Probabilistic Graphical Models

10-708

Hidden Markov models and extensions

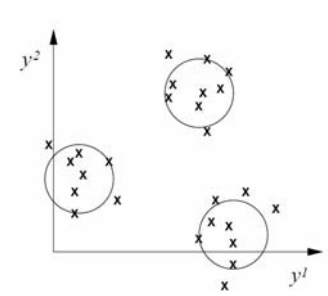
Eric Xing

Lecture 14, Oct 31, 2005

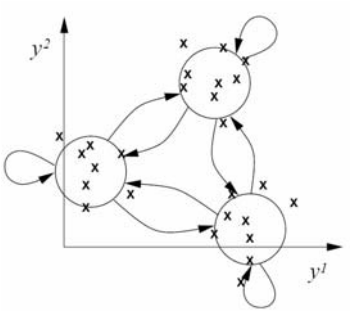
Reading: MJ-Chap. 11, 18

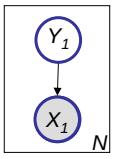
Hidden Markov Model:
from static to dynamic mixture models

Static mixture



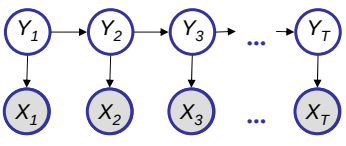
Dynamic mixture





The underlying source:
Speech signal, dice,

The sequence:
Phonemes, sequence of rolls,

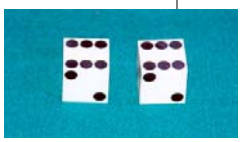


Example: The Dishonest Casino

A casino has two dice:

- Fair die
  P(1) = P(2) = P(3) = P(5) = P(6) = 1/6
- Loaded die
  P(1) = P(2) = P(3) = P(5) = 1/10
  P(6) = 1/2

Casino player switches back-&-forth between fair and loaded die once every 20 turns



Game:

1. You bet $1
2. You roll (always with a fair die)
3. Casino player rolls (maybe with fair die, maybe with loaded die)
4. Highest number wins $2

Puzzles regarding the dishonest casino

GIVEN: A sequence of rolls by the casino player

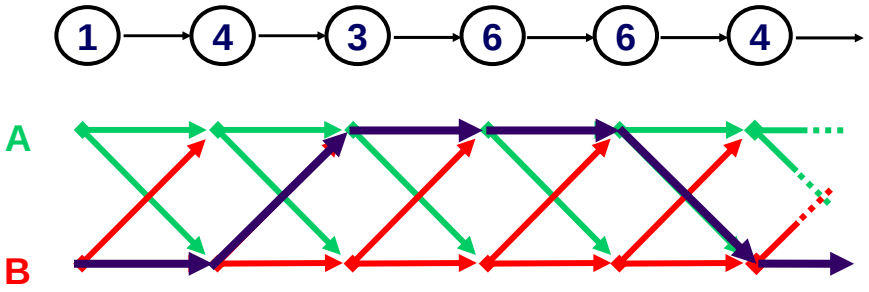
12455264621461461361366616646661636616366163616515615115146123562344

QUESTION

- How likely is this sequence, given our model of how the casino works?
  - This is the **EVALUATION** problem in HMMs
- What portion of the sequence was generated with the fair die, and what portion with the loaded die?
  - This is the **DECODING** question in HMMs
- How "loaded" is the loaded die? How "fair" is the fair die? How often does the casino player change from fair to loaded, and back?
  - This is the **LEARNING** question in HMMs

A stochastic generative model

- Observed sequence:



- Hidden sequence (a parse or segmentation):

Definition (of HMM)

- Observation space
  - **Alphabetic set:** $\mathbb{B} = \{b_1, b_2, \cdots, b_K\}$
  - **Euclidean space:** $\mathbb{R}^d$
- Index set of hidden states

  $\mathbb{I} = \{1,2,\cdots,M\}$
- Transition probabilities between any two states

  $p(y_t^j = 1 \mid y_{t-1}^i = 1) = a_{i,j}$,

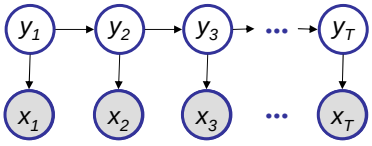
  or $p(y_t \mid y_{t-1}^i = 1) \sim \text{Multinomial}(a_{i,1}, a_{i,1}, \ldots, a_{i,M}), \forall i \in \mathbb{I}.$
- Start probabilities

  $p(y_1) \sim \text{Multinomial}(\pi_1, \pi_2, \ldots, \pi_M)$.
- Emission probabilities associated with each state

  $p(x_t \mid y_t^i = 1) \sim \text{Multinomial}(b_{i,1}, b_{i,1}, \ldots, b_{i,K}), \forall i \in \mathbb{I}.$

