Agenda: Hidden Markov Models

- Hidden Markov models
- Forward algorithm for state estimation (filtering)
- Forward-backward algorithm for state estimation (smoothing)
- Viterbi algorithm for most likely explanation
- EM algorithm (Baum Welch) for parameter estimation
Hidden Markov model

- \( \{H_t\} \) & \( \{Y_t\} \) discrete time stochastic processes
- \( \{H_t\} \) Markov chain and not directly observable ("hidden")
- \( \{Y_t\} \) directly observable

\[
P(Y_t \mid H_{1:T}) = P(Y_t \mid H_{1:t}) = P(Y_t \mid H_t)
\]

- Terminology: Transition probabilities \( P(H_{t+1} \mid H_t) \)
  - Emission probabilities \( P(Y_t \mid H_t) \)
  
  Homogeneous if above probabilities time independent (assumed henceforth)
Examples

▶ Speech recognition
▶ Finance forecasting
▶ DNA motif discovery
▶ ...

Speech recognition

$X = x_1 x_2 \ldots x_T$

$W^* = \arg \max_W p(X|W) p(W)$

Acoustic model

Language model
Copy number variations

How many duplications do we have? Do we have deletions?
Inference problems

1. What is the prob/likelihood of an observed sequence $Y_{1:T}$?

2. What is the prob/likelihood of the latent variable given $Y$?

3. What is the most likely value of a latent variable? (‘decoding problem’)

4. Given one or several observed sequences, how would we estimate model parameters? i.e. transition & emission probabilities (and distribution of initial latent variable)
Probability of an observed sequence

\[ P(y_{1:T}) = \sum_{h_{1:T}} P(y_{1:T}, h_{1:T}) = \sum_{h_{1:T}} P(h_1) \prod_{t=2}^{T} P(h_t \mid h_{t-1}) \prod_{t=1}^{T} P(y_t \mid h_t) \]  

\[ (*) = \sum_{h_T} P(y_T \mid h_T) \sum_{h_{1:T-1}} P(h_T \mid h_{T-1}) P(h_1) \prod_{t=2}^{T-1} P(h_t \mid h_{t-1}) \prod_{t=1}^{T-1} P(y_t \mid h_t) \]

Suggests dynamic programming solution
Forward algorithm

- Forward probabilities: $\alpha_t(h_t) = P(y_{1:t}, h_t)$  \( y_{1:T} \) is given throughout

- Recursion: $\alpha_1(h_1) = P(h_1)P(y_1, h_1)$ &

\[
\alpha_{t+1}(h_{t+1}) = \sum_{h_t} P(y_{t+1} \mid h_{t+1}, h_t, y_{1:t}) P(h_{t+1} \mid h_t, y_{1:t}) \alpha_t(h_t)
\]

\[
= P(y_{t+1} \mid h_{t+1}) \sum_{h_t} P(h_{t+1} \mid h_t) \alpha_t(h_t)
\]

One matrix-vector product per time step!

- Likelihood of an observed sequence is: $P(y_{1:T}) = \sum_h \alpha_T(h)$
Probability of latent variables

Interested in conditional distribution of latent variables:

(a) Filtering : $P(h_t \mid y_{1:t})$ [forward algorithm]

(b) Smoothing (hindsight) : $P(h_t \mid y_{1:T})$ [forward-backward algorithm]

(c) Most likely explanation : $(\arg\max_{h_{1:T}} P(h_{1:T} \mid y_{1:T}))$ [Viterbi algorithm]
Filtering: $\mathbb{P}(h_t \mid y_{1:t})$

Forward probabilities $\alpha_t(h_t) = \mathbb{P}(h_t, y_{1:t})$

$$\mathbb{P}(h_t \mid y_{1:t}) = \frac{\alpha_t(h_t)}{\sum_h \alpha_t(h)}$$
Smoothing: $\mathbb{P}(h_t \mid y_{1:T})$

Conditional independence of $y_{t+1:T} \& y_{1:t} \mid h_t$ + Bayes’ rule give

$$\mathbb{P}(h_t \mid y_{1:T}) = \mathbb{P}(h_t \mid y_{1:t}, y_{t+1:T}) \propto \mathbb{P}(y_{t+1:T} \mid h_t) \mathbb{P}(h_t \mid y_{1:t})$$

Backward probabilities:

$$\beta_T(h_T) = 1$$

$$\beta_t(h_t) = \mathbb{P}(y_{t+1:T} \mid h_t)$$

Recursion

$$\beta_t(h_t) = \sum_{h_{t+1}} \mathbb{P}(y_{t+1:T}, h_{t+1} \mid h_t) = \sum_{h_{t+1}} \mathbb{P}(h_{t+1} \mid h_t) \mathbb{P}(y_{t+1} \mid h_{t+1}) \beta_{t+1}(h_{t+1})$$

One matrix-vector product per time step!
Forward-backward algorithm

- Compute forward probabilities: \( \alpha_t(h_t) = \mathbb{P}(y_{1:t}, h_t) \)

- Backward: \( \beta_t(h_t) = \mathbb{P}(y_{t+1:T} \mid h_t) \)

\[
\mathbb{P}(h_t \mid y_{1:T}) = \frac{\mathbb{P}(y_{1:t}, h_t)\mathbb{P}(y_{t+1:T} \mid h_t)}{\mathbb{P}(y_{1:T})} = \frac{\alpha_t(h_t)\beta_t(h_t)}{\sum_h \alpha_T(h)} = \gamma_t(h_t)
\]

