Sequential Data

Greg Mori - CMPT 419/726

Bishop PRML Ch. 13
Russell and Norvig, AIMA
Outline

Hidden Markov Models

Inference for HMMs

Learning for HMMs
Outline

Hidden Markov Models

Inference for HMMs

Learning for HMMs
Temporal Models

• The world changes over time
  • Explicitly model this change using Bayesian networks
  • Undirected models also exist (will not cover)

• Basic idea: copy state and evidence variables for each time step

• e.g. Diabetes management

• $z_t$ is set of unobservable state variables at time $t$
  • $bloodSugar_t$, $stomachContents_t$, ...

• $x_t$ is set of observable evidence variables at time $t$
  • $measuredBloodSugar_t$, $foodEaten_t$, ...

• Assume discrete time step, fixed

• Notation: $x_{a:b} = x_a, x_{a+1}, \ldots, x_{b-1}, x_b$
Markov Chain

- Construct Bayesian network from these variables
  - parents? distributions? for state variables $z_t$:
• Construct Bayesian network from these variables
  • parents? distributions? for state variables $z_t$:
• Markov assumption: $z_t$ depends on bounded subset of $z_{1:t-1}$
  • First-order Markov process: $p(z_t|z_{1:t-1}) = p(z_t|z_{t-1})$
  • Second-order Markov process: $p(z_t|z_{1:t-1}) = p(z_t|z_{t-2}, z_{t-1})$
• Stationary process: $p(z_t|z_{t-1})$ fixed for all $t$
Hidden Markov Model (HMM)

- **Sensor Markov assumption**: \( p(x_t|z_{1:t}, x_{1:t-1}) = p(x_t|z_t) \)

- **Stationary process**: transition model \( p(z_t|z_{t-1}) \) and sensor model \( p(x_t|z_t) \) fixed for all \( t \) (separate \( p(z_1) \))

- HMM special type of Bayesian network, \( z_t \) is a single discrete random variable:

  ![Diagram of HMM]

  - Joint distribution:
    \[
p(z_{1:t}, x_{1:t}) = \]

Hidden Markov Model (HMM)

- **Sensor Markov assumption**: $p(x_t|z_{1:t}, x_{1:t-1}) = p(x_t|z_t)$
- **Stationary process**: transition model $p(z_t|z_{t-1})$ and sensor model $p(x_t|z_t)$ fixed for all $t$ (separate $p(z_1)$)
- **HMM special type of Bayesian network**, $z_t$ is a **single discrete random variable**:

  ![Diagram](image)

- **Joint distribution**:
  
  $p(z_{1:t}, x_{1:t}) = p(z_1) \prod_{i=2:t} p(z_i|z_{i-1}) \prod_{i=1:t} p(x_i|z_i)$
HMM Example

- First-order Markov assumption not true in real world
- Possible fixes:
  - Increase order of Markov process
  - Augment state, add $temp_t$, $pressure_t$
• $z_n$ takes one of 3 values
• Using one-of-$K$ coding scheme, $z_{nk} = 1$ if in state $k$ at time $n$
• Transition matrix $A$ where $p(z_{nk} = 1 | z_{n-1,j} = 1) = A_{jk}$
The lattice or trellis representation shows possible paths through the latent state variables $z_n$. 

**Lattice / Trellis Representation**
Outline

Hidden Markov Models

Inference for HMMs

Learning for HMMs
Inference Tasks

- **Filtering**: $p(z_t|x_{1:t})$
  - Estimate current unobservable state given all observations to date

- **Prediction**: $p(z_n|x_{1:t})$ for $n > t$
  - Similar to filtering, without evidence

- **Smoothing**: $p(z_n|x_{1:t})$ for $n < t$
  - Better estimate of past states

- **Most likely explanation**: $\arg\max_{z_{1:t}} p(z_{1:t}|x_{1:t})$
  - e.g. speech recognition, decoding noisy input sequence
Filtering

- **Aim:** devise a *recursive* state estimation algorithm:

\[ p(z_{t+1}|x_{1:t+1}) = f(x_{t+1}, p(z_t|x_{1:t})) \]

\[
p(z_{t+1}|x_{1:t+1}) = p(z_{t+1}|x_{1:t}, x_{t+1}) \]
\[
= \alpha p(x_{t+1}|x_{1:t}, z_{t+1}) p(z_{t+1}|x_{1:t}) \]
\[
= \alpha p(x_{t+1}|z_{t+1}) p(z_{t+1}|x_{1:t})
\]

