Outline

- Why probabilistic logic?
- Abduction as an underlying framework
- Abduction by logic programming
- Relationship between graphical models and probabilistic logic
- Research topics

Why Relational + Probabilistic

- Structure, but also uncertainty
- Medical diagnosis, pedigrees, etc.
- Social network structures, citation analysis, etc.


Representations

- We have see generalisations of propositional logic:

$$\mathit{Talented} \wedge \mathit{GoodTeacher} \rightarrow \mathit{PassCourse}$$

Extensions in different directions:

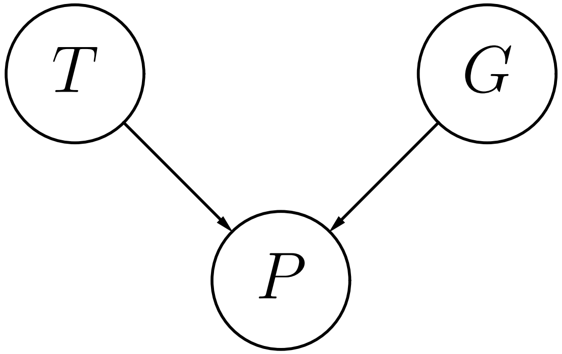
1. How to incorporate uncertainty?
   - Rule-based uncertainty (1970's and 1980's)

$$\mathit{Talented} \wedge \mathit{GoodTeacher} \xrightarrow{\text{CF}=0.9} \mathit{PassCourse}$$

   - Bayesian networks (1990's – now)

2. How to incorporate relations?
   - First-order logic as a general language

Bayesian networks



$$P(p \mid t, g) = 0.9$$
$$P(p \mid \neg t, g) = 0.5$$
$$P(p \mid t, \neg g) = 0.7$$
$$P(p \mid \neg t, \neg g) = 0.1$$
$$P(g) = 0.6$$
$$P(t) = 0.9$$

$$P(P, S, T) = P(P \mid S, T)P(S)P(T)$$

Allows computing arbitrary probabilities:

$$\begin{aligned} P(p \mid t, g) &= 0.9 \\ P(p \mid t) &= \ldots \\ P(t \mid p) &= \ldots \end{aligned}$$

First-order logic

First-order logic allows modelling relations; consider this formula φ:

$$\forall s \, \mathit{Talented}(s) \wedge \mathit{GoodTeacher} \rightarrow \mathit{PassCourse}(s) \qquad (1)$$

Example reasoning (first-order, abductive):

- If John does not pass the course, then (obviously) it is because of the teachers

$$\{\varphi, \forall s \mathit{Talented}(s), \neg \mathit{PassCourse}(J)\} \models \neg \mathit{GoodTeacher}$$

- This might be bad news for Mary, because now there is no hypothesis H such that:

$$\{\varphi, \forall s \mathit{Talented}(s), \neg \mathit{PassCourse}(J), H\} \models \mathit{PassCourse}(M)$$

Probability and First-Order Domains

- FOL can talk about constants and relations
- Propositional/BN only about fixed settings
- So, FOL really opens up new possibilities for representing, reasoning and learning, but...
- We have seen before (with CF) that naive combinations of logic and probability need to be approached with care
- A key general issue: what does it mean to say:
  - P(Some randomly chosen bird can fly) ≥ 0.8
  - P(Tweety can fly) ≥ 0.8 (Tweety is a particular bird)
- Halpern (1990): type-1 and type-2 probability

Probabilistic relational reasoning

- First-order logic: good for relational reasoning in various ways about classes of objects
- Probabilistic graphical models such as Bayesian networks are good for reasoning with uncertainty

$\Rightarrow$ *Is there no way to combine them?*

Solutions for:

- Probability that all students are talented
- Probability that Mary will pass the course, given the observations about John

Many combinations

- Hot topic in AI, many approaches since end of the nineties
- Start from logic programming: KBMC, SLP, PRISM, LPAD, ICL, CPLogic, SCFG, etc.
- Start from probabilistic graphical models: BLP, BLN, SRM, RMM, MLN, RDN, etc.

Probabilistic programming languages starting to appear

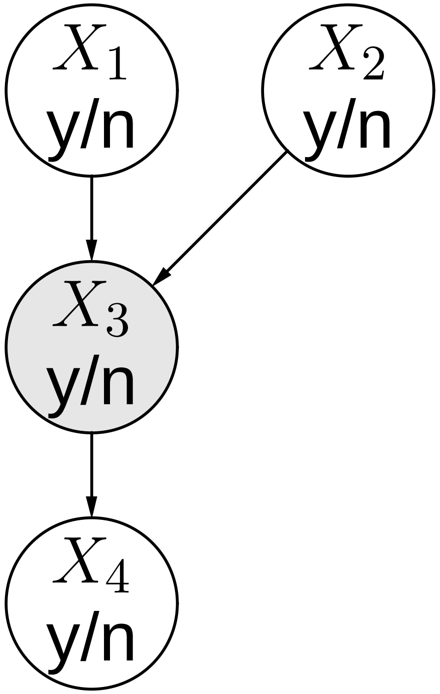
- Imperative-Functional: infer.net, Church, Factorie, Scala, etc.
- Horn-Logic and Prolog: PRISM, ProbLog, ICL, Dyna, etc.

