Bayesian Networks – Inference

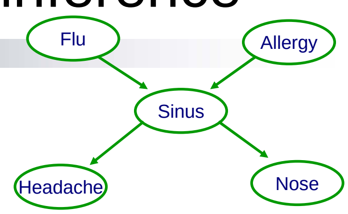
Machine Learning – 10701/15781

Carlos Guestrin

Carnegie Mellon University

November 5th, 2007

General probabilistic inference



- Query: $P(X \mid e)$

$P(A=t \mid H=t, N=t)$

- Using Bayes rule: (Defn. cond. prob.)

$$P(X \mid e) = \frac{P(X, e)}{P(e)}$$

$P(x, e)$

- Normalization:

$$P(X \mid e) \propto P(X, e)$$

$P(A, H=t, N=t) =$

A	
t	0.2
f	0.15

$P(A=t \mid H=t, N=t) = \frac{0.2}{0.2+0.15} = \ldots$

Marginalization

$P(F,S,N) = P(F) \cdot P(S|F) \cdot P(N|S)$

$P(F \mid N=t)$

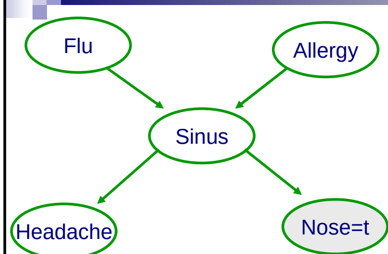
$\propto P(F{=}t, N=t) = P(F=t, S=t, N=t) + P(F=t, S=f, N=t)$

$= \sum_s P(F=t, s, N=t)$

$= \sum_s P(F=t) \cdot P(s \mid F=t) \cdot P(N=t \mid s)$

$= P(F=t) \sum_s P(s \mid F=t)\, P(N=t \mid s)$

Probabilistic inference example



$P(F \mid N=t)\ ? \propto P(F, N=t)$

$P(F,A,S,H,N) = P(F) \cdot P(A)\, P(S|F,A)\, P(N|S)\, P(H|S)$

$P(F, N=t) = \sum_{a,s,h} P(F,a,s,h,N=t)$

$2^3 = 8$ terms

if $\sum_{\substack{x_1, x_2, \ldots x_{54} \\ \text{binary}}}$ 2^{54}

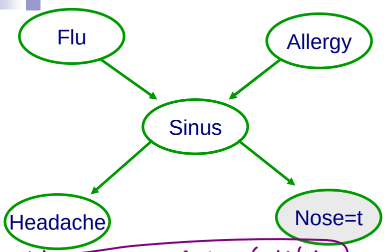
8 sum, in each 4 multiplies

$\simeq$ 32 multiplies

Inference seems exponential in number of variables!
Actually, inference in graphical models is NP-hard ☹

Fast probabilistic inference example – Variable elimination

sums 8
multiplies 8



$$P(F, N=t) = \sum_{a,s,h} P(F) \cdot P(a)\, P(s|F,a)\, P(h|s) \cdot P(N=t|s)$$

$\sum_a \sum_s \sum_h$

2 sums
0 multiplies

$$= \sum_a \sum_s P(F) P(a) P(s|F,a) P(N=t|s) \cancel{\sum_h P(h|s)}^{1}$$

$$= \sum_a P(F)\, P(a) \underbrace{\sum_s P(s|F,a)\, P(N=t|s)}_{g_1(F,a)} = P(N=t|F,a)$$

$\boxed{\text{computing } g_1\text{: for each value of } F,A \ (4) \ \ 1 \text{ sum}, 1 \text{ multiply}}$
4 sums 4 multiplies

$$= P(F) \underbrace{\sum_a P(a) \cdot g_1(F,a)}_{g_2(F)} = \sum_a P(a) \cdot P(N=t|F,a) = P(N=t|F)$$

$\boxed{g_2\text{: 2 values of } F\text{: 1 sum, 1 multiply}}$
2 sums 2 multiplies

$$P(F, N=t) = P(F) \cdot g_2(F) = P(F) \cdot P(N=t|F)$$

1 multiplic for each value of F

(Potential for) Exponential reduction in computation!

