semantics:** Templates for Markov net features
- **Learning:**
  - **Parameters:** Generative or discriminative
  - **Structure:** ILP with arbitrary clauses and MAP score
- **Inference:**
  - **MAP:** Weighted satisfiability
  - **Marginal:** MCMC with moves proposed by SAT solver
  - Partial grounding + Lazy inference



# Markov Logic

- A **Markov Logic Network (MLN)** is a set of pairs **(F, w)** where

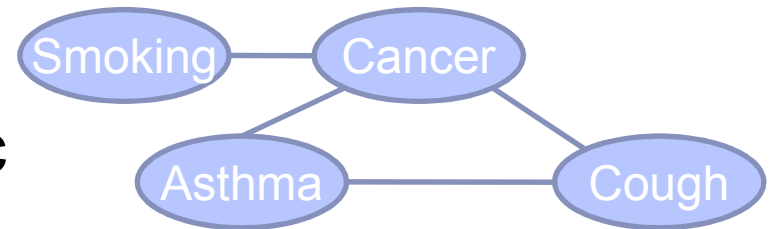
- **F** is a formula in first-order logic
- **w** is a real number

$$P(X) = \frac{1}{Z} \exp \left( \sum_{i \in F} w_i n_i(x) \right)$$

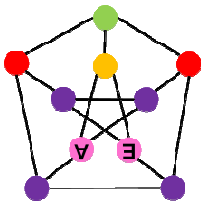
# true groundings of *i*th clause

Normalization constant

Iterate over all first-order MLN formulas



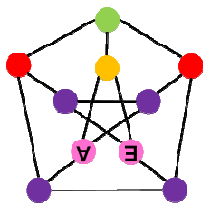
- Together with a finite set of constants, it defines a Markov network with
- Kind of undirected BLPs



---

# Example of First-Order KB

- High quality papers get accepted
- Co-authors are either both smart or both not



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010

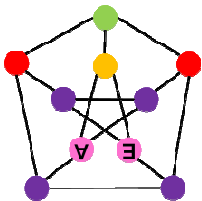


**Fraunhofer**





# Example of First-Order KB

$$\forall x \text{ high\_quality}(p) \Rightarrow \text{accepted}(p)$$
$$\forall x, y \text{ co\_author}(x, y) \Rightarrow (\text{smart}(x) \Leftrightarrow \text{smart}(y))$$


K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



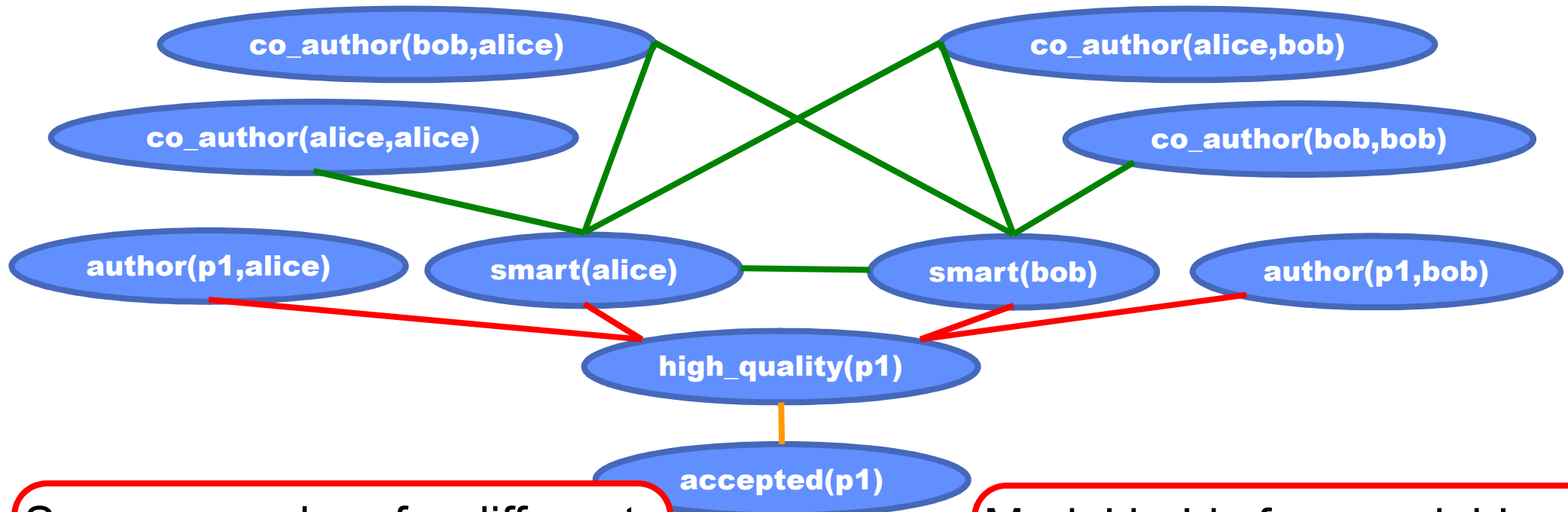
**Fraunhofer**



# Markov Logic

Suppose we have constants: **alice**, **bob** and **p1**

- |          |  |
|----------|--|
| 1.5      | $\forall x \text{ author}(x, p) \wedge \text{smart}(x) \Rightarrow \text{high\_quality}(p)$                  |
| 1.1      | $\forall x \text{ high\_quality}(p) \Rightarrow \text{accepted}(p)$  |
| 1.2      | $\forall x, y \text{ co\_author}(x, y) \Rightarrow (\text{smart}(x) \Leftrightarrow \text{smart}(y))$        |
| $\infty$ | $\forall x, y \exists p \text{ author}(x, p) \wedge \text{author}(y, p) \Rightarrow \text{co\_author}(x, y)$ |

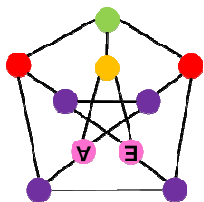


Same procedure for different (numbers of) papers and conference

Model holds for a variable number of objects and relations among objects

# ProbLog

- **Logical language:** Prolog / Horn clauses
- **Probabilistic language:** “Naïve Bayes”
  - Attach probabilities to clauses
- **Learning:** ILP + EM-BDD, Least-squares
- **Inference:**
  - Do all proofs, add probabilities, use BDDs for efficient inference, Abduction, Explanation-based



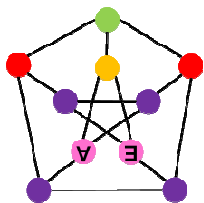
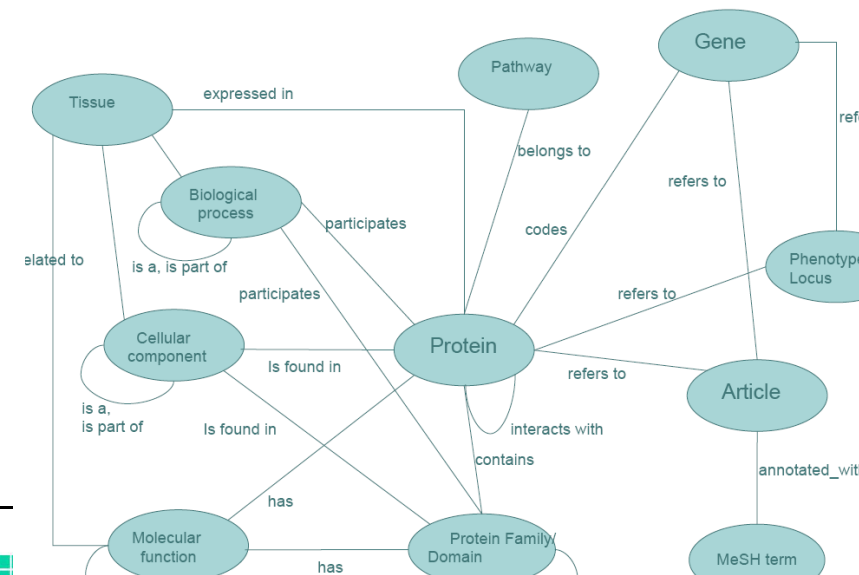
# ProbLog

- **Prolog Clauses + Probability Labels**  $\{p_1 : c_1, \dots, p_n : c_n\}$
- Label of a clause/fact  $c$  indicates the probability that  $c$  belongs to the target program
- Each fact/clause independent of other clauses
- **Defines a distribution over programs**  $L \subseteq \{c_1, \dots, c_n\}$

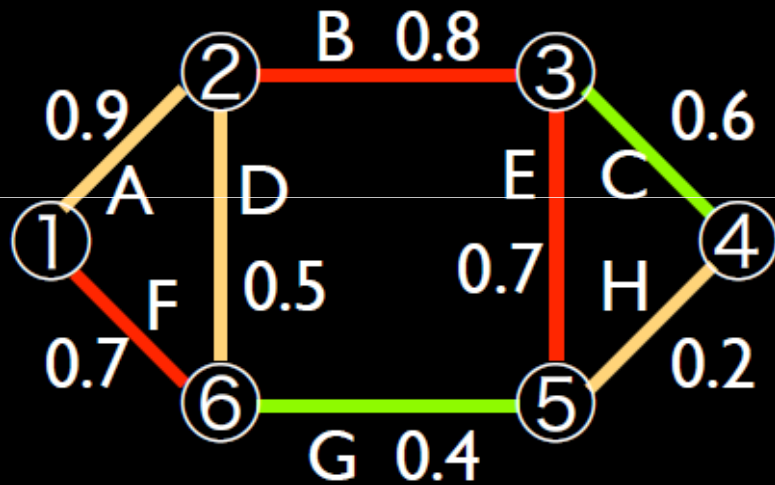
$$P(L|Program) = \prod_{c_i \in L} p_i \prod_{c_j \notin L} (1 - p_j)$$

$$P(q|Program) = \sum_{L \subseteq \{c_1, \dots, c_n\}} P(q|L) \times P(L|Program)$$

$$P(q|L) = \begin{cases} 1 & \text{if } L \models q \\ 0 & \text{otherwise} \end{cases}$$



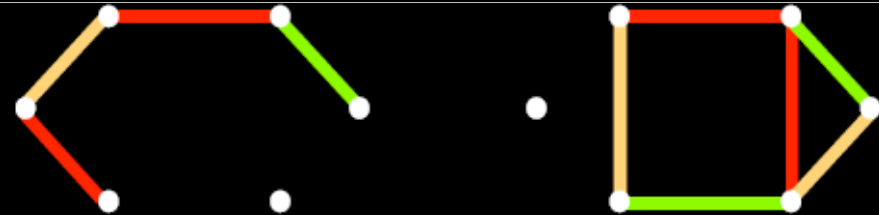
# ProbLog



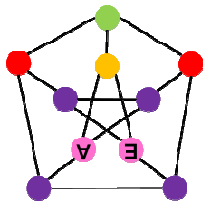
$\text{path}(x,y) \text{ :- edge}(x,y)$   
 $\text{path}(x,y) \text{ :- edge}(x,z), \text{path}(y,z)$

$$P(q|T) = \sum_{S \subseteq L, S \models q} P(S|T)$$

① → ④

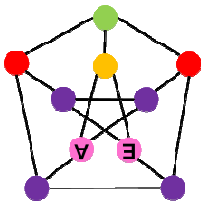


...



# Most common approach to semantics and inference

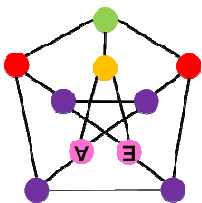
- **Propositionalization** followed by **graphical model inference** respectively **(probabilistic) model checking**
- **Propositionalization:**  
Create all ground atoms and clauses using essentially forward or backward chaining. Can be query directed. There even exists first-order Bayes' ball variants
- **Variable elimination, Belief Propagation, Gibbs Sampling, Weighted (MAX)-SAT, BDD-based, ...**



# Recent Advances in SR Inference

- Preprocessing for Inference: FROG – Shavlik & Natarajan (2009)
- Lifted Exact Inference
  - Lifted Variable Elimination – Poole (2003), Braz et al. (2005), Milch et al. (2008)
  - Lifted VE + Aggregation – Kisynski & Poole (2009)
- Sampling Methods
  - MCMC techniques – Milch & Russell (2006)
  - Logical Particle Filter – Natarajan et al. (2008), Zettlemoyer et al. (2007)
  - Lazy Inference – Poon et al. (2008)
  - MC-SAT – Poon & Domingos (2006)
- Approximate Methods
  - Bi-simulated Lifted Variable Elimination – Sen et al. (2008, 2009)
  - Lifted First-Order Belief Propagation – Singla & Domingos (2008), Kersting et al. (2009, 2010)
  - MAP Inference – Riedel (2008)
  - Formula based – Gogate & Domingos (2010)
- Bounds Propagation: Anytime Lifted Belief Propagation – Braz et al. (2009)

More about this later !



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**

**IAIS universität bonn**  
RHEINISCHE FRIEDRICH-WILHELMS-  
UNIVERSITÄT

---

# Costs and Benefits of the SRL soup

## ▪ Benefits

- Rich pool of different languages
- Very likely that there is a language that fits your task at hand well
- A lot research remains to be done, ;-)

## ▪ Costs

- “Learning” SRL is much harder
- Not all frameworks support all kinds of inference and learning settings

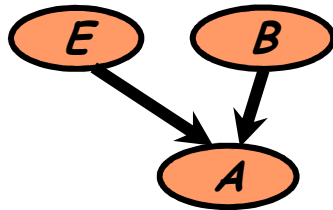
**Quite similar to propositional ones!**

**How do we actually learn relational models from data?**



# Excuse: BN Learning Known Structure, Complete Data

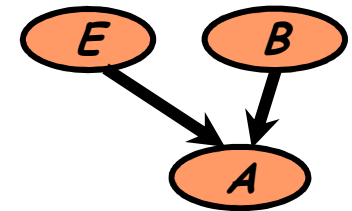
E, B, A
<Y,N,N>
<Y,N,Y>
<N,N,Y>
<N,Y,Y>
.
.
<N,Y,Y>



$E$	$B$	$P(A \mid E, B)$	
$e$	$b$	?	?
$e$	$\bar{b}$	?	?
$\bar{e}$	$b$	?	?
$\bar{e}$	$\bar{b}$	?	?

Learning  
algorithm

- Network structure is specified
  - Learner needs to estimate parameters
- Data does not contain missing values



$E$	$B$	$P(A \mid E, B)$	
$e$	$b$	.9	.1
$e$	$\bar{b}$	.7	.3
$\bar{e}$	$b$	.8	.2
$\bar{e}$	$\bar{b}$	.99	.01

# ML Parameter Estimation

A1	A2	A3	A4	A5	A6
true	true	false	true	false	false
false	true	true	true	false	false
...	...	...	...	...	...
true	false	false	false	true	true

$$\begin{aligned}
 \mathcal{LL}(\Theta|\mathcal{X}) &= \log P(X_1, X_2, \dots, X_n|\Theta) \\
 &\stackrel{\text{(iid)}}{=} \log \prod_{i=1}^n P(X_i|\Theta) \\
 &\stackrel{\log \prod}{=} \sum \log \\
 &= \sum_{i=1}^n \log P(X_i|\Theta) = \sum_{i=1}^n \log P(x_i^1, x_i^2, \dots, x_i^m|\Theta) \\
 &= \sum_{i=1}^n \log \left( \prod_{j=1}^m P(x_i^j | \text{pa}(x_i^j), \Theta) \right) \quad \text{(BN semantics)} \\
 &= \sum_{i=1}^n \sum_{j=1}^m \log P(x_i^j | \text{pa}(x_i^j), \Theta) \\
 &= \sum_{j=1}^m \sum_{i=1}^n \log P(x_i^j | \text{pa}(x_i^j), \Theta_j) \quad \text{Only local parameters of family of } A_j \text{ involved} \\
 &= \sum_{j=1}^m \mathcal{LL}(\Theta_j|\mathcal{X}) \quad \text{Each factor individually !!}
 \end{aligned}$$

# ML Parameter Estimation

$$\mathcal{LL}(\Theta|\mathcal{X}) = \log P(X_1, X_2, \dots, X_n|\Theta)$$

A1	A2	A3	A4	A5	A6
true	true	false	true	false	false
false	true	true	true	false	false
				...	
				true	

**Decomposability** of the likelihood

$$= \sum_{i=1}^n \log \left( \prod_{j=1}^m P(x_i^j | \text{pa}(x_i^j), \Theta) \right) \quad \text{(BN semantics)}$$

$$= \sum_{i=1}^n \sum_{j=1}^m \log P(x_i^j | \text{pa}(x_i^j), \Theta)$$

$$= \sum_{j=1}^m \left( \sum_{i=1}^n \log P(x_i^j | \text{pa}(x_i^j), \Theta_j) \right)$$

$$= \sum_{j=1}^m \mathcal{LL}(\Theta_j|\mathcal{X})$$

Only local parameters of family of  $A_j$  involved

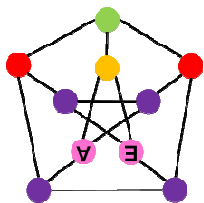
Each factor individually !!

---

## Decomposability of Likelihood

If data set is **complete/fully observed** (i.e. no “?”)

- we can maximize each local likelihood function **independently**, and
- then **combine** the solutions to get an MLE solution
- This **decomposition** of the global problem to independent, local sub-problems allows us to come up with efficient solutions to the MLE problem



## Likelihood for Multinominals

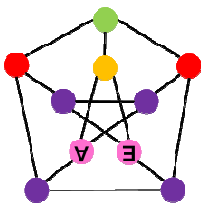
Random variable  $V$  with  $1, \dots, K$  values

$$P(V = k) = \theta_k \quad \sum_{k=1}^K \theta_k = 1$$

This constraint implies that the choice on  $\theta_i$  influences the choice on  $\theta_j$  ( $i \neq j$ )

$$\mathcal{LL}(\Theta_v | \mathcal{X}) = \sum_{k=1}^K \log \theta_k^{N_k} = \sum_{k=1}^K N_k \cdot \log \theta_k$$

where  $N_k$  denotes the number of times we observe state  $k$  in the data (**the counts**)



# Likelihood for Binominals (2 states only)

Compute partial derivative

$$\begin{aligned}\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v | \mathcal{X}) &= \frac{\partial}{\partial \theta_i} (N_1 \log \theta_1 + N_2 \log(1 - \theta_1)) \\ &= \frac{N_1}{\theta_1} + \frac{N_2}{1 - \theta_1}\end{aligned}$$

$$\theta_1 + \theta_2 = 1$$

Set partial derivative zero

$$\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v | \mathcal{X}) = 0 \Leftrightarrow \frac{N_1}{\theta_1} + \frac{N_2}{1 - \theta_1} = 0$$

=> MLE is

$$\theta_1^* = \frac{N_1}{N_1 + N_2}$$

# Likelihood for Multinomials

## Compute partial derivative

$$\begin{aligned}\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v | \mathcal{X}) &= \frac{\partial}{\partial \theta_i} (N_1 \log \theta_1 + N_2 \log(1 - \theta_1)) \\ &= \frac{N_1}{\theta_1} + \frac{N_2}{1 - \theta_1}\end{aligned}$$

$$\theta_1 + \theta_2 = 1$$

## Set partial derivative zero

$$\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v | \mathcal{X}) = 0 \Leftrightarrow \frac{N_1}{\theta_1} + \frac{N_2}{1 - \theta_1} = 0$$

In general, for multinomials (>2 states),  
the MLE is

$$\theta_i^* = \frac{N_i}{\sum_j N_j}$$

# Likelihood for Conditional Multinominals

$P(V = k | \text{pa}(V) = \text{pa})$  multinomial for each joint state  $\text{pa}$  of the parents of  $V$ :

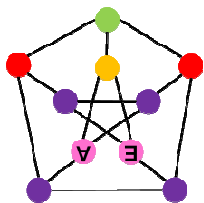
$$P(k|1, 1), P(k|1, 2), P(k|2, 1), P(k|2, 2)$$

$$\mathcal{LL}(\Theta_v | \mathcal{X})$$

$$= \sum_{\text{pa}} \sum_{k=1}^K \log \theta_{k|\text{pa}}^{N_{k,\text{pa}}} = \sum_{\text{pa}} \sum_{k=1}^K N_{k,\text{pa}} \cdot \theta_{k|\text{pa}}$$

MLE

$$\theta_{k|\text{pa}}^* = \frac{N_{k,\text{pa}}}{N_{\text{pa}}}$$



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



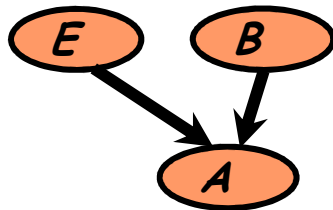
Fraunhofer



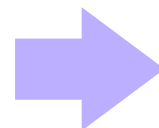


# Known Structure, Incomplete Data

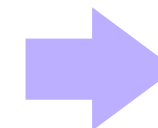
E, B, A
<Y,?,N>
<Y,N,?>
<N,N,Y>
<N,Y,Y>
.
.
<?,Y,Y>



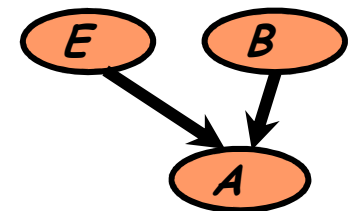
$E$	$B$	$P(A   E, B)$	
$e$	$b$	?	?
$e$	$\bar{b}$	?	?
$\bar{e}$	$b$	?	?
$\bar{e}$	$\bar{b}$	?	?



Learning  
algorithm



- Network structure is specified
- Data contains missing values
  - Need to consider assignments to missing values



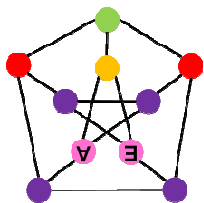
$E$	$B$	$P(A   E, B)$	
$e$	$b$	.9	.1
$e$	$\bar{b}$	.7	.3
$\bar{e}$	$b$	.8	.2
$\bar{e}$	$\bar{b}$	.99	.01

---

# EM Idea

- If **data is complete**, ML parameter estimation is easy:  
**simple counting** (1 iteration)
- But what if there are missing values, i.e., we are facing **incomplete data**?

1. **Complete data** (Imputation)
  - most probable?, average?, ... value
2. **Count**
3. **Iterate**



# EM Idea: complete the data

$$\theta_{A=\text{true}} = \frac{1}{2}$$

$$\theta_{B=\text{true}|A=\text{true}} = \frac{1}{2}$$

$$\theta_{B=\text{true}|A=\text{false}} = \frac{1}{2}$$

**complete**

$$P(B = \text{true} | A = \text{true}) = 0.5$$

$$P(B = \text{true} | A = \text{false}) = 0.5$$

**incomplete data**

A	B
true	true
true	?
false	true
true	false
false	?



**complete data**

expected counts



A	B	N
true	true	1.5
true	false	1.5
false	true	1.5
false	false	0.5

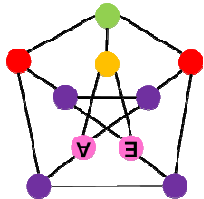
**maximize**

**iterate**

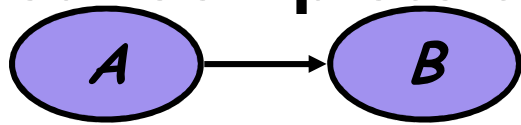
$$\theta_{A=\text{true}} = \frac{1.5+1.5}{1.5+1.5+1.5+0.5} = 0.6$$

$$\theta_{B=\text{true}|A=\text{true}} = \frac{1.5}{1.5+1.5} = 0.5$$

$$\theta_{B=\text{true}|A=\text{false}} = \frac{1.5}{1.5+0.5} = 0.75$$



# EM Idea: complete the data



$$\theta_{A=\text{true}} = 0.6$$

$$\theta_{B=\text{true}|A=\text{true}} = 0.5$$

$$\theta_{B=\text{true}|A=\text{false}} = 0.875$$

**complete**

$$P(B = \text{true} | A = \text{true}) = 0.5$$

$$P(B = \text{true} | A = \text{false}) = 0.875$$

**incomplete data**

A	B
true	true
true	?
false	true
true	false
false	?



**complete data**

expected counts



A	B	N
true	true	1.5
true	false	1.5
false	true	1.875
false	false	0.125

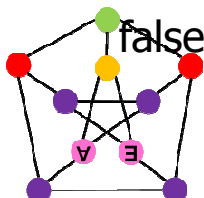
**maximize**

**iterate**

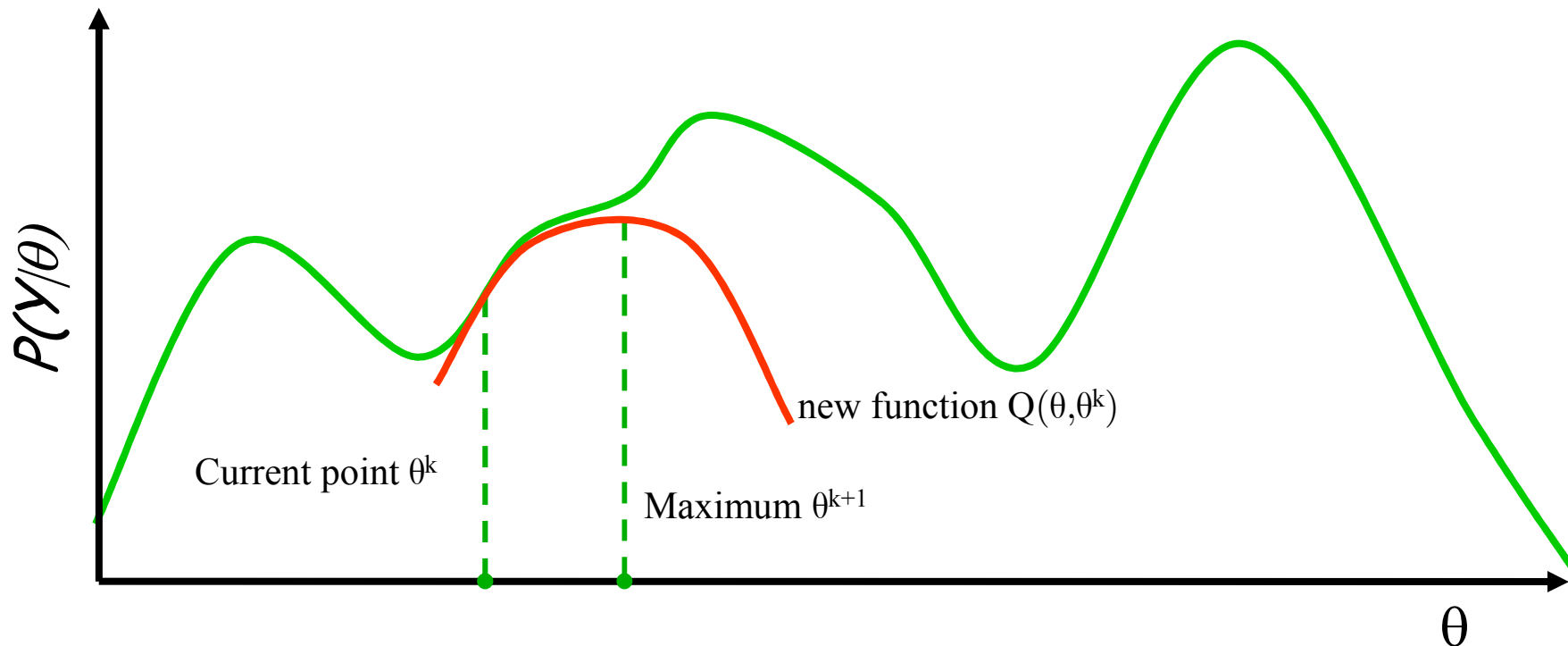
$$\theta_{A=\text{true}} = \frac{1.5+1.5}{1.5+1.5+1.875+0.125} = 0.6$$

$$\theta_{B=\text{true}|A=\text{true}} = \frac{1.5}{1.5+1.5} = 0.5$$

$$\theta_{B=\text{true}|A=\text{false}} = \frac{1.875}{1.875+0.125} = 0.9375$$



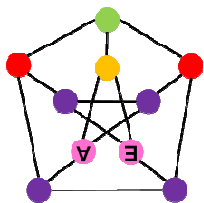
# EM Algorithm Principle



## Expectation Maximization (EM):

Construct a new function based on the “current point” (which “behaves well”)

Property: The maximum of the new function has a better scoring than the current point.



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



## EM for Multinominals

Random variable  $V$  with  $1, \dots, K$  values

$$P(V = k) = \theta_k \quad \sum_{k=1}^K \theta_k = 1$$

$$Q(\Theta_v, \Theta') = \sum_{k=1}^K \log \theta_k^{EN_k} = \sum_{k=1}^K \log EN_k \cdot \theta_k$$

where  $EN_k$  are the **expected counts** of state  $k$  in the data, i.e.

$$EN_k = \sum_{i=1}^m P(k|X_i)$$

„MLE“:

$$\frac{EN_i}{\sum_k EN_k}$$

## EM for Conditional Multinomials

$P(V = k | \text{pa}(V) = \text{pa})$  multinomial for each joint state  $\text{pa}$  of the parents of  $V$ :

$$P(k|1, 1), P(k|1, 2), P(k|2, 1), P(k|2, 2)$$

$$Q(\Theta_v, \Theta')$$

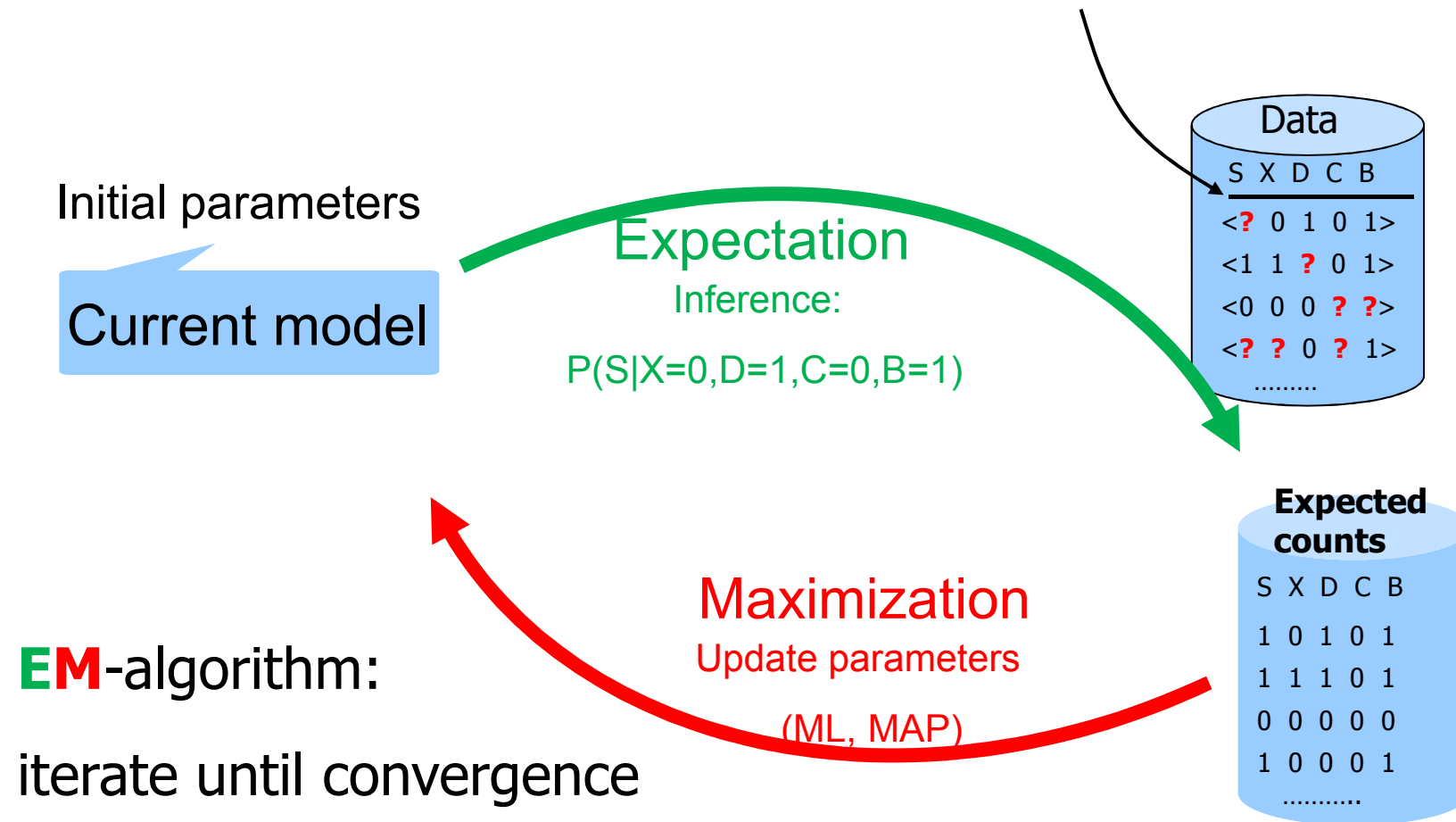
$$= \sum_{\text{pa}} \sum_{k=1}^K \log \theta_{k|\text{pa}}^{EN_{k,\text{pa}}} = \sum_{\text{pa}} \sum_{k=1}^K EN_{k,\text{pa}} \cdot \theta_{k|\text{pa}}$$

