

# Bayesian Networks – Inference

Machine Learning – 10701/15781  
 Carlos Guestrin  
 Carnegie Mellon University

November 5<sup>th</sup>, 2007

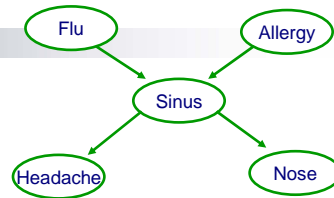
©2005-2007 Carlos Guestrin

1

## General probabilistic inference

■ Query:  $P(X | e)$

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



■ Using Bayes rule: (Defn. cond. probs.)

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

$$P(x, e)$$

■ Normalization:

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

$$P(A, H=t, N=t) = \begin{array}{l|l} A & \\ \hline t & 0.2 \\ f & 0.15 \end{array}$$

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

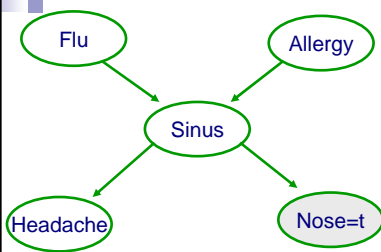
# Marginalization



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

$$\begin{aligned}
 & P(F|N=t) \\
 \propto & P(F, N=t) = P(F=t, S=t, N=t) + P(F=t, S=f, N=t) \\
 & = \sum_s P(F=t, s, N=t) \\
 & = \sum_s \underbrace{P(F=t)} \cdot P(s|F=t) \cdot P(N=t|s) \\
 & = P(F=t) \sum_s P(s|F=t) P(N=t|s)
 \end{aligned}$$

# Probabilistic inference example



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

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

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

$\sim 2^3 = 8 \text{ terms}$

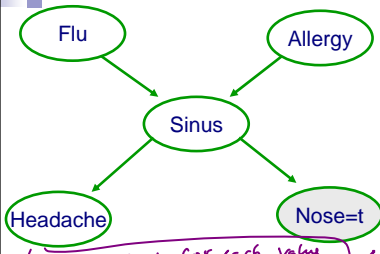
if  $\sum_{x_1, x_2, \dots, x_n} \dots$   
 binary  $\leftarrow 2^5$

8 sum, in each  
 4 multiplies  
 $\sim 32$  multiplies

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

# Fast probabilistic inference example – Variable elimination

sums & multipliers



$$\begin{aligned}
 P(F, N=t) &= \sum_a P(F) \cdot P(a) P(s|F,a) P(h|s) \cdot P(N=t|s) \\
 &= \sum_a \sum_s P(F) P(a) P(s|F,a) P(N=t|s) \sum_h P(h|s) \\
 &= \sum_a P(F) P(a) \sum_s P(s|F,a) P(N=t|s) \\
 &= P(F) \sum_a P(a) \cdot g_1(F,a)
 \end{aligned}$$

Computing  $g_1$ : for each value of  $F, a$  4 sums, 4 multiplies

$g_2$ : 2 values of  $F$ ; 1 sum, 2 multiplies

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

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

multiply for each value of  $F$

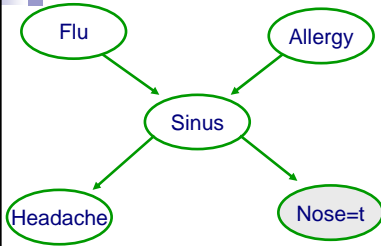
**(Potential for) Exponential reduction in computation!**

# Understanding variable elimination – Exploiting distributivity $a(b+c) = ab+ac$

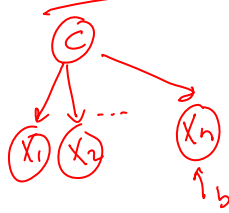


$$\begin{aligned}
 P(F=t, N=t) &= P(F=t) \cdot P(S=t|F=t) \cdot P(N=t|S=t) + \\
 &\quad P(F=t) P(S=f|F=t) \cdot P(N=t|S=f) \\
 &= P(F=t) [P(S=t|F=t) \cdot P(N=t|S=t) + P(S=f|F=t) \cdot P(N=t|S=f)]
 \end{aligned}$$

