"The mind is a neural computer, fitted by natural selection with combinatorial algorithms for causal and probabilistic reasoning about plants, animals, objects, and people."

. . .

"In a universe with any regularities at all, decisions informed about the past are better than decisions made at random. That has always been true, and we would expect organisms, especially informavores such as humans, to have evolved acute intuitions about probability. The founders of probability, like the founders of logic, assumed they were just formalizing common sense."

Steven Pinker, How the Mind Works, 1997, pp. 524, 343.

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. . . . [Your] 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]

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

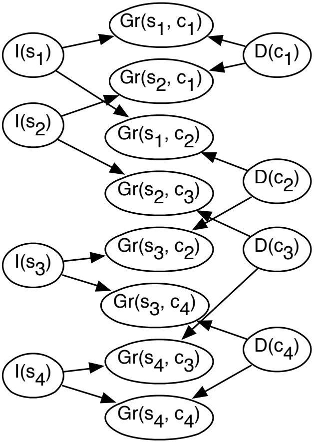
- build a probabilistic model before knowing the individuals
- learn the model for one set of individuals
- apply the model to new individuals
- allow complex relationships between individuals

Example: Predicting Relations

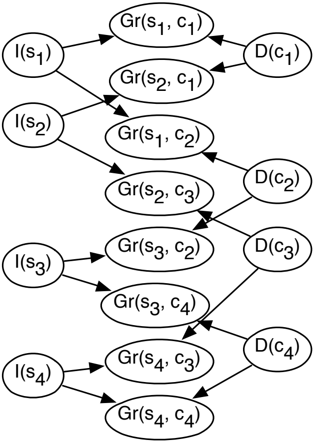
Student	*Course*	*Grade*
s_1	c_1	A
s_2	c_1	C
s_1	c_2	B
s_2	c_3	B
s_3	c_2	B
s_4	c_3	B
s_3	c_4	?
s_4	c_4	?

- 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



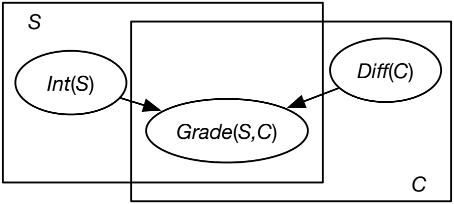
$I(S)$	$D(C)$	$Gr(S,C)$		
		A	B	C
true	*true*	0.5	0.4	0.1
true	*false*	0.9	0.09	0.01
false	*true*	0.01	0.1	0.9
false	*false*	0.1	0.4	0.5

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

"parameter sharing"

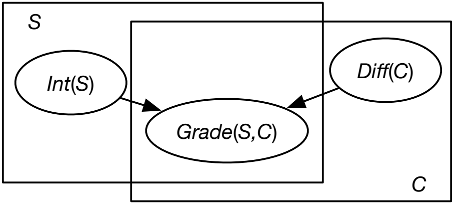
http://artint.info/code/aispace/grades.xml

Plate Notation



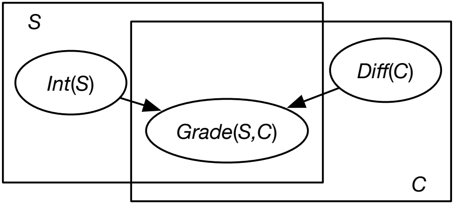
- S is a logical variable representing students
- C is a logical variable representing courses
- the set of all individuals of some type is called a population

Plate Notation



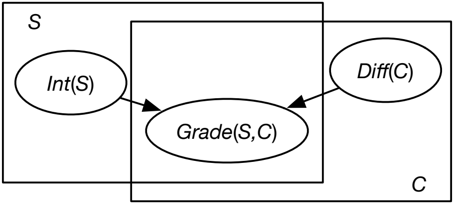
- S is a logical variable representing students
- C is a logical variable representing courses
- the set of all individuals of some type is called a population
- $I(S)$, $Gr(S,C)$, $D(C)$ are parametrized random variables

Plate Notation



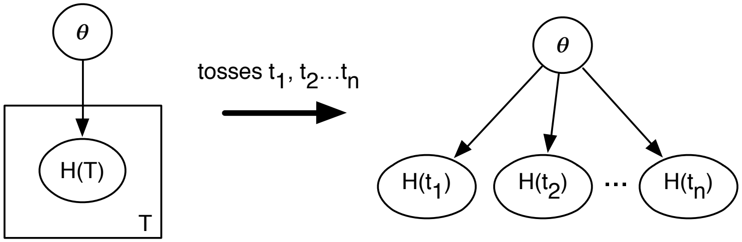
- S is a logical variable representing students
- C is a logical variable representing courses
- the set of all individuals of some type is called a population
- $I(S)$, $Gr(S, C)$, $D(C)$ are parametrized random variables
- for every student s, there is a random variable $I(s)$
- for every course c,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

