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. The code is mostly modular. . . , with modules for dealing with different kinds of objects and modules generalizing across many kinds of 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 B. Baum, What is Thought? [2004]

Lecture 17.3

Topics:

- mapping between relational probabilistic models and their groundings
- 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

Crowdsourcing

Consider crowdsourcing (such as Amazon mechanical Turk) to pay people to answer yes/no questions to determine what is true.

User	*Question*	*Answer*
u_1	q_1	*Yes*
u_2	q_1	*No*
u_1	q_2	*No*
u_3	q_2	*No*
u_2	q_2	*Yes*
u_3	q_3	*No*
$\dots$	$\dots$	$\dots$

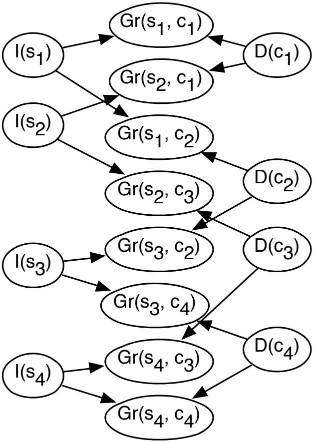
- How do you determine what is true from the responses?
- Truth: Majority vote? $> 90\%$?
- How confident should you be in the predictions?
- How do you determine who to pay?
- What about users guessing at random just to get paid?

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



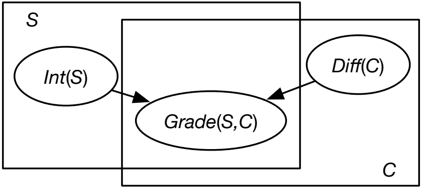
$I(S)$	$D(C)$	$Gr(S,C)$		
		A	B	C
true	*true*	0.5	0.4	0.1
true	*false*	0.9	0.09	0.01
false	*true*	0.01	0.1	0.9
false	*false*	0.1	0.4	0.5

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

"parameter sharing"

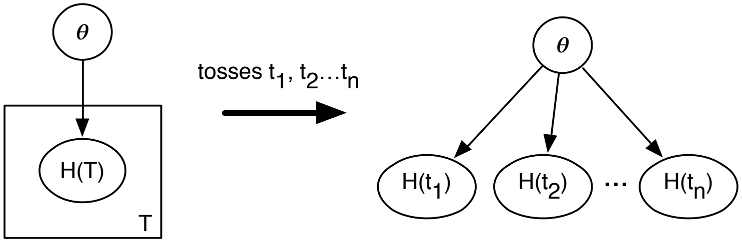
See `relnProbModels` in `AIPython.org`

Plate Notation



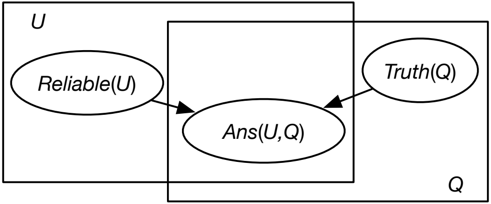
- S is a logical variable representing students
- C is a logical variable representing courses
- the set of all individuals of some type is called a population
- $Int(S)$, $Grade(S, C)$, $Diff(C)$ are parametrized random variables
- for every student s, there is a random variable $Int(s)$
- for every course c, there is a random variable $Diff(c)$
- for every student s and course c pair there is a random variable $Grade(s, c)$
- 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.
- Domain 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.

Crowdsourcing

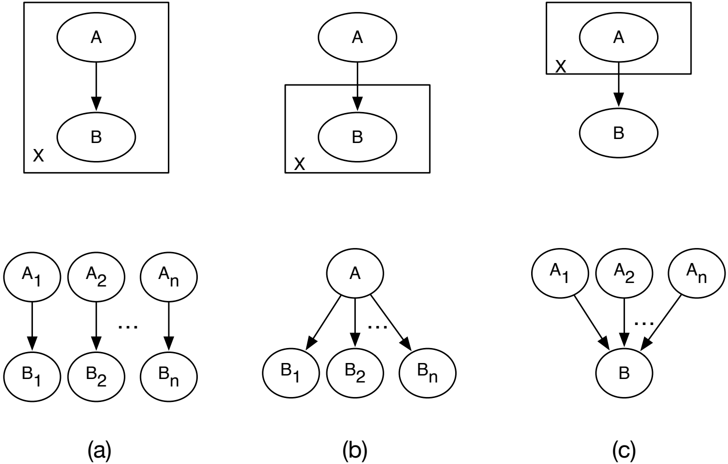


- Q is a logical variable representing questions
- U is a logical variable representing users who answer questions
- $Ans(U, Q)$ is the answer (yes/no) given by U to question Q
- $Truth(Q)$ represents whether Q is true or false
- $Reliable(U)$ represents how reliable U is (or maybe is just guessing). The more reliable the more likely the answer corresponds to the truth.
- It is unlikely that unreliable users will keep guessing the same answer.

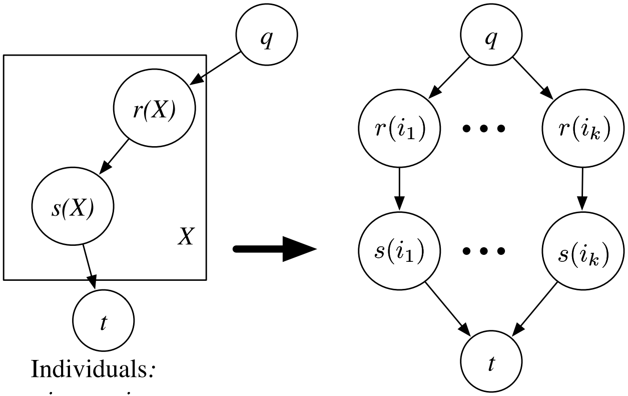
Parametrized belief networks

- 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).

Plates and the Grounding: Interaction of arcs and plates



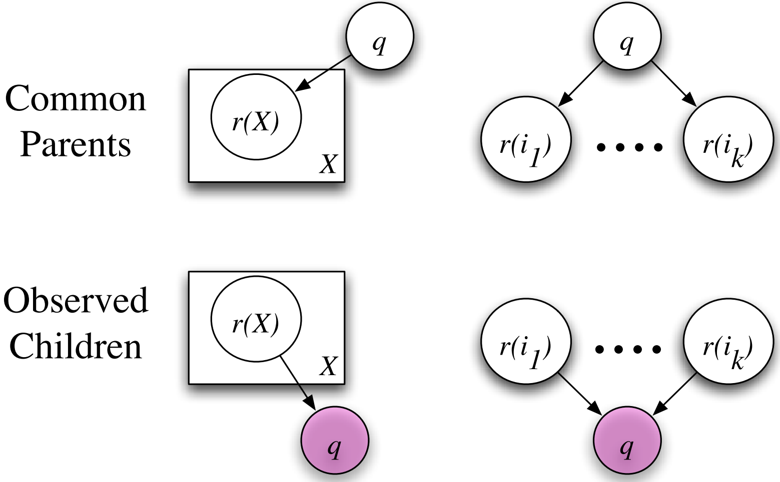
The population of logical variable X is $\{x_1, \ldots, x_n\}$
Random variable $A(x_i)$ is written as A_i
What independencies hold in (a), (b), (c)?



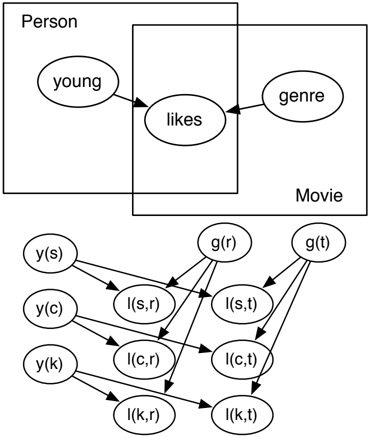
Individuals: $i_1, \ldots, i_k$

Creating Dependencies

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

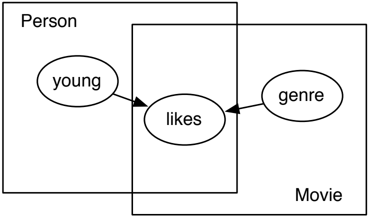


Overlapping plates



Relations: $likes(P, M)$, $young(P)$, $genre(M)$
likes is Boolean, *young* is Boolean,
genre has domain $\{action, romance, family\}$
Three people: sam (s), chris (c), kim (k)

Overlapping plates

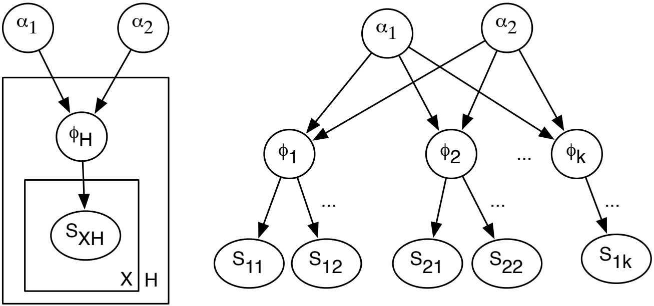


- Relations: *likes*(P, M), *young*(P), *genre*(M)
- *likes* is Boolean, *young* is Boolean, *genre* has domain $\{action, romance, family\}$
- If there are 1000 people and 100 movies,
  Grounding contains: 100,000 likes + 1,000 age + 100 genre
  = 101,100 random variables
- How many numbers need to 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. Also called "weight sharing" or "convolutional"
- $P(happy(X) | friend(X, Y), mean(Y))$ — needs aggregation — $happy(a)$ depends on an unbounded number of parents.
- There can be more structure about the individuals...

Example: 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?



(a) (b)

Aggregation

- When an arc comes out of a plate, the child has an unbounded number of parents $\longrightarrow$ aggregation.
- The other cases are the same as in a belief network and can use: table, rules, decision tree, logistic regression, neural network....
- Aggregation requires a method to represent the conditional probability given an unbounded number of parents in a finite way.

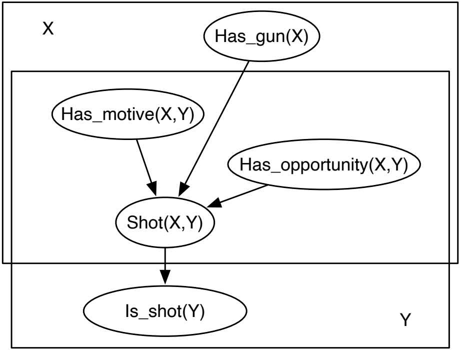
Standard Aggregators

- A probabilistic logic program, such as $b \leftarrow (\exists X \; a(X) \land n(X)) \lor n_0$ uses "noisy or" as the aggregator.
- The model can be specified using weighted logical formulae, extended to first-order logic: "relational logistic regression".
- Standard database aggregators such as average, sum or max of some values of the parents (common in convolutional graph neural networks).
- Latent Dirichlet allocation when population of a plate is the domain of the child (e.g., topic of a word).
  The value of $A(v)$ can be used to compute $P(B{=}v)$:

  $$P(B{=}v) = \frac{exp(A(v))}{\sum_{v'} exp(A(v'))}$$

  E.g., B is a topic, A is the topic of a particular word instance.
  Each word instance is paying attention to a topic.

Example: Aggregation

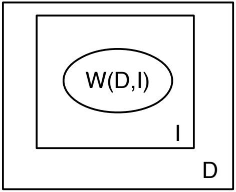


$$is\_shot(Y) \leftrightarrow \exists X\ shot(X,Y))$$
$$\lor\ shot\_by\_no\_one(Y)$$

noisy-or

Example: Language Models

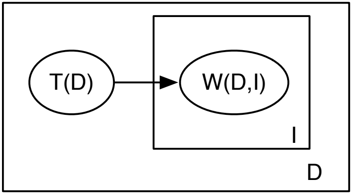
Unigram Model:



- 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 domain of W is the set of all words.

Example: Language Models

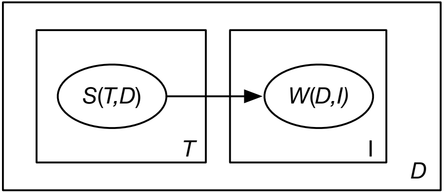
Topic Mixture:



- 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 domain of W is the set of all words.
- $T(d)$ is the topic of document d. The domain of T is the set of all topics.

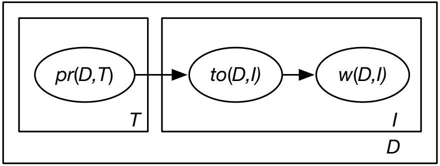
Example: Language Models

Mixture of topics, bag of words (unigram):



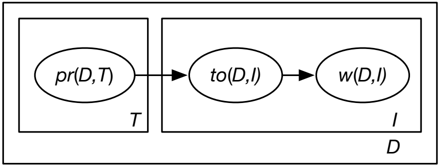
- D is the set of all documents
- I is the set of indexes of words in the document. I ranges from 1 to the number of words in the document.
- T is the set of all topics
- $W(d, i)$ is the i'th word in document d. The domain of W is the set of all words.
- $S(t, d)$ is true if topic t is a subject of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

Example: Latent Dirichlet Allocation



- $P(to(D,I) \mid pr(D,T))$ requires aggregation over T
- domain of $P(to(D,I)$ is T.
- could use (assuming $pr(D,T) \geq 0$ and $\sum_T pr(D,T) = 1$ for each D):

$$P(to(D,I){=}t) = pr(D,t)$$

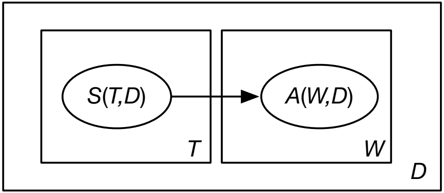
- alternative (assuming $pr(D,T)$ is real):

$$P(to(D,I){=}t \mid pr(D,T)) = \frac{exp(pr(D,t))}{\sum_{t'} exp(pr(D,t'))}$$

Each word is paying attention to a topic.

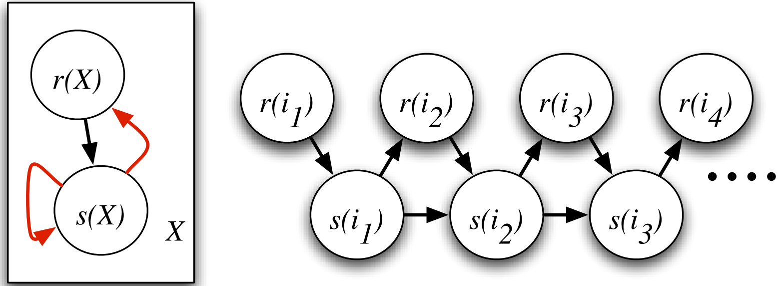
Example: Language Models

Mixture of topics, set of words:



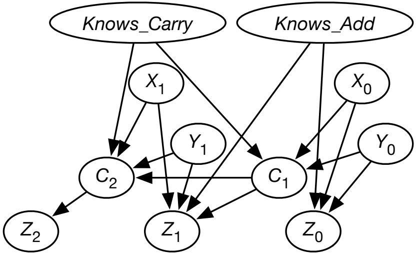
- D is the set of all documents
- W is the set of all words
- T is the set of all topics
- Boolean $A(w, d)$ is true if word w appears in document d
- Boolean $S(t, d)$ is true if topic t is a subject of document d.
- Rephil (Google) has 900,000 topics, 12,000,000 "words", 350,000,000 links.

Creating Dependencies: Exploit Domain Structure



Predicting students errors

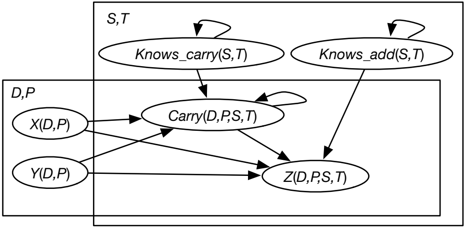
$$\begin{array}{rrrr} & & x_1 & x_0 \\ + & & y_1 & y_0 \\ \hline & z_2 & z_1 & z_0 \end{array}$$



- 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

$$\begin{array}{cccccc} & x_j & \cdots & x_1 & x_0 \\ + & y_j & \cdots & y_1 & y_0 \\ \hline & z_j & \cdots & z_1 & z_0 \end{array}$$



- 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 $x(D, P)$. . . .

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, Mysticism and Logic and Other Essays [1917]