# Understanding variable elimination – Order can make a HUGE difference



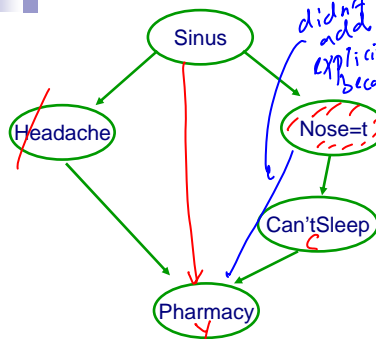
$$\begin{aligned}
 P(F, N=t) &= \sum_{a, s, h} P(F) \cdot P(a) \cdot P(s|F, a) \cdot P(h|s) \cdot P(N=t|s) \\
 &= \sum_{a, h} P(F) \cdot P(a) \underbrace{\sum_s P(s|F, a) P(h|s)}_{g_1(F, a, h)} \cdot P(N=t|s)
 \end{aligned}$$



$$\begin{aligned}
 P(X_n) &= \sum_{c, x_1, x_2, \dots, x_{n-1}} P(c) \cdot \prod_i P(x_i|c) \\
 &= \sum_{x_1, x_2, \dots, x_{n-1}} \underbrace{\sum_c P(c) \prod_i P(x_i|c)}_{g_1(x_1, \dots, x_{n-1})}
 \end{aligned}$$

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: HCS  
 $g_i \leftarrow$  factor

$$\begin{aligned}
 P(Y, N=t) &= \sum_{s, h, c} P(s) \cdot P(h|s) \cdot P(N=t|s) \cdot P(c|N=t) \cdot P(Y|h, c) \\
 &= \sum_{s, c} P(s) P(N=t|s) P(c|N=t) \underbrace{\sum_h P(h|s) \cdot P(Y|h, c)}_{g_1(s, Y, c) = P(Y|s, c)} \\
 &= \sum_s P(s) P(N=t|s) \underbrace{\sum_c P(c|N=t) \cdot g_1(s, Y, c)}_{g_2(s, Y)} \\
 &= \sum_s P(s) P(N=t|s) \cdot g_2(s, Y)
 \end{aligned}$$

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

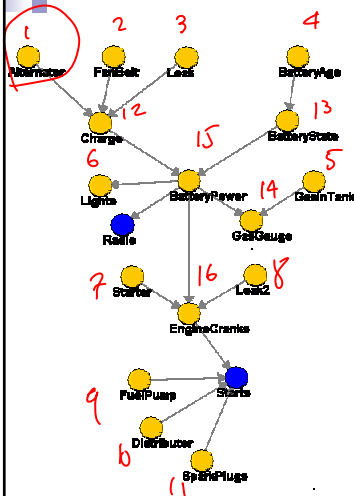
$$g = \sum_{X_i} \prod_{j=1}^k f_j$$

*eg,  $g_7(X_2, X_4) = \sum_{X_7} g_3(X_2, X_4, X_7) \cdot P(X_7|Z_2)$*

- Variable  $X_i$  has been eliminated!

- Normalize  $P(X,e)$  to obtain  $P(X|e)$

## Complexity of variable elimination – (Poly)-tree graphs



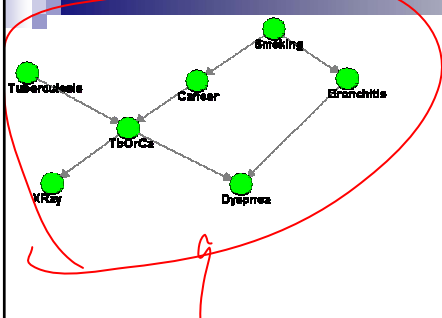
**Variable elimination order:**  
Start from "leaves" up –  
find topological order, eliminate  
variables in reverse order

*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  $\rightarrow 2^n$*

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

# Complexity of variable elimination – Graphs with loops

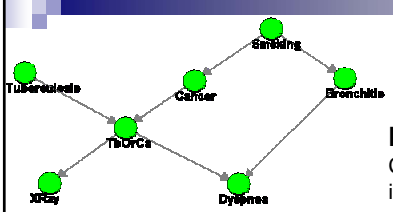


$$g = \sum_{x_i} \prod_{j=1}^k f_j$$

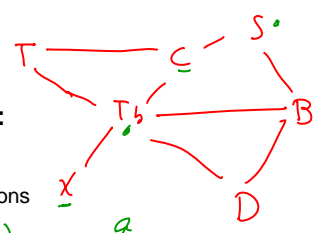
how many vars appear in  $g$ :  
 $g(x_1, x_2, \dots, x_m) \in 2^m$  terms  
 ↑ binary