Plate Notation



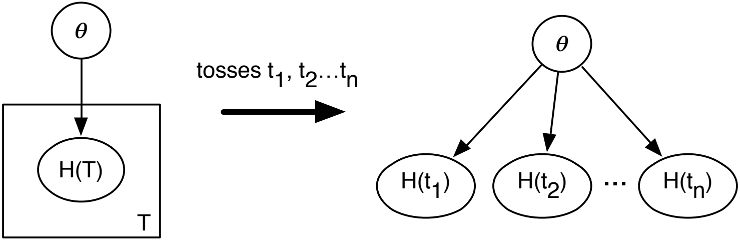
- S is a logical variable representing students
- C is a logical variable representing courses
- the set of all individuals of some type is called a population
- $I(S)$, $Gr(S, C)$, $D(C)$ are parametrized random variables
- for every student s, there is a random variable $I(s)$
- for every course c, 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.
- Domain of θ is $\{0.0, 0.01, 0.02, \ldots, 0.99, 1.0\}$ or interval $[0, 1]$.
- $P(H(t_i){=}true|\theta{=}p) =$

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.

Parametrized belief networks

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

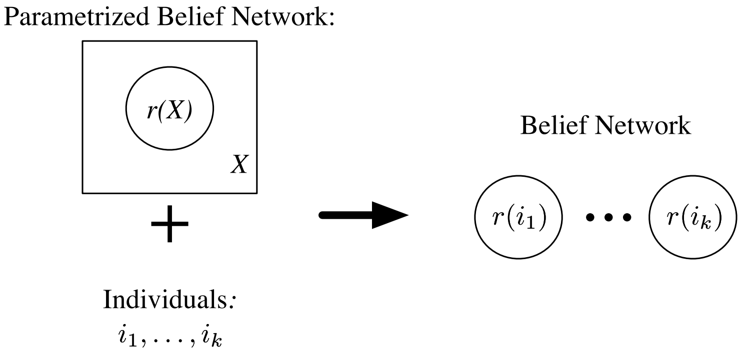
Parametrized belief networks

- 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. $|person| = 1000000$

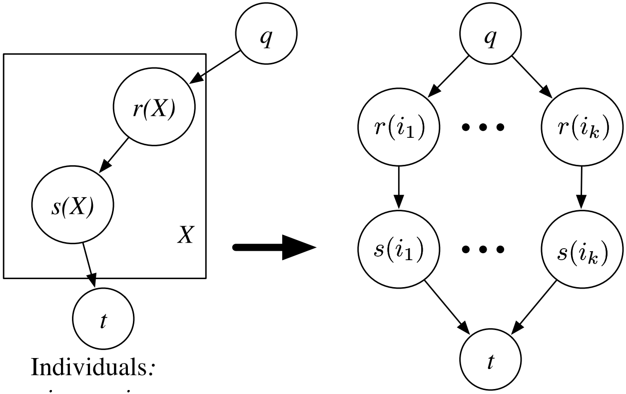
Parametrized belief networks

- 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. $interested(p_1) \dots interested(p_{1000000})$
- Instances are independent (but can have common ancestors and descendants).

Parametrized Belief Networks / Plates



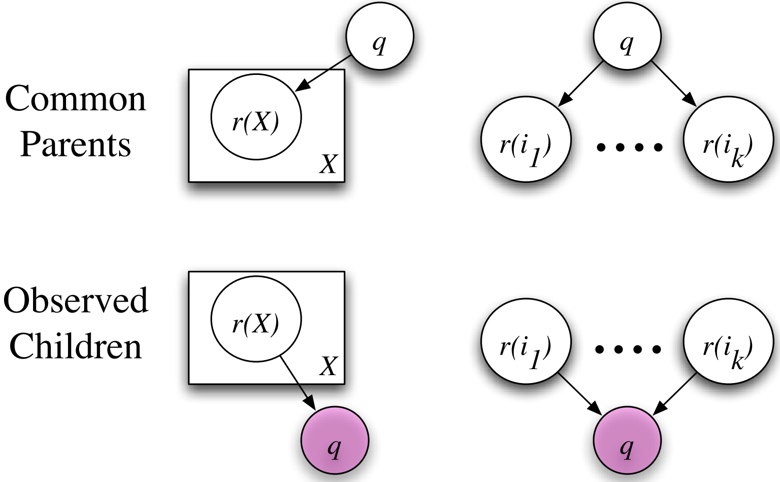
Parametrized Belief Networks / Plates (2)



