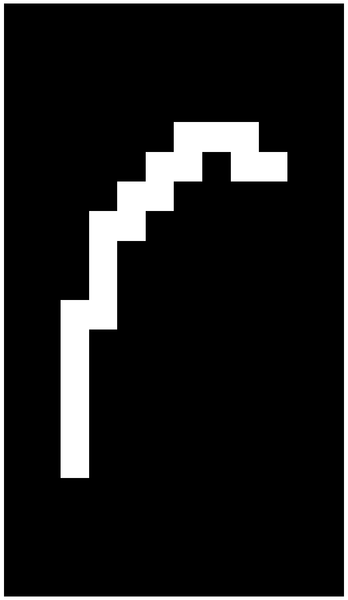
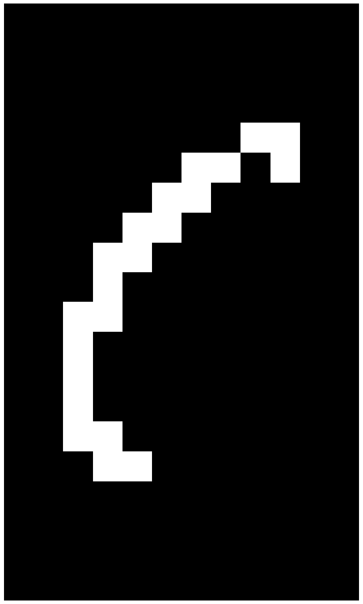
Hidden Markov Models

Machine Learning 10-601
March 17, 2008

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

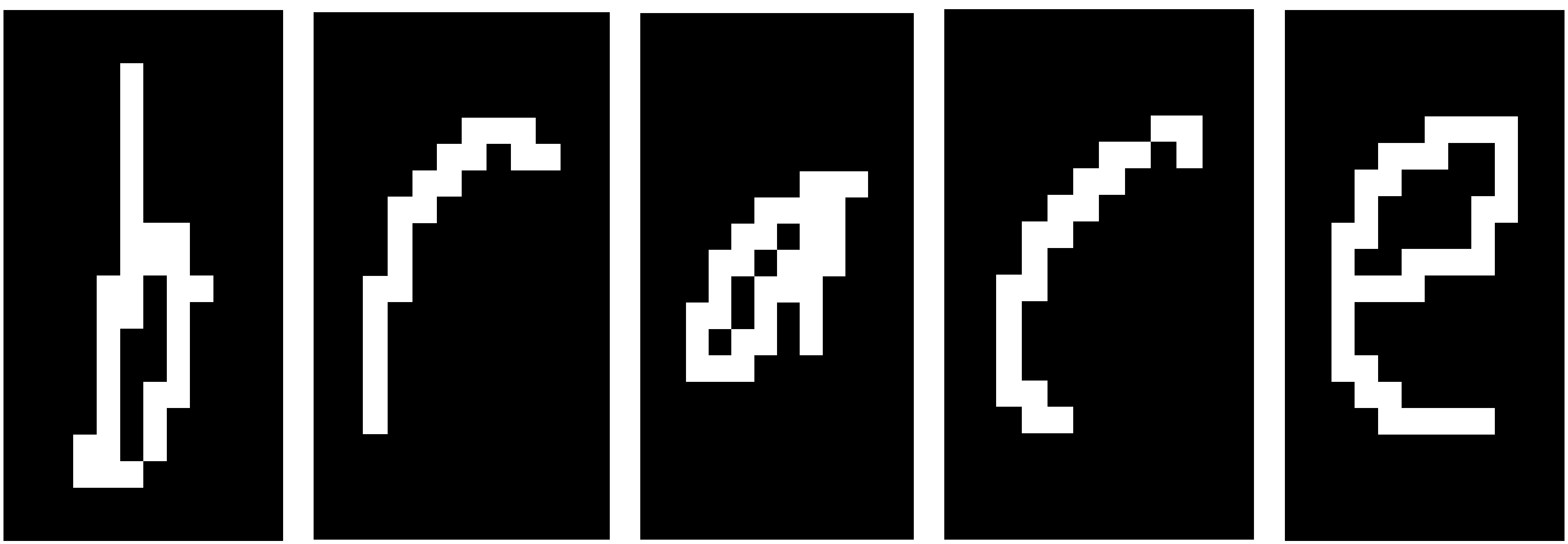
With thanks to Prof. Carlos Guestrin for some of these slides