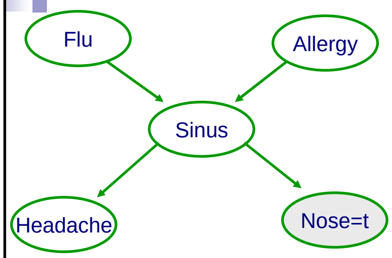
Understanding variable elimination – Exploiting distributivity

$a(b+c) = ab + ac$

$$P(F=t, N=t) = P(F=t) \cdot P(S=t|F=t) \cdot P(N=t|S=t) + P(F=t)\, P(S=f|F=t) \cdot P(N=t|S=f)$$

$$= P(F=t)\left[P(S=t|F=t) \cdot P(N=t|S=t) + P(S=f|F=t) \cdot P(N=t|S=f)\right]$$

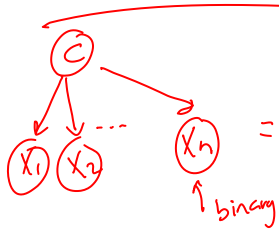
Understanding variable elimination – Order can make a HUGE difference



$P(F, N=t)$

$= \sum_{a,s,h} P(F) \cdot P(a)\, P(s|F,a) \cdot P(h|s) \cdot P(N=t|s)$

$= \sum_{a,h} P(F) \cdot P(a) \underbrace{\sum_{s} P(s|F,a)\, P(h|s) \cdot P(N=t|s)}_{g_1(F,a,h)}$



$P(X_n) = \sum_{c, x_1, x_2, \ldots x_{n-1}} P(c) \cdot \prod_i P(x_i|c)$

$= \sum_{x_1 \ldots x_{n-1}} \underbrace{\sum_{C} P(c) \prod_i P(x_i|c)}_{g_1(x_1, \ldots x_n) \Leftarrow 2^n}$

opt. order
X_1
$\vdots$
X_{n-1}
C

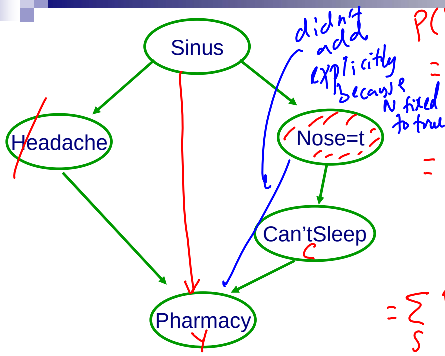
eliminating X_i only

$g_i(c) = \sum_{x_i} P(x_i|c)$

Understanding variable elimination – Another example

order: H C S

$g_i \Leftarrow$ factor



didn't add explicitly because N fixed to true

$P(Y, N=t)$

$= \sum_{s,h,c} P(s) \cdot P(h|s) \cdot P(N=t|s) \cdot P(c|N=t) \cdot P(y|h,c)$

$= \sum_{s,c} P(s)\, P(N=t|s)\, P(c|N=t) \underbrace{\sum_{h} P(h|s) \cdot P(y|h,c)}_{g_1(s,y,c) = P(y|s,c)}$

$= \sum_{s} P(s)\, P(N=t|s) \underbrace{\sum_{c} P(c|N=t) \cdot g_1(s,y,c)}_{g_2(s,y)}$

$= \sum_{s} P(s)\, P(N=t|s) \cdot g_2(s,y)$

eliminate variables
"lose" independencies:
a priori $S \perp Y | H, N$
but after eliminate H
$\neg\ S \perp Y | N$
$\neg\ S \perp Y$

Variable elimination algorithm

- Given a BN and a query P(X|e) query $\propto$ P(X,e)
- Instantiate evidence e in every CPT **IMPORTANT!!!**
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \{X,e\}$, eliminate X_i
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$

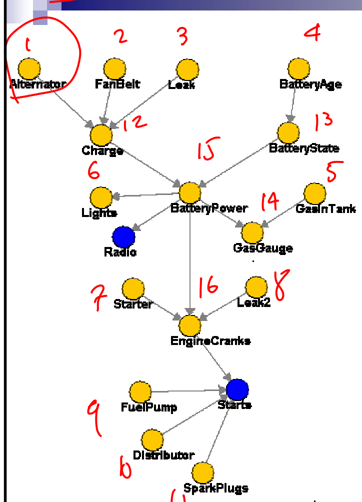
e.g., $g_7(x_2,x_4) = \sum_{x_7} g_3(x_2,x_4,x_7) \cdot P(x_4|x_7)$

  - Variable X_i has been eliminated!
- Normalize P(X,e) to obtain P(X|e)