„MLE“

$$\theta_{k|\text{pa}}^* = \frac{EN_{k,\text{pa}}}{EN_{\text{pa}}}$$



## Non-decomposable likelihood (missing value, hidden nodes)

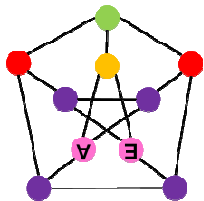
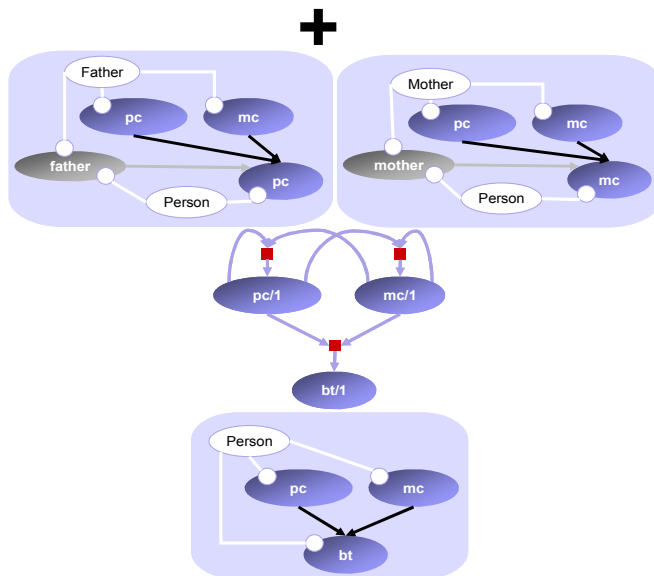
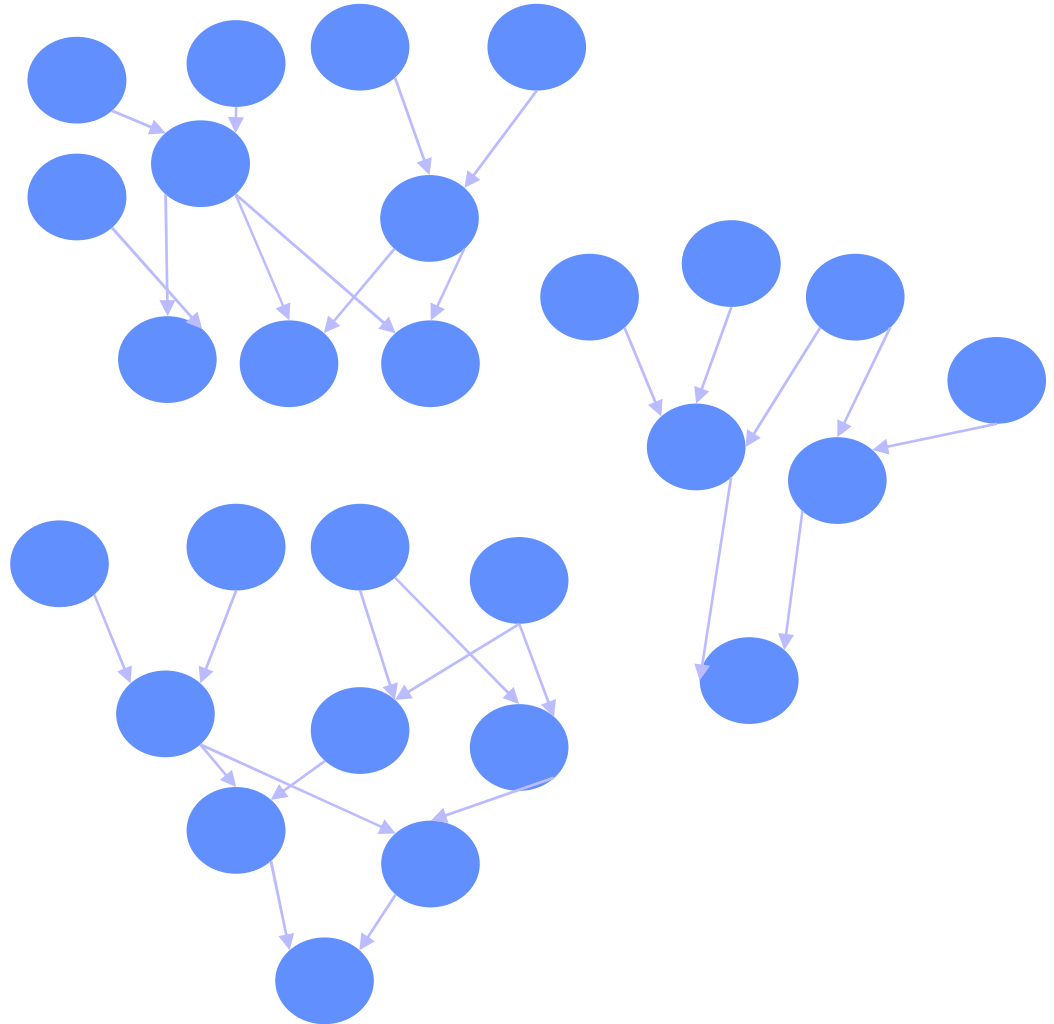
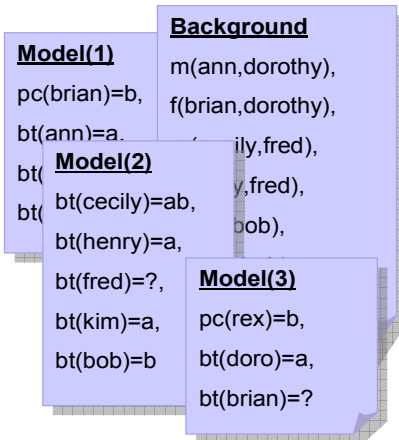


$$\sum_{i=1}^m P(k, \text{pa} | X_i)$$

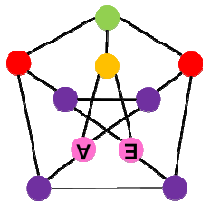
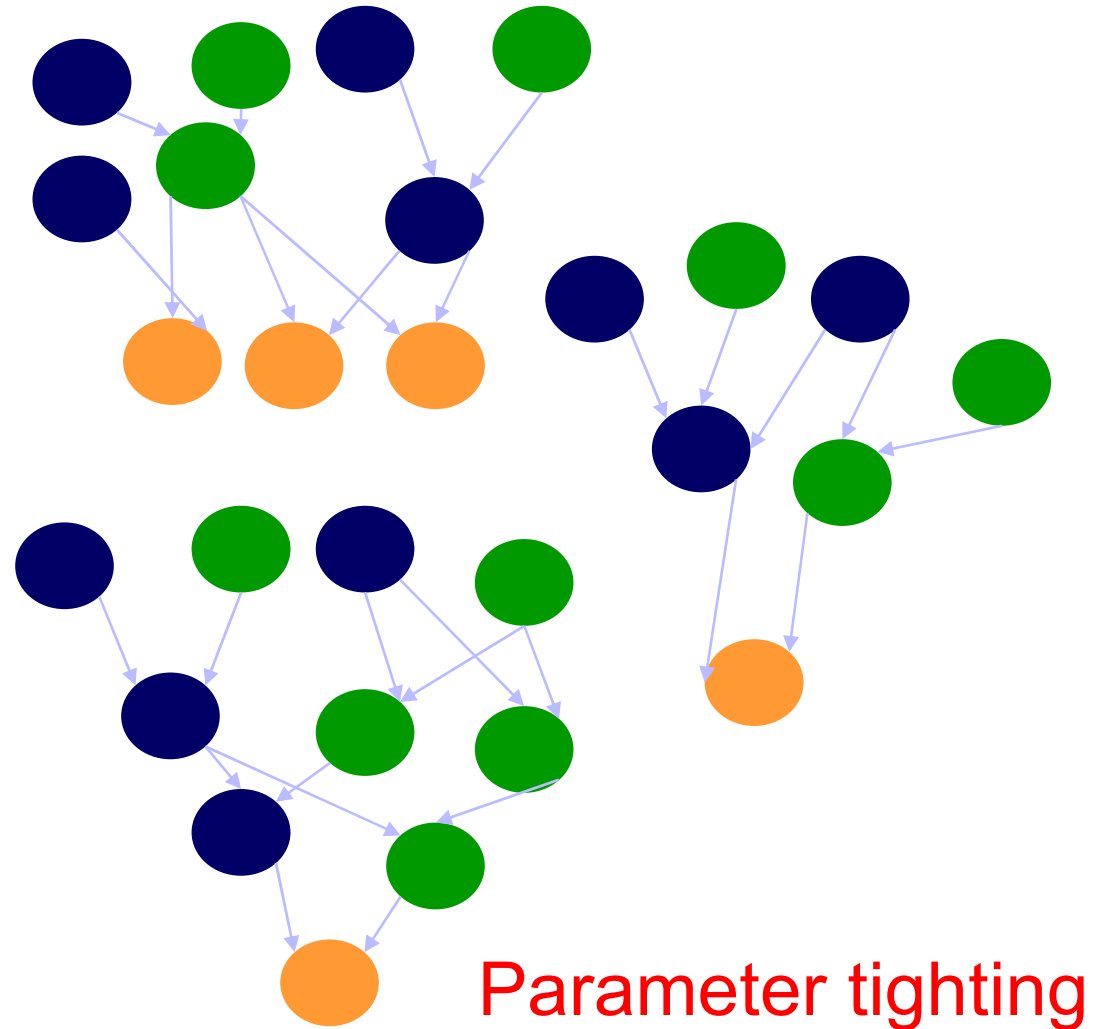
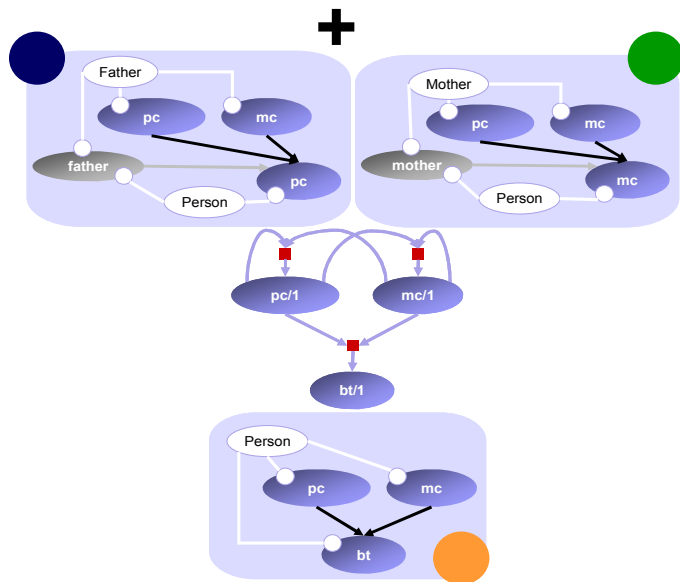
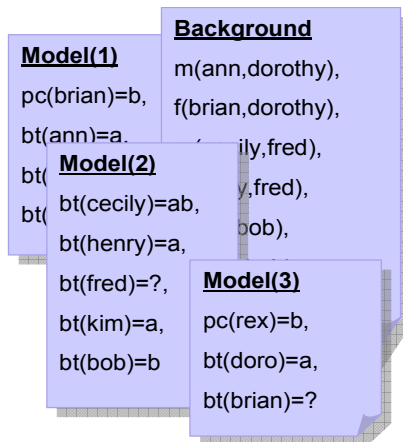
OK, but how do we finally do relational PE?



# Relational Parameter Estimation

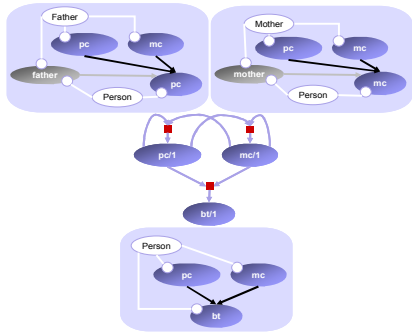


# Relational Parameter Estimation



# So, apply „standard“ EM

## Logic Program L



## Initial Parameters $\theta_0$

**Current Model**  
( $M, \theta_k$ )

iterate until convergence

Expectation

Inference

Expected counts of a clause

$$\sum_{\text{Ground Instance } GI} \sum_{\text{DataCase } DC} P(\text{head}(GI), \text{body}(GI) \mid DC)$$

$$\sum_{\text{Ground Instance } GI} \sum_{\text{DataCase } DC} P(\text{head}(GI), \text{body}(GI) \mid DC)$$

$$\sum_{\text{Ground Instance } GI} \sum_{\text{DataCase } DC} P(\text{body}(GI) \mid DC)$$

Maximization

Update parameters (ML, MAP)

But how do we select a model ?

### Model(1)

pc(brian)=b,  
bt(ann)=a,

Model(2)  
bt(cecily)=ab,  
bt(henry)=a,

bt(fred)=?,  
bt(kim)=a,  
bt(bob)=b

### Background

m(ann,dorothy),  
f(brian,dorothy),  
f(cecily,fred),

bt(cecily)=ab,  
bt(henry)=a,  
bt(bob)=b,

### Model(3)

pc(rex)=b,  
bt(doro)=a,  
bt(brian)=?

Variants exists! Combining Rules, Generative, discriminative, max-margin, ...

# Relational Model Selection / Structure Learning

## ILP= Machine Learning + Logic Programming

[Muggleton, De Raedt JLP96]

Find set of general rules

mutagenic(X) :- atom(X,A,c),charge(X,A,0.82)

mutagenic(X) :- atom(X,A,n),...

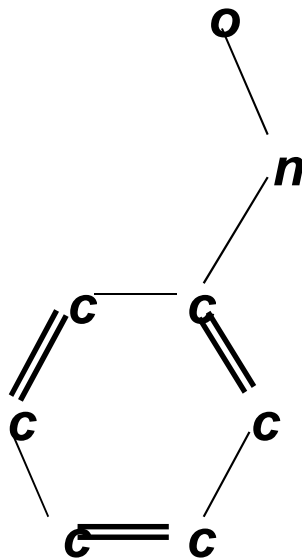
Examples E

pos(mutagenic(m<sub>1</sub>))

neg(mutagenic(m<sub>2</sub>))

pos(mutagenic(m<sub>3</sub>))

...



### Background Knowledge B

molecule(m<sub>1</sub>)

molecule(m<sub>2</sub>)

atom(m<sub>1</sub>,a<sub>11</sub>,c)

atom(m<sub>2</sub>,a<sub>21</sub>,o)

atom(m<sub>1</sub>,a<sub>12</sub>,n)

atom(m<sub>2</sub>,a<sub>22</sub>,n)

bond(m<sub>1</sub>,a<sub>11</sub>,a<sub>12</sub>)

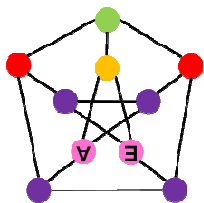
bond(m<sub>2</sub>,a<sub>21</sub>,a<sub>22</sub>)

charge(m<sub>1</sub>,a<sub>11</sub>,0.82)

charge(m<sub>2</sub>,a<sub>21</sub>,0.82)

...

...



# Example ILP Algorithm: FOIL [Quinlan MLJ 5:239-266, 1990]

mutagenic(X) :- atom(X,A,n),charge(A,0.82)

0

mutagenic(X) :- atom(X,A,c),bond(A,B)

$\vee 1 \equiv 1$

$\vee \dots$

:- atom(X,A,c)

Coverage = 0.5, 0.7

:- atom(X,A,c),bond(A,B)

Coverage = 0.8

:- atom(X,A,n)

Coverage = 0.6, 0.3

:- atom(X,A,n),charge(A,0.82)

Coverage = 0.6

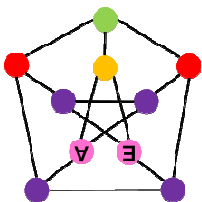
:- true

:- atom(X,A,f)

Coverage = 0.4, 0.6

Some objective function, e.g.

percentage of covered positive examples



## Vanilla SRL [De Raedt, K ALT04]

mutagenic(X) :- atom(X,A,n),charge(A,0.82)

mutagenic(X) :- atom(X,A,c),bond(A,B)

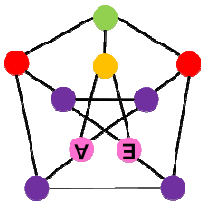
=0.882

...

- Traverses the hypotheses space a la ILP
- Replaces ILP's 0-1 covers relation by a “smooth”, probabilistic one  $[0,1]$

$$\text{cover}(e, H, B) = P(e|H, B)$$

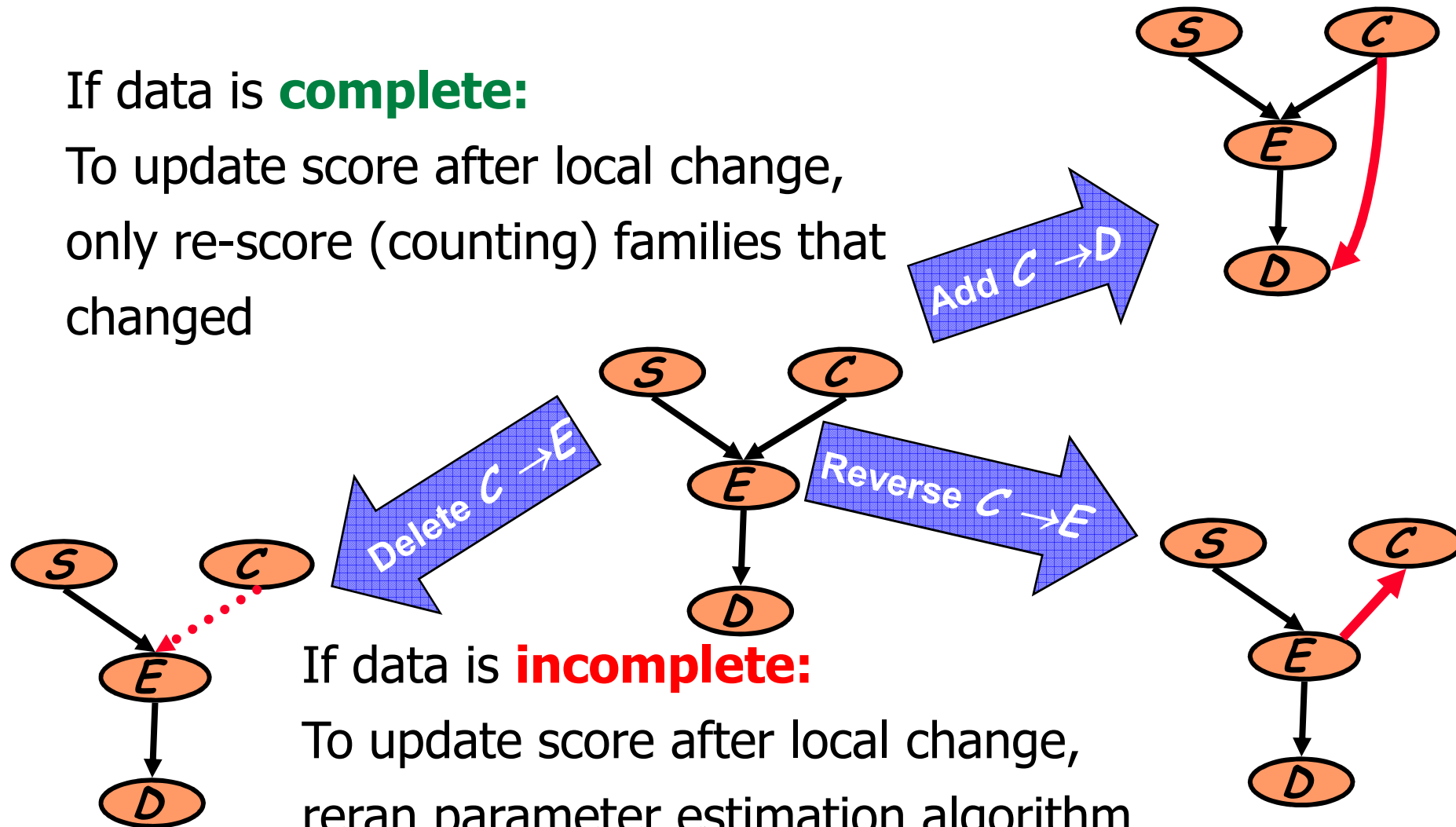
$$\text{cover}(E, H, B) = \prod_{e \in E} \text{cover}(e, H, B)$$



## So, essentially like in the propositional case !

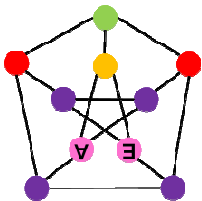
If data is **complete**:

To update score after local change,  
only re-score (counting) families that  
changed

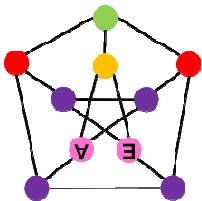
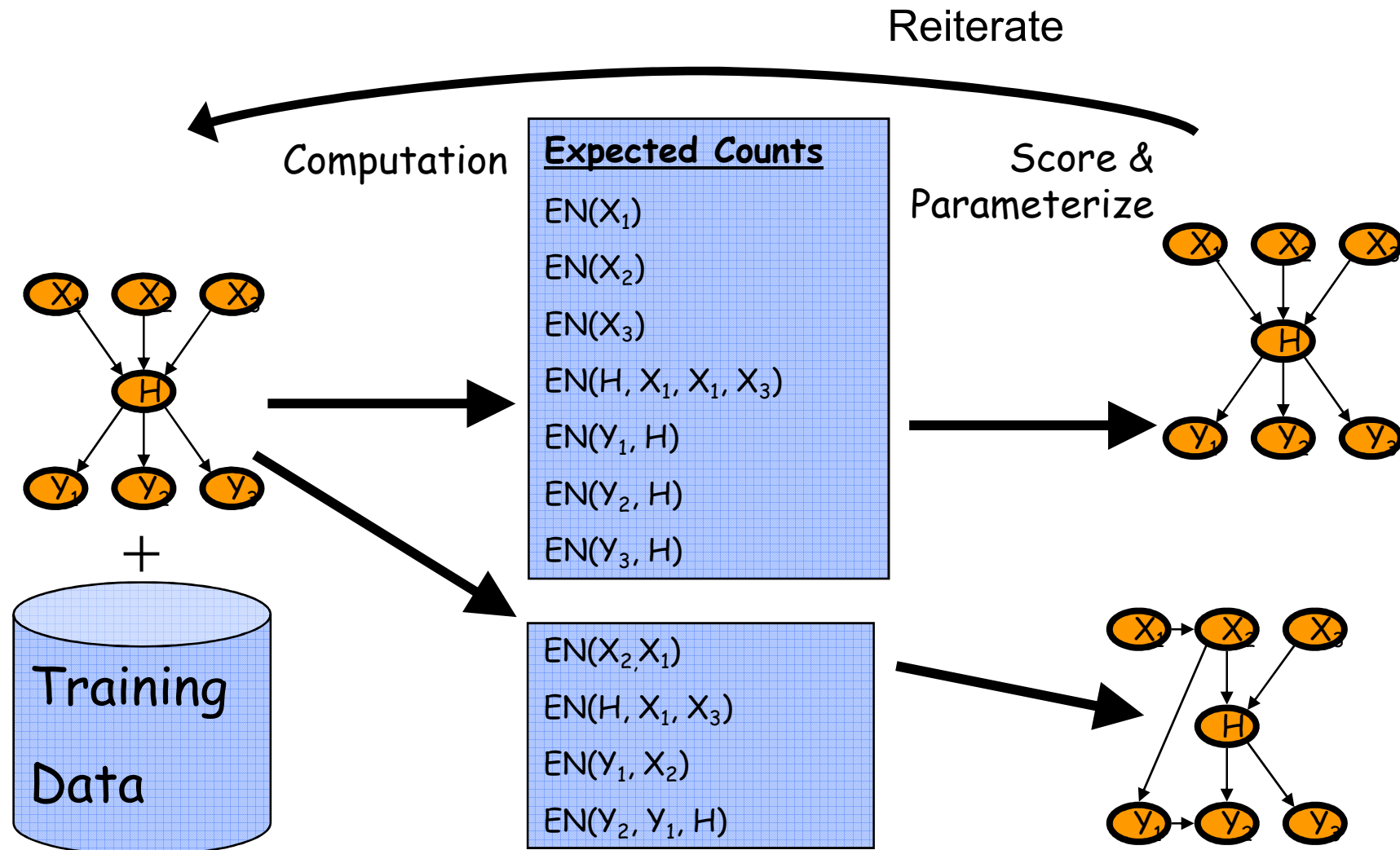


If data is **incomplete**:

To update score after local change,  
reran parameter estimation algorithm



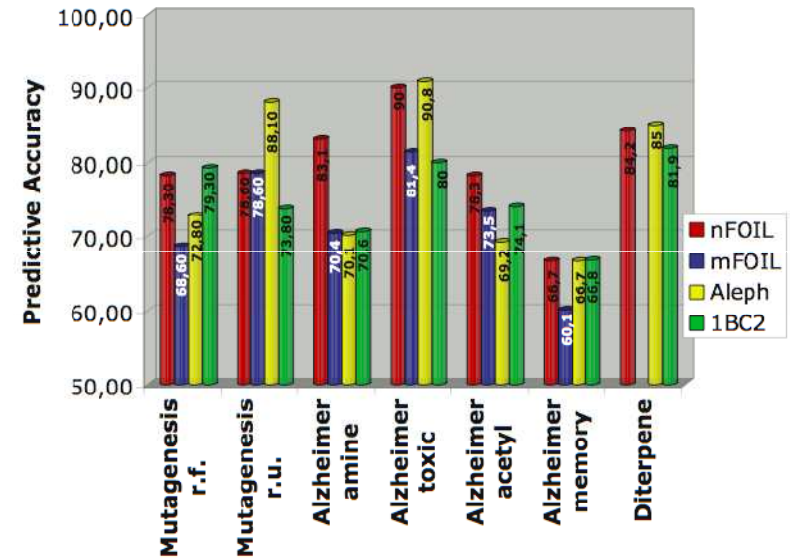
# Structural EM [Friedman et al. 98]





## nFOIL = FOIL + Naive Bayes

- Clauses are independent features
- Likelihood for parameter estimation
- Conditional likelihood for scoring clauses



atom(X,A,n),charge(A,0.82)

atom(X,A,c),bond(A,B)

mutagenic(X)

...

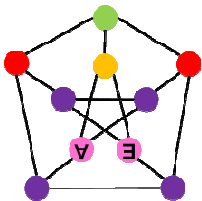
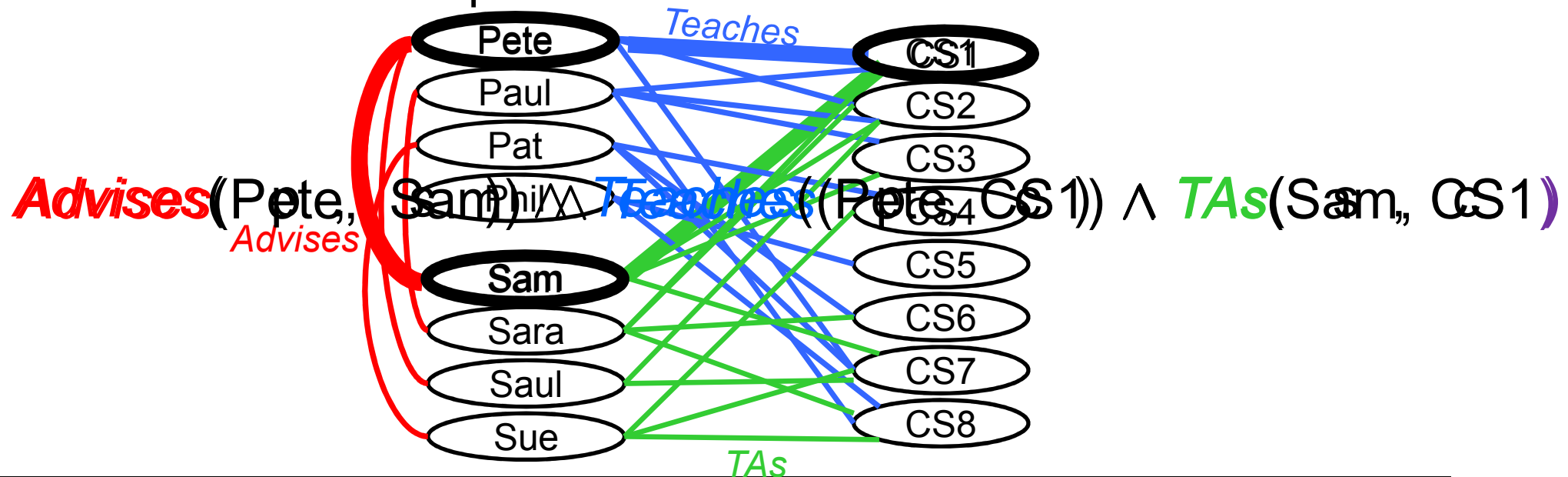
$P(\text{truth value clauses} | \text{truth value target predicate}) \times P(\text{truth value target predicate})$

Let's have a look at bottom-up, i.e. data-driven approaches

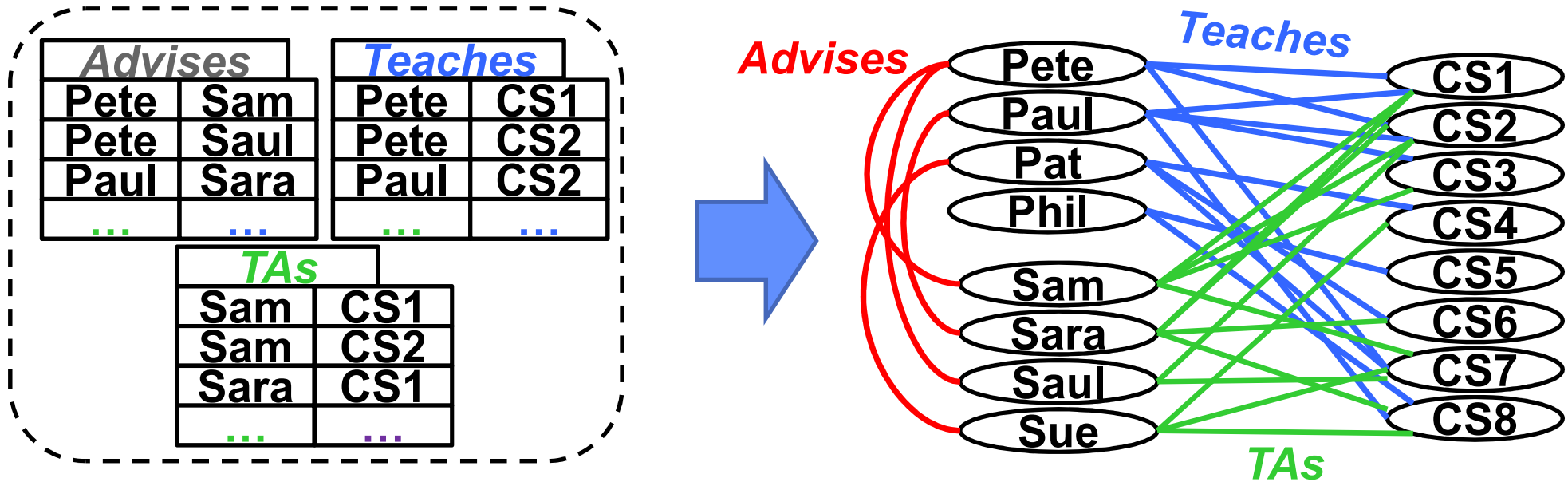
Several variants exists! Top-down, bottom-up, boosting, transfer learning, among others

# Relational Pathfinding [Richards & Mooney, AAAI'92]

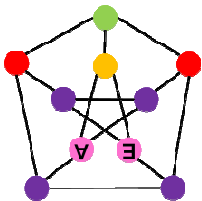
- Find paths of linked ground atoms  $\rightarrow$  formulas
- Path  $\equiv$  conjunction that is true at least once
- Exponential search space of paths
- Restricted to short paths



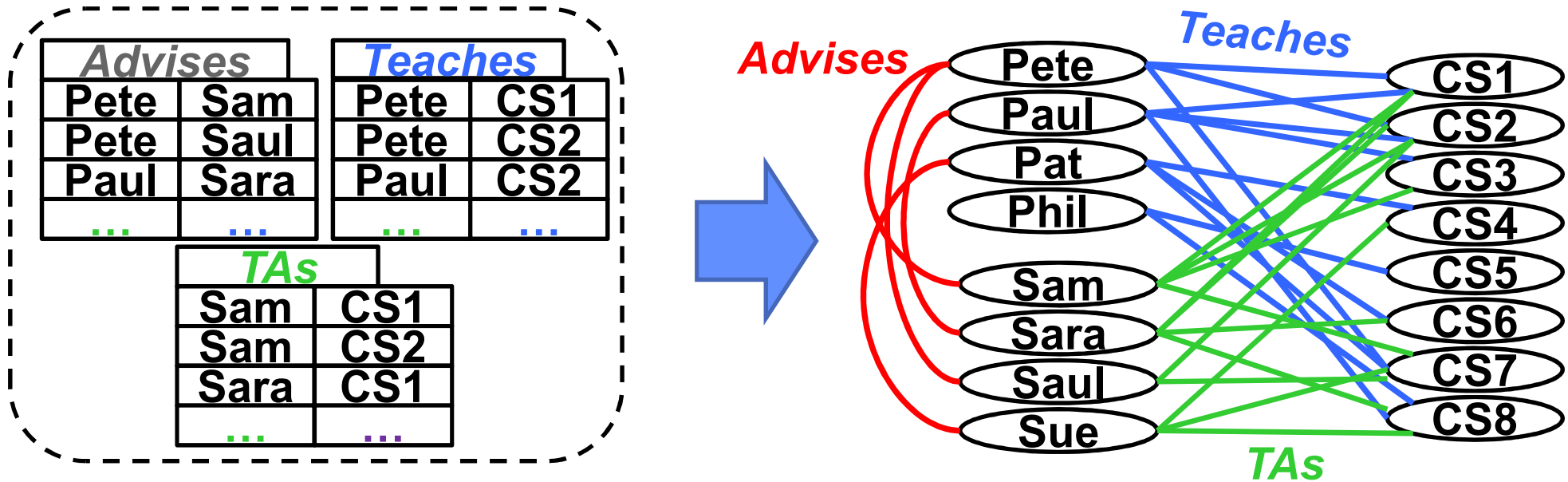
# Learning via Hypergraph Lifting [Kok & Domingos, ICML'09]



- Relational DB can be viewed as hypergraph
  - Nodes  $\equiv$  Constants
  - Hyperedges  $\equiv$  True ground atoms

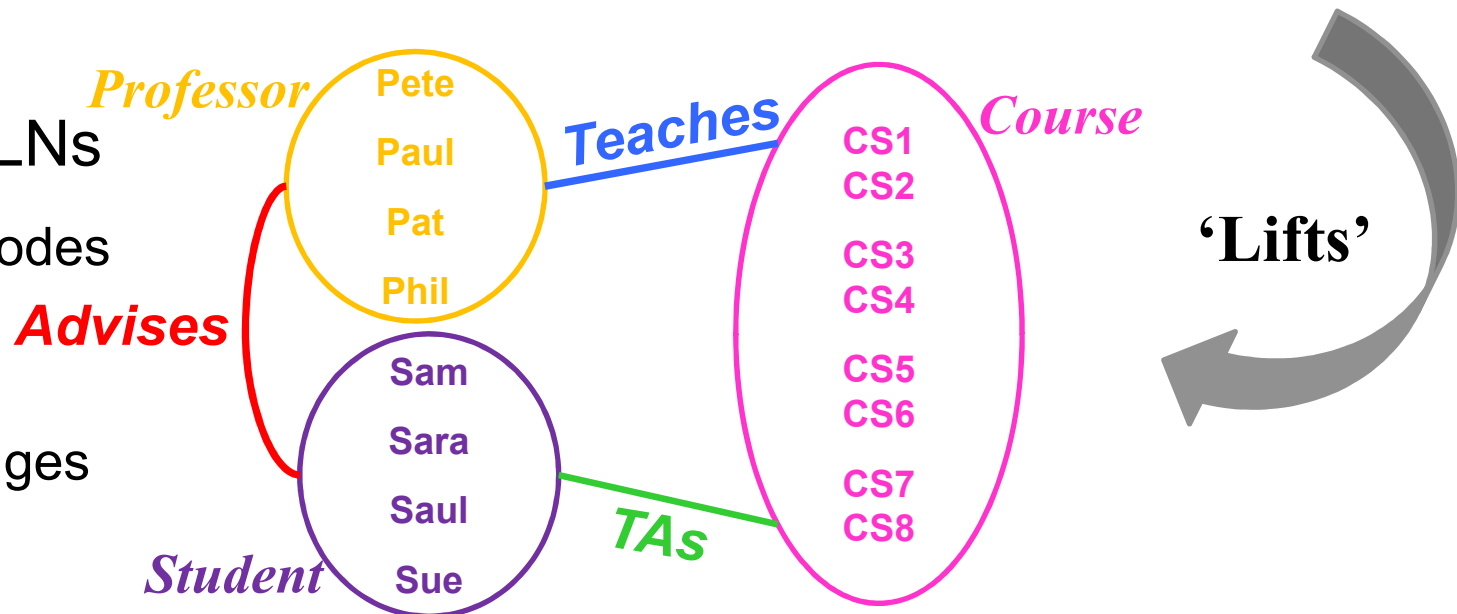


# Learning via Hypergraph Lifting [Kok & Domingos, ICML'09]

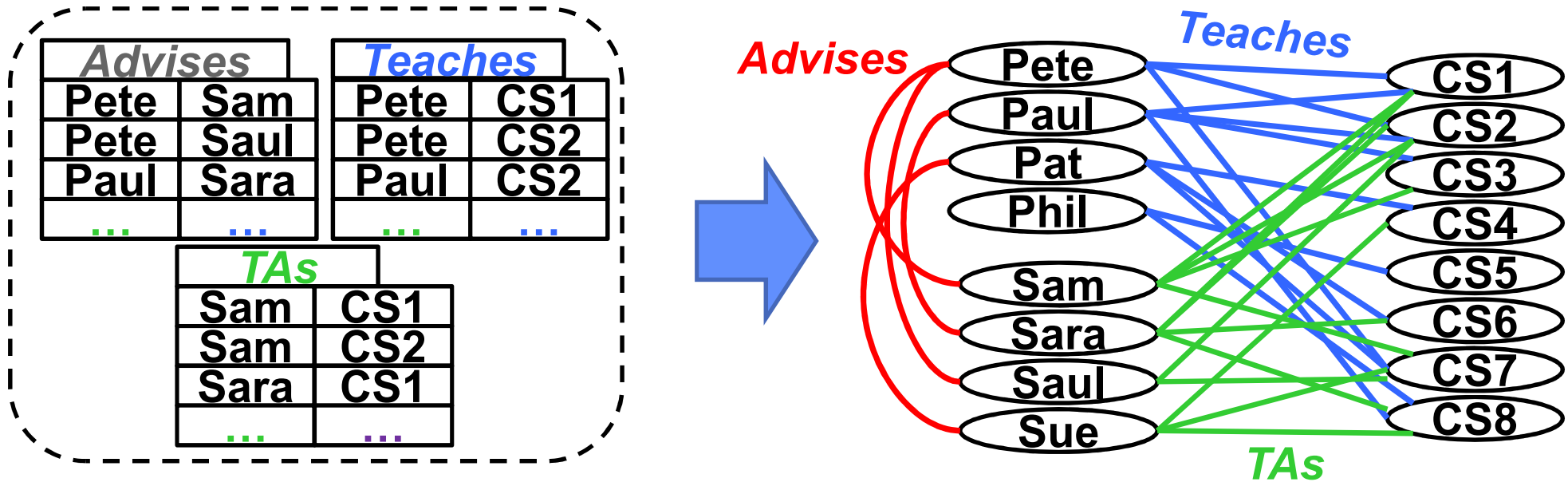


Using “2<sup>nd</sup>”-order MLNs

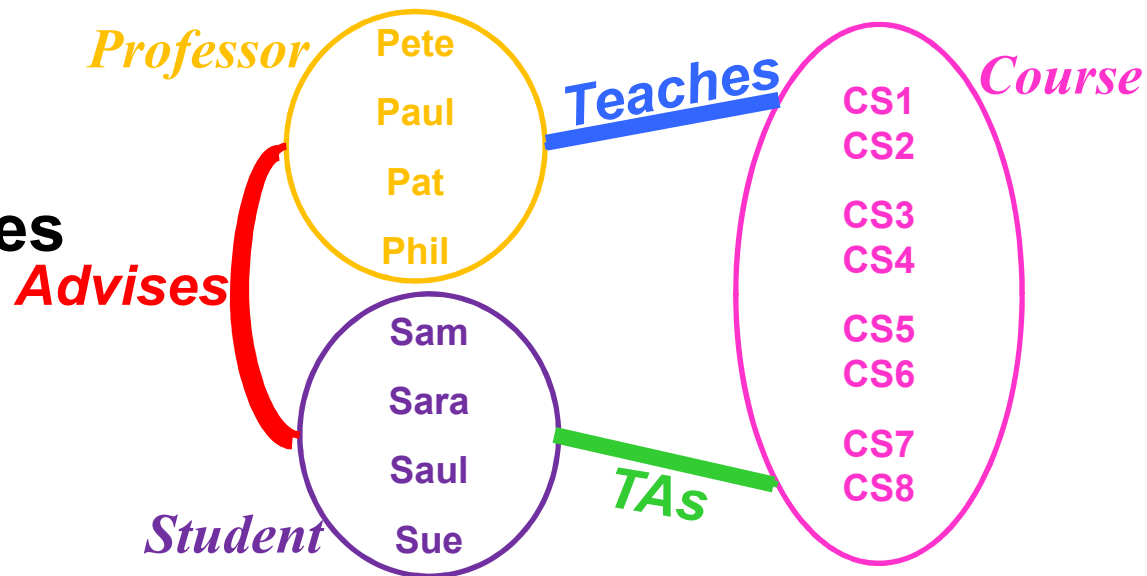
- Jointly clusters nodes into higher-level concepts
- Clusters hyperedges



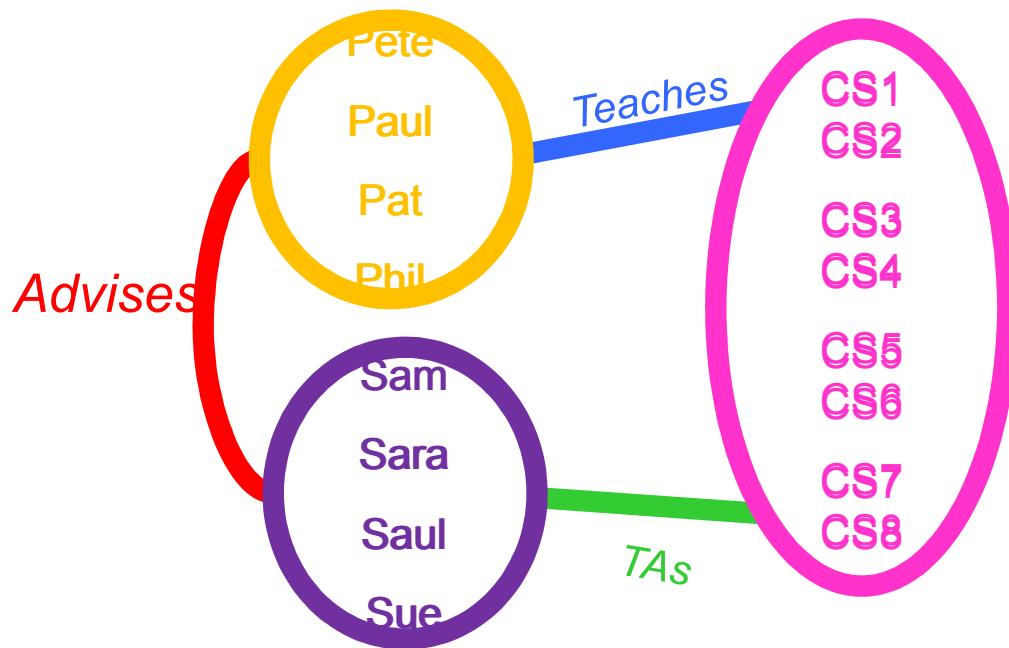
# Learning via Hypergraph Lifting [Kok & Domingos, ICML'09]



