Bayes' Nets

Representation ✔

Conditional Independences ✔

Probabilistic Inference (Next).

- Enumeration (exact, exponential complexity)
- Variable elimination (exact, worst-case exponential complexity, often better)
- Probabilistic inference is NP-complete
- Sampling (approximate)

Learning Bayes' Nets from Data

Examples:

- Posterior probability
- Most likely explanation:

Inference

Inference: calculating some useful quantity from a joint probability distribution

Inference by Enumeration

General case:

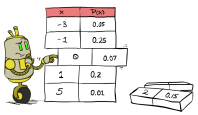
- Evidence variables: $E_1, \ldots, E_k = e_1, \ldots, e_k$
- Query* variable: Q
- Hidden variables: $H_1, \ldots, H_r$

All Variables:
Evidence $\cup$ {Q} $\cup$ Hidden.

We want: $Pr(Q|e_1, \ldots, e_k)$

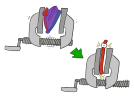
* Works fine with multiple query variables, too

Step 1: Select the entries consistent with the evidence



x	P(x)
-3	0.05
-1	0.25
0	0.07
1	0.2
5	0.01

Step 2: Sum out H to get joint of Query and evidence



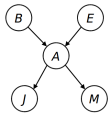
$P(Q, e_1, \ldots, e_k) = \sum_{h_1, \ldots, h_r} P(Q, h_1, \ldots, h_r, e_1, \ldots, e_k)$

Step 3: Normalize.

$Z = \sum_q P(Q = q, e_1, \ldots, e_k)$

$P(Q|e_1, \ldots, e_k) = \frac{1}{Z} P(Q, e_1, \ldots, e_k)$

Inference by Enumeration in Bayes' Net



Given unlimited time, inference in BNs is easy.

$P(B|+j,+m) \propto P(B,+j,+m)$

$= \sum_{e,a} P(B,e,a,+j,+m)$

$= \sum_{e,a} P(B)P(e)P(a|B,e)P(+j|a)P(+m|a)$

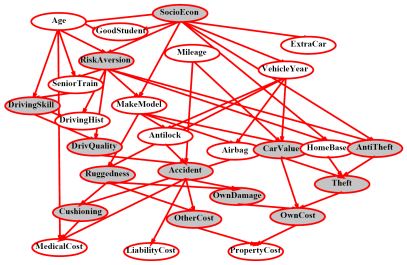
$= P(B)P(+e)P(+a|B,+e)P(+j|+a)P(+m|+a)$

$+P(B)P(+e)P(-a|B,+e)P(+j|-a)P(+m|-a)+$

$P(B)P(-e)P(+a|B,-e)P(+j|+a)P(+m|+a)$

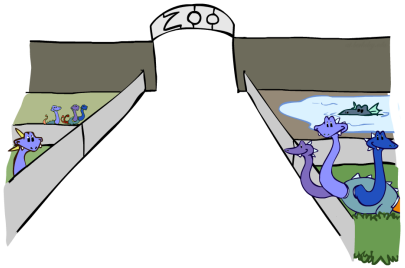
$+P(B)P(-e)P(-a|B,-e)P(+j|-a)$

Inference by Enumeration?



$P(antilock|observedvariables) = ?$

Factor Zoo



Factor Zoo I

Joint distribution: P(X,Y)
- Entries P(x,y) for all x, y
- Sums to? 1

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Selected joint: P(x,Y)
- Slice of joint distribution
- Entries P(x,y): fixed x, all y
- Table P(cold, W)?
- Sums to? P(x).

$P(cold, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3

Number of capital letters (variables)= dimensionality of the table

Factor Zoo II

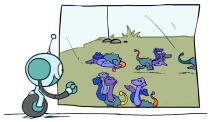
Single conditional: $P(Y|x)$
- Entries $P(y|x)$ for fixed x, all y
- Sum of entries? 1

T	W	P
cold	sun	0.4
cold	rain	0.6

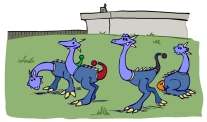
Family of conditionals:

$P(X|Y)$
- Multiple conditionals
- Entries $P(x|y)$ for all x, y
- Sum? $|Y|$

T	W	P
hot	sun	0.8
hot	rain	0.2
cold	sun	0.4
cold	rain	0.6



Factor Zoo III

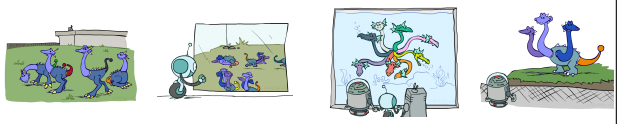


Specified family: $P(y|X)$
- Entries $P(y|x)$ for fixed y, but for all x
- Sums to? who knows!

