Learning Objectives

At the end of the class you should be able to:

- describe the mapping between relational probabilistic models and their groundings
- read plate notation
- build a relational probabilistic model for a domain

Relational Probabilistic Models

- flat or modular or hierarchical
- explicit states or features or individuals and relations
- static or finite stage or indefinite stage or infinite stage
- fully observable or partially observable
- deterministic or stochastic dynamics
- goals or complex preferences
- single agent or multiple agents
- knowledge is given or knowledge is learned
- perfect rationality or bounded rationality

Relational Probabilistic Models

Often we want random variables for combinations of individual in populations

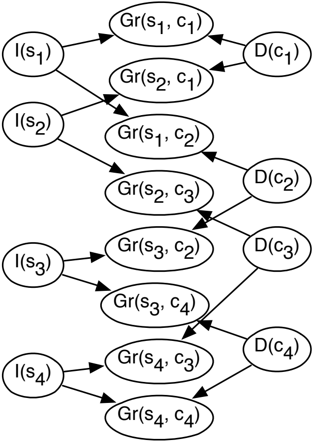
- build a probabilistic model before knowing the individuals
- learn the model for one set of individuals
- apply the model to new individuals
- allow complex relationships between individuals

Example: Predicting Relations

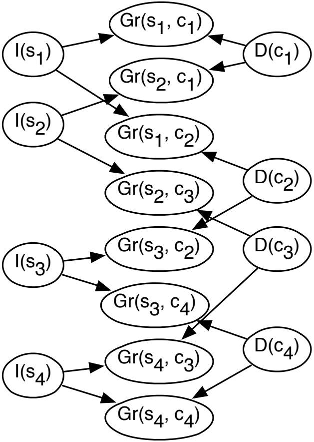
Student	*Course*	*Grade*
s_1	c_1	A
s_2	c_1	C
s_1	c_2	B
s_2	c_3	B
s_3	c_2	B
s_4	c_3	B
s_3	c_4	?
s_4	c_4	?

- Students s_3 and s_4 have the same averages, on courses with the same averages. Why should we make different predictions?
- How can we make predictions when the values of properties *Student* and *Course* are individuals?

From Relations to Belief Networks



From Relations to Belief Networks

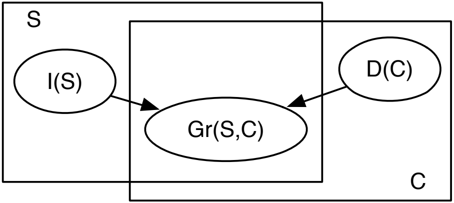


$I(S)$	$D(C)$	$Gr(S,C)$		
		A	B	C
true	*true*	0.5	0.4	0.1
true	*false*	0.9	0.09	0.01
false	*true*	0.01	0.1	0.9
false	*false*	0.1	0.4	0.5

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

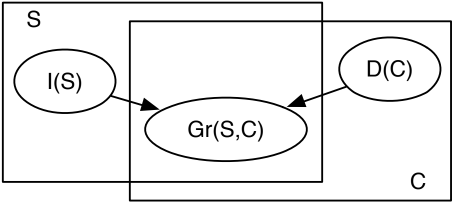
"parameter sharing"

Plate Notation



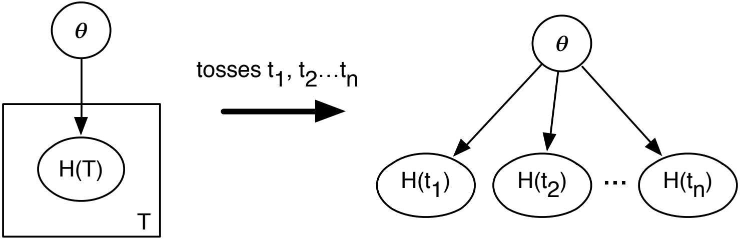
- S is a logical variable representing students
- C is a logical variable representing courses
- the set of all individuals of some type is called a population
- $I(S)$, $Gr(S,C)$, $D(C)$ are parametrized random variables

Plate Notation



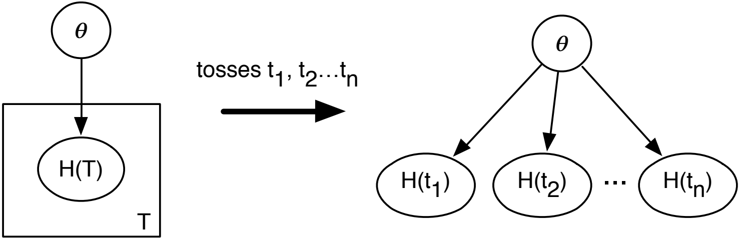
- S is a logical variable representing students
- C is a logical variable representing courses
- the set of all individuals of some type is called a population
- $I(S)$, $Gr(S, C)$, $D(C)$ are parametrized random variables
- for every student s, there is a random variable $I(s)$
- for every course c, there is a random variable $D(c)$
- for every student s and course c pair there is a random variable $Gr(s, c)$
- all instances share the same structure and parameters