Handwriting recognition

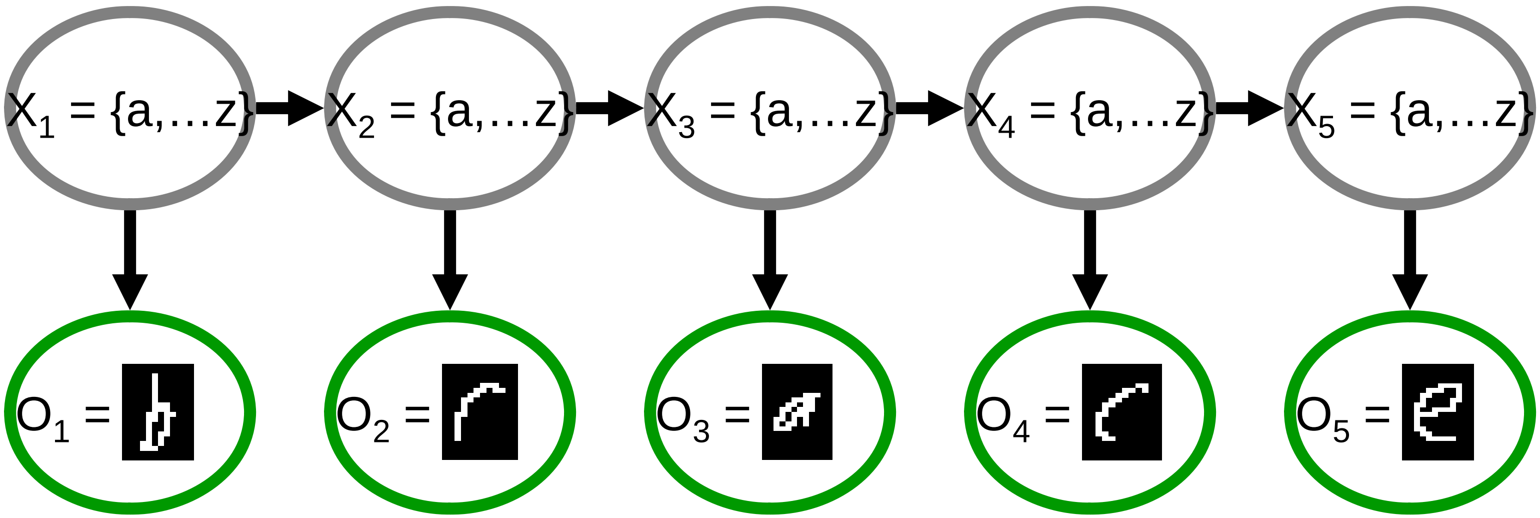
 

Character recognition, e.g., logistic regression

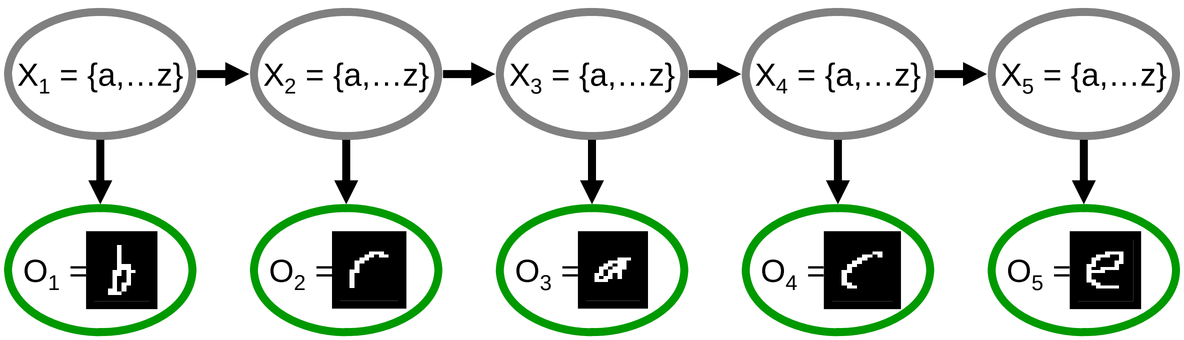
Example of a hidden Markov model (HMM)