$P(rain|T)$

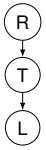
T	W	P
rain	hot	0.2
rain	cold	0.6

Factor Zoo Summary



In general, when we write $P(Y_1, \ldots, Y_N | X_1, \ldots, X_M)$
- It is a "factor": a multi-dimensional array
- Its entries are $P(y_1, \ldots, y_N | x_1, \ldots, x_M)$
- Any assigned (=lower-case) X or Y is a dimension missing (selected) from the array

Example: Traffic Domain



$P(R)$

T	L
+r	0.1
-r	0.9

$P(T|R)$

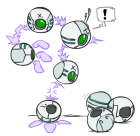
+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Random Variables
- R: Raining
- T: Traffic
- L: Late for class!

Inference by Enumeration: Procedural Outline



Track objects called factors.

Initial factors are local CPTs (one per node)

$P(R)$

T	L
+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Any known values are selected.
E.g. if we know $L = +l$, the initial factors are

$P(R)$

T	L
+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Procedure: Join all factors, then eliminate all hidden variables

Operation 1: Join Factors

First basic operation: joining factors

Combining factors:

- Just like a database join
- Get all factors over the joining variable
- Build a new factor over the union of the variables involved

Example: Join on R

$P(R)$

T	L
+r	0.1
-r	0.9

$\times$

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$\rightarrow$

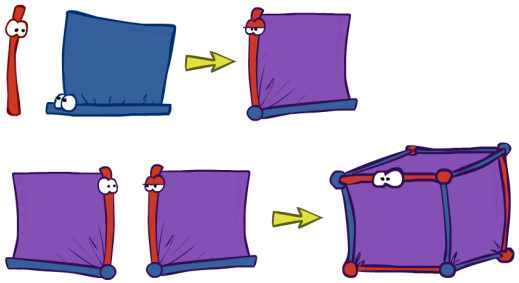
R,T

$P(R,T)$

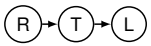
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

- Computation for each entry: pointwise products

Example: Multiple Joins



Example: Multiple Joins



$P(R)$

T	L
+r	0.1
-r	0.9

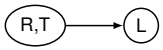
$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R $\rightarrow$



$P(R,T)$

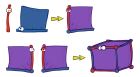
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join T. $\rightarrow$

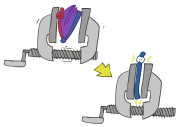
R,T,L



$P(R,T,L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

Operation 2: Eliminate



Second basic operation: marginalization

Take a factor and sum out a variable

- Shrinks a factor to a smaller one
- A projection operation

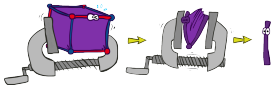
Example:

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

Sum R $\rightarrow$

+t	0.17
-t	0.83

Multiple Elimination



R,T,L

$P(R,T,L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

Sum out R $\rightarrow$

T,L

+t	+l	0.026
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

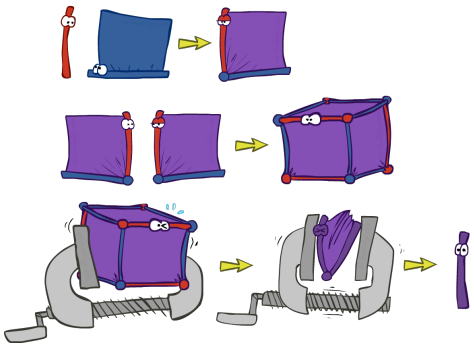
Sum out T $\rightarrow$

L

+l	0.134
-l	0.866

Thus Far: Multiple Join, Multiple Eliminate

Inference by Enumeration!



Example: Traffic Domain (revisited)

$P(R)$

T	L
+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

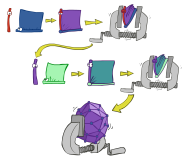
Random Variables

- R: Raining
- T: Traffic
- L: Late for class!

$P(L) = \sum_{r,t} P(r,t,L) = \sum_{r,t} P(r)P(t|r)P(L|t)$.

or $P(L) = \sum_t P(t)P(L|t)$ and $P(T) = \sum_r P(r)P(T|R)$.