Complexity of algorithm is \( O(Tn^2) \) where \( n \) \# hidden states
Most likely explanation: \[
\arg \max_{h_{1:T}} \mathbb{P}(h_{1:T} \mid y_{1:T})
\]

Value function: \[
V_1(j) = \mathbb{P}(y_1, h_1 = j) = \mathbb{P}(y_1 \mid h_1 = j)\mathbb{P}(h_1 = j)
\]

\[
V_t(j) = \max_{h_{1:t-1}} \mathbb{P}(h_{1:t-1}, y_{1:t}, h_t = j)
\]

Recursion

\[
V_{t+1}(j) = \max_{h_{1:t}} \mathbb{P}(h_{1:t-1}, y_{1:t}, h_t, y_{t+1}, h_{t+1} = j)
\]

\[
= \max_{h_{1:t-1}} \max_i \mathbb{P}(h_{1:t-1}, y_{1:t}, h_t = i)\mathbb{P}(h_{t+1} = j \mid h_t = i)\mathbb{P}(y_{t+1} \mid h_{t+1} = j)
\]

\[
= \max_i V_t(i)\mathbb{P}(h_{t+1} = j \mid h_t = i)\mathbb{P}(y_{t+1} \mid h_{t+1} = j) \quad (\ast)
\]

Keeping track of optimal state: \[
\psi_t(j) = \arg \max \text{ in } (\ast)
\]

\[
\psi_T() = \arg \max_j V_T(j)
\]
Viterbi algorithm

- Compute $V_t$ for $t = 1, \ldots, T$

- $\psi_t$

- Backtrack for most likely sequence:

  $$H_T = \psi_T(\ )$$

  For $t = T - 1, T - 2, \ldots, 1$

  $$H_t = \psi_t(H_{t+1})$$
Parameter estimation (learning)

Given observed sequence(s) $Y_{1:T}$, how can we estimate the model parameters?

Parameters (collectively denoted by $\theta$)

- $\pi$: distribution of $X_1$
- $T(x, x')$: transition probabilities
- $E(x, y) = \mathbb{P}(Y_t = y \mid X_t = x)$: emission probabilities

Solution via maximum likelihood & Baum-Welch (EM) algorithm
Baum-Welch algorithm

- Complete likelihood

\[ P(X_{1:T}, Y_{1:T}) = P(X_1) \prod_{t=2}^{T} P(X_t \mid X_{t-1}) \prod_{t=1}^{T} P(Y_t \mid X_t) \]

- Parameter estimation from complete likelihood is easy (why?)

- EM iteration: current parameter value \( \theta_k \)

E-step: Compute \( Q(\theta \mid \mid \theta_k) = \mathbb{E}_X [\log P_{\theta}(X, Y) \mid Y, \theta_k] \)

M-step: \( \theta_{k+1} = \arg \max_{\theta} Q(\theta \mid \mid \theta_k) \)
E-step

\[ \gamma_t(x) = \mathbb{P}(X_t = x \mid y_{1:T}, \theta_k) \quad \text{[forward-backward algorithm]} \]

\[ \xi_t(x, x') = \mathbb{P}(X_t = x, X_{t+1} = x' \mid y_{1:T}, \theta_k) \propto \alpha_t(x) T^{(k)}(x, x') \beta_{t+1}(x') E^{(k)}(x', y_{t+1}) \]

Above relation not hard to show...

Probabilities use current guesses of model parameters

\[ P^{(k)}(x, x') = \mathbb{P}(X_{t+1} = x' \mid X_t = x, \theta_k) \quad E^{(k)}(x, y) = \mathbb{P}(Y_t = y \mid X_t = x, \theta_k) \]

Log-likelihood:

\[ \log P(X, Y) = \log \pi(X_1) + \sum_{t=1}^{T-1} \log T(X_{t+1}, X_t) + \sum_{t=1}^{T} \log E(X_t, y_t) \]

E-step

\[ Q(\theta \mid \theta_k) = \sum_x \gamma_1(x) \log \pi(x) + \sum_{t=1}^{T-1} \sum_{x, x'} \xi_t(x, x') \log T(x, x') + \sum_{t=1}^{T} \sum_x \gamma_t(x) \log E(x, y) \]
Q-step

\[ Q(\theta \mid \theta_k) = \sum_x \gamma_1(x) \log \pi(x) + \sum_{x,x'} \sum_{t=1}^{T-1} \xi_t(x, x') \log T(x, x') + \sum_x \sum_{t=1}^{T} \gamma_t(x) \log E(x, y_t) \]

Q-step: update parameters \( \pi(\cdot), T(\cdot, \cdot), E(\cdot, \cdot) \)

\[ \pi^+(x) = \gamma_1(x) \]

\[ T^+(x, x') = \frac{\sum_{t=1}^{T-1} \xi_t(x, x')}{\sum_z \sum_{t=1}^{T-1} \xi_t(x, z)} = \frac{\sum_{t=1}^{T-1} \xi_t(x, x')}{\sum_{t=1}^{T-1} \gamma_t(x)} \]

\[ E^+(x, v) = \frac{\sum_{t=1}^{T} 1\{y_t = v\} \gamma_x(t)}{\sum_{t=1}^{T} \gamma_x(t)} \]
That’s all folks!
Enjoy the summer