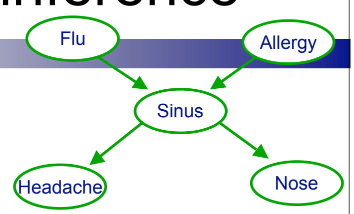
# Bayesian Networks – Inference

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

November 5th, 2007




## General probabilistic inference




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

- Using Bayes rule:

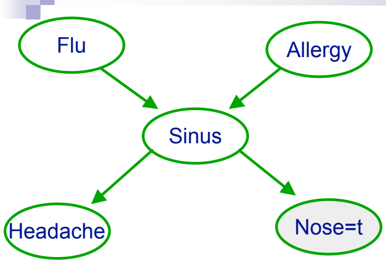
$$P(X \mid e) = \frac{P(X, e)}{P(e)}$$

- Normalization:

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

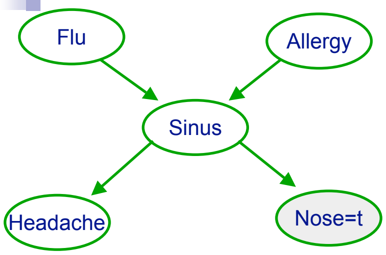
# Marginalization


# Probabilistic inference example




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

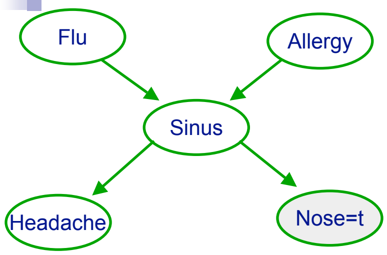
## Fast probabilistic inference example – Variable elimination



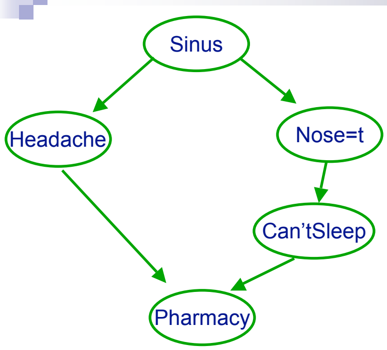

**(Potential for) Exponential reduction in computation!**

## Understanding variable elimination – Exploiting distributivity

# Understanding variable elimination – Order can make a HUGE difference




# Understanding variable elimination – Another example



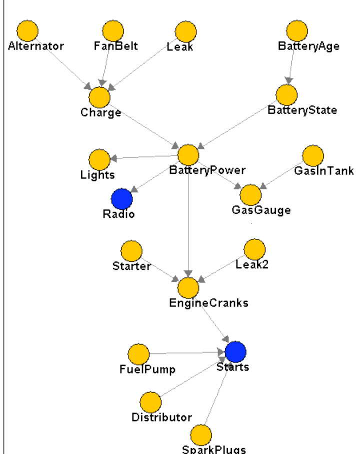
## Variable elimination algorithm

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

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

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

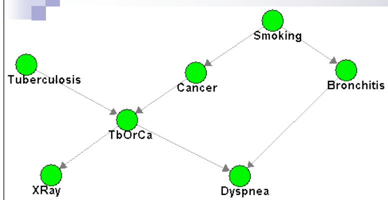
## Complexity of variable elimination – (Poly)-tree graphs




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

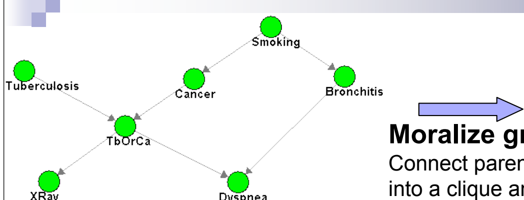
**Linear in number of variables!!! (versus exponential)**

# Complexity of variable elimination – Graphs with loops




**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

**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

**Compact representation ⇏ 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
- 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

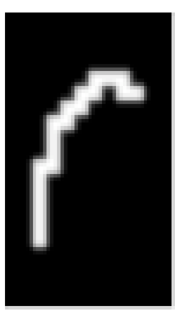
- Compact representation for probability distributions
- Fast inference
- Fast learning

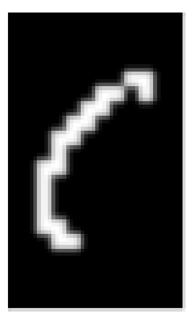
- But… Who are the most popular kids?

**1. Naïve Bayes**

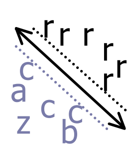
**2 and 3.**
**Hidden Markov models (HMMs)**
**Kalman Filters**