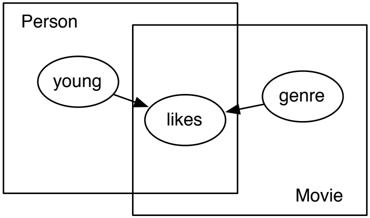
Individuals: $i_1, \ldots, i_k$

Creating Dependencies

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

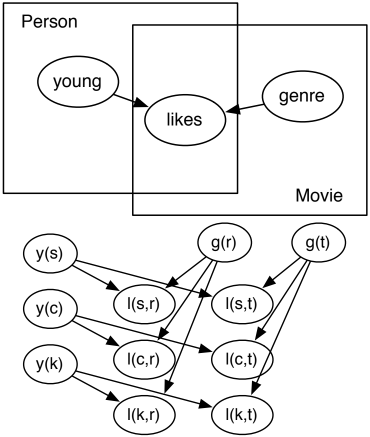


Overlapping plates



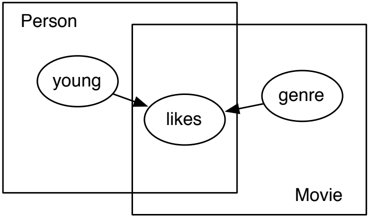
Relations: $likes(P, M)$, $young(P)$, $genre(M)$
likes is Boolean, *young* is Boolean,
genre has domain $\{action, romance, family\}$

Overlapping plates



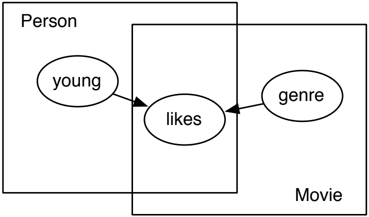
Relations: *likes*(P, M), *young*(P), *genre*(M)
likes is Boolean, *young* is Boolean,
genre has domain $\{$*action*, *romance*, *family*$\}$
Three people: sam (s), chris (c), kim (k)

Overlapping plates



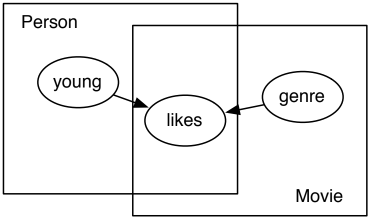
- Relations: $likes(P, M)$, $young(P)$, $genre(M)$
- $likes$ is Boolean, $young$ is Boolean, $genre$ has domain $\{action, romance, family\}$
- If there are 1000 people and 100 movies,
  Grounding contains:
  random variables

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?

Overlapping plates

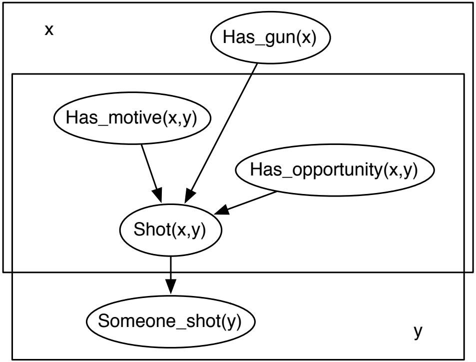


- Relations: $likes(P, M)$, $young(P)$, $genre(M)$
- *likes* is Boolean, *young* is Boolean, *genre* has domain $\{action, romance, family\}$
- If there are 1000 people and 100 movies,
  Grounding contains: 100,000 likes + 1,000 age + 100 genre = 101,100 random variables
- How many numbers need to be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

Representing Conditional Probabilities

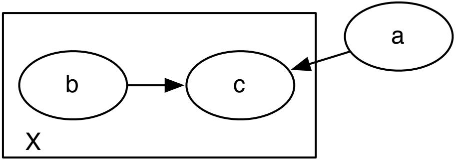
- $P(likes(P, M)|young(P), genre(M))$ — parameter sharing — individuals share probability parameters.
- $P(happy(X)|friend(X, Y), mean(Y))$ — needs aggregation — $happy(a)$ depends on an unbounded number of parents.
- There can be more structure about the individuals. . .

Example: Aggregation



Exercise #1

For the relational probabilistic model:

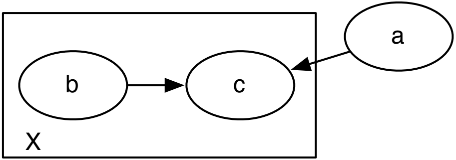


Suppose the the population of X is n and all variables are Boolean.

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

Exercise #1

For the relational probabilistic model:



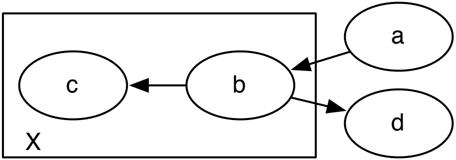
Suppose the the population of X is n and all variables are Boolean.

