ECE 8873
Data Compression and Modeling

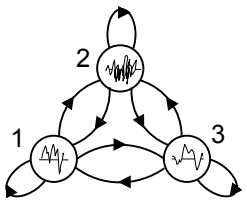
Lecture 12: Hidden Markov Model

School of Electrical and Computer Engineering
Georgia Institute of Technology
Spring, 2004

Non-stationarity & Non-memoryless

- Many real world signals have characteristics that change with time – it is what makes them interesting and "informative."
- Recall the structural component of information – it often changes at irregular times.
- Finite state Markov chain is one of the simplest process that allows us to model varying characteristics with "memory."
- A simple model that can be used to approximate non-stationary, non-memoryless sources is the hidden Markov model.

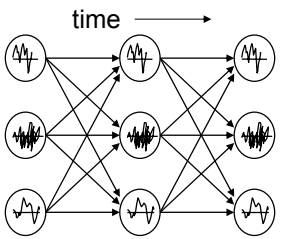
Hidden Markov Models



$$P(X \mid \Lambda) = \sum_{\mathbf{q}} P(X, \mathbf{q} \mid \Lambda)$$

$$P(X, \mathbf{q} \mid \Lambda) = \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1} q_t} b_{q_t}(x_t)$$

$$X = (x_1, x_2, \cdots, x_T) \qquad \mathbf{q} = (q_0, q_1, q_2, \cdots, q_T)$$



- Each state represents a process of measurable observations.
- Inter-process transition is governed by a finite state Markov chain.
- Processes are stochastic and individual observations do not immediately identify the state.

Hidden Markov Models - Specifications

$X = (x_1, x_2, \cdots, x_T)$ is the sequence of observations
$\mathbf{q} = (q_0, q_1, \cdots, q_T)$ is the sequence of states the system is in

- Number of states of the Markov chain, N
- State transition probability matrix, $A = \left[a_{ij} \right]_{N \times N}$

  $a_{ij} = \Pr[q_t = j \mid q_{t-1} = i] \qquad \sum_{j=1}^{N} a_{ij} = 1$ for all i

- In-state observation probability distribution functions

  $B = \{b_i(x)\}_{i=1}^{N} \qquad b_i(x) \Rightarrow b_i(x, \lambda_i)$ i.e., parameterized by λ_i
- Initial state probability distribution,

  $\boldsymbol{\pi}^t = [\pi_1, \pi_2, \cdots, \pi_N]$ where $\pi_i = \Pr[q_0 = i] \qquad \sum_{i=1}^{N} \pi_i = 1$

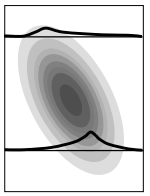
The triple $\Lambda = (\boldsymbol{\pi}, A, B)$ defines a hidden Markov model.

In-State (Local) Observation Distributions

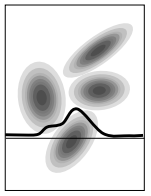
- Discrete distributions
  $x \in \{s_k\}_{k=1}^{M}$ and $b_i(x = s_k) = \Pr\{x = s_k, q = i\} = b_{ik}$
- Log-concave probability density functions
  x is continuous - valued and $\log b_i(x) = \log f_i(x)$ is a concave function
- Elliptically symmetric probability density functions
  $b_i(x) = \int f(x, g) d\mu(g)$
- General mixture probability density functions

$$b_i(x) = \sum_{k=1}^{M} c_{ik} f_{ik}(x)$$

where $\sum_{k=1}^{M} c_{ik} = 1$



Elliptically Symmetric Distribution



Mixture of Elliptically Symmetric Distribution

marginal

Three Basic Problems of HMM

- Given the observation sequence $X = (x_1, x_2, \cdots, x_T)$ and a model $\Lambda = (\boldsymbol{\pi}, A, B)$, how do we efficiently compute $P(X \mid \Lambda)$?
- Given the observation sequence $X = (x_1, x_2, \cdots, x_T)$ and the model $\Lambda = (\boldsymbol{\pi}, A, B)$, how do we find a corresponding state sequence $\mathbf{q} = (q_0, q_1, q_2, \cdots, q_T)$ that is optimal in some sense?
- Given an observation sequence X, or a number of sequences $\{X^{(i)}\}_i$, how to estimate parameters in the model set $\Lambda = (\boldsymbol{\pi}, A, B)$?

Evaluation of HMM Probability

$$P(X \mid \Lambda) = \sum_{\mathbf{q}} P(X, \mathbf{q} \mid \Lambda) \qquad P(X, \mathbf{q} \mid \Lambda) = \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1} q_t} b_{q_t}(x_t)$$

$$P(X \mid \Lambda) = \sum_{\mathbf{q}} P(X, \mathbf{q} \mid \Lambda) = \sum_{\mathbf{q}} \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1} q_t} b_{q_t}(x_t)$$

Direct evaluation will involve $2T \bullet N^T$ calculations.