Lessons learned

Consider:

- if $P(a) = 0.3$ and $P(b) = 0.6$, what is $P(a \wedge b)$?
- if $P(a) = 0.3$ and $P(b) = 0.6$, what is $P(a \vee b)$?
- if $P(h \mid e) = 0.3$ and $P(h \mid e') = 0.3$, what is $P(h \mid e, e')$?

Probabilistic reasoning

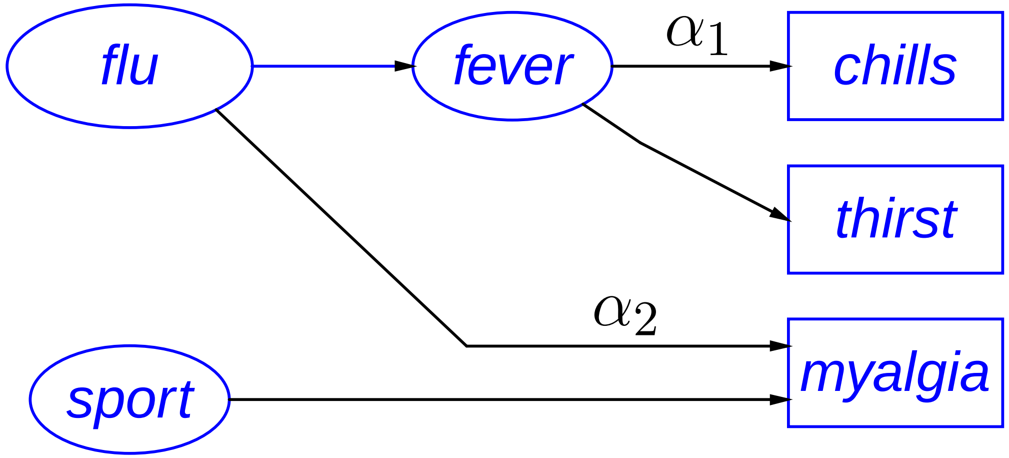


$$P(x_4 \mid x_3) = 0.4$$
$$P(x_4 \mid \neg x_3) = 0.1$$
$$P(x_3 \mid x_1, x_2) = 0.3$$
$$P(x_3 \mid \neg x_1, x_2) = 0.5$$
$$P(x_3 \mid x_1, \neg x_2) = 0.7$$
$$P(x_3 \mid \neg x_1, \neg x_2) = 0.9$$
$$P(x_1) = 0.6$$
$$P(x_2) = 0.2$$

$$P(x_3) = \sum_{X_1, X_2} P(x_3 \mid X_1, X_2) P(X_1) P(X_2)$$

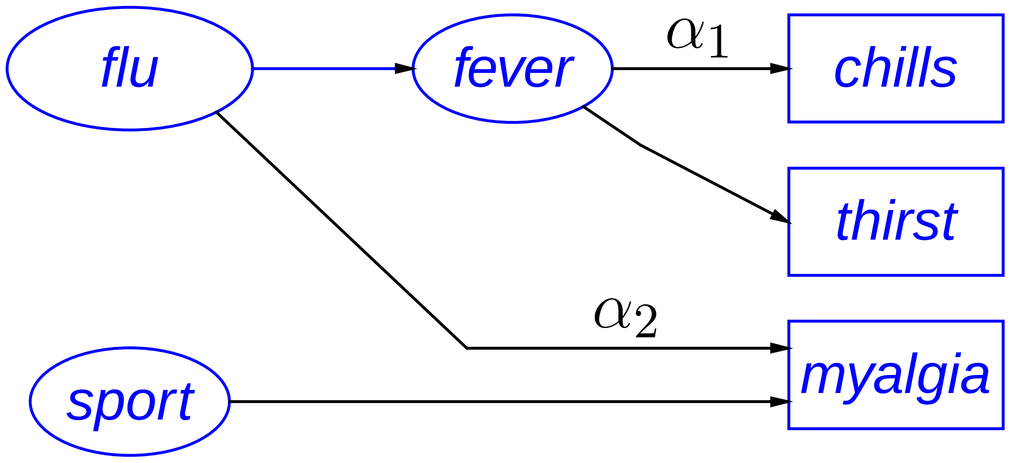
probabilistic reasoning = abduction?