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

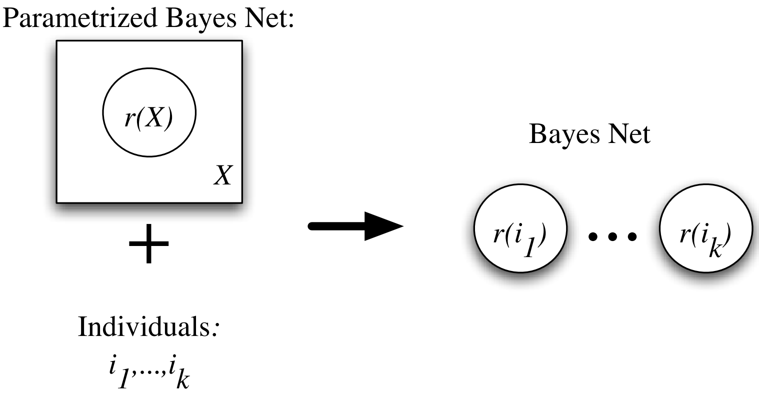


- T is a logical variable representing tosses of a thumb tack
- $H(t)$ is a Boolean variable that is true if toss t is heads.
- θ is a random variable representing the probability of heads.
- Range of θ is $\{0.0, 0.01, 0.02, \ldots, 0.99, 1.0\}$ or interval $[0, 1]$.
- $P(H(t_i){=}true|\theta{=}p) =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

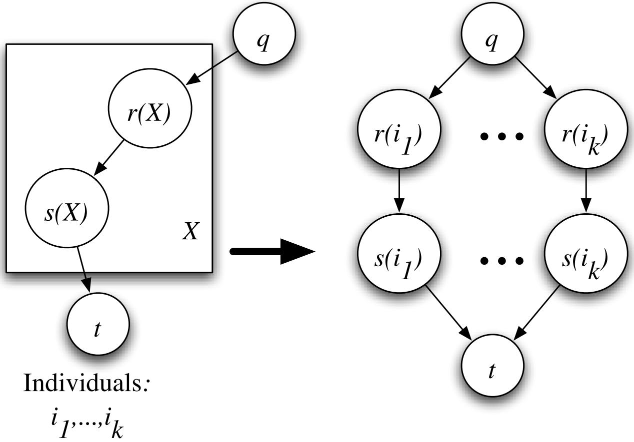
Parametrized belief networks

- Allow random variables to be parametrized. $interested(X)$
- Parameters correspond to logical variables. X
  Parameters 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. $interested(p_1) \dots interested(p_{1000000})$
- Instances are independent (but can have common ancestors and descendants).

Parametrized Bayesian networks / Plates

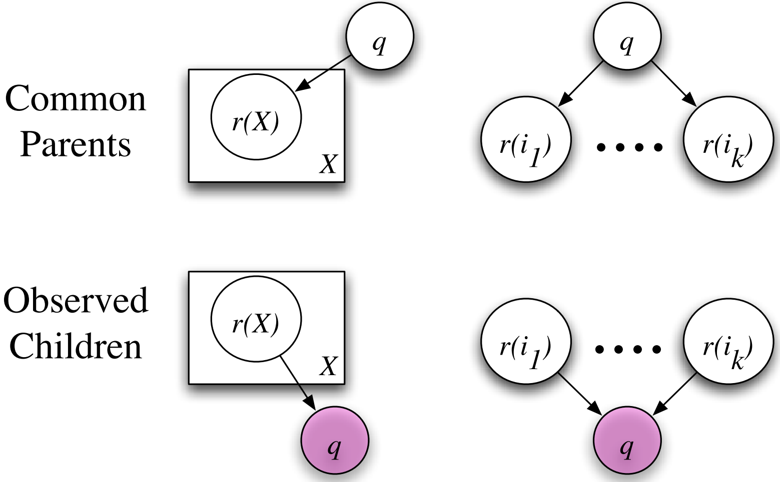


Parametrized Bayesian networks / Plates (2)

