Logic, Probability and Computation: Statistical Relational AI and Beyond

David Poole

Department of Computer Science,
University of British Columbia

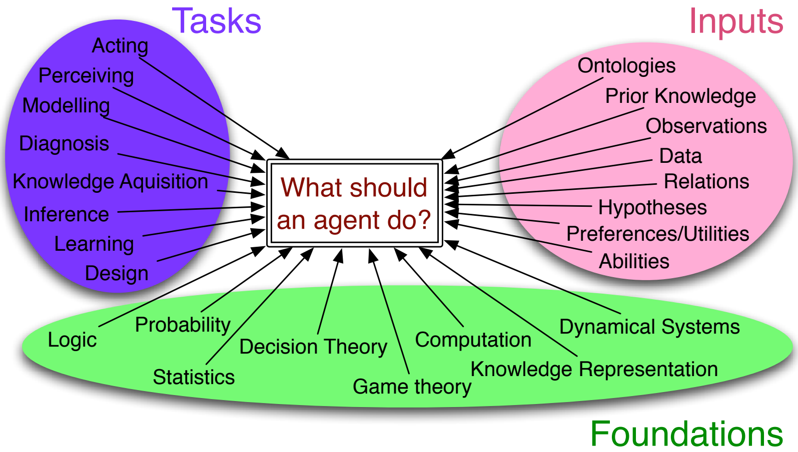
March 20, 2018

There is a real world with real structure. The program of mind has been trained on vast interaction with this world and so contains code that reflects the structure of the world and knows how to exploit it. This code contains representations of real objects in the world and represents the interactions of real objects. ...

You exploit the structure of the world to make decisions and take actions. Where you draw the line on categories, what constitutes a single object or a single class of objects for you, is determined by the program of your mind, which does the classification. This classification is not random but reflects a compact description of the world, and in particular a description useful for exploiting the structure of the world.

Eric Baum, *What is Thought?*, 2004, pages 169-170

AI: computational agents that act intelligently



Outline

1. Logic and Probability
   - Relational Probabilistic Models
   - Probabilistic Logic Programs
2. Lifted Inference
   - Lifted Inference
   - Recursive Conditioning
   - Lifted Recursive Conditioning
3. Undirected models, Directed models, and Weighted Formulae
4. Existence and Identity Uncertainty

First-order Predicate Calculus

The world (we want to represent) is made up of individuals (things) with relationships among them.

First-order Predicate Calculus

> *The world (we want to represent) is made up of individuals (things) with relationships among them.*

There isn't anything else!

First-order Predicate Calculus

The world (we want to represent) is made up of individuals (things) with relationships among them.

There isn't anything else!

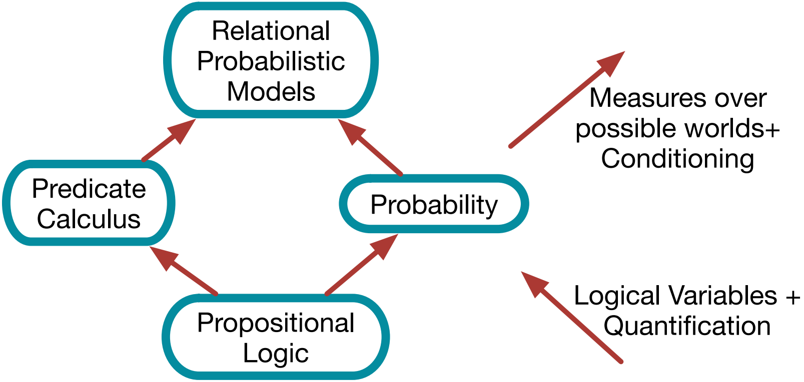
Classical (first order) logic lets us represent:

- individuals in the world
- relations amongst those individuals
- conjunctions, disjunctions, negations of relations
- quantification over individuals

Why Probability?

- There is lots of uncertainty about the world, but agents still need to act.
- Predictions are needed to decide what to do:
  - definitive predictions: you will be run over tomorrow
  - point probabilities: probability you will be run over tomorrow is 0.002 if you are not careful and 0.000001 if you are careful.
  - probability ranges: you will be run over with probability in range [0.001,0.34]
- Acting is gambling: agents who don't use probabilities will lose to those who do — Dutch books.
- Probabilities can be learned from data.
  Bayes' rule specifies how to combine data and prior knowledge.

Statistical Relational AI



Bayes' Rule

Probability provides a calculus for how knowledge (observations) affects belief.

Bayes' Rule

Probability provides a calculus for how knowledge (observations) affects belief.

$$P(h|e) = \frac{\overset{\text{Likelihood}}{P(e|h)}\ \ \overset{\text{Prior}}{P(h)}}{\underset{\text{Normalizing constant}}{P(e)}}$$

Bayes' Rule

Probability provides a calculus for how knowledge (observations) affects belief.

$$P(h|e) = \frac{\overbrace{P(e|h)}^{\text{Likelihood}}\ \overbrace{P(h)}^{\text{Prior}}}{\underbrace{P(e)}_{\text{Normalizing constant}}}$$

- What if e is a patient's electronic health record and h is the effect of a particular treatment on a particular patient?

Bayes' Rule

Probability provides a calculus for how knowledge (observations) affects belief.

Likelihood Prior

$$P(h|e) = \frac{P(e|h)\ P(h)}{P(e)}$$

Normalizing constant

- What if e is a patient's electronic health record and h is the effect of a particular treatment on a particular patient?
- What if e is the electronic health records for all of the people in the province?

Bayes' Rule

Probability provides a calculus for how knowledge (observations) affects belief.

Likelihood Prior

$$P(h|e) = \frac{P(e|h)\ P(h)}{P(e)}$$

Normalizing constant

- What if e is a patient's electronic health record and h is the effect of a particular treatment on a particular patient?
- What if e is the electronic health records for all of the people in the province?
- What if e is a collection of student records in a university?

Bayes' Rule

Probability provides a calculus for how knowledge (observations) affects belief.

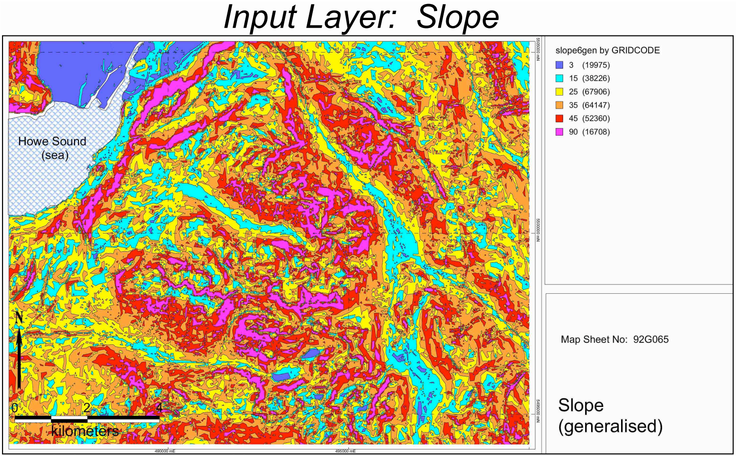
Likelihood Prior

$$P(h|e) = \frac{P(e|h)\ P(h)}{P(e)}$$

Normalizing constant

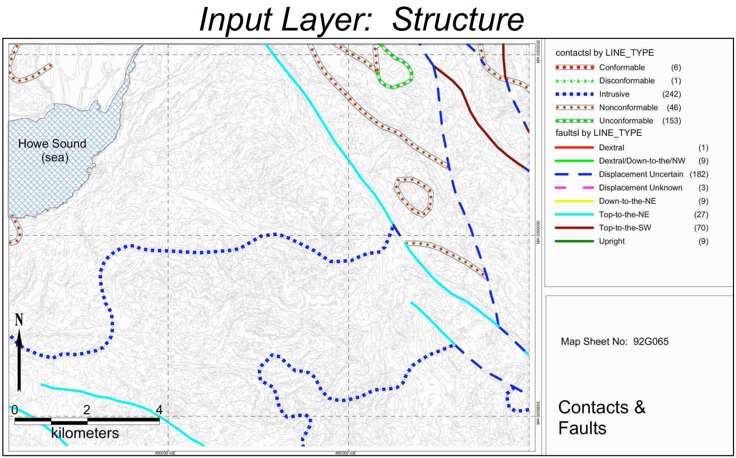
- What if e is a patient's electronic health record and h is the effect of a particular treatment on a particular patient?
- What if e is the electronic health records for all of the people in the province?
- What if e is a collection of student records in a university?
- What if e is everything known about the geology of Earth?

Example Observation, Geology



[Clinton Smyth, Georeference Online.]

Example Observation, Geology



[Clinton Smyth, Georeference Online.]

Outline

1. Logic and Probability
   - Relational Probabilistic Models
   - Probabilistic Logic Programs
2. Lifted Inference
   - Lifted Inference
   - Recursive Conditioning
   - Lifted Recursive Conditioning
3. Undirected models, Directed models, and Weighted Formulae
4. Existence and Identity Uncertainty

Relational Learning

- Machine learning typically assumes informative feature values. But often the values are names of individuals.
- It is the properties of these individuals and their relationship to other individuals that needs to be learned.
- Relational learning has been studied under the umbrella of "Inductive Logic Programming" as the representations were traditionally logic programs.

Example: trading agent

What does Joe like?

Individual	Property	Value
joe	*likes*	*resort_14*
joe	*dislikes*	*resort_35*
...	...	...
resort_14	*type*	*resort*
resort_14	*near*	*beach_18*
beach_18	*type*	*beach*
beach_18	*covered_in*	*ws*
ws	*type*	*sand*
ws	*color*	*white*
...	...	...

Example: trading agent

Possible hypothesis that could be learned:

Example: trading agent

Possible hypothesis that could be learned:
"Joe likes resorts that are near sandy beaches."

Example: trading agent

Possible hypothesis that could be learned:
"Joe likes resorts that are near sandy beaches."

$$
\begin{aligned}
& prop(joe, likes, R) \leftarrow \\
& \qquad prop(R, type, resort) \wedge \\
& \qquad prop(R, near, B) \wedge \\
& \qquad prop(B, type, beach) \wedge \\
& \qquad prop(B, covered\_in, S) \wedge \\
& \qquad prop(S, type, sand).
\end{aligned}
$$

Example: trading agent

Possible hypothesis that could be learned:
"Joe likes resorts that are near sandy beaches."

$$
\begin{aligned}
& prop(joe, likes, R) \leftarrow \\
& \qquad prop(R, type, resort) \wedge \\
& \qquad prop(R, near, B) \wedge \\
& \qquad prop(B, type, beach) \wedge \\
& \qquad prop(B, covered\_in, S) \wedge \\
& \qquad prop(S, type, sand).
\end{aligned}
$$

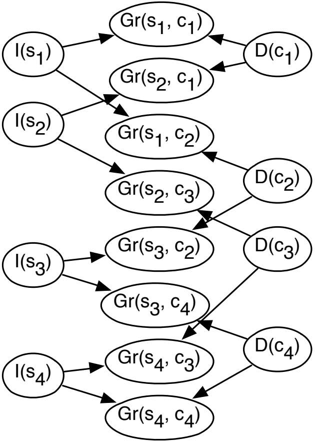
- But we want probabilistic predictions.

Example: Predicting Relations

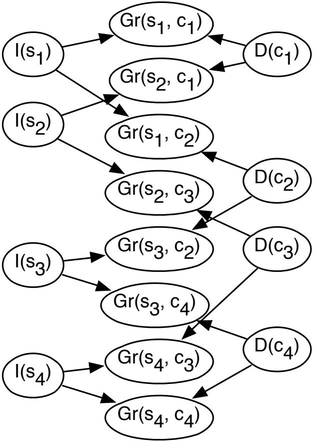
Student	*Course*	*Grade*
s_1	c_1	A
s_2	c_1	C
s_1	c_2	B
s_2	c_3	B
s_3	c_2	B
s_4	c_3	B
s_3	c_4	?
s_4	c_4	?

- Students s_3 and s_4 have the same averages, on courses with the same averages.
- Which student would you expect to better?

From Relations to Bayesian Belief Networks



From Relations to Bayesian Belief Networks

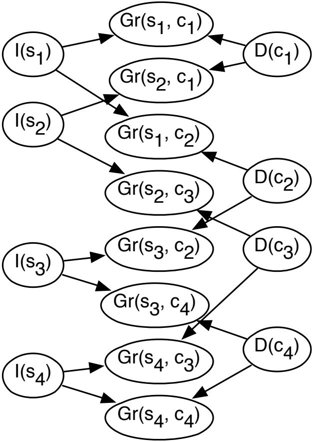


$I(S)$	$D(C)$	$Gr(S, C)$		
		A	B	C
true	*true*	0.5	0.4	0.1
true	*false*	0.9	0.09	0.01
false	*true*	0.01	0.09	0.9
false	*false*	0.1	0.4	0.5

$P(I(S)) = 0.5$
$P(D(C)) = 0.5$

"parameter sharing"

From Relations to Bayesian Belief Networks



$I(S)$	$D(C)$	$Gr(S,C)$ A	B	C
true	*true*	0.5	0.4	0.1
true	*false*	0.9	0.09	0.01
false	*true*	0.01	0.09	0.9
false	*false*	0.1	0.4	0.5

$P(I(S)) = 0.5$
$P(D(C)) = 0.5$

"parameter sharing"

http://artint.info/code/aispace/grades.xml

Example: Predicting Relations

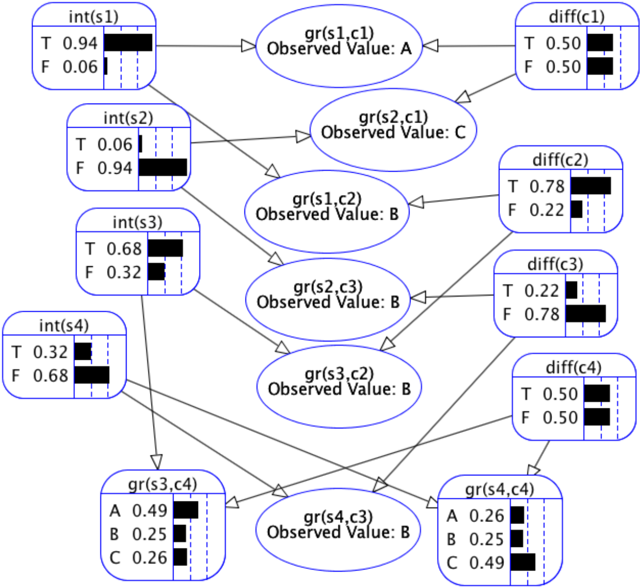
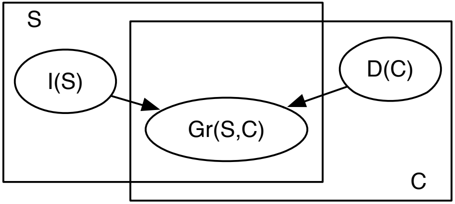
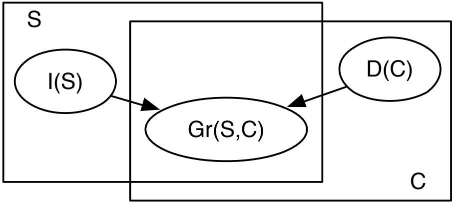


Plate Notation



- S, C logical variable representing students, courses
- the set of individuals of a type is called a population
- $I(S)$, $Gr(S, C)$, $D(C)$ are parametrized random variables