Trace paths &  
convert paths to  
first-order clauses



# FindPaths



## Paths Found

*Advises*(, )

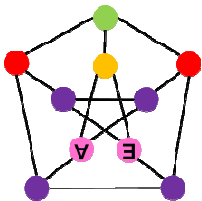
*Advises*(, ) ,

*Teaches* (, )

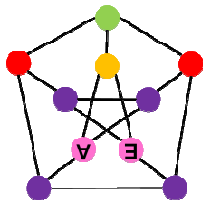
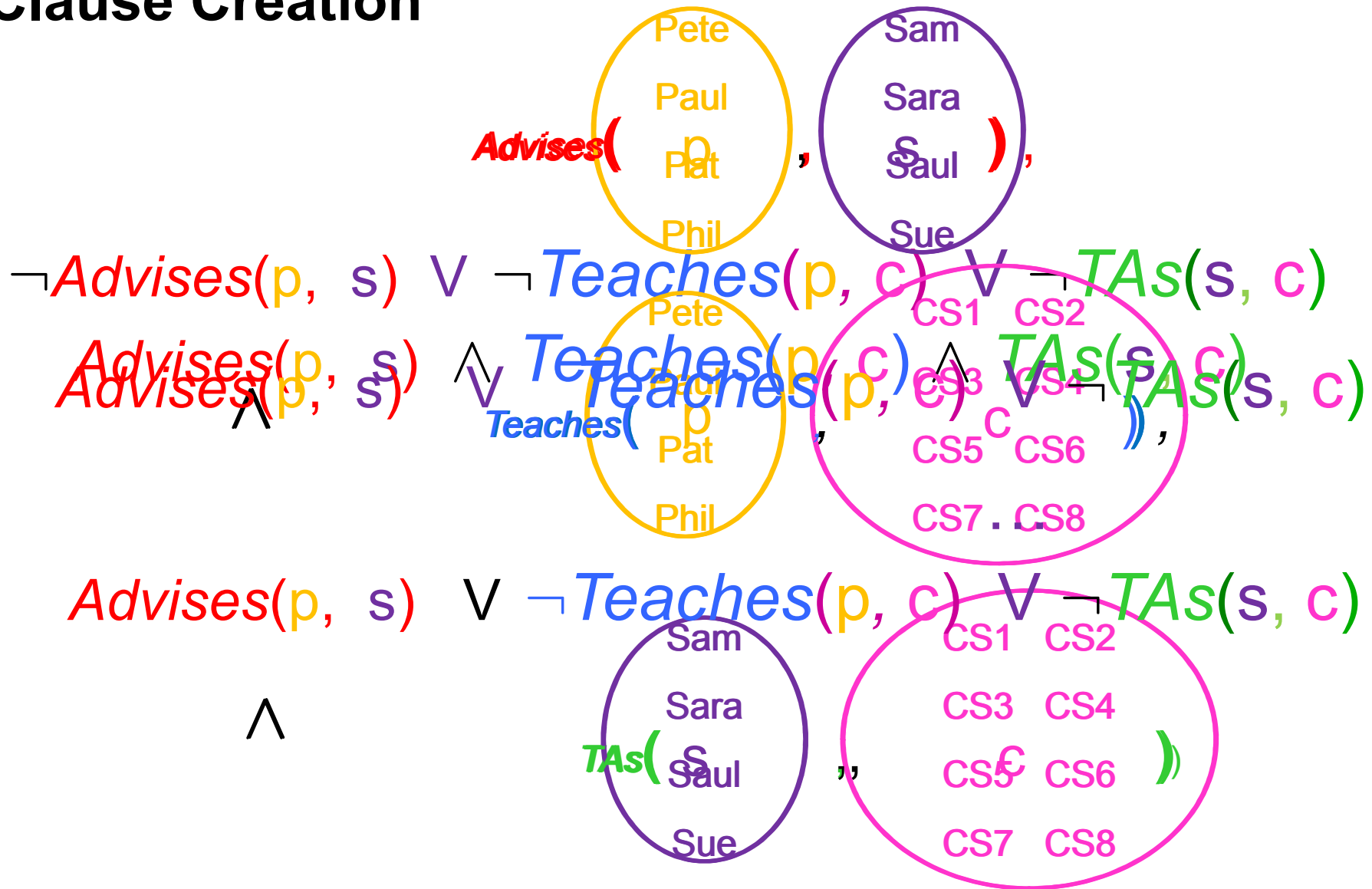
*Advises*(, ) ,

*Teaches* (, ) ,

*TAs*( , )

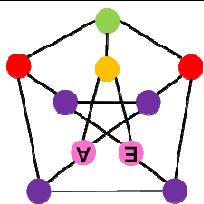
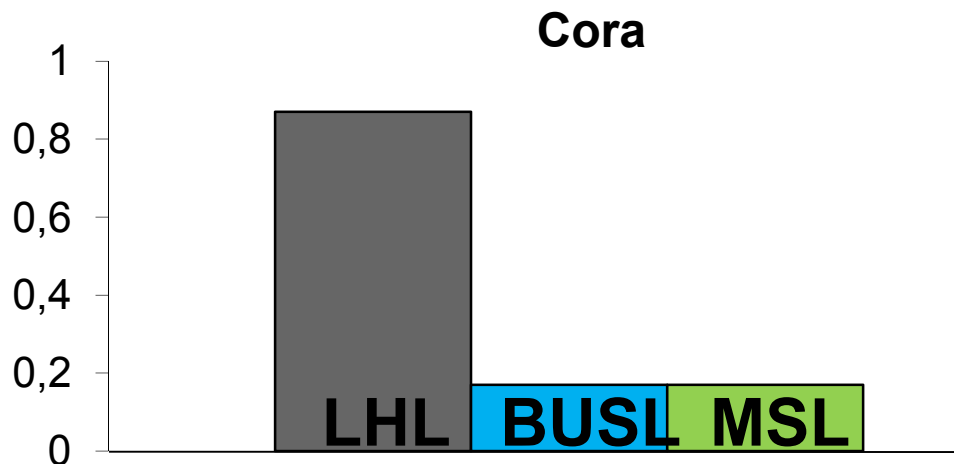
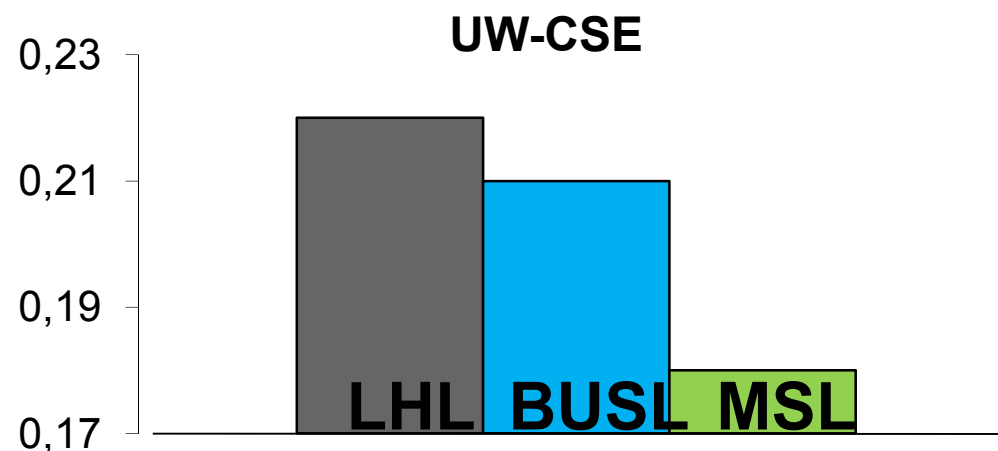
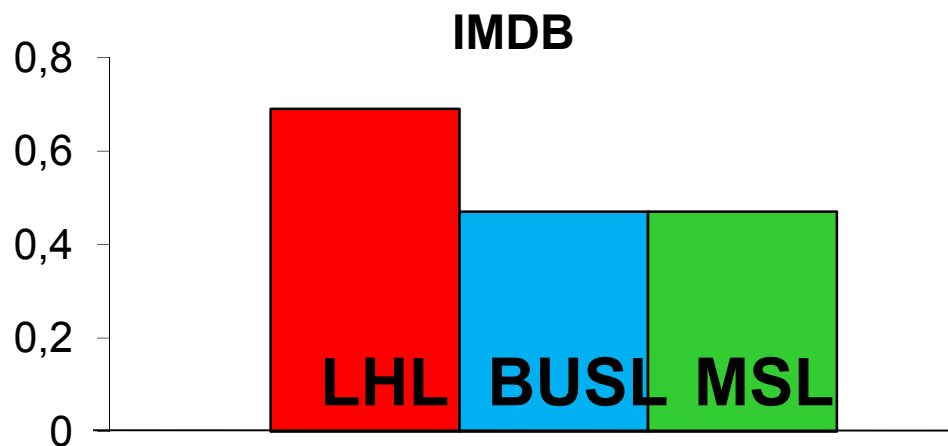


# Clause Creation



## LHL vs. BUSL vs. MSL

# Area under Prec-Recall Curve

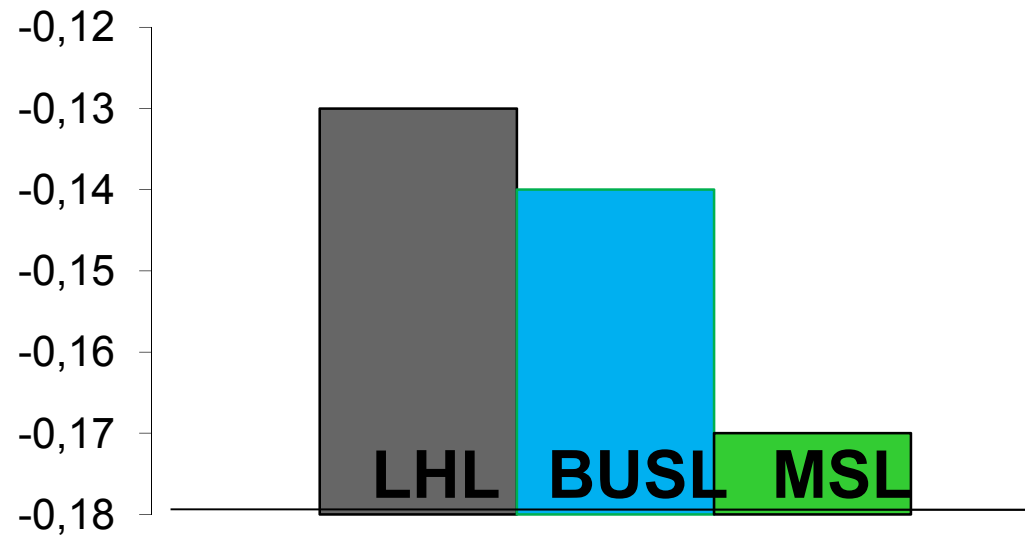




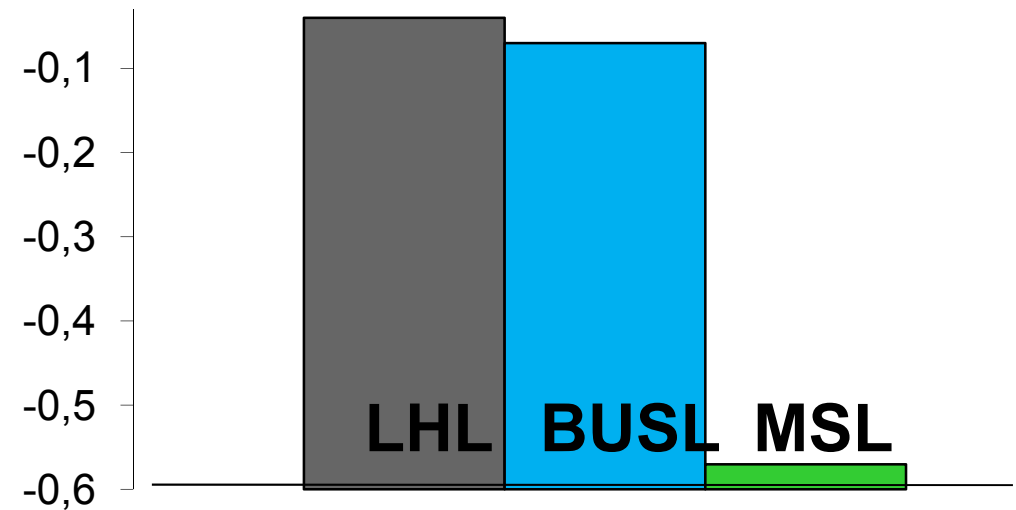
## LHL vs. BUSL vs. MSL

# Conditional Log-likelihood

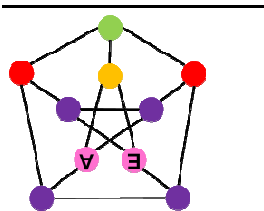
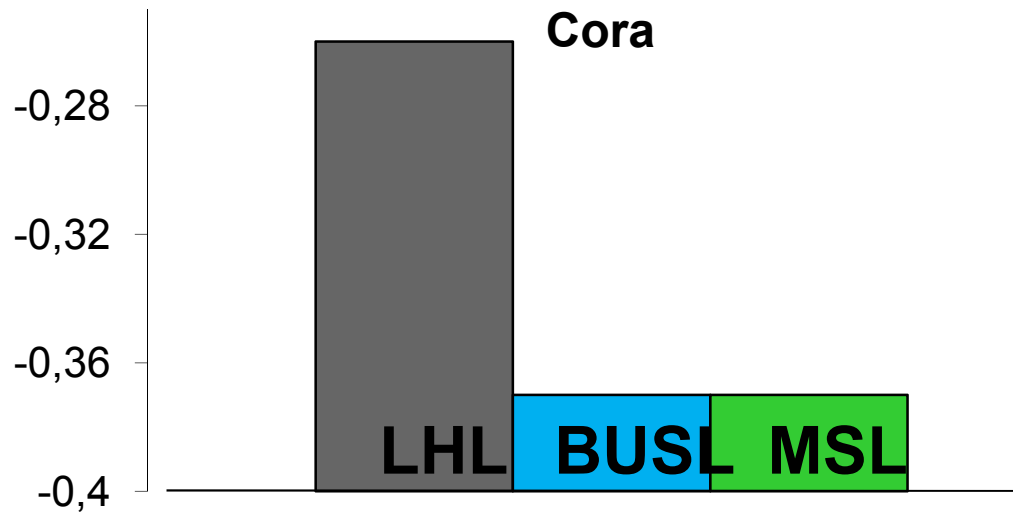
IMDB



UW-CSE

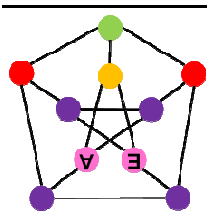
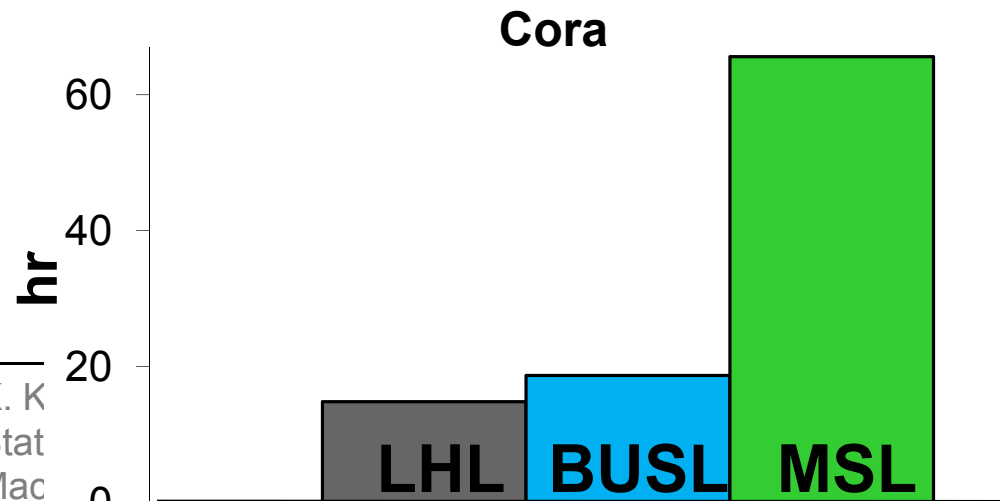
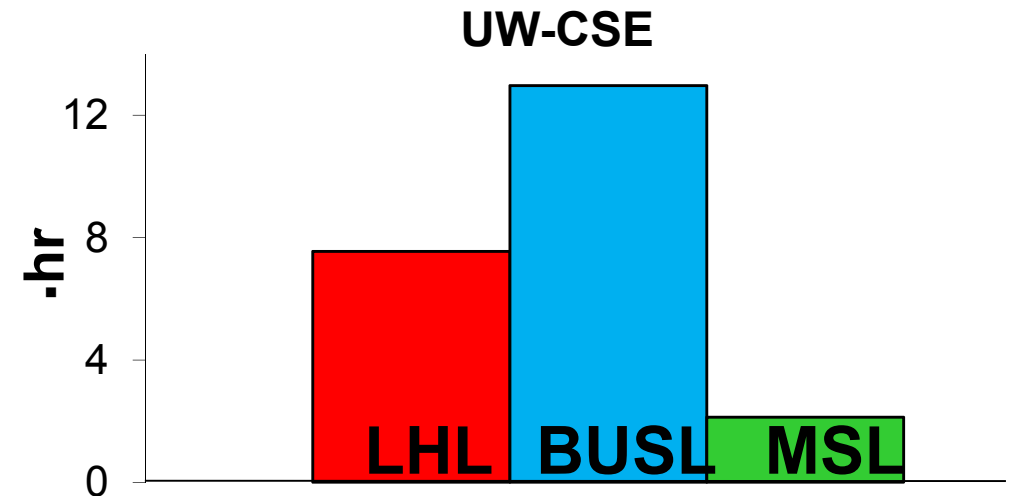
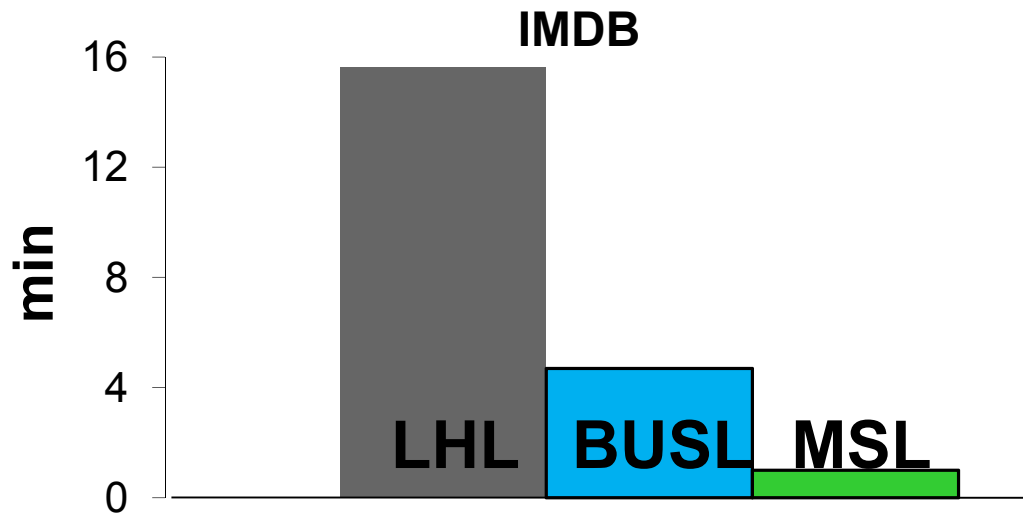


Cora



# LHL vs. BUSL vs. MSL

## Runtime

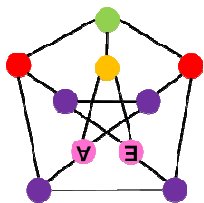


K. K  
Stat  
Mac  
ANL

# Boosted Statistical Relational Learning

Most SRL approaches seek to find models  
with a **finite** set of parameters ...  
... but we deal within **infinite** domains!

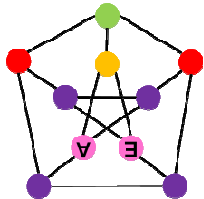
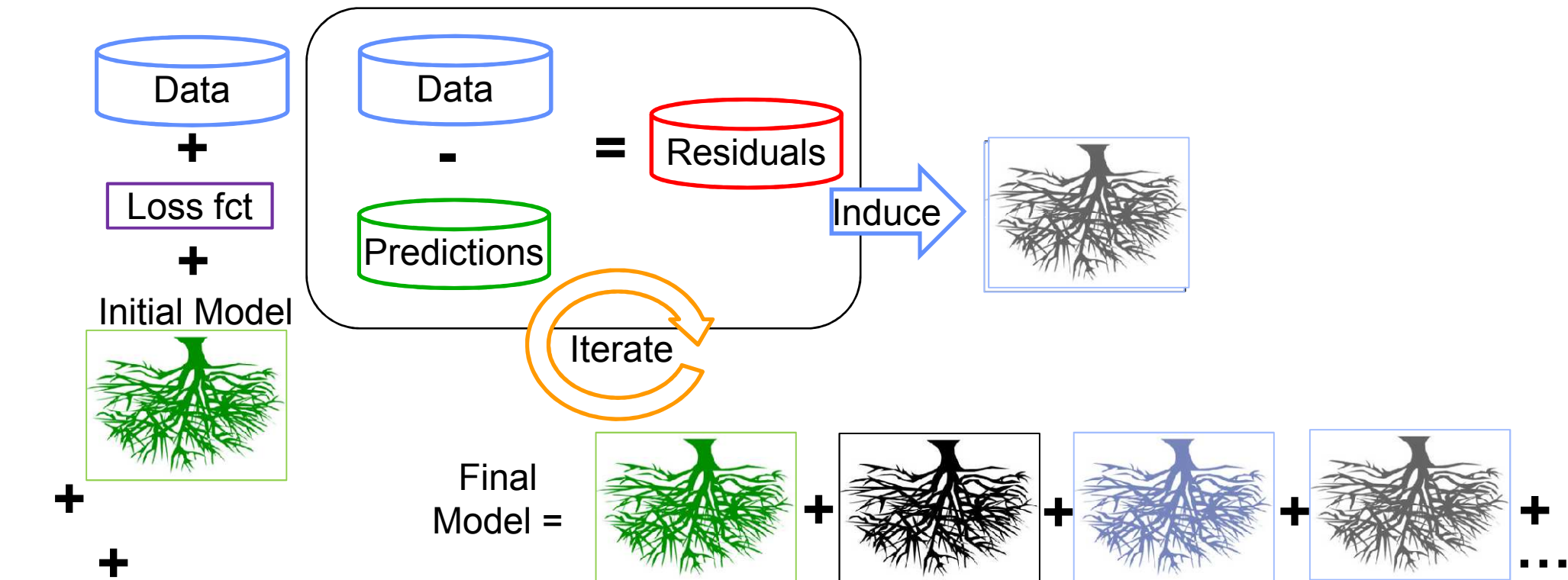
Idea: drop the finite model assumption



# Gradient (Tree) Boosting

[Friedman Annals of Statistics 29(5):1189-1232, 2001]

- Models = weighted combination of a large number of small trees (models)
- Intuition: Generate an additive model by sequentially fitting small trees to pseudo-residuals from a regression at each iteration...

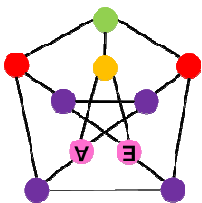


---

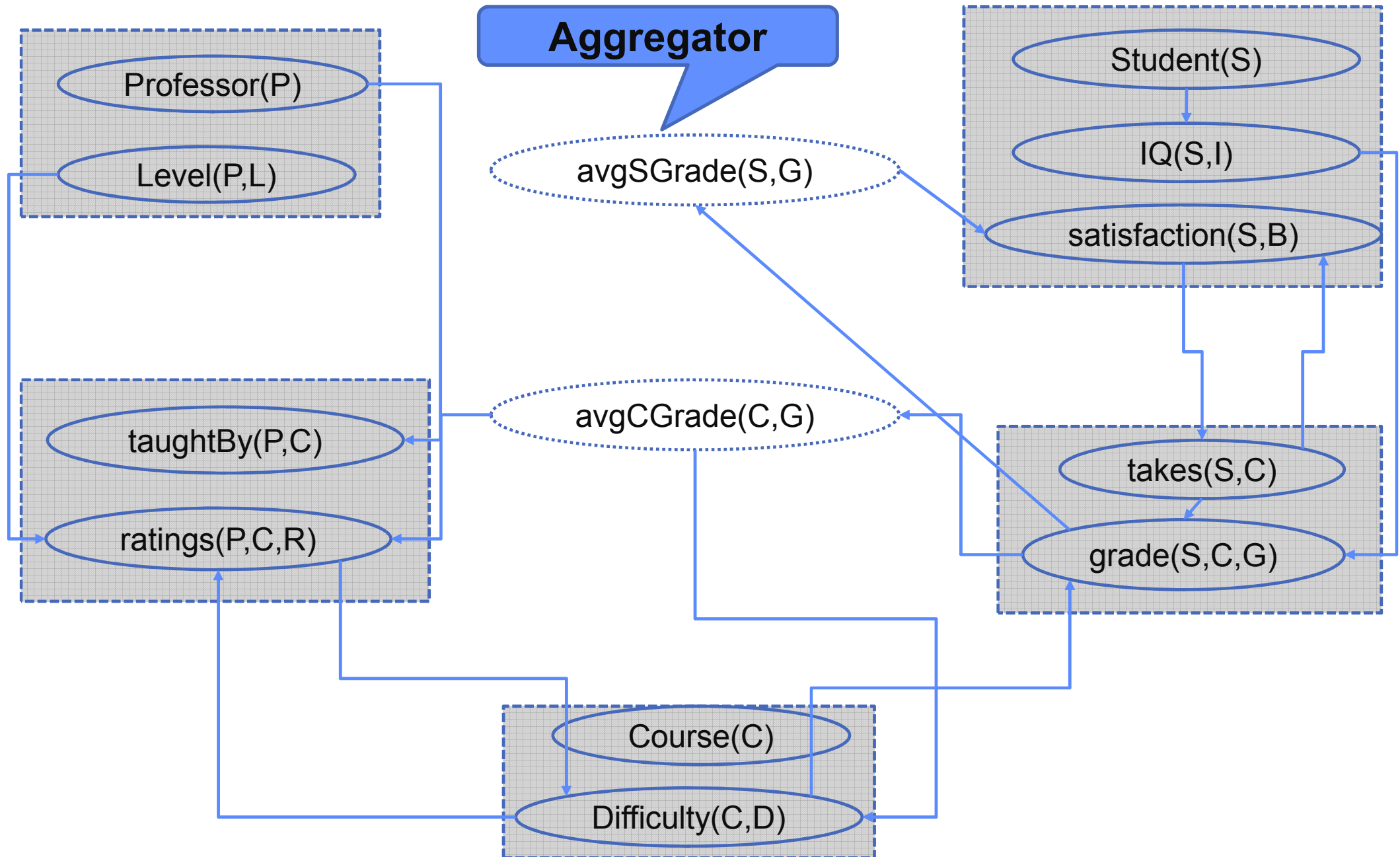
# Gradient (Tree) Boosting

Main step: estimate a relational regression model

- Relational linear-chain CRFs [Gutmann, K. ECML06]
- Policy Gradients [K., Driessens ICML08]
- Aligning relational sequences [Karwath et al. ICDM08]
- Learning Relational Dependency Networks [Natarajan et al. ILP10]



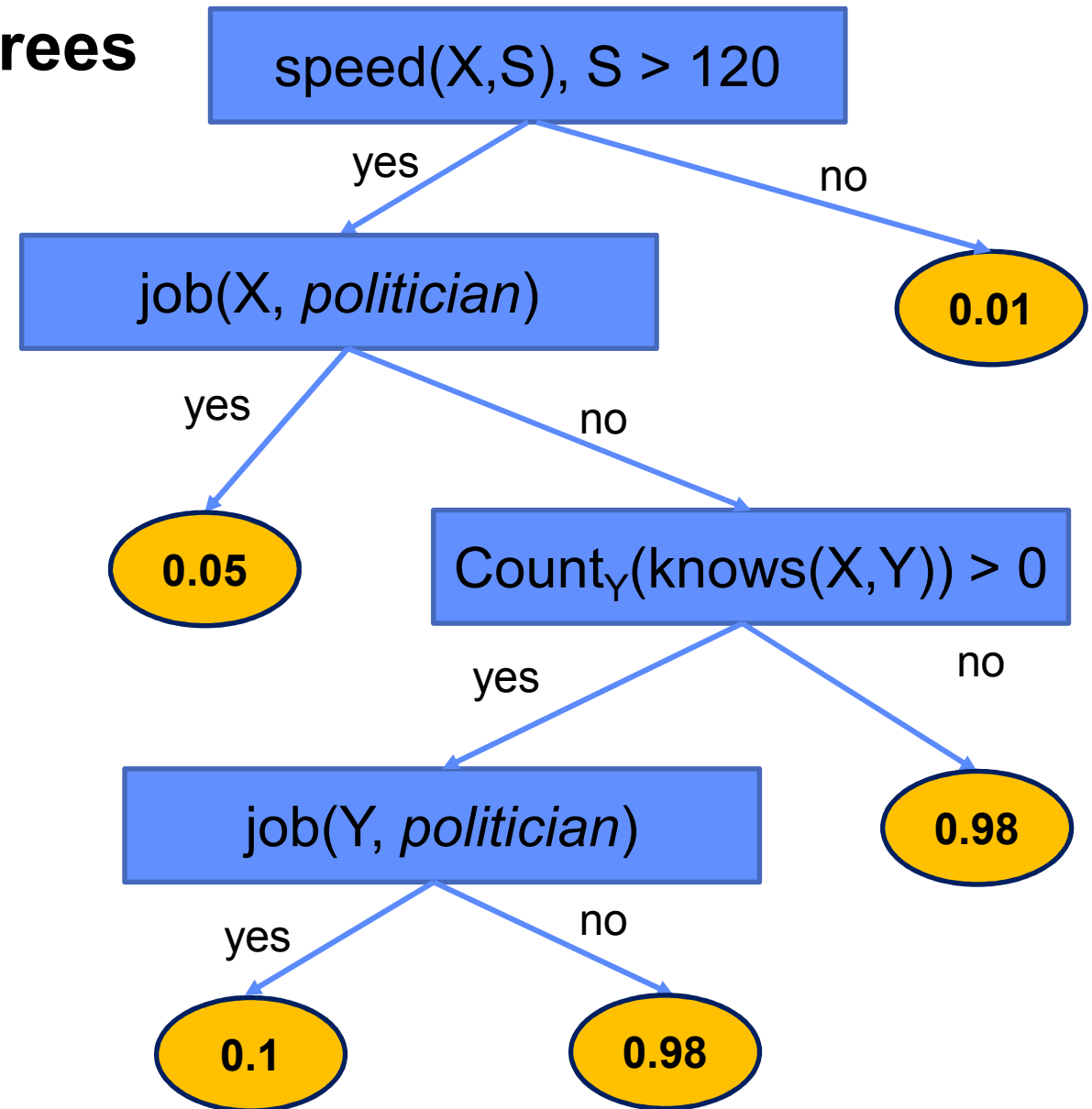
# Relational Dependency Network-Example



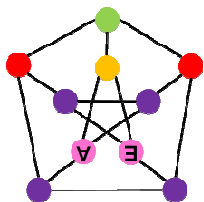
To predict  $Fine(X)$

## Relational Probability Trees

- Each conditional probability distribution can be learned as a tree
- Leaves are probabilities
- The final RDN is the set of these RPTs



Essentially [Blockeel & De Raedt '98]



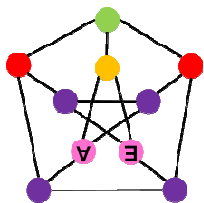
---

# Gradient Tree Boosting

- Find ML parameters, i.e. maximize  $\log P(Y|X)$  without fixing the model structure/features
- Functional Gradient

$$F_m = F_0 + \Delta_1 + \dots + \Delta_m$$

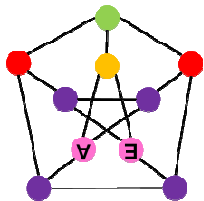
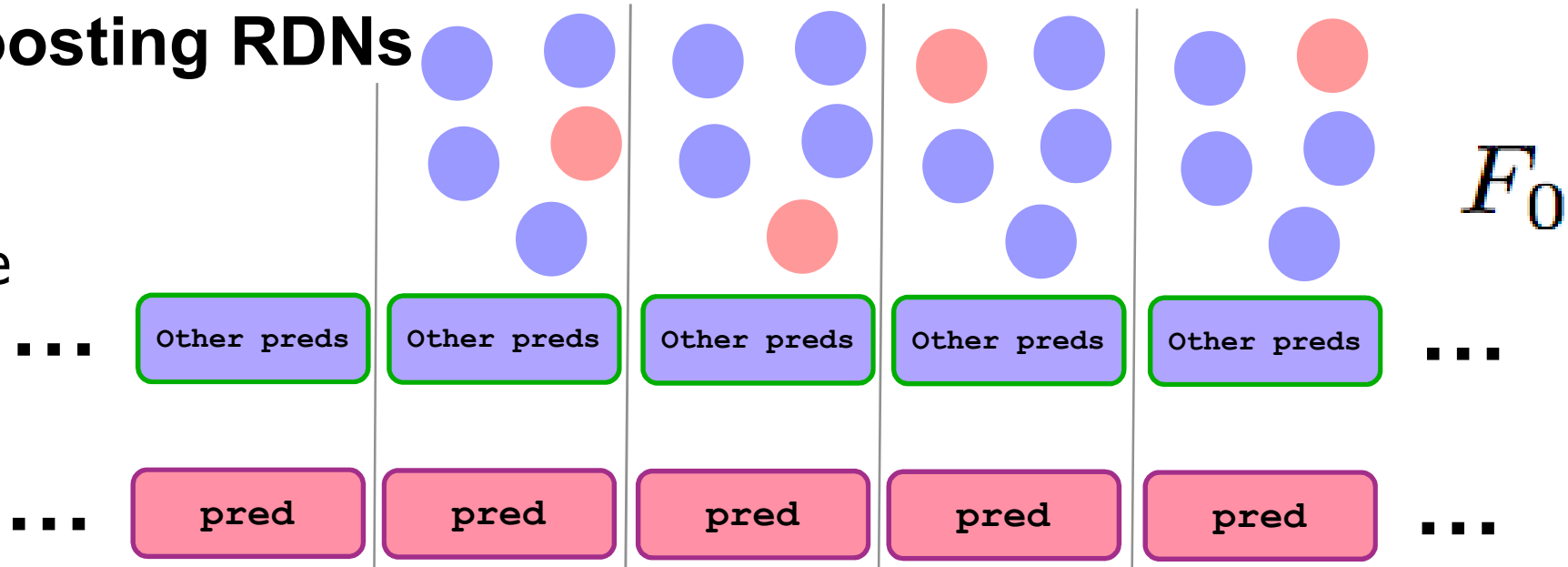
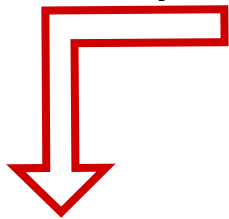
$$\Delta_m = \eta_m \cdot E_{x,y} \left[ \frac{\partial}{\partial F_{m-1}} \log P(y|x; F_{m-1}) \right]$$





# Boosting RDNs

Generate  
Example



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



Fraunhofer



**Boosting RDNs**

Generate sample ...

... Other preds ...

... pred ...

