# Biostatistics 615/815 Lecture 10: Hidden Markov Models

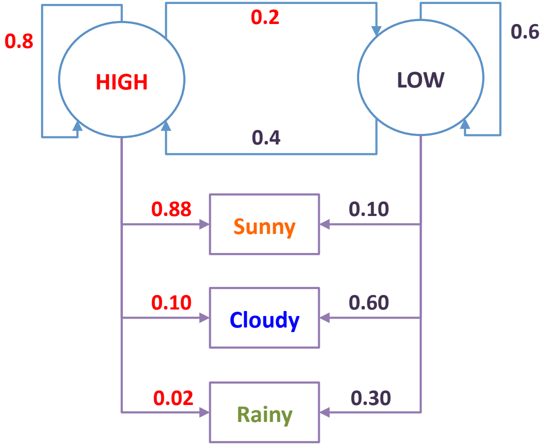
Hyun Min Kang

October 4th, 2012

# Hidden Markov Models (HMMs)

- A Markov model where actual state is unobserved
  - Transition between states are probabilistically modeled just like the Markov process
- Typically there are observable outputs associated with hidden states
  - The probability distribution of observable outputs given an hidden states can be obtained.

# An example of HMM




- Direct Observation : (SUNNY, CLOUDY, RAINY)
- Hidden States : (HIGH, LOW)

# Mathematical representation of the HMM example

States $S = \{S_1, S_2\} = $ (HIGH, LOW)

Outcomes $O = \{O_1, O_2, O_3\} = $ (SUNNY, CLOUDY, RAINY)

Initial States $\pi_i = \Pr(q_1 = S_i)$, $\pi = \{0.7, 0.3\}$

Transition $A_{ij} = \Pr(q_{t+1} = S_j | q_t = S_i)$

$$A = \begin{pmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{pmatrix}$$

Emission $B_{ij} = b_{q_t}(o_t) = b_{S_i}(O_j) = \Pr(o_t = O_j | q_t = S_i)$

$$B = \begin{pmatrix} 0.88 & 0.10 & 0.02 \\ 0.10 & 0.60 & 0.30 \end{pmatrix}$$

# Unconditional marginal probabilities

## What is the chance of rain in the day 4?

$$\mathbf{f}(\mathbf{q}_4) = \begin{pmatrix} \Pr(q_4 = S_1) \\ \Pr(q_4 = S_2) \end{pmatrix} = (A^T)^3 \pi = \begin{pmatrix} 0.669 \\ 0.331 \end{pmatrix}$$

$$\mathbf{g}(o_4) = \begin{pmatrix} \Pr(o_4 = O_1) \\ \Pr(o_4 = O_2) \\ \Pr(o_4 = O_3) \end{pmatrix} = B^T \mathbf{f}(\mathbf{q}_4) = \begin{pmatrix} 0.621 \\ 0.266 \\ 0.233 \end{pmatrix}$$

The chance of rain in day 4 is 23.3%

# Marginal likelihood of data in HMM

- Let $\lambda = (A, B, \pi)$
- For a sequence of observation $\mathbf{o} = \{o_1, \cdots, o_t\}$,

$$\Pr(\mathbf{o}|\lambda) = \sum_{\mathbf{q}} \Pr(\mathbf{o}|\mathbf{q},\lambda) \Pr(\mathbf{q}|\lambda)$$

$$\Pr(\mathbf{o}|\mathbf{q},\lambda) = \prod_{i=1}^{t} \Pr(o_i|q_i,\lambda) = \prod_{i=1}^{t} b_{q_i}(o_i)$$

$$\Pr(\mathbf{q}|\lambda) = \pi_{q_1} \prod_{i=2}^{t} a_{q_{i-1}q_i}$$

$$\Pr(\mathbf{o}|\lambda) = \sum_{\mathbf{q}} \pi_{q_1} b_{q_1}(o_1) \prod_{i=2}^{t} a_{q_{i-1}q_i} b_{q_i}(o_i)$$

# Naive computation of the likelihood

$$\Pr(\mathbf{o}|\lambda) = \sum_{\mathbf{q}} \pi_{q_1} b_{q_1}(o_1) \prod_{i=2}^{t} a_{q_{i-1} q_i} b_{q_i}(o_i)$$

- Number of possible $q = 2^t$ are exponentially growing with the number of observations
- Computational would be infeasible for large number of observations
- Algorithmic solution required for efficient computation.