Plate Notation

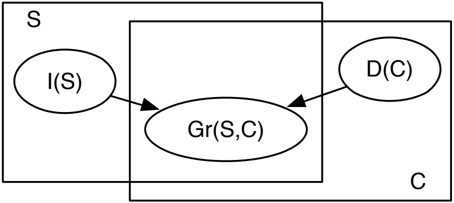


- S, C logical variable representing students, courses
- the set of individuals of a type is called a population
- $I(S)$, $Gr(S,C)$, $D(C)$ are parametrized random variables

Grounding:

- for every student s, there is

Plate Notation

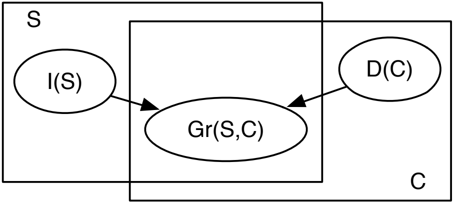


- S, C logical variable representing students, courses
- the set of individuals of a type is called a population
- $I(S)$, $Gr(S,C)$, $D(C)$ are parametrized random variables

Grounding:

- for every student s, there is a random variable $I(s)$
- for every course c, there is

Plate Notation

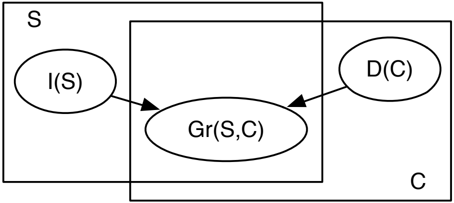


- S, C logical variable representing students, courses
- the set of individuals of a type is called a population
- $I(S)$, $Gr(S, C)$, $D(C)$ are parametrized random variables

Grounding:

- for every student s, there is a random variable $I(s)$
- for every course c, there is a random variable $D(c)$
- for every s, c pair there is

Plate Notation

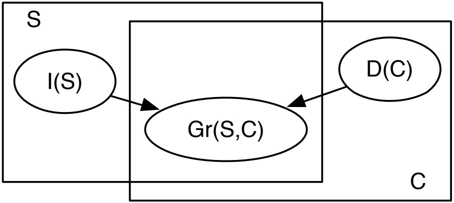


- S, C logical variable representing students, courses
- the set of individuals of a type is called a population
- $I(S)$, $Gr(S,C)$, $D(C)$ are parametrized random variables

Grounding:

- for every student s, there is a random variable $I(s)$
- for every course c, there is a random variable $D(c)$
- for every s, c pair there is a random variable $Gr(s,c)$

Plate Notation

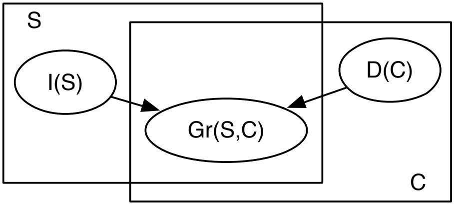


- S, C logical variable representing students, courses
- the set of individuals of a type is called a population
- $I(S)$, $Gr(S, C)$, $D(C)$ are parametrized random variables

Grounding:

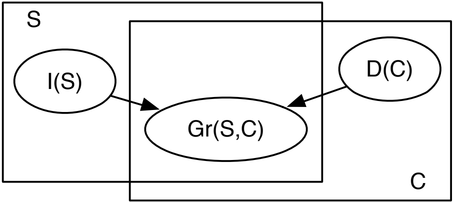
- for every student s, there is a random variable $I(s)$
- for every course c, there is a random variable $D(c)$
- for every s, c pair there is a random variable $Gr(s, c)$
- all instances share the same structure and parameters

Plate Notation



- If there were 1000 students and 100 courses:
  Grounding contains

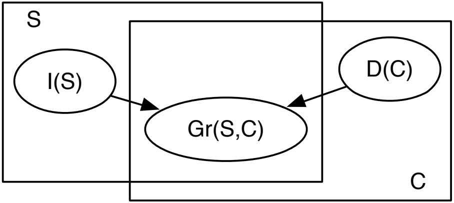
Plate Notation



- If there were 1000 students and 100 courses: Grounding contains
  - 1000 $I(s)$ variables
  - 100 $D(c)$ variables
  - 100000 $Gr(s,c)$ variables

  total: 101100 variables
- Numbers to be specified to define the probabilities:

Plate Notation



- If there were 1000 students and 100 courses:
  Grounding contains
  - 1000 $I(s)$ variables
  - 100 $D(c)$ variables
  - 100000 $Gr(s,c)$ variables

  total: 101100 variables
- Numbers to be specified to define the probabilities:
  1 for $I(S)$, 1 for $D(C)$, 8 for $Gr(S,C)$ = 10 parameters.

Exchangeability

- Before we know anything about individuals, they are indistinguishable, and so should be treated identically. exchangeability — names can be exchanged and the model doesn't change.

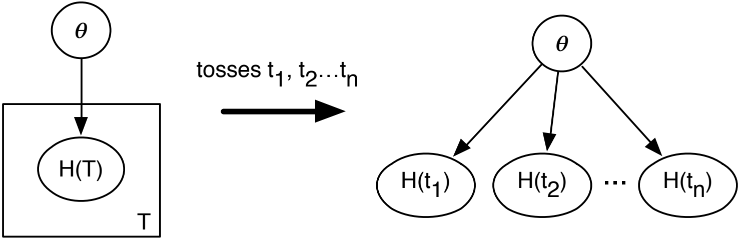
Exchangeability

- Before we know anything about individuals, they are indistinguishable, and so should be treated identically.
  exchangeability — names can be exchanged and the model doesn't change.

We model uncertainty about:

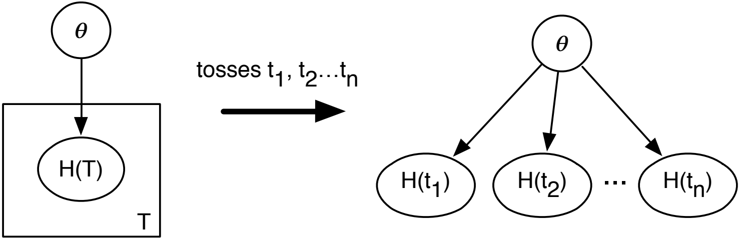
- Properties of individuals
- Relationships among individuals
- How properties and relations interrelate
- Identity (equality) of individuals
- Existence (and number) of individuals

Plate Notation for Learning Parameters



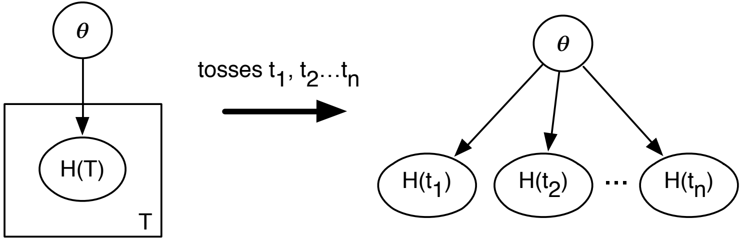
- T is a

Plate Notation for Learning Parameters



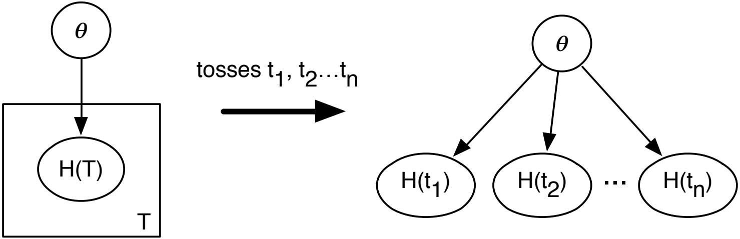
- T is a logical variable representing tosses of a thumb tack
- $H(t)$ is a

Plate Notation for Learning Parameters



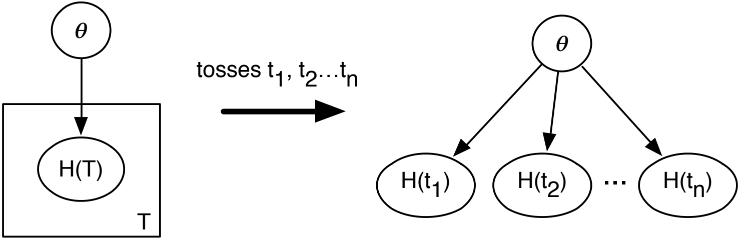
- T is a logical variable representing tosses of a thumb tack
- $H(t)$ is a Boolean variable that is true if toss t is heads.
- θ is a

Plate Notation for Learning Parameters



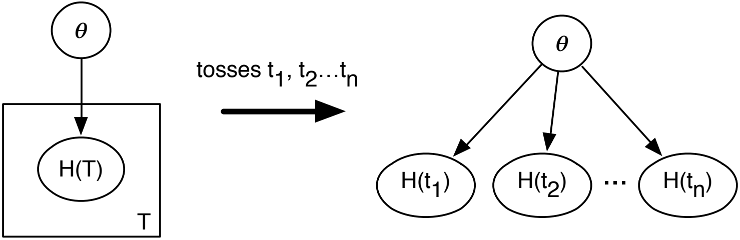
- T is a logical variable representing tosses of a thumb tack
- $H(t)$ is a Boolean variable that is true if toss t is heads.
- θ is a random variable representing the probability of heads.
- Range of θ is

Plate Notation for Learning Parameters



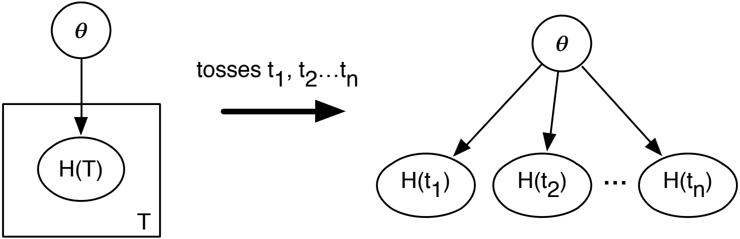
- T is a logical variable representing tosses of a thumb tack
- $H(t)$ is a Boolean variable that is true if toss t is heads.
- θ is a random variable representing the probability of heads.
- Range of θ is $\{0.0, 0.01, 0.02, \ldots, 0.99, 1.0\}$ or interval $[0, 1]$.
- $P(H(t_i){=}true|\theta{=}p) =$

Plate Notation for Learning Parameters



- T is a logical variable representing tosses of a thumb tack
- $H(t)$ is a Boolean variable that is true if toss t is heads.
- θ is a random variable representing the probability of heads.
- Range of θ is $\{0.0, 0.01, 0.02, \ldots, 0.99, 1.0\}$ or interval $[0, 1]$.
- $P(H(t_i){=}true|\theta{=}p) = p$

Plate Notation for Learning Parameters



- T is a logical variable representing tosses of a thumb tack
- $H(t)$ is a Boolean variable that is true if toss t is heads.
- θ is a random variable representing the probability of heads.
- Range of θ is $\{0.0, 0.01, 0.02, \ldots, 0.99, 1.0\}$ or interval $[0, 1]$.
- $P(H(t_i){=}true|\theta{=}p) = p$
- Independence: for $i \neq j$, $H(t_i)$ is independent of $H(t_j)$ given θ: i.i.d. or independent and identically distributed.

Parametrized belief networks

- Allow random variables to be parametrized. $interested(X)$
- Parameters correspond to logical variables. X
  logical variables can be drawn as plates.

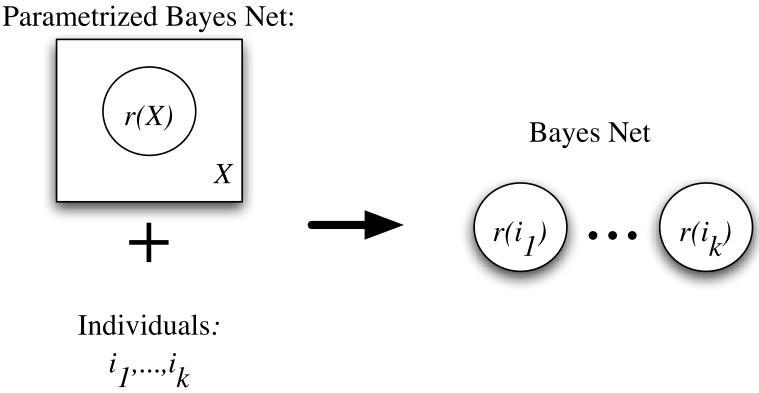
Parametrized belief networks

- Allow random variables to be parametrized. $interested(X)$
- Parameters correspond to logical variables. X
  logical variables can be drawn as plates.
- Each logical variable is typed with a population. $X : person$
- A population is a set of individuals.
- Each population has a size. $|person| = 1000000$

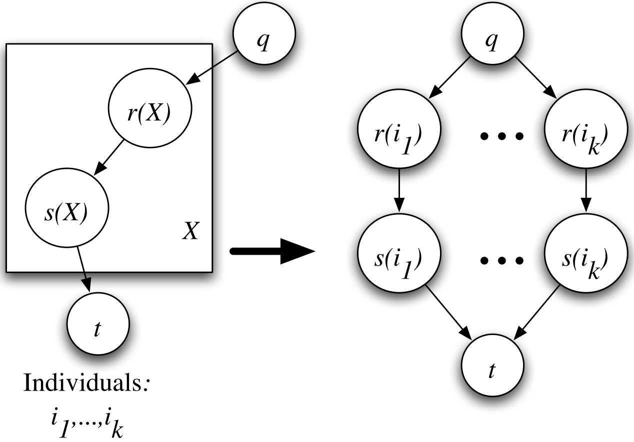
Parametrized belief networks

- Allow random variables to be parametrized. $interested(X)$
- Parameters correspond to logical variables. X
  logical variables can be drawn as plates.
- Each logical variable is typed with a population. $X : person$
- A population is a set of individuals.
- Each population has a size. $|person| = 1000000$
- Parametrized belief network means its grounding: an instance of each random variable for each assignment of an individual to a logical variable. $interested(p_1) \ldots interested(p_{1000000})$
- Instances are independent (but can have common ancestors and descendants).

Parametrized Bayesian networks / Plates

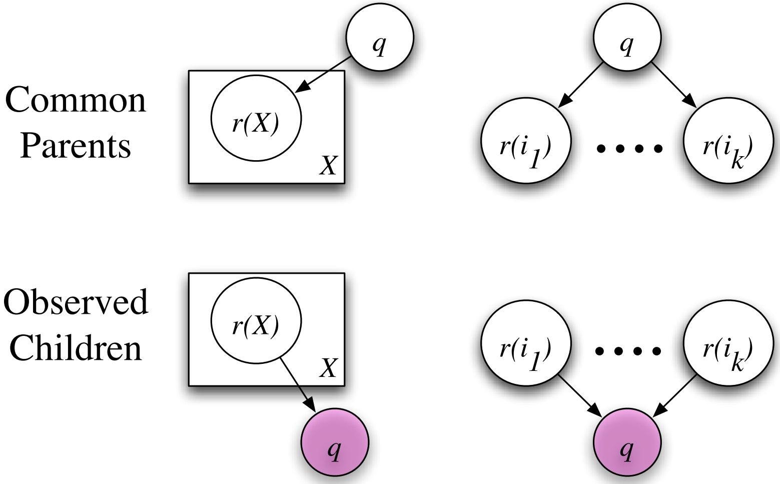


Parametrized Bayesian networks / Plates (2)

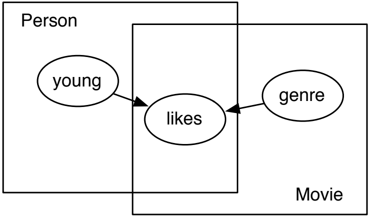


Creating Dependencies

Instances of plates are independent, except by common parents or children.