Complexity of variable elimination – (Poly)-tree graphs

(no loops in graph)

inference is linear time



Variable elimination order:
Start from "leaves" up –
find topological order, eliminate variables in reverse order

eliminate A

$\sum_a P(a) \cdot P(c|a) = g_1(c)$

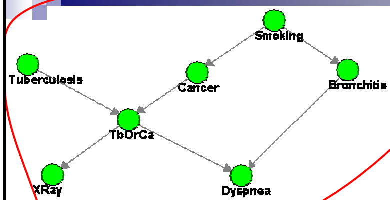
never generate any factor that is larger than an original CPT

$\Rightarrow$ running time linear in number of nodes n

if all vars were binary 2^n

Linear in number of variables!!! (versus exponential)

Complexity of variable elimination – Graphs with loops



g

$$g = \sum_{x_i} \prod_{j=1}^{k} f_j$$

g

how many vars appear in g:

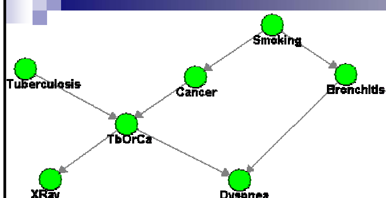
how many?

$g(x_1, x_2, \ldots, x_m) \leftarrow 2^m$ terms

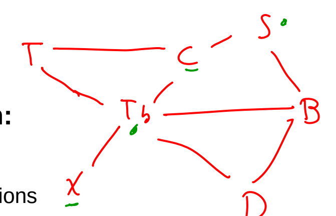
↑ binary

Exponential in number of variables in largest factor generated

Complexity of variable elimination – Tree-width



Moralize graph:
Connect parents into a clique and remove edge directions



Treewidth: $\forall$ $\max_{i,j}$ $X_i \subseteq X, X_j \subseteq X$ (not neighbors) x_i, x_j :

$= 3$

min number nodes I remove to separate them

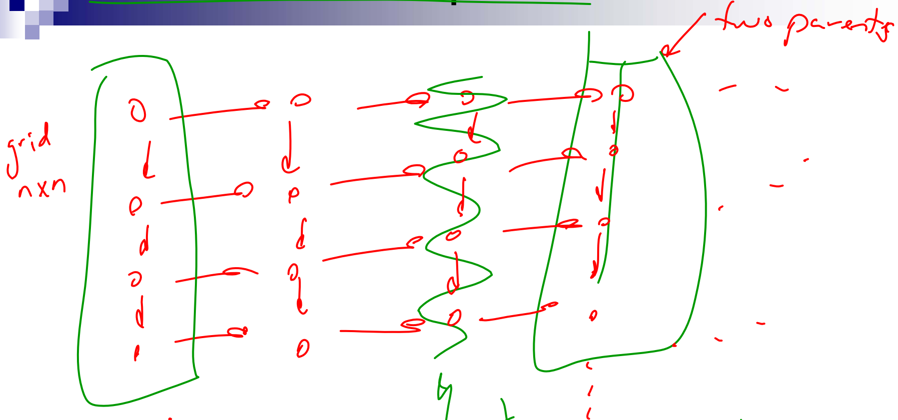
+1

g

cut you need ≤ 2

Complexity of VE elimination:
(“Only”) exponential in tree-width
Tree-width is maximum node cut +1

Example: Large tree-width with small number of parents



must remove whole column, tree width is n+1

inference hard on grids, and many other graphs: approx. inf.

Compact representation $\not\Rightarrow$ Easy inference ☹

Choosing an elimination order

- Choosing best order is NP-complete
  - Reduction from MAX-Clique
- Many good heuristics (some with guarantees)
- Ultimately, can't beat NP-hardness of inference
  - Even optimal order can lead to exponential variable elimination computation
- In practice
  - Variable elimination often very effective
  - Many (many many) approximate inference approaches available when variable elimination too expensive

Announcements

- HW4 out later today
  will come out in two installments, no interest
  no hidden fees
- Project milestone
  - Next Monday (11/12 in class)

HMMs

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

November 5th, 2007

Adventures of our BN hero

- Compact representation for probability distributions
- Fast inference
- Fast learning

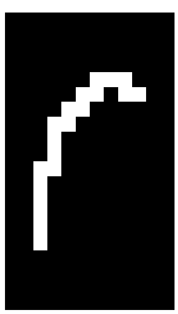
- But… Who are the most popular kids?