The Forward Procedure

Define $\alpha_t(i) = \Pr(x_1, x_2, \cdots, x_t, q_t = i \mid \Lambda)$

$\alpha_t(i)$ Is the probability of the partial observation sequence $x_1, x_2, \cdots, x_t$, up to time t, and the system is at state i at time t.

Forward Procedure

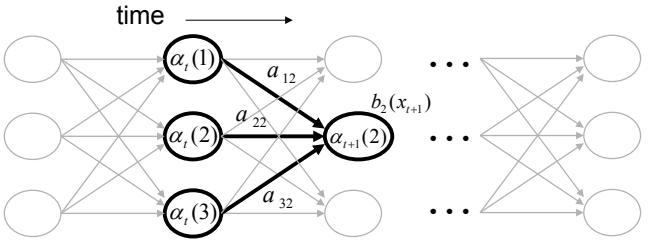
Initialization: $\alpha_0(i) = \pi_i, \quad i = 1, 2, \cdots, N$

Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(x_{t+1}), \quad 0 \le t \le T-1, 1 \le j \le N$$

Termination: $P(X \mid \Lambda) = \sum_{i=1}^{N} \alpha_T(i)$

~N^2T calculations



Backward Procedure

Define $\beta_t(i) = \Pr\{(x_{t+1}, x_{t+2}, \cdots, x_T \mid q_t = i, \Lambda)$

$\beta_t(i)$ Is the probability of the partial observation sequence $x_{t+1}, x_{t+2}, \cdots, x_T$, given that the system is at state i at time t.

Initialization: $\beta_T(i) = 1, \quad i = 1, 2, \cdots, N$

Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(x_{t+1})\beta_{t+1}(j), \quad t = T-1, T-2, \cdots, 1, \quad 1 \le i \le N$$

Optimal State Sequence

- Several possibilities
  - The state sequence that maximizes the joint state-observation probability

$$\mathbf{q}_{opt} = \arg\max_{\mathbf{q}} P(X, \mathbf{q} \mid \Lambda) = \arg\max_{\mathbf{q}} \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(x_t)$$

  - The state sequence that consists of individual states maximizing the a posteriori probability given the observation

$$\gamma_t(i) = P(q_t = i \mid X, \Lambda)$$

i.e. probability of being in state i at time t, given X

$$\gamma_t(i) = P(q_t = i \mid X, \Lambda) = P(X, q_t = i \mid \Lambda)[P(X \mid \Lambda)]^{-1}$$

$$= P(X, q_t = i \mid \Lambda)\left[\sum_{i=1}^{N} P(X, q_t = i \mid \Lambda)\right]^{-1}$$

$$= \alpha_t(i)\beta_t(i)\left[\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)\right]^{-1}$$

The Viterbi Algorithm

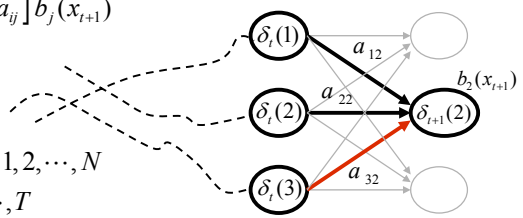
Find $\mathbf{q}^* = \arg\max_{\mathbf{q}} P(X, \mathbf{q} \mid \Lambda) = \arg\max_{\mathbf{q}} \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(x_t)$

$$\delta_t(i) \equiv \max_{q_1, q_2, \cdots, q_{t-1}} P\{q_1, q_2, \cdots, q_{t-1}, [q_t = i], x_1, x_2, \cdots, x_t \mid \Lambda\}$$

$$\delta_{t+1}(j) \equiv \max_{q_1, q_2, \cdots, q_t} P\{q_1, q_2, \cdots, q_t, [q_{t+1} = j], x_1, x_2, \cdots, x_{t+1} \mid \Lambda\}$$

$$= \max_i \left\{ \max_{q_1, q_2, \cdots, q_{t-1}} P\{q_1, q_2, \cdots, q_{t-1}, [q_t = i], x_1, x_2, \cdots, x_t \mid \Lambda\} a_{ij} \right\} b_j(x_{t+1})$$

$$= \max_i \left[\delta_t(i) a_{ij}\right] b_j(x_{t+1})$$



Do for all $j = 1, 2, \cdots, N$ and $t = 1, 2, \cdots, T$

The Viterbi Algorithm

- Initialization

$$\delta_0(i) = \pi_i, \quad i = 1, 2, \cdots, N$$

- Recursion

$$\delta_t(j) = \max_i \left[\delta_{t-1}(i) a_{ij}\right] b_j(x_t) \qquad j = 1, 2, \cdots, N$$

$$\psi_t(j) = \arg\max_i \left[\delta_{t-1}(i) a_{ij}\right] \qquad t = 1, 2, \cdots, T$$

- Termination

$$P^* = \max_i \left[\delta_T(i)\right]$$

$$q_T^* = \arg\max_i \left[\delta_T(i)\right]$$

- Backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \cdots, 1$$

Parameter Estimation

- Maximum likelihood (ML)
- Maximum mutual information (MMI)
- Minimum discrimination information (MDI)
- Minimum classification error (MCE)