**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

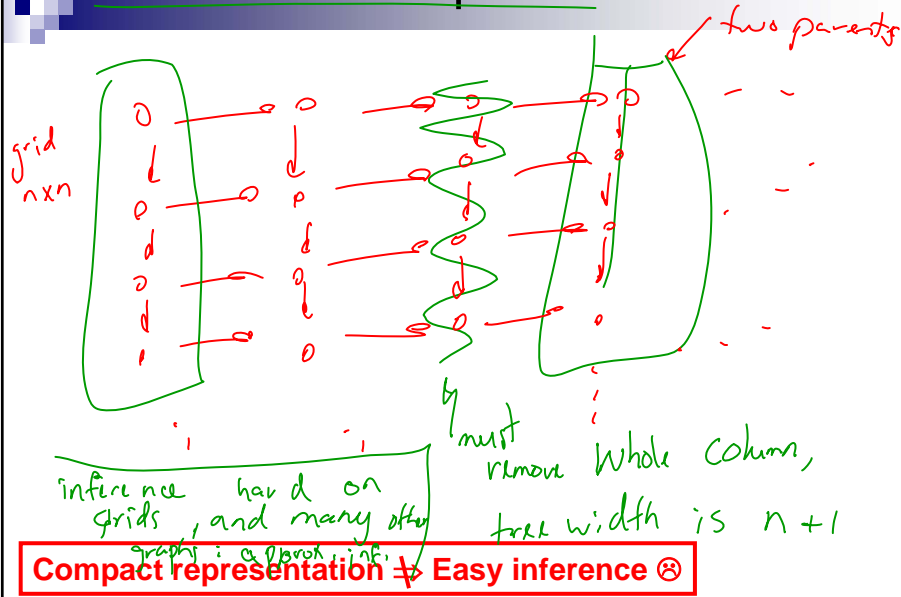


Tree-width:  $\# X_i, X_j$  (not neighbors)  
 $= 3$   
 min number nodes  $\uparrow$  remove to separate from  $+1$

cut you need  $\leq 2$

**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



## 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
  - $\rightarrow$  □ Variable elimination often very effective
  - $\rightarrow$  □ Many (many many) approximate inference approaches available when variable elimination too expensive

# Announcements

- HW4 out later today

*It 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

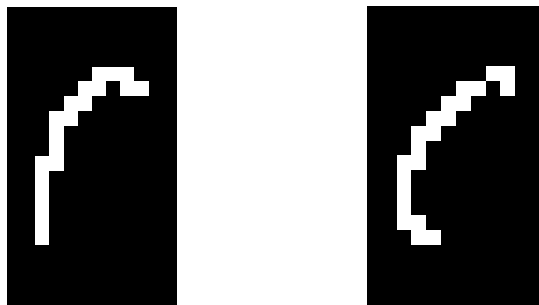
Carnegie Mellon University

November 5<sup>th</sup>, 2007

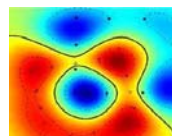
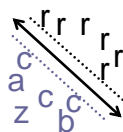
# Adventures of our BN hero