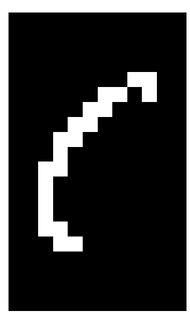
1. Naïve Bayes

2 and 3.
Hidden Markov models (HMMs)
Kalman Filters

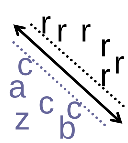
HMM with Gaussian

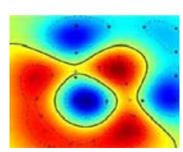
Handwriting recognition





Character recognition, e.g., kernel SVMs





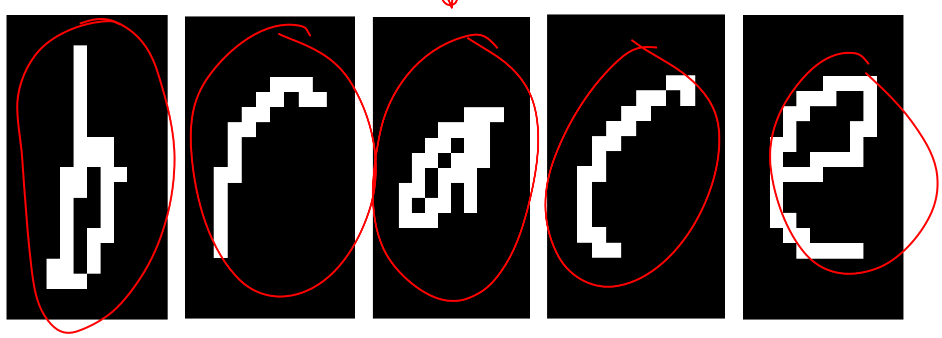
Example of a hidden Markov model (HMM)

$P(X_3|X_2) \leftarrow P(X_3=a|X_2=r)$

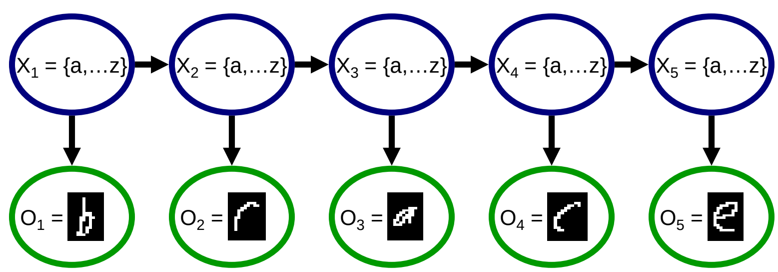
prob an a follows an r in third position

$\{a,\ldots,z\}$

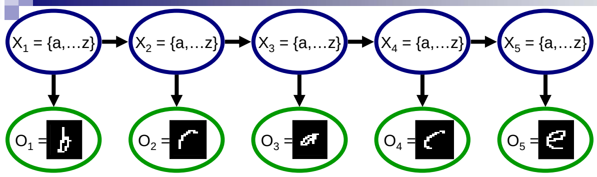
$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow X_5$



Understanding the HMM Semantics



HMMs semantics: Details



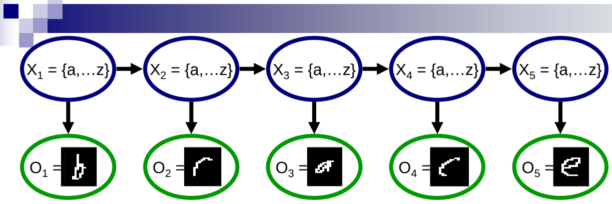
Just 3 distributions:

$$P(X_1)$$

$$P(X_i \mid X_{i-1})$$

$$P(O_i \mid X_i)$$

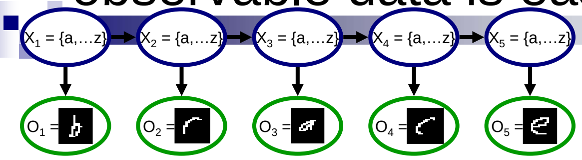
HMMs semantics: Joint distribution



$P(X_1)$
$P(X_i \mid X_{i-1})$
$P(O_i \mid X_i)$

$$P(X_1, \ldots, X_n \mid o_1, \ldots, o_n) = P(X_{1:n} \mid o_{1:n})$$
$$\propto P(X_1)P(o_1 \mid X_1) \prod_{i=2}^{n} P(X_i \mid X_{i-1})P(o_i \mid X_i)$$