- i.e. prediction + estimation. Prediction by summing out \( z_t \):

\[
p(z_{t+1}|x_{1:t+1}) = \alpha p(x_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}, z_t|x_{1:t}) \]
\[
= \alpha p(x_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}|z_t, x_{1:t}) p(z_t|x_{1:t}) \]
\[
= \alpha p(x_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}|z_t) p(z_t|x_{1:t})
\]
Filtering

- **Aim:** devise a **recursive** state estimation algorithm:

\[
p(z_{t+1}|x_{1:t+1}) = f(x_{t+1}, p(z_t|x_{1:t}))
\]

\[
p(z_{t+1}|x_{1:t+1}) = p(z_{t+1}|x_{1:t}, x_{t+1})
\]

\[
= \alpha p(x_{t+1}|x_{1:t}, z_{t+1}) p(z_{t+1}|x_{1:t})
\]

\[
= \alpha p(x_{t+1}|z_{t+1}) p(z_{t+1}|x_{1:t})
\]

- **I.e. prediction + estimation.** Prediction by summing out \(z_t:\)

\[
p(z_{t+1}|x_{1:t+1}) = \alpha p(x_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}, z_t|x_{1:t})
\]

\[
= \alpha p(x_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}|z_t, x_{1:t}) p(z_t|x_{1:t})
\]

\[
= \alpha p(x_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}|z_t) p(z_t|x_{1:t})
\]
Filtering

- **Aim**: devise a recursive state estimation algorithm:

\[ p(z_{t+1}|x_{1:t+1}) = f(x_{t+1}, p(z_t|x_{1:t})) \]

\[ p(z_{t+1}|x_{1:t+1}) = p(z_{t+1}|x_{1:t}, x_{t+1}) \]
\[ = \alpha p(x_{t+1}|x_{1:t}, z_{t+1})p(z_{t+1}|x_{1:t}) \]
\[ = \alpha p(x_{t+1}|z_{t+1})p(z_{t+1}|x_{1:t}) \]

- **I.e. prediction + estimation**. Prediction by summing out \( z_t \):

\[ p(z_{t+1}|x_{1:t+1}) = \alpha p(x_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}, z_t|x_{1:t}) \]
\[ = \alpha p(x_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}|z_t, x_{1:t})p(z_t|x_{1:t}) \]
\[ = \alpha p(x_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}|z_t)p(z_t|x_{1:t}) \]
Filtering Example

\[
p(rain_1 = true) = 0.5
\]

\[
p(z_{t+1}|x_{1:t+1}) = \alpha p(x_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}|z_t)p(z_t|x_{1:t})
\]

<table>
<thead>
<tr>
<th>(R_{t-1})</th>
<th>(P(R_t))</th>
<th>(R_t)</th>
<th>(P(U_t))</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>0.7</td>
<td>t</td>
<td>0.9</td>
</tr>
<tr>
<td>f</td>
<td>0.3</td>
<td>f</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Filtering - Lattice

- Using notation in PRML, forward message is $\alpha(z_n)$
- Compute $\alpha(z_{n,i})$ using sum over $k$ of $\alpha(z_{n-1,k})$ multiplied by $A_{ki}$, then multiplying in evidence $p(x_t|z_{ni})$
- Each step, computing $\alpha(z_n)$ takes $O(K^2)$ time, with $K$ values for $z_n$
**Smoothing**

- Divide evidence $x_{1:t}$ into $x_{1:n}$, $x_{n+1:t}$

\[
p(z_n|x_{1:t}) = p(z_n|x_{1:n}, x_{n+1:t})
\]

\[
= \alpha p(z_n|x_{1:n}) p(x_{n+1:t}|z_n, x_{1:n})
\]

\[
= \alpha p(z_n|x_{1:n}) p(x_{n+1:t}|z_n)
\]

\[
= \alpha \alpha(z_n) \beta(z_n)
\]