(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:

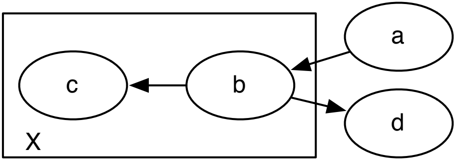


Suppose the the population of X is n and all variables are Boolean.

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

Exercise #2

For the relational probabilistic model:



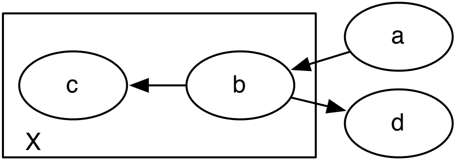
Suppose the the population of X is n and all variables are Boolean.

(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.

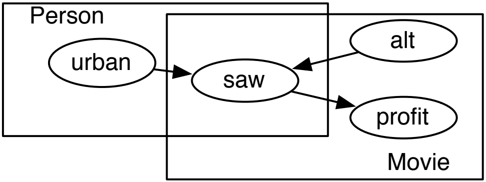
(a) Which of the conditional probabilities cannot be defined as a table?

(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:

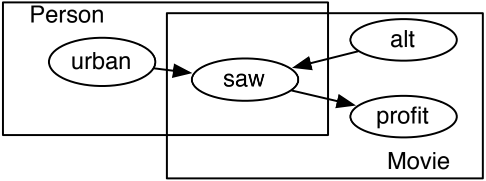


Suppose the population of *Person* is n and the population of *Movie* is m, and all variables are Boolean.

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

Exercise #3

For the relational probabilistic model:



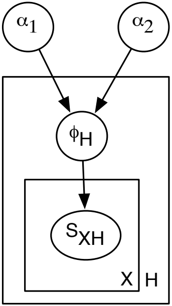
Suppose the population of *Person* is n and the population of *Movie* is m, and all variables are Boolean.

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

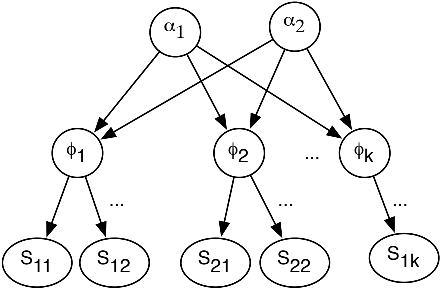
(b) How many numbers are required to specify the conditional probabilities? (Assume an "sum" is the aggregator and the rest are defined by tables).

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?



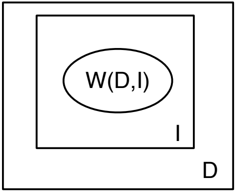
(a)



(b)

Example: Language Models

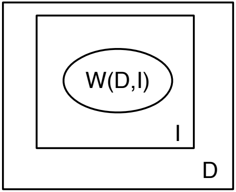
Unigram Model:



- 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.

Example: Language Models

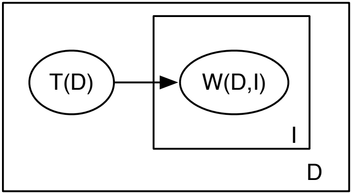
Unigram Model:



- 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

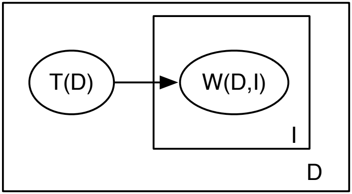
Topic Mixture:



- 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.

Example: Language Models

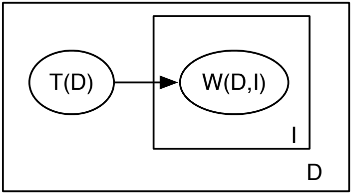
Topic Mixture:



- 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.

Example: Language Models

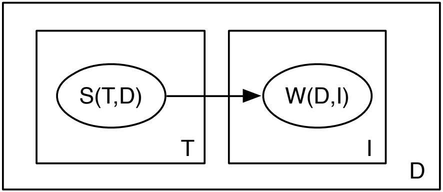
Topic Mixture:



- 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

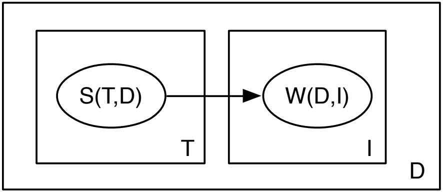
Mixture of topics, bag of words (unigram):



- 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

Example: Language Models

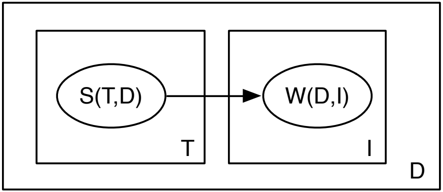
Mixture of topics, bag of words (unigram):



- 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.