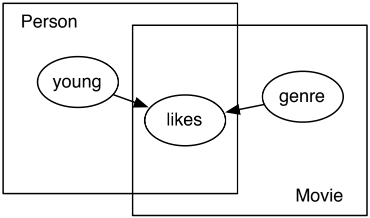


Overlapping plates



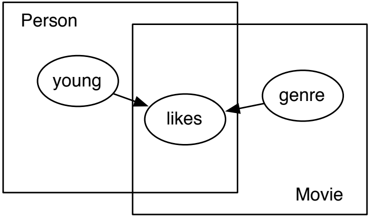
Relations:

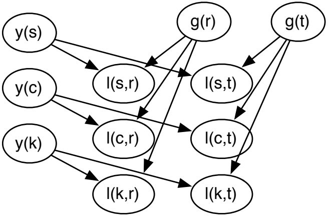
Overlapping plates



Relations: $likes(P, M)$, $young(P)$, $genre(M)$
likes is Boolean, *young* is Boolean,
genre has range $\{action, romance, family\}$
Three people: sam (s), chris (c), kim (k)
Two movies: rango (r), terminator (t)

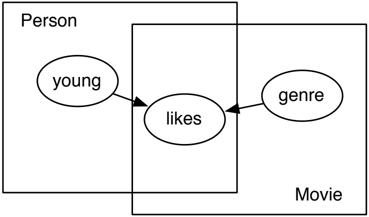
Overlapping plates





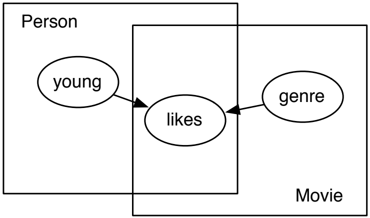
Relations: $likes(P, M)$, $young(P)$, $genre(M)$
likes is Boolean, *young* is Boolean,
genre has range $\{action, romance, family\}$
Three people: sam (s), chris (c), kim (k)
Two movies: rango (r), terminator (t)

Overlapping plates



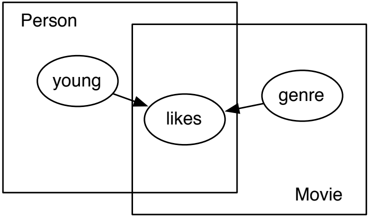
- Relations: $likes(P, M)$, $young(P)$, $genre(M)$
- $likes$ is Boolean, $young$ is Boolean, $genre$ has range $\{action, romance, family\}$
- If there are 1000 people and 100 movies,
  Grounding contains:
  random variables

Overlapping plates



- Relations: $likes(P, M)$, $young(P)$, $genre(M)$
- *likes* is Boolean, *young* is Boolean, *genre* has range $\{action, romance, family\}$
- If there are 1000 people and 100 movies,
  Grounding contains: 100,000 likes + 1,000 age + 100 genre = 101,100 random variables
- How many numbers need to be specified to define the probabilities required?

Overlapping plates

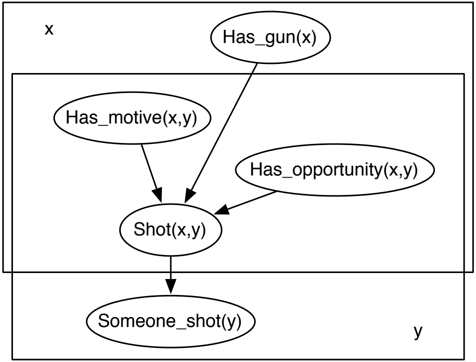


- Relations: $likes(P, M)$, $young(P)$, $genre(M)$
- *likes* is Boolean, *young* is Boolean, *genre* has range $\{action, romance, family\}$
- If there are 1000 people and 100 movies,
  Grounding contains: 100,000 likes + 1,000 age + 100 genre = 101,100 random variables
- How many numbers need to be specified to define the probabilities required?
  1 for *young*, 2 for *genre*, 6 for *likes* = 9 total.

Representing Conditional Probabilities

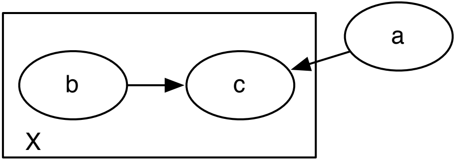
- $P(likes(P, M)|young(P), genre(M))$ — parameter sharing — individuals share probability parameters.
- $P(happy(X)|friend(X, Y), mean(Y))$ — needs aggregation — $happy(a)$ depends on an unbounded number of parents.
- There can be more structure about the individuals...

Example: Aggregation



Exercise #1

For the relational probabilistic model:

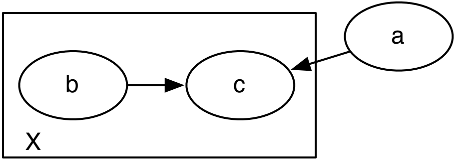


Suppose the the population of X is n and all variables are Boolean.

(a) How many random variables are in the grounding?

Exercise #1

For the relational probabilistic model:



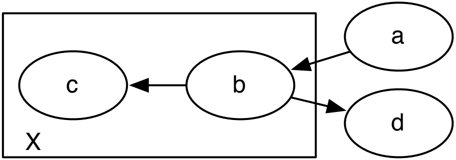
Suppose the the population of X is n and all variables are Boolean.

(a) How many random variables are in the grounding?

(b) How many numbers need to be specified for a tabular representation of the conditional probabilities?

Exercise #2

For the relational probabilistic model:

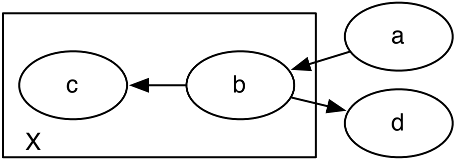


Suppose the the population of X is n and all variables are Boolean.

(a) Which of the conditional probabilities cannot be defined as a table?

Exercise #2

For the relational probabilistic model:



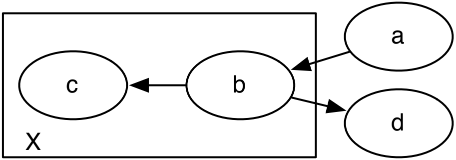
Suppose the the population of X is n and all variables are Boolean.

(a) Which of the conditional probabilities cannot be defined as a table?

(b) How many random variables are in the grounding?

Exercise #2

For the relational probabilistic model:



Suppose the the population of X is n and all variables are Boolean.

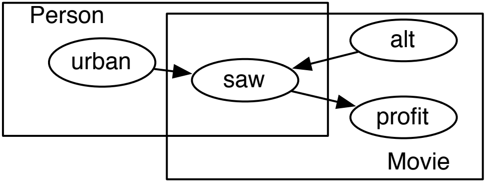
(a) Which of the conditional probabilities cannot be defined as a table?

(b) How many random variables are in the grounding?

(c) How many numbers need to be specified for a tabular representation of those conditional probabilities that can be defined using a table? (Assume an aggregator is an “or” which uses no numbers).

Exercise #3

For the relational probabilistic model:

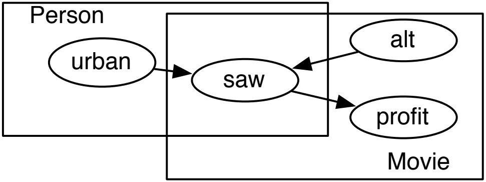


Suppose the population of *Person* is n and the population of *Movie* is m, and all variables are Boolean.

(a) How many random variables are in the grounding?

Exercise #3

For the relational probabilistic model:



Suppose the population of *Person* is n and the population of *Movie* is m, and all variables are Boolean.

(a) How many random variables are in the grounding?

(b) How many numbers are required to specify the conditional probabilities? (Assume an "or" is the aggregator and the rest are defined by tables).

Hierarchical Bayesian Model

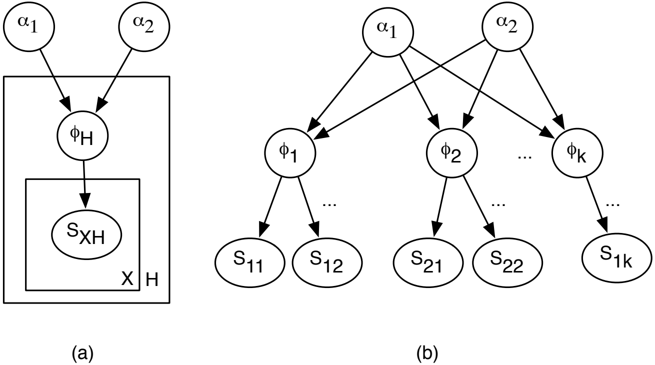
Example: S_{XH} is true when patient X is sick in hospital H.
We want to learn the probability of Sick for each hospital.

Hierarchical Bayesian Model

Example: S_{XH} is true when patient X is sick in hospital H.
We want to learn the probability of Sick for each hospital.
Where do the prior probabilities for the hospitals come from?

Hierarchical Bayesian Model

Example: S_{XH} is true when patient X is sick in hospital H.
We want to learn the probability of Sick for each hospital.
Where do the prior probabilities for the hospitals come from?

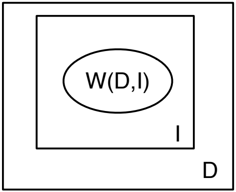


Example: Language Models

Unigram Model:

Example: Language Models

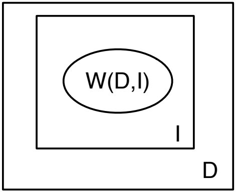
Unigram Model:



- D is the document
- I is the index of a word in the document. I ranges from 1 to the number of words in document D.

Example: Language Models

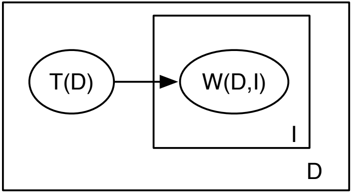
Unigram Model:



- D is the document
- I is the index of a word in the document. I ranges from 1 to the number of words in document D.
- $W(D, I)$ is the I'th word in document D. The range of W is the set of all words.

Example: Language Models

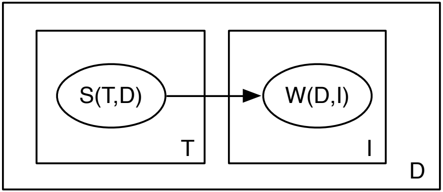
Topic Mixture:



- D is the document
- I is the index of a word in the document. I ranges from 1 to the number of words in document D.
- $W(d,i)$ is the i'th word in document d. The range of W is the set of all words.
- $T(d)$ is the topic of document d. The range of T is the set of all topics.

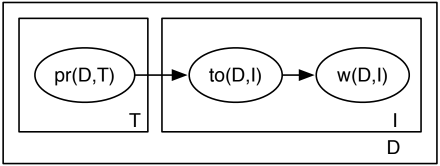
Example: Language Models

Mixture of topics, bag of words (unigram):



- D is the set of all documents
- I is the set of indexes of words in the document. I ranges from 1 to the number of words in the document.
- T is the set of all topics
- $W(d,i)$ is the i'th word in document d. The range of W is the set of all words.
- $S(t,d)$ is true if topic t is a subject of document d. S is Boolean.

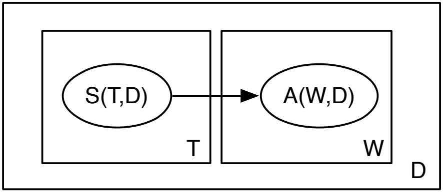
Example:Latent Dirichlet Allocation



- D is the document
- I is the index of a word in the document. I ranges from 1 to the number of words in document D.
- T is the topic
- $w(d, i)$ is the i'th word in document d. The range of w is the set of all words.
- $to(d, i)$ is the topic of the ith-word of document d. The range of to is the set of all topics.
- $pr(d, t)$ is is the proportion of document d that is about topic t. The range of pr is the reals.

Example: Language Models

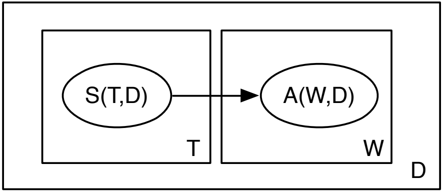
Mixture of topics, set of words:



- D is the set of all documents
- W is the set of all words.
- T is the set of all topics
- Boolean $A(w, d)$ is true if word w appears in document d.
- Boolean $S(t, d)$ is true if topic t is a subject of document d.

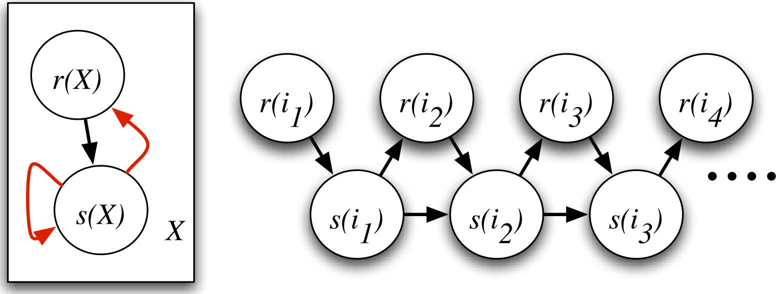
Example: Language Models

Mixture of topics, set of words:



- D is the set of all documents
- W is the set of all words.
- T is the set of all topics
- Boolean $A(w, d)$ is true if word w appears in document d.
- Boolean $S(t, d)$ is true if topic t is a subject of document d.
- Rephil (Google) has 900,000 topics, 12,000,000 "words", 350,000,000 links.

Creating Dependencies: Exploit Domain Structure

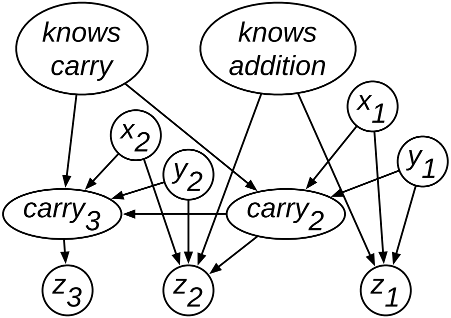


Example: diagnosing addition

$$
\begin{array}{rccc}
 & & x_2 & x_1 \\
+ & & y_2 & y_1 \\
\hline
 & z_3 & z_2 & z_1
\end{array}
$$

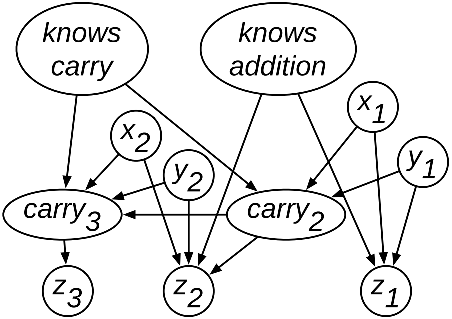
Example: diagnosing addition

$$\begin{array}{rrrr} & & x_2 & x_1 \\ + & & y_2 & y_1 \\ \hline & z_3 & z_2 & z_1 \end{array}$$



Example: diagnosing addition

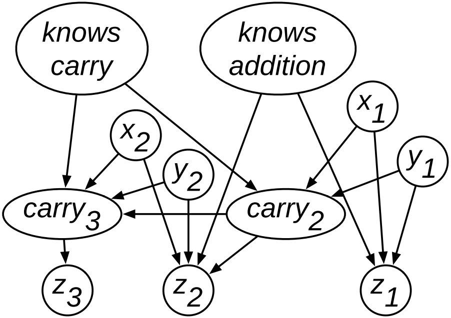
$$\begin{array}{cccc} & & x_2 & x_1 \\ + & & y_2 & y_1 \\ \hline & z_3 & z_2 & z_1 \end{array}$$



What if there were multiple digits

Example: diagnosing addition

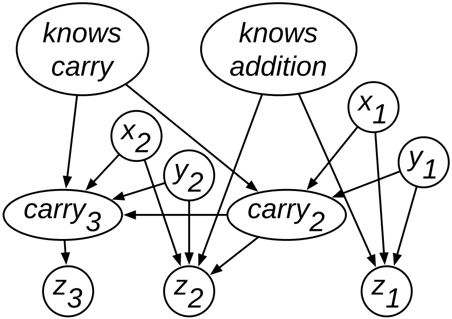
$$\begin{array}{rrrr} & & x_2 & x_1 \\ + & & y_2 & y_1 \\ \hline & z_3 & z_2 & z_1 \end{array}$$



What if there were multiple digits, problems

Example: diagnosing addition

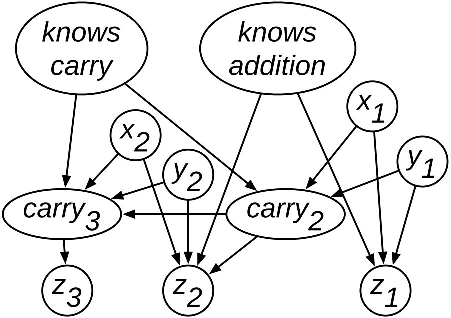
$$\begin{array}{rccc} & & x_2 & x_1 \\ + & & y_2 & y_1 \\ \hline & z_3 & z_2 & z_1 \end{array}$$



What if there were multiple digits, problems, students

Example: diagnosing addition

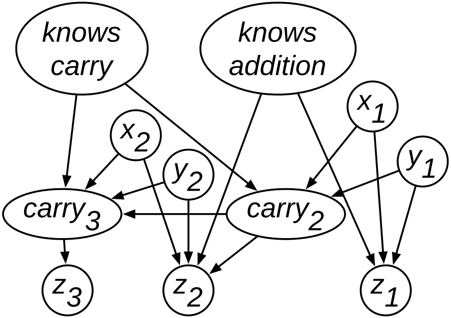
$$
\begin{array}{cccc}
 & & x_2 & x_1 \\
+ & & y_2 & y_1 \\
\hline
 & z_3 & z_2 & z_1
\end{array}
$$



What if there were multiple digits, problems, students, times?