Understanding the HMM Semantics



HMMs semantics: Details



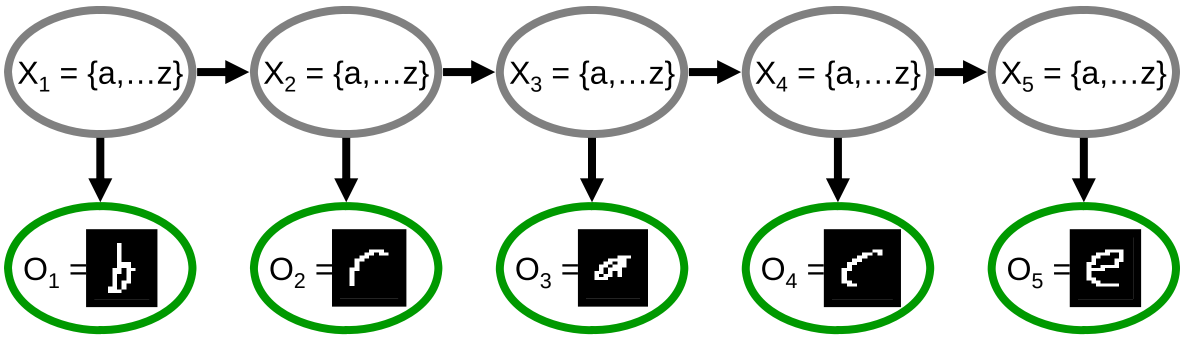
Just 3 distributions:

$$P(X_1)$$

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

$$P(O_i \mid X_i)$$

Using and Learning HMM's



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

Core HMM questions:

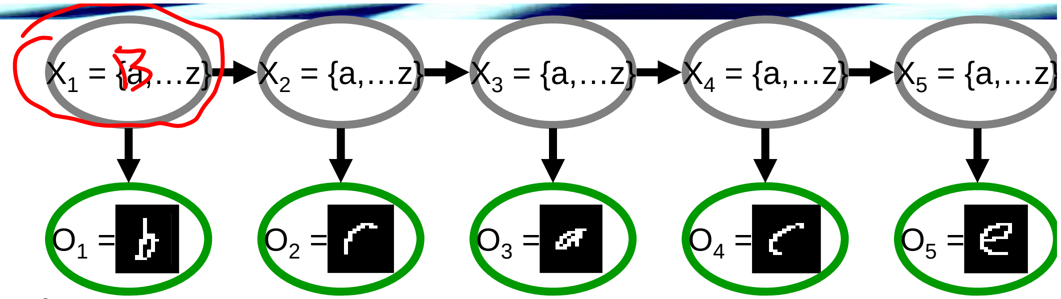
1. How do we calculate $P(o_1, o_2, \ldots o_n)$?

2. How do we calculate argmax over $x_1, x_2, \ldots x_n$ of $P(x_1, x_2, \ldots x_n \mid o_1, o_2, \ldots o_n)$?

3. How do we train the HMM, given its structure and
   3a. Fully observed training examples: $< x_1, \ldots x_n, o_1, \ldots o_n >$

   3b. Partially observed training examples: $< o_1, \ldots o_n >$

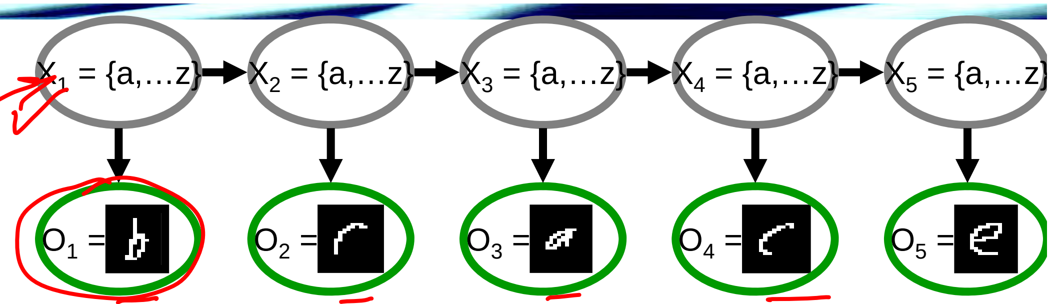
How do we generate a random output sequence following the HMM $P(o_1, o_2, \ldots o_T)$