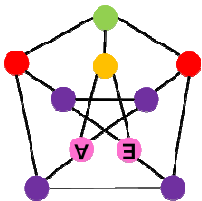
$F_0$

Legend: Blue circle (positive), Red circle (negative, weight -0.5)

$$\frac{\partial \log P(y_i | \mathbf{x}_i)}{\partial \psi(y_i=1 | \mathbf{x}_i)} = I(y_i = 1; x_i) - \frac{e^{\psi(y_i=1; x_i)}}{\sum_{y'} e^{\psi(y'; x_i)}}$$

$$\Delta_m(y_i, \mathbf{x}_i) = I(y_i = 1; \mathbf{x}_i) - P_{m-1}(y_i = 1; \mathbf{x}_i)$$

## “Weight” of each example



# Boosting RDNs

$F_0$

Generate  
Example

Other preds

Other preds

Other preds

Other preds

Other preds

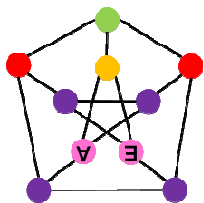
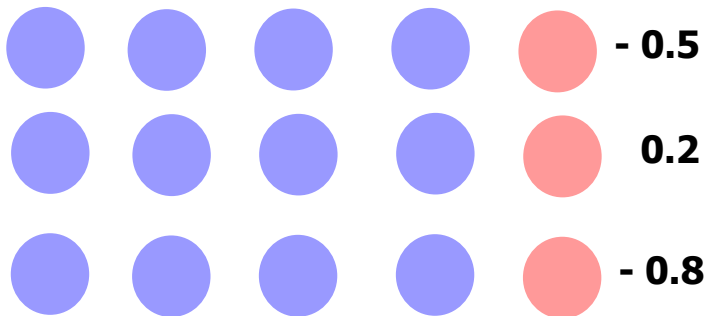
pred

pred

pred

pred

pred



# Boosting RDNs

$F_0$

Generate  
Example

Other preds

Other preds

Other preds

Other preds

Other preds

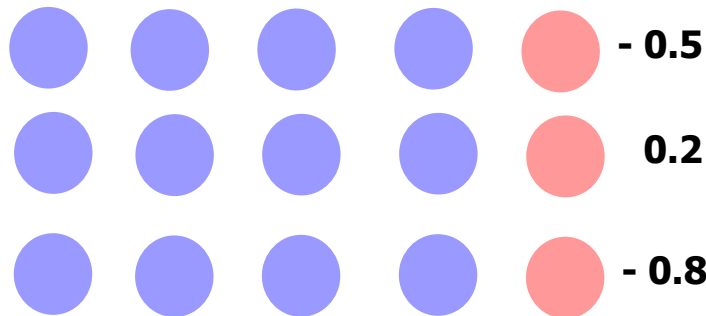
pred

pred

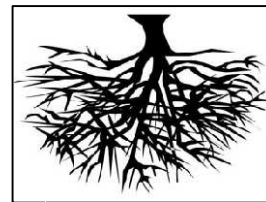
pred

pred

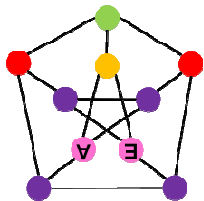
pred



Induce  
Regression  
Tree

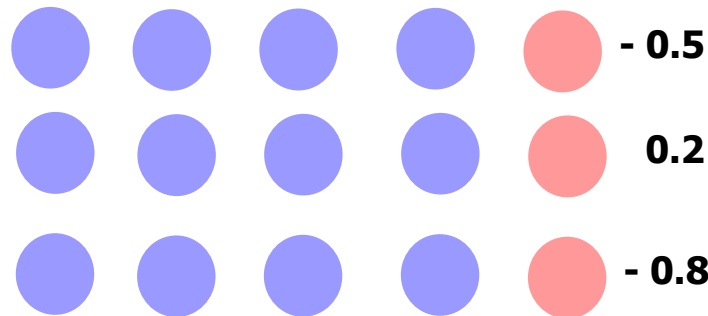
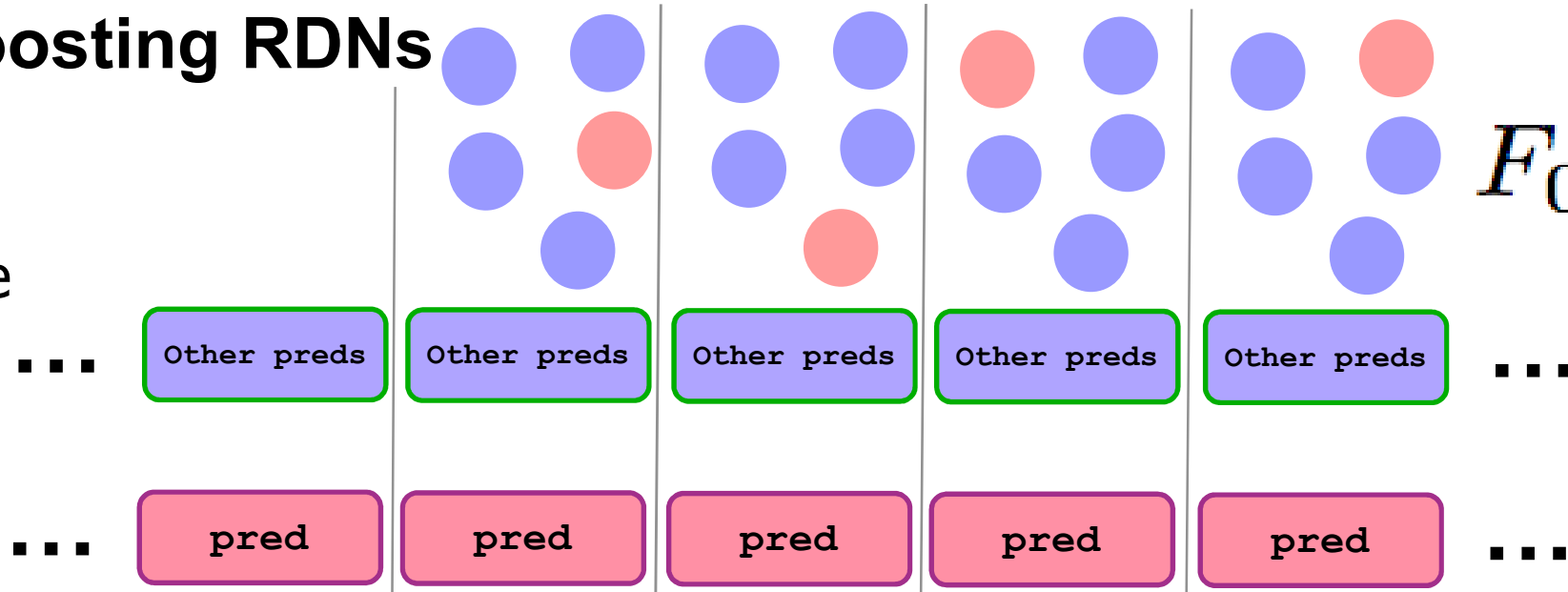
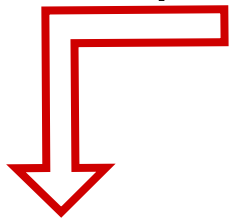


$$E_{x,y} \left[ \frac{\partial}{\partial F_{m-1}} \log P(y|x; F_{m-1}) \right]$$

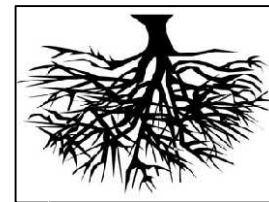
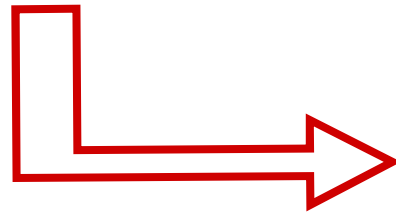


# Boosting RDNs

Generate Example

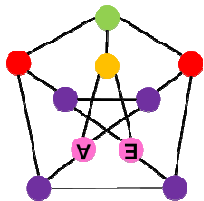


Induce Regression Tree



$$E_{x,y} \left[ \frac{\partial}{\partial F_{m-1}} \log P(y|x; F_{m-1}) \right]$$

Update Model



# Boosting RDNs

Generate Example ...

Other preds

Other preds

Other preds

Other preds

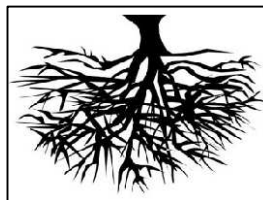
Other preds ...

$$F_0 + \Delta_1$$

Final Model =



+



+



+



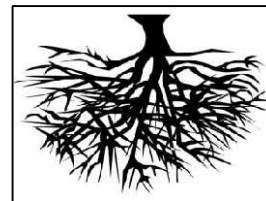
+

...

0.2

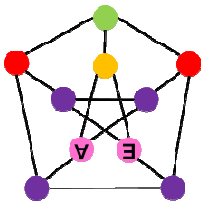
- 0.8

Induce Regression Tree



$$E_{x,y} \left[ \frac{\partial}{\partial F_{m-1}} \log P(y|x; F_{m-1}) \right]$$

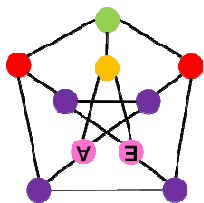
Update Model



# UW-CSE Results

- Task: *Entity Relationship* prediction
  - Predict *advisedBy* relation
  - Train in 4 areas and test in 1
  - Used RDN with Regression Tree Learner

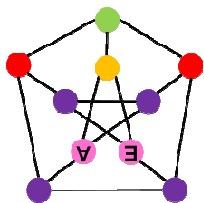
	AUC-ROC	AUC-PR	Likelihood	Training Time
Boosting	0.961	0.930	0.810	9 s
RDN	0.888	0.781	0.805	1 s
Alchemy	0.535	0.621	0.731	93 hrs



# OMOP Results

- Task: Predict *Adverse-drug events*
  - **Input:** Drugs and conditions (side-effects)
  - **Goal:** Predict if a patient is on a given drug ( $onDrug(D,P)$ )
  - Learning “in reverse”
  - Averaged over 5 train-test sets
  - Each set is a different drug

	AUC-ROC	AUC-PR	Accuracy	Training Time
Boosting	0.824	0.839	0.753	497.8 s
RDN	0.738	0.736	0.697	39.4 s
ILP + Noisy-Or	0.420	0.582	0.687	2400 s

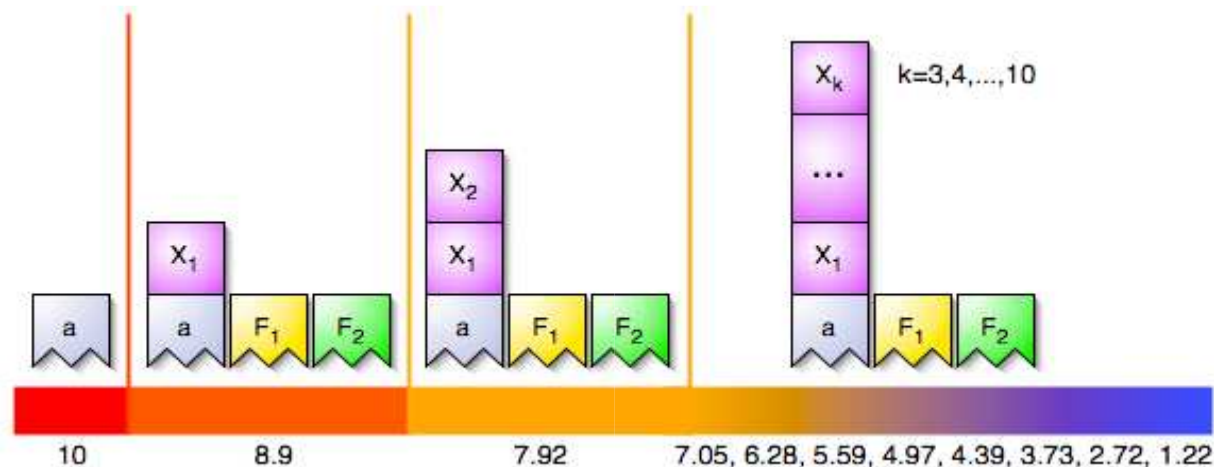




# Direct Policy Learning

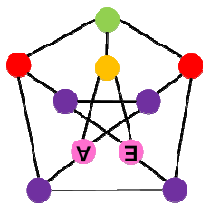
- Value functions can often be much more complex to represent than the corresponding policy

Goal:  $cl(a)$



- When policies have much simpler representations than the corresponding value functions, **direct search in policy space can be a good idea**

**Policy:** put each block on top of a on the floor



# Non-Parametric Policy Gradients [K, Driessens ICML08]

- Assume policy to be expressed using an arbitrary potential function

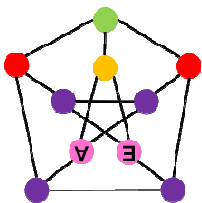
$$\pi(s, a, \Psi) = \frac{e^{\Psi(s, a)}}{\sum_b e^{\Psi(s, b)}}$$

- Do functional gradient search w.r.t. world-value

sample

$$\begin{aligned} \frac{\partial \rho}{\partial \Psi} &= \frac{\partial}{\partial \Psi} \sum_{s, a} d^{\pi}(s) \pi(s, a) Q^{\pi}(s, a) \\ &= \sum_{s, a} d^{\pi}(s) Q^{\pi}(s, a) \frac{\partial \pi(s, a)}{\partial \Psi} \end{aligned}$$

compute locally



# Local Evaluation

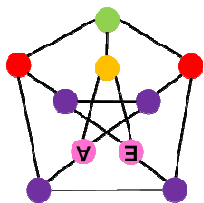
$$Q^\pi(s, a)$$

Monte-Carlo estimate or actor critic

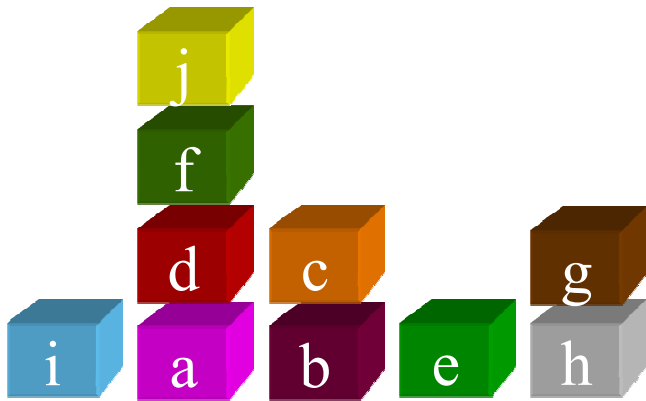
$$\pi(s, a) = \frac{e^{\Psi(s, a)}}{\sum_b e^{\Psi(s, b)}}$$

$$\frac{\partial \pi(s, a)}{\partial \Psi(s, a)} = \pi(s, a)(1 - \pi(s, a))$$

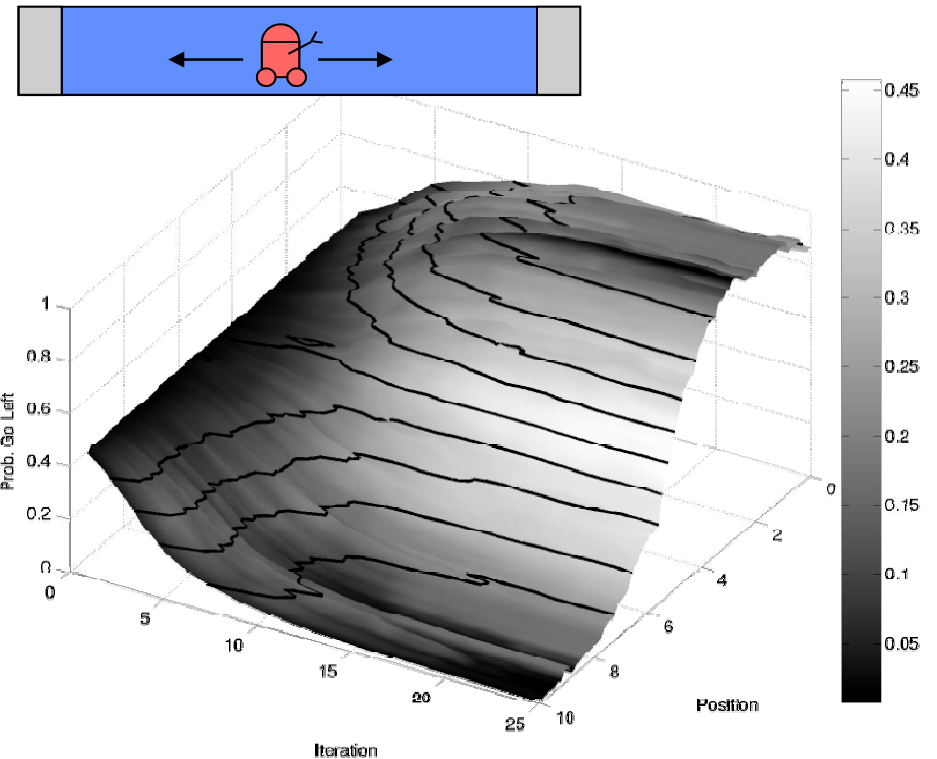
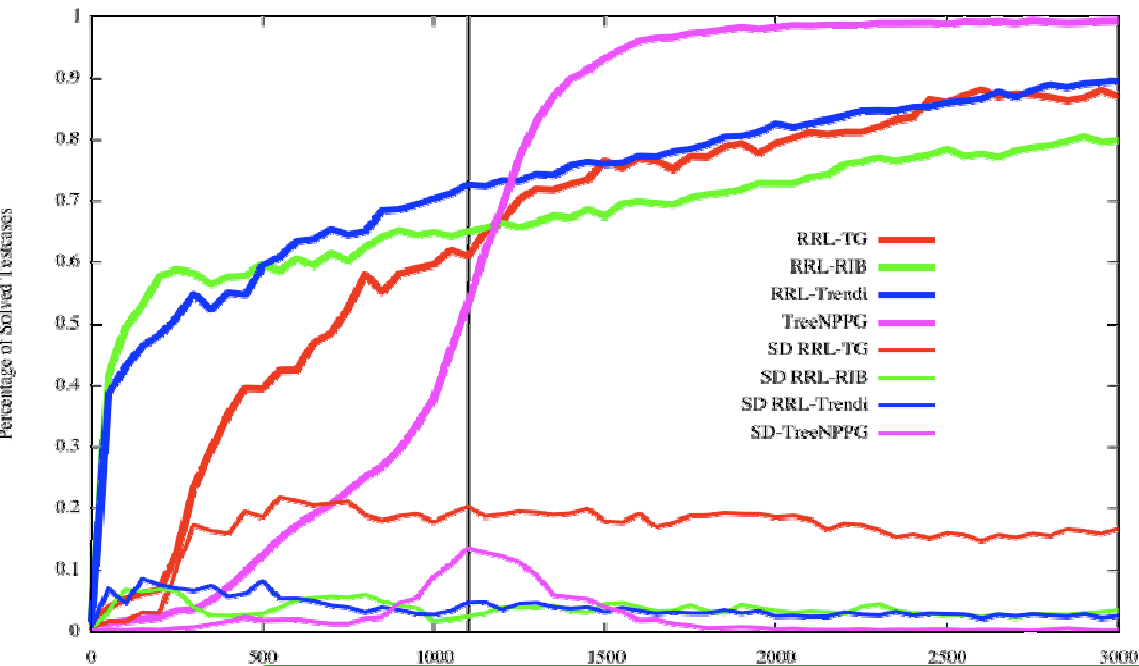
$$\frac{\partial \pi(s, a)}{\partial \Psi(s, b)} = -\pi(s, a)\pi(s, b)$$



# Some Experimental Results



On(a,b) = 10 blocks



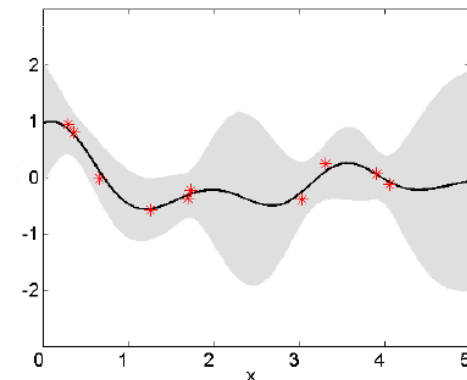
Allows us to treat propositional, continuous and relational features in a unified way!

# Networks of Continuous Values

Gaussian Processes [Rasmussen and Williams 2006] ... are effective tools to deal with continuous variables

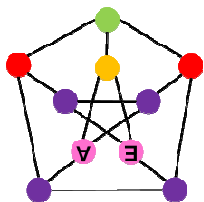
## (Not-so-Formal) Definition of Gaussian Processes

- It is a **generalization of multivariate** Gaussian distributions over finite dimensional vectors **to infinite dimensionality**.
- **Each draw** from a Gaussian process **is a function**.



Most existing SRL approaches do **NOT** deal with **continuous values**

Many AI/ML tasks can be elegantly solved using **continuous values**: terrain mapping, water quality prediction, preference learning, **topic models**...



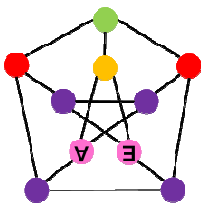
# Dirichlet-Multinomial Regression Topic Model

Condition LDA models on arbitrary document metadata

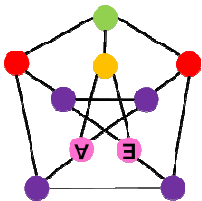
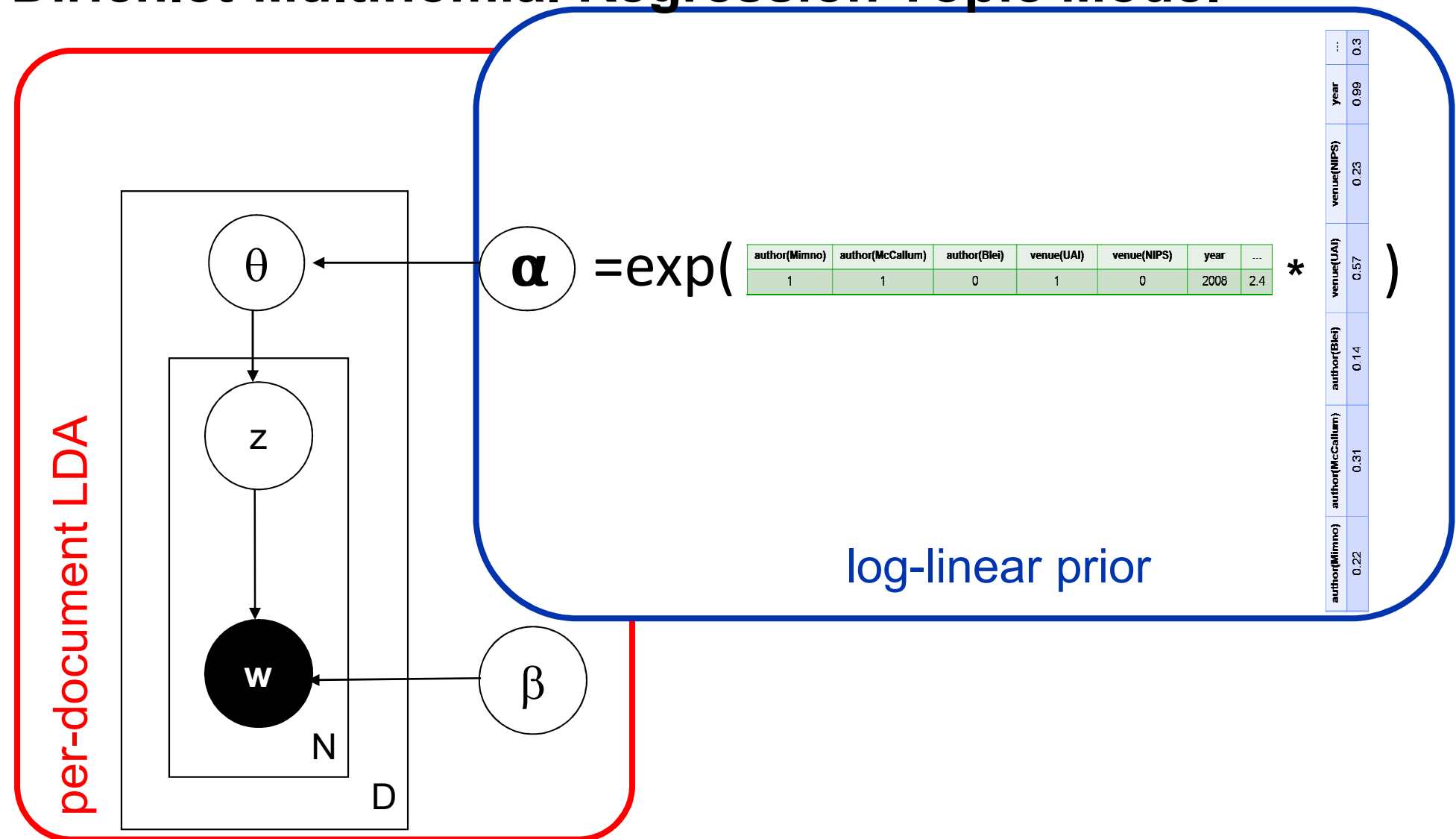
- Log-linear prior on document-topic distributions that is a function of observed features of the document,
  - author, publication venue, references, dates, ....

author(Mimno)	author(McCallum)	author(Blei)	venue(UAI)	venue(NIPS)	year	...
1	1	0	1	0	2008	2.4

author(Mimno)	author(McCallum)	author(Blei)	venue(UAI)	venue(NIPS)	year	...
0.22	0.31	0.14	0.57	0.23	0.99	0.3



# Dirichlet-Multinomial Regression Topic Model



# Dirichlet-Multinomial Regression Topic Model

1. For each topic  $t$ ,

(a) Draw  $\lambda_t \sim \mathcal{N}(0, \sigma^2 I)$

(b) Draw  $\phi_t \sim \mathcal{D}(\beta)$

▪ Lack of dependencies manifested in diagonal covariance matrix

2. For each document  $d$ ,

(a) For each topic  $t$  let  $\alpha_{dt} = \exp(\mathbf{x}_d^T \lambda_t)$ .

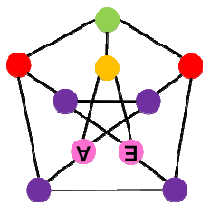
(b) Draw  $\theta_d \sim \mathcal{D}(\alpha_d)$ .

(c) For each word  $i$ ,

i. Draw  $z_i \sim \mathcal{M}(\theta_d)$ .

ii. Draw  $w_i \sim \mathcal{M}(\phi_{z_i})$ .

Idea: Plug-in SRL approach





# Gaussian Processes [Rasmussen and Williams 2006] . . .

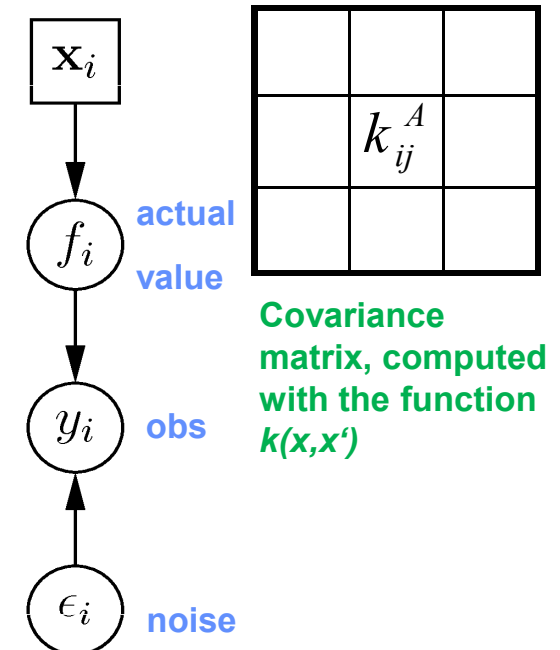
- ... are effective tools to deal with continuous variables
- **Prior distribution** of actual value:

$$f | \theta_a \sim GP(0, k_a(x, x'))$$

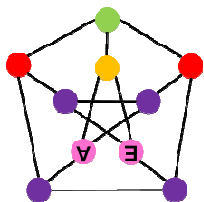
- $k(x, x')$  : **covariance functions**. "The more

**Relational knowledge** can further reveal **additional pairwise correlations** between variables of interest

Correlation between entities is often modelled via a kernel function using input attributes of the entities only



$$*, X)(K_a + \sigma^2 I)^{-1} k_a(x_*, X)^T$$





# Relational Gaussian Process Models

- Relational GPs

(Chu, Ghahramani NIPS06)

- Single relation modeled as random variable
- Known relations d-separate entities

- XGPs

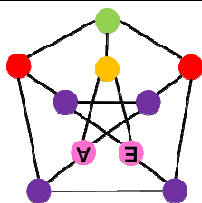
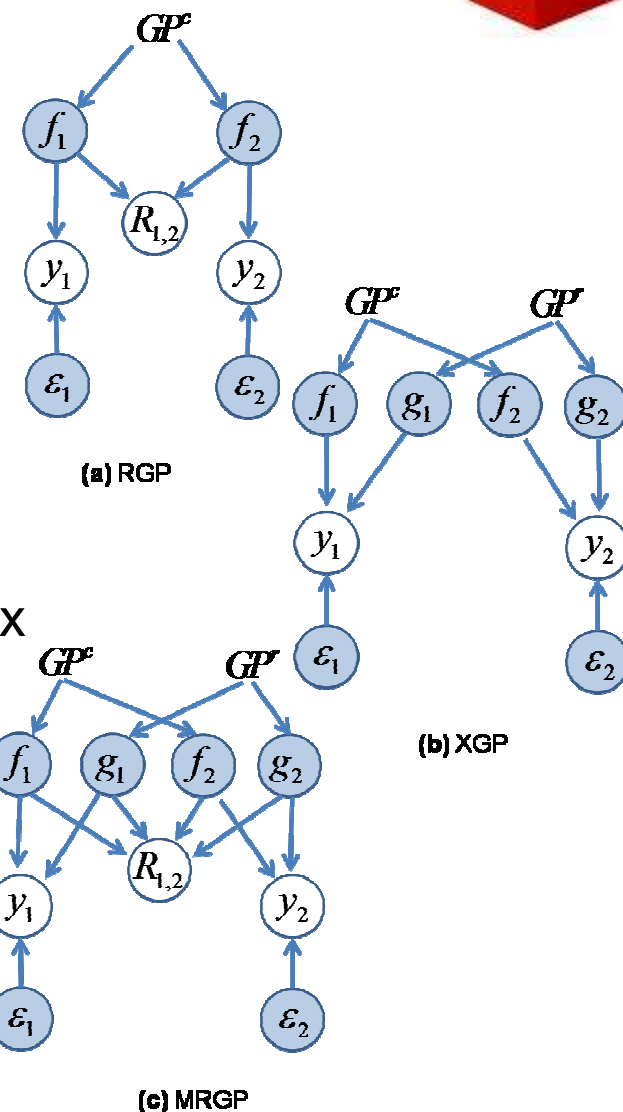
(Silva, Chu, Ghahramani NIPS07)

- Single relation modeled via covariance matrix
- No “link” prediction possible

- Multi-Relational GPs

(Xu, K, Tresp IJCAI09)

- Multiple relations (multiple colors) modeled
- Combining RGPs and XGPs

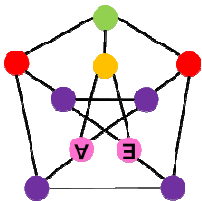
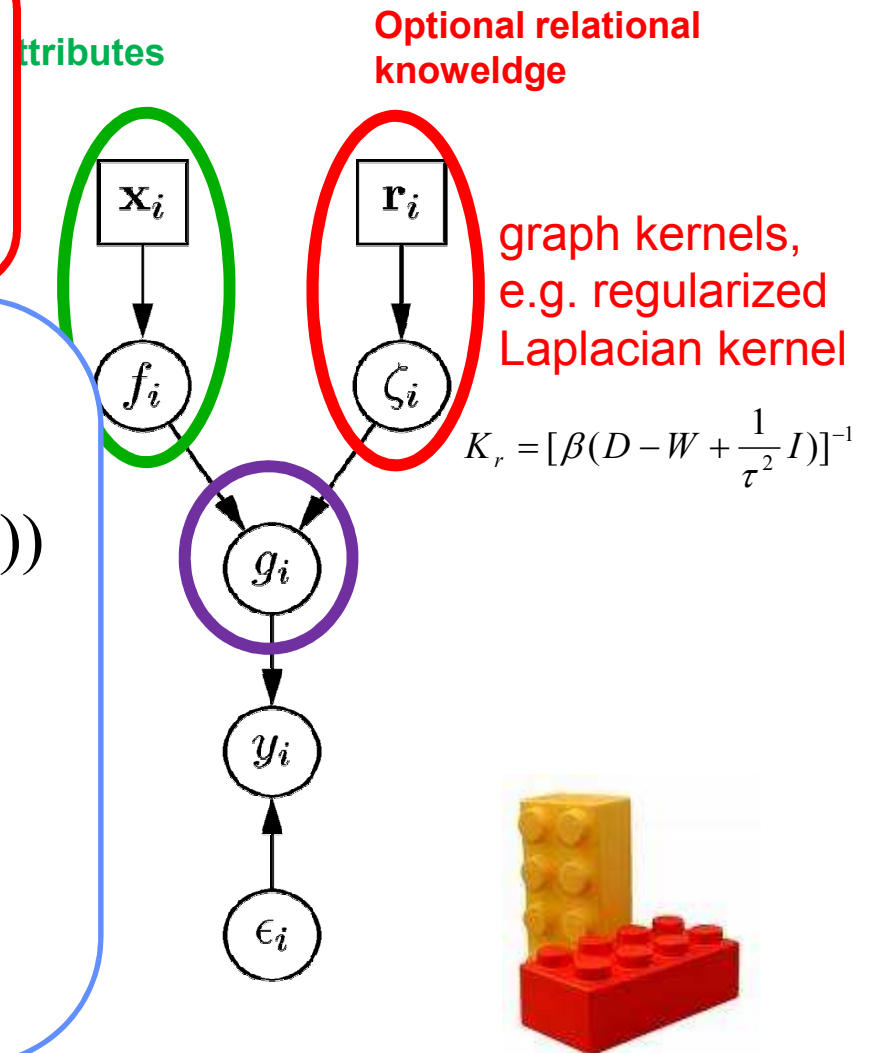
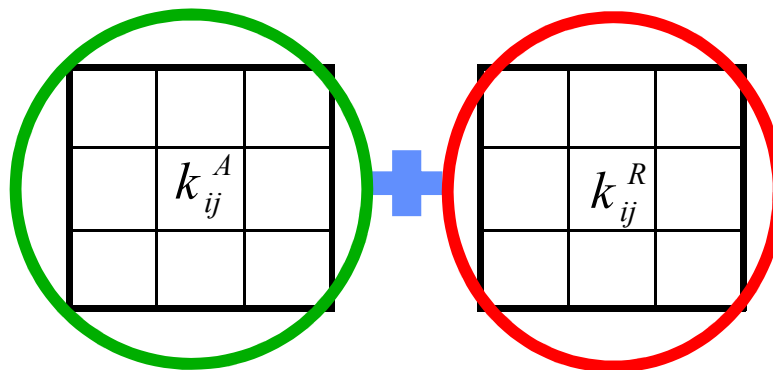


# GPs with Hidden Common Cause Relations [Silva et al. NIPS07]

Essentially, a GP with graph kernels

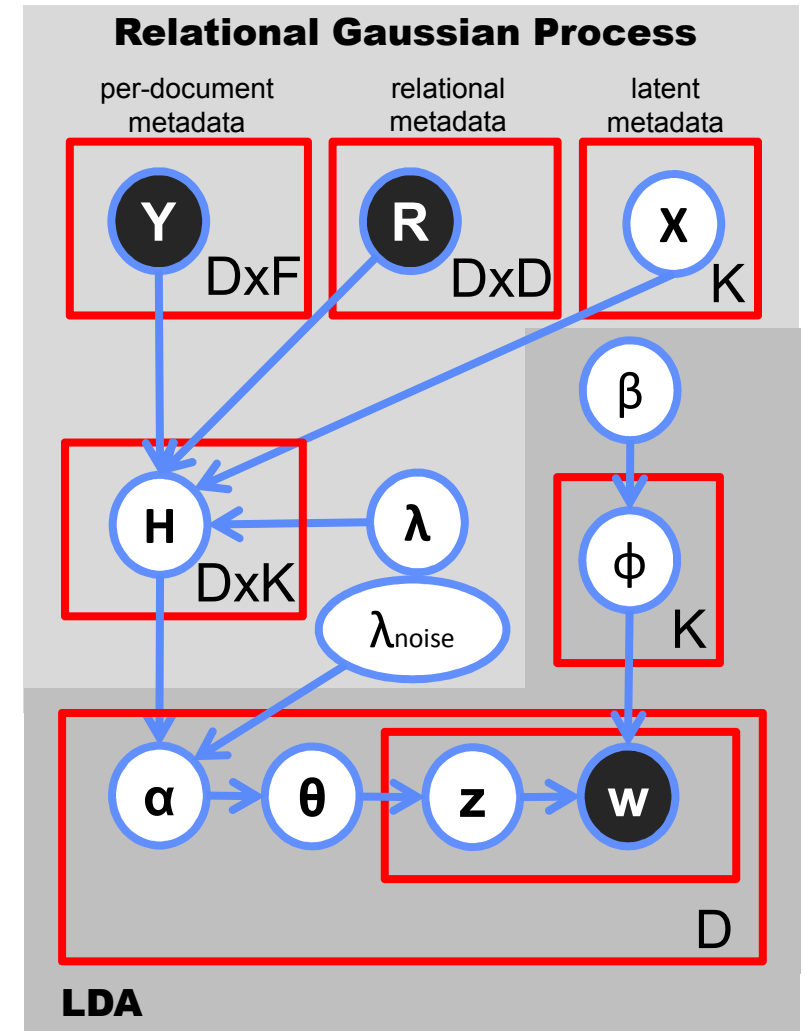
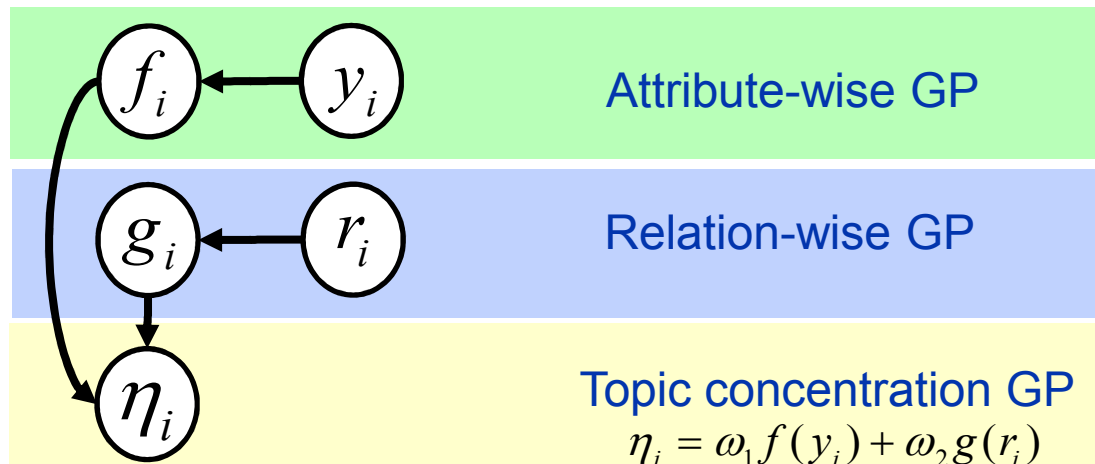
We get the prior distribution as

$$g \mid \theta_a, \theta_r \sim GP(0, \omega_1^2 k_a(x, x') + \omega_2^2 k_r(r, r'))$$