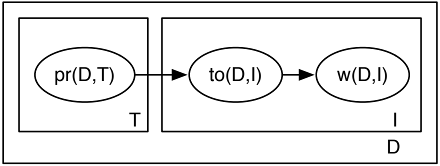
Example: Language Models

Mixture of topics, bag of words (unigram):



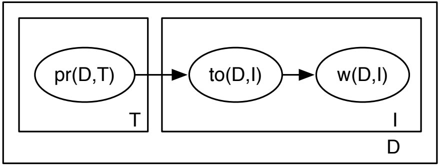
- D is the set of all documents
- I is the set of indexes of words in the document. I ranges from 1 to the number of words in the document.
- T is the set of all topics
- $W(d, i)$ is the i'th word in document d. The domain of W is the set of all words.
- $S(t, d)$ is true if topic t is a subject of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

Example:Latent Dirichlet Allocation



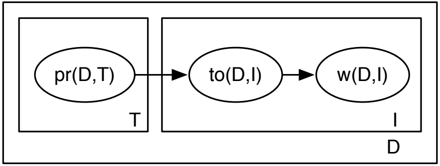
- D is the document
- I is the index of a word in the document. I ranges from 1 to the number of words in document D.
- T is the topic

Example:Latent Dirichlet Allocation



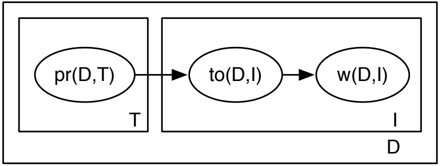
- D is the document
- I is the index of a word in the document. I ranges from 1 to the number of words in document D.
- T is the topic
- $w(d, i)$ is the i'th word in document d. The domain of w is the set of all words.

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 ith-word of document d. The domain of to is the set of all topics.

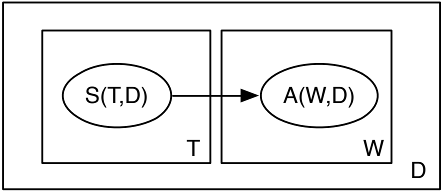
Example:Latent Dirichlet Allocation



- 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 ith-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: Language Models

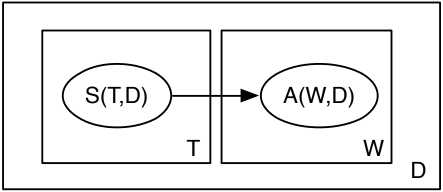
Mixture of topics, set of words:



- D is the set of all documents
- W is the set of all words.
- T is the set of all topics

Example: Language Models

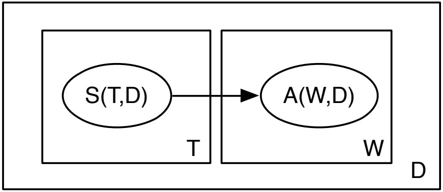
Mixture of topics, set of words:



- 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.

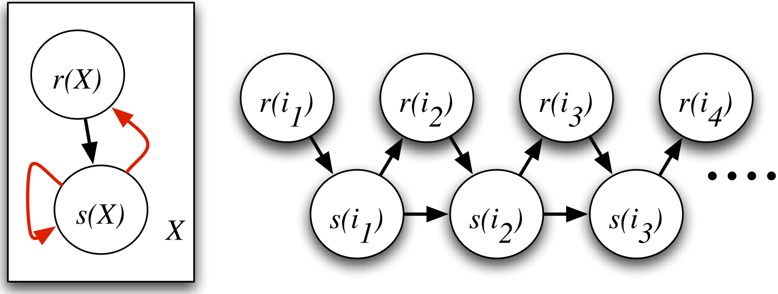
Example: Language Models

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

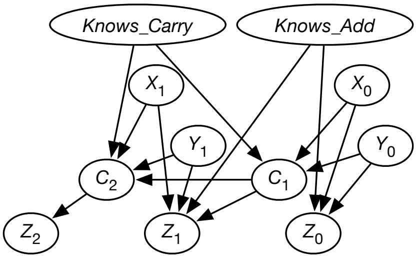


Predicting students errors

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

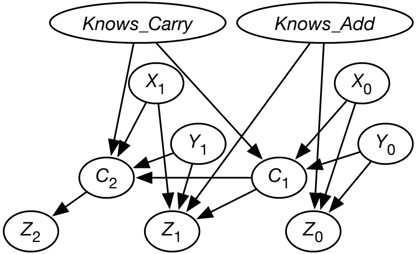
Predicting students errors

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



Predicting students errors

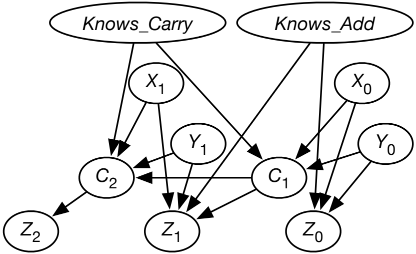
$$
\begin{array}{cccc}
 & & x_1 & x_0 \\
+ & & y_1 & y_0 \\
\hline
 & z_2 & z_1 & z_0
\end{array}
$$