# More Markov Chain Question

- If the observation was (SUNNY,SUNNY,CLOUDY,RAINY,RAINY) from day 1 through day 5, what is the distribution of hidden states for each day?
- Need to know $\Pr(q_t|\mathbf{o}, \lambda)$

# Forward and backward probabilities

$$
\begin{aligned}
\mathbf{q}_t^- &= (q_1, \cdots, q_{t-1}), \qquad \mathbf{q}_t^+ = (q_{t+1}, \cdots, q_T) \\
\mathbf{o}_t^- &= (o_1, \cdots, o_{t-1}), \qquad \mathbf{o}_t^+ = (o_{t+1}, \cdots, o_T) \\
\Pr(q_t = i | \mathbf{o}, \lambda) &= \frac{\Pr(q_t = i, \mathbf{o}|\lambda)}{\Pr(\mathbf{o}|\lambda)} = \frac{\Pr(q_t = i, \mathbf{o}|\lambda)}{\sum_{j=1}^n \Pr(q_t = j, \mathbf{o}|\lambda)} \\
\Pr(q_t, \mathbf{o}|\lambda) &= \Pr(q_t, \mathbf{o}_t^-, o_t, \mathbf{o}_t^+ | \lambda) \\
&= \Pr(\mathbf{o}_t^+ | q_t, \lambda) \Pr(\mathbf{o}_t^- | q_t, \lambda) \Pr(o_t | q_t, \lambda) \Pr(q_t | \lambda) \\
&= \Pr(\mathbf{o}_t^+ | q_t, \lambda) \Pr(\mathbf{o}_t^-, o_t, q_t | \lambda) \\
&= \beta_t(q_t) \alpha_t(q_t)
\end{aligned}
$$

If $\alpha_t(q_t)$ and $\beta_t(q_t)$ is known, $\Pr(q_t|\mathbf{o}, \lambda)$ can be computed in a linear time.

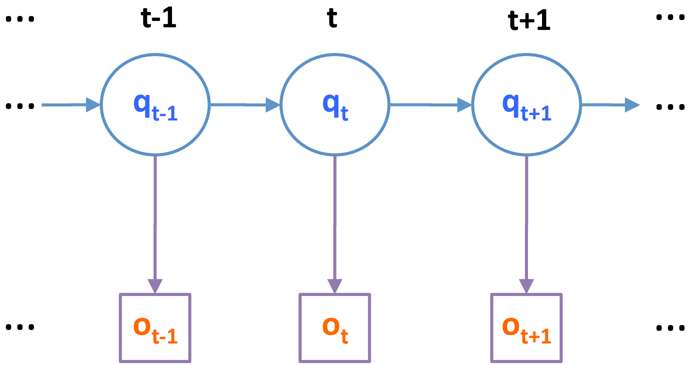
# DP algorithm for calculating forward probability

- Key idea is to use $(q_t, o_t) \perp \mathbf{o}_t^- | \mathbf{q}_{t-1}$.
- Each of $q_{t-1}$, $q_t$, and $q_{t+1}$ is a Markov blanket.

$$
\begin{aligned}
\alpha_t(i) &= \Pr(o_1, \cdots, o_t, q_t = i|\lambda) \\
&= \sum_{j=1}^{n} \Pr(\mathbf{o}_t^-, o_t, q_{t-1} = j, q_t = i|\lambda) \\
&= \sum_{j=1}^{n} \Pr(\mathbf{o}_t^-, q_{t-1} = j|\lambda) \Pr(q_t = i|q_{t-1} = j, \lambda) \Pr(o_t|q_t = i, \lambda) \\
&= \sum_{j=1}^{n} \alpha_{t-1}(j) a_{ji} b_i(o_t) \\
\alpha_1(i) &= \pi_i b_i(o_1)
\end{aligned}
$$

# Conditional dependency in forward-backward algorithms

- Forward : $(q_t, o_t) \perp \mathbf{o}_t^- | \mathbf{q}_{t-1}$.
- Backward : $o_{t+1} \perp \mathbf{o}_{t+1}^+ | \mathbf{q}_{t+1}$.



# DP algorithm for calculating backward probability

- Key idea is to use $o_{t+1} \perp \mathbf{o}_{t+1}^{+} | \mathbf{q}_{t+1}$.

$$
\begin{aligned}
\beta_t(i) &= \Pr(o_{t+1}, \cdots, o_T | q_t = i, \lambda) \\
&= \sum_{j=1}^{n} \Pr(o_{t+1}, \mathbf{o}_{t+1}^{+}, q_{t+1} = j | q_t = i, \lambda) \\
&= \sum_{j=1}^{n} \Pr(o_{t+1} | q_{t+1}, \lambda) \Pr(\mathbf{o}_{t+1}^{+} | q_{t+1} = j, \lambda) \Pr(q_{t+1} = j | q_t = i, \lambda) \\
&= \sum_{j=1}^{n} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}) \\
\beta_T(i) &= 1
\end{aligned}
$$

# Putting forward and backward probabilities together

- Conditional probability of states given data

$$\Pr(q_t = i|\mathbf{o}, \lambda) = \frac{\Pr(\mathbf{o}, q_t = S_i|\lambda)}{\sum_{j=1}^{n} \Pr(\mathbf{o}, q_t = S_j|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{n} \alpha_t(j)\beta_t(j)}$$

- Time complexity is $\Theta(n^2 T)$.