Example: diagnosing addition

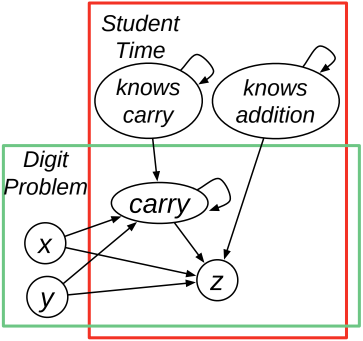
$$\begin{array}{rrrr} & & x_2 & x_1 \\ + & & y_2 & y_1 \\ \hline & z_3 & z_2 & z_1 \end{array}$$



What if there were multiple digits, problems, students, times?
How can we build a model before we know the individuals?

Multi-digit addition with parametrized BNs / plates

$$
\begin{array}{ccccc}
 & x_{j_x} & \cdots & x_2 & x_1 \\
+ & y_{j_z} & \cdots & y_2 & y_1 \\
\hline
 & z_{j_z} & \cdots & z_2 & z_1
\end{array}
$$



Random Variables: $x(D, P)$, $y(D, P)$, $knowsCarry(S, T)$, $knowsAddition(S, T)$, $carry(D, P, S, T)$, $z(D, P, S, T)$ for each: digit D, problem P, student S, time T

Relational Probabilistic Models

Often we want random variables for combinations of individuals in populations

- build a probabilistic model before knowing the individuals
- learn the model for one set of individuals
- apply the model to new individuals
- allow complex relationships between individuals

Outline

1. Logic and Probability
   - Relational Probabilistic Models
   - Probabilistic Logic Programs
2. Lifted Inference
   - Lifted Inference
   - Recursive Conditioning
   - Lifted Recursive Conditioning
3. Undirected models, Directed models, and Weighted Formulae
4. Existence and Identity Uncertainty

Independent Choice Logic (ICL)

- A language for relational probabilistic models.
- Idea: combine logic and probability, where all uncertainty in handled in terms of Bayesian decision theory, and logic specifies consequences of choices.

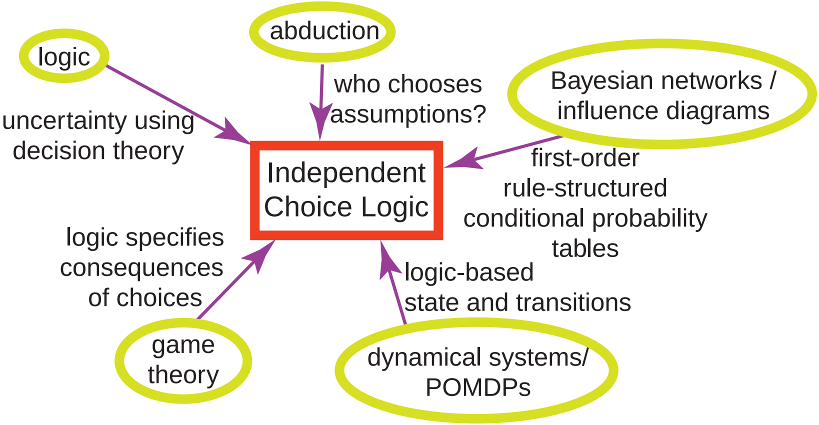
Independent Choice Logic (ICL)

- A language for relational probabilistic models.
- Idea: combine logic and probability, where all uncertainty in handled in terms of Bayesian decision theory, and logic specifies consequences of choices.
- An ICL theory consists of a choice space with probabilities over choices and a logic program that gives consequences of choices.

Independent Choice Logic (ICL)

- A language for relational probabilistic models.
- Idea: combine logic and probability, where all uncertainty in handled in terms of Bayesian decision theory, and logic specifies consequences of choices.
- An ICL theory consists of a choice space with probabilities over choices and a logic program that gives consequences of choices.
- History: parametrized Bayesian belief networks, abduction and default reasoning $\longrightarrow$ probabilistic Horn abduction (IJCAI-91); richer language (negation as failure + choices by other agents $\longrightarrow$ independent choice logic (AIJ 1997) $\longrightarrow$ Problog (probabilistic programming language)

The independent choice logic influences



Independent Choice Logic

- An atomic hypothesis is an atomic formula.
  An alternative is a set of atomic hypotheses.
  $\mathcal{C}$, the choice space is a set of disjoint alternatives.

Independent Choice Logic

- An atomic hypothesis is an atomic formula.
  An alternative is a set of atomic hypotheses.
  $\mathcal{C}$, the choice space is a set of disjoint alternatives.
- $\mathcal{F}$, the facts is an acyclic logic program that gives consequences of choices (can contain negation as failure).
  No atomic hypothesis is the head of a rule.

Independent Choice Logic

- An atomic hypothesis is an atomic formula.
  An alternative is a set of atomic hypotheses.
  $\mathcal{C}$, the choice space is a set of disjoint alternatives.
- $\mathcal{F}$, the facts is an acyclic logic program that gives consequences of choices (can contain negation as failure).
  No atomic hypothesis is the head of a rule.
- P_0 a probability distribution over alternatives:

$$\forall A \in \mathcal{C} \ \sum_{a \in A} P_0(a) = 1.$$

Meaningless Example

$$\mathcal{C} = \{\{c_1, c_2, c_3\}, \{b_1, b_2\}\}$$

$$\begin{aligned}\mathcal{F} = \{\ & f \leftarrow c_1 \wedge b_1, & & f \leftarrow c_3 \wedge b_2, \\ & d \leftarrow c_1, & & d \leftarrow \ \sim c_2 \wedge b_1, \\ & e \leftarrow f, & & e \leftarrow \ \sim d\}\end{aligned}$$

$$\begin{array}{lll} P_0(c_1) = 0.5 & P_0(c_2) = 0.3 & P_0(c_3) = 0.2 \\ P_0(b_1) = 0.9 & P_0(b_2) = 0.1 & \end{array}$$

Semantics of ICL

- There is a possible world for each selection of one element from each alternative.
- The logic program together with the selected atoms specifies what is true in each possible world.
- The elements of different alternatives are probabilistically independent.

Meaningless Example: Semantics

$$\begin{aligned}\mathcal{F} = \{ \; & f \leftarrow c_1 \wedge b_1, & & f \leftarrow c_3 \wedge b_2, \\ & d \leftarrow c_1, & & d \leftarrow \sim c_2 \wedge b_1, \\ & e \leftarrow f, & & e \leftarrow \sim d \}\end{aligned}$$

$$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$$
$$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$$

$$w_1 \models \overbrace{c_1 \quad b_1}^{\text{selection}} \quad \overbrace{\qquad\qquad\qquad}^{\text{logic program}}$$

Meaningless Example: Semantics

$$\mathcal{F} = \{\ f \leftarrow c_1 \wedge b_1,\quad f \leftarrow c_3 \wedge b_2,\quad d \leftarrow c_1,\quad d \leftarrow\ \sim c_2 \wedge b_1,\quad e \leftarrow f,\quad e \leftarrow\ \sim d\}$$

$$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$$
$$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$$

$$w_1 \models \underbrace{c_1 \quad b_1}_{\text{selection}} \quad \underbrace{f \quad d \quad e}_{\text{logic program}} \qquad P(w_1) =$$

Meaningless Example: Semantics

$$
\begin{aligned}
\mathcal{F} = \{\; & f \leftarrow c_1 \wedge b_1, && f \leftarrow c_3 \wedge b_2, \\
& d \leftarrow c_1, && d \leftarrow \sim c_2 \wedge b_1, \\
& e \leftarrow f, && e \leftarrow \sim d \}
\end{aligned}
$$

$$
\begin{array}{lll}
P_0(c_1) = 0.5 & P_0(c_2) = 0.3 & P_0(c_3) = 0.2 \\
P_0(b_1) = 0.9 & P_0(b_2) = 0.1 &
\end{array}
$$

$$
\begin{array}{llllllll}
 & & \overbrace{\quad\quad\quad\quad}^{\text{selection}} & & \overbrace{\quad\quad\quad\quad\quad\quad}^{\text{logic program}} & & & \\
w_1 & \models & c_1 & b_1 & f & d & e & P(w_1) = 0.45 \\
w_2 & \models & c_2 & b_1 & & & &
\end{array}
$$

Meaningless Example: Semantics

$$\mathcal{F} = \{ \; f \leftarrow c_1 \wedge b_1, \quad f \leftarrow c_3 \wedge b_2,$$
$$d \leftarrow c_1, \quad d \leftarrow \sim c_2 \wedge b_1,$$
$$e \leftarrow f, \quad e \leftarrow \sim d \}$$

$$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$$
$$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$$

$$\begin{array}{lllllllll} w_1 & \models & \overbrace{c_1 \quad b_1}^{\text{selection}} & \overbrace{f \quad d \quad e}^{\text{logic program}} & & P(w_1) = 0.45 \\ w_2 & \models & c_2 \quad b_1 & \sim f \quad \sim d \quad e & & P(w_2) = \end{array}$$

Meaningless Example: Semantics

$$\mathcal{F} = \{\ f \leftarrow c_1 \wedge b_1,\quad f \leftarrow c_3 \wedge b_2,\\ d \leftarrow c_1,\quad d \leftarrow \sim c_2 \wedge b_1,\\ e \leftarrow f,\quad e \leftarrow \sim d\}$$

$$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$$
$$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$$

$$\begin{array}{lllllllll} & & \overbrace{\qquad\qquad}^{\text{selection}} & & \overbrace{\qquad\qquad\qquad}^{\text{logic program}} & & & \\ w_1 & \models & c_1 & b_1 & f & d & e & P(w_1) = 0.45 \\ w_2 & \models & c_2 & b_1 & \sim f & \sim d & e & P(w_2) = 0.27 \\ w_3 & \models & c_3 & b_1 & & & & \end{array}$$

Meaningless Example: Semantics

$$\begin{aligned}\mathcal{F} = \{\ & f \leftarrow c_1 \wedge b_1, & & f \leftarrow c_3 \wedge b_2,\\ & d \leftarrow c_1, & & d \leftarrow\ \sim c_2 \wedge b_1,\\ & e \leftarrow f, & & e \leftarrow\ \sim d\}\end{aligned}$$

$$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$$
$$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$$

$$\begin{array}{lllllll} & & \overbrace{\quad\quad\quad\quad}^{\text{selection}} & & \overbrace{\quad\quad\quad\quad\quad\quad}^{\text{logic program}} & & \\ w_1 & \models & c_1 \quad b_1 & & f \quad d \quad e & & P(w_1) = 0.45 \\ w_2 & \models & c_2 \quad b_1 & & \sim f \quad \sim d \quad e & & P(w_2) = 0.27 \\ w_3 & \models & c_3 \quad b_1 & & \sim f \quad d \quad \sim e & & P(w_3) = \end{array}$$

Meaningless Example: Semantics

$$\mathcal{F} = \{\ f \leftarrow c_1 \wedge b_1,\quad f \leftarrow c_3 \wedge b_2,$$
$$d \leftarrow c_1,\quad d \leftarrow\ \sim c_2 \wedge b_1,$$
$$e \leftarrow f,\quad e \leftarrow\ \sim d\}$$

$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$
$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$

$$\begin{array}{llllllll}
 & & \overbrace{\qquad\qquad}^{\text{selection}} & & \overbrace{\qquad\qquad\qquad\qquad}^{\text{logic program}} & & & \\
w_1 & \models & c_1 & b_1 & f & d & e & P(w_1) = 0.45 \\
w_2 & \models & c_2 & b_1 & \sim f & \sim d & e & P(w_2) = 0.27 \\
w_3 & \models & c_3 & b_1 & \sim f & d & \sim e & P(w_3) = 0.18 \\
w_4 & \models & c_1 & b_2 & & & &
\end{array}$$

Meaningless Example: Semantics

$$\mathcal{F} = \{\; f \leftarrow c_1 \wedge b_1, \quad f \leftarrow c_3 \wedge b_2,$$
$$d \leftarrow c_1, \quad d \leftarrow\ \sim c_2 \wedge b_1,$$
$$e \leftarrow f, \quad e \leftarrow\ \sim d\}$$

$$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$$
$$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$$

		selection		logic program			
w_1	$\models$	c_1	b_1	f	d	e	$P(w_1) = 0.45$
w_2	$\models$	c_2	b_1	$\sim f$	$\sim d$	e	$P(w_2) = 0.27$
w_3	$\models$	c_3	b_1	$\sim f$	d	$\sim e$	$P(w_3) = 0.18$
w_4	$\models$	c_1	b_2	$\sim f$	d	$\sim e$	$P(w_4) =$

Meaningless Example: Semantics

$$
\begin{aligned}
\mathcal{F} = \{\ & f \leftarrow c_1 \wedge b_1, & & f \leftarrow c_3 \wedge b_2, \\
& d \leftarrow c_1, & & d \leftarrow\ \sim c_2 \wedge b_1, \\
& e \leftarrow f, & & e \leftarrow\ \sim d\}
\end{aligned}
$$

$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$
$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$

$$
\begin{array}{llcccccc}
 & & \overbrace{\quad\quad\quad\quad}^{\text{selection}} & & \overbrace{\quad\quad\quad\quad\quad\quad}^{\text{logic program}} & & & \\
w_1 & \models & c_1 & b_1 & f & d & e & P(w_1) = 0.45 \\
w_2 & \models & c_2 & b_1 & \sim f & \sim d & e & P(w_2) = 0.27 \\
w_3 & \models & c_3 & b_1 & \sim f & d & \sim e & P(w_3) = 0.18 \\
w_4 & \models & c_1 & b_2 & \sim f & d & \sim e & P(w_4) = 0.05 \\
w_5 & \models & c_2 & b_2 & & & &
\end{array}
$$