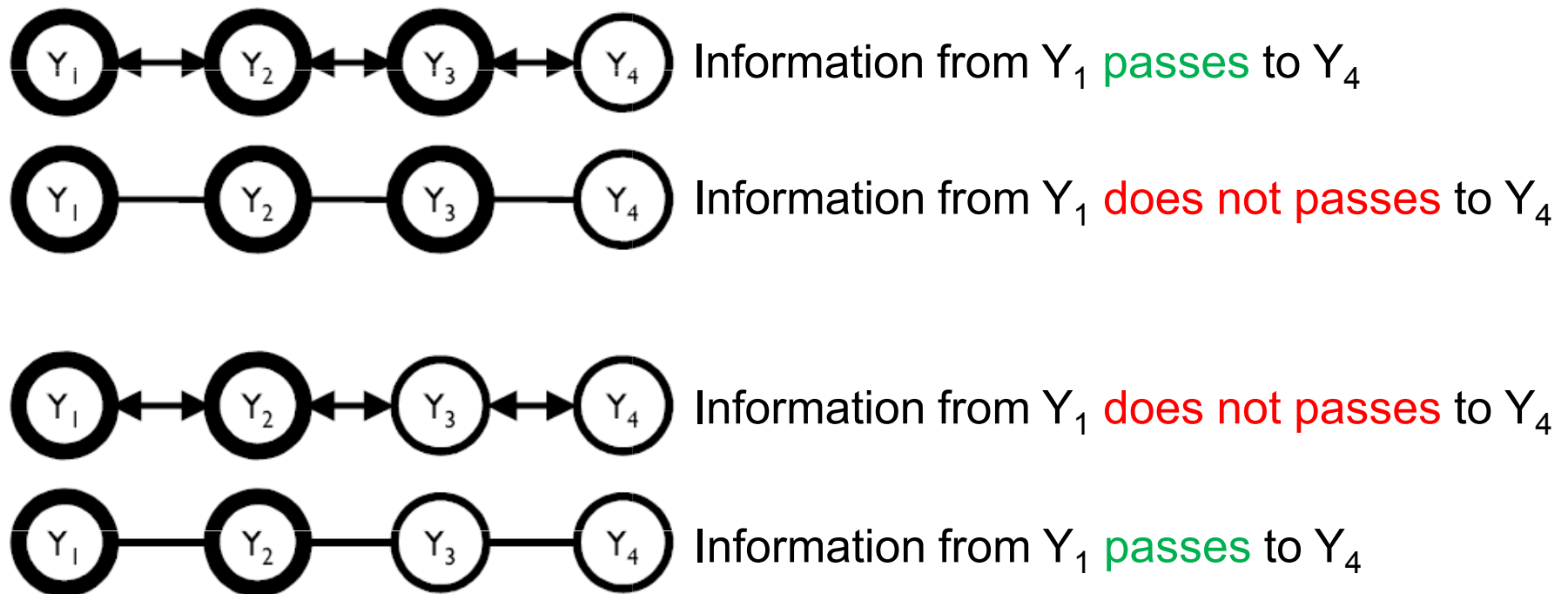
## xLDA – Topic Model

- Relations are modeled as hidden common causes of topic concentrations
- Graph-kernels provide an efficient tool to capture the information in relations
- The model is **easy to implement**
  - weighted sum of covariance matrixes
- Can be extended to also estimate latent topic metadata  $X$



# Comparison to Markov (Logic) Network / CRF

- XGPs employ **mixed-graph models**



Markov network would ignore all training documents in a chain besides the endpoints due to the Markov assumption



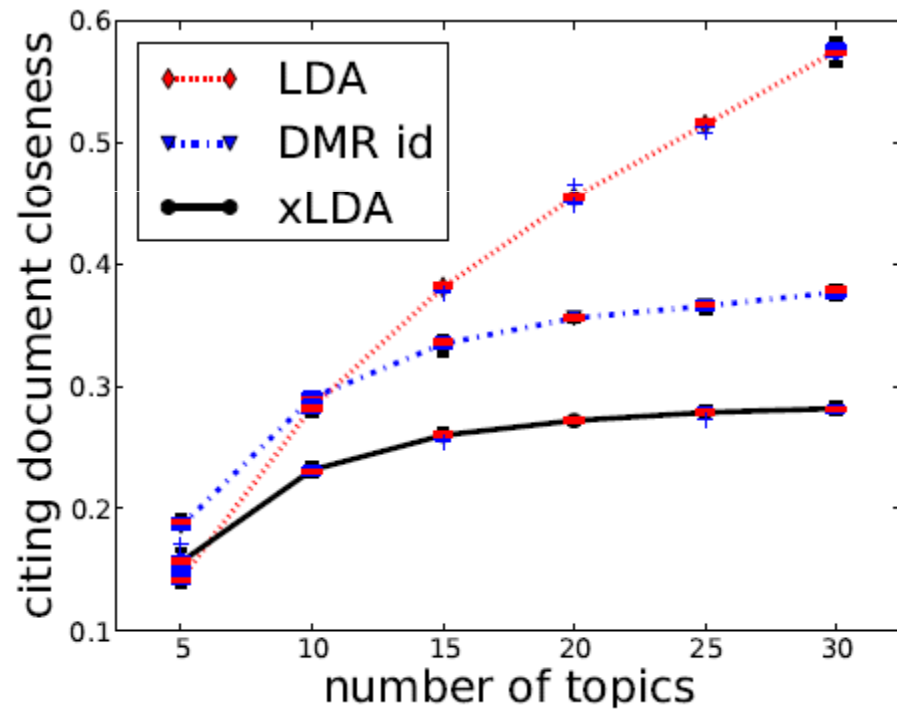
: observed node



: unobserved node

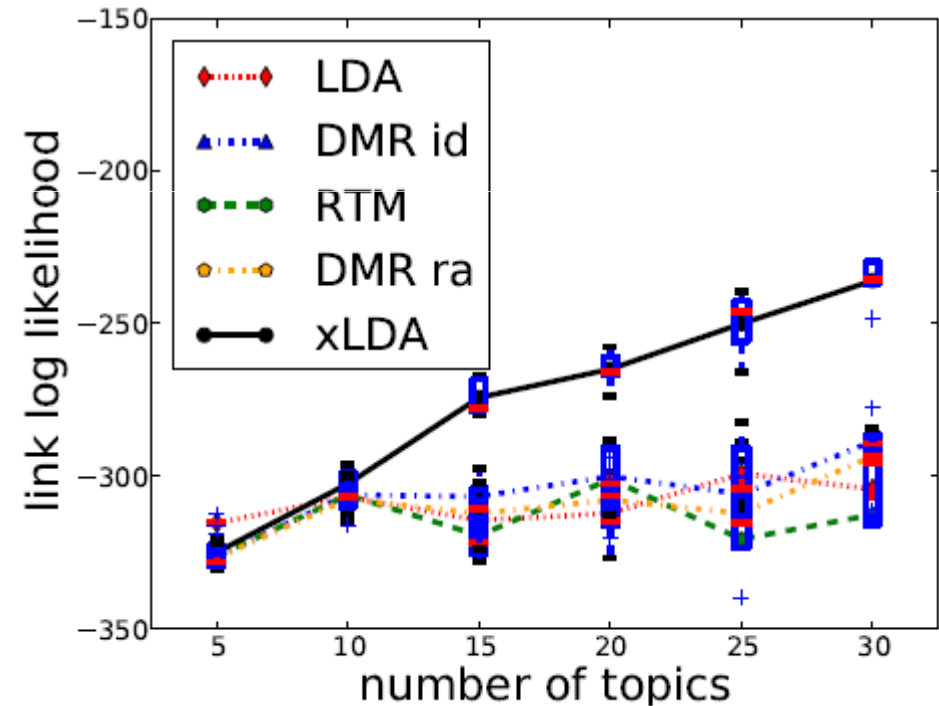
**Fraunhofer**

# Cora

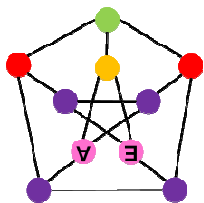


(c) Hellinger distance (the lower, the better) of linked documents for different number of topics.

# Wikipedia



(d) Average link log likelihood (the higher, the better) for different number of topics.

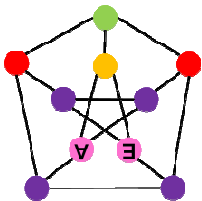


---

## So far

- Motivation
- Several SR frameworks
- Semantics
- Inference
- Learning

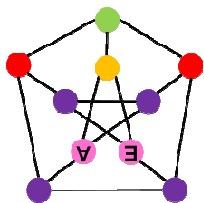
So, essentially we are done !



## But wait ...

- Inference in first-order logic is not „ground“
- ... it is lifted, i.e., it never „touches“ the ground
- **Can we do similar things?**

- **Yes, we can.** Resulting lifted approaches are often
  - Faster
  - More compact
  - More intuitive
  - Higher level – more information/structure available for optimization





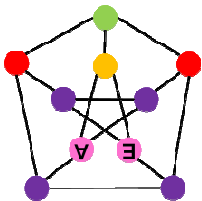
## Why does it work? Typically tons of redundancies

$$\forall x, y, z \quad \text{GradStudent}(x) \wedge \text{Prof}(y) \wedge \text{Prof}(z) \wedge \text{TA}(x, z) \wedge \text{SameGroup}(y, z) \\ \rightarrow \text{AdvisedBy}(x, y)$$

10,000	People at some school	The Evidence
2000	Graduate students	
1000	Professors	
1000	TAs	
500	Pairs of professors in the same group	

Total Num of Groundings =  $|x| \times |y| \times |z| = 10^{12}$

$10^{12}$



$\text{GradStudent}(x) \wedge \text{Prof}(y) \wedge \text{Prof}(z) \wedge \text{TA}(x,z) \wedge \text{SameGroup}(y,z) \rightarrow \text{AdvisedBy}(x,y)$

FROG keeps only these  $X$  values

**GradStudent( $x$ )**

GradStudent(P1)

$\neg$  GradStudent(P2)

GradStudent(P3)

True

GradStudent(P1)

GradStudent(P3)

...

False

$\neg$  GradStudent(P2)

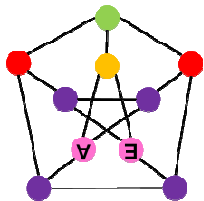
$\neg$  GradStudent(P4)

...

All these values for  $X$  satisfy the clause, regardless of  $Y$  and  $Z$

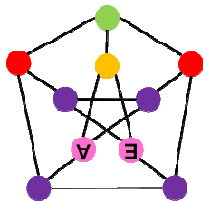
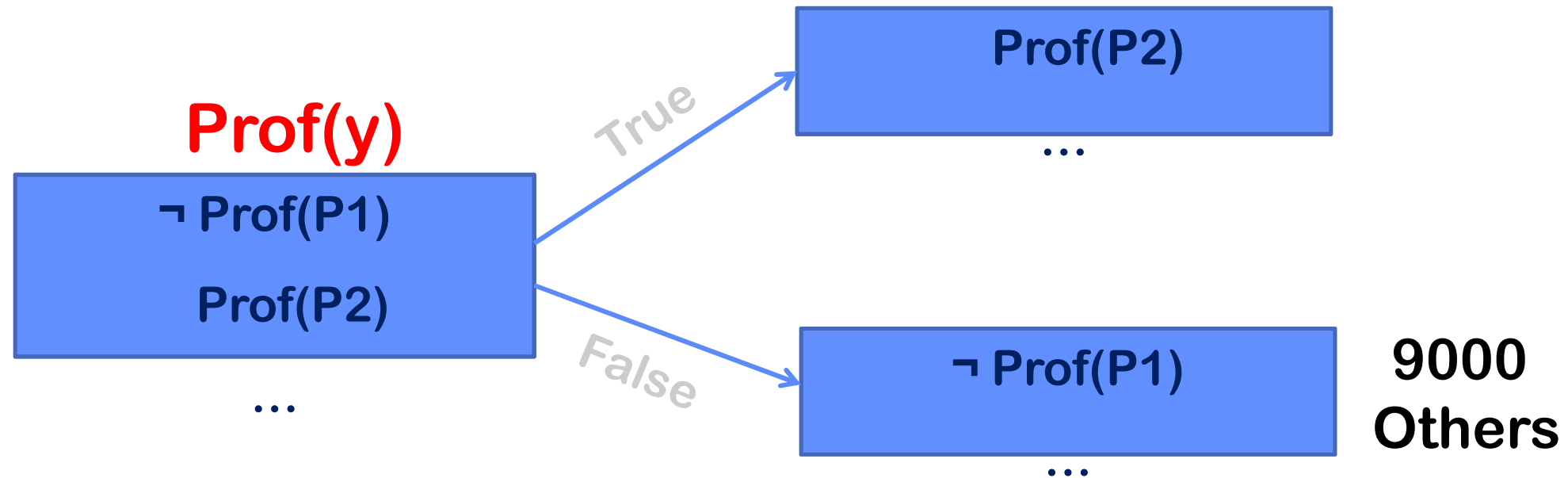
Instead of  $10^4$  values for  $X$ ,  
have  $2 \times 10^3$

$2 \times 10^{11}$



$\text{GradStudent}(x) \wedge \text{Prof}(y) \wedge \text{Prof}(z) \wedge \text{TA}(x,z) \wedge \text{SameGroup}(y,z) \rightarrow \text{AdvisedBy}(x,y)$

---



K. Kersting  
 Statistical Relational Learning  
 Machine Learning Summer School (MLSS)  
 ANU, Canberra, Australia, Oct. 4, 2010



Fraunhofer

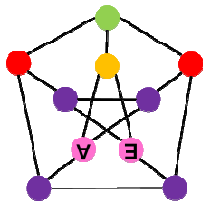
IAIS Universität Bonn  
 RHEINISCHE FRIEDRICH-WILHELMS-  
 UNIVERSITÄT

$2 \times 10^{10}$

$\text{GradStudent}(x) \wedge \text{Prof}(y) \wedge \text{Prof}(z) \wedge \text{TA}(x,z) \wedge \text{SameGroup}(y,z) \rightarrow \text{AdvisedBy}(x,y)$

---

<<< Same as Prof(y) >>>



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



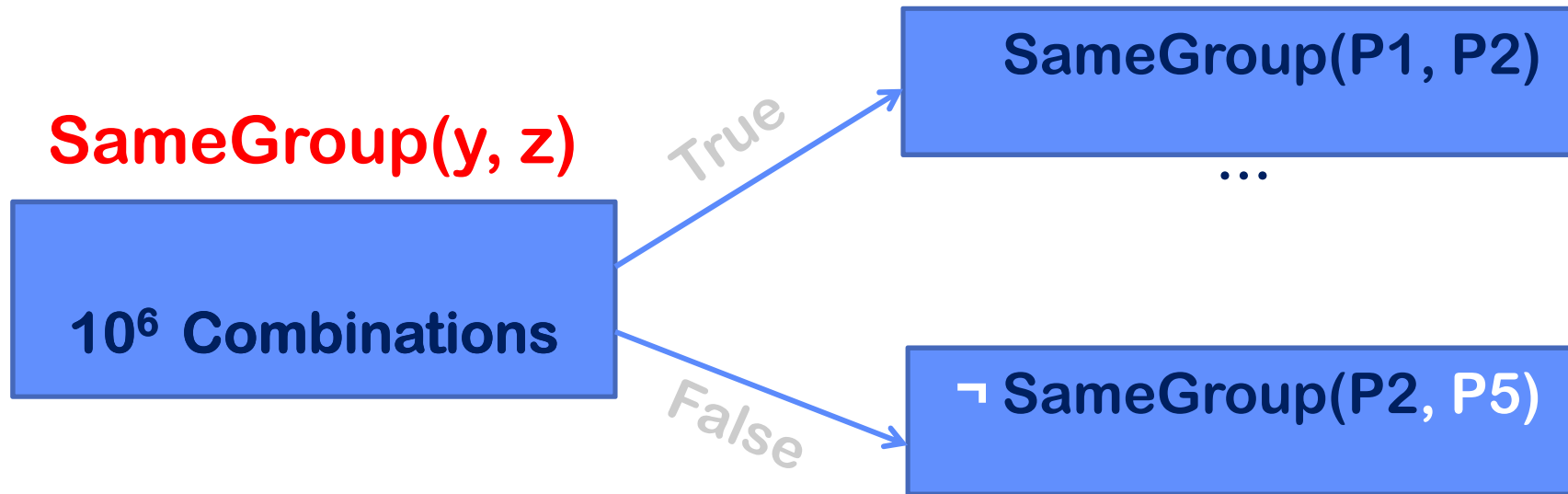
Fraunhofer

IAIS Universität Bonn  
RHEINISCHE FRIEDRICH-WILHELMS-  
UNIVERSITÄT

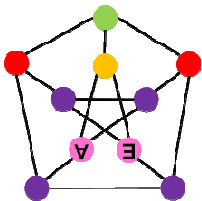
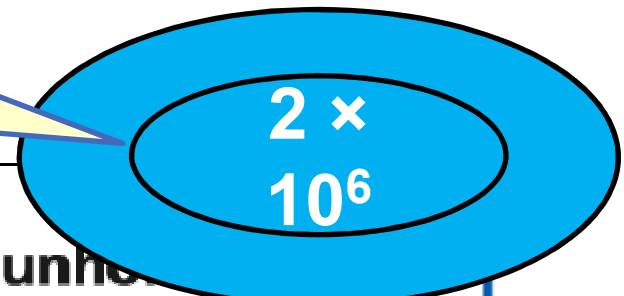
$2 \times 10^9$

▪  $\text{GradStudent}(x) \wedge \text{Prof}(y) \wedge \text{Prof}(z) \wedge \text{TA}(x,z) \wedge \text{SameGroup}(y,z) \rightarrow \text{AdvisedBy}(x,y)$

---



**only**  
**1000 Y:Z combinations**

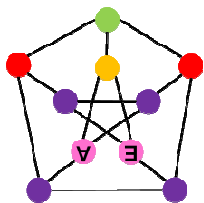
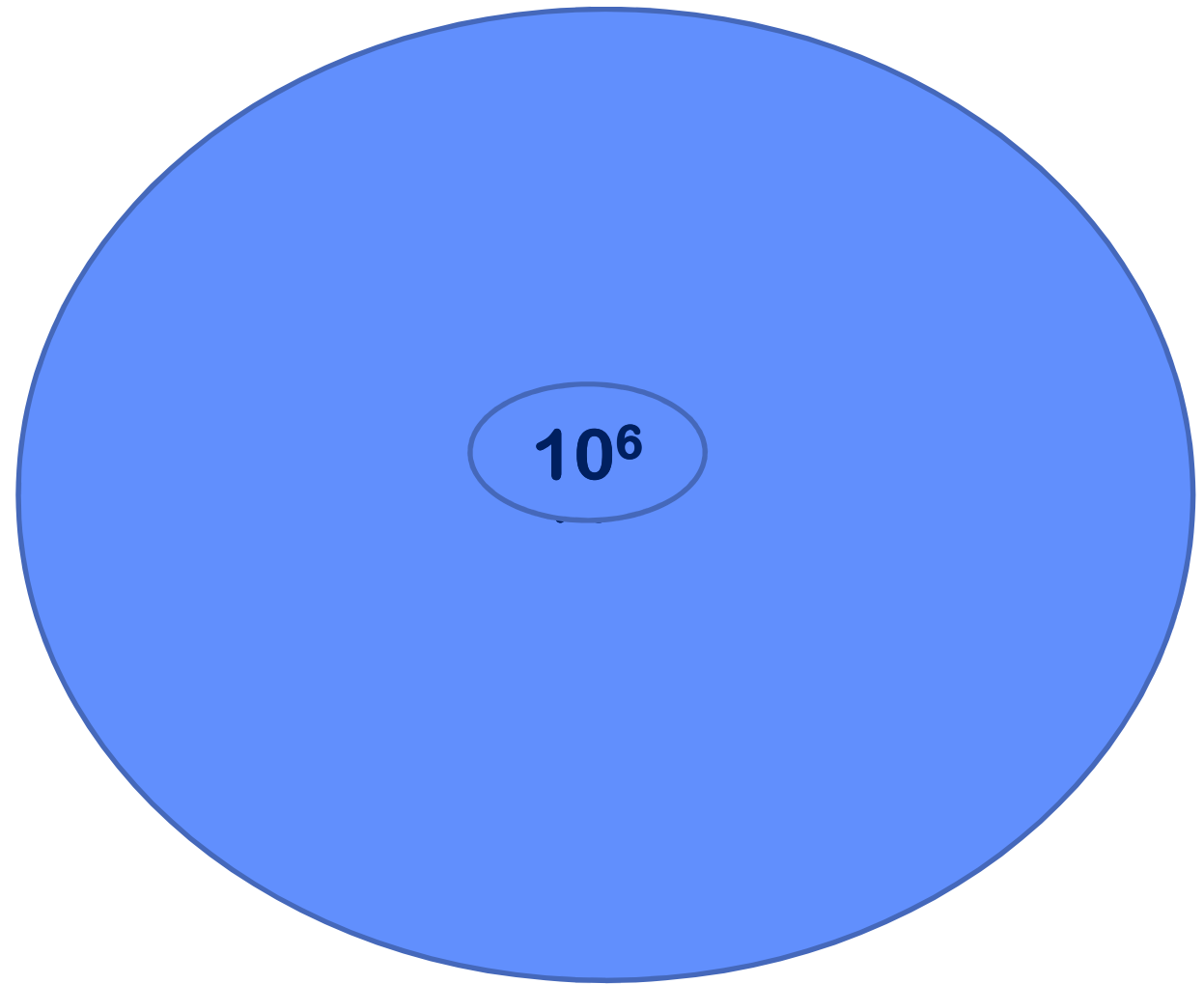


$$\text{GradStudent}(x) \wedge \text{Prof}(y) \wedge \text{Prof}(z) \wedge \text{TA}(x,z) \wedge \text{SameGroup}(y,z) \rightarrow \text{AdvisedBy}(x,y)$$

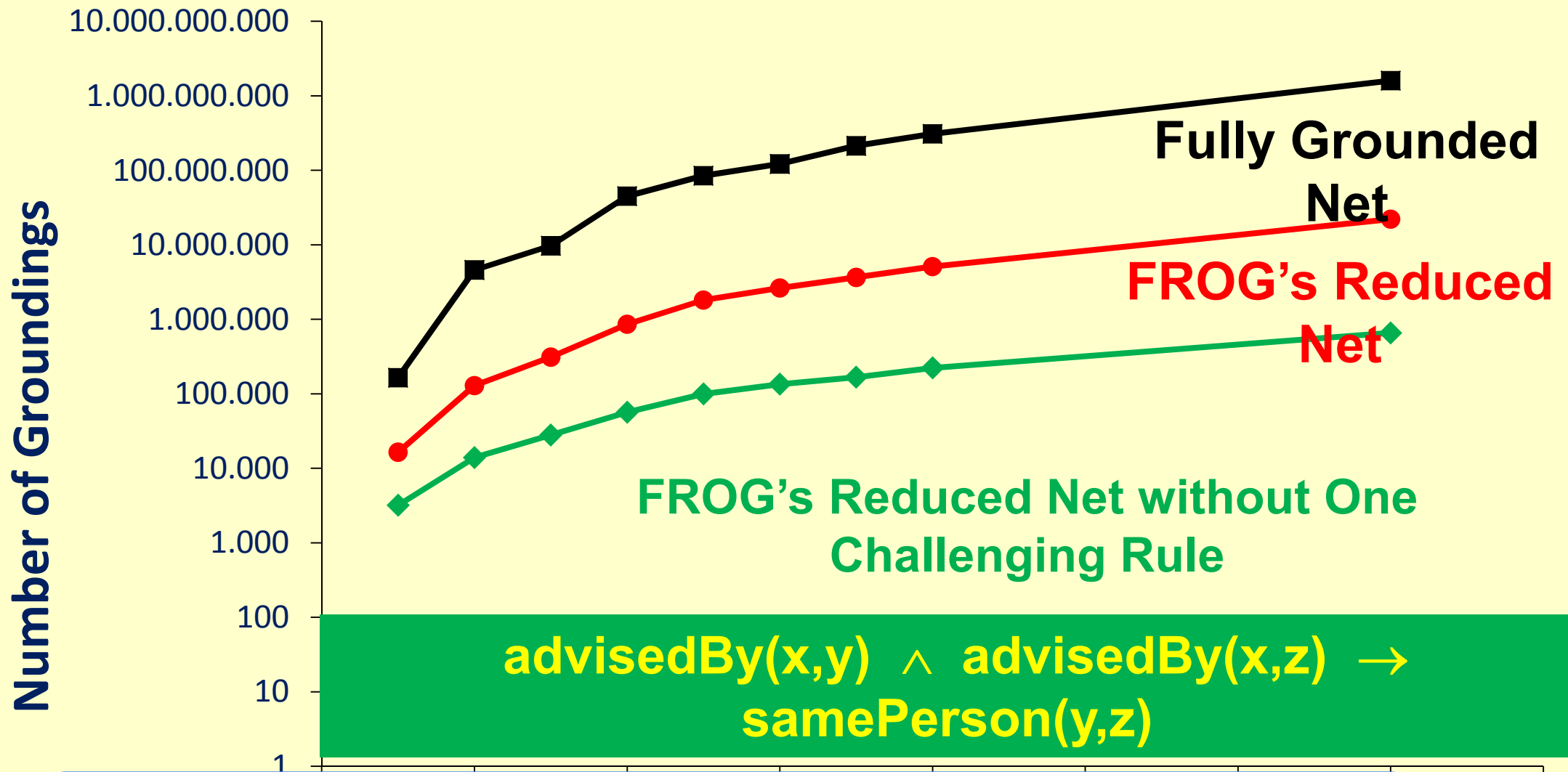
---

**Original number of  
groundings =  $10^{12}$**

**Final number of  
groundings  $\leq 10^6$**

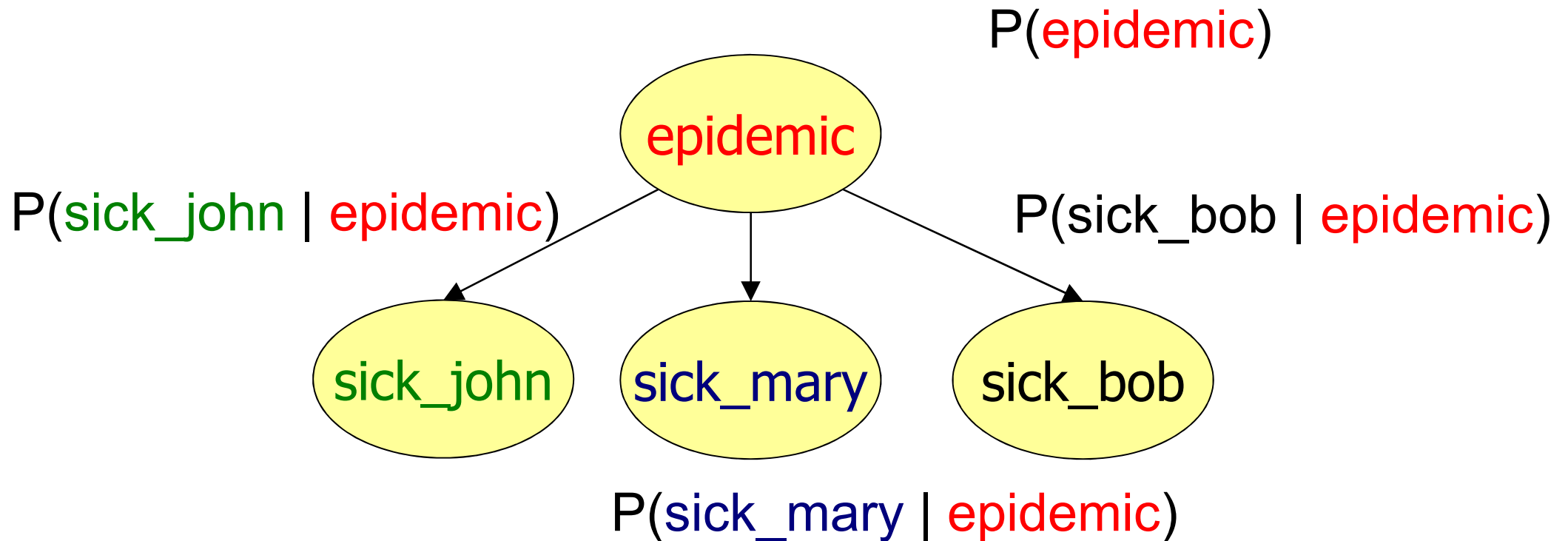


## Sample Results: UWash-CSE

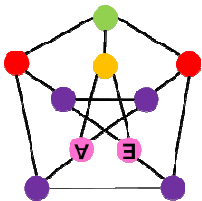


OK, this is preprocessing. Can we exploit redundancies also at inference time?

# Bayesian Networks (directed)



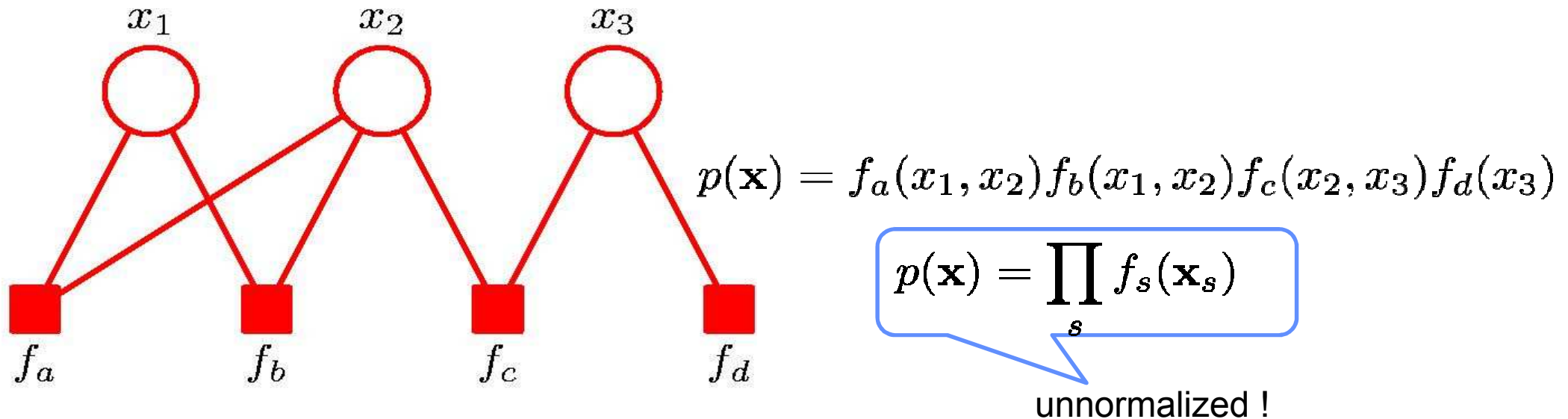
$$\begin{aligned} & P(\text{sick\_john}, \text{sick\_mary}, \text{sick\_bob}, \text{epidemic}) \\ = & P(\text{sick\_john} \mid \text{epidemic}) * P(\text{sick\_mary} \mid \text{epidemic}) \\ & * P(\text{sick\_bob} \mid \text{epidemic}) * P(\text{epidemic}) \end{aligned}$$



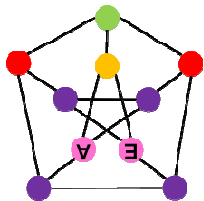


# Distributions can naturally be represented as **factor graphs**

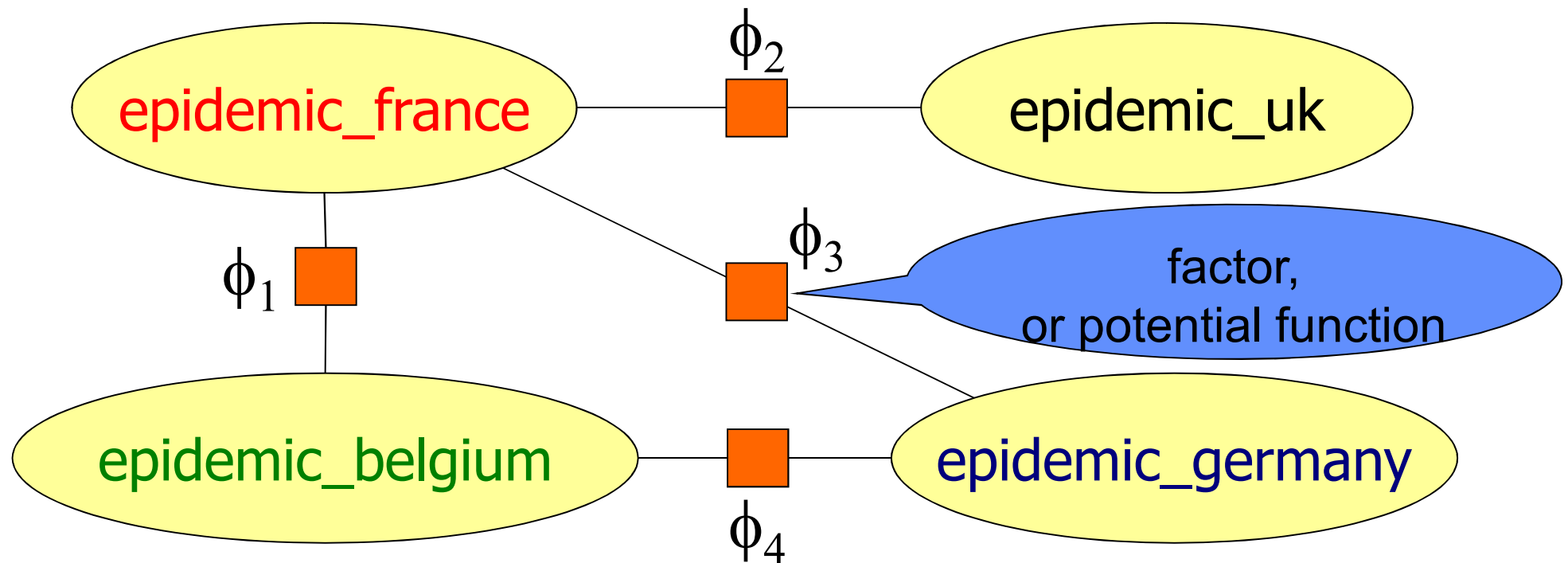
- Each circle denotes a (random) variable, each box denotes a factor (potential)



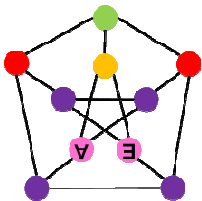
- There is an edge between a circle and a box if the variable is in the domain/scope of the factor



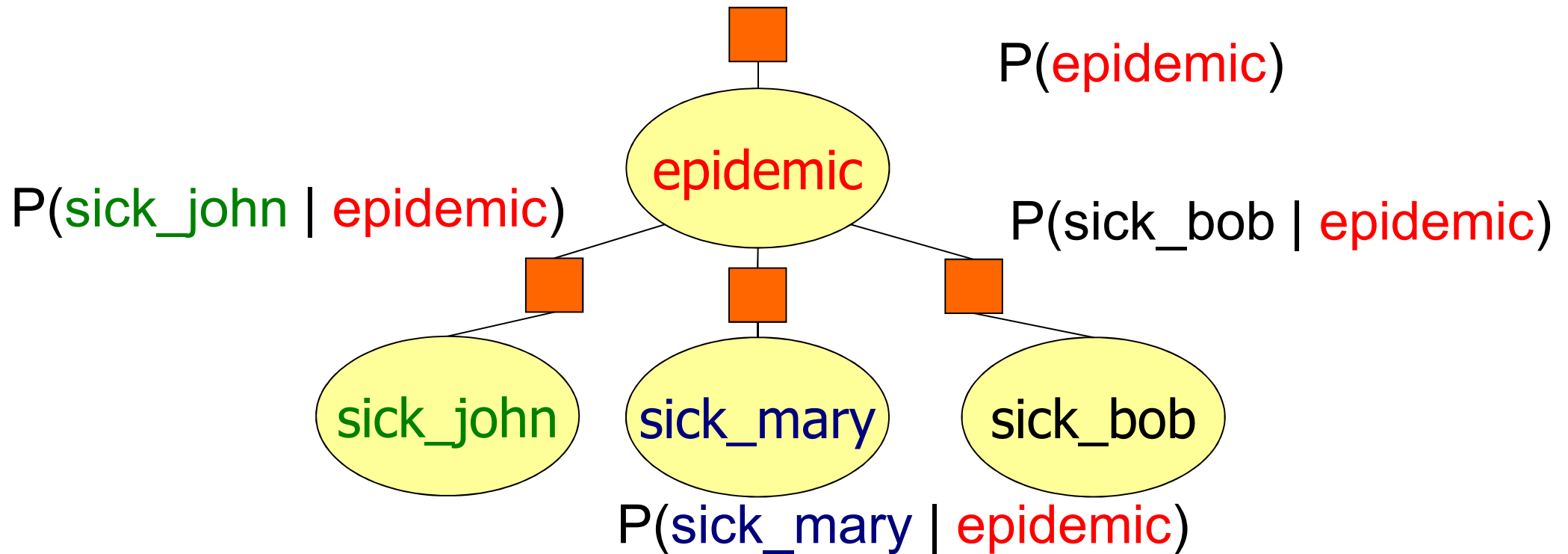
# Factor Networks (undirected)



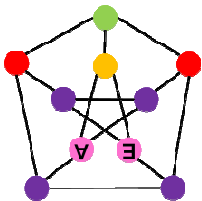
$$\begin{aligned} &P(\text{epi\_france, epi\_belgium, epi\_uk, epi\_germany}) \\ &\propto \phi_1(\text{epi\_france, epi\_belgium}) * \phi_2(\text{epi\_france, epi\_uk}) \\ &* \phi_3(\text{epi\_france, epi\_germany}) * \phi_4(\text{epi\_belgium, epi\_germany}) \end{aligned}$$