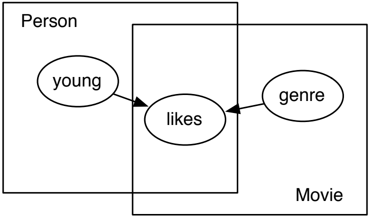


Creating Dependencies

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

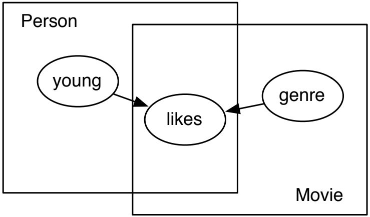


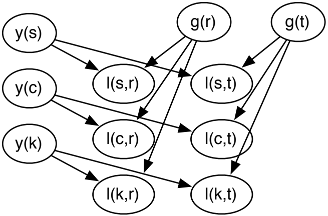
Overlapping plates



Relations: $likes(P, M)$, $young(P)$, $genre(M)$
likes is Boolean, *young* is Boolean,
genre has range $\{action, romance, family\}$

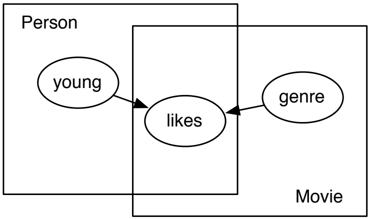
Overlapping plates





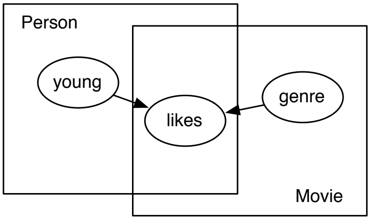
Relations: *likes*(P, M), *young*(P), *genre*(M)
likes is Boolean, *young* is Boolean,
genre has range $\{$*action*, *romance*, *family*$\}$
Three people: sam (s), chris (c), kim (k)
Two movies: rango (r), terminator (t)

Overlapping plates



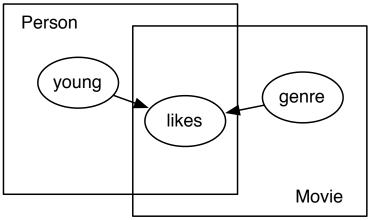
- Relations: $likes(P, M)$, $young(P)$, $genre(M)$
- $likes$ is Boolean, $young$ is Boolean, $genre$ has range $\{action, romance, family\}$
- If there are 1000 people and 100 movies,
  Grounding contains:
  random variables

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

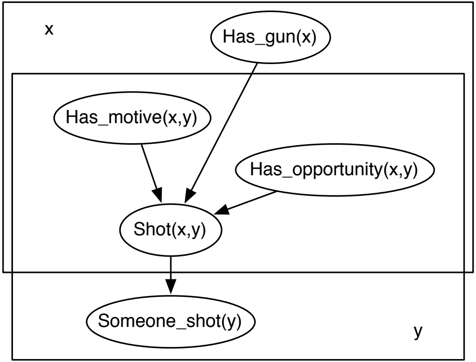


- Relations: $likes(P, M)$, $young(P)$, $genre(M)$
- *likes* is Boolean, *young* is Boolean, *genre* has range $\{action, romance, family\}$
- If there are 1000 people and 100 movies,
  Grounding contains: 100,000 likes + 1,000 age + 100 genre = 101,100 random variables
- How many numbers need to be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

Representing Conditional Probabilities

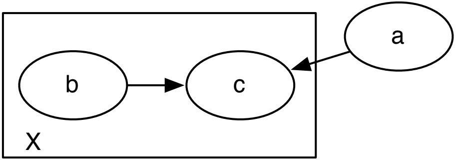
- $P(likes(P, M)|young(P), genre(M))$ — parameter sharing — individuals share probability parameters.
- $P(happy(X)|friend(X, Y), mean(Y))$ — needs aggregation — $happy(a)$ depends on an unbounded number of parents.
- There can be more structure about the individuals. . .

Example: Aggregation



Exercise #1

For the relational probabilistic model:



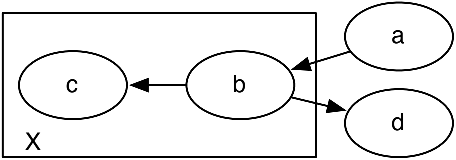
Suppose the the population of X is n and all variables are Boolean.

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

(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.

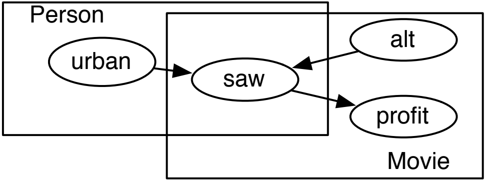
(a) Which of the conditional probabilities cannot be defined as a table?