Meaningless Example: Semantics

$$\mathcal{F} = \{ \begin{array}{ll} f \leftarrow c_1 \wedge b_1, & f \leftarrow c_3 \wedge b_2, \\ d \leftarrow c_1, & d \leftarrow \sim c_2 \wedge b_1, \\ e \leftarrow f, & e \leftarrow \sim d \} \end{array}$$

$$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$$
$$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$$

$$\begin{array}{llllllll} & & \overbrace{\qquad\qquad}^{\text{selection}} & & \overbrace{\qquad\qquad\qquad}^{\text{logic program}} & & & \\ w_1 & \models & c_1 & b_1 & f & d & e & P(w_1) = 0.45 \\ w_2 & \models & c_2 & b_1 & \sim f & \sim d & e & P(w_2) = 0.27 \\ w_3 & \models & c_3 & b_1 & \sim f & d & \sim e & P(w_3) = 0.18 \\ w_4 & \models & c_1 & b_2 & \sim f & d & \sim e & P(w_4) = 0.05 \\ w_5 & \models & c_2 & b_2 & \sim f & \sim d & e & P(w_5) \end{array}$$

Meaningless Example: Semantics

$$\mathcal{F} = \{ \begin{array}{ll} f \leftarrow c_1 \wedge b_1, & f \leftarrow c_3 \wedge b_2, \\ d \leftarrow c_1, & d \leftarrow \sim c_2 \wedge b_1, \\ e \leftarrow f, & e \leftarrow \sim d \} \end{array}$$

$$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$$
$$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$$

$$\begin{array}{cccccccc} & & \overbrace{\hspace{4em}}^{\text{selection}} & & \overbrace{\hspace{8em}}^{\text{logic program}} & & & \\ w_1 & \models & c_1 & b_1 & f & d & e & P(w_1) = 0.45 \\ w_2 & \models & c_2 & b_1 & \sim f & \sim d & e & P(w_2) = 0.27 \\ w_3 & \models & c_3 & b_1 & \sim f & d & \sim e & P(w_3) = 0.18 \\ w_4 & \models & c_1 & b_2 & \sim f & d & \sim e & P(w_4) = 0.05 \\ w_5 & \models & c_2 & b_2 & \sim f & \sim d & e & P(w_5) = 0.03 \\ w_6 & \models & c_3 & b_2 & & & & \end{array}$$

Meaningless Example: Semantics

$$\mathcal{F} = \{ \begin{array}{ll} f \leftarrow c_1 \wedge b_1, & f \leftarrow c_3 \wedge b_2, \\ d \leftarrow c_1, & d \leftarrow \sim c_2 \wedge b_1, \\ e \leftarrow f, & e \leftarrow \sim d \} \end{array}$$

$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$
$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$

		selection		logic program			
w_1	$\models$	c_1	b_1	f	d	e	$P(w_1) = 0.45$
w_2	$\models$	c_2	b_1	$\sim f$	$\sim d$	e	$P(w_2) = 0.27$
w_3	$\models$	c_3	b_1	$\sim f$	d	$\sim e$	$P(w_3) = 0.18$
w_4	$\models$	c_1	b_2	$\sim f$	d	$\sim e$	$P(w_4) = 0.05$
w_5	$\models$	c_2	b_2	$\sim f$	$\sim d$	e	$P(w_5) = 0.03$
w_6	$\models$	c_3	b_2	f	$\sim d$	e	$P(w_6) =$

Meaningless Example: Semantics

$$\begin{aligned}
\mathcal{F} = \{\ & f \leftarrow c_1 \wedge b_1, && f \leftarrow c_3 \wedge b_2, \\
& d \leftarrow c_1, && d \leftarrow \sim c_2 \wedge b_1, \\
& e \leftarrow f, && e \leftarrow \sim d\}
\end{aligned}$$

$$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$$
$$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$$

$$\begin{array}{ccccccccc}
 & & \overbrace{\quad\quad\quad\quad}^{\text{selection}} & & \overbrace{\quad\quad\quad\quad\quad\quad}^{\text{logic program}} & & & & \\
w_1 & \models & c_1 & b_1 & f & d & e & & P(w_1) = 0.45 \\
w_2 & \models & c_2 & b_1 & \sim f & \sim d & e & & P(w_2) = 0.27 \\
w_3 & \models & c_3 & b_1 & \sim f & d & \sim e & & P(w_3) = 0.18 \\
w_4 & \models & c_1 & b_2 & \sim f & d & \sim e & & P(w_4) = 0.05 \\
w_5 & \models & c_2 & b_2 & \sim f & \sim d & e & & P(w_5) = 0.03 \\
w_6 & \models & c_3 & b_2 & f & \sim d & e & & P(w_6) = 0.02
\end{array}$$

$$P(e) = 0.45 + 0.27 + 0.03 + 0.02 = 0.77$$

Contingently Acyclic Logic Programs

Disallowed

- $a \leftarrow \sim b. \qquad b \leftarrow \sim a.$

Contingently Acyclic Logic Programs

Disallowed

- $a \leftarrow\sim b. \qquad b \leftarrow\sim a.$
  two stable models $a \wedge \neg b$ and $\neg a \wedge b$.

Contingently Acyclic Logic Programs

Disallowed

- $a \leftarrow \sim b. \qquad b \leftarrow \sim a.$
  two stable models $a \wedge \neg b$ and $\neg a \wedge b$.
- $a \leftarrow \sim a.$

Contingently Acyclic Logic Programs

Disallowed

- $a \leftarrow\sim b. \qquad b \leftarrow\sim a.$
  two stable models $a \wedge \neg b$ and $\neg a \wedge b$.
- $a \leftarrow\sim a.$
  no stable models

Contingently Acyclic Logic Programs

Disallowed

- $a \leftarrow \sim b. \qquad b \leftarrow \sim a.$
  two stable models $a \wedge \neg b$ and $\neg a \wedge b$.
- $a \leftarrow \sim a.$
  no stable models

Allowed

- $p(do(A, X)) \leftarrow p(X) \wedge rest. \qquad p(init).$
  well founded recursions are good!

Contingently Acyclic Logic Programs

Disallowed

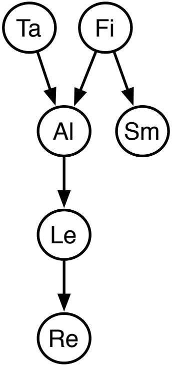
- $a \leftarrow\sim b.\qquad b \leftarrow\sim a.$
  two stable models $a \wedge \neg b$ and $\neg a \wedge b$.
- $a \leftarrow\sim a.$
  no stable models

Allowed

- $p(do(A, X)) \leftarrow p(X) \wedge rest.\qquad p(init).$
  well founded recursions are good!
- $a \leftarrow b \wedge c.\qquad b \leftarrow a \wedge \sim c.$
  only one body will be true in any possible world.

Belief Networks, Decision trees and ICL rules

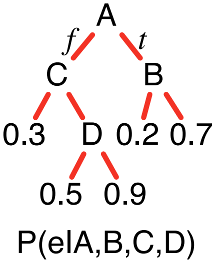
- There is a local mapping from Bayesian belief networks into ICL.



$$\begin{array}{l}
\text{prob } ta : 0.02.\\
\text{prob } fire : 0.01.\\
alarm \leftarrow ta \wedge fire \wedge atf.\\
alarm \leftarrow \sim ta \wedge fire \wedge antf.\\
alarm \leftarrow ta \wedge \sim fire \wedge atnf.\\
alarm \leftarrow \sim ta \wedge \sim fire \wedge antnf.\\
\text{prob } atf : 0.5.\\
\text{prob } antf : 0.99.\\
\text{prob } atnf : 0.85.\\
\text{prob } antnf : 0.0001.\\
smoke \leftarrow fire \wedge sf.\\
\text{prob } sf : 0.9.\\
smoke \leftarrow \sim fire \wedge snf.\\
\text{prob } snf : 0.01.
\end{array}$$

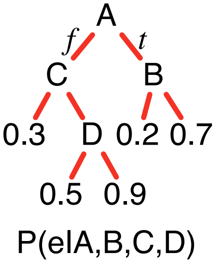
Belief Networks, Decision trees and ICL rules

- Rules can represent decision tree with probabilities:



Belief Networks, Decision trees and ICL rules

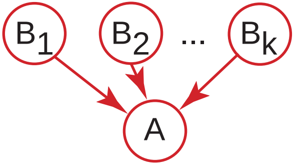
- Rules can represent decision tree with probabilities:



$e \leftarrow a \wedge b \wedge h_1.$	$P_0(h_1) = 0.7$
$e \leftarrow a \wedge \sim b \wedge h_2.$	$P_0(h_2) = 0.2$
$e \leftarrow \sim a \wedge c \wedge d \wedge h_3.$	$P_0(h_3) = 0.9$
$e \leftarrow \sim a \wedge c \wedge \sim d \wedge h_4.$	$P_0(h_4) = 0.5$
$e \leftarrow \sim a \wedge \sim c \wedge h_5.$	$P_0(h_5) = 0.3$

Mapping belief networks into ICL

There is a local mapping from belief networks into ICL:



is translated into the rules

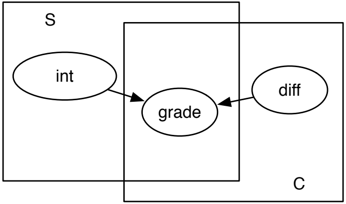
$$a(V) \leftarrow b_1(V_1) \wedge \cdots \wedge b_k(V_k) \wedge h(V, V_1, \ldots, V_k).$$

and the alternatives

$$\forall v_1 \cdots \forall v_k \{h(v, v_1, \ldots, v_k) \mid v \in domain(a)\} \in \mathcal{C}$$

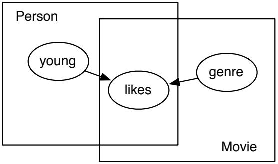
Predicting Grades

Plates correspond to logical variables.



prob $int(S) : 0.5.$
prob $diff(C) : 0.5.$
$grade(S, C, G) \leftarrow int(S) \wedge diff(C) \wedge idg(S, C, G).$
prob $idg(S, C, a) : 0.5,\ idg(S, C, b) : 0.4,\ idg(S, C, c) : 0.1.$
$grade(S, C, G) \leftarrow int(S) \wedge \sim diff(C) \wedge indg(S, C, G).$
prob $indg(S, C, a) : 0.9,\ indg(S, C, b) : 0.09,\ indg(S, C, c) : 0.01.$
...

Movie Ratings



prob $young(P) : 0.4.$

prob $genre(M, action) : 0.4, genre(M, romance) : 0.3,$
$\qquad genre(M, family) : 0.4.$

$likes(P, M) \leftarrow young(P) \wedge genre(M, G) \wedge ly(P, M, G).$

$likes(P, M) \leftarrow \sim young(P) \wedge genre(M, G) \wedge lny(P, M, G).$

prob $ly(P, M, action) : 0.7.$

prob $ly(P, M, romance) : 0.3.$

prob $ly(P, M, family) : 0.8.$

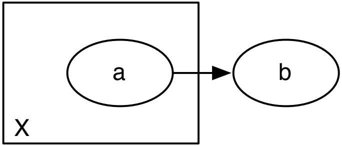
prob $lny(P, M, action) : 0.2.$

prob $lny(P, M, romance) : 0.9.$

prob $lny(P, M, family) : 0.3.$

Aggregation

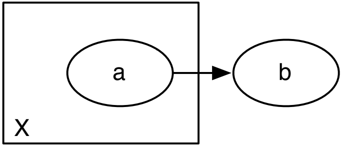
The relational probabilistic model:



Cannot be represented using tables. Why?

Aggregation

The relational probabilistic model:



Cannot be represented using tables. Why?

- This can be represented in ICL by

  $$b \leftarrow a(X) \& n(X).$$

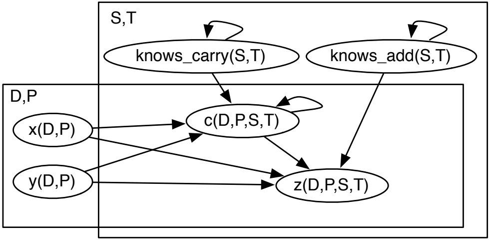
  "noisy-or", where $n(X)$ is a noise term, $\{n(c), \sim n(c)\} \in \mathcal{C}$ for each individual c.
- If $a(c)$ is observed for each individual c:

  $$P(b) = 1 - (1 - p)^k$$

  Where $p = P(n(X))$ and k is the number of $a(c)$ that are true.

Example: Multi-digit addition

$$\begin{array}{rcccc} & x_{j_x} & \cdots & x_2 & x_1 \\ + & y_{j_z} & \cdots & y_2 & y_1 \\ \hline & z_{j_z} & \cdots & z_2 & z_1 \end{array}$$



ICL rules for multi-digit addition

$$
\begin{aligned}
z(D,P,S,T) = V \leftarrow\ & \\
& x(D,P) = Vx \wedge \\
& y(D,P) = Vy \wedge \\
& c(D,P,S,T) = Vc \wedge \\
& knows\_add(S,T) \wedge \\
& \neg mistake(D,P,S,T) \wedge \\
& V \text{ is } (Vx + Vy + Vc) \text{ div } 10.
\end{aligned}
$$

$$
\begin{aligned}
z(D,P,S,T) = V \leftarrow\ & \\
& knows\_add(S,T) \wedge \\
& mistake(D,P,S,T) \wedge \\
& selectDig(D,P,S,T) = V. \\
z(D,P,S,T) = V \leftarrow\ & \\
& \neg knows\_add(S,T) \wedge \\
& selectDig(D,P,S,T) = V.
\end{aligned}
$$

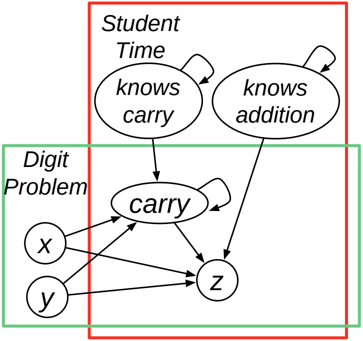
Alternatives:

$\forall DPST\{noMistake(D,P,S,T), mistake(D,P,S,T)\}$

$\forall DPST\{selectDig(D,P,S,T) = V \mid V \in \{0..9\}\}$

Multi-digit addition with parametrized BNs / plates

$$
\begin{array}{cccc}
 & x_{j_x} & \cdots & x_2 & x_1 \\
+ & y_{j_z} & \cdots & y_2 & y_1 \\
\hline
 & z_{j_z} & \cdots & z_2 & z_1
\end{array}
$$



Random Variables: $x(D, P)$, $y(D, P)$, $knowsCarry(S, T)$, $knowsAddition(S, T)$, $carry(D, P, S, T)$, $z(D, P, S, T)$
for each: digit D, problem P, student S, time T

☛ parametrized random variables

ICL rules for multi-digit addition

$$
\begin{aligned}
z(D,P,S,T) = V \leftarrow\\
&x(D,P) = Vx \wedge\\
&y(D,P) = Vy \wedge\\
&carry(D,P,S,T) = Vc \wedge\\
&knowsAddition(S,T) \wedge\\
&\neg mistake(D,P,S,T) \wedge\\
&V \text{ is } (Vx + Vy + Vc) \text{ div } 10.
\end{aligned}
$$

$$
\begin{aligned}
z(D,P,S,T) = V \leftarrow\\
&knowsAddition(S,T) \wedge\\
&mistake(D,P,S,T) \wedge\\
&selectDig(D,P,S,T) = V.\\
z(D,P,S,T) = V \leftarrow\\
&\neg knowsAddition(S,T) \wedge\\
&selectDig(D,P,S,T) = V.
\end{aligned}
$$

Alternatives:

$\forall DPST\{noMistake(D,P,S,T), mistake(D,P,S,T)\}$

$\forall DPST\{selectDig(D,P,S,T) = V \mid V \in \{0..9\}\}$

Outline

1. Logic and Probability
   - Relational Probabilistic Models
   - Probabilistic Logic Programs
2. Lifted Inference
   - Lifted Inference
   - Recursive Conditioning
   - Lifted Recursive Conditioning
3. Undirected models, Directed models, and Weighted Formulae
4. Existence and Identity Uncertainty

Why Exact Inference?