How do we compute

$$P(o_1, o_2, \ldots o_T)$$

N poss values



1 2 . . . T

1. brute force:

2. Forward algorithm (dynamic progr., variable elimination):

$N^2 T$

define $\alpha_t(k) = \boxed{P(o_1, o_2, \ldots o_t, X_t = k)}$

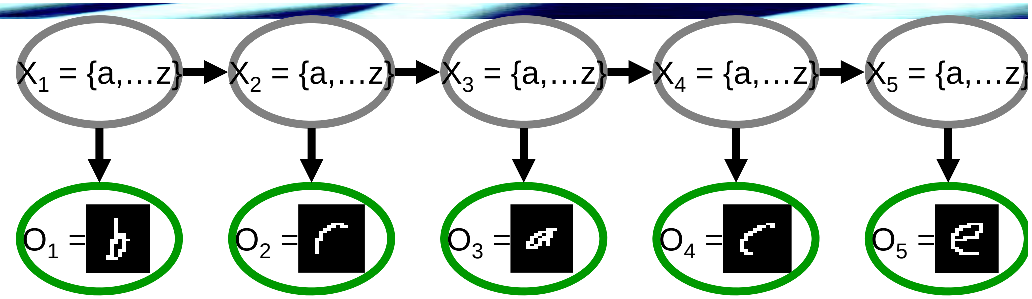
$$\alpha_1(k) = P(X_1=k)\, P(O_1=o_1 \mid X_1=k) = P(o_1, X_1=k)$$

$$\alpha_{t+1}(k) = \left[\sum_{j=1}^{N} \alpha_t(j)\, P(X_{t+1}=k \mid X_t=j)\right] P(O_{t+1}=o_{t+1} \mid X_{t+1}=k)$$

$$P(o_1, o_2, \ldots o_T) = \sum_k \alpha_T(k)$$

How do we compute

$$P(X_t = k | o_1, o_2, \ldots o_T)$$



2. Backward algorithm (dynamic progr., variable elimination):

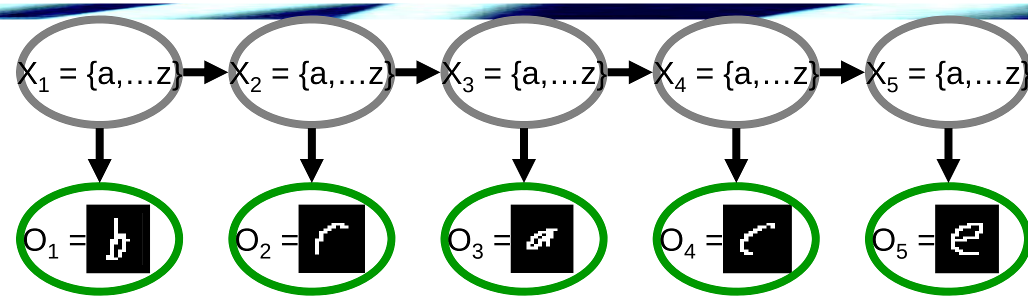
$$\alpha_t(k) = P(o_1, o_2, \ldots o_t, X_t = k)$$

define $\beta_t(k) = P(o_{t+1}, o_{t+2}, \ldots o_T | X_t = k)$

$\alpha_t(k)\,\beta_t(k)$

$$P(X_t = k | o_1, o_2, \ldots o_T) = \frac{P(X_t = k, o_1, o_2, \ldots o_T)}{P(o_1, o_2, \ldots o_T)}$$