- Backwards message $\beta(z_n)$ another recursion:
Smoothing

- Divide evidence $x_{1:t}$ into $x_{1:n}$, $x_{n+1:t}$

$$p(z_n|x_{1:t}) = p(z_n|x_{1:n}, x_{n+1:t})$$
$$= \alpha p(z_n|x_{1:n}) p(x_{n+1:t}|z_n, x_{1:n})$$
$$= \alpha p(z_n|x_{1:n}) p(x_{n+1:t}|z_n)$$
$$= \alpha \alpha(z_n) \beta(z_n)$$

- Backwards message $\beta(z_n)$ another recursion:
• Divide evidence $x_{1:t}$ into $x_{1:n}$, $x_{n+1:t}$, $p(z_n|x_{1:t}) = \alpha(z_n)\beta(z_n)$

• Backwards message another recursion:

$$p(x_{n+1:t} | z_n) = \sum_{z_{n+1}} p(x_{n+1:t}, z_{n+1} | z_n)$$

$$= \sum_{z_{n+1}} p(x_{n+1:t} | z_{n+1}, z_n)p(z_{n+1} | z_n)$$

$$= \sum_{z_{n+1}} p(x_{n+1:t} | z_{n+1})p(z_{n+1} | z_n)$$

$$= \sum_{z_{n+1}} p(x_{n+1} | z_{n+1})p(x_{n+2:t} | z_{n+1})p(z_{n+1} | z_n)$$
Smoothing Example

Rain₁ → Rain₂

Umbrella₁ → Umbrella₂

forward
smoothed
backward

0.500
0.500

0.618
0.182

0.627
0.373

0.883
0.117

0.883
0.117

0.690
0.410

1.000
1.000
• Using notation in PRML, backward message is $\beta(z_n)$
• Compute $\beta(z_{n,i})$ using sum over $k$ of $\beta(z_{n+1,k})$ multiplied by $A_{ik}$ and evidence $p(x_{n+1}|z_{n+1,k})$
• Each step, computing $\beta(z_n)$ takes $O(K^2)$ time, with $K$ values for $z_n$
Forward-Backward Algorithm

- Filter from time 1 to $N$, and cache forward messages $\alpha(z_n)$
- Smooth from time $N$ to 1, and cache backward messages $\beta(z_n)$
- Can now compute $p(z_n|x_1, x_2, \ldots, x_N)$ for all $n$
- Total complexity $O(NK^2)$
- a.k.a Baum-Welch algorithm
Outline

Hidden Markov Models

Inference for HMMs

Learning for HMMs
The parameters of an HMM are:

- **Transition matrix** $A$ where $p(z_{nk} = 1 | z_{n-1,j} = 1) = A_{jk}$
- **Sensor model** $\phi_k$ parameters to each $p(x_n | z_{nk} = 1, \phi_k)$ (e.g. $\phi_k$ could be mean and variance of Gaussian)
- **Prior for initial state** $z_1$, model as multinomial $p(z_{1k} = 1) = \pi_k$, parameters $\pi$

Call these parameters $\theta = (A, \pi, \phi)$

Learning problem: given one sequence $x$, find best $\theta$

- Extension to multiple sequences straight-forward (assume independent, log of product is sum)
Maximum Likelihood for HMMs

- We can use maximum likelihood to choose the best parameters:

\[ \theta_{ML} = \arg \max p(x|\theta) \]

- Unfortunately this is hard to do: we can get \( p(x|\theta) \) by summing out from the joint distribution:

\[
p(x|\theta) = \sum_{z_1} \sum_{z_2} \cdots \sum_{z_N} p(x, z_1, z_2, \ldots, z_N|\theta)
\]

\[
\equiv \sum_z p(x, z|\theta)
\]

- But this sum has \( K^N \) terms in it
- And, as in the mixture distribution case, no simple closed-form solution
- Instead, use expectation-maximization (EM)
EM for HMMs

- Start with initial guess for parameters $\theta^{old} = (A, \pi, \phi)$
- **E-step**: Calculate posterior on latent variables $p(z|x, \theta^{old})$
- **M-step**: Maximize $Q(\theta, \theta^{old}) = \sum_z p(z|x, \theta^{old}) \ln p(x, z|\theta)$ wrt $\theta$
- Let’s look at the M-step, and see how the HMM structure helps us
HMM M-step