Why do we care about exact inference?

- Gold standard

Why Exact Inference?

Why do we care about exact inference?

- Gold standard
- Size of problems amenable to exact inference is growing

Why Exact Inference?

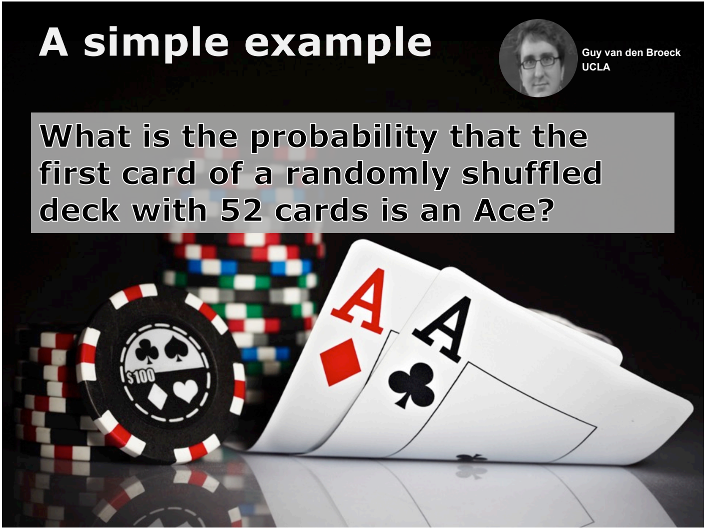
Why do we care about exact inference?

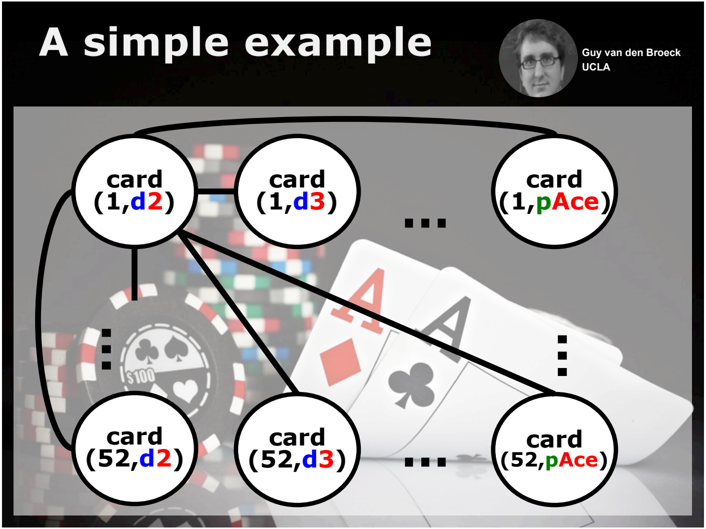
- Gold standard
- Size of problems amenable to exact inference is growing
- Learning for inference

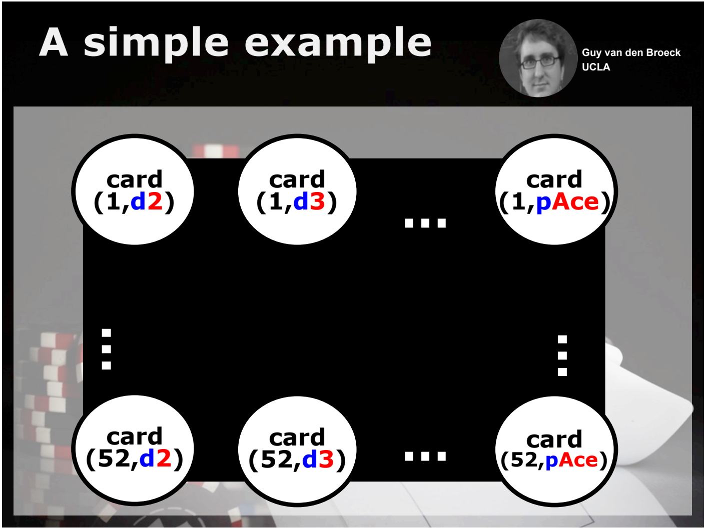
Why Exact Inference?

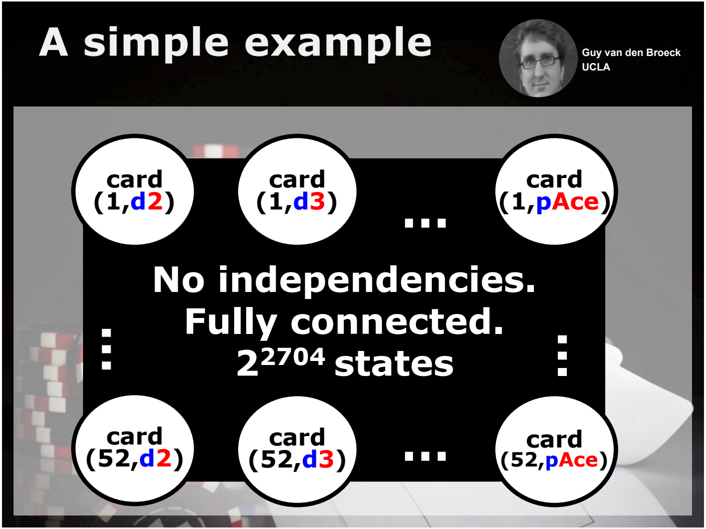
Why do we care about exact inference?

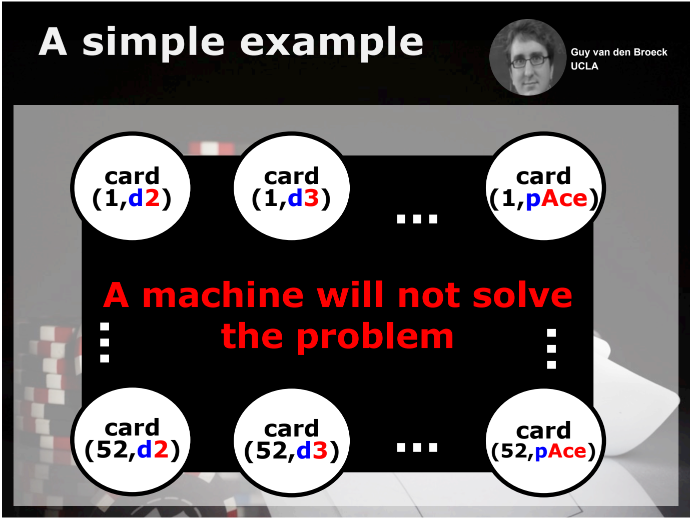
- Gold standard
- Size of problems amenable to exact inference is growing
- Learning for inference
- Basis for efficient approximate inference:
    - Rao-Blackwellization
    - Variational Methods

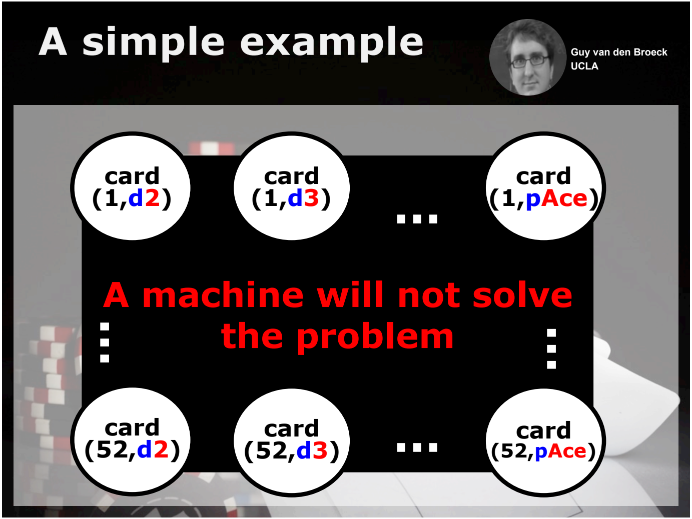

A simple example
Guy van den Broeck
UCLA
What is the probability that the first card of a randomly shuffled deck with 52 cards is an Ace?
$100
A
A


A simple example
Guy van den Broeck
UCLA
card (1,d2)
card (1,d3)
...
card (1,pAce)
card (52,d2)
card (52,d3)
...
card (52,pAce)


A simple example
Guy van den Broeck
UCLA
card (1,d2)
card (1,d3)
...
card (1,pAce)
card (52,d2)
card (52,d3)
...
card (52,pAce)


A simple example
Guy van den Broeck
UCLA
card (1,d2)
card (1,d3)
...
card (1,pAce)
No independencies.
Fully connected.
2^2704 states
card (52,d2)
card (52,d3)
...
card (52,pAce)


A simple example
Guy van den Broeck
UCLA
card (1,d2)
card (1,d3)
...
card (1,pAce)
A machine will not solve the problem
card (52,d2)
card (52,d3)
...
card (52,pAce)



... unless it can represent and exploit symmetry.

Outline

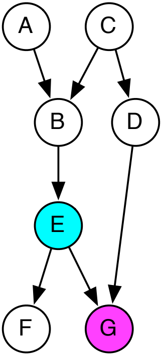
Lifted Inference

- Idea: treat those individuals about which you have the same information as a block; just count them.
- Use the ideas from lifted theorem proving - no need to ground.
- Potential to be exponentially faster in the number of non-differentialed individuals.
- Relies on knowing the number of individuals (the population size).

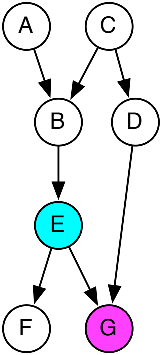
Outline

Inference via factorization in graphical models

$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$



Inference via factorization in graphical models



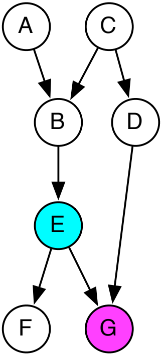
$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

$$P(E \wedge g)$$

$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC) P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$=$$

Inference via factorization in graphical models



$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

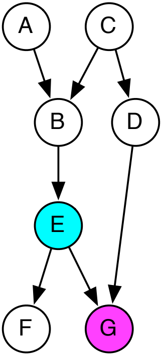
$$P(E \wedge g)$$

$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC)$$
$$P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$=$$

$$\left(\sum_D P(D \mid C)P(g \mid ED)\right)\Bigg)$$

Inference via factorization in graphical models



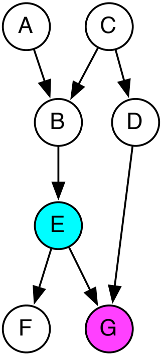
$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

$$P(E \wedge g)$$

$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC) \\ P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$= \quad \left(\sum_A P(A)P(B \mid AC)\right) \\ \left(\sum_D P(D \mid C)P(g \mid ED)\right)\Bigg)$$

Inference via factorization in graphical models



$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

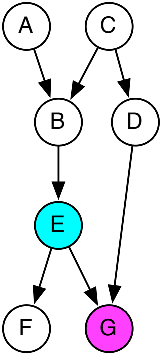
$$P(E \wedge g)$$

$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC)$$
$$P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$=$$

$$\sum_C \left(P(C) \left(\sum_A P(A)P(B \mid AC) \right) \right.$$
$$\left. \left(\sum_D P(D \mid C)P(g \mid ED) \right) \right)$$

Inference via factorization in graphical models



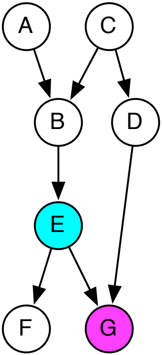
$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

$$P(E \wedge g)$$

$$= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC) P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED)$$

$$= \sum_B P(E \mid B) \sum_C \left(P(C) \left(\sum_A P(A)P(B \mid AC) \right) \left(\sum_D P(D \mid C)P(g \mid ED) \right) \right)$$

Inference via factorization in graphical models



$$P(E \mid g) = \frac{P(E \wedge g)}{\sum_E P(E \wedge g)}$$

$$\begin{aligned} P(E \wedge g) \\ &= \sum_F \sum_B \sum_C \sum_A \sum_D P(A)P(B \mid AC) \\ &\quad P(C)P(D \mid C)P(E \mid B)P(F \mid E)P(g \mid ED) \\ &= \left(\sum_F P(F \mid E)\right) \\ &\quad \sum_B P(E \mid B) \sum_C \left(P(C) \left(\sum_A P(A)P(B \mid AC)\right)\right. \\ &\quad \left.\left(\sum_D P(D \mid C)P(g \mid ED)\right)\right) \end{aligned}$$

Recursive Conditioning

- Computes sum (partition function) from outside in

Input:

- Context - assignment of values to variables
- Set of factors

Output: value of summing out other variables (partition function)

- Evaluate a factor as soon as all its variables are assigned
- Cache values already computed
- Recognize disconnected components
- Recursively branch on a variable

Variable Elimination and Recursive Conditioning

- Variable elimination is the dynamic programming variant of recursive conditioning.
- Recursive Conditioning is the search variant of variable elimination
- They do the same additions and multiplications.
- Complexity $O(nr^t)$, for n variables, range size r, and treewidth t.

Outline

Weighted Formula

A Weighted formula is a pair $\langle F, v \rangle$ where

- F a formula on parametrized random variables
- v number

Example:

$\langle X \neq Y \wedge likes(X, Y) \wedge rich(Y), 0.001 \rangle$
$\langle likes(X, X) \wedge rich(X), 0.7 \rangle$

...