# Finding the most likely trajectory of hidden states

- Given a series of observations, we want to compute

$$\arg\max_{\mathbf{q}} \Pr(\mathbf{q}|\mathbf{o}, \lambda)$$

- Define $\delta_t(i)$ as

$$\delta_t(i) = \max_{\mathbf{q}} \Pr(\mathbf{q}, \mathbf{o}|\lambda)$$

- Use dynamic programming algorithm to find the 'most likely' path

# The Viterbi algorithm

**Initialization** $\delta_1(i) = \pi b_i(o_1)$ for $1 \leq i \leq n$.

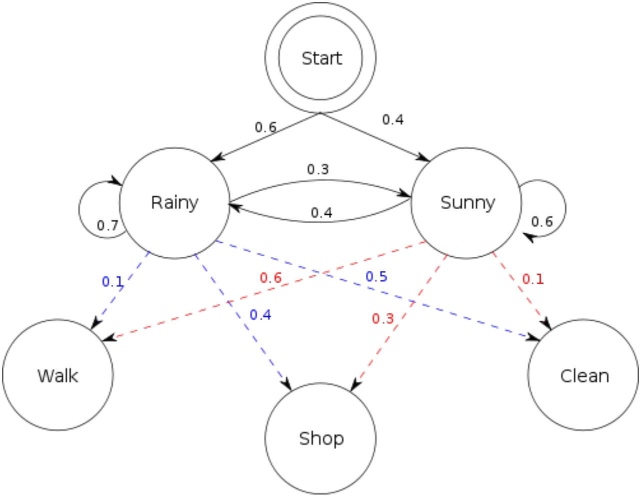
**Maintenance** $\delta_t(i) = \max_j \delta_{t-1}(j) a_{ji} b_i(o_t)$

$\phi_t(i) = \arg\max_j \delta_{t-1}(j) a_{ji}$

**Termination** Max likelihood is $\max_i \delta_T(i)$

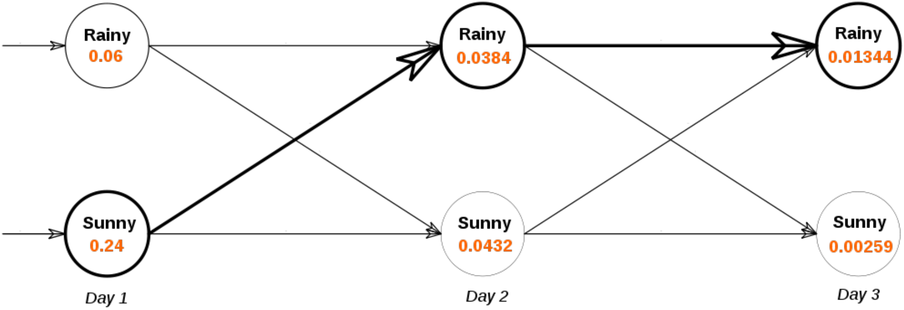
Optimal path can be backtracked using $\phi_t(i)$

# An HMM example



# An example Viterbi path

- When observations were (walk, shop, clean)
- Similar to Manhattan tourist problem.



# Statistical analysis with HMM

## HMM for a deterministic problem

- Given
  - Given parameters $\lambda = \{\pi, A, B\}$
  - and data $\mathbf{o} = (o_1, \cdots, o_T)$
- Forward-backward algorithm
  - Compute $\Pr(q_t|\mathbf{o}, \lambda)$
- Viterbi algorithm
  - Compute $\arg\max_{\mathbf{q}} \Pr(\mathbf{q}|\mathbf{o}, \lambda)$

## HMM for a stochastic process / algorithm

- Generate random samples of $\mathbf{o}$ given $\lambda$

# Deterministic Inference using HMM

- If we know the exact set of parameters, the inference is deterministic given data
  - No stochastic process involved in the inference procedure
  - Inference is deterministic just as estimation of sample mean is deterministic
- The computational complexity of the inference procedure is exponential using naive algorithms
- Using dynamic programming, the complexity can be reduced to $O(n^2 T)$.