Recall: abductive explanations



- Causal specification: $\Sigma = (\Delta, \Phi, \mathcal{R})$, met:
  - Δ: potential causes and incompleteness assumptions (assumables)
  - Φ: facts that can be observed
  - $\mathcal{R}$: causal model
- Explanations (prediction) $E \subseteq \Delta$: $\boxed{\mathcal{R} \cup E \vDash F}$
- Let $\mathcal{E}(F)$ be the set of all explanations of F

Example explanations



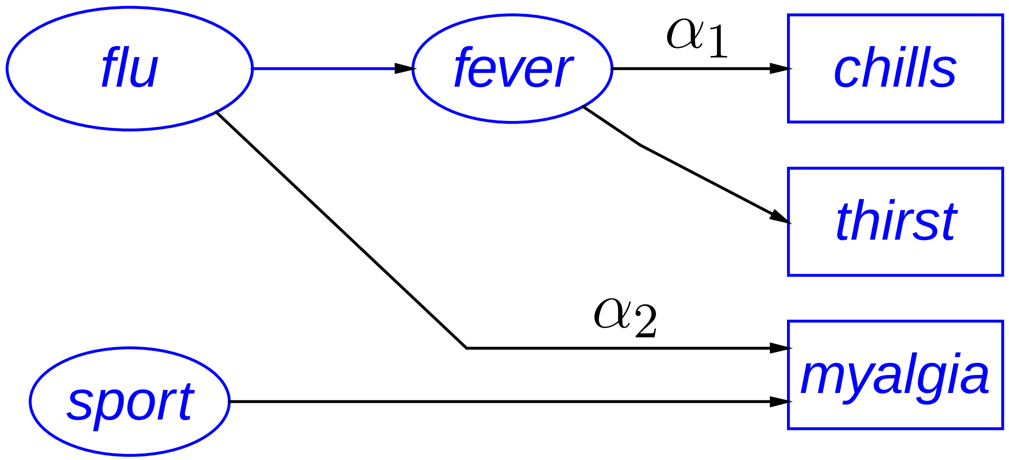
Causal specification: $\Sigma = (\Delta, \Phi, \mathcal{R})$

- Example 1: $\mathcal{R} \cup \{\textit{flu}, \alpha_1\} \models \textit{chills} \wedge \textit{thirst}$
- Example 2: $\mathcal{R} \cup \{\textit{flu}, \alpha_1, \alpha_2\} \models \textit{chills} \wedge \textit{thirst}$

The set of all explanations for chills and thirst contains:

$$\mathcal{E}(\textit{chills} \wedge \textit{thirst}) = \{\{\textit{flu}, \alpha_1\}, \{\textit{flu}, \alpha_1, \alpha_2\}, \{\textit{flu}, \alpha_1, \textit{sport}\}, \{\textit{flu}, \alpha_1, \textit{sport}, \alpha_2\}\}$$

Closed world assumption

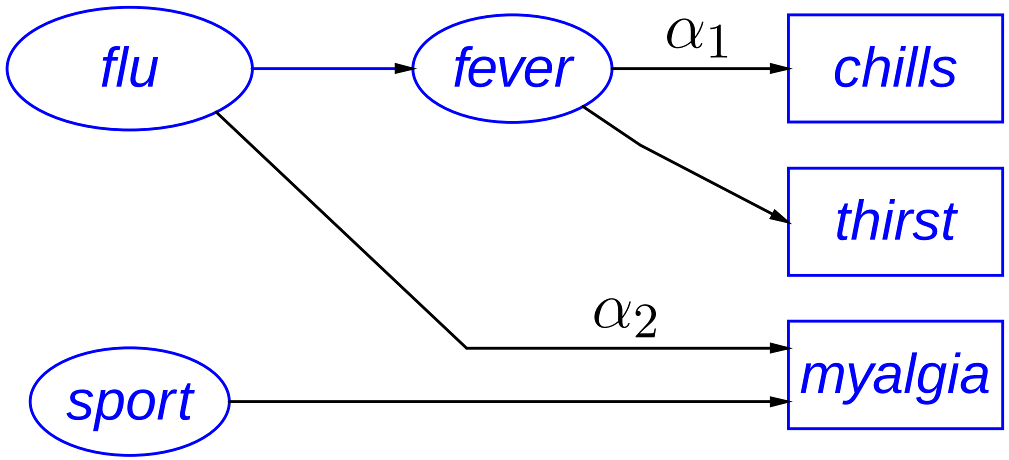


Closed world assumption: F is only true if and only if one of its explanations is true:

$$F = \bigvee_{E_i \in \mathcal{E}(F)} E_i$$

e.g: $\textit{chills} \wedge \textit{thirst} = (\textit{flu} \wedge \alpha_1) \vee (\textit{flu} \wedge \alpha_1 \wedge \alpha_2)$
$\vee(\textit{flu} \wedge \alpha_1 \wedge \textit{sport}) \vee (\textit{flu} \wedge \alpha_1 \wedge \textit{sport} \wedge \alpha_2)$

Idea for adding probabilities



Suppose we have a probability distribution over Δ, i.e., $P(\Delta)$, then we can compute $P(F)$, because:

$$\boxed{P(F) = P(\bigvee_{E_i \in \mathcal{E}(F)} E_i)}$$

$$\begin{aligned} P(\textit{chills} \wedge \textit{thirst}) &= P((\textit{flu} \wedge \alpha_1) \vee (\textit{flu} \wedge \alpha_1 \wedge \alpha_2) \\ &\quad \vee (\textit{flu} \wedge \alpha_1 \wedge \textit{sport}) \vee (\textit{flu} \wedge \alpha_1 \wedge \textit{sport} \wedge \alpha_2)) \end{aligned}$$

Sufficiency of minimal explanations

Definition. A minimal explanation E for F is an explanation E for F s.t. there is no $E' \subset E$ where E' is an explanation for F.