- 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 Gaussians*

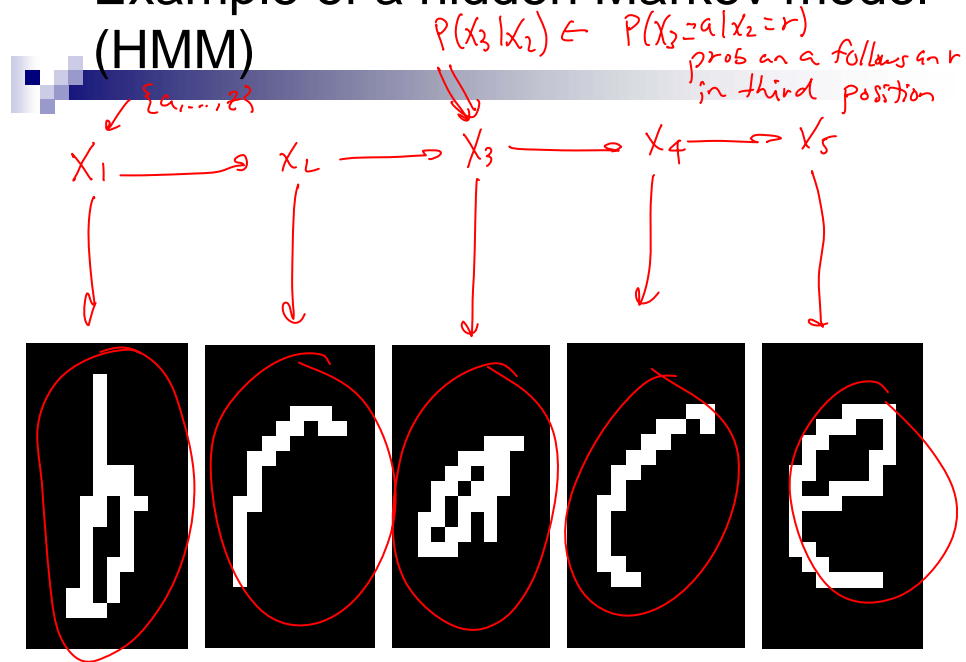
# Handwriting recognition



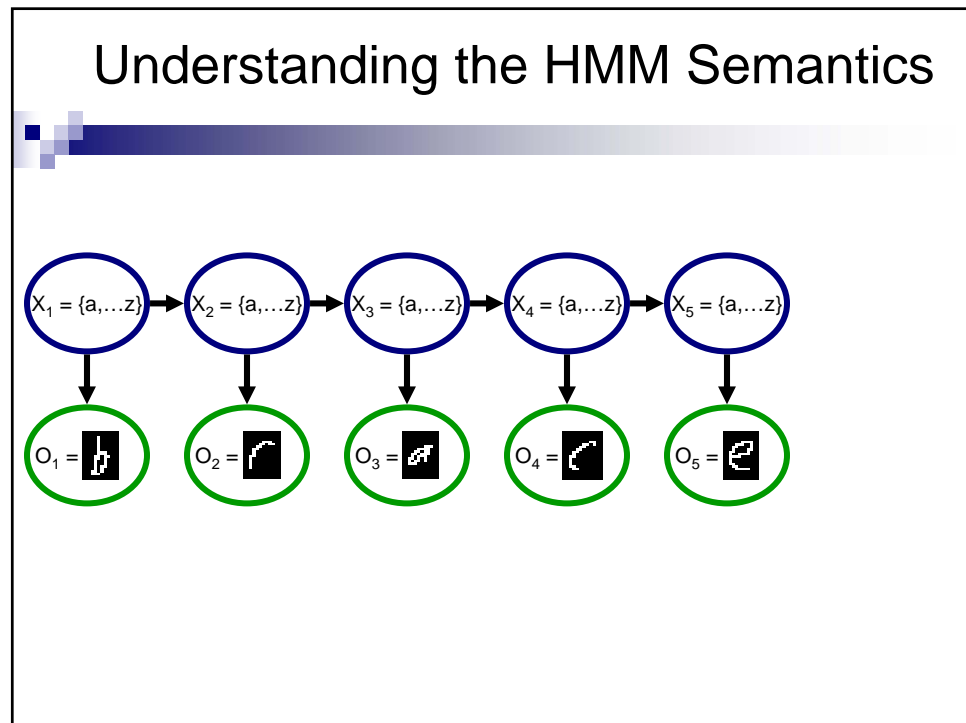
Character recognition, e.g., kernel SVMs



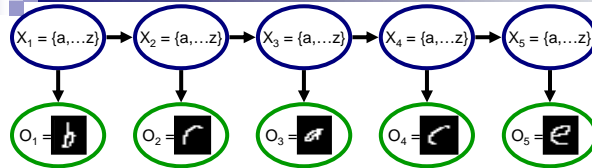
# Example of a hidden Markov model (HMM)



## Understanding the HMM Semantics



## HMMs semantics: Details



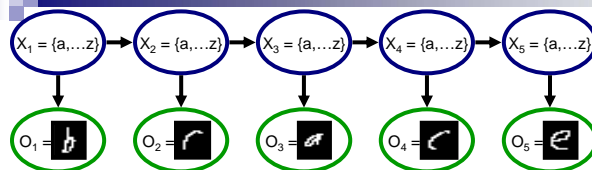
Just 3 distributions:

$$P(X_1)$$

$$P(X_i | X_{i-1})$$

$$P(O_i | X_i)$$

## HMMs semantics: Joint distribution



$$P(X_1)$$

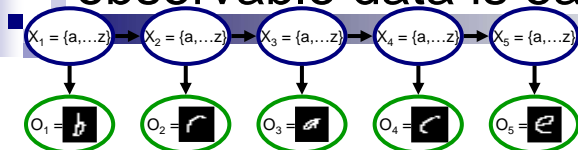
$$P(X_i | X_{i-1})$$

$$P(O_i | X_i)$$

$$P(X_1, \dots, X_n | o_1, \dots, o_n) = P(X_{1:n} | o_{1:n})$$

$$\propto P(X_1)P(o_1 | X_1) \prod_{i=2}^n P(X_i | X_{i-1})P(o_i | X_i)$$

## Learning HMMs from fully observable data is easy



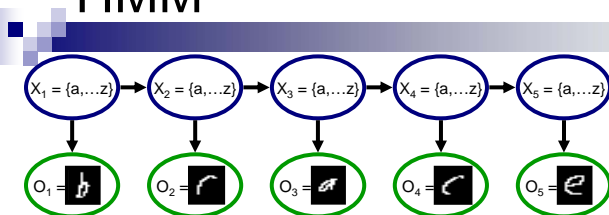
Learn 3 distributions:

$$P(X_1)$$

$$P(O_i | X_i)$$

$$P(X_i | X_{i-1})$$

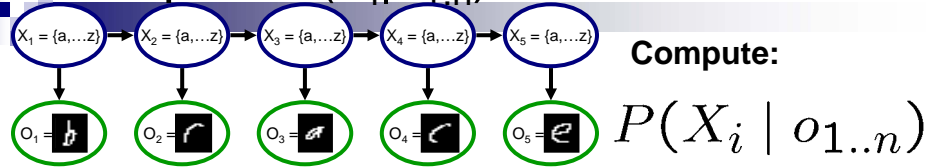
## Possible inference tasks in an HMM



Marginal probability of a hidden variable:

Viterbi decoding – most likely trajectory for hidden vars:

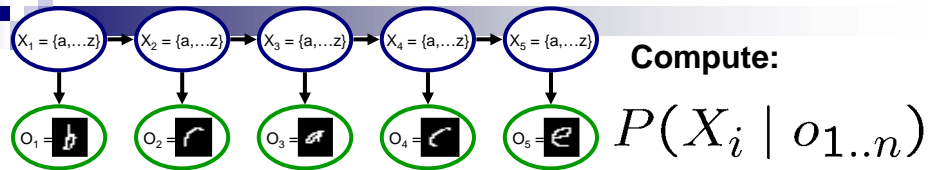
Using variable elimination to compute  $P(X_i | o_{1:n})$



Variable elimination order?

Example:

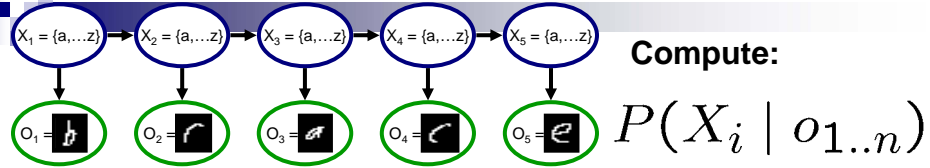
What if I want to compute  $P(X_i | o_{1:n})$  for each  $i$ ?



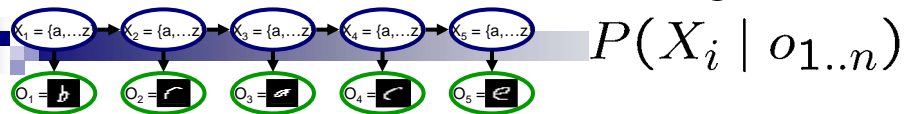
Variable elimination for each  $i$ ?

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

## Reusing computation



## The forwards-backwards algorithm



■ Initialization:  $\alpha_1(X_1) = P(X_1)P(o_1 | X_1)$

■ For  $i = 2$  to  $n$

□ Generate a forwards factor by eliminating  $X_{i-1}$

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

■ Initialization:  $\beta_n(X_n) = 1$

■ For  $i = n-1$  to  $1$

□ Generate a backwards factor by eliminating  $X_{i+1}$

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

■ For  $i$ , probability is:  $P(X_i | o_{1..n}) \propto \alpha_i(X_i)\beta_i(X_i)$

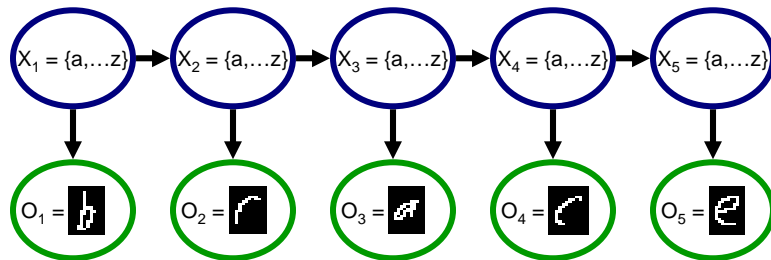
## What you'll implement 1: multiplication

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

## What you'll implement 2: marginalization

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

## 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