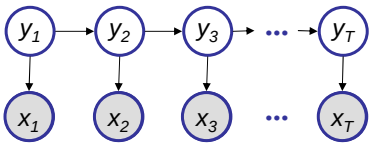
  or in general:

  $p(x_t \mid y_t^i = 1) \sim \text{f}(\cdot \mid \theta_i), \forall i \in \mathbb{I}.$



Probability of a parse

- Given a sequence $\mathbf{x} = x_1 \ldots\ldots x_T$ and a parse $\mathbf{y} = y_1, \ldots\ldots, y_T$,
- To find how likely is the parse: (given our HMM and the sequence)



$p(\mathbf{x}, \mathbf{y}) = p(x_1 \ldots\ldots x_T, y_1, \ldots\ldots, y_T)$ (Joint probability)

$= p(y_1)\, p(x_1 \mid y_1)\, p(y_2 \mid y_1)\, p(x_2 \mid y_2) \ldots p(y_T \mid y_{T-1})\, p(x_T \mid y_T)$

$= p(\pi_1)\, P(y_2 \mid y_1) \ldots p(y_T \mid y_{T-1}) \times p(x_1 \mid y_1)\, p(x_2 \mid y_2) \ldots p(x_T \mid y_T)$

$= p(y_1, \ldots\ldots, y_T)\, p(x_1 \ldots\ldots x_T \mid y_1, \ldots\ldots, y_T)$

Let $\pi_{y_1} \overset{\text{def}}{=} \prod_{i=1}^{M} [\pi_i]^{y_1^i}$, $a_{y_t, y_{t-1}} \overset{\text{def}}{=} \prod_{i,j=1}^{M} [a_{ij}]^{y_t^i y_{t-1}^j}$, and $\eta_{y_t, x_t} \overset{\text{def}}{=} \prod_{i=1}^{M} \prod_{k=1}^{K} [\eta_{ik}]^{y_t^i x_t^k}$,

$= \pi_{y_1} a_{y_1, y_2} \cdots a_{y_{T-1}, y_T}\, \eta_{y_1, x_1} \cdots \eta_{y_T, x_T}$

- Marginal probability: $p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) = \sum_{y_1} \sum_{y_2} \cdots \sum_{y_N} \pi_{y_1} \prod_{t=2}^{T} a_{y_{t-1}, y_t} \prod_{t=1}^{T} p(x_t \mid y_t)$
- Posterior probability: $p(\mathbf{y} \mid \mathbf{x}) = p(\mathbf{x}, \mathbf{y}) / p(\mathbf{x})$

Three main questions on HMMs

1. Evaluation

GIVEN	an HMM ***M***,	and a sequence $\mathbf{x}$,
FIND	Prob ($\mathbf{x}$ | ***M***)	
ALGO.	Forward	

2. Decoding

GIVEN	an HMM ***M***,	and a sequence $\mathbf{x}$,
FIND	the sequence $\mathbf{y}$ of states that maximizes, e.g., P($\mathbf{y}$ | $\mathbf{x}$, ***M***), or the most probable subsequence of states	
ALGO.	Viterbi, Forward-backward	

3. Learning

GIVEN	an HMM ***M***, with unspecified transition/emission probs., and a sequence $\mathbf{x}$,
FIND	parameters $\theta = (\pi_i, a_{ij}, \eta_{ik})$ that maximize P($\mathbf{x}$ | θ)
ALGO.	Baum-Welch (EM)

The Forward Algorithm

- We want to calculate $P(\mathbf{x})$, the likelihood of $\mathbf{x}$, given the HMM
  - Sum over all possible ways of generating $\mathbf{x}$:

$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}) = \sum_{y_1} \sum_{y_2} \cdots \sum_{y_N} \pi_{y_1} \prod_{t=2}^{T} a_{y_{t-1}, y_t} \prod_{t=1}^{T} p(x_t \mid y_t)$$

  - To avoid summing over an exponential number of paths $\mathbf{y}$, define

$$\alpha(y_t^k = 1) = \alpha_t^k \overset{\text{def}}{=} P(x_1, \ldots, x_t, y_t^k = 1)$$ (the forward probability)

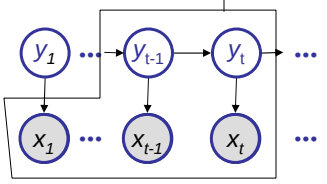
  - The recursion:

$$\alpha_t^k = p(x_t \mid y_t^k = 1) \sum_i \alpha_{t-1}^i a_{i,k}$$

$$P(\mathbf{x}) = \sum_k \alpha_T^k$$

The Forward Algorithm – derivation

- Compute the forward probability:

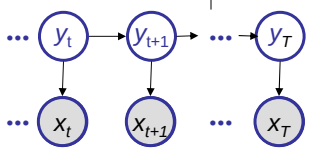


$$\begin{aligned}
\alpha_t^k &= P(x_1, \ldots, x_{t-1}, x_t, y_t^k = 1) \\
&= \sum_{y_{t-1}} P(x_1, \ldots, x_{t-1}, x_t, y_{t-1}, y_t^k = 1) \\
&= \sum_{y_{t-1}} P(x_1, \ldots, x_{t-1}, y_{t-1}) P(y_t^k = 1 \mid y_{t-1}, x_1, \ldots, x_{t-1}) P(x_t \mid y_t^k = 1, x_1, \ldots, x_{t-1}, y_{t-1}) \\
&= \sum_{y_{t-1}} P(x_1, \ldots, x_{t-1}, y_{t-1}) P(y_t^k = 1 \mid y_{t-1}) P(x_t \mid y_t^k = 1) \\
&= P(x_t \mid y_t^k = 1) \sum_i P(x_1, \ldots, x_{t-1}, y_{t-1}^i = 1) P(y_t^k = 1 \mid y_{t-1}^i = 1) \\
&= P(x_t \mid y_t^k = 1) \sum_i P(x_1, \ldots, x_{t-1}, y_{t-1}^i = 1) P(y_t^k = 1 \mid y_{t-1}^i = 1) \\
&= P(x_t \mid y_t^k = 1) \sum_i \alpha_{t-1}^i a_{i,k}
\end{aligned}$$

The Backward Algorithm

- We want to compute $P(y_t^k = 1 \mid \mathbf{x})$,

  the posterior probability distribution on the t^{th} position, given $\mathbf{x}$



  - We start by computing

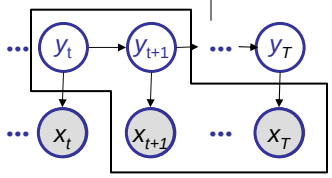
$$P(y_t^k = 1, \mathbf{x}) = P(x_1,\ldots,x_t, y_t^k = 1, x_{t+1},\ldots,x_T)$$
$$= P(x_1,\ldots,x_t, y_t^k = 1) P(x_{t+1},\ldots,x_T \mid x_1,\ldots,x_t, y_t^k = 1)$$
$$= P(x_1 \ldots x_t, y_t^k = 1) P(x_{t+1} \ldots x_T \mid y_t^k = 1)$$

Forward, α_t^k **Backward,** $\beta_t^k = P(x_{t+1},\ldots,x_T \mid y_t^k = 1)$

  - The recursion: $\beta_t^k = \sum_i a_{k,i} p(x_{t+1} \mid y_{t+1}^i = 1) \beta_{t+1}^i$

The Backward Algorithm – derivation

- Define the backward probability:



$$\beta_t^k = P(x_{t+1},\ldots,x_T \mid y_t^k = 1)$$
$$= \sum_{y_{t+1}} P(x_{t+1},\ldots,x_T, y_{t+1} \mid y_t^k = 1)$$
$$= \sum_i P(y_{t+1}^i = 1 \mid y_t^k = 1) p(x_{t+1} \mid y_{t+1}^i = 1) P(x_{t+2},\ldots,x_T \mid y_{t+1}^i = 1)$$
$$= \sum_i a_{k,i} p(x_{t+1} \mid y_{t+1}^i = 1) \beta_{t+1}^i$$

Posterior decoding

- We can now calculate

$$P(y_t^k = 1 \mid \mathbf{x}) = \frac{P(y_t^k = 1, \mathbf{x})}{P(\mathbf{x})} = \frac{\alpha_t^k \beta_t^k}{P(\mathbf{x})}$$

- Then, we can ask
  - What is the most likely state at position *t* of sequence **x**:

$$k_t^* = \arg\max_k P(y_t^k = 1 \mid \mathbf{x})$$

  - Note that this is an MPA of a single hidden state, what if we want to a MPA of a whole hidden state sequence?
  - Posterior Decoding: $\left\{ y_t^{k_t^*} = 1 : t = 1 \cdots T \right\}$
  - This is different from MPA of a whole sequence of hidden states
  - This can be understood as *bit error rate* vs. *word error rate*

Example:
MPA of *X* ?
MPA of *(X, Y)* ?

x	y	P(x,y)
0	0	0.35
0	1	0.05
1	0	0.3
1	1	0.3

Viterbi decoding

- GIVEN $\mathbf{x} = x_1, \ldots, x_T$, we want to find $\mathbf{y} = y_1, \ldots, y_T$, such that $P(\mathbf{y}|\mathbf{x})$ is maximized:

$$\mathbf{y}^* = \text{argmax}_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x}) = \text{argmax}_{\pi} P(\mathbf{y}, \mathbf{x})$$

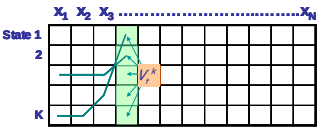