Lifted Recursive Conditioning

LiftedRC(Context, WeightedFormulas)

- *Context* is a set of assignments to random variables and counts to assignments of instances of relations. e.g.:

$$\begin{aligned}\{\neg a,\ & \#_X f(X) \land g(X) = 7, \\ & \#_X f(X) \land \neg g(X) = 5, \\ & \#_X \neg f(X) \land g(X) = 18, \\ & \#_X \neg f(X) \land \neg g(X) = 0\}\end{aligned}$$

- *WeightedFormulas* is a set of weighted formulae, e.g.,

$$\begin{aligned}\{ & \langle \neg a \land \neg f(X) \land g(X), 0.1\rangle, \\ & \langle a \land \neg f(X) \land g(X), 0.2\rangle, \\ & \langle f(X) \land g(Y), 0.3\rangle, \\ & \langle f(X) \land h(X), 0.4\rangle\}\end{aligned}$$

Evaluating Weighted Formulae

Context:

$$\begin{aligned}\{\neg a, \quad & \#_X f(X) \wedge g(X) = 7, \\ & \#_X f(X) \wedge \neg g(X) = 5, \\ & \#_X \neg f(X) \wedge g(X) = 18, \\ & \#_X \neg f(X) \wedge \neg g(X) = 0\}\end{aligned}$$

WeightedFormulas:

$$\begin{aligned}\{ \quad & \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1\rangle, \\ & \langle a \wedge \neg f(X) \wedge g(X), 0.2\rangle, \\ & \langle f(X) \wedge g(Y), 0.3\rangle, \\ & \langle f(X) \wedge h(X), 0.4\rangle\}\end{aligned}$$

LiftedRC(*Context*, *WeightedFormulas*) returns:

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \wedge g(X) = 7,$$
$$\#_X f(X) \wedge \neg g(X) = 5,$$
$$\#_X \neg f(X) \wedge g(X) = 18,$$
$$\#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \quad \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1 \rangle,$$
$$\langle a \wedge \neg f(X) \wedge g(X), 0.2 \rangle,$$
$$\langle f(X) \wedge g(Y), 0.3 \rangle,$$
$$\langle f(X) \wedge h(X), 0.4 \rangle \}$$

LiftedRC(*Context*, *WeightedFormulas*) returns:

$$0.1^{18} *$$

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \wedge g(X) = 7, \\ \#_X f(X) \wedge \neg g(X) = 5, \\ \#_X \neg f(X) \wedge g(X) = 18, \\ \#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \quad \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1\rangle, \\ \langle a \wedge \neg f(X) \wedge g(X), 0.2\rangle, \\ \langle f(X) \wedge g(Y), 0.3\rangle, \\ \langle f(X) \wedge h(X), 0.4\rangle\}$$

$LiftedRC(Context, WeightedFormulas)$ returns:

$$0.1^{18} * 1 *$$

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \wedge g(X) = 7,$$
$$\#_X f(X) \wedge \neg g(X) = 5,$$
$$\#_X \neg f(X) \wedge g(X) = 18,$$
$$\#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \quad \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1 \rangle,$$
$$\langle a \wedge \neg f(X) \wedge g(X), 0.2 \rangle,$$
$$\langle f(X) \wedge g(Y), 0.3 \rangle,$$
$$\langle f(X) \wedge h(X), 0.4 \rangle\}$$

LiftedRC(*Context*, *WeightedFormulas*) returns:

$$0.1^{18} * 1 * 0.3^{12*}$$

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \wedge g(X) = 7,$$
$$\#_X f(X) \wedge \neg g(X) = 5,$$
$$\#_X \neg f(X) \wedge g(X) = 18,$$
$$\#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \quad \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1 \rangle,$$
$$\langle a \wedge \neg f(X) \wedge g(X), 0.2 \rangle,$$
$$\langle f(X) \wedge g(Y), 0.3 \rangle,$$
$$\langle f(X) \wedge h(X), 0.4 \rangle\}$$

$LiftedRC(Context, WeightedFormulas)$ returns:

$$0.1^{18} * 1 * 0.3^{12*25} *$$

Evaluating Weighted Formulae

Context:

$$\{\neg a, \quad \#_X f(X) \wedge g(X) = 7,$$
$$\#_X f(X) \wedge \neg g(X) = 5,$$
$$\#_X \neg f(X) \wedge g(X) = 18,$$
$$\#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas:

$$\{ \quad \langle \neg a \wedge \neg f(X) \wedge g(X), 0.1 \rangle,$$
$$\langle a \wedge \neg f(X) \wedge g(X), 0.2 \rangle,$$
$$\langle f(X) \wedge g(Y), 0.3 \rangle,$$
$$\langle f(X) \wedge h(X), 0.4 \rangle \}$$

$LiftedRC(Context, WeightedFormulas)$ returns:

$$0.1^{18} * 1 * 0.3^{12*25} * LiftedRC(Context, \{\langle f(X) \wedge h(X), 0.4 \rangle\})$$

Branching

Context:

$$\{\neg a,\ \#_X f(X) \wedge g(X) = 7,$$
$$\#_X f(X) \wedge \neg g(X) = 5,$$
$$\#_X \neg f(X) \wedge g(X) = 18,$$
$$\#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas: $\{\langle f(X) \wedge h(X), 0.4\rangle, \ldots\}$

Branching on H for the 7 "X" individuals s.th. $f(X) \wedge g(X)$:

$LiftedRC(Context, WeightedFormulas) =$

Branching

Context:

$$\begin{aligned}\{\neg a,\ & \#_X f(X) \wedge g(X) = 7, \\ & \#_X f(X) \wedge \neg g(X) = 5, \\ & \#_X \neg f(X) \wedge g(X) = 18, \\ & \#_X \neg f(X) \wedge \neg g(X) = 0\}\end{aligned}$$

WeightedFormulas: $\{\langle f(X) \wedge h(X), 0.4\rangle, \dots\}$
Branching on H for the 7 "X" individuals s.th. $f(X) \wedge g(X)$:
$LiftedRC(Context, WeightedFormulas) =$

$$\sum_{i=0}^{7} \binom{7}{i} LiftedRC(\{\neg a,\quad \begin{aligned} & \#_X f(X) \wedge g(X) \wedge h(X) = i, \\ & \#_X f(X) \wedge g(X) \wedge \neg h(X) = 7 - i, \\ & \#_X f(X) \wedge \neg g(X) = 5, \dots\}, \end{aligned}$$

$$WeightedFormulas)$$

Branching

Context:

$$\{\neg a,\ \#_X f(X) \wedge g(X) = 7,\\ \#_X f(X) \wedge \neg g(X) = 5,\\ \#_X \neg f(X) \wedge g(X) = 18,\\ \#_X \neg f(X) \wedge \neg g(X) = 0\}$$

WeightedFormulas: $\{\langle f(X) \wedge h(X), 0.4\rangle, \ldots\}$
Branching on H for the 7 "X" individuals s.th. $f(X) \wedge g(X)$:
$LiftedRC(Context, WeightedFormulas) =$

$$\sum_{i=0}^{7} \binom{7}{i} LiftedRC(\{\neg a,\ \#_X f(X) \wedge g(X) \wedge h(X) = i,\\ \#_X f(X) \wedge g(X) \wedge \neg h(X) = 7 - i,\\ \#_X f(X) \wedge \neg g(X) = 5, \ldots\},\\ WeightedFormulas)$$

Branching

Context:

$$\{\neg a,\ \#_X f(X) \land g(X) = 7,$$
$$\#_X f(X) \land \neg g(X) = 5,$$
$$\#_X \neg f(X) \land g(X) = 18,$$
$$\#_X \neg f(X) \land \neg g(X) = 0\}$$

WeightedFormulas: $\{\langle f(X) \land h(X), 0.4\rangle, \dots\}$
Branching on H for the 7 "X" individuals s.th. $f(X) \land g(X)$:
$LiftedRC(Context, WeightedFormulas) =$

$$\sum_{i=0}^{7} \binom{7}{i} LiftedRC(\{\neg a,\ \begin{array}{l} \#_X f(X) \land g(X) \land h(X) = i, \\ \#_X f(X) \land g(X) \land \neg h(X) = 7 - i, \\ \#_X f(X) \land \neg g(X) = 5, \dots\}, \end{array}$$
$$WeightedFormulas)$$

Branching

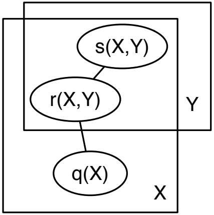
Context:

$$\begin{aligned}\{\neg a,\ & \#_X f(X) \wedge g(X) = 7, \\ & \#_X f(X) \wedge \neg g(X) = 5, \\ & \#_X \neg f(X) \wedge g(X) = 18, \\ & \#_X \neg f(X) \wedge \neg g(X) = 0\}\end{aligned}$$

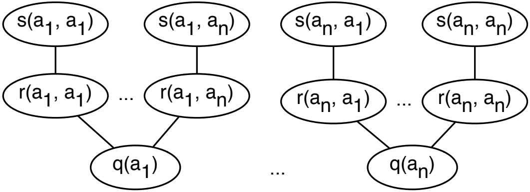
WeightedFormulas: $\{\langle f(X) \wedge h(X), 0.4\rangle, \ldots\}$
Branching on H for the 7 "X" individuals s.th. $f(X) \wedge g(X)$:
$LiftedRC(Context, WeightedFormulas) =$

$$\sum_{i=0}^{7} \binom{7}{i} LiftedRC(\{\neg a,\ \begin{aligned} & \#_X f(X) \wedge g(X) \wedge h(X) = i, \\ & \#_X f(X) \wedge g(X) \wedge \neg h(X) = 7 - i, \\ & \#_X f(X) \wedge \neg g(X) = 5, \ldots\}, \end{aligned}$$
$$WeightedFormulas)$$

Recognizing Disconnectedness



Relational Model

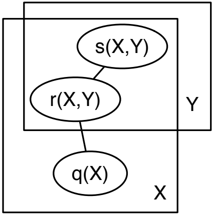


Grounding

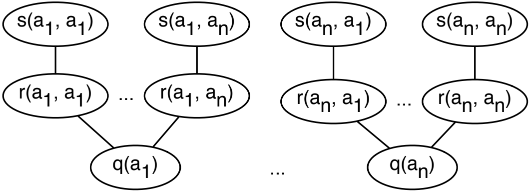
Weighted formulae *WeightedFormulas*:

$$\{ \langle \{ s(X, Y) \land r(X, Y) \}, t_1 \rangle$$
$$\langle \{ q(X) \land r(X, Y) \}, t_2 \rangle \}$$

Recognizing Disconnectedness



Relational Model



Grounding

Weighted formulae *WeightedFormulas*:

$$\{\langle\{s(X,Y)\wedge r(X,Y)\},t_1\rangle$$
$$\langle\{q(X)\wedge r(X,Y)\},t_2\rangle\}$$

$$\begin{aligned}&LiftedRC(Context, WeightedFormulas)\\ &\qquad = LiftedRC(Context, WeightedFormulas\{X/c\})^n\end{aligned}$$

...now we only have unary predicates

Observations and Queries

- Observations become the initial context.
  Observations can be ground or lifted.
- $$P(q|obs) = \frac{LiftedRC(q \wedge obs, WFs)}{LiftedRC(q \wedge obs, WFs) + LiftedRC(\neg q \wedge obs, WFs)}$$

  calls can share the cache
- "How many?" queries are also allowed

Complexity

As the population size n of undifferentiated individuals increases:

- If grounding is polynomial — instances must be disconnected — lifted inference is constant in n (taking r^n for real r)

Complexity

As the population size n of undifferentiated individuals increases:

- If grounding is polynomial — instances must be disconnected — lifted inference is constant in n (taking r^n for real r)
- Otherwise, for unary relations, grounding is exponential and lifted inference is polynomial.

Complexity

As the population size n of undifferentiated individuals increases:

- If grounding is polynomial — instances must be disconnected — lifted inference is constant in n (taking r^n for real r)
- Otherwise, for unary relations, grounding is exponential and lifted inference is polynomial.
- If non-unary relations become unary, above holds.

Complexity

As the population size n of undifferentiated individuals increases:

- If grounding is polynomial — instances must be disconnected — lifted inference is constant in n (taking r^n for real r)
- Otherwise, for unary relations, grounding is exponential and lifted inference is polynomial.
- If non-unary relations become unary, above holds.
- Otherwise, ground one individual from population, recurse. Sometimes this domain recursion is linear, but is typically exponential (as is grounding the population).

Complexity

As the population size n of undifferentiated individuals increases:

- If grounding is polynomial — instances must be disconnected — lifted inference is constant in n (taking r^n for real r)
- Otherwise, for unary relations, grounding is exponential and lifted inference is polynomial.
- If non-unary relations become unary, above holds.
- Otherwise, ground one individual from population, recurse. Sometimes this domain recursion is linear, but is typically exponential (as is grounding the population).

Always exponentially faster than grounding everything.

What we can and cannot lift

We can lift a model that consists just of

$$\langle\{f(X) \wedge g(Z)\}, \alpha_4\rangle$$

What we can and cannot lift

We can lift a model that consists just of

$$\langle\{f(X) \wedge g(Z)\}, \alpha_4\rangle$$

or just of

$$\langle\{f(X, Z) \wedge g(Y, Z)\}, \alpha_2\rangle$$

What we can and cannot lift

We can lift a model that consists just of

$$\langle \{f(X) \wedge g(Z)\}, \alpha_4 \rangle$$

or just of

$$\langle \{f(X, Z) \wedge g(Y, Z)\}, \alpha_2 \rangle$$

or just of

$$\langle \{f(X, Z) \wedge g(Y, Z) \wedge h(Y)\}, \alpha_3 \rangle$$

What we can and cannot lift

We can lift a model that consists just of

$$\langle\{f(X) \wedge g(Z)\}, \alpha_4\rangle$$

or just of

$$\langle\{f(X,Z) \wedge g(Y,Z)\}, \alpha_2\rangle$$

or just of

$$\langle\{f(X,Z) \wedge g(Y,Z) \wedge h(Y)\}, \alpha_3\rangle$$

We cannot lift (still exponential) a model that consists just of:

$$\langle\{f(X,Z) \wedge g(Y,Z) \wedge h(Y,W)\}, \alpha_3\rangle$$

What we can and cannot lift

We can lift a model that consists just of

$$\langle \{f(X) \wedge g(Z)\}, \alpha_4 \rangle$$

or just of

$$\langle \{f(X,Z) \wedge g(Y,Z)\}, \alpha_2 \rangle$$

or just of

$$\langle \{f(X,Z) \wedge g(Y,Z) \wedge h(Y)\}, \alpha_3 \rangle$$

We cannot lift (still exponential) a model that consists just of:

$$\langle \{f(X,Z) \wedge g(Y,Z) \wedge h(Y,W)\}, \alpha_3 \rangle$$

or

$$\langle \{f(X,Z) \wedge g(Y,Z) \wedge h(Y,X)\}, \alpha_3 \rangle$$

Compilation

- The computation reduces to products and sums
- The structure can be determined at compile time
- Orders of magnitude faster than lifted recursive conditioning
- Often abstracted as weighted model counting (WMC)

Take Home

- Lifted inference exploits symmetries ("for all")
- Instead of considering which individuals a predicate is true for, count how many individuals it is true for, and determine appropriate probabilities.
- Always exponentially better in the number of undifferentiated individuals than grounding everything.
- Open problem: finding a dichotomy of those problems we know we can lift and those we know it is impossible to lift.

Potential of Lifted Inference

- Lifting reduces complexity:

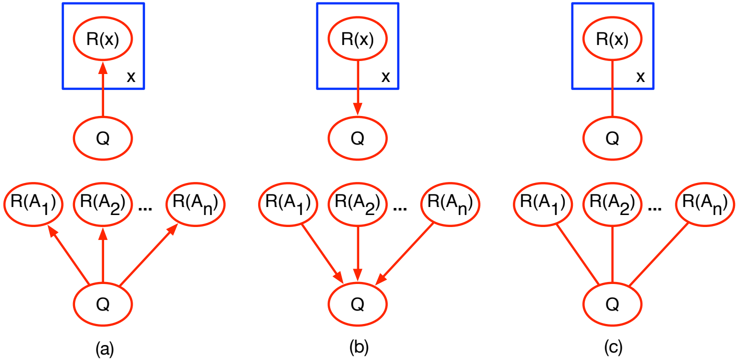
  polynomial $\longrightarrow$ *logarithmic*

  exponential $\longrightarrow$ *polynomial*

  in the population size of undifferentiated individuals compared to grounding
- We can now lift all unary relations, but we know we can't do all binary relations [Guy Van den Broeck, 2013].
  Always exponentially faster.
- Current most efficient algorithm compile to secondary representations. (E.g. Mehran Kazemi compiles to C++).
- Great potential for approximate inference

Outline

Three Elementary Models

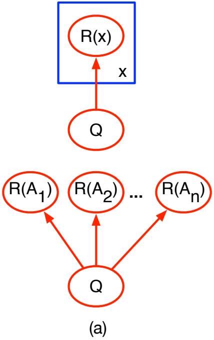


(a) Naïve Bayes

(b) (Relational) Logistic Regression

(c) Markov network

Independence Assumptions



- Naïve Bayes (a) and Markov network (c): $R(A_i)$ and $R(A_j)$
  - are independent given Q
  - are dependent not given Q.
- Directed model with aggregation (b): $R(A_i)$ and $R(A_j)$
  - are dependent given Q,
  - are independent not given Q.