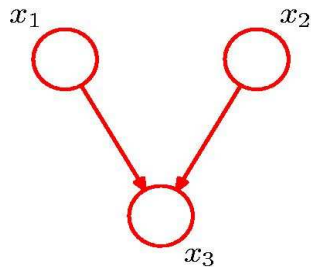
# Bayesian Nets as Factor Networks



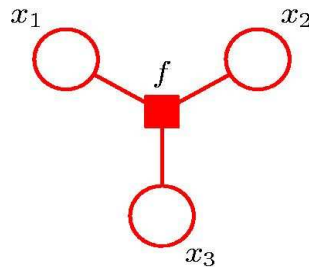
$$P(\text{sick\_john}, \text{sick\_mary}, \text{sick\_bob}, \text{epidemic}) \\ \propto P(\text{sick\_john} \mid \text{epidemic}) * P(\text{sick\_mary} \mid \text{epidemic}) \\ * P(\text{sick\_bob} \mid \text{epidemic}) * P(\text{epidemic})$$



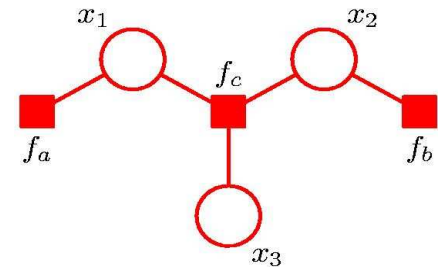
# Factor Graphs from Graphical Models



$$p(\mathbf{x}) = p(x_1)p(x_2) \\ p(x_3|x_1, x_2)$$



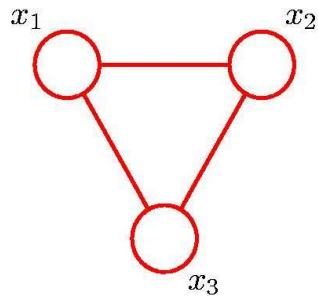
$$f(x_1, x_2, x_3) = \\ p(x_1)p(x_2)p(x_3|x_1, x_2)$$



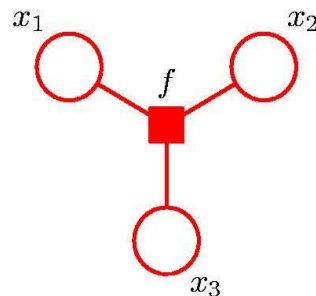
$$f_a(x_1) = p(x_1)$$

$$f_b(x_2) = p(x_2)$$

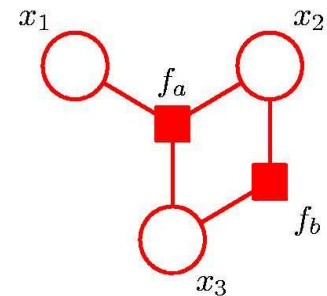
$$f_c(x_1, x_2, x_3) = p(x_3|x_1, x_2)$$



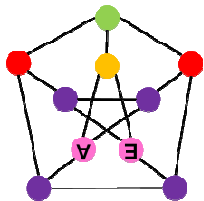
$$\psi(x_1, x_2, x_3)$$



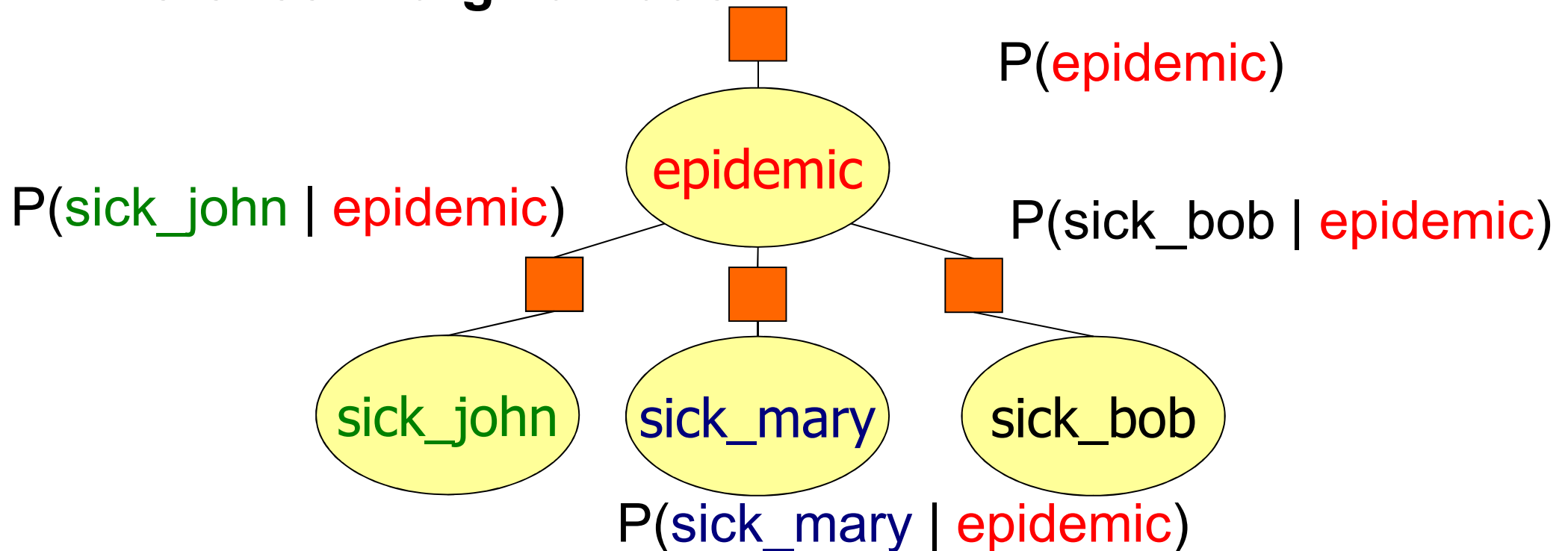
$$f(x_1, x_2, x_3) \\ = \psi(x_1, x_2, x_3)$$



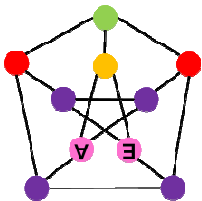
$$f_a(x_1, x_2, x_3)f_b(x_2, x_3) \\ = \psi(x_1, x_2, x_3)$$



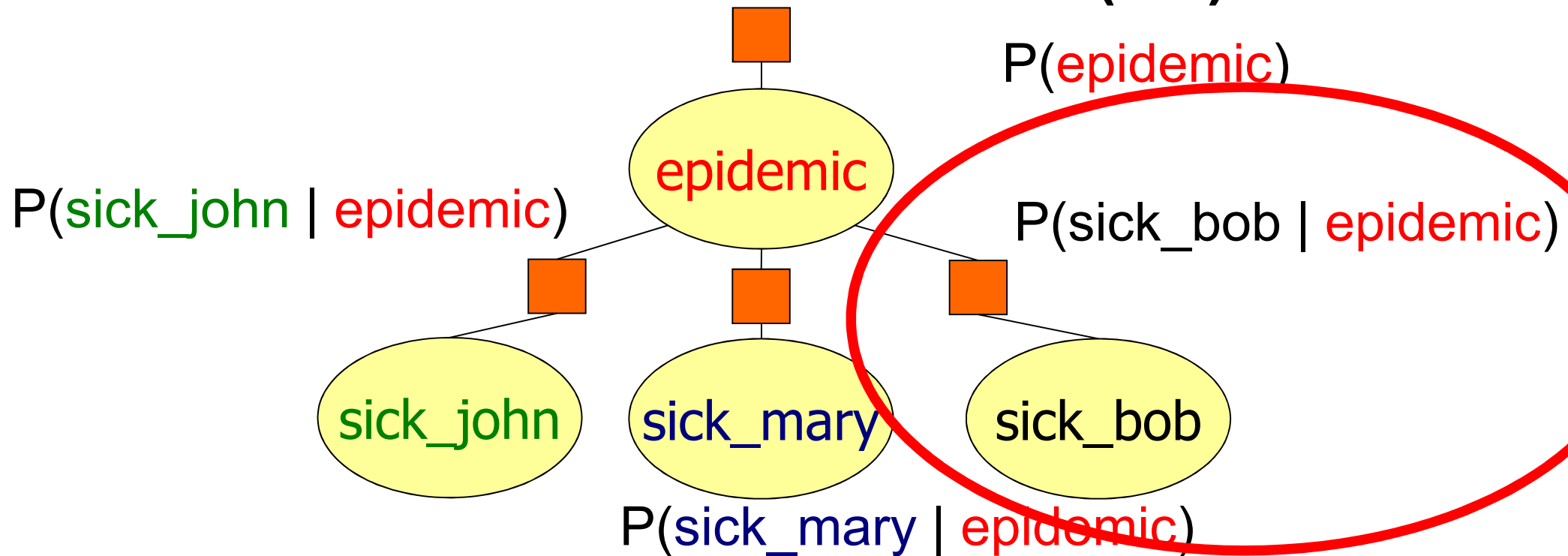
# Inference: Marginalization



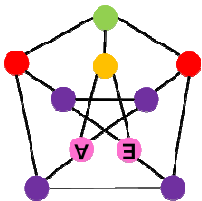
$$\begin{aligned}
 P(\text{sick\_john}) \propto & \sum_{\text{epidemic}} \sum_{\text{sick\_mary}} \sum_{\text{sick\_bob}} \\
 & P(\text{sick\_john} \mid \text{epidemic}) \\
 & * P(\text{sick\_mary} \mid \text{epidemic}) * P(\text{sick\_bob} \mid \text{epidemic}) \\
 & * P(\text{epidemic})
 \end{aligned}$$



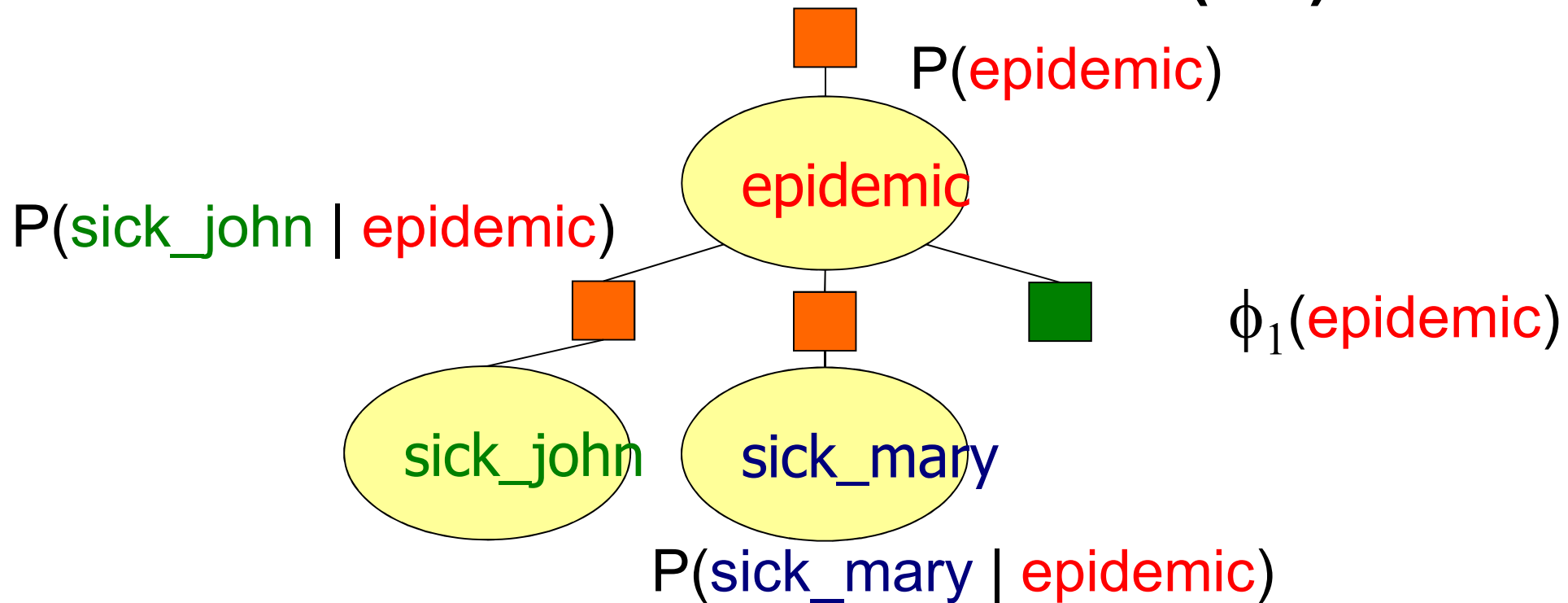
# Inference: Variable Elimination (VE)



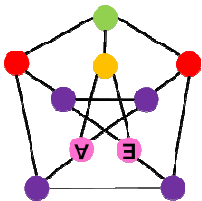
$$\begin{aligned}
 P(\text{sick\_john}) &\propto \sum_{\text{epidemic}} P(\text{sick\_john} \mid \text{epidemic}) * P(\text{epidemic}) \\
 &\quad * \sum_{\text{sick\_mary}} P(\text{sick\_mary} \mid \text{epidemic}) \\
 &\quad * \sum_{\text{sick\_bob}} P(\text{sick\_bob} \mid \text{epidemic})
 \end{aligned}$$



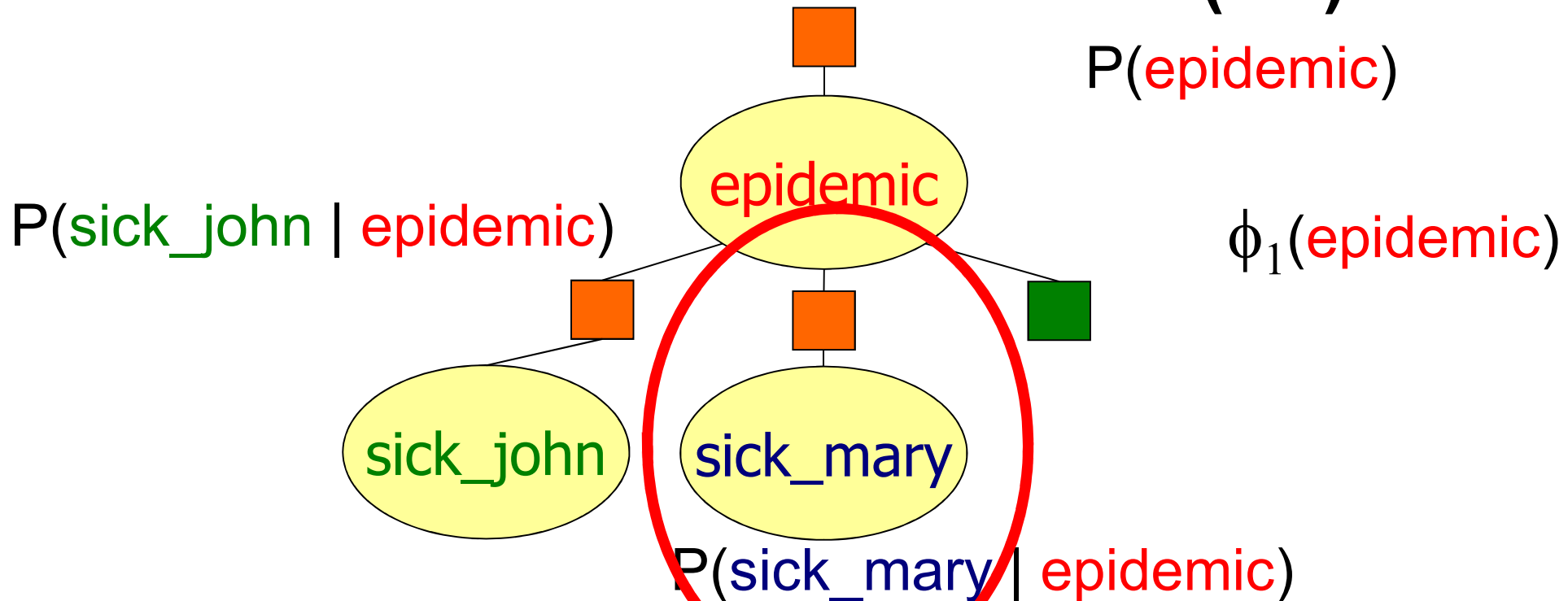
# Inference: Variable Elimination (VE)



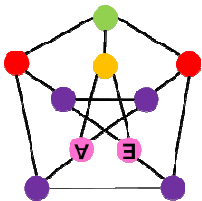
$$\begin{aligned}
 P(\text{sick\_john}) &\propto \sum_{\text{epidemic}} P(\text{sick\_john} \mid \text{epidemic}) * P(\text{epidemic}) \\
 &\quad * \sum_{\text{sick\_mary}} P(\text{sick\_mary} \mid \text{epidemic}) \\
 &\quad * \phi_1(\text{epidemic})
 \end{aligned}$$



# Inference: Variable Elimination (VE)

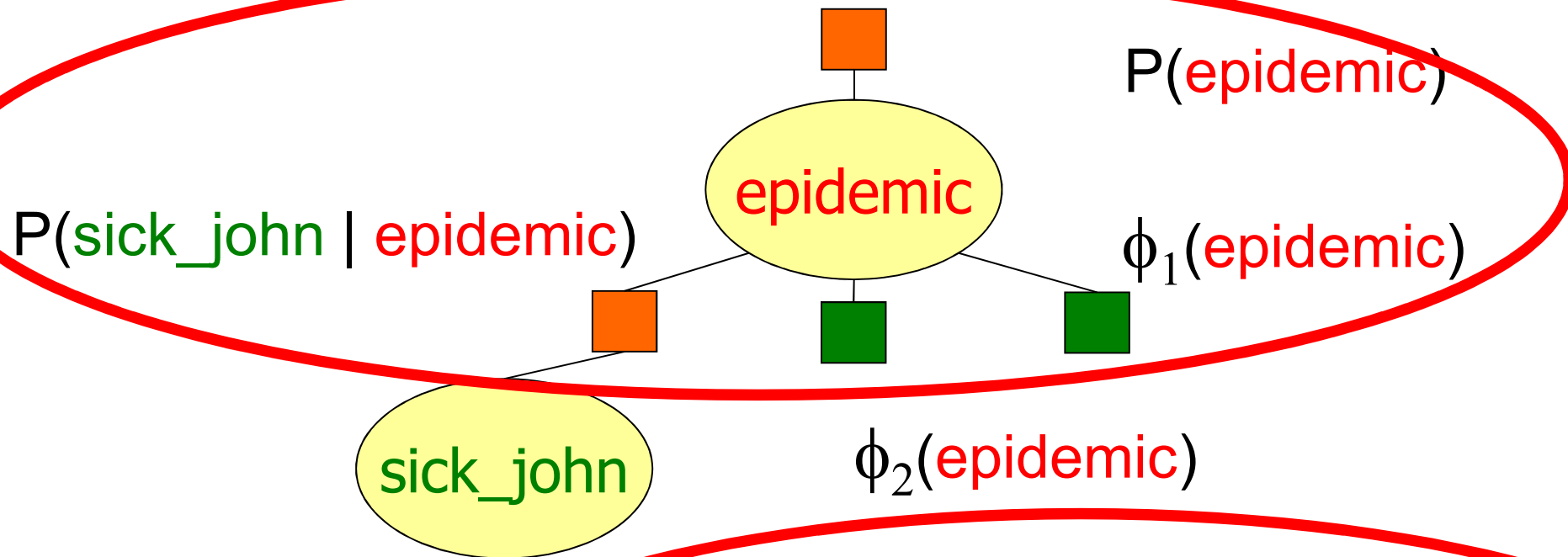


$$P(\text{sick\_john}) \propto \sum_{\text{epidemic}} P(\text{sick\_john} \mid \text{epidemic}) * P(\text{epidemic}) * \phi_1(\text{epidemic}) * \sum_{\text{sick\_mary}} P(\text{sick\_mary} \mid \text{epidemic})$$

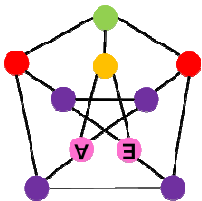




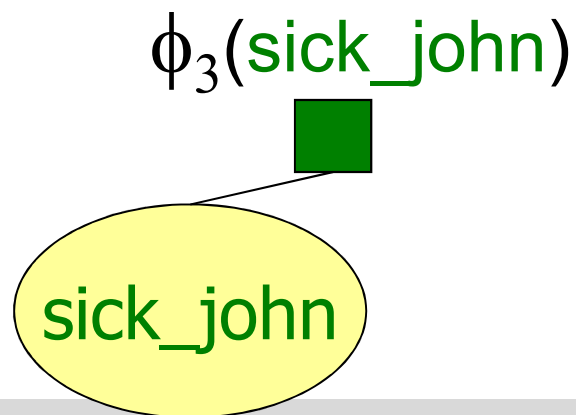
# Inference: Variable Elimination (VE)



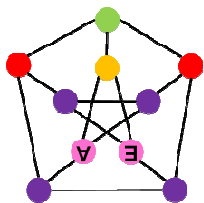
$$P(\text{sick\_john}) \propto \sum_{\text{epidemic}} P(\text{sick\_john} \mid \text{epidemic}) * P(\text{epidemic}) * \phi_1(\text{epidemic}) * \phi_2(\text{epidemic})$$



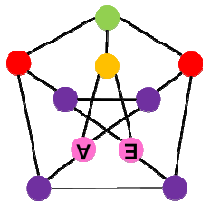
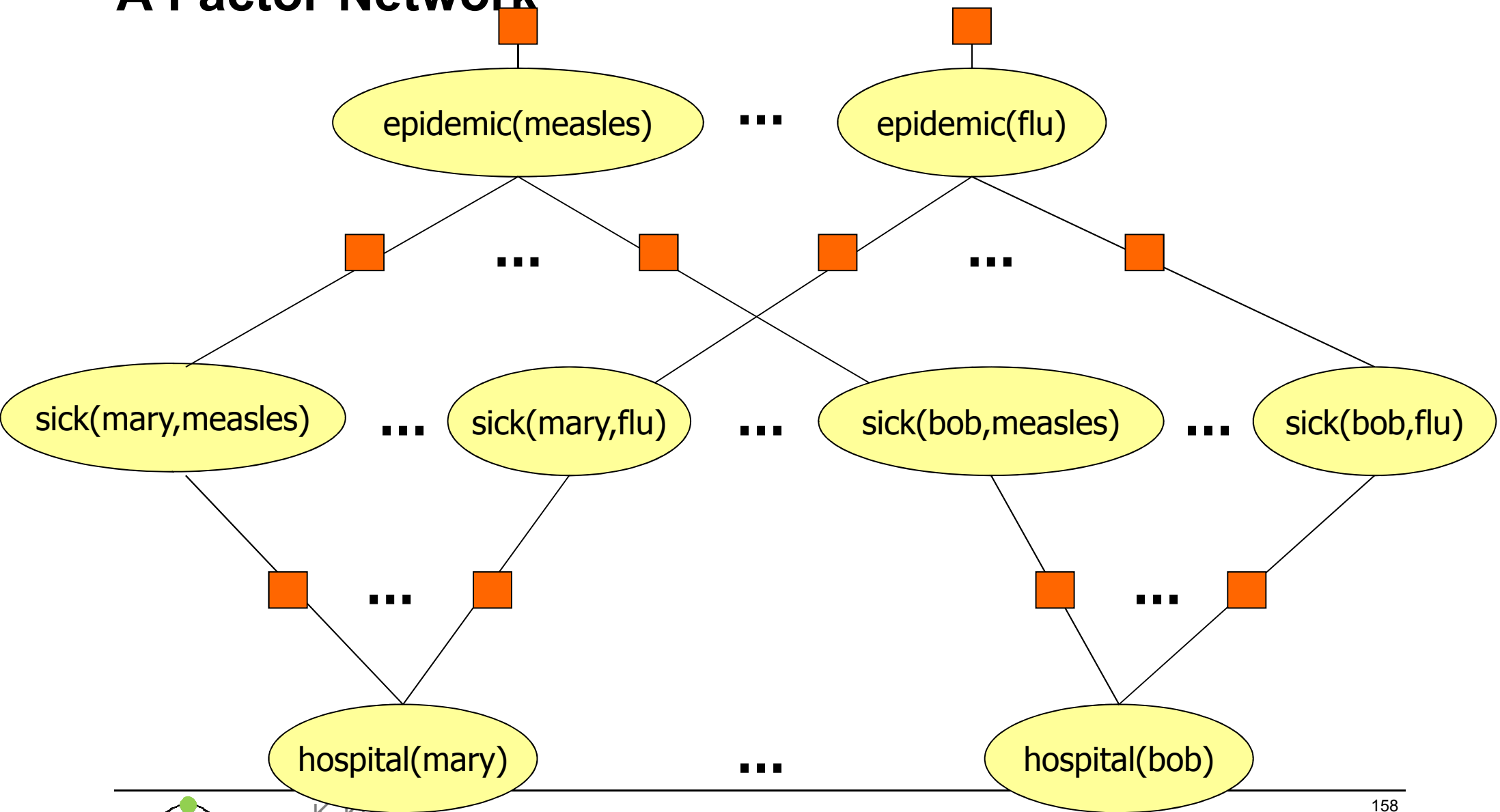
# Inference: Variable Elimination (VE)



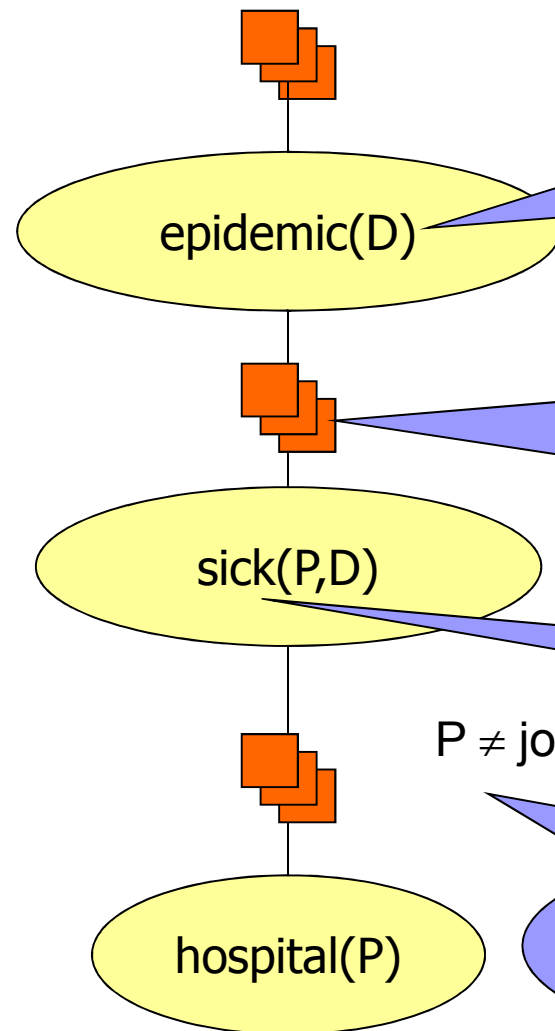
$$P(\text{sick\_john}) \propto \phi_3(\text{sick\_john})$$



# A Factor Network



# First-order Representation



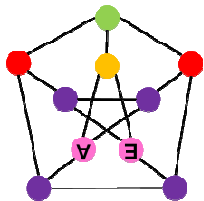
Logical Variables

parameterize random variables

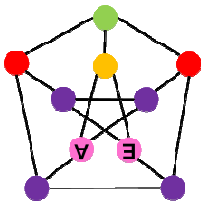
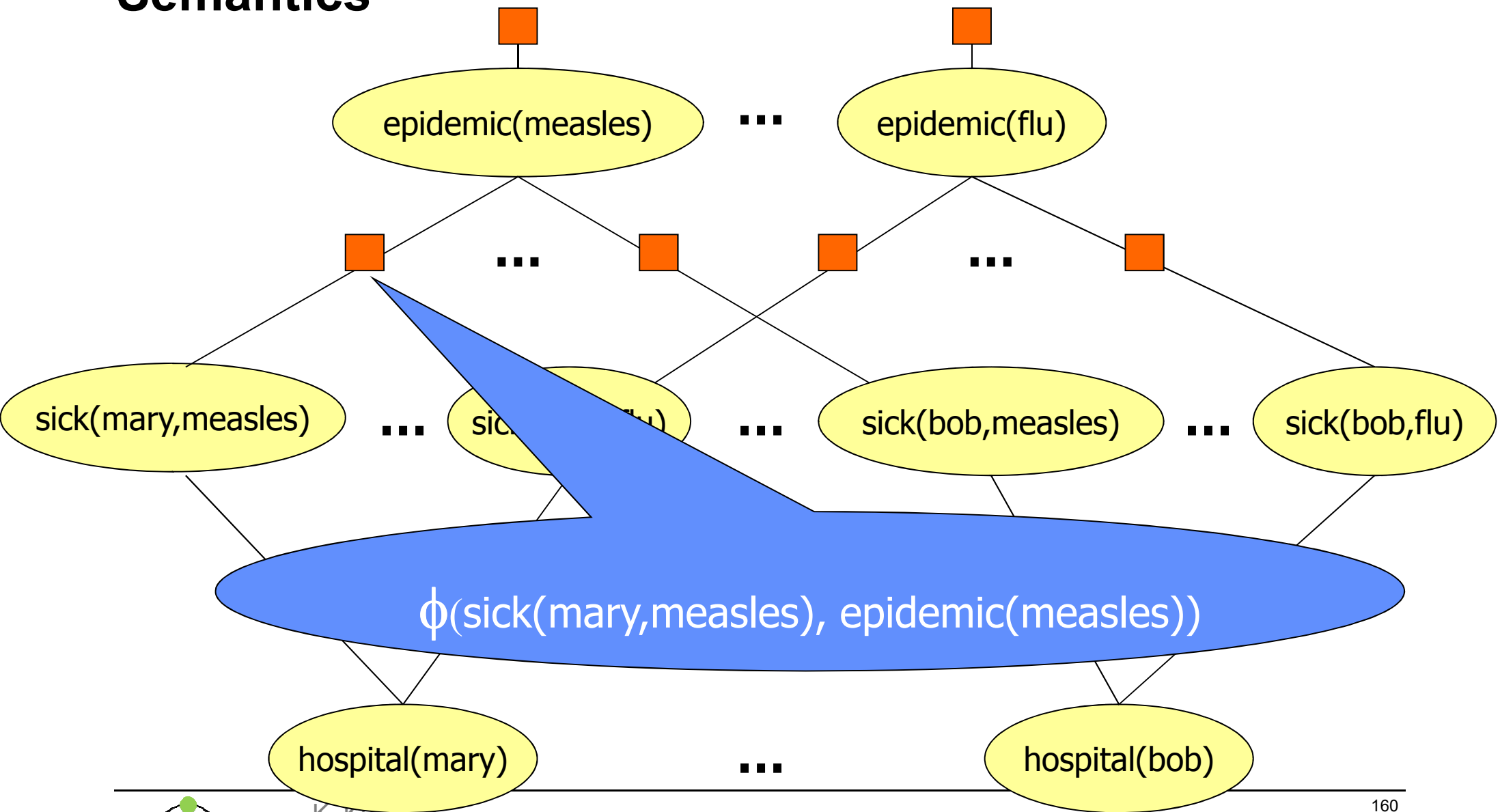
Parfactors, for  
parameterized factors

Atoms represent a set of  
random variables

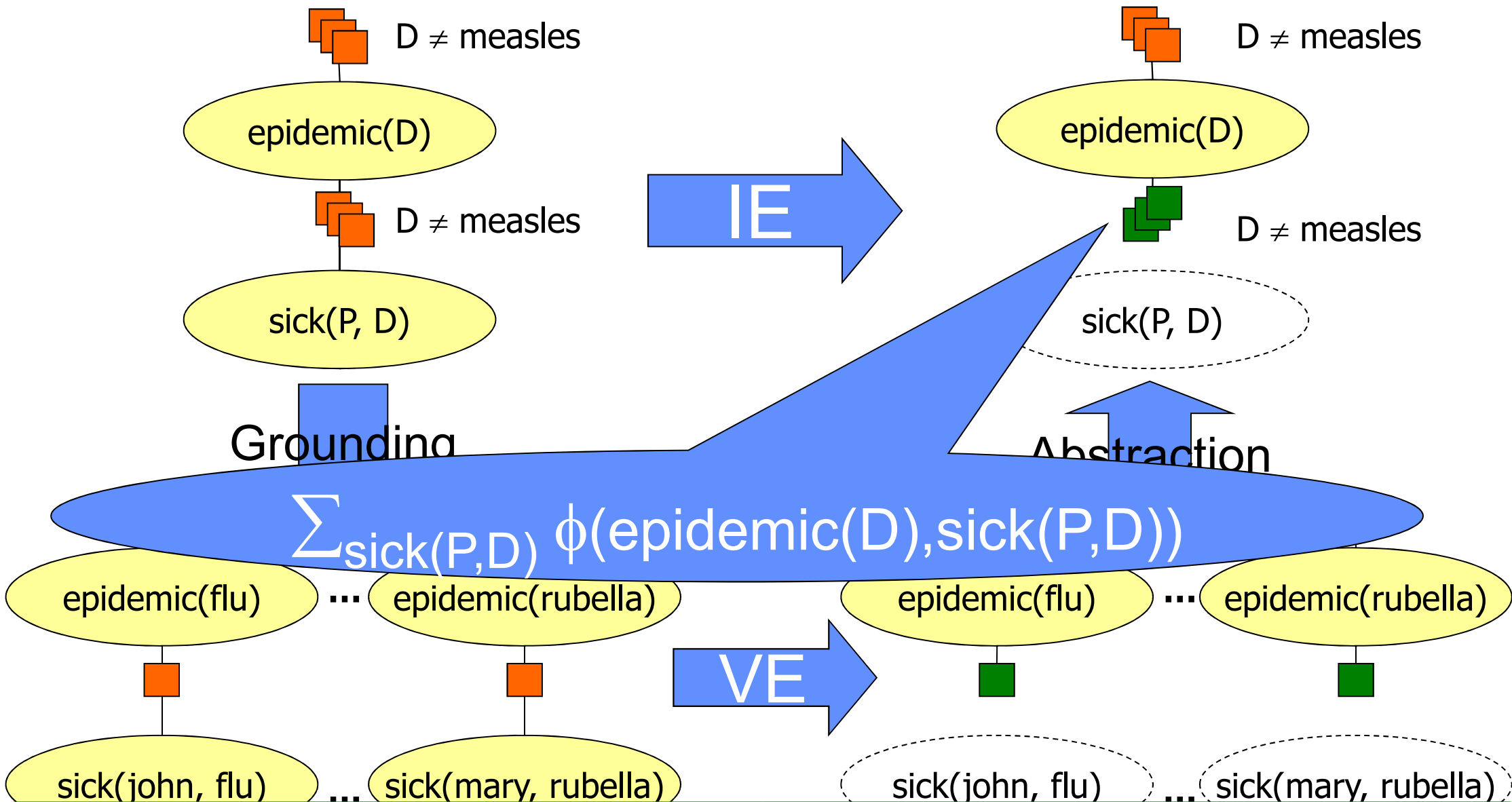
Constraints to logical  
variables



# Semantics



# Inversion Elimination (IE)

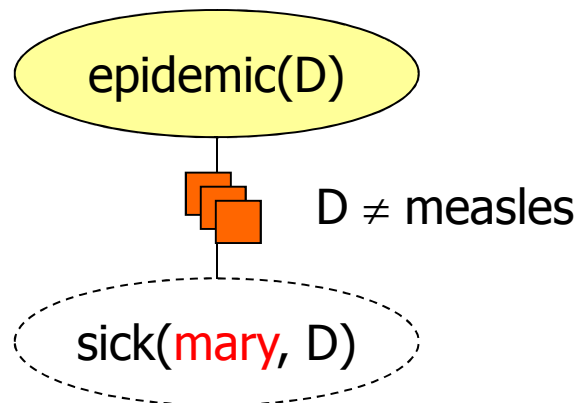


Grounding does **not** exploit symmetries encoded in the structure of the model

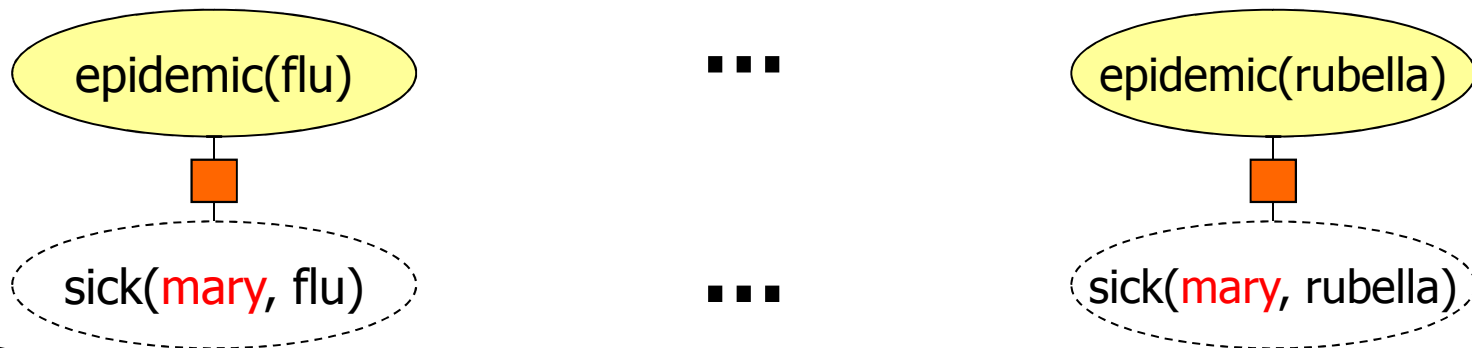
# Inversion Elimination - Limitations

Requires eliminated RVs to occur in separate instances of parfactor

$\phi'(\text{epidemic}(D))^n$



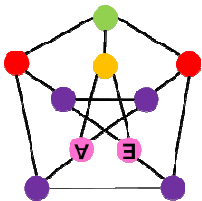
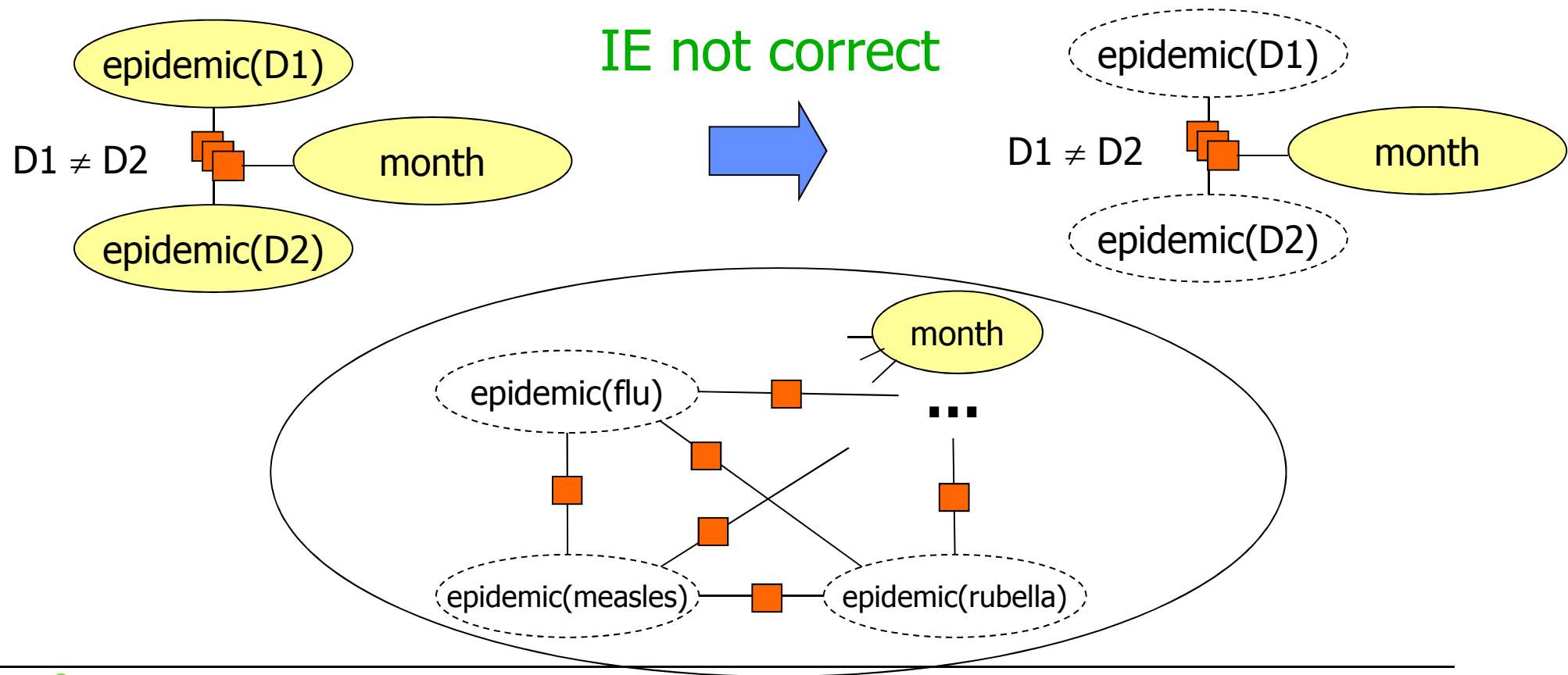
Inversion  
Elimination  
correct



