CPSC 532P: Statistical Relational AI

David Poole

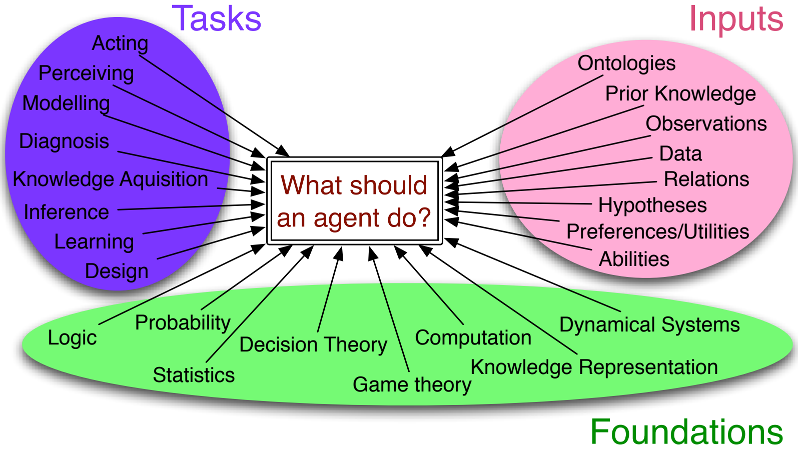
Department of Computer Science,
University of British Columbia

January 2017

> 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



Tasks

Acting
Perceiving
Modelling
Diagnosis
Knowledge Aquisition
Inference
Learning
Design

What should an agent do?

Inputs

Ontologies
Prior Knowledge
Observations
Data
Relations
Hypotheses
Preferences/Utilities
Abilities

Foundations

Logic
Probability
Statistics
Decision Theory
Game theory
Computation
Knowledge Representation
Dynamical Systems

Outline

1. Logic and Probability
    - Relational Probabilistic Models

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!

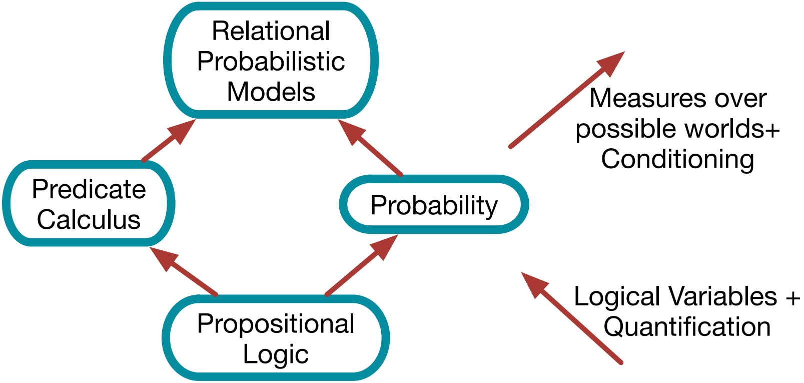
Classical (first order) logic lets us represent:

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

Why Probability?

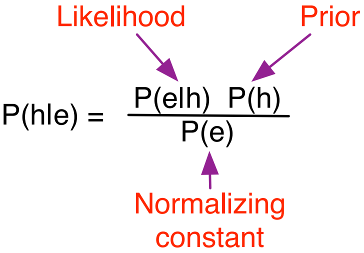
- There is lots of uncertainty about the world, but agents still need to act.
- Predictions are needed to decide what to do:
  - definitive predictions: you will be run over tomorrow
  - point probabilities: probability you will be run over tomorrow is 0.002 if you are not careful and 0.000001 if you 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


Relational Probabilistic Models
Predicate Calculus
Probability
Propositional Logic
Measures over possible worlds+ Conditioning
Logical Variables + Quantification