Logistic Regression

Logistic Regression, write $R(a_i)$ as R_i:

$$P(Q|R_1, \ldots, R_n) = sigmoid(w_0 + w_1 R_1 + \cdots + w_n R_n)$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

Logistic Regression

Logistic Regression, write $R(a_i)$ as R_i:

$$P(Q|R_1,\ldots,R_n) = sigmoid(w_0 + w_1 R_1 + \cdots + w_n R_n)$$

$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

If all of the R_i are exchangeable $w_1,\ldots,w_n$ must all be the same:

$$P(Q|R_1,\ldots,R_n) = sigmoid(w_0 + w_1 \sum_i R_i))$$

Logistic Regression

Logistic Regression, write $R(a_i)$ as R_i:

$$P(Q|R_1, \ldots, R_n) = sigmoid(w_0 + w_1 R_1 + \cdots + w_n R_n)$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

If all of the R_i are exchangeable $w_1, \ldots, w_n$ must all be the same:

$$P(Q|R_1, \ldots, R_n) = sigmoid(w_0 + w_1 \sum_i R_i))$$

If we learn the parameters for $n = 10$ the prediction for $n = 20$ depends on how values R_i are represented numerically:

- If $True = 1$ and $False = 0$ then $P(Q|R_1, \ldots, R_n)$ depends on the number of R_i that are true.

Logistic Regression

Logistic Regression, write $R(a_i)$ as R_i:

$$P(Q|R_1,\ldots,R_n) = sigmoid(w_0 + w_1 R_1 + \cdots + w_n R_n)$$

$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

If all of the R_i are exchangeable $w_1,\ldots,w_n$ must all be the same:

$$P(Q|R_1,\ldots,R_n) = sigmoid(w_0 + w_1 \sum_i R_i))$$

If we learn the parameters for $n=10$ the prediction for $n=20$ depends on how values R_i are represented numerically:

- If $True = 1$ and $False = 0$ then $P(Q|R_1,\ldots,R_n)$ depends on the number of R_i that are true.
- If $True = 1$ and $False = -1$ then $P(Q|R_1,\ldots,R_n)$ depends on how many more of R_i are true than false.

Logistic Regression

Logistic Regression, write $R(a_i)$ as R_i:

$$P(Q|R_1, \ldots, R_n) = sigmoid(w_0 + w_1 R_1 + \cdots + w_n R_n)$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

If all of the R_i are exchangeable $w_1, \ldots, w_n$ must all be the same:

$$P(Q|R_1, \ldots, R_n) = sigmoid(w_0 + w_1 \sum_i R_i))$$

If we learn the parameters for $n = 10$ the prediction for $n = 20$ depends on how values R_i are represented numerically:

- If $True = 1$ and $False = 0$ then $P(Q|R_1, \ldots, R_n)$ depends on the number of R_i that are true.
- If $True = 1$ and $False = -1$ then $P(Q|R_1, \ldots, R_n)$ depends on how many more of R_i are true than false.
- If $True = 0$ and $False = -1$ then $P(Q|R_1, \ldots, R_n)$ depends on the number of R_i that are false.

Directed and Undirected models

- Weighted formula (WF): $\langle L, F, w \rangle$
  - L is a set of logical variables,
  - F is a logical formula: $\{$free logical variables in $F\} \subseteq L$
  - w is a real-valued weight.
- Instances of weighted formule obtained by assigning individuals to variables in L.

Directed and Undirected models

- Weighted formula (WF): $\langle L, F, w \rangle$
  - L is a set of logical variables,
  - F is a logical formula: $\{$free logical variables in $F\} \subseteq L$
  - w is a real-valued weight.
- Instances of weighted formule obtained by assigning individuals to variables in L.
- A world is an assignment of a value to each ground instance of each atom.
- Markov logic network (MLN): "undirected model" weighted formulae define measures on worlds.

Directed and Undirected models

- Weighted formula (WF): $\langle L, F, w \rangle$
  - L is a set of logical variables,
  - F is a logical formula: $\{$free logical variables in $F\} \subseteq L$
  - w is a real-valued weight.
- Instances of weighted formule obtained by assigning individuals to variables in L.
- A world is an assignment of a value to each ground instance of each atom.
- Markov logic network (MLN): "undirected model" weighted formulae define measures on worlds.
- Relational logistic regression (RLR): "directed model" weighted formulae define conditional probabilities.

Weighted formulae for conditionals $\rightarrow$ logistic regression

Weighted formulae:

$$\langle \{x\}, \mathit{funFor}(x), -5\rangle$$
$$\langle \{x,y\}, \mathit{funFor}(x) \wedge \mathit{friends}(x,y) \wedge \mathit{social}(y), 10\rangle$$
$$\langle \{x,y\}, \mathit{funFor}(x) \wedge \mathit{friends}(x,y) \wedge \neg \mathit{social}(y), -3\rangle$$

If *obs* includes observations for all $\mathit{friends}(x,y)$ and $\mathit{social}(y)$:

$$P(\mathit{funFor}(x) \mid \mathit{obs}) = \mathit{sigmoid}(-5 + 10 n_s(x) - 3 n_a(x))$$

$$n_s(x) = |\{y \mid \mathit{friends}(x,y) \wedge \mathit{social}(y)\}|$$

$$n_a(x) = |\{y \mid \mathit{friends}(x,y) \wedge \neg \mathit{social}(y)\}|$$

Weighted formulae for conditionals → logistic regression

Weighted formulae:

$$\langle \{x\}, \mathit{funFor}(x), -5 \rangle$$
$$\langle \{x, y\}, \mathit{funFor}(x) \wedge \mathit{friends}(x, y) \wedge \mathit{social}(y), 10 \rangle$$
$$\langle \{x, y\}, \mathit{funFor}(x) \wedge \mathit{friends}(x, y) \wedge \neg \mathit{social}(y), -3 \rangle$$

If *obs* includes observations for all $\mathit{friends}(x, y)$ and $\mathit{social}(y)$:

$$P(\mathit{funFor}(x) \mid \mathit{obs}) = \mathit{sigmoid}(-5 + 10 n_s(x) - 3 n_a(x))$$

$$n_s(x) = |\{y \mid \mathit{friends}(x, y) \wedge \mathit{social}(y)\}|$$

$$n_a(x) = |\{y \mid \mathit{friends}(x, y) \wedge \neg \mathit{social}(y)\}|$$

- Weighted formulae give arbitrary polynomials of counts.

Representation Issues

- Probabilities of directed model can be interpreted locally

Representation Issues

- Probabilities of directed model can be interpreted locally
- Directed models are modular — e.g., adding a dependent variable without side effects is straightforward, but impossible for MLNs
  [Buchman and Poole, AAAI 2015]

Representation Issues

- Probabilities of directed model can be interpreted locally
- Directed models are modular — e.g., adding a dependent variable without side effects is straightforward, but impossible for MLNs
  [Buchman and Poole, AAAI 2015]
- Directed models allow for pruning in inference.

Representation Issues

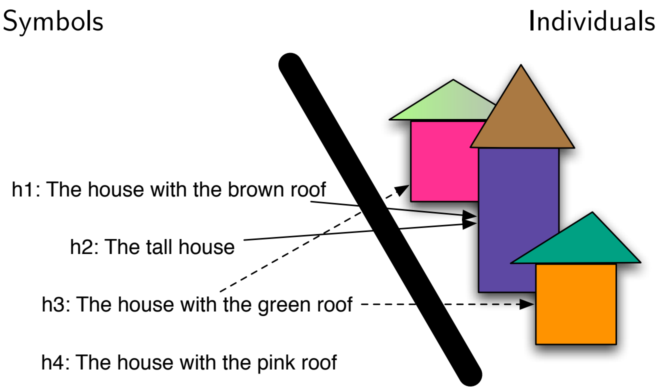
- Probabilities of directed model can be interpreted locally
- Directed models are modular — e.g., adding a dependent variable without side effects is straightforward, but impossible for MLNs
  [Buchman and Poole, AAAI 2015]
- Directed models allow for pruning in inference.
- Directed models require the structure of the conditional probabilities to be acyclic. Or maybe not...

Representation Issues

- Probabilities of directed model can be interpreted locally
- Directed models are modular — e.g., adding a dependent variable without side effects is straightforward, but impossible for MLNs
  [Buchman and Poole, AAAI 2015]
- Directed models allow for pruning in inference.
- Directed models require the structure of the conditional probabilities to be acyclic. Or maybe not...
- Noisy-or aggregation corresponds to logic programs.
  With layered relational logistic regression, can we get relational neural networks?

Outline

Correspondence Problem



c symbols and i individuals $\longrightarrow c^{i+1}$ correspondences

Clarity Principle

Clarity principle: probabilities must be over well-defined propositions.

- What if an individual doesn't exist?
  - $house(h4) \wedge roof\_colour(h4, pink) \wedge \neg exists(h4)$

Clarity Principle

Clarity principle: probabilities must be over well-defined propositions.

- What if an individual doesn't exist?
  - $house(h4) \wedge roof\_colour(h4, pink) \wedge \neg exists(h4)$

- What if more than one individual exists? Which one are we referring to?
  —In a house with three bedrooms, which is the second bedroom?

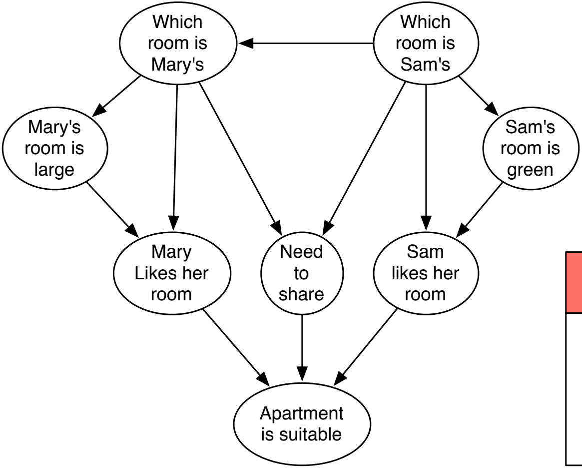
Role assignments

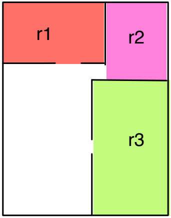
Hypothesis about what apartment Mary would like.

Whether Mary likes an apartment depends on:

- Whether there is a bedroom for daughter Sam
- Whether Sam's room is green
- Whether there is a bedroom for Mary
- Whether Mary's room is large
- Whether they share

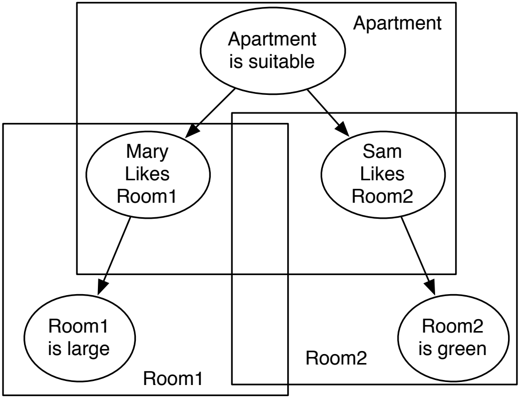
Bayesian Belief Network Representation

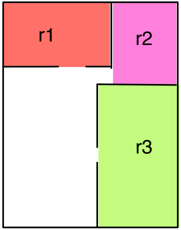




How can we condition on the observation of the apartment?

Naive Bayes representation





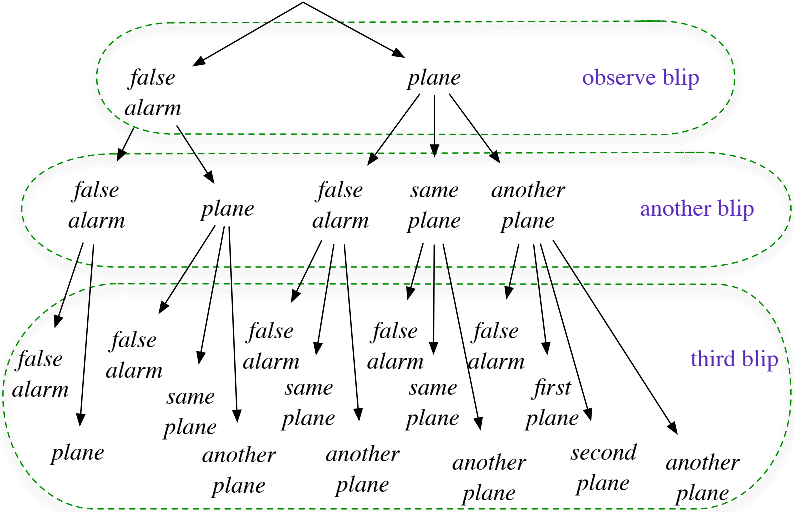
How do we specify that Mary chooses a room?
What about the case where they (have to) share?

Number and Existence Uncertainty

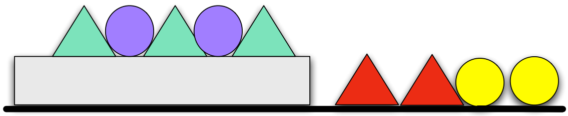
- PRMs (Pfeffer et al.), BLOG (Milch et al.): distribution over the number of individuals. For each number, reason about the correspondence.
- NP-BLOG (Carbonetto et al.): keep asking: is there one more?

  e.g., if you observe a radar blip, there are three hypotheses:
  - the blip was produced by plane you already hypothesized
  - the blip was produced by another plane
  - the blip wasn't produced by a plane

Existence Example



Observation Protocols

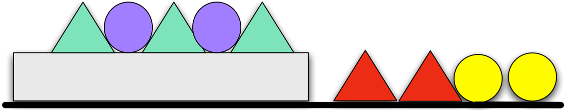


Observe a triangle and a circle touching. What is the probability the triangle is green?

$$P(green(x) \mid triangle(x) \wedge \exists y\ circle(y) \wedge touching(x, y))$$

The answer depends on how the x and y were chosen!

Protocol for Observing



$$P(green(x) \mid triangle(x) \wedge \exists y\ circle(y) \wedge touching(x, y))$$

$select(x)$	$select(y)$	$select(x, y)$
$select(y)$	$select(x)$	
$3/4$	$2/3$	$4/5$

Other Issues

- Probabilistic programming
- Much data is being published with respect to formal ontologies.
  How can probabilistic models interact with such data?
- We'd like to publish hypotheses that make probabilistic predictions so they interoperate with data.
- Identity uncertainty. Probability of equality.
  Do these citations refer to the same publication?
- To make decisions, probabilistic models need to interact with utility models.
- Representing actions, time,...

Conclusion

- The field of "statistical relational AI" studies how to combine first-order logic and probabilistic reasoning.

Challenges

- Representation: heuristically and epistemologically adequate representations for probabilistic models + observations (+ causation + actions + utilities + ontologies)
- Inference: exploit structure + exchangeability
  compute posterior probabilities (or optimal actions) quickly enough to be useful
- Learning: find best hypotheses conditioned on all observations
  ....just inference?

Age of Relations (100 years later)

What is now required is to give the greatest possible development to mathematical logic, to allow to the full the importance of relations, and then to found upon this secure basis a new philosophical logic, which may hope to borrow some of the exactitude and certainty of its mathematical foundation. If this can be successfully accomplished, there is every reason to hope that the near future will be as great an epoch in pure philosophy as the immediate past has been in the principles of mathematics. Great triumphs inspire great hopes; and pure thought may achieve, within our generation, such results as will place our time, in this respect, on a level with the greatest age of Greece.

– Bertrand Russell [1917]

AI: computational agents that act intelligently