- What if there were multiple digits

Predicting students errors

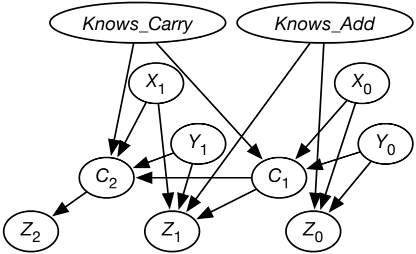
$$\begin{array}{rrrr} & & x_1 & x_0 \\ + & & y_1 & y_0 \\ \hline & z_2 & z_1 & z_0 \end{array}$$



- What if there were multiple digits, problems

Predicting students errors

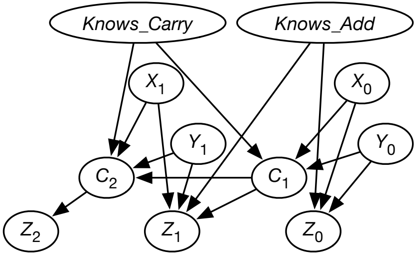
$$\begin{array}{cccc} & & x_1 & x_0 \\ + & & y_1 & y_0 \\ \hline & z_2 & z_1 & z_0 \end{array}$$



- What if there were multiple digits, problems, students

Predicting students errors

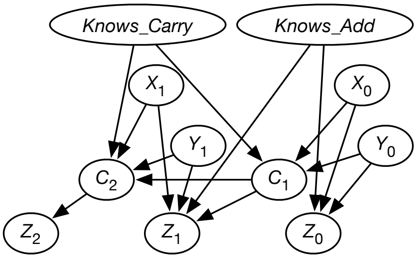
$$\begin{array}{cccc} & & x_1 & x_0 \\ + & & y_1 & y_0 \\ \hline & z_2 & z_1 & z_0 \end{array}$$



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

Predicting students errors

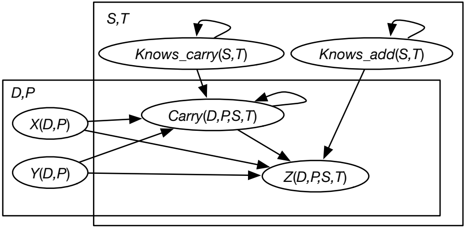
$$\begin{array}{cccc} & & x_1 & x_0 \\ + & & y_1 & y_0 \\ \hline & z_2 & z_1 & z_0 \end{array}$$



- 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

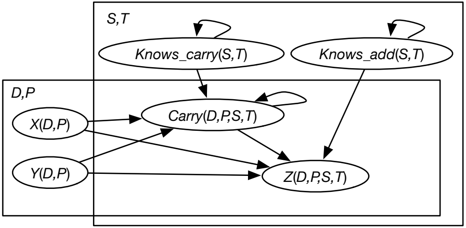
$$\begin{array}{rcccc} & x_j & \cdots & x_1 & x_0 \\ + & y_j & \cdots & y_1 & y_0 \\ \hline & z_j & \cdots & z_1 & z_0 \end{array}$$



- Parametrized Random Variables:

Multi-digit addition with parametrized BNs / plates

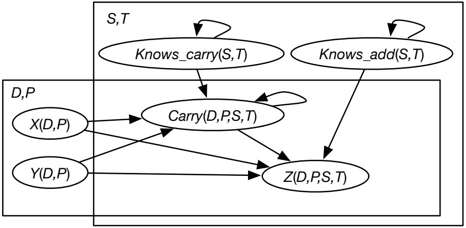
$$
\begin{array}{rcccc}
 & x_j & \cdots & x_1 & x_0 \\
+ & y_j & \cdots & y_1 & y_0 \\
\hline
 & z_j & \cdots & z_1 & z_0
\end{array}
$$



- 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:

Multi-digit addition with parametrized BNs / plates

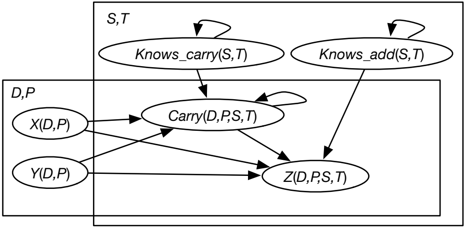
$$
\begin{array}{cccccc}
 & x_j & \cdots & x_1 & x_0 \\
+ & y_j & \cdots & y_1 & y_0 \\
\hline
 & z_j & \cdots & z_1 & z_0
\end{array}
$$



- 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:

Multi-digit addition with parametrized BNs / plates

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



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

Independent Choice Logic

- A language for relational 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.