Bayes' Rule

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



Likelihood

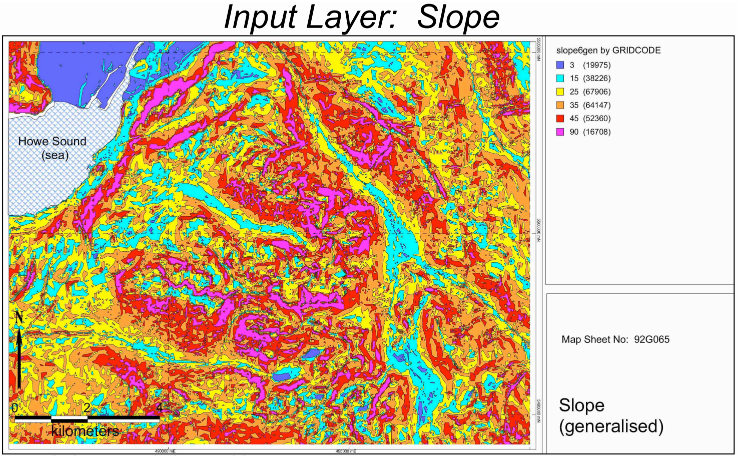
Prior

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

Normalizing constant

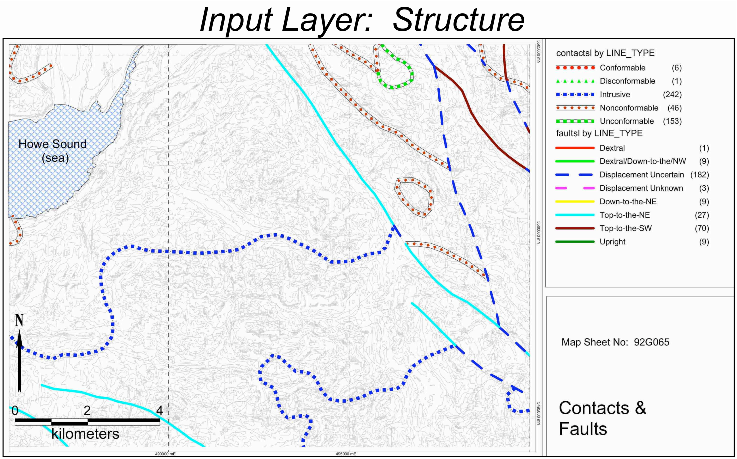
- What if *e* is a patient's electronic health record and *h* is the effect of a particular treatment on a particular patient?
- What if *e* is the electronic health records for all of the people in the province?
- What if *e* is a collection of student records in a university?
- What if *e* is everything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

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:
"Joe likes resorts that are near sandy beaches."

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

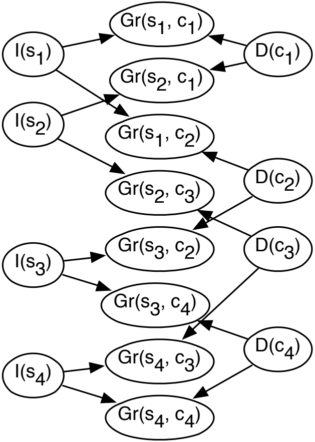
- But we want probabilistic predictions.

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



$I(S)$	$D(C)$	$Gr(S, C)$		
		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

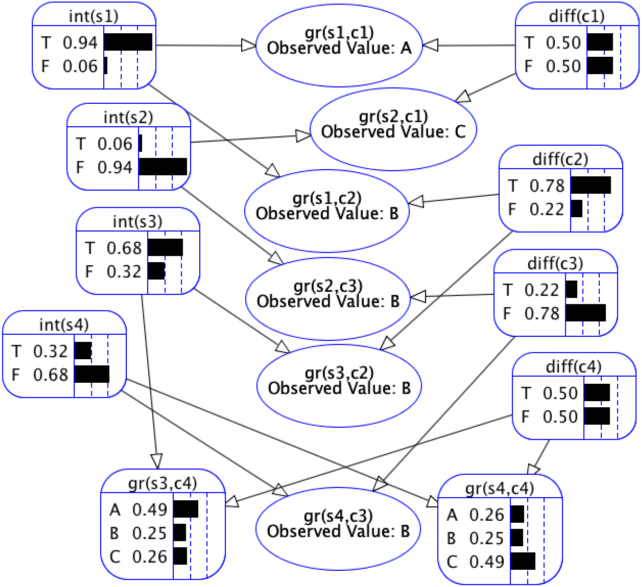
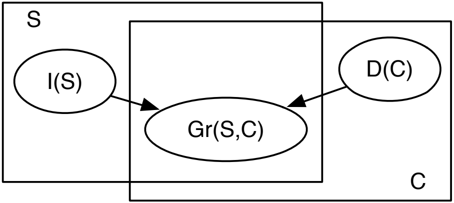