$\sum_k \alpha_T(k)$



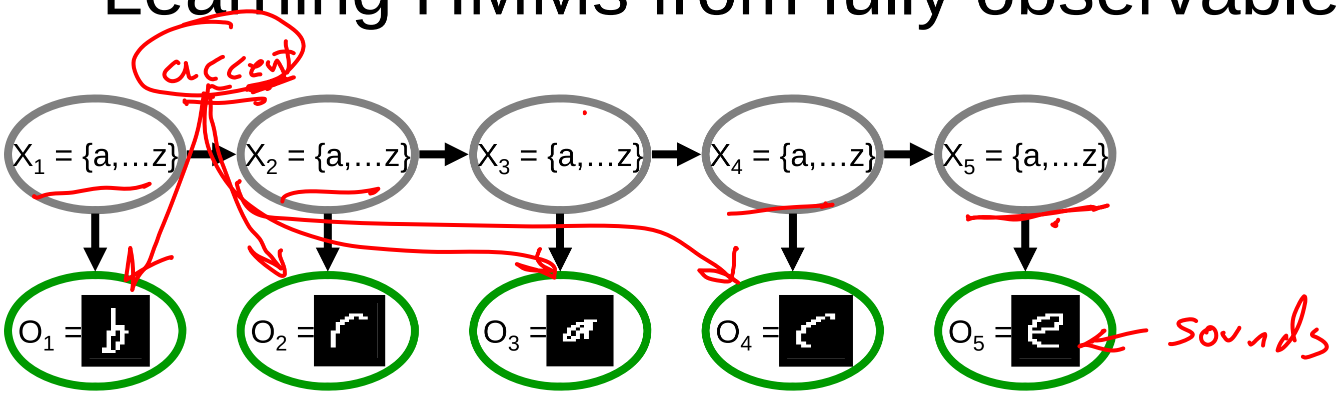
How do we compute

$$\arg\max_{x_1,\ldots x_T} P(x_1,\ldots x_T \mid o_1, o_2, \ldots o_T)$$

Viterbi algorithm, based on recursive computation of

$$\delta_t(k) = \max_{x_1,\ldots x_{t-1}} P(x_1,\ldots x_{t-1} X_t = k, o_1, o_2, \ldots o_t)$$

Learning HMMs from fully observable data: easy



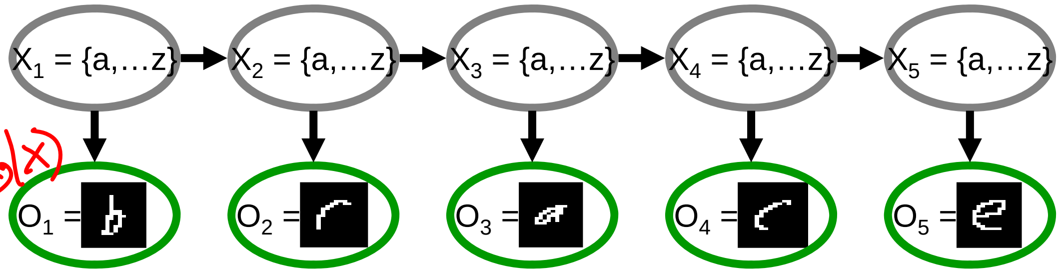
Learn 3 distributions:

$$P(X_1)$$

$$P(O_i \mid X_i)$$

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

Learning HMMs when only observe o1...oT



$P(o|x)$

} unobs Z

} obs Y

EM

Initial model params θ

Loop to convergence.

Estep: use current θ to calc. expected values of Z

(eg. Viterbi alg)

$E[\#\text{ times trans from } x_i \text{ to } x_j]$

$E[\#\text{ times in } x_i]$

$E[\#\text{ times in } x_i \text{ and obs } o_j]$

M step re-maximize likelihood by choosing θ'

$$\theta' \leftarrow \underset{\theta'}{\arg\max}\ E\left[P(o_1, o_2 \dots o_T, \hat{x}_1, \dots \hat{x}_T \mid \theta')\right]$$

$P(z \mid Y, \theta)$

What you need to know

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

Thanks to Carlos Guestrin for many slides