# Using Stochastic Process for HMM Inference

Using random process for the inference

- Randomly sampling **o** from $\Pr(\mathbf{o}|\lambda)$.
- Estimating $\arg\max_\lambda \Pr(\mathbf{o}|\lambda)$.
  - No deterministic algorithm available
  - Simplex, E-M algorithm, or Simulated Annealing is possible apply
- Estimating the distribution $\Pr(\lambda|\mathbf{o})$.
  - Gibbs Sampling

# Recap : The E-M Algorithm

## Expectation step (E-step)

- Given the current estimates of parameters $\theta^{(t)}$, calculate the conditional distribution of latent variable $\mathbf{z}$.
- Then the expected log-likelihood of data given the conditional distribution of $\mathbf{z}$ can be obtained

$$Q(\theta|\theta^{(t)}) = \mathbf{E}_{\mathbf{z}|\mathbf{x},\theta^{(t)}} \left[\log p(\mathbf{x}, \mathbf{z}|\theta)\right]$$

## Maximization step (M-step)

- Find the parameter that maximize the expected log-likelihood

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^t)$$

# Baum-Welch for estimating $\arg\max_\lambda \Pr(\mathbf{o}|\lambda)$

Assumptions

- Transition matrix is identical between states
  - $a_{ij} = \Pr(\mathbf{q}_{t+1} = i|\mathbf{q}_t = j) = \Pr(\mathbf{q}_t = i|\mathbf{q}_{t-1} = j)$
- Emission matrix is identical between states
  - $b_i(j) = \Pr(\mathbf{o}_t = j|\mathbf{q}_t = i) = \Pr(\mathbf{o}_{t=1} = j|\mathbf{q}_{t-1} = i)$
- This is NOT the only possible assumption.
  - For example, $a_{ij}$ can be parameterized as a function of $t$.
  - Multiple sets of **o** independently drawn from the same distribution can be provided.
  - Other assumptions will result in different formulation of E-M algorithm

# E-step of the Baum-Welch Algorithm

1. Run the forward-backward algorithm given $\lambda^{(\tau)}$

$$\begin{aligned} \alpha_t(i) &= \Pr(o_1, \cdots, o_t, q_t = i | \lambda^{(\tau)}) \\ \beta_t(i) &= \Pr(o_{t+1}, \cdots, o_T | q_t = i, \lambda^{(\tau)}) \\ \gamma_t(i) &= \Pr(q_t = i | \mathbf{o}, \lambda^{(\tau)}) = \frac{\alpha_t(i)\beta_t(i)}{\sum_k \alpha_t(k)\beta_t(k)} \end{aligned}$$

2. Compute $\xi_t(i,j)$ using $\alpha_t(i)$ and $\beta_t(i)$

$$\begin{aligned} \xi_t(i,j) &= \Pr(q_t = i, q_{t+1} = j | \mathbf{o}, \lambda^{(\tau)}) \\ &= \frac{\alpha_t(i) a_{ji} b_j(o_{t+1}) \beta_{t+1}(j)}{\Pr(\mathbf{o}|\lambda^{(\tau)})} \\ &= \frac{\alpha_t(i) a_{ji} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{(k,l)} \alpha_t(k) a_{lk} b_l(o_{t+1}) \beta_{t+1}(l)} \end{aligned}$$

# M-step of the Baum-Welch Algorithm

Let $\lambda^{(\tau+1)} = (\pi^{(\tau+1)}, A^{(\tau+1)}, B^{(\tau+1)})$

$$
\begin{aligned}
\pi^{(\tau+1)}(i) &= \frac{\sum_{t=1}^{T} \Pr(q_t = i|\mathbf{o}, \lambda^{(\tau)})}{T} = \frac{\sum_{t=1}^{T} \gamma_t(i)}{T} \\
a_{ij}^{(\tau+1)} &= \frac{\sum_{t=1}^{T-1} \Pr(q_t = j, q_{t+1} = i|\mathbf{o}, \lambda^{(\tau)})}{\sum_{t=1}^{T-1} \Pr(q_t = j|\mathbf{o}, \lambda^{(\tau)})} = \frac{\sum_{t=1}^{T-1} \xi_t(j,i)}{\sum_{t=1}^{T-1} \gamma_t(j)} \\
b_i(k)^{(\tau+1)} &= \frac{\sum_{t=1}^{T} \Pr(q_t = i, o_t = k|\mathbf{o}, \lambda^{(\tau)})}{\sum_{t=1}^{T} \Pr(q_t = i|\mathbf{o}, \lambda^{(\tau)})} = \frac{\sum_{t=1}^{T} \gamma_t(i) I(o_t = k)}{\sum_{t=1}^{T} \gamma_t(i)}
\end{aligned}
$$

A detailed derivation can be found at

- Welch, "Hidden Markov Models and The Baum Welch Algorithm", IEEE Information Theory Society News Letter, Dec 2003