(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:



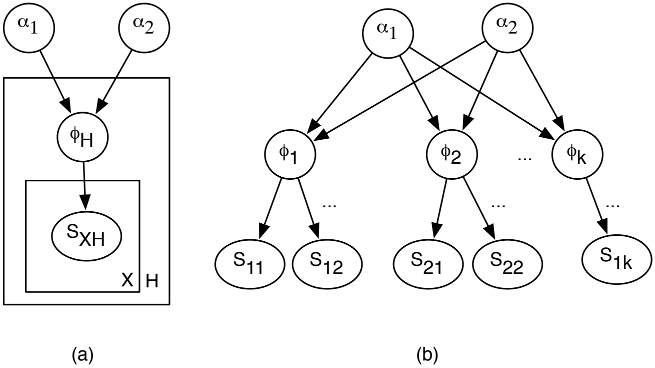
Suppose the population of *Person* is n and the population of *Movie* is m, and all variables are Boolean.

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

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

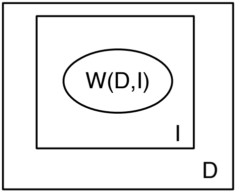
Hierarchical Bayesian Model

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



Example: Language Models

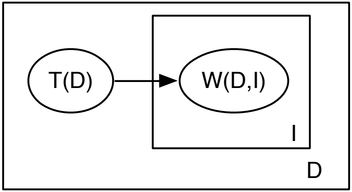
Unigram Model:



- 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

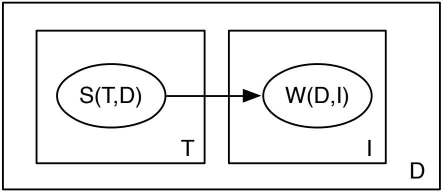
Topic Mixture:



- 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

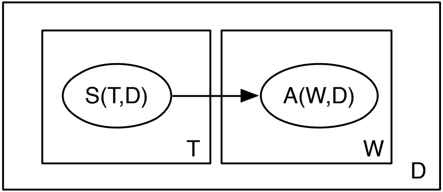
Mixture of topics, bag of words (unigram):



- 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: Language Models

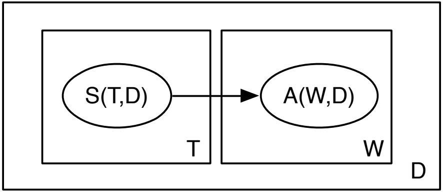
Mixture of topics, set of words:



- 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.

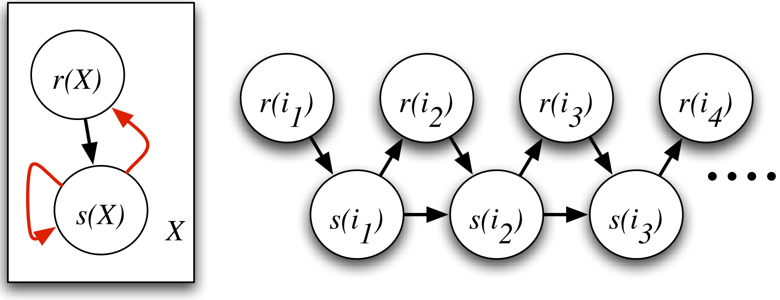
Example: Language Models

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

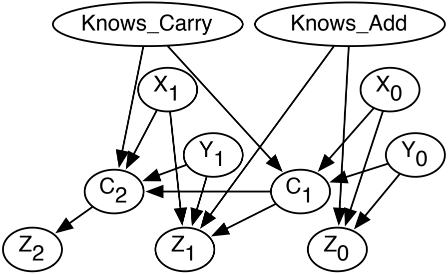


Predicting students errors

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

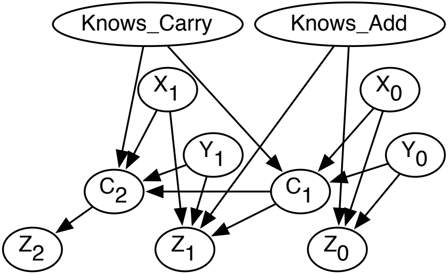
Predicting students errors

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



Predicting students errors

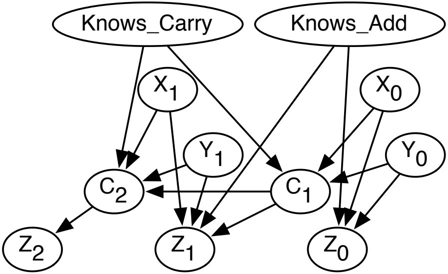
$$
\begin{array}{rccc}
 & & x_2 & x_1 \\
+ & & y_2 & y_1 \\
\hline
 & z_3 & z_2 & z_1
\end{array}
$$



- What if there were multiple digits

Predicting students errors

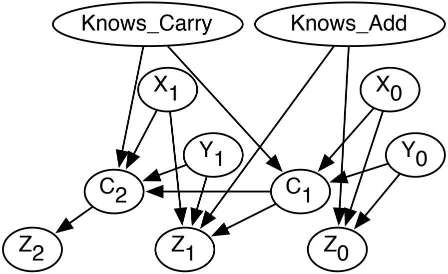
$$\begin{array}{cccc} & & x_2 & x_1 \\ + & & y_2 & y_1 \\ \hline & z_3 & z_2 & z_1 \end{array}$$



- What if there were multiple digits, problems

Predicting students errors

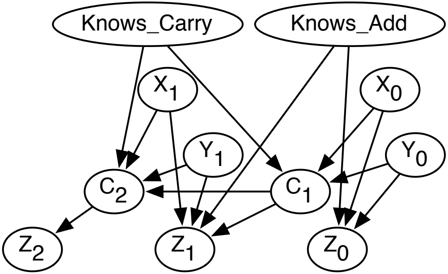
$$\begin{array}{rccc} & & x_2 & x_1 \\ + & & y_2 & y_1 \\ \hline & z_3 & z_2 & z_1 \end{array}$$



- What if there were multiple digits, problems, students

Predicting students errors

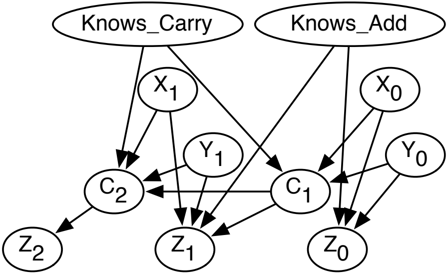
$$\begin{array}{cccc} & & x_2 & x_1 \\ + & & y_2 & y_1 \\ \hline & z_3 & z_2 & z_1 \end{array}$$



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

Predicting students errors

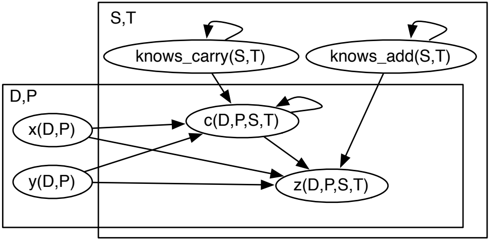
$$
\begin{array}{cccc}
 & & x_2 & x_1 \\
+ & & y_2 & y_1 \\
\hline
 & z_3 & z_2 & z_1
\end{array}
$$



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

Multi-digit addition with parametrized BNs / plates

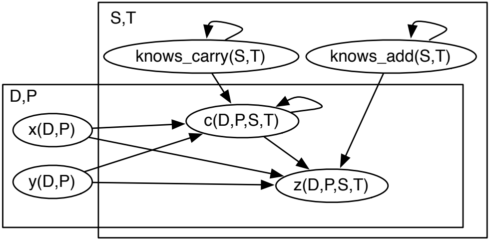
$$\begin{array}{rcccc} & x_{j_x} & \cdots & x_2 & x_1 \\ + & y_{j_y} & \cdots & y_2 & y_1 \\ \hline & z_{j_z} & \cdots & z_2 & z_1 \end{array}$$



- Parametrized Random Variables:

Multi-digit addition with parametrized BNs / plates

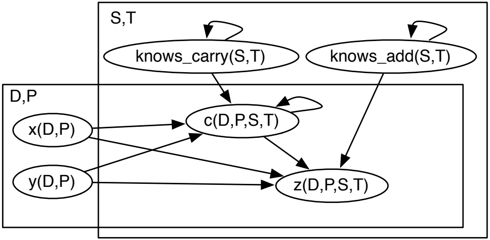
$$\begin{array}{cccccc} & x_{j_x} & \cdots & x_2 & x_1 \\ + & y_{j_y} & \cdots & y_2 & y_1 \\ \hline & z_{j_z} & \cdots & z_2 & z_1 \end{array}$$



- Parametrized Random Variables: $x(D,P)$, $y(D,P)$, $knows\_carry(S,T)$, $knows\_add(S,T)$, $c(D,P,S,T)$, $z(D,P,S,T)$
- Logical variables:

Multi-digit addition with parametrized BNs / plates

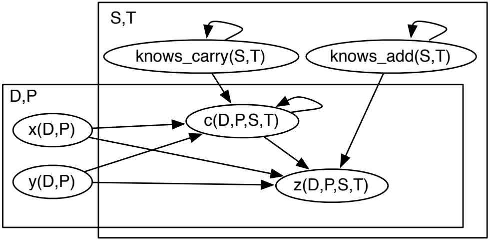
$$
\begin{array}{rcccc}
 & x_{j_x} & \cdots & x_2 & x_1 \\
+ & y_{j_y} & \cdots & y_2 & y_1 \\
\hline
 & z_{j_z} & \cdots & z_2 & z_1
\end{array}
$$



- Parametrized Random Variables: $x(D,P)$, $y(D,P)$, $knows\_carry(S,T)$, $knows\_add(S,T)$, $c(D,P,S,T)$, $z(D,P,S,T)$
- Logical variables: digit D, problem P, student S, time T.
- Random variables:

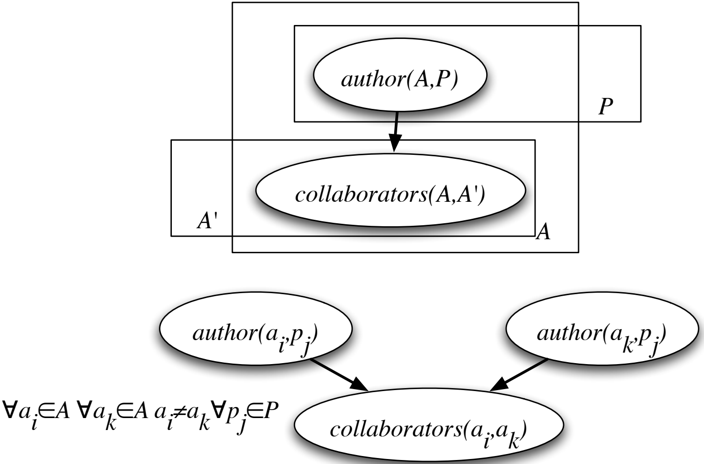
Multi-digit addition with parametrized BNs / plates

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



- Parametrized Random Variables: $x(D,P)$, $y(D,P)$, $knows\_carry(S,T)$, $knows\_add(S,T)$, $c(D,P,S,T)$, $z(D,P,S,T)$
- Logical variables: digit D, problem P, student S, time T.
- Random variables: There is a random variable for each assignment of a value to D and a value to P in $x(D,P)$....

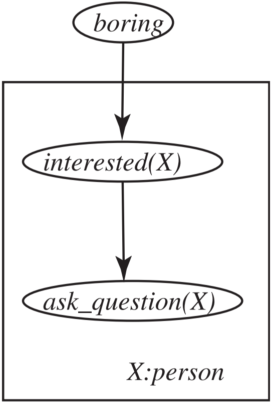
Creating Dependencies: Relational Structure



Lifted Inference

- Idea: treat those individuals about which you have the same information as a block; just count them.
- Potential to be exponentially faster in the number of non-differentialed individuals.
- Relies on knowing the number of individuals (the population size).

Example parametrized belief network



$P(boring)$
$\forall X \; P(interested(X)|boring)$
$\forall X \; P(ask\_question(X)|interested(X))$

First-order probabilistic inference

$$
\begin{tikzcd}
\text{Parametrized Belief Network} \arrow[r, "\text{FOVE}"] \arrow[d, "\text{ground}"] & \text{Parametrized Posterior} \arrow[d, "\text{ground}"] \\
\text{Belief Network} \arrow[r, "\text{VE}"] & \text{Posterior}
\end{tikzcd}
$$

Independent Choice Logic

- A language for first-order probabilistic models.
- Idea : combine logic and probability, where all uncertainty in handled in terms of Bayesian decision theory, and a logic program specifies consequences of choices.
- Parametrized random variables are represented as logical atoms, and plates correspond to logical variables.

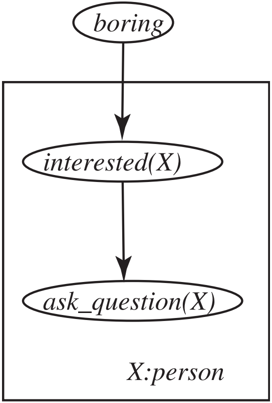
Parametric Factors

A parametric factor is a triple $\langle C, V, t \rangle$ where

- C is a set of inequality constraints on parameters,
- V is a set of parametrized random variables
- t is a table representing a factor from the random variables to the non-negative reals.

$$\left\langle \{X \neq sue\}, \{interested(X), boring\}, \begin{array}{|ll|l|} \hline interested & boring & Val \\ \hline yes & yes & 0.001 \\ yes & no & 0.01 \\ & \dots & \\ \hline \end{array} \right\rangle$$

Removing a parameter when summing



n people

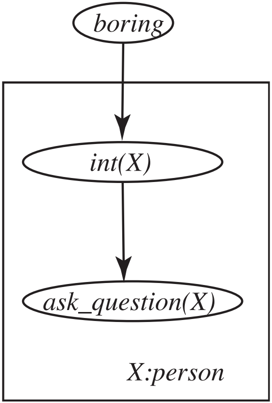
we observe no questions

Eliminate *interested*:

$$\langle \{\}, \{boring, interested(X)\}, t_1 \rangle$$
$$\langle \{\}, \{interested(X)\}, t_2 \rangle$$
$$\downarrow$$
$$\langle \{\}, \{boring\}, (t_1 \times t_2)^n \rangle$$

$(t_1 \times t_2)^n$ is computed pointwise; we can compute it in time $O(\log n)$.

Counting Elimination



$|people| = n$

[de Salvo Braz et al. 2007] and [Milch et al. 08]

Eliminate *boring*:

VE: factor on $\{int(p_1), \ldots, int(p_n)\}$

Size is $O(d^n)$ where d is size of range of interested.

Exchangeable: only the number of interested individuals matters.

Counting Formula:

#interested	Value
0	v_0
1	v_1
...	...
n	v_n

Complexity: $O(n^{d-1})$.

Potential of Lifted Inference

- Reduce complexity:

  polynomial $\longrightarrow$ *logarithmic*

  exponential $\longrightarrow$ *polynomial*

- We need a representation for the intermediate (lifted) factors that is closed under multiplication and summing out (lifted) variables.
- Still an open research problem.

Independent Choice Logic

- An alternative is a set of ground atomic formulas.
  $\mathcal{C}$, the choice space is a set of disjoint alternatives.
- $\mathcal{F}$, the facts is a logic program that gives consequences of choices.
- 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 independent.

Meaningless Example: Semantics

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

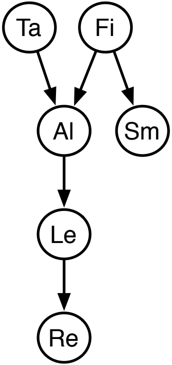
$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$
$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$

		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	$\sim e$	$P(w_4) = 0.05$
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	$P(w_6) = 0.02$

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

Belief Networks, Decision trees and ICL rules

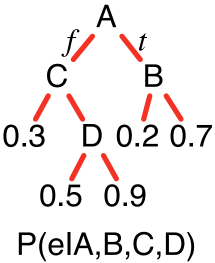
- There is a local mapping from belief networks into ICL.



prob $ta : 0.02.$
prob $fire : 0.01.$
$alarm \leftarrow ta \wedge fire \wedge atf.$
$alarm \leftarrow \sim ta \wedge fire \wedge antf.$
$alarm \leftarrow ta \wedge \sim fire \wedge atnf.$
$alarm \leftarrow \sim ta \wedge \sim fire \wedge antnf.$
prob $atf : 0.5.$
prob $antf : 0.99.$
prob $atnf : 0.85.$
prob $antnf : 0.0001.$
$smoke \leftarrow fire \wedge sf.$
prob $sf : 0.9.$
$smoke \leftarrow \sim fire \wedge snf.$
prob $snf : 0.01.$

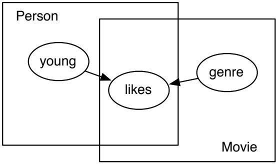
Belief Networks, Decision trees and ICL rules

- Rules can represent decision tree with probabilities:



$$
\begin{array}{ll}
e \leftarrow a \wedge b \wedge h_1. & P_0(h_1) = 0.7 \\
e \leftarrow a \wedge \sim b \wedge h_2. & P_0(h_2) = 0.2 \\
e \leftarrow \sim a \wedge c \wedge d \wedge h_3. & P_0(h_3) = 0.9 \\
e \leftarrow \sim a \wedge c \wedge \sim d \wedge h_4. & P_0(h_4) = 0.5 \\
e \leftarrow \sim a \wedge \sim c \wedge h_5. & P_0(h_5) = 0.3
\end{array}
$$

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

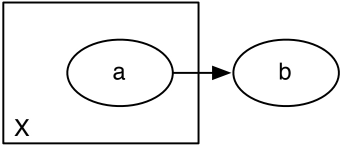
prob $lny(P, M, action) : 0.2.$

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

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

Aggregation

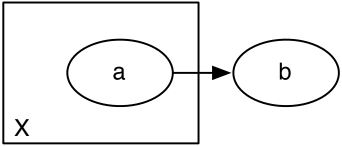
The relational probabilistic model:



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.

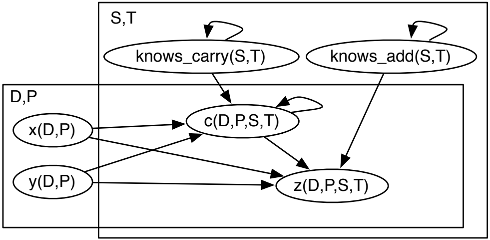
- If $a(c)$ is observed for each individual c:

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

  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

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

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

Alternatives:

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

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

Learning Relational Models with Hidden Variables

User	Item	Date	Rating
Sam	Terminator	2009-03-22	5
Sam	Rango	2011-03-22	4
Sam	The Holiday	2010-12-25	1
Chris	The Holiday	2010-12-25	4
...	...	...	

Netflix: 500,000 users, 17,000 movies, 100,000,000 ratings.

Learning Relational Models with Hidden Variables

User	Item	Date	Rating
Sam	Terminator	2009-03-22	5
Sam	Rango	2011-03-22	4
Sam	The Holiday	2010-12-25	1
Chris	The Holiday	2010-12-25	4
...	...	...	

Netflix: 500,000 users, 17,000 movies, 100,000,000 ratings.
r_{ui} = rating of user u on item i
$\widehat{r_{ui}}$ = predicted rating of user u on item i
D = set of (u, i, r) tuples in the training set (ignoring Date)
Sum squares error:

$$\sum_{(u,i,r)\in D} (\widehat{r_{ui}} - r)^2$$

Learning Relational Models with Hidden Variables

- Predict same for all ratings: $\widehat{r_{ui}} = \mu$

Learning Relational Models with Hidden Variables

- Predict same for all ratings: $\widehat{r_{ui}} = \mu$
- Adjust for each user and item: $\widehat{r_{ui}} = \mu + b_i + c_u$

Learning Relational Models with Hidden Variables

- Predict same for all ratings: $\widehat{r_{ui}} = \mu$
- Adjust for each user and item: $\widehat{r_{ui}} = \mu + b_i + c_u$
- One hidden feature: f_i for each item and g_u for each user

$$\widehat{r_{ui}} = \mu + b_i + c_u + f_i g_u$$

Learning Relational Models with Hidden Variables

- Predict same for all ratings: $\widehat{r_{ui}} = \mu$
- Adjust for each user and item: $\widehat{r_{ui}} = \mu + b_i + c_u$
- One hidden feature: f_i for each item and g_u for each user

$$\widehat{r_{ui}} = \mu + b_i + c_u + f_i g_u$$

- k hidden features:

$$\widehat{r_{ui}} = \mu + b_i + c_u + \sum_k f_{ik} g_{ku}$$

Learning Relational Models with Hidden Variables

- Predict same for all ratings: $\widehat{r_{ui}} = \mu$
- Adjust for each user and item: $\widehat{r_{ui}} = \mu + b_i + c_u$
- One hidden feature: f_i for each item and g_u for each user

$$\widehat{r_{ui}} = \mu + b_i + c_u + f_i g_u$$

- k hidden features:

$$\widehat{r_{ui}} = \mu + b_i + c_u + \sum_k f_{ik} g_{ku}$$

- Regularize

$$\begin{aligned} \mathit{minimize} \sum_{(u,i)\in K} & \left(\mu + b_i + c_u + \sum_k f_{ik} g_{ku} - r_{ui}\right)^2 \\ & + \lambda\left(b_i^2 + c_u^2 + \sum_k f_{ik}^2 + g_{ku}^2\right) \end{aligned}$$

Parameter Learning using Gradient Descent

$\mu \leftarrow$ average rating
assign $f[i,k]$, $g[k,u]$ randomly
assign $b[i], c[u]$ arbitrarily
repeat:
    for each $(u,i,r) \in D$:
        $e \leftarrow \mu + b[i] + c[u] + \sum_k f[i,k] * g[k,u] - r$
        $b[i] \leftarrow b[i] - \eta * e - \eta * \lambda * b[i]$
        $c[u] \leftarrow c[u] - \eta * e - \eta * \lambda * c[u]$
        for each feature k:
            $f[i,k] \leftarrow f[i,k] - \eta * e * g[k,u] - \eta * \lambda * f[i,k]$
            $g[k,u] \leftarrow g[k,u] - \eta * e * f[i,k] - \eta * \lambda * g[k,u]$