Plate Notation

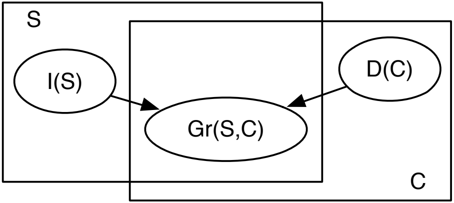


- S, C logical variable representing students, courses
- the set of individuals of a type is called a population
- $I(S)$, $Gr(S,C)$, $D(C)$ are parametrized random variables

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
  - 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.

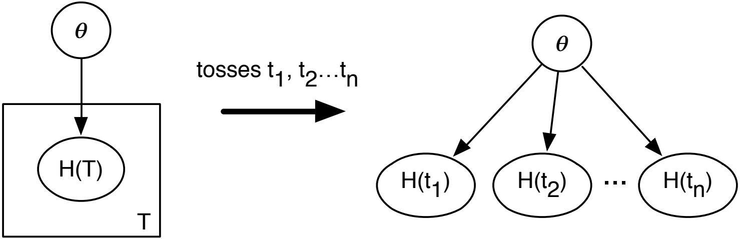
Exchangeability

- 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.

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

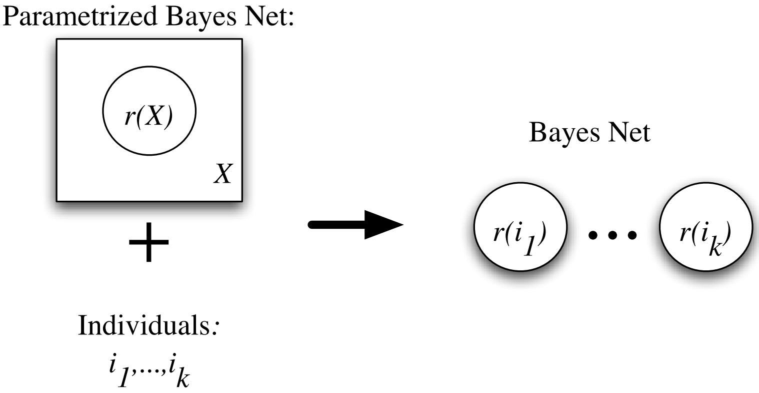


- T is a logical variable representing tosses of a thumb tack
- $H(t)$ is a Boolean variable that is true if toss t is heads.
- θ is a random variable representing the probability of heads.
- Range of θ is $\{0.0, 0.01, 0.02, \ldots, 0.99, 1.0\}$ or interval $[0, 1]$.
- $P(H(t_i){=}true|\theta{=}p) =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

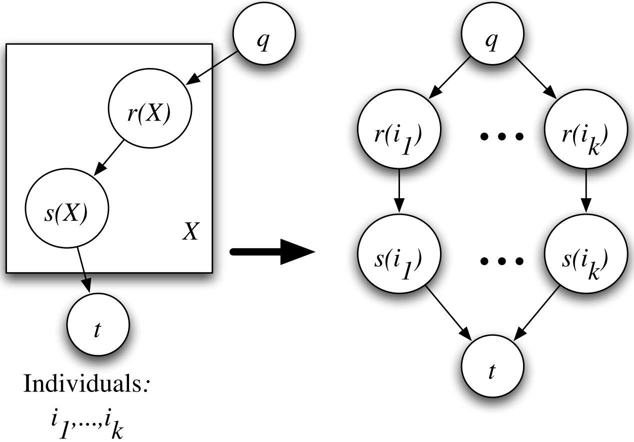


Parametrized Bayes Net:

Individuals: $i_1, \ldots, i_k$

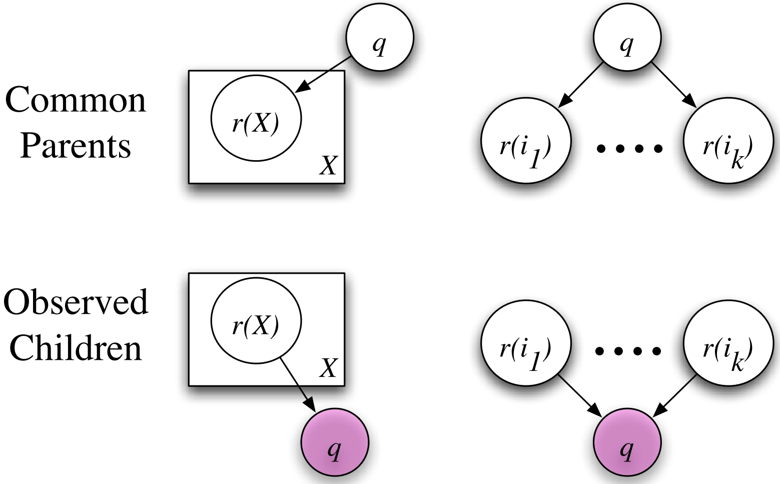
Bayes Net

Parametrized Bayesian networks / Plates (2)

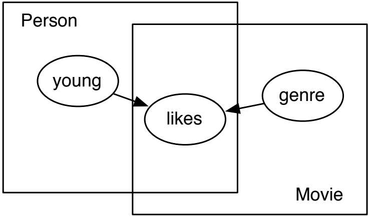


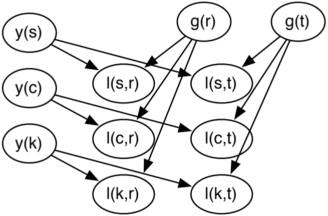
Creating Dependencies

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



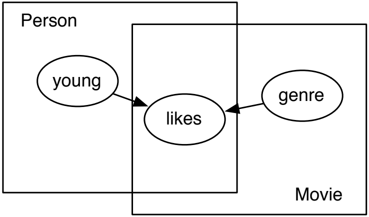
Overlapping plates





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

Overlapping plates

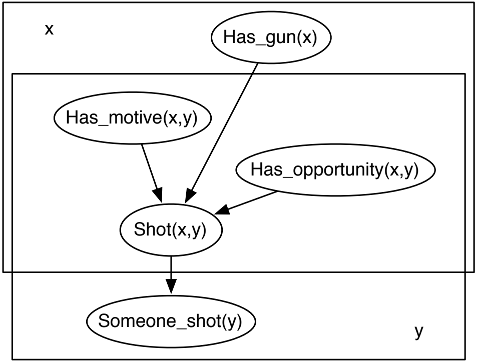


- Relations: *likes*(*P*, *M*), *young*(*P*), *genre*(*M*)
- *likes* is Boolean, *young* is Boolean, *genre* has range {*action*, *romance*, *family*}
- If there are 1000 people and 100 movies,
  Grounding contains: 100,000 likes + 1,000 age + 100 genre = 101,100 random variables
- How many numbers need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

Representing Conditional Probabilities

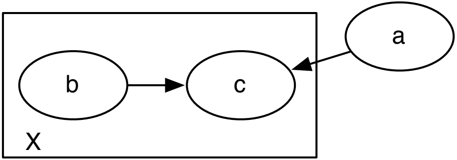
- $P(likes(P, M) | young(P), genre(M))$ — parameter sharing — individuals share probability parameters.
- $P(happy(X) | friend(X, Y), mean(Y))$ — needs aggregation — $happy(a)$ depends on an unbounded number of parents.
- There can be more structure about the individuals. . .