Exploits symmetries encoded in the structure of the model

# Inversion Elimination - Limitations

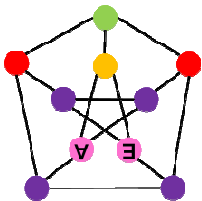
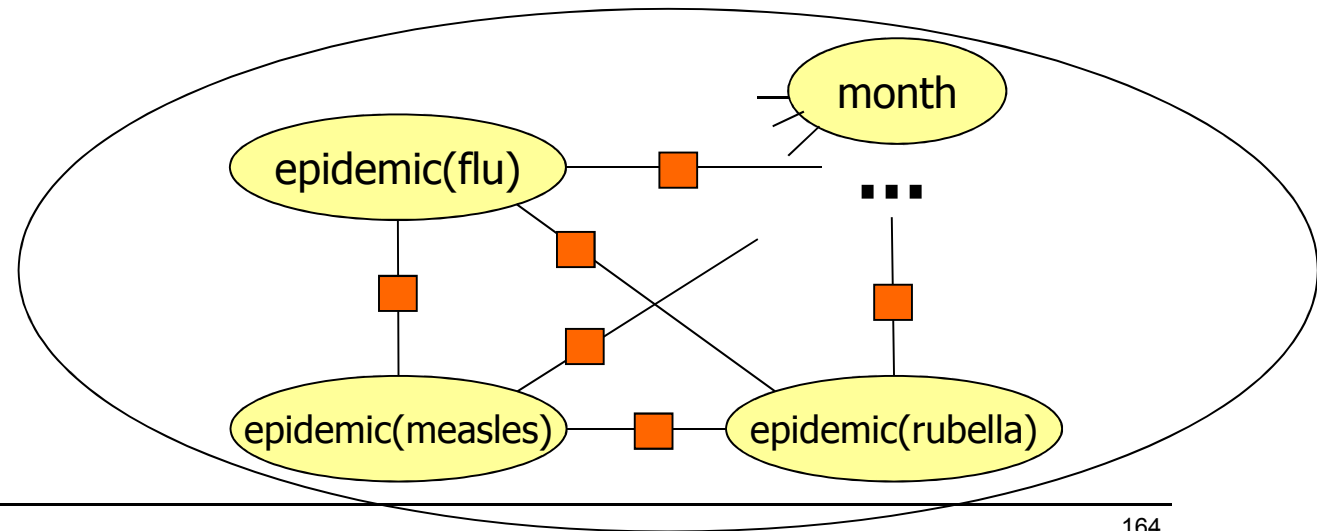
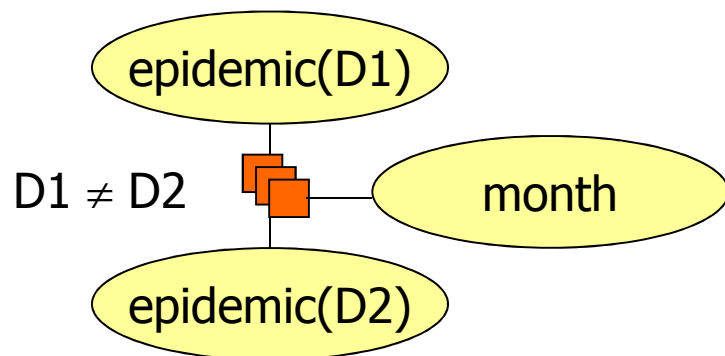
Requires eliminated RVs to occur in separate instances of parfactor



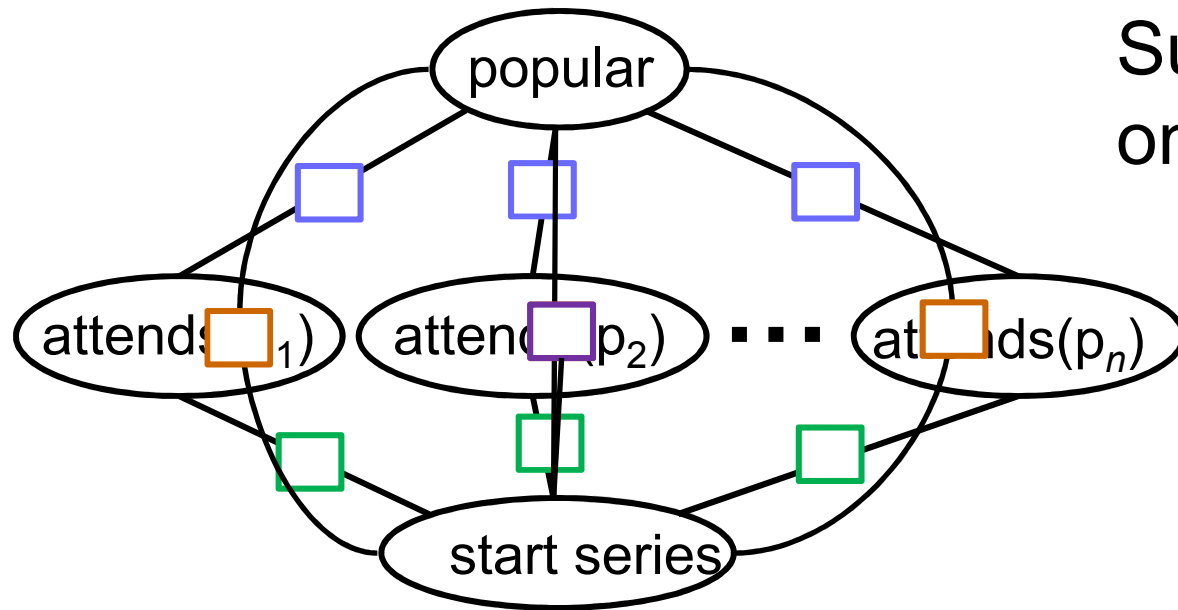


# Counting Elimination

- Need to consider joint assignments;
- Exponential number of those;
- But actually, potential depends on histogram of values in assignment only: 00101 the same as 11000;
- Polynomial number of assignments instead.



## A simpler Example: Inviting $n$ people to a workshop



Sum out non-query variables one by one

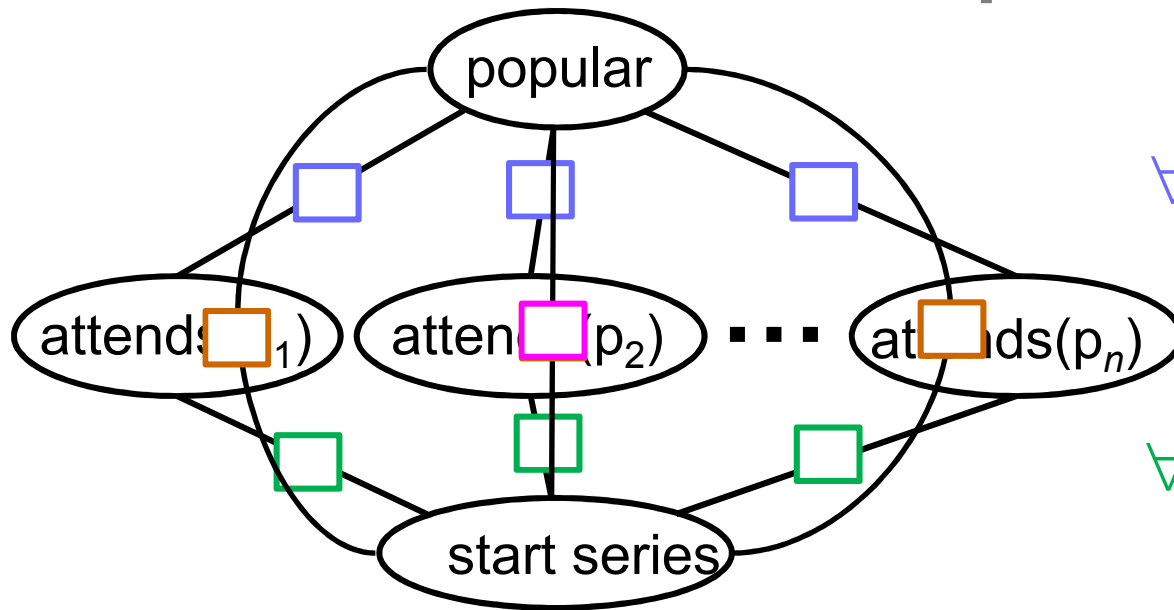
Time is **linear** in number of invitees  $n$

$$\sum_{\text{att}(p_1)} \underbrace{\phi_1(\text{pop}, \text{att}(p_1)) \phi_2(\text{att}(p_1), \text{ser})}_{\phi'(\text{pop}, \text{ser})}$$

Does **not** exploit symmetries encoded in the structure of the model

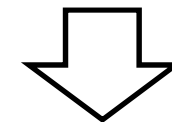
# First-Order Variable Elimination

[Poole 2003; de Salvo Braz *et al.* 2005]

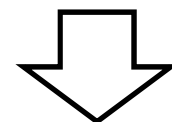


$$\forall X. \phi_1(\text{popular}, \text{attends}(X))$$

$$\forall X. \phi_2(\text{attends}(X), \text{series})$$



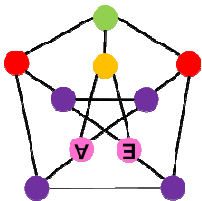
$$\forall X. \phi'(\text{popular}, \text{series})$$



$$\phi'(\text{popular}, \text{series})^n$$

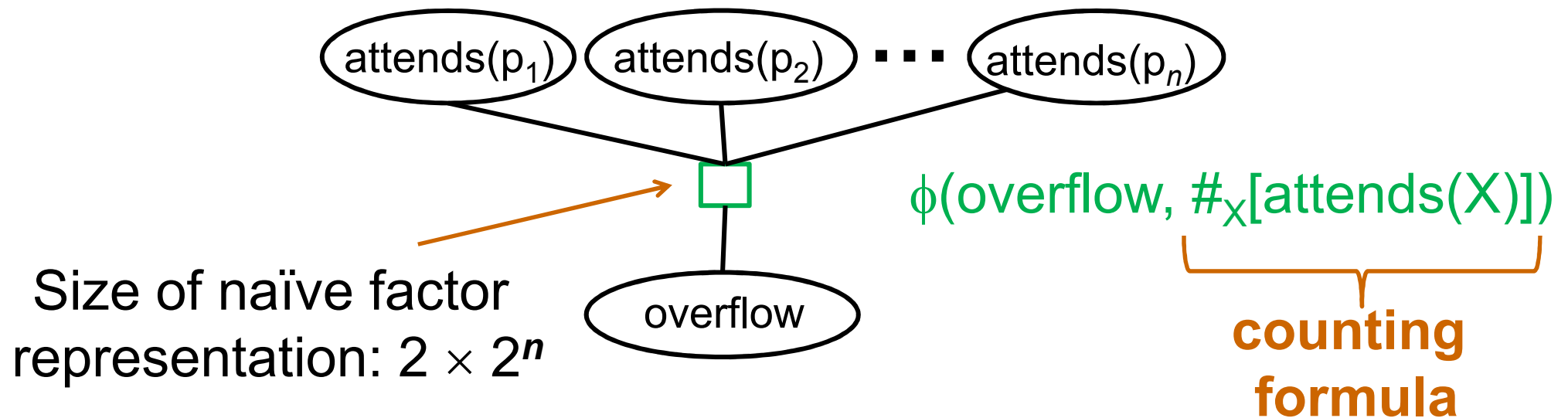
Sum out all  $\text{attends}(X)$  variables at once

Time is **constant** in  $n$

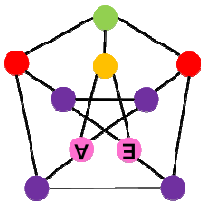


# Symmetry Within Factors

[Milch, Zettlemoyer, Haims, K, Kaelbling AAAI08]

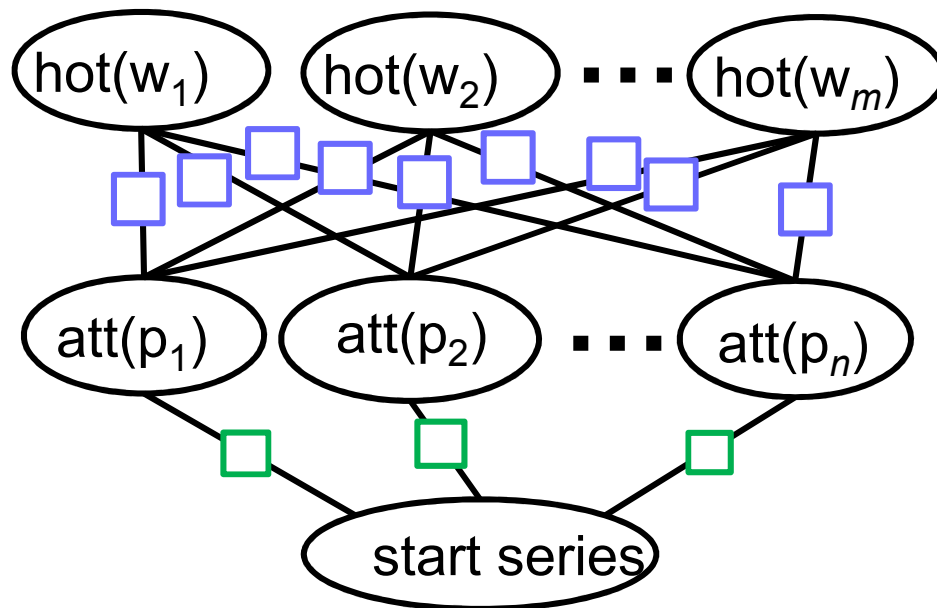


- Values of counting formula are **histograms** counting how many objects  $X$  yield each possible value of  $\text{attends}(X)$ 
  - Only  $n+1$  histograms, e.g.,  $[50, 0]$ ,  $[49, 1]$ , ...,  $[0, 50]$
  - Factor size now  $2 \times (n+1)$ : **linear** in  $n$

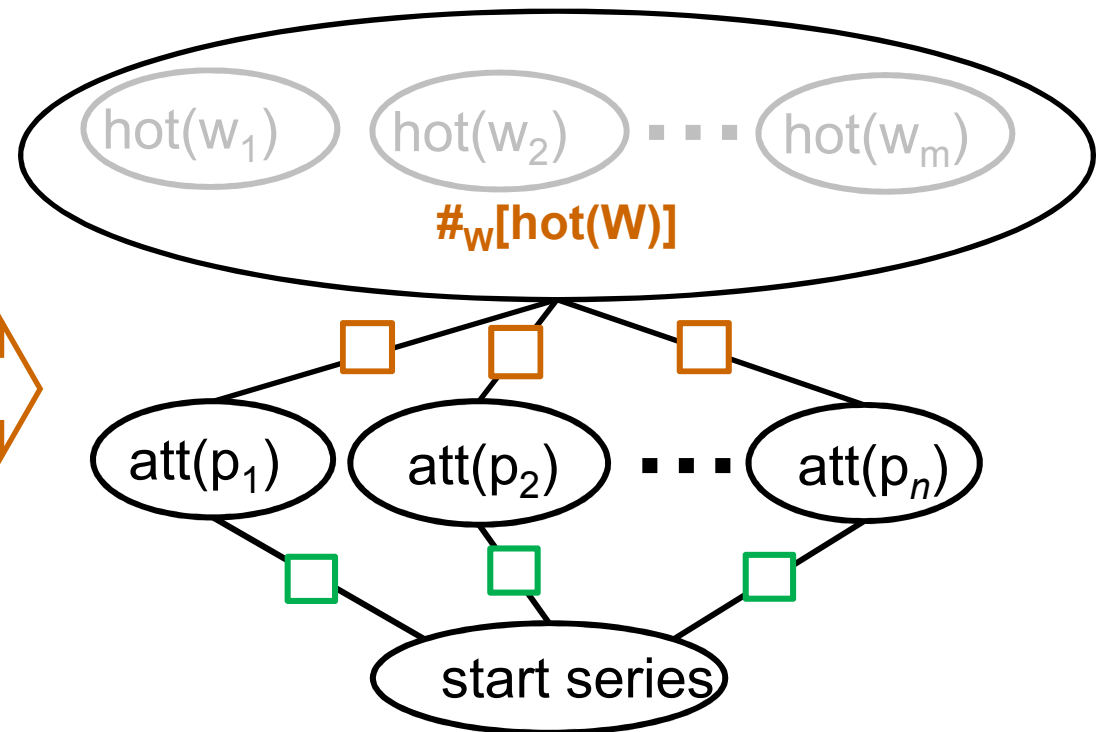


## Example: Competing Workshops

$$\forall W \forall X. \phi(\text{hot}(W), \text{att}(X))$$



Can't sum out  $\text{attends}(X)$  without joining all the  $\text{hot}(W)$  variables



Create counting formula on  $\text{hot}(W)$ , then sum out  $\text{attends}(X)$  at lifted level

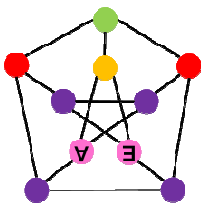
Conversion to counting formulas creates new opportunities for lifted elimination

# Results: Competing Workshops

What about approximate inference approaches?

Time (ms)

- These exact inference approaches are rather complex
- so far do not easily scale to realistic domains,
- and hence have only been applied to rather small artificial problems



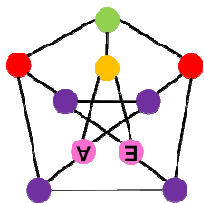
# How do you spend your spare time?



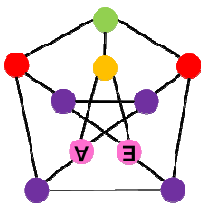
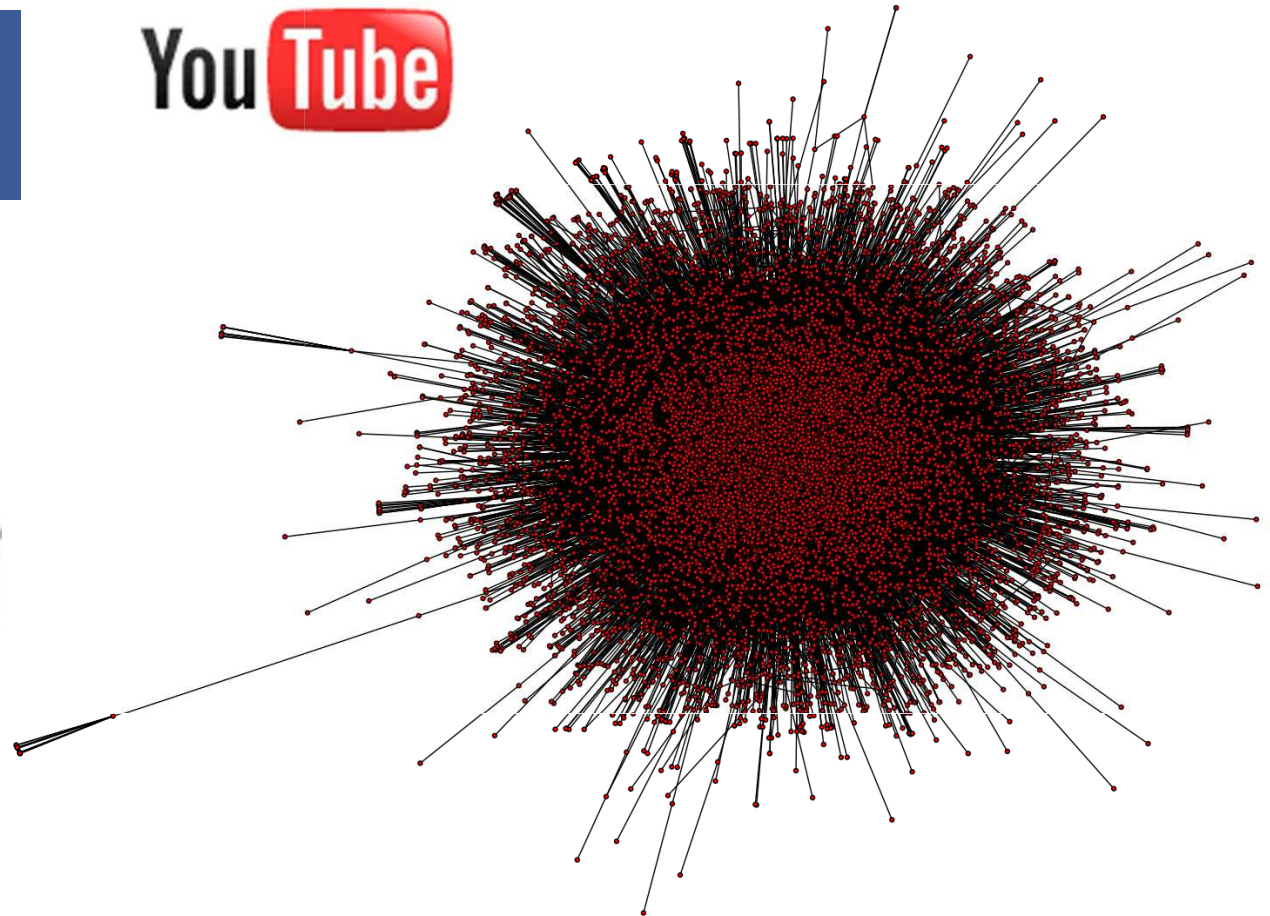
YouTube like media portals have changed the way users access media content in the Internet

Every day, millions of people visit social media sites such as Flickr, YouTube, and Jumpcut, among others, to share their photos and videos, ...

while others enjoy themselves by searching, watching, commenting, and rating the photos and videos; what your friends like will bear great significance for you.



# How do you efficiently broadcast information?



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



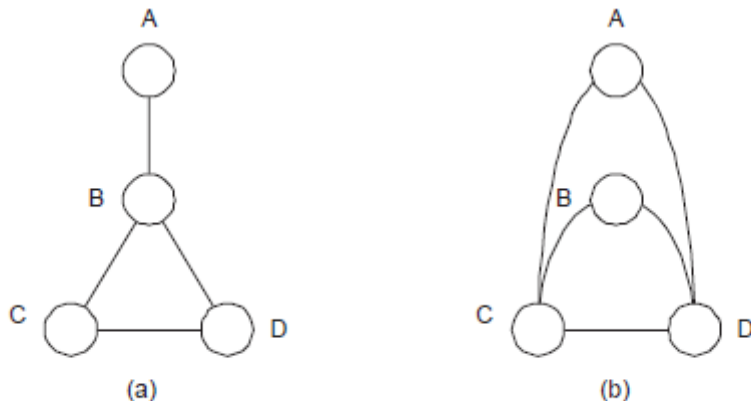
**Fraunhofer**





# Content Distribution using Stochastic Policies

[Bickson et al. WDAS04]



Add an edge potential between any two nodes which can take a part from a common neighboring node.

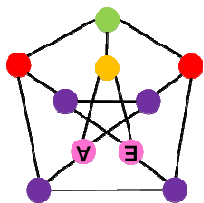
$\psi_{BB}(X_B)$	3 from A	1 from C	3 from C	3 from D
Uniform policy	1/4	1/4	1/4	1/4
Rarest part first	1/6	1/2	1/6	1/6

Table 2. Possible actions for the node B in the example graph shown in Figure 1

$\psi_{BD}(X_B, X_D)$	3 from A	2 from C	3 from C	3 from D
1 from B	1/8	1/8	1/8	1/8
2 from C	1/2	$\epsilon$	$\epsilon$	1/2
3 from C	1/2	$\epsilon$	$\epsilon$	1/2

Table 3. Example of the edge potentials for the edge BD for the graph shown in Figure 2. The matrix rows are then normalized.

Large (distributed) networks, so Bickson et al. propose to use (loopy) belief propagation



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



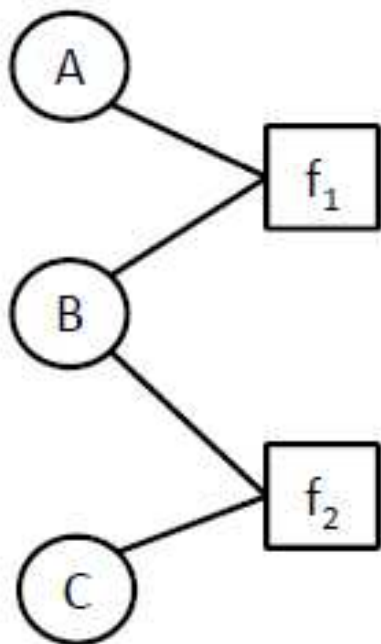
Fraunhofer



# The Sum-Product Algorithm aka Belief Propagation

- Iterative process in which neighboring variables “talk” to each other, passing **messages such as:**

*“I (variable  $x_3$ ) think that you (variable  $x_2$ ) belong in these states with various likelihoods...”*



A	B	$f_1$
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4

C	B	$f_2$
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4



$$\mu_{X \rightarrow f}(x) = \prod_{h \in \text{nb}(X) \setminus \{f\}} \mu_{h \rightarrow X}(x)$$

$$\mu_{f \rightarrow X}(x) = \sum_{\neg\{x\}} \left( f(\mathbf{x}) \prod_{y \in \text{nb}(f) \setminus \{X\}} \mu_{y \rightarrow f}(y) \right)$$

# Loopy Belief Propagation

- After enough iterations, this series of conversations is **likely** to converge to a consensus that determines the marginal probabilities of all the variables.
- Sum-Product/BP
  - **(1) update messages until convergence**
  - **(2) compute single node marginals**
- Variants exist for solving
  - SAT problems,
  - systems of linear equations,
  - matching problems and
- for arbitrary distributions (based on sampling)

**A lot of shared factors, so use lifted belief propagation**



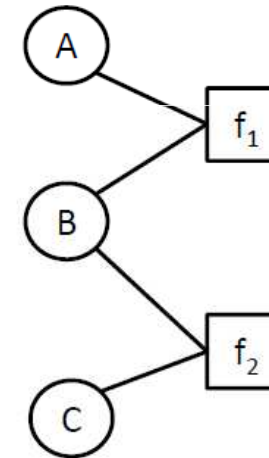
# Lifted Belief Propagation

[Singla, Domingos AAAI08, K, Ahmadi, Natarajan UAI09]

Counting shared factors can result in great efficiency gains for (loopy) belief propagation



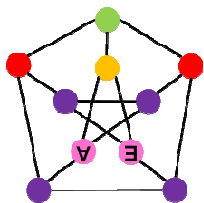
Shared factors appear more often than you think in relevant real world problems



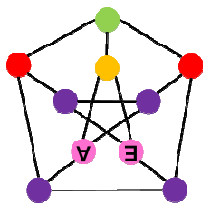
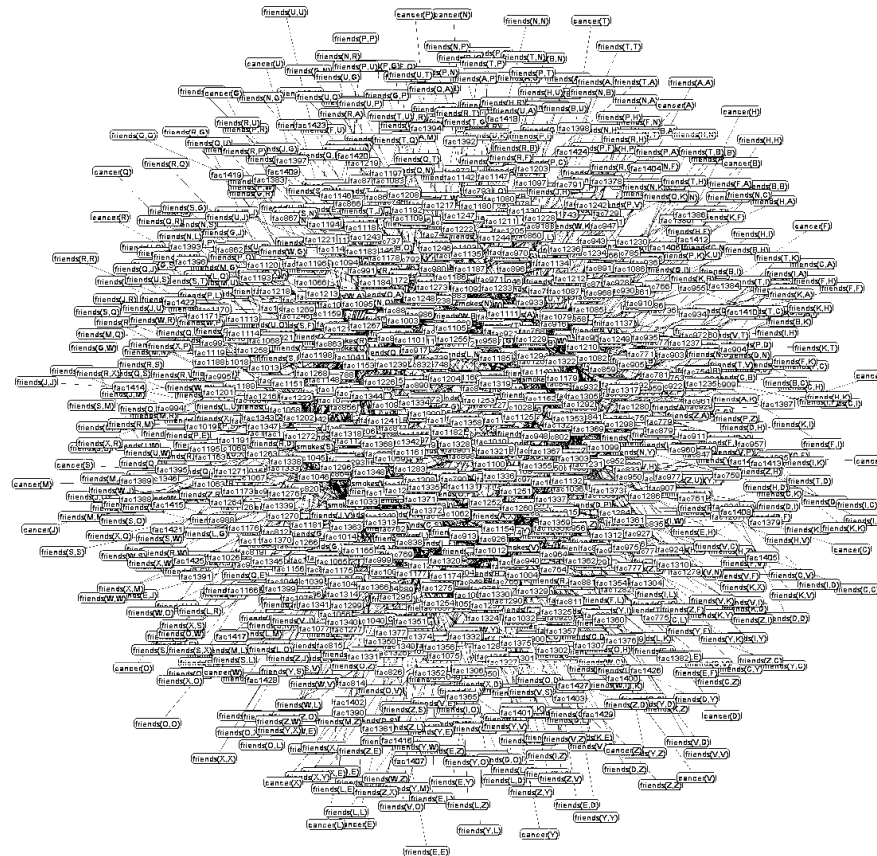
A	B	$f_1$
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4

C	B	$f_2$
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4

identical



# Social Network Analysis



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010

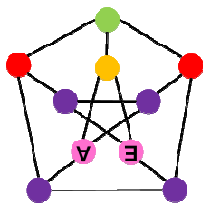
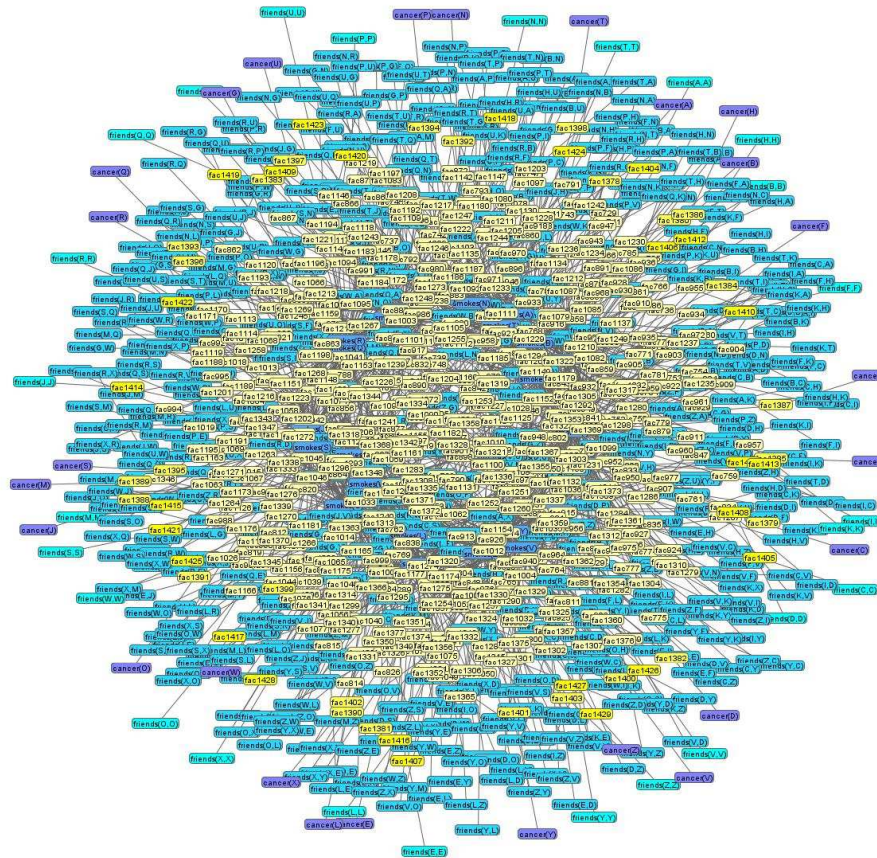


**Fraunhofer**





# Social Network Analysis



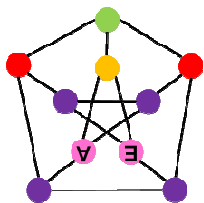
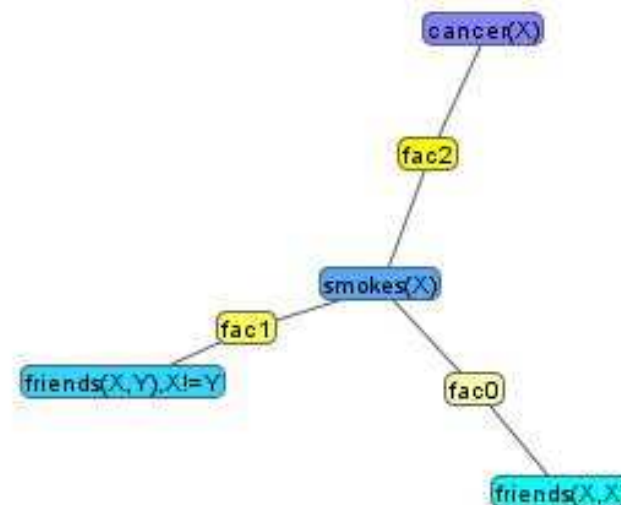
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



Fraunhofer



# Social Network Analysis



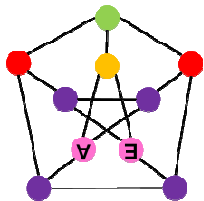
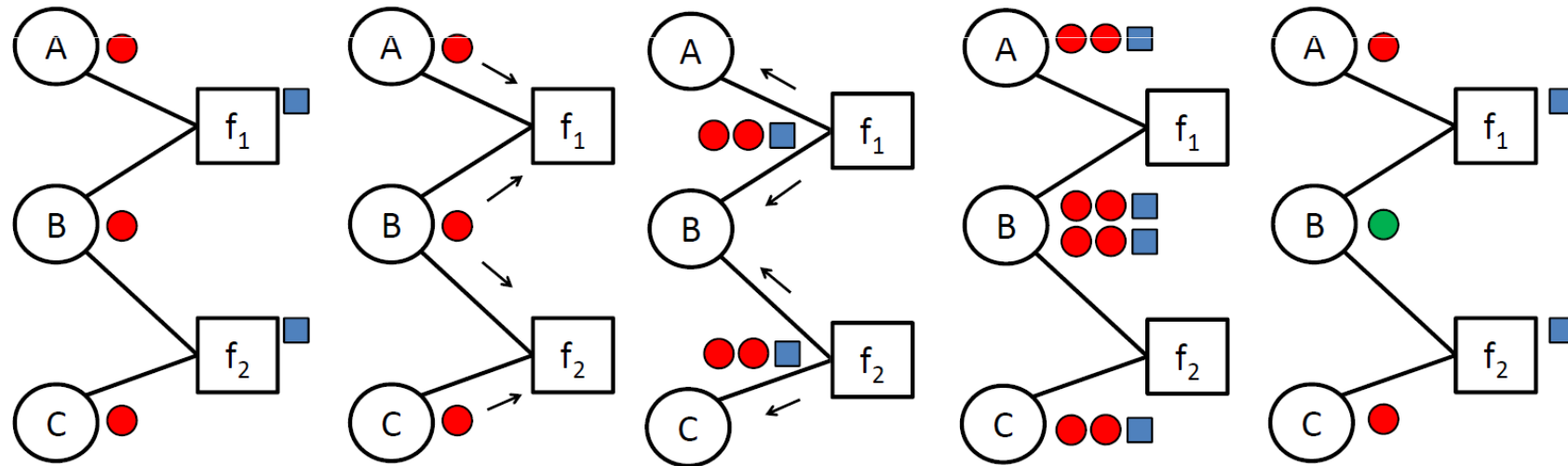
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**

