Machine Learning

4771

Instructors:
Adrian Weller and Ilia Vovsha
Lecture 24

• 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 with EM
• So far, we have dealt with mixture models with IID:

\[ y_t = (\text{speed}, \text{size}) \]

• Recall mixture of Gaussians and EM...
• Variable \( q \) was a multinomial
• Roll a die to determine sub-population:
  \( q = \{\text{compact}, \text{midsize}, \text{luxury}\} \)
• Then sample appropriate Gaussian mean and covariance to get
  \( y = (\text{speed}, \text{size}) \)
• Can consider other mixtures too, multinomials, Poisson...

\[ p(q_t | \theta) = [0.5, 0.2, 0.3] \]
Hidden Markov Models

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

$$p(q_t | \theta) = \begin{bmatrix} .2 & .6 & .2 \end{bmatrix}$$

- Example: a crooked casino croupier using mixture of dice.
- You win if he rolls 1,2,3. You lose if he rolls 4,5,6.
- Croupier has three dice (one fair & two weighted):
Hidden Markov Models

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

- What if the dealer has a memory or mood? Not IID!

- Dealer might start to like you and roll 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 die depends on previous one...

\[ q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \]

\[ y_0 \rightarrow y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow y_4 \]

Order of \( y_0, \ldots, y_T \) matters
Temporal or sequence model!

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

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

• 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 \( q_{t-1} \) ...

\[ p(q_t \mid q_{t-1} = 1) \]

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

\[ p(q_t \mid q_{t-1} = 3) \]
HMMs as State Machines

• HMMs have two variable types: state \( q \) and emission \( y \)
• Typically, we don’t know \( q \) (hidden variable, e.g. 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 Rec** (Rabiner): phonemes from audio cepstral vectors
- **Language** (Jelinek): parts of speech from words
- **Biology** (Baldi): splice site from gene sequence
- **Gesture** (Starner): word from hand coordinates
- **Emotion** (Picard): emotion from EEG

\[ q_0 \rightarrow q_1 \rightarrow q_2 \]

\[ y_0 \rightarrow y_1 \rightarrow y_2 \]

- **Ba-ra-kk-oo-oо-dd-ah**
- **Noun Verb Noun**: John Ate Pizza
- **-Intron- | -Exon- | -Promoter-**
- **GATTACATTATATACCACCAG**
- **Pass The Salt**
- **Happy Neutral Sad**
Hidden Markov Models

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

- Haven’t yet specified the types of variables or cpts...

1) \( q \) can be discrete, example: finite state machine

\[ p(q_t | q_{t-1}) = \prod_{i=1}^{M} \prod_{j=1}^{M} [a_{ij}]^{q_{t-1}^i q_t^j} \]

2) \( y \) can be vectors, example: time series

\[ p(y_t | q_t) = N(y_t | \mu_{q_t}, \Sigma_{q_t}) \]

3) \( y \) can be discrete, example: strings

\[ p(y_t | q_t) = \prod_{i=1}^{M} \prod_{j=1}^{N} [\eta_{ij}]^{q_t^i y_t^j} \]

4) \( q \) and \( y \) can be vectors, example: Kalman filter

\[ p(q_t | q_{t-1}) = N(q_t | Aq_{t-1}, Q) \quad \text{and} \quad N(y_t | Cq_t, R) \]

Kalman Filters, Linear dynamical systems
Used in tracking, control (see ch. 14)
HMMs: Parameters

- We focus on HMMs with: discrete state $q$ (of size $M$) and discrete emission $y$ (of size $N$).
- Input will be arbitrary length string: $y_1, \ldots, y_T$.
- The pdf or (complete) likelihood is:
  \[
p(q, y) = p(q_0) \prod_{t=1}^{T} p(q_t | q_{t-1}) \prod_{t=0}^{T} p(y_t | q_t)
  \]
- 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:
  \[
  \theta = \left\{ \pi, \alpha, \eta \right\}
  \]

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

| $p(y_t | q_t)$ | $M \times N$ |
|----------------|-------------|
| $\prod_{i=1}^{M} \prod_{j=1}^{N} [\eta_{ij}]^{q_t^i y_t^j}$ | \[
  \sum_{j=1}^{N} \eta_{ij} = 1
  \] |

<table>
<thead>
<tr>
<th>$p(q_0)$</th>
<th>$M$</th>
</tr>
</thead>
</table>
| $\prod_{i=1}^{M} [\pi_i]^{q_0^i}$ | \[
  \sum_{j=1}^{M} \pi_j = 1
  \] |
HMMs: Basic Operations

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

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

HMMs easily get Junction Tree
HMMs: JTA Init & Verify

• Init: \( \psi(q_0, y_0) = p(q_0)p(y_0 | q_0) \)
  \( \psi(q_t, q_{t+1}) = p(q_{t+1} | q_t) = \alpha_{q_t, q_{t+1}} \)
  \( \psi(q_t, y_t) = p(y_t | q_t) \)

  \[ \psi(q_0, y_0) \quad \psi(q_0, q_1) \quad \psi(q_1, q_2) \quad \cdots \quad \psi(q_{T-1}, q_T) \]

  \[ \phi(q_0) \quad \phi(q_1) \quad \phi(q_2) \quad \cdots \quad \phi(q_T) \]

  \[ \varsigma(q_1) \quad \varsigma(q_2) \quad \cdots \quad \varsigma(q_T) \]

  \( \psi(q_1, y_1) \quad \psi(q_2, y_2) \quad \cdots \quad \psi(q_T, y_T) \)

• Collect up from leaves: don’t change zeta separators

  \( \varsigma^*(q_t) = \sum_{y_t} \psi(q_t, y_t) = \sum_{y_t} p(y_t | q_t) = 1 \)
  \( \psi^*(q_{t-1}, q_t) = \frac{\varsigma^*}{\varsigma} \psi(q_{t-1}, q_t) = \psi(q_{t-1}, q_t) \)

• Collect left-right via phi’s: change backbone to marginals

  \( \phi^*(q_0) = \sum_{y_0} \psi(q_0, y_0) = p(q_0) \)
  \( \phi^*(q_t) = \sum_{q_{t-1}} \psi^*(q_{t-1}, q_t) = p(q_t) \)

  \( \psi^*(q_0, q_1) = \frac{\phi^*}{\phi} \psi(q_0, q_1) = p(q_0, q_1) \)
  \( \psi^*(q_{t-1}, q_t) = \frac{p(q_{t-1})}{p(q_t | q_{t-1})} p(q_t | q_{t-1}) = p(q_{t-1}, q_t) \)

• Distribute:

  \( \varsigma^{**}(q_t) = \sum_{q_{t-1}} \psi^*(q_{t-1}, q_t) = \sum_{q_{t-1}} p(q_{t-1}, q_t) = p(q_t) \)

  \( \psi^{**}(q_t, y_t) = \frac{\varsigma^{**}}{\varsigma^*} \psi(q_t, y_t) = \frac{p(q_t)}{p(y_t | q_t)} p(y_t | q_t) = p(y_t, q_t) \)

...done!