Theorem. Let $\mathcal{E}_m(F)$ be the set of all minimal explanations for F. Then:

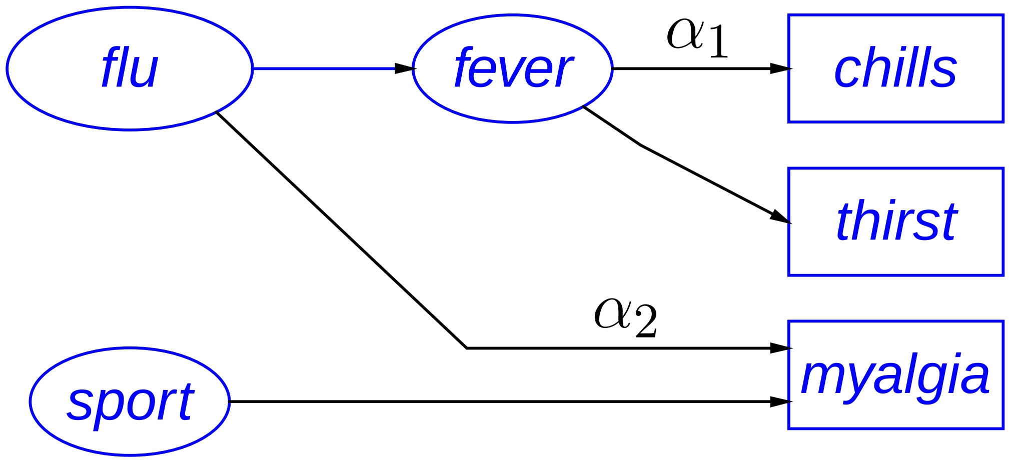
$$F = \bigvee_{E_i \in \mathcal{E}_m(F)} E_i$$

Proof (sketch). Note that if $E_i \in \mathcal{E}_m(F)$ and $E_j \supset E_i$, then:

$$E_i \vee E_j = E_i$$

Proof by induction on the number of non-minimal explanations

Minimal explanations: example



$$\begin{aligned} \textit{chills} \wedge \textit{thirst} &= (\textit{flu} \wedge \alpha_1) \vee (\textit{flu} \wedge \alpha_1 \wedge \textit{sport}) \vee \cdots \\ &= \textit{flu} \wedge \alpha_1 \end{aligned}$$

Recall that this is the solution formula S for F: the most specific formula consisting only of abducible literals, such that

$$\text{COMP}[\mathcal{R}; N] \cup F \models S$$

Defining a probability distribution

- We assume a very simple distribution consisting of a set of independent random variables
- Partition $V \subseteq \Delta$ is associated to a random variable X_V where V is the domain of X

Example:

$P(X = \textit{sport}) = 0.3$

$P(X = \textit{flu}) = 0.1$

$P(X = \textit{not\_sport\_or\_flu}) = 0.6$

$P(Y = \alpha_1) = 0.9$

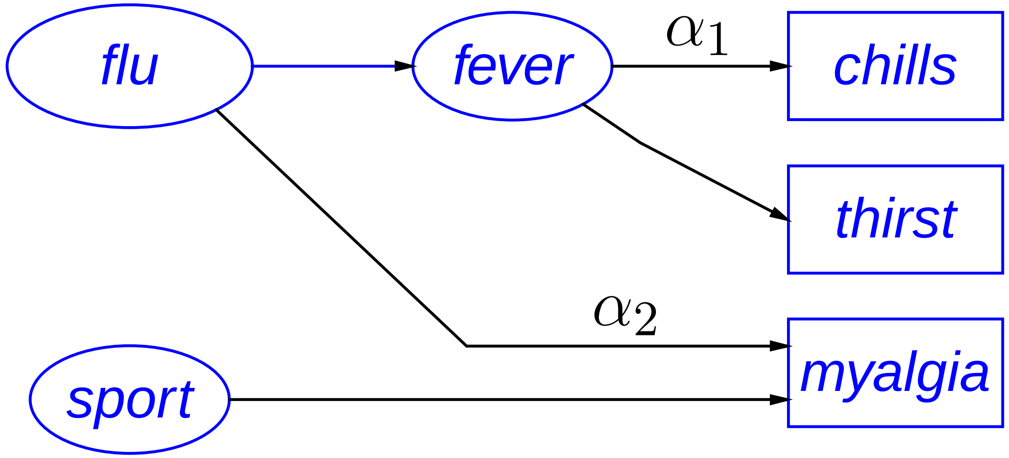
$P(Y = \textit{other}_1) = 0.1$

$P(Z = \alpha_2) = 0.7$

$P(Z = \textit{other}_2) = 0.3$

(Assumption: sport and flu are mutually exclusive)