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.
  $\mathcal{F}$ can include negation as failure
  No member of an alternative unifies with the head of a clause.
- 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

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

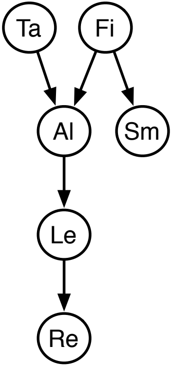
$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$
$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$

		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 and ICL rules

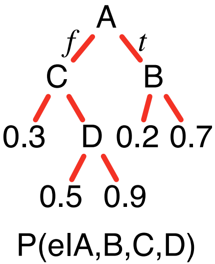
- (Discrete) belief networks can be directly mapped into ICL. There is an alternative for each free parameter.



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.

Decision Trees and ICL rules

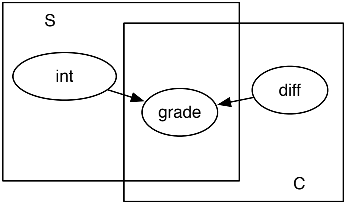
- Rules can represent decision tree with probabilities:



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

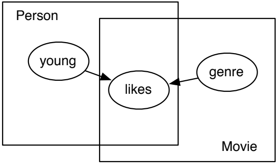
Predicting Grades

Plates correspond to logical variables.



prob $int(S) : 0.5.$
prob $diff(C) : 0.5.$
$grade(S, C, G) \leftarrow int(S) \land diff(C) \land idg(S, C, G).$
prob $idg(S, C, a) : 0.5,\ idg(S, C, b) : 0.4,\ idg(S, C, c) : 0.1.$
$grade(S, C, G) \leftarrow int(S) \land \sim diff(C) \land indg(S, C, G).$
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 $genre(M, action) : 0.4, genre(M, romance) : 0.3,$
$\quad genre(M, family) : 0.4.$

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

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

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

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

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

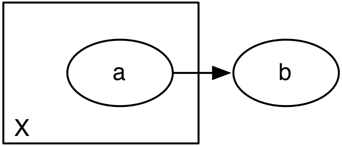
prob $lny(P, M, action) : 0.2.$

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

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

Aggregation

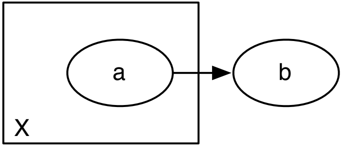
The relational probabilistic model:



Cannot be represented using tables. Why?

Aggregation

The relational probabilistic model:



Cannot be represented using tables. Why?

- This can be represented in ICL by

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

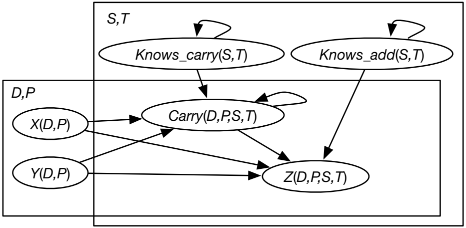
  "noisy-or", where $n(X)$ is a noise term, $\{n(X), \sim n(X)\} \in \mathcal{C}$
- If $a(c)$ is observed for each individual c:

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

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

Example: Multi-digit addition

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



ICL rules for multi-digit addition

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

ICL rules for multi-digit addition

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

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

Alternatives:

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

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

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: 500K users, 17k movies, 100M ratings (now unavailable).
Movielens: multiple datasets from 100K to 25M ratings,
also with links to IMDB, plus user properties for smaller datasets

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: 500K users, 17k movies, 100M ratings (now unavailable).
Movielens: multiple datasets from 100K to 25M ratings,
also with links to IMDB, plus user properties for smaller datasets
$\widehat{r_{ui}}$ = predicted rating of user u on item i
Es = set of (u, i, r) tuples in the training set (ignoring Date)
Sum squares error:

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

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 parameters except

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 parameters except μ.