Inference by Enumeration vs. Variable Elimination



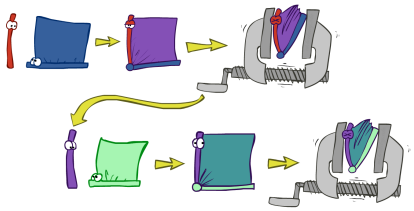
Why is inference by enumeration so slow?

- Join whole joint distribution before sum out the hidden variables

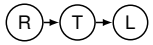
Idea: interleave joining and marginalizing!

- Called "Variable Elimination"
- Still NP-hard, but usually much faster than inference by enumeration

Marginalizing Early (= Variable Elimination)



Traffic Domain



Inference by Enumeration:

$P(L) = \sum_t \sum_r P(r)P(t|r)P(L|t)$.

Join on r: $P(r)P(t|r)$

Join on t: $\times P(L|t)$

Eliminate r: $\sum_r$

Eliminate t: $\sum_t$

Variable Elimination:

$P(L) = \sum_t P(L|t) \sum_r P(r)P(t|r)$.

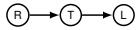
Join on r: $P(r)P(t|r)$

Eliminate r: $\sum_r$

Join on t: $\times P(L|t)$

Eliminate t: $\sum_t$

Marginalizing Early! (aka VE)



$P(R)$

T	L
+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Sum out R

+t	0.17
-t	0.83

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join T

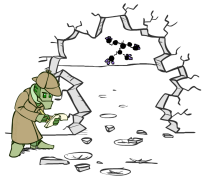
$P(T,L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

Sum out T

+l	0.134
-l	0.866

Evidence



Start with factors that select evidence

- Initial factors:

$P(R)$

T	L
+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- Computing $P(L|+r)$, the initial factors become:

$P(R)$

T	L
+r	0.1

$P(T|R)$

+r	+t	0.8
+r	-t	0.2

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Then eliminate variables to answer query.

Evidence II

Result will be a selected joint of query and evidence

- E.g. for $P(L|+r)$, we would end up with:

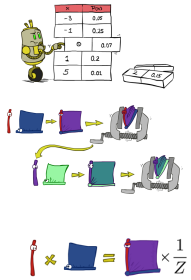
+r	+ l	0.026
+r	- l	0.074

To get our answer, just normalize this!

+r	+ l	0.26
+r	- l	0.74

That 's it!

General Variable Elimination



Query: Q

Start with initial factors:

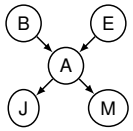
- Local CPTs (but instantiated by evidence)

While still hidden variables (not Q or evidence):

- Pick a hidden variable H
- Join all factors mentioning H
- Eliminate (sum out) H

Join all remaining factors and normalize

Example



$P(B|j,m) \propto P(B,j,m)$

Selected CPT:

$P(B)$ $P(E)$ $P(A|B,E)$
$P(j|A)$ $P(m|A)$

Choose Hidden Variable A.
Factors: $P(A|B,E)$, $P(j|A)$,$P(m|A)$.
Join: $P(j,m,A|B,E)$. Sum: $P(j,m|B,E)$

Remaining Factors: $P(B)$,$P(E)$,$P(j,m|B,E)$

Choose Hidden Variable E:
Factors: $P(j,m|B,E)$,$P(E)$. Join: $P(j,m,E|B)$. Sum: $P(j,m|B)$

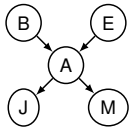
Remaining Factors: $P(B)$,$P(j,m|B)$

Finish with B: Factors: $P(j,m|B,E)$,$P(B)$. Join: $P(j,m,B)$.

And Finally....
Normalize: $P(B|j,m)$

Same Example in Equations

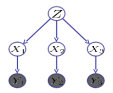
Factors: $P(B)$ $P(E)$ $P(A|B,E)$ $P(j,A)$ $P(m,A)$



	$P(B\|j,m) \propto P(B,j,m)$
Sum out to get marginal.	$= \sum_{e,a} P(B,j,m,e,a)$
use Bayes' net joint distribution expression	$= \sum_{e,a} P(B)P(e)P(a\|B,e)P(j,a)P(m\|a)$
use x*(y+z) = xy + xz	$= \sum_e P(B)P(e)\sum_a P(a\|B,e)P(j,a)P(m\|a)$
joining on a, and then summing out gives f1	$= \sum_e P(B)P(e)f_1(j,m\|B,e)$
use x*(y+z) = xy + xz	$= P(B)\sum_e P(e)f_1(j,m\|B,e)$
joining on e, and then summing out gives f2	$= P(B)f_2(j,m\|B)$

Exploits:
$uwy + uwz + uxy + uxz + vwy + vwz + vxy + vxz = (u+v)(w+x)(y+z)$.