Learning HMMs from fully observable data is easy



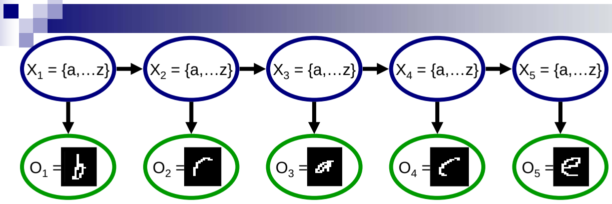
Learn 3 distributions:

$$P(X_1)$$

$$P(O_i \mid X_i)$$

$$P(X_i \mid X_{i-1})$$

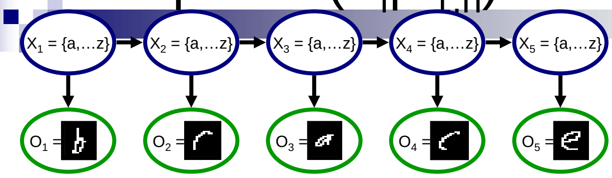
Possible inference tasks in an HMM



Marginal probability of a hidden variable:

Viterbi decoding – most likely trajectory for hidden vars:

Using variable elimination to compute $P(X_i|o_{1:n})$



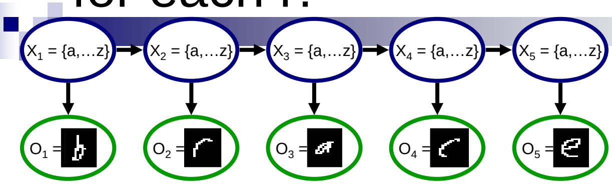
Compute:

$$P(X_i \mid o_{1..n})$$

Variable elimination order?

Example:

What if I want to compute $P(X_i|o_{1:n})$ for each i?



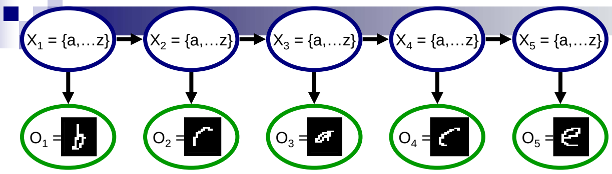
Compute:

$$P(X_i \mid o_{1..n})$$

Variable elimination for each i?

Variable elimination for each i, what's the complexity?

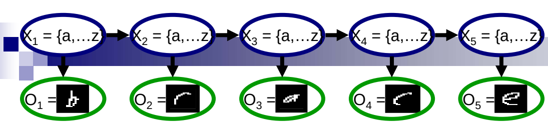
Reusing computation



Compute:

$$P(X_i \mid o_{1..n})$$

The forwards-backwards algorithm



$$P(X_i \mid o_{1..n})$$

- Initialization: $\alpha_1(X_1) = P(X_1)P(o_1 \mid X_1)$
- For i = 2 to n
  - Generate a forwards factor by eliminating X_{i-1}

$$\alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})$$

- Initialization: $\beta_n(X_n) = 1$
- For i = n-1 to 1
  - Generate a backwards factor by eliminating X_{i+1}

$$\beta_i(X_i) = \sum_{x_{i+1}} P(o_{i+1} \mid x_{i+1})P(x_{i+1} \mid X_i)\beta_{i+1}(x_{i+1})$$

- 8 i, probability is: $\boxed{P(X_i \mid o_{1..n}) \propto \alpha_i(X_i)\beta_i(X_i)}$

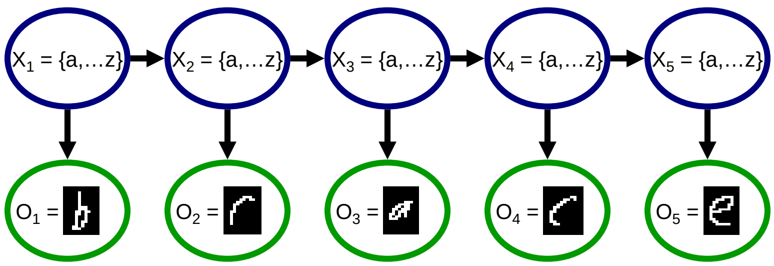
What you'll implement 1: multiplication

$$\alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i) P(X_i \mid X_{i-1} = x_{i-1}) \alpha_{i-1}(x_{i-1})$$

What you'll implement 2: marginalization

$$\alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i) P(X_i \mid X_{i-1} = x_{i-1}) \alpha_{i-1}(x_{i-1})$$

Higher-order HMMs



Add dependencies further back in time !

better representation, harder to learn

What you need to know

- Hidden Markov models (HMMs)
  - Very useful, very powerful!
  - Speech, OCR,…
  - Parameter sharing, only learn 3 distributions
  - Trick reduces inference from $O(n^2)$ to $O(n)$
  - Special case of BN