Example



$$
\begin{aligned}
P(\textit{myalgia}) &= P((\textit{flu} \wedge \alpha_2) \vee \textit{sport}) \\
&= P(\textit{flu} \wedge \alpha_2) + P(\textit{sport}) \\
&= P(\textit{flu})P(\alpha_2) + P(\textit{sport}) \\
&= 0.1 \cdot 0.7 + 0.3 = 0.37
\end{aligned}
$$

Problem: how to obtain the minimal explanations?

Recall: logic programming

- A substitution θ is a finite set of the form $\theta = \{t_1/x_1, \ldots, t_n/x_n\}$, with x_i a variabele and t_i a term; $x_i \neq t_i$ and $x_i \neq x_j$, $i \neq j$
- A grounded expression does not contain variables
- A substitution θ is called a unifier of E and E' if $E\theta = E'\theta$; E and E' are then called unifiable
- SLD resolution (for Horn clauses):

$$\frac{\leftarrow (B_1, \ldots, B_n)\theta, \qquad (B_i \leftarrow A_1, \ldots, A_m)\theta}{\leftarrow (B_1, \ldots, B_{i-1}, A_1, \ldots, A_m, B_{i+1}, \ldots, B_n)\theta}$$

such that B_i unifies given substitution θ

SLD derivation = backward reasoning + unification

Explanations: by resolution

Given the following specification:

chills $\leftarrow$ *fever*	*sport* $: 0.3$, *flu* $: 0.1$
thirst $\leftarrow$ *fever*	$\alpha_1 : 0.9$
fever $\leftarrow$ *flu*	$\alpha_2 : 0.7$
myalgia $\leftarrow$ *flu*	
myalgia $\leftarrow$ *sport*	

Suppose $F =$ *myalgia*:

$$\cfrac{\cfrac{\leftarrow \textit{myalgia} \quad \textit{myalgia} \leftarrow \textit{sport}}{\leftarrow \textit{sport}} \qquad \textit{sport} : 0.3}{\Box}$$

Explanations: by resolution

Given the following specification:

chills $\leftarrow$ *fever*	*sport* $: 0.3,$ *flu* $: 0.1$
thirst $\leftarrow$ *fever*	$\alpha_1 : 0.9$
fever $\leftarrow$ *flu*	$\alpha_2 : 0.7$
myalgia $\leftarrow$ *flu*	
myalgia $\leftarrow$ *sport*	

Suppose $F =$ *myalgia*:

$$\cfrac{\cfrac{\cfrac{\leftarrow \textit{myalgia} \quad \textit{myalgia} \leftarrow \textit{flu}, \alpha_2}{\leftarrow \textit{flu}, \alpha_2 \qquad \textit{flu} : 0.1}}{\leftarrow \alpha_2 \qquad \alpha_2 : 0.7}}{\Box}$$

The AILog system

```
prob flu : 0.1, sport : 0.3, dummy : 0.6.
prob a1 : 0.9.
prob a2 : 0.7.

chills  <- fever & a1.
fever   <- flu.
thirst  <- fever.
myalgia <- flu & a2.
myalgia <- sport.
```

gives:

```
ailog: predict myalgia.
Answer: P(myalgia|Obs)=0.37.
  [ok,more,explanations,worlds,help]: explanations.
  0: ass([],[a2,flu],0.06999999999999999)
  1: ass([],[sport],0.3)
```

The first-order case

- Explanations are sets of ground assumables
- In particular: ground assumables used in an SLD proof
- A declaration:

$$a_1 : p, a_2 : p_2, \ldots, a_n : p_n$$

now defines a random variable X_i for every grounding of $a_1, \ldots, a_n$ such that $P(X_i = a_j\theta_i) = p_j$

Example:

$$\mathit{Flu}(p) : 0.1, \mathit{Sport}(p) : 0.3, \mathit{Other}(p) : 0.6$$

implies e.g. $\mathit{Flu}(\mathit{Arjen}) = 0.1$

First-order inference: example

Given is:

$$\textit{PassCourse}(s) \leftarrow \alpha(s), \textit{GoodTeacher}$$

$$\textit{GoodTeacher} : 0.7$$

$$\alpha(s) : 0.9$$

What is $P(\textit{PassCourse}(M))$?

$$\cfrac{\cfrac{\cfrac{\leftarrow \textit{PassCourse}(M) \quad \textit{PassCourse}(s) \leftarrow \alpha(s), \textit{GoodTeacher}}{\leftarrow \alpha(M), \textit{GoodTeacher}} \quad \alpha(s) : 0.9}{\leftarrow \textit{GoodTeacher}} \quad \textit{GoodTeacher} : 0.7}{\square}$$

$$P(\textit{PassCourse}(M)) = P(\alpha(M) \wedge \textit{GoodTeacher}) = 0.63$$

Conditional probabilities

By the definition of conditional probability:

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)} = \frac{P(\bigvee_{E_i \in \mathcal{E}_m(A \wedge B)} E_i)}{P(\bigvee_{E_i \in \mathcal{E}_m(B)} E_i)}$$

Example:

$$\mathcal{E}_m(\mathit{PassCourse}(J)) = \{\{\mathit{GoodTeacher}, \alpha(J)\}\}$$

$$\mathcal{E}_m(\mathit{PassCourse}(M) \wedge \mathit{PassCourse}(J))$$

$$= \{\{\mathit{GoodTeacher}, \alpha(J), \alpha(M)\}\}$$

$$\begin{aligned} P(\mathit{PC}(M) \mid \mathit{PC}(J)) &= \frac{P(\mathit{GoodTeacher}) \cdot P(\alpha(J)) \cdot P(\alpha(M))}{P(\mathit{GoodTeacher}) \cdot P(\alpha(J))} \\ &= P(\alpha(M)) = 0.9 \end{aligned}$$

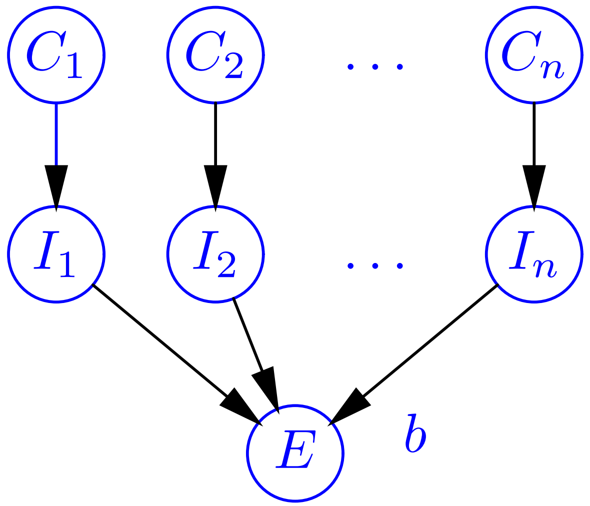
Expressiveness

- Every Bayesian network can be translated to probabilistic logic (Assignment 2)
  - Intuition: each variable given its parent in the graph becomes an implication
- Every ground probabilistic program can be converted into a Bayesian network
  - What happens to multiple rules with the same head?
- Every non-ground probabilistic program can be seen as a template for a Bayesian network:

$$A(x) \leftarrow \alpha(x), B(x)$$

can be seen as a piece of Bayesian network for every instantiation for x

Recall: causal independence

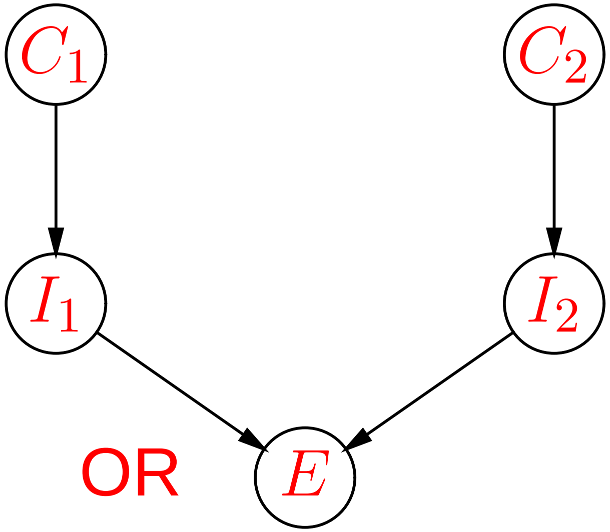


$$P(e \mid C_1, \ldots, C_n) = \sum_{I_1,\ldots,I_n} P(e \mid I_1, \ldots, I_n) \prod_{k=1}^{n} P(I_k \mid C_k)$$

$$= \sum_{b(I_1,\ldots,I_n)=e} \prod_{k=1}^{n} P(I_k \mid C_k)$$

Boolean functions: $P(E \mid I_1, \ldots, I_n) \in \{0, 1\}$ with $b(I_1, \ldots, I_n) = 1$ if $P(e \mid I_1, \ldots, I_n) = 1$

Noisy-OR model



Example: suppose we have an OR model and two (true) causes, it follows:

$$\begin{aligned} P(e \mid c_1, c_2) &= P(i_1 \mid c_1)P(\neg i_2 \mid c_2) + P(i_1 \mid c_1)P(\neg i_2 \mid c_2) \\ &+ P(i_1 \mid c_1)P(i_2 \mid c_2) \end{aligned}$$

Compare: $\mathcal{R} = \{E(x) \leftarrow \alpha(x,k,y), C(k,y)\}$ with $\alpha(x,k,y)$ and $C(k,y)$ assumables