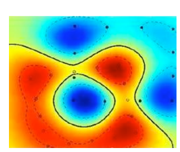
# Handwriting recognition



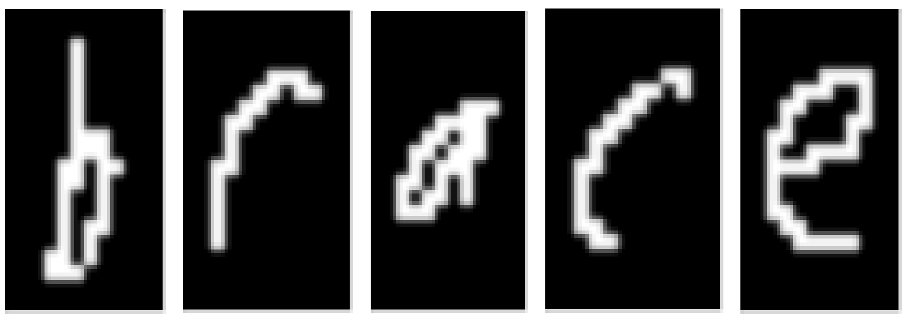


Character recognition, e.g., kernel SVMs

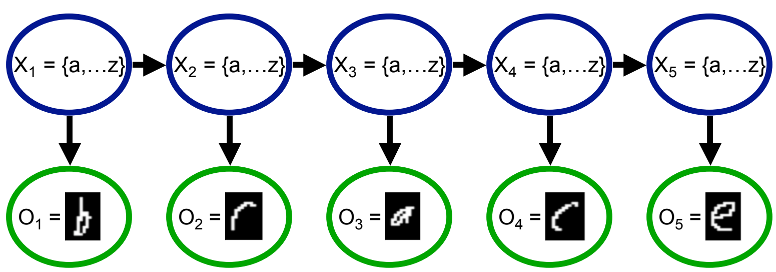




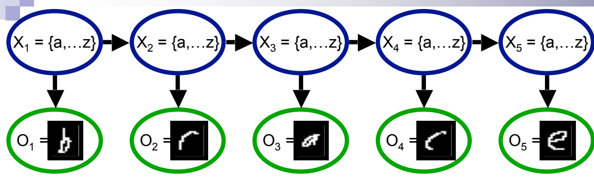
# Example of a hidden Markov model (HMM)



# Understanding the HMM Semantics



# HMMs semantics: Details



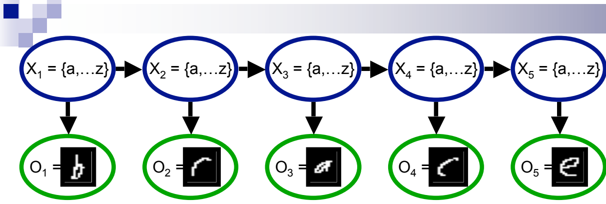

**Just 3 distributions:**

$$P(X_1)$$

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

$$P(O_i \mid X_i)$$

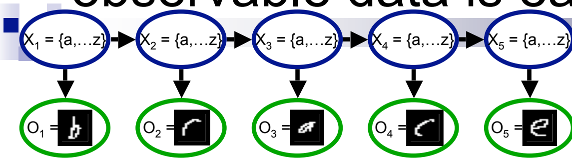
# HMMs semantics: Joint distribution




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

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

# Learning HMMs from fully observable data is easy



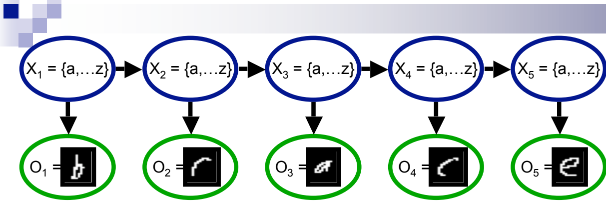

**Learn 3 distributions:**

$$P(X_1)$$

$$P(O_i \mid X_i)$$

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

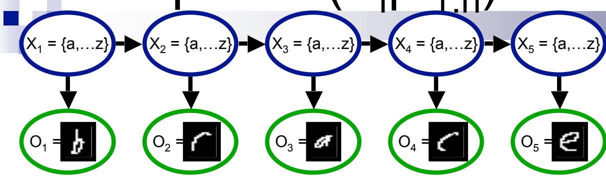
# Possible inference tasks in an HMM




**Marginal probability of a hidden variable:**

**Viterbi decoding – most likely trajectory for hidden vars:**

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



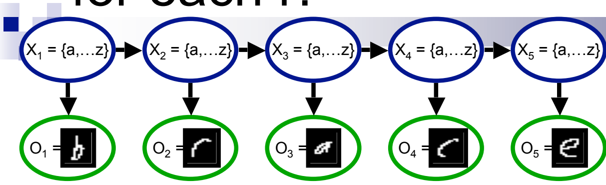

**Compute:**

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

**Variable elimination order?**

**Example:**

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



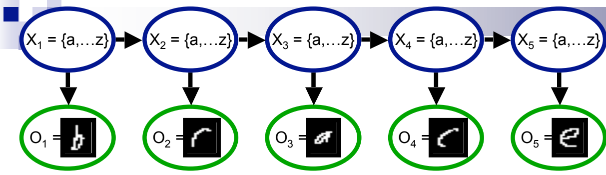

**Compute:**

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

**Variable elimination for each i?**

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

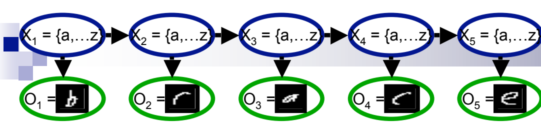
# Reusing computation




**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 $X_{i+1}$

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

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

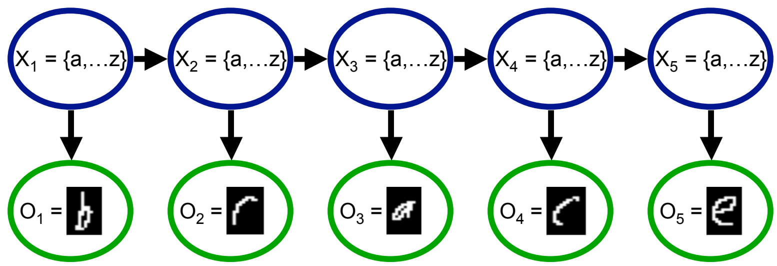
# What you'll implement 1: multiplication

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

# 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