Another Variable Elimination Example



Size of factor = number of entries.

All factors of size 2 — only one variable (Z, Z, X_3).

Query: $P(X_3|y_1,y_2,y_3)$.

Evidence:
$P(Z), P(X_1|Z), P(X_2|Z), P(X_3|Z),$
$P(y_1|X_1), P(y_2|X_2), P(y_3|X_3)$

Eliminate X_1, get $f_1(y_1|Z) = \sum_{x_1} P(x_1|Z)P(y_1|x_1)$.

$P(Z), P(X_1|Z), P(X_2|Z), P(X_3|Z), P(y_2|X_2), f_1(y_1|Z)$

Eliminate X_2, get $f_2(y_2|Z) = \sum_{x_2} P(x_2|Z)P(y_2|x_2)$.

$P(Z), P(X_1|Z), P(X_3|Z), f_2(y_2|Z), f_1(y_1|Z)$

Eliminate Z,
get $f_3(y_1,y_2,X_3) = \sum_z P(z)P(X_3|z)f_1(y_1|z)f_2(y_2|z)$.

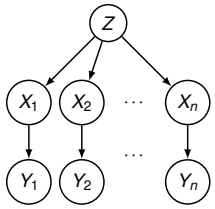
$P(y_3|X_3), f_3(y_1,y_2,X_3)$

No more hidden variables. Join.

$f_4(y_1,y_2,y_3,X_3) = P(y_3|X_3)f_3(y_1,y_2,X_3)$

Normalizing gives result.

Variable Elimination Ordering



Query $P(X_n|y_1,\ldots,y_n)$.

Consider the following orderings:
$Z, X_1, \ldots, X_{n-1}$ and
$X_1, \ldots, X_{n-1}, Z$.

What is the size of the maximum factor generated for each of the orderings?

Eliminate Z? New factor with $X_1, \ldots, X_n$ of size 2^n.
Uh oh.

Eliminate X_1? New factor with Z and y_1 of size 2.
Eliminate X_2? New factor with Z and y_2 of size 2.
... Cool!

Answer: 2 versus 2^n (assuming binary variables.)

In general: the ordering can greatly affect efficiency.

VE: Computational and Space Complexity

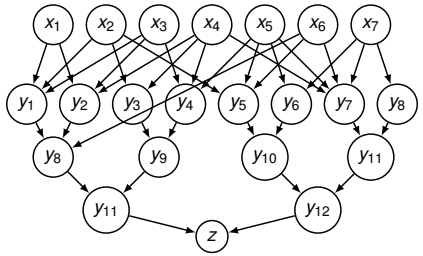
Computational and space complexity of variable elimination depends on largest factor

Elimination ordering can greatly affect size of the largest factor.
- E.g., previous slide's example 2^n vs. 2

Does there always exist an ordering that only results in small factors?
- No!

Worst Case Complexity?

CSP:
$(x_1 \vee x_2 \neg x_3), (\neg x_1 \vee x_3 \vee \neg x_4), (x_2 \vee \neg x_2 \vee x_4), (\neg x_3 \vee \neg x_4 \vee \neg x_3),$
$(x_3 \vee x_5 \vee x_7), (x_4 \vee x_5 \vee x_6), (\neg x_5 \vee x_6 \vee \neg x_7), (\neg x_5 \vee \neg x_6 \vee x_7)$



$P(X_i = 1) = 0.5,$
$P(X_i = 0) = 0.5$

$Y_1 = X_1 \vee X_2 \vee X_3$

...

$P(z)$?

If $P(z) \neq 0$, the 3-SAT problem has a solution!

Inference in Bayes' nets is NP-hard.
No known efficient probabilistic inference in general.

"Easy" Structures: Polytrees

A polytree is a directed graph with no undirected cycles

For poly-trees you can always find an ordering that is efficient
- Try it!!

Cut-set conditioning for Bayes' net inference
- Choose set of variables such that if removed only a polytree remains
- Exercise: Think about how the specifics would work out!

Bayes' Nets

Representation ✔

D-separation. ✔

Probabilistic Inference. ✔

- E Enumeration (exact, exponential complexity). ✔
- Variable elimination
(exact, worst-case exponential complexity, often better) . ✔
- Inference is NP-complete. ✔
- Sampling (approximate) (Next up.)

Learning Bayes' Nets from Data. (Later.)