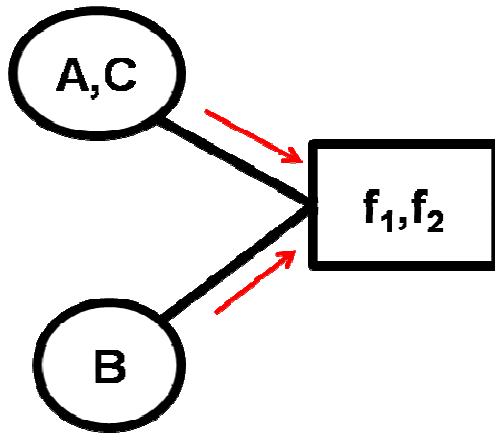


## Step 1: Compression

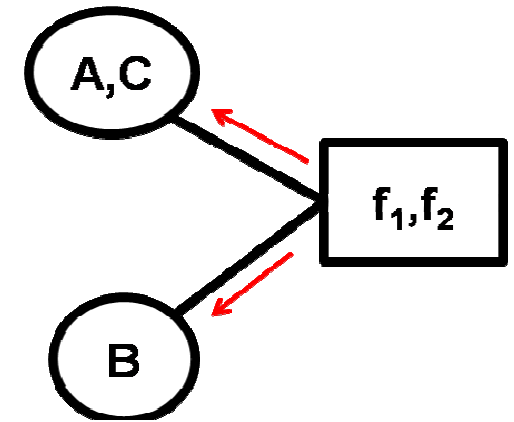




## Step 2: Modified Belief Propagation

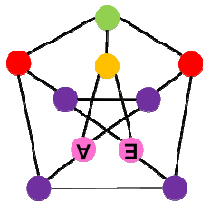


$$\mu_{\mathcal{X} \rightarrow f}(x) = u_{f \rightarrow \mathcal{X}}(x)^{c(f, \mathcal{X})-1} \cdot \prod_{h \in \text{nb}(\mathcal{X}) \setminus \{f\}} \mu_{h \rightarrow \mathcal{X}}(x)^{c(h, \mathcal{X})}$$



$$\mu_{f \rightarrow \mathcal{X}}(x) = \prod_{f \in \text{nb}(\mathcal{X}_i)} \mu_{f \rightarrow \mathcal{X}_i}(x_i)$$

$$b_i(x_i) = \prod_{f \in \text{nb}(\mathcal{X}_i)} \mu_{f \rightarrow \mathcal{X}_i}(x_i)^{c(f, \mathcal{X})}$$



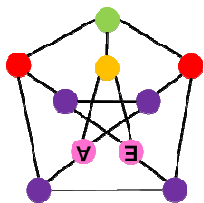
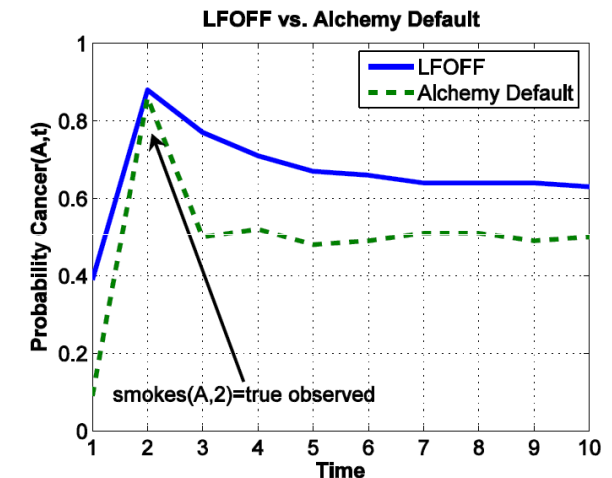
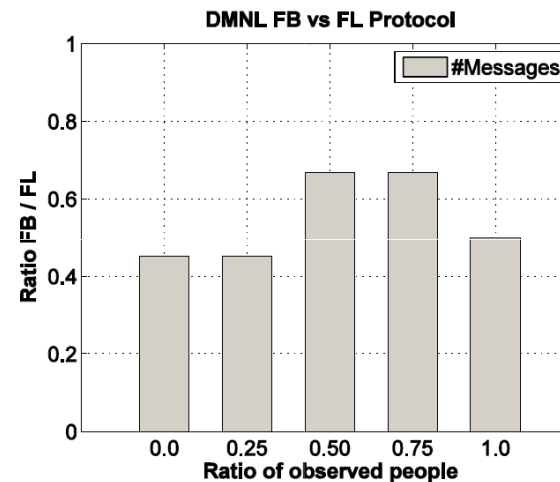
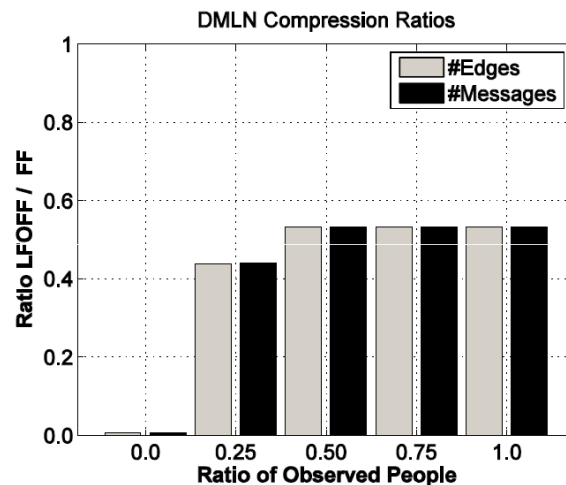
# Lifted Factored Frontier



Apriori most people do not smoke  
 Apriori most people do not have cancer  
 Apriori most people are not friends  
 Smoking causes cancer  
 Friends have similar smoking habits  
 Most friends stay friends  
 Most smokers stay smokers

$\neg \text{Smokes}(x, 0)$   
 $\neg \text{Cancer}(x, 0)$   
 $\neg \text{Friends}(x, y, 0)$   
 $\text{Smokes}(x, t) \Rightarrow \text{Cancer}(x, t)$   
 $\text{Friends}(x, y, t) \Rightarrow (\text{Smokes}(x, t) \Leftrightarrow \text{Smokes}(y, t))$   
 $\text{Friends}(x, y, t) \Leftrightarrow \text{Friends}(x, y, \text{succ}(t))$   
 $\text{Smokes}(x, t) \Leftrightarrow \text{Smokes}(x, \text{succ}(t))$

20 people over 10 time steps. Max number of friends 5. Cancer never observed.  
 Time step randomly selected.



# Lower Bound on Model Counts of CNF

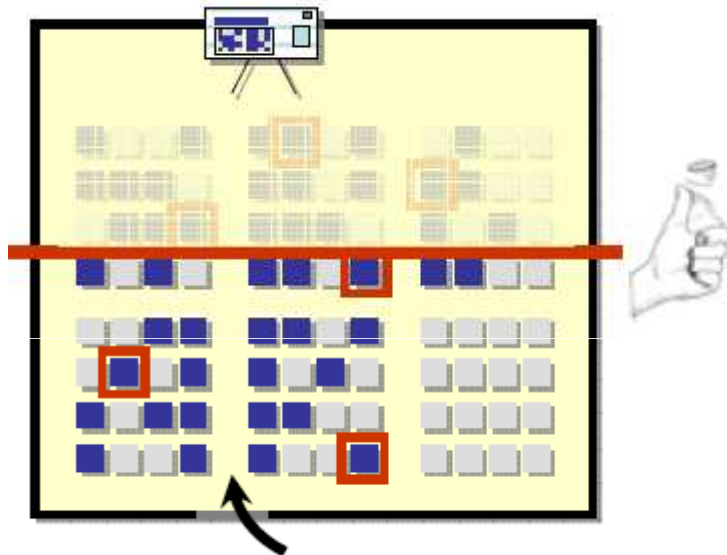
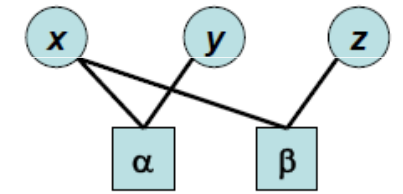
- BPCount [Kroc et al 08]
  - BP used to estimate marginals
  - Provable bound

e.g. SAT Problem:

$$\underbrace{(x \vee y)}_{\alpha} \wedge \underbrace{(\neg x \vee z)}_{\beta}$$



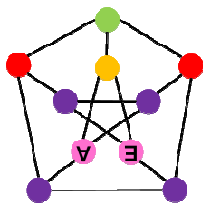
Factor Graph:



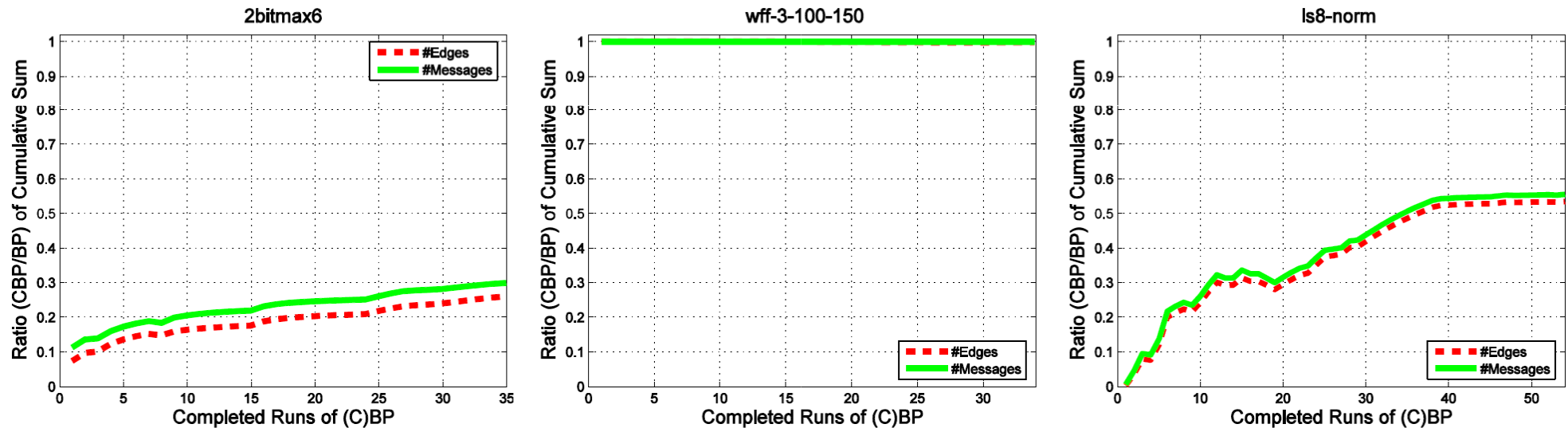
Idea:

- Identify a “balanced” row split or column split (roughly equal number of solutions on each side)
  - Use marginals for estimate
- Pick one side *at random*
- Count on that side recursively
- Multiply result by 2

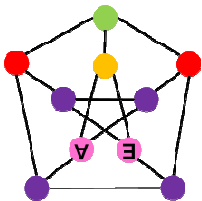
[similar to decimation]



# Model Counting



Satisfied by Lifted Message Passing?



K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010

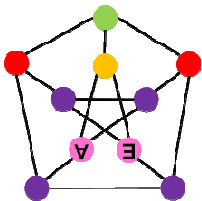
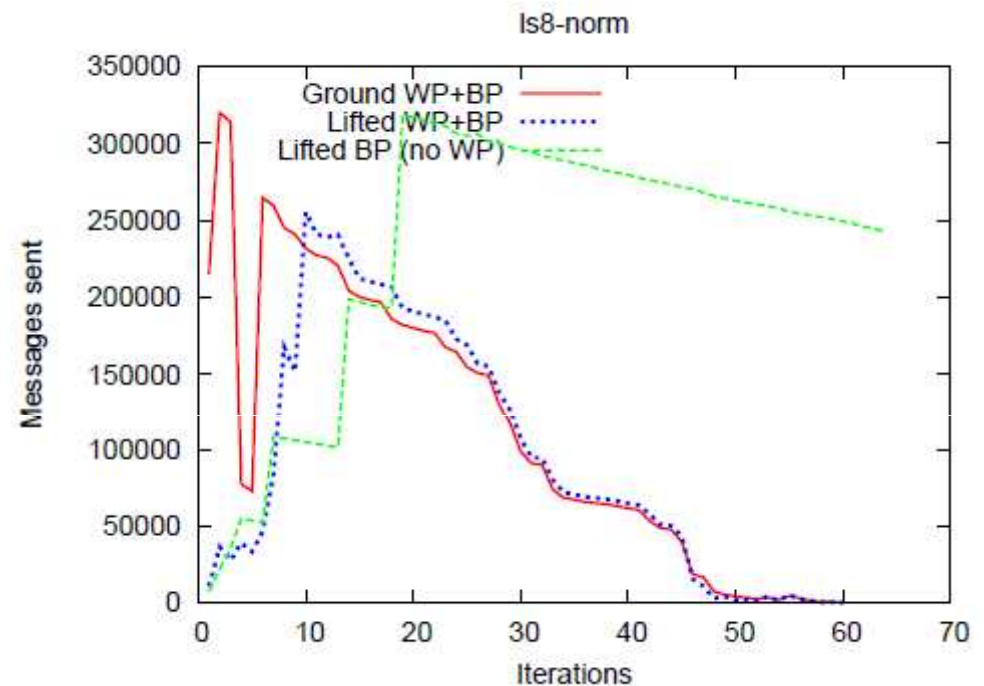
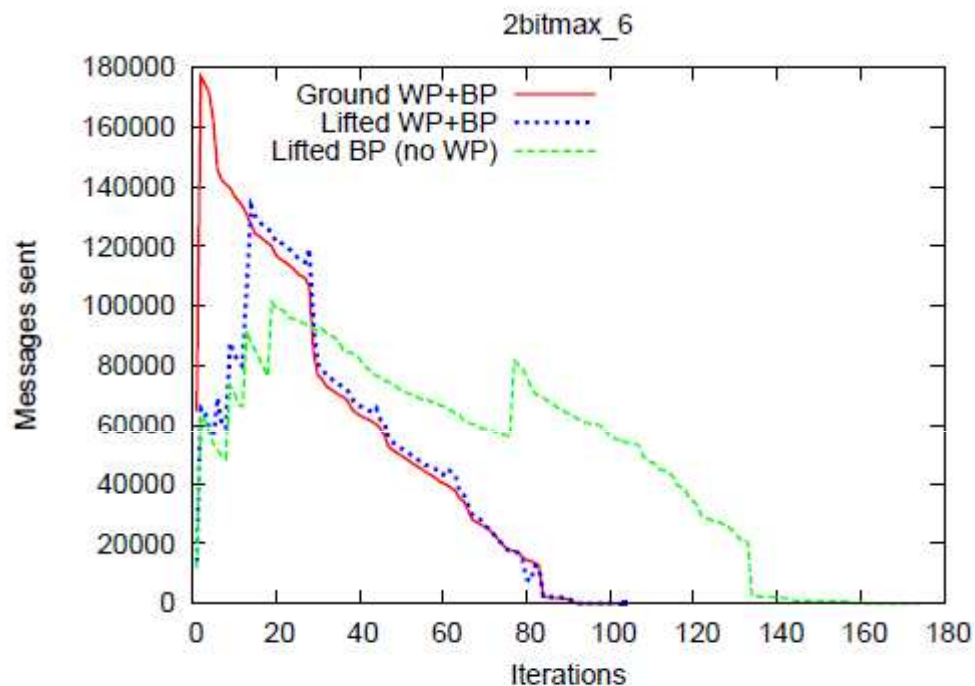


Fraunhofer



# Lifted Satisfiability [Hadiji, K, Ahmadi StarAI10]

- Warning and survey propagation can also be lifted
- Enables lifted treatment of both prob. and det. knowledge



[Ongoing work]

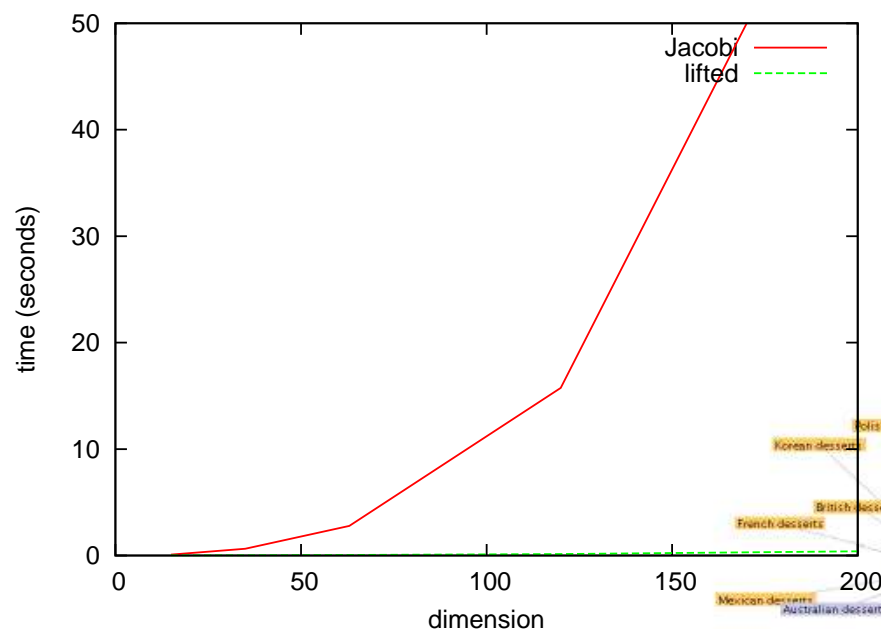
Bayati et al. ICDM09]

## Lifted Linear Equations

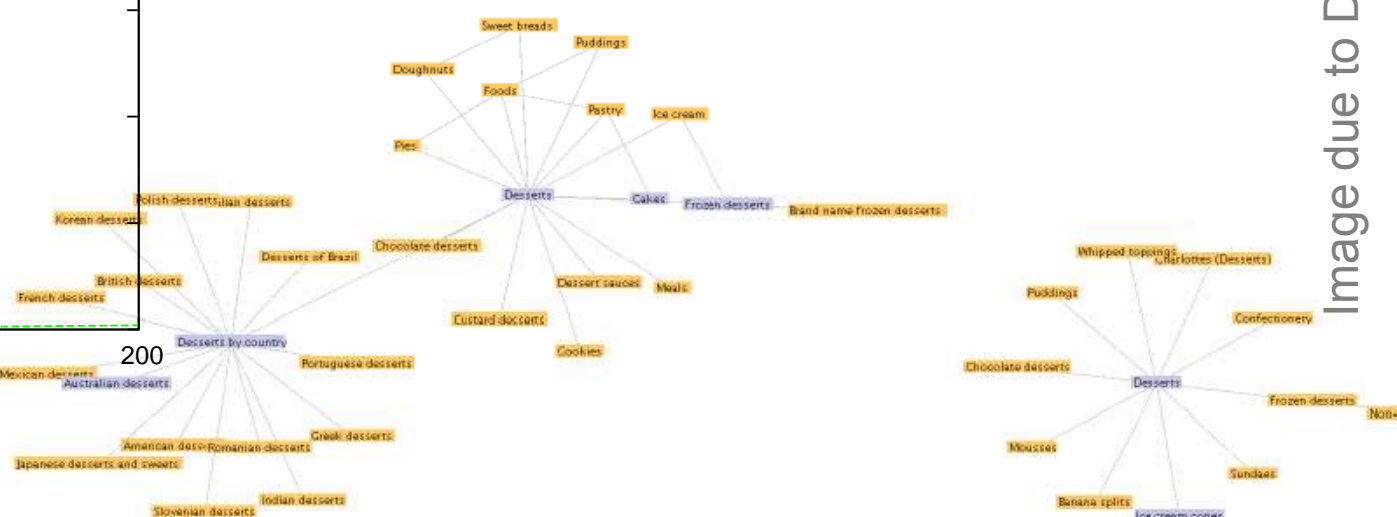
- Gaussian Belief Propagation can also be lifted
- Enables lifted page rank, HITS, Kalman filters, ...

## Lifted Matching

- NetAlignBP, Min-Sum, ... can also be lifted



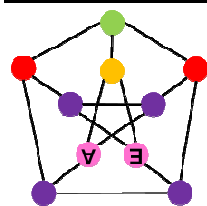
## Library of Congress vs. Wikipedia



WC

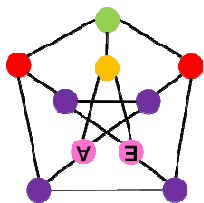
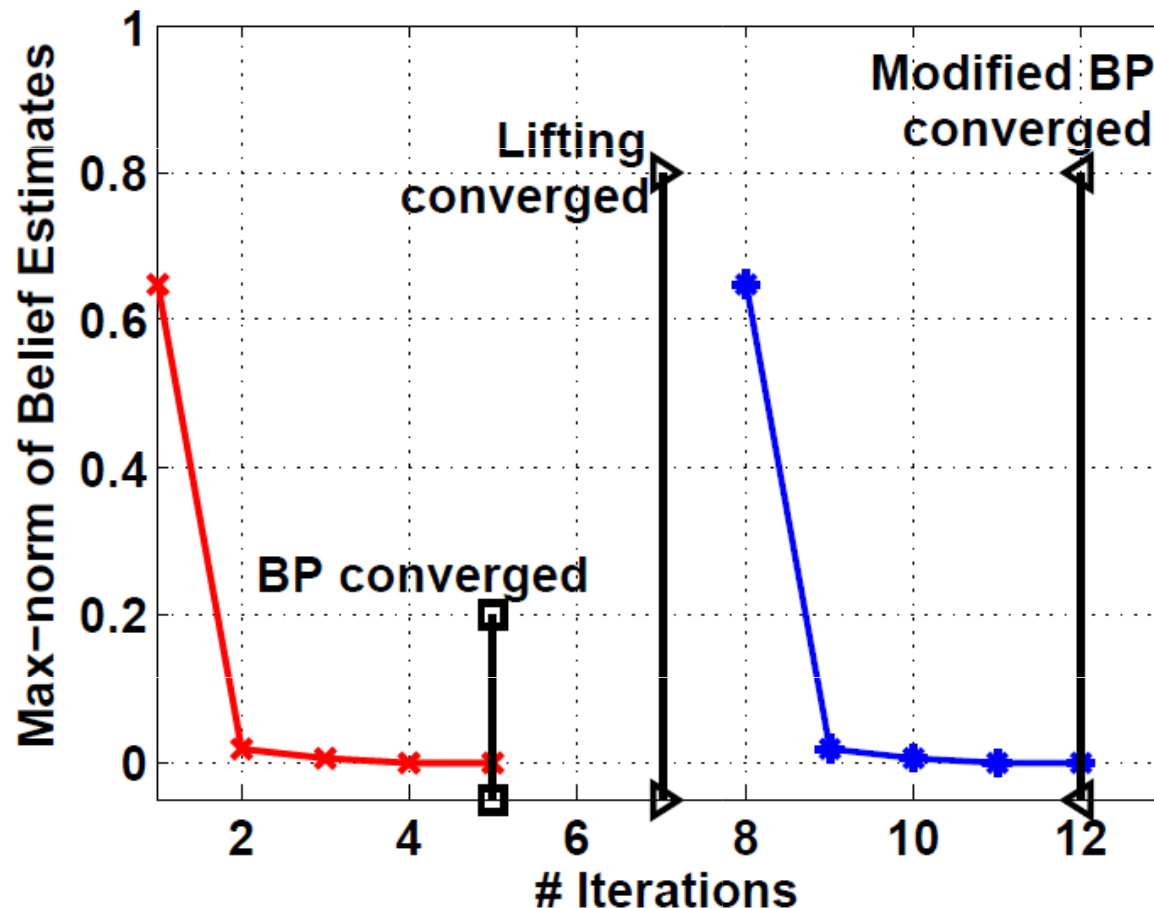
LCSH

**Are they similar?**



K. Kersting  
Statistical Relational Machine Learning  
ANU, Canberra, Australia

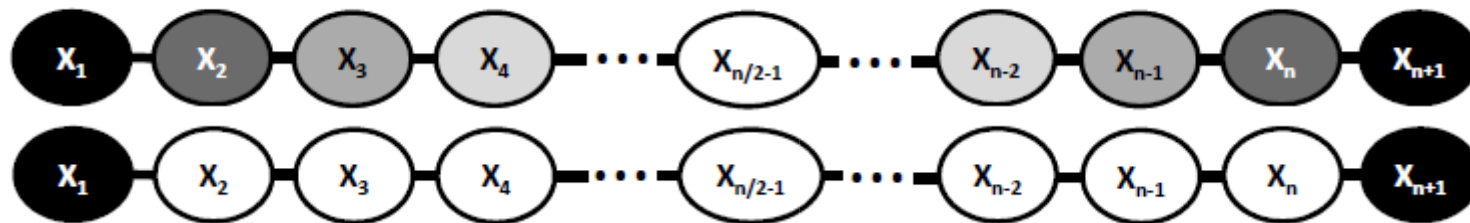
# Content Distribution (Gnutella): Lifted BP vs. BP



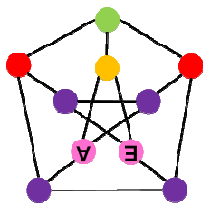


# Message Errors to the Rescue!

- **Ihler et al. 05:** BP message errors decay along paths
- LBP may spuriously assume some nodes send and receive different messages and, hence, produce **pessimistic lifted network**



Make use of decaying message errors already at lifting time





# Informed Lifted Belief Propagation

[El Massaoudi, K, Ahmadi, Hadiji AAAI10]

**Algorithm 1:** iLBP – informed Lifted BP. We use  $b_i(x_i)$  resp.  $m_i(x_i)$  to denote the unnormalized beliefs resp. messages of both variable node  $X_i$  and variable nodes covered by supernodes  $\mathfrak{X}_i$ .

**Data:** A factor graph  $G$  with variable nodes  $X$  and factors  $f$ , Evidence  $E$

**Result:** Unnormalized marginals  $b_i(x_i)$  for all supernodes and, hence, for all variable nodes

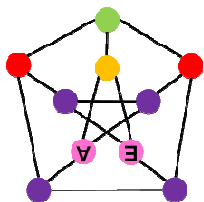
```

1 Colorize  $X$  and  $f$  w.r.t.  $E$ ;
2  $\mathfrak{G} \leftarrow$  one iteration CP;
3 Initialize messages for  $\mathfrak{G}$ ;
4  $(b_i(x_i), m_i(x_i)) \leftarrow$  one iteration MBP on  $\mathfrak{G}$ ;
5 Colorize all  $X_i$ s according to  $m_i(x_i)$ ;
6 while  $b_i(x_i)$ s have not converged do
7    $\mathfrak{G}' \leftarrow$  one iteration CP (based on new colors);
8   Initialize novel supernodes using  $b_i(x_i)$  and  $m_i(x_i)$ ;
9    $(b_i(x_i), m(x_i)) \leftarrow$  one iteration of MBP on  $\mathfrak{G}'$ ;
10  foreach supernode  $\mathfrak{X}$  in  $\mathfrak{G}$  do
11    if the  $m(x_i)$ s of the  $X_i$ s in  $\mathfrak{X}$  differ then
12      | Colorize all  $X_i$  in  $\mathfrak{X}$  according to  $m(x_i)$ 
13    |
14  |
15 Return  $b_i(x_i)$  for all supernodes
  
```

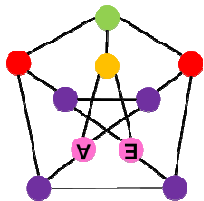
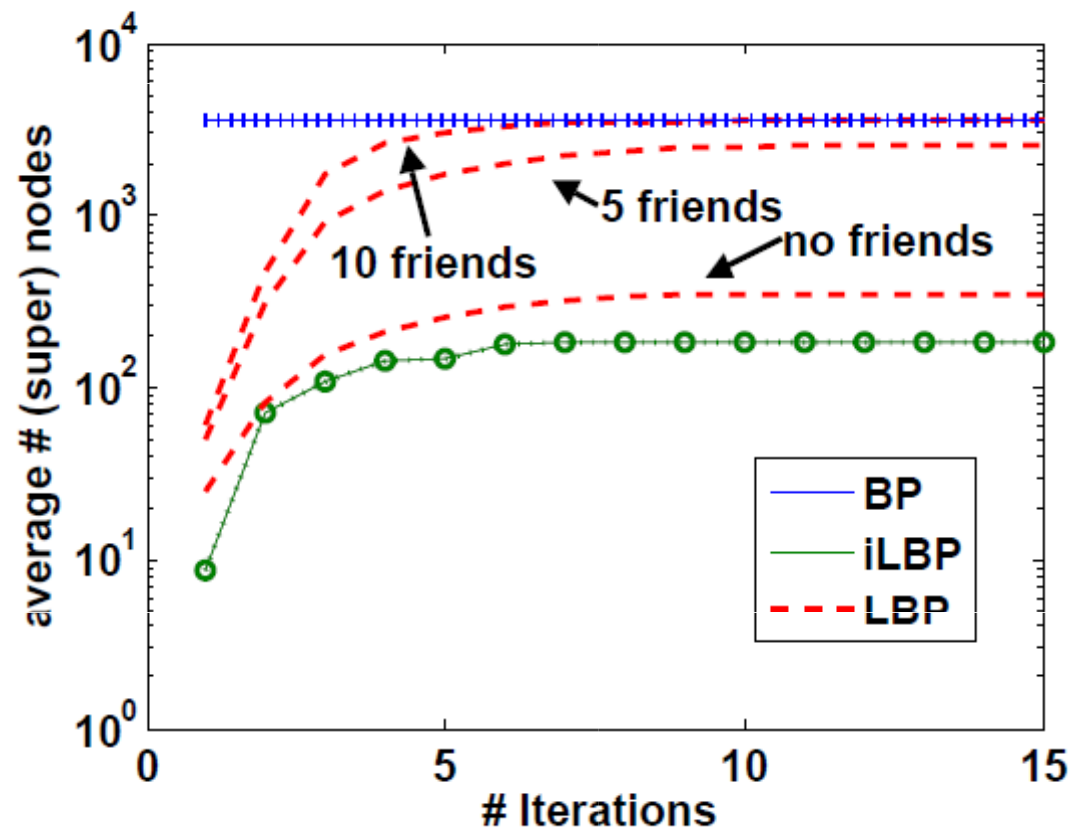
Iterate

Refine Lifting

Modified BP

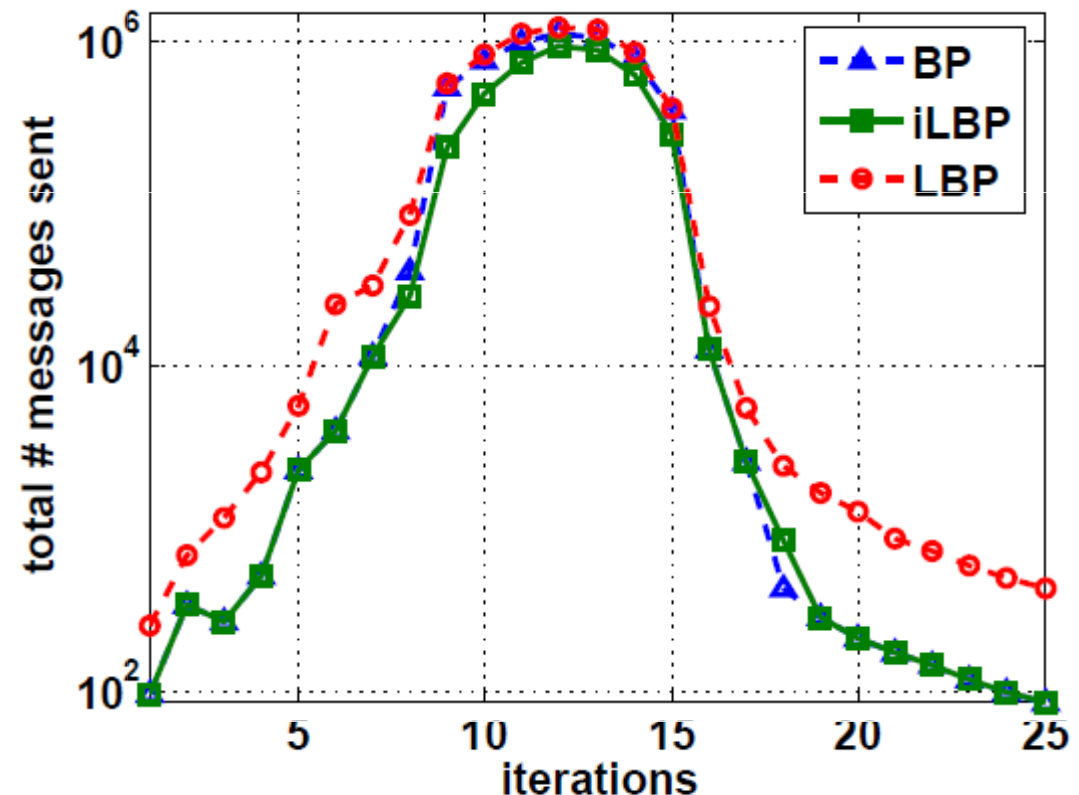


# Social Networks

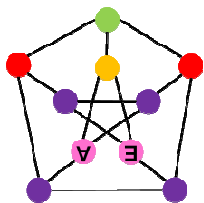


# Lifted Content Distribution

- 1 file, Gnutella snapshot
  - 10876 nodes
  - 39994 edges
- iLBP 4.272.164 mess.
- < BP 5.761.952 mess.
- < LBP 6.381.516 mess.

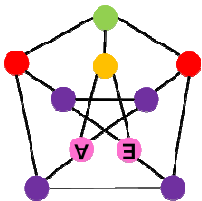


- On a different network:
  - iLBP 1.972.662 < LBP 2.962.311 < BP 5.761.952



# Conclusions

- $\text{StarAI} \geq \text{Objects\&Relations} + \text{Probabilities} + \text{Machine Learning}$
- It covers the whole AI spectrum
  - Lifted SAT, Relational Topic Models, Relational (PO)MDPs, ...
- Lifted/efficient reasoning crucial to StarAI
  - Exploit symmetries revealed by (relational) model
- **More tasks and applications!** NLP, Computer Vision, Robotics?
- Relational linear/quadratic programs?
- Relational combinatorial problems and solvers?
- Lifted junction trees? The Cortex of AI?
- Lifted inference for arbitrary distributions?



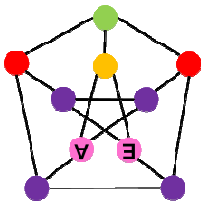
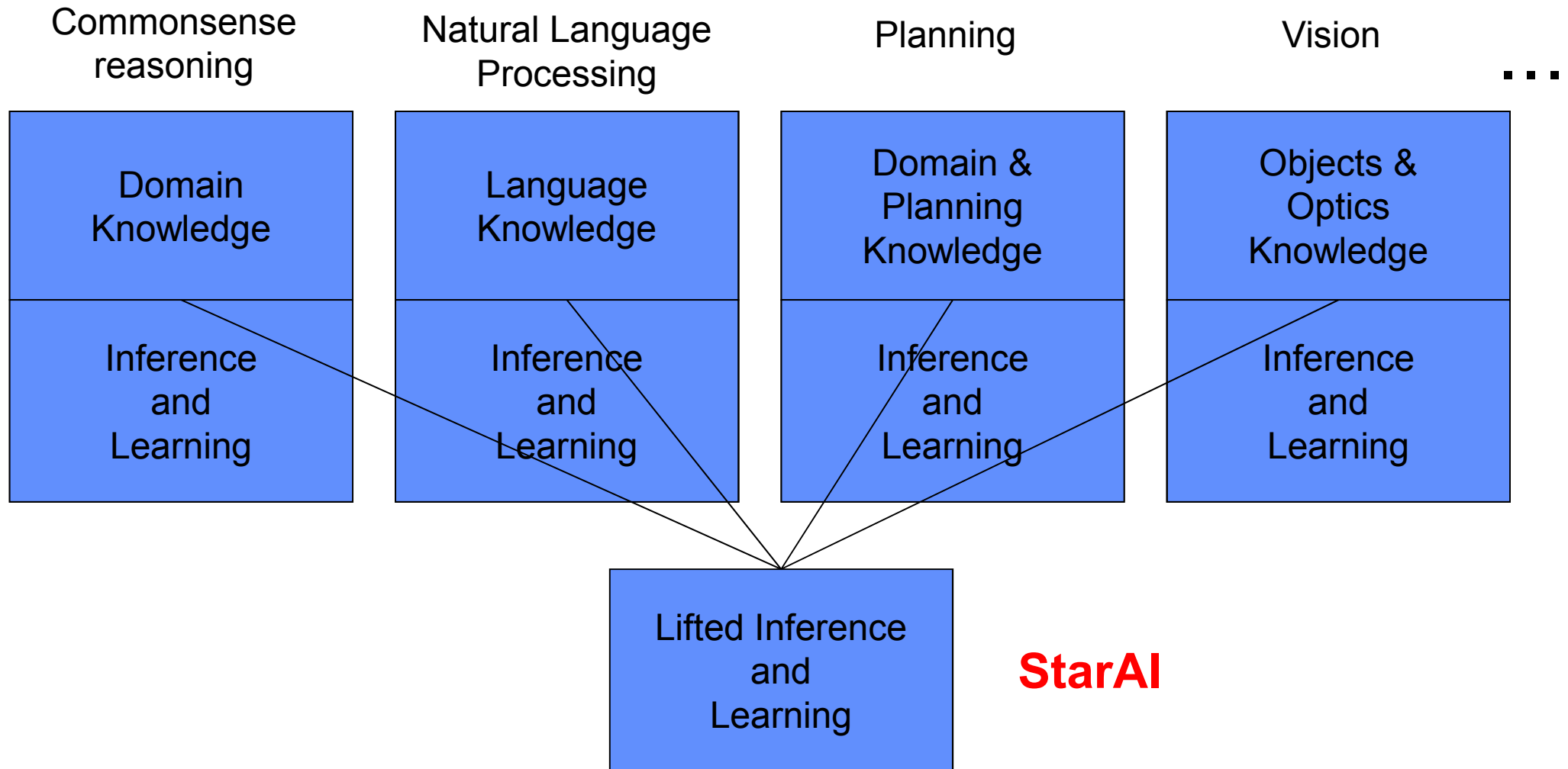
K. Kersting  
Statistical Relational Learning  
Machine Learning Summer School (MLSS)  
ANU, Canberra, Australia, Oct. 4, 2010



**Fraunhofer**



# The Big Picture



**Thanks for your attention**