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

\[ X_1 = \{a, \ldots, z\} \]
\[ X_2 = \{a, \ldots, z\} \]
\[ X_3 = \{a, \ldots, z\} \]
\[ X_4 = \{a, \ldots, z\} \]
\[ X_5 = \{a, \ldots, z\} \]

\[ O_1 = \text{[Image]} \]
\[ O_2 = \text{[Image]} \]
\[ O_3 = \text{[Image]} \]
\[ O_4 = \text{[Image]} \]
\[ O_5 = \text{[Image]} \]
HMMs semantics: Details

Just 3 distributions:

\[ P(X_1) \]
\[ P(X_i \mid X_{i-1}) \]
\[ P(O_i \mid X_i) \]
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 $P(x_1, x_2, \ldots, x_n | o_1, o_2, \ldots, o_n)$?

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

1. Brute force:

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

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

   $\begin{align*}
   \alpha_1(k) &= \frac{P(X_1 = k)}{P(O_1 = o_1 | X_1 = k)} = P(o_1, X_1 = k) \\
   \alpha_{t+1}(k) &= \frac{\sum_{j=1}^{N} \alpha_t(j) P(X_{t+1} = k | X_t = j)}{P(O_{t+1} = o_{t+1} | X_t = k)} \\
   P(o_1, o_2, \ldots o_T) &= \frac{\sum_{k} \alpha_T(k)}{k}
   \end{align*}$
How do we compute
\[ P(X_t = k | o_1, o_2, \ldots, o_T) \]

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

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

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

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

\[ \sum_{k} \alpha_T(k) \beta_t(k) \]
How do we compute

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

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 $o_1...o_T$

$X_1 = \{a,...,z\}$
$X_2 = \{a,...,z\}$
$X_3 = \{a,...,z\}$
$X_4 = \{a,...,z\}$
$X_5 = \{a,...,z\}$

$O_1 = \text{obs}$
$O_2 = \text{obs}$
$O_3 = \text{unobs}$
$O_4 = \text{unobs}$
$O_5 = \text{unobs}$

EM

Initial model params $\Theta$

Loop to convergence.

$E \text{step: use current } \Theta \text{ to calc. expected values of } \{ z_i \}$

(Calc. Viterbi alg.)

M step: re-maximize likelihood by choosing $\Theta'$

$\Theta' = \arg \max_{\Theta} E \left[ \log P(o_1, o_2 ... o_T, x_1 ... x_T | \Theta') \right]$
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