Maximum Likelihood:

Given X, find $\overline{\Lambda} = \arg\max_{\Lambda} P(X \mid \Lambda)$

Or, given $\{X^{(i)}\}_{i=1}^{N}$ find

$$\overline{\Lambda} = \arg\max_{\Lambda} P(\{X^{(i)}\}_{i=1}^{N} \mid \Lambda)$$

X

Baum-Welch Re-estimation

Define $Q(\Lambda, \Lambda') = \sum_{\mathbf{q}} P(X, \mathbf{q} \mid \Lambda) \log P(X, \mathbf{q} \mid \Lambda')$ Q = Auxiliary function

Theorem: If $Q(\Lambda, \Lambda') \geq Q(\Lambda, \Lambda)$ then $P(X \mid \Lambda') \geq P(X \mid \Lambda)$. The inequality is strict unless $P(X, \mathbf{q} \mid \Lambda') \geq P(X, \mathbf{q} \mid \Lambda)$ almost everywhere.

Reestimation:

1. Given Λ, define the auxiliary function as a function of Λ';
2. Maximize the auxiliary function over Λ' and obtain $\overline{\Lambda}$

$$\overline{\Lambda} = T(\Lambda) \in \Psi = \{\hat{\Lambda} \mid Q(\Lambda, \hat{\Lambda}) = \max_{\Lambda'} Q(\Lambda, \Lambda')\}$$

3. Replace Λ with $\overline{\Lambda}$ and repeat the above until a stationary point is reached.

A general hill-climbing algorithm; similar to EM (expectation-maximization) algorithm

Reestimation Transformation

$\overline{\Lambda} = T(\Lambda) \in \Psi = \{\hat{\Lambda} \mid Q(\Lambda, \hat{\Lambda}) = \max_{\Lambda'} Q(\Lambda, \Lambda')\}$

For Gaussian mixture density HMM: $b_i(x) = \sum_{k=1}^{M} c_{ik} f(x; \mu_{ik}, \Sigma_{ik})$

Initial state probability: $\overline{\pi}_i = P(X, q_0 = i \mid \Lambda)[P(X \mid \Lambda)]^{-1}$

State transition probability:

$$\overline{a}_{ij} = \sum_{t=1}^{T} P(X, q_{t-1} = i, q_t = j \mid \Lambda) \left[\sum_{t=1}^{T} P(X, q_{t-1} = i \mid \Lambda)\right]^{-1}$$

Mixture weights:

$$\overline{c}_{ik} = \sum_{t=1}^{T} P(X, q_t = i, u_t = k \mid \Lambda) \left[\sum_{t=1}^{T} P(X, q_t = i \mid \Lambda)\right]^{-1}$$

Gaussian parameters:

$$\overline{\mu}_{ik} = \sum_{t=1}^{T} x_t P(X, q_t = i, u_t = k \mid \Lambda) \left[\sum_{t=1}^{T} P(X, q_t = i, u_t = k \mid \Lambda)\right]^{-1}$$

$$\overline{\Sigma}_{ik} = \sum_{t=1}^{T} (x_t - \mu_{ik})(x_t - \mu_{ik})^t P(X, q_t = i, u_t = k \mid \Lambda) \left[\sum_{t=1}^{T} P(X, q_t = i, u_t = k \mid \Lambda)\right]^{-1}$$

Forward & Backward Probability

$$\alpha_t(i) = \Pr\{(x_1, x_2, \cdots, x_t, q_t = i \mid \Lambda) \qquad \beta_t(i) = \Pr\{(x_{t+1}, x_{t+2}, \cdots, x_T \mid q_t = i, \Lambda)$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(x_{t+1}), \quad 0 \leq t \leq T-1, 1 \leq j \leq N$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(x_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \cdots, 1, \quad 1 \leq i \leq N$$

$$\gamma_t(i,k) = P(x_1, x_1, \cdots, x_1, q_t = i, u_t = k \mid \Lambda)$$

$$= \sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} c_{ik} f_{ik}(x_t), \quad t = 1, 2, \cdots, T, \quad 1 \leq i \leq N, \ 1 \leq k \leq M$$

$P(X, q_t = i \mid \Lambda) = \alpha_t(i)\beta_t(i)$ $\qquad P(X \mid \Lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) = \sum_{i=1}^{N} \alpha_T(i)$

$$P(X, q_{t-1} = i, q_t = j \mid \Lambda) = \alpha_{t-1}(i) a_{ij} \left[\sum_{k=1}^{M} c_{jk} f_{jk}(x_t)\right] \beta_t(j)$$

$$P(X, q_t = i, u_t = k \mid \Lambda) = \sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} c_{jk} f_{jk}(x_t) \beta_t(i)$$

Reestimation formulas can be expressed all in terms of the forward & backward probabilities.

Hidden Markov Model

- A reasonable modeling tool
- Useful in extracting the structural component of the information (signal)
- Can be used in coding schemes (e.g., finite state coding schemes)
- Certainly useful in speech recognition & language processing