Machine Learning
4771

Instructor: Tony Jebara
Topic 19

• Hidden Markov Models
• HMMs as State Machines & Applications
• HMMs Basic Operations
• HMMs via the Junction Tree Algorithm
Hidden Markov Models

• A great application of Junction Tree Algorithm and EM
• Recall mixture of Gaussians model on IID data

\[
p(q_t | \theta) = \begin{bmatrix} .5 & .2 & .3 \end{bmatrix}
\]

• Example: location data of a single parent as a mixture of Gaussians
  Parent has 3 internal states:
  \( q = \{ \text{home, daycare, work} \} \)
• Based on \( q \), sample from appropriate Gaussian mean and covariance to get \( y = (\text{latitude, longitude}) \)
Hidden Markov Models

- Parent drops child at daycare before & after work. Not IID!
- Have dependence on previous state
- Can’t go straight from home to work!
- Now, order of $y_0, \ldots, y_T$ matters (in IID order doesn’t matter)
Hidden Markov Models

• Consider mixture of multinomials (dice) \( y = \{1, 2, 3, 4, 5, 6\} \)

• Example: a crooked casino croupier using mixture of dice.
• You win if he rolls 1, 2, 3. You lose he rolls 4, 5, 6.
• Croupier has 3 internal states \( q = \{\text{helpful, fair, adversarial}\} \)
• Based on \( q \), sample different ‘dice’ multinomial

\[
p(q_t | \theta) = \left[ \begin{array}{c} .2 \\ .6 \\ .2 \end{array} \right]
\]
Hidden Markov Models

• But if the dealer has a memory or mood? Not IID!

5 6 4 6 1 6 6 1 6 6  4 3 2 1 5 3 4 1 6 1 4 1 4 3 4 1 6 3 4  1 1 1 3 1 1 4 1 2 1

• If you tip, dealer starts to like you and rolls the helpful die
• Dealer has a memory of his mood and last type of die \( q_{t-1} \)
• Will often use same die for \( q_t \) as was rolled before...
• Now, order of \( y_0, ..., y_T \) matters (if IID order doesn’t matter)
Hidden Markov Models

• Since next choice of the dice depends on previous one...

• Add left-right arrows. This is a hidden Markov model

• Markov: future $\parallel$ past $\mid$ present

\[ p(q_t \mid q_{t-1}, q_{t-2}, \ldots, q_1, q_0) = p(q_t \mid q_{t-1}) \]

• From graph, have the following general pdf:

\[ p(X_U) = p(q_0) \prod_{t=1}^{T} p(q_t \mid q_{t-1}) \prod_{t=0}^{T} p(y_t \mid q_t) \]

• So \( p(q_t) \) depends on previous state \( q_{t-1} \) ...

\[
\begin{align*}
 p\left( q_t \mid q_{t-1} = 1 \right) & = \frac{1}{3} \\
 p\left( q_t \mid q_{t-1} = 2 \right) & = \frac{2}{3} \\
 p\left( q_t \mid q_{t-1} = 3 \right) & = \frac{1}{3}
\end{align*}
\]
HMMs as State Machines

• HMMs have two variables: state \( q \) and emission \( y \)
• Typically, we don’t know \( q \) (hidden variable 1, 2, 3, ?)
• HMMs are like stochastic automata or finite state machines...
  next state depends on previous one...
  (helpful, fair, adversarial)
• Can’t observe state \( q \) directly, just a random related emission \( y \) outcome (dice roll) so...
  doubly-stochastic automaton
HMM Applications

• Speech Recognition
  phonemes from audio cepstral vectors

• Language Parsing
  parts of speech from words

• Genomics
  splice site from gene sequence

Ba-ra-kk-oo-oo-dd-ah

Noun  Verb  Noun
John  Ate  Pizza

-Intron- | -Exon- | -Promoter-
GATTACATTATACCCACCATAACG
HMMs: Parameters

• We focus on HMMs with: discrete state $q$ (of size $M$) 
discrete emission $y$ (of size $N$)

• Input will be arbitrary length string: $y_0, \ldots, y_T$

• The pdf or (complete) likelihood is:

$$p(q, y) = p(q_0) \prod_{t=1}^{T} p(q_t \mid q_{t-1}) \prod_{t=0}^{T} p(y_t \mid q_t)$$

• We don’t know hidden states, the incomplete likelihood is:

$$p(y) = \sum_{q_0} \cdots \sum_{q_T} p(q, y)$$

• Assume HMM is stationary, tables are repeated: $\theta = \{\pi, \eta, \alpha\}$

$$p(q_t \mid q_{t-1}) = \prod_{i=1}^{M} \prod_{j=1}^{M} \left[ \alpha_{ij} \right]^{q_{t-1}^i q_t^j} \quad \sum_{j=1}^{M} \alpha_{ij} = 1 \quad M \times M$$

$$p(y_t \mid q_t) = \prod_{i=1}^{M} \prod_{j=1}^{N} \left[ \eta_{ij} \right]^{q_t^i y_t^j} \quad \sum_{j=1}^{N} \eta_{ij} = 1 \quad M \times N$$

$$p(q_0) = \prod_{i=1}^{M} \left[ \pi_i \right]^{q_0^i} \quad \sum_{j=1}^{M} \pi_j = 1 \quad M$$
HMMs: Basic Operations

- Would like to do 3 basic things with our HMMs:
  1) **Evaluate**: given $y_0, \ldots, y_T$ & $\theta$ compute $p(y_1, \ldots, y_T)$
  2) **Decode**: given $y_0, \ldots, y_T$ & $\theta$ find $q_0, \ldots, q_T$ or $p(q_0), \ldots, p(q_T)$
  3) **Max Likelihood**: given $y_0, \ldots, y_T$ learn parameters $\theta$

- Typically use Baum-Welch ($\alpha$-$\beta$ algo)... JTA is more general:

HMMs easily get Junction Tree
HMMs: JTA Init & Verify

- **Init:**
  \[
  \psi(q_0, y_0) = p(q_0) p(y_0 | q_0) \quad \psi(q_t, q_{t+1}) = \alpha_{q_t, q_{t+1}} \quad \psi(q_t, y_t) = p(y_t | q_t)
  \]

- **Collect up** (this time it actually doesn’t change the zetas)
  \[
  \varsigma^*(q_t) = \sum_{y_t} \psi(q_t, y_t) = \sum_{y_t} p(y_t | q_t) = 1 \\
  \psi^*(q_{t-1}, q_t) = \frac{\varsigma^*(q_{t-1}, q_t)}{\varsigma(q_{t-1})} \psi(q_{t-1}, q_t) = \psi(q_{t-1}, q_t)
  \]

- **Collect left-right** via phi’s: change backbone to marginals
  \[
  \phi^*(q_0) = \sum_{q_0} \psi(q_0, y_0) = p(q_0) \quad \psi^*(q_0, q_1) = \frac{\phi^*(q_0)}{\phi(q_0)} \psi(q_0, q_1) = p(q_0, q_1) \\
  \phi^*(q_t) = \sum_{q_{t-1}} \psi(q_{t-1}, q_t) = p(q_t) \quad \psi^*(q_{t-1}, q_t) = \frac{p(q_{t-1})}{p(q_t | q_{t-1})} p(q_t | q_{t-1}) = p(q_{t-1}, q_t)
  \]

- **Distribute:**
  \[
  \varsigma^{**}(q_t) = \sum_{q_{t-1}} \psi^*(q_{t-1}, q_t) = \sum_{q_{t-1}} p(q_{t-1}, q_t) = p(q_t) \\
  \psi^{**}(q_t, y_t) = \frac{\varsigma^{**}(q_t)}{\varsigma(q_t)} \psi(q_t, y_t) = \frac{p(q_t)}{p(y_t | q_t)} p(y_t | q_t) = p(y_t, q_t)
  \]
  ...done!