- **M-step**: Maximize $Q(\theta, \theta^{old}) = \sum_z p(z|x, \theta^{old}) \ln p(x, z|\theta)$ wrt $\theta$:

- The **complete data log-likelihood** factors nicely:

\[
\ln p(x, z|\theta) = \ln \left\{ p(z_1|\pi) \prod_{i=2:N} p(z_i|z_{i-1}, A) \prod_{i=1:N} p(x_i|z_i, \phi) \right\} \\
= \ln p(z_1|\pi) + \sum_{i=2:N} \ln p(z_i|z_{i-1}, A) + \sum_{i=1:N} \ln p(x_i|z_i, \phi)
\]

- To maximize $Q$ we now have 3 separate problems, one for each parameter
  - Let’s consider each in turn
Prior $\pi$

- Maximize $Q$ wrt prior on initial state $\pi$:

$$Q(\pi, \theta^{old}) = \sum_z p(z|x, \theta^{old}) \ln p(z_1 | \pi)$$

$$= \sum_z p(z|x, \theta^{old}) \ln \prod_{k=1}^{K} \pi_{zk}$$

$$= \sum_z p(z|x, \theta^{old}) \sum_{z_1k} \ln \pi_k$$

$$= \ln \pi_k \sum_{z_1k} p(z|x, \theta^{old})z_{1k}$$

$$= \sum_{z_1k} p(z_{1k} = 1|x, \theta^{old}) \ln \pi_k$$

- I.e. smoothed value for $z_1$ being in state $k$
\[ Q(\pi, \theta^{old}) = \sum_{k=1}^{K} p(z_{1k} = 1|x, \theta^{old}) \ln \pi_k \]

- Can solve for best \( \pi \)
- Use Lagrange multiplier to enforce constraint \( \sum_k \pi_k = 1 \)

\[ \pi_k = \frac{p(z_{1k} = 1|x, \theta^{old})}{\sum_{j=1}^{K} p(z_{1j} = 1|x, \theta^{old})} \]

- Intuitively sensible result: new \( \pi_k \) is smoothed probability of being in state \( k \) at time 1 using old parameters
- E-step needs to calculate smoothed \( p(z_{1k} = 1|x, \theta^{old}) \) – this is fast \( O(NK^2) \)
Transition Matrix $A$

- **Maximize** $Q$ wrt transition matrix $A$:

$$Q(A, \theta^{old}) = \sum_z p(z|x, \theta^{old}) \sum_{i=2:N} \ln p(z_i|z_{i-1}, A)$$

$$= \sum_z p(z|x, \theta^{old}) \sum_{i=2:N} \ln \prod_{k=1:K} \prod_{j=1:K} A^{z_{i-1},jz_i,k}$$

$$= \sum_z p(z|x, \theta^{old}) \sum_{i=2:N} \sum_{k=1:K} \sum_{j=1:K} z_{i-1},jz_i,k \ln A_{jk}$$

$$= \sum_{k=1:K} \sum_{j=1:K} \ln A_{jk} \sum_{i=2:N} \sum_z p(z|x, \theta^{old})z_{i-1},jz_i,k$$

$$= \sum_{k=1:K} \sum_{j=1:K} \ln A_{jk} \sum_{i=2:N} p(z_{i-1} = j, z_i = k|x, \theta^{old})$$

- **E-step** needs to calculate $p(z_{i-1} = j, z_i = k|x, \theta^{old})$ – can be done quickly using forward and backward messages
\[
Q(A, \theta^{old}) = \sum_{k=1:K} \sum_{j=1:K} \ln A_{jk} \sum_{i=2:N} p(z_{n-1} = j, z_n = k | x, \theta^{old})
\]

- Can solve for best A
- Again use Lagrange multipliers to enforce constraint \(\sum_k A_{jk} = 1\)

\[
A_{jk} = \frac{\sum_{n=2:N} p(z_{n-1} = j, z_n = k | x, \theta^{old})}{\sum_{l=1:K} \sum_{n=2:N} p(z_{n-1} = j, z_n = l | x, \theta^{old})}
\]