Regularizing

Two possible methods for regularization:

- Minimize for each example

$$\sum_{(u,i,r)\in Es} \left((\widehat{r_{ui}} - r)^2 + \lambda \sum_{parameter\ p} p^2 \right)$$

- Minimize for whole dataset

$$\left(\sum_{(u,i,r)\in Es} (\widehat{r_{ui}} - r)^2 \right) + \lambda \sum_{parameter\ p} p^2$$

- For standard supervised learning,

Regularizing

Two possible methods for regularization:

- Minimize for each example

$$\sum_{(u,i,r)\in Es}\left((\widehat{r_{ui}}-r)^2+\lambda\sum_{parameter\ p}p^2\right)$$

- Minimize for whole dataset

$$\left(\sum_{(u,i,r)\in Es}(\widehat{r_{ui}}-r)^2\right)+\lambda\sum_{parameter\ p}p^2$$

- For standard supervised learning, it doesn't matter. The λs differ by a factor of $|Es|$.
- For collaborative filtering,

Regularizing

Two possible methods for regularization:

- Minimize for each example

$$\sum_{(u,i,r)\in Es} \left((\widehat{r_{ui}} - r)^2 + \lambda \sum_{parameter\ p} p^2 \right)$$

- Minimize for whole dataset

$$\left(\sum_{(u,i,r)\in Es} (\widehat{r_{ui}} - r)^2 \right) + \lambda \sum_{parameter\ p} p^2$$

- For standard supervised learning, it doesn't matter. The λs differ by a factor of $|Es|$.
- For collaborative filtering, it does matter, as there are varied number of ratings for each movie, and for each user.

Proposal 1: Regularize for each example

Minimize:

Proposal 1: Regularize for each example

Minimize:

$$\sum_{(u,i,r)\in Es} \left((\mu + b_i + c_u + \sum_k f_{ik} g_{ku} - r)^2 \right.$$
$$\left. + \lambda(b_i^2 + c_u^2 + \sum_k f_{ik}^2 + g_{ku}^2) \right)$$

where λ is a regularization parameter.

Proposal 1: Regularize for each example

Minimize:

$$\sum_{(u,i,r)\in Es} \left((\mu + b_i + c_u + \sum_k f_{ik} g_{ku} - r)^2 \right.$$
$$\left. + \lambda (b_i^2 + c_u^2 + \sum_k f_{ik}^2 + g_{ku}^2) \right)$$

where λ is a regularization parameter.

To find minimizing parameters:

- Gradient descent
- Iterative least squares

Proposal 1: Regularize for each example, Gradient Descent

input:
  Es is set of (u, i, r) triples
  η is learning rate
  λ is regularization parameter
$\mu :=$ average rating
assign $f[i, k]$, $g[k, u]$ randomly
assign $b[i], c[u]$ arbitrarily
repeat:
  for each $(u, i, r) \in Es$:
    $e := \mu + b[i] + c[u] + \sum_k f[i, k] * g[k, u] - r$
    $b[i] := b[i] - \eta * e - \eta * \lambda * b[i]$
    $c[u] := c[u] - \eta * e - \eta * \lambda * c[u]$
    for each feature k:
      $f[i, k] := f[i, k] - \eta * e * g[k, u] - \eta * \lambda * f[i, k]$
      $g[k, u] := g[k, u] - \eta * e * f[i, k] - \eta * \lambda * g[k, u]$

Proposal 2: Regularize Globally

Minimize:

Proposal 2: Regularize Globally

Minimize:

$$\left(\sum_{(u,i,r)\in Es} (\mu + b_i + c_u + \sum_k f_{ik} g_{ku} - r)^2\right)$$
$$+ \lambda \left(\sum_i (b_i^2 + \sum_k f_{ik}^2) + \sum_u (c_u^2 + \sum_k g_{ku}^2)\right)$$

where λ is a regularization parameter.

Proposal 2: Regularize Globally

Minimize:

$$\left(\sum_{(u,i,r) \in Es} (\mu + b_i + c_u + \sum_k f_{ik} g_{ku} - r)^2 \right)$$
$$+ \lambda \left(\sum_i (b_i^2 + \sum_k f_{ik}^2) + \sum_u (c_u^2 + \sum_k g_{ku}^2) \right)$$

where λ is a regularization parameter.

To find minimizing parameters:

- Gradient descent
- Iterative least squares

$\mu :=$ average rating
assign $f[i,k]$, $g[k,u]$ randomly and assign $b[i], c[u]$ arbitrarily
repeat:

  for each $(u,i,r) \in Es$:
    $e := \mu + b[i] + c[u] + \sum_k f[i,k] * g[k,u] - r$
    $b[i] := b[i] - \eta * e$
    $c[u] := c[u] - \eta * e$
    for each feature k:
      $f[i,k] := f[i,k] - \eta * e * g[k,u]$
      $g[k,u] := g[k,u] - \eta * e * f[i,k]$
  for each item i:
    $b[i] := b[i] - \eta * \lambda * b[i]$
    for each feature k:
      $f[i,k] := f[i,k] - \eta * \lambda * f[i,k]$
  for each user u:
    $c[u] := c[u] - \eta * \lambda * c[u]$
    for each feature k:
      $g[k,u] := g[k,u] - \eta * \lambda * g[k,u]$