Noisy-OR model (2)

$$P(E(t) \mid C_1(t), C_2(t))$$

$$= \frac{P(\mathcal{E}(E(t), C_1(t), C_2(t)))}{P(\mathcal{E}(C_1(t), C_2(t)))}$$

$$= \frac{P((\alpha(t,1,t) \wedge C_1(t) \wedge C_2(t)) \vee (\alpha(t,2,t) \wedge C_1(t) \wedge C_2(t)))}{P(C_1(t) \wedge C_2(t))}$$

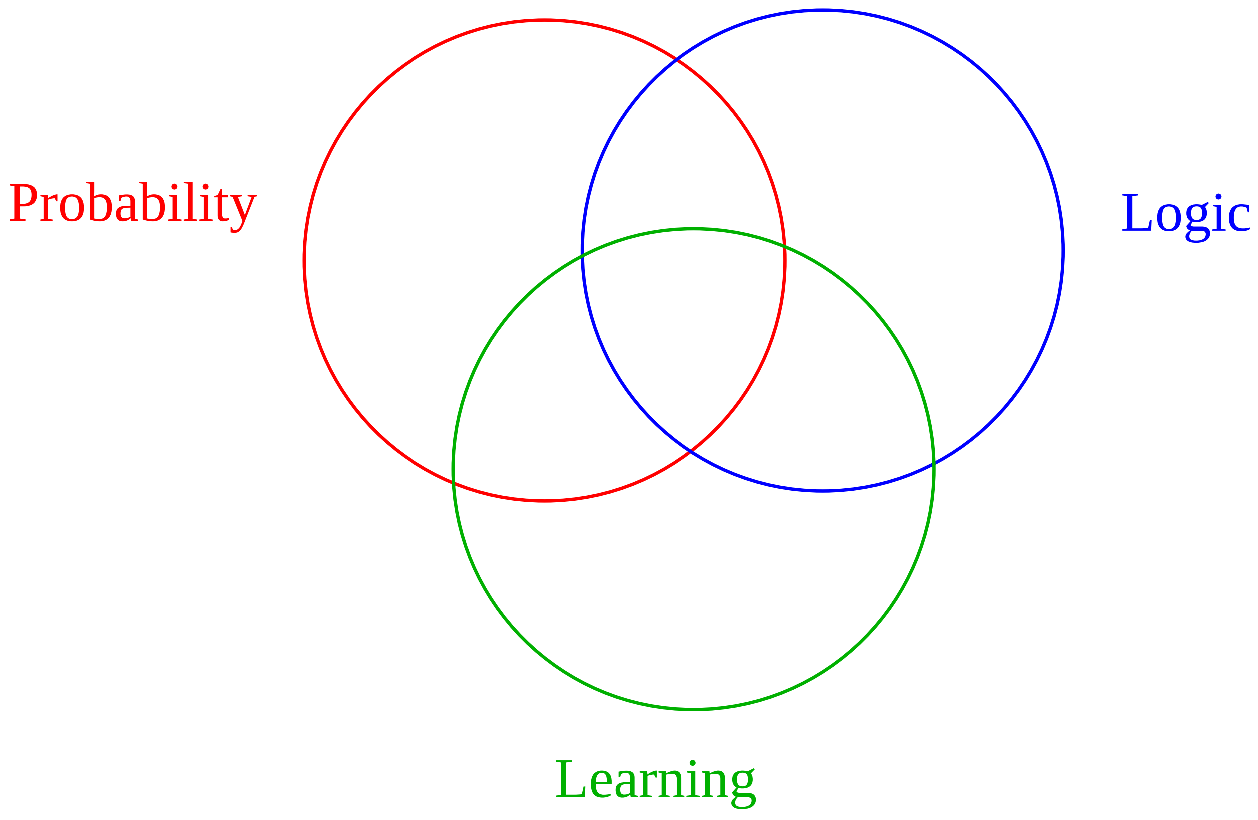
$$= \frac{P(C_1(t) \wedge C_2(t) \wedge (\alpha(t,1,t) \vee \alpha(t,2,t)))}{P(C_1(t) \wedge C_2(t))}$$

$$= P(\alpha(t, 1, t) \vee \alpha(t, 2, t))$$

$$= P(\alpha(t, 1, t) \wedge \alpha(t, 2, t)) + P(\alpha(t, 1, t) \wedge \alpha(f, 2, t)) + P(\alpha(f, 1, t) \wedge \alpha(t, 2, t))$$

$$= P(\alpha(t, 1, t))P(\alpha(t, 2, t)) + P(\alpha(t, 1, t))P(\alpha(f, 2, t)) + P(\alpha(f, 1, t))P(\alpha(t, 2, t))$$

Goal: Probabilistic Logic Learning



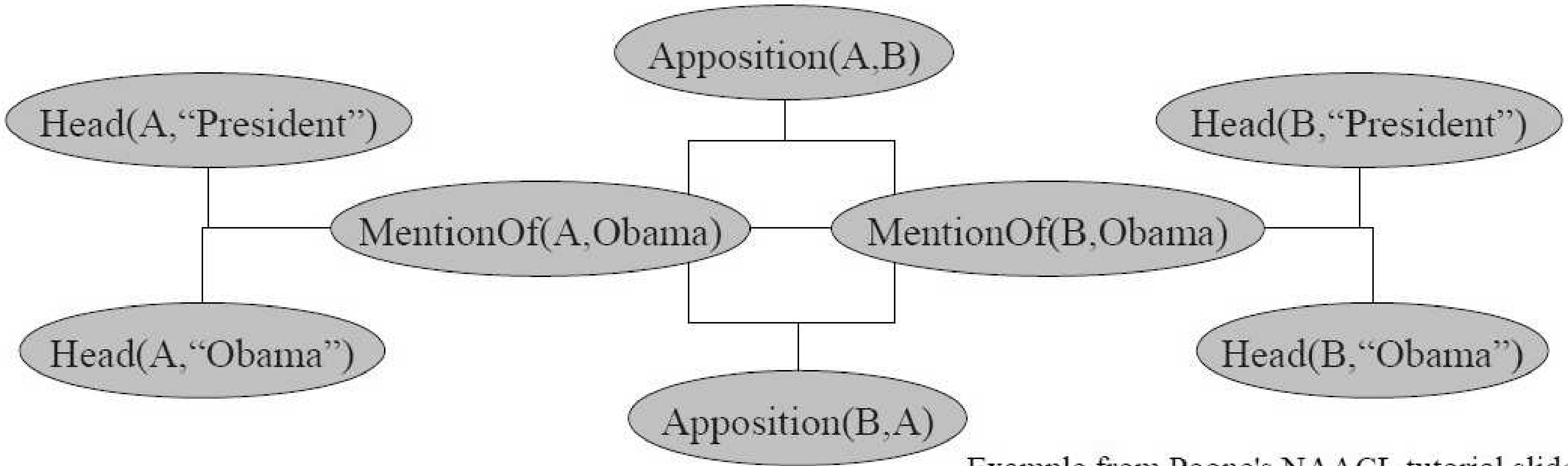
Example: Markov Logic Networks

- Specify undirected models (Markov networks) with the use of first-order logic
- An interface layer for AI (Domingos)
- Syntax: weighted first-order formulas
- Semantics: feature templates for Markov networks
- Intuition: soften logical constraints
  - Each formula has a weight
  - Higher weights mean stronger constraints
  - P(world)   exp(sum weights of formulas it satisfies)

Example: Markov Logic Networks(2)

1.5	$\forall x \; \text{MentionOf}(x, \text{Obama}) \Rightarrow \text{Head}(x, \text{"Obama"})$
0.8	$\forall x \; \text{MentionOf}(x, \text{Obama}) \Rightarrow \text{Head}(x, \text{"President"})$
100	$\forall x, y, c \; \text{Apposition}(x, y) \wedge \text{MentionOf}(x, c) \Rightarrow \text{MentionOf}(y, c)$

Two mention constants: **A** and **B**



Example from Poone's NAACL tutorial slides

Example: ProbLog

- Probabilistic Prolog (Problog) at KU Leuven
- Probabilistic programming language (extends YAP Prolog, as opposed to meta-interpreter such as AILog)
- Supports probabilistic inference/learning in Prolog
- Fast and Free at http://dtai.cs.kuleuven.be/problog/
- Standard clauses: $\texttt{path}(\texttt{X},\texttt{Y},\texttt{A}) :- \texttt{X} \neq \texttt{Y}, \texttt{edge}(\texttt{X},\texttt{Y})$, etc.
- Probabilistic facts: $0.9 :: \texttt{edge}(\texttt{y}, 1, 2)$
- Syntactic sugar:
  $0.3 :: \texttt{fall}(\texttt{X}) \vee 0.7 :: \texttt{stay}(\texttt{X}) \leftarrow \texttt{manipulate}(\texttt{X})$
- Basically extends standard Prolog queries like $?-\texttt{q}(\texttt{X}).$ and $?-\texttt{mother}(\texttt{a},\texttt{Y})$ to $?-\texttt{prob}(\texttt{q}(\texttt{X})).$ and $?-\texttt{prob}(\texttt{mother}(\texttt{a},\texttt{Y}))$

Research topics

- Efficient inference: given the explanations, it is not so easy to compute the probability, e.g., consider this:

$$P(a \vee b \vee c) = P(a) + P(a \wedge \neg b) + P(a \wedge \neg b \wedge \neg c)$$

- Semantical questions such as dealing with hard constraints. Recall that in abduction we have `nogoods`,

  ```
  false <- chills
  ```

  What does this mean for the probability distribution? And would it possible to add soft constraints?
- Learning clauses and parameters from data
- Application oriented: many application fields are both relational as well as probabilistic