- Again sensible result: \(A_{jk}\) set to expected number of times we transition from state \(j\) to \(k\) using the smoothed results from old parameters
Sensor Model

- Similar derivation for sensor model parameters $\phi$
- Again end up with weighted parameter estimated based on expected values of states given smoothed estimates
HMM EM Summary

- Start with initial guess for parameters $\theta^{old} = (A, \pi, \phi)$
- Run forward-backward algorithm to get all messages $\alpha(z_n)$, $\beta(z_n)$ (E-step)
  - $O(NK^2)$ time complexity
  - Can use these to compute any smoothed posterior $p(z_{nk} = 1|x, \theta^{old})$
  - Also can compute any $p(z_{n-1,j} = 1, z_{n,k} = 1|x, \theta^{old})$
- Using these, update values for parameters (M-step)
  - $\pi_k$ is smoothed probability of being in in state $k$ at time 1
  - $A_{jk}$ is smoothed probability of transitioning from state $j$ to $k$ averaged over all time steps
  - $\phi$ is weighted sensor parameters using smoothed probabilities (e.g. similar to mixture of Gaussians)
- Repeat until convergence
HMM EM Summary

- Start with initial guess for parameters $\theta^{old} = (A, \pi, \phi)$
- Run forward-backward algorithm to get all messages $\alpha(z_n)$, $\beta(z_n)$ (E-step)
  - $O(NK^2)$ time complexity
  - Can use these to compute any smoothed posterior $p(z_{nk} = 1|x, \theta^{old})$
  - Also can compute any $p(z_{n-1,j} = 1, z_{n,k} = 1|x, \theta^{old})$
- Using these, update values for parameters (M-step)
  - $\pi_k$ is smoothed probability of being in in state $k$ at time 1
  - $A_{jk}$ is smoothed probability of transitioning from state $j$ to $k$ averaged over all time steps
  - $\phi$ is weighted sensor parameters using smoothed probabilities (e.g. similar to mixture of Gaussians)
- Repeat until convergence
HMM EM Summary

- Start with initial guess for parameters $\theta^{old} = (A, \pi, \phi)$
- Run forward-backward algorithm to get all messages $\alpha(z_n), \beta(z_n)$ (E-step)
  - $O(NK^2)$ time complexity
  - Can use these to compute any smoothed posterior $p(z_{nk} = 1|x, \theta^{old})$
  - Also can compute any $p(z_{n-1,j} = 1, z_n, k = 1|x, \theta^{old})$
- Using these, update values for parameters (M-step)
  - $\pi_k$ is smoothed probability of being in in state $k$ at time 1
  - $A_{jk}$ is smoothed probability of transitioning from state $j$ to $k$ averaged over all time steps
  - $\phi$ is weighted sensor parameters using smoothed probabilities (e.g. similar to mixture of Gaussians)
- Repeat until convergence
HMM EM Summary

- Start with initial guess for parameters $\theta^{old} = (A, \pi, \phi)$
- Run forward-backward algorithm to get all messages $\alpha(z_n), \beta(z_n)$ (E-step)
  - $O(NK^2)$ time complexity
  - Can use these to compute any smoothed posterior $p(z_{nk} = 1|x, \theta^{old})$
  - Also can compute any $p(z_{n-1,j} = 1, z_{n,k} = 1|x, \theta^{old})$
- Using these, update values for parameters (M-step)
  - $\pi_k$ is smoothed probability of being in state $k$ at time 1
  - $A_{jk}$ is smoothed probability of transitioning from state $j$ to $k$ averaged over all time steps
  - $\phi$ is weighted sensor parameters using smoothed probabilities (e.g. similar to mixture of Gaussians)
- Repeat until convergence
Conclusion

- Readings: Ch. 13.2, 13.2.1, 13.2.2
- HMM - Probabilistic model of temporal data
  - Discrete hidden (unobserved, latent) state variable at each time
    - Continuous (next)
  - Observation (can be discrete / continuous) at each time
  - Conditional independence assumptions (Markov)
  - Assumptions on distributions (stationary)
- Inference
  - Filtering
  - Smoothing
  - Most likely sequence (next)
- Maximum likelihood learning
  - EM – efficient computation $O(NK^2)$ time using forward-backward smoothing