Example: Aggregation



Exercise #1

For the relational probabilistic model:



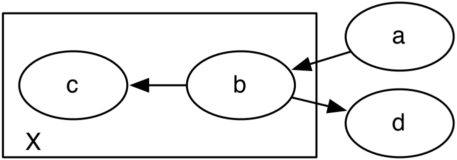
Suppose the the population of X is n and all variables are Boolean.

(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.

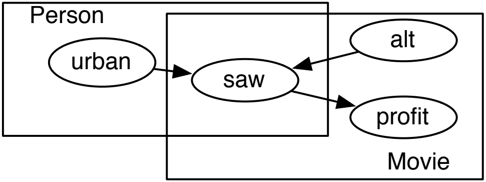
(a) Which of the conditional probabilities cannot be defined as a table?

(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:



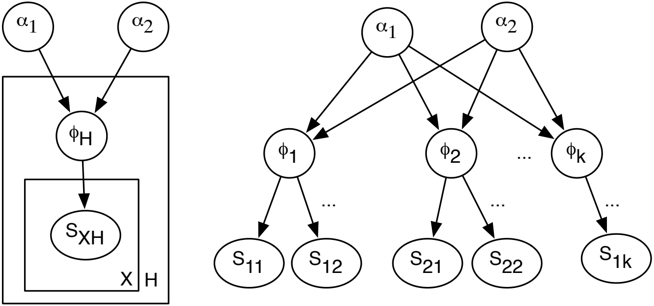
Suppose the population of *Person* is *n* and the population of *Movie* is *m*, and all variables are Boolean.

(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?

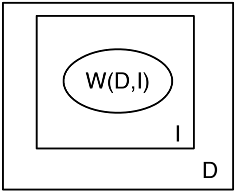


(a)

(b)

Example: Language Models

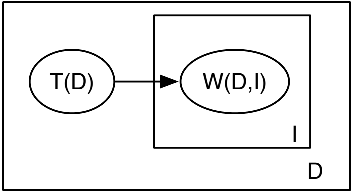
Unigram Model:



- 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

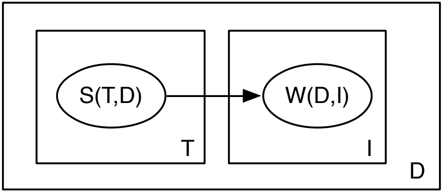
Topic Mixture:



- 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.

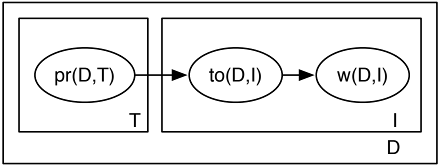
Example: Language Models

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.

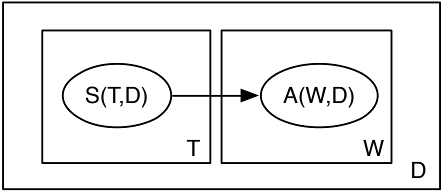
Example:Latent Dirichlet Allocation



- 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.

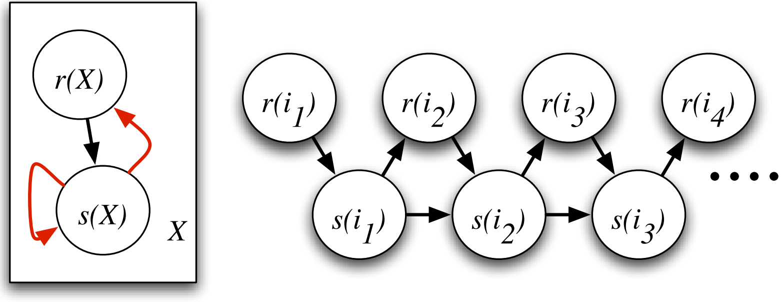
Example: Language Models

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.
- Rephil (Google) has 900,000 topics, 12,000,000 “words”, 350,000,000 links.

Creating Dependencies: Exploit Domain Structure


r(X)
s(X)
X
r(i_1)
r(i_2)
r(i_3)
r(i_4)
s(i_1)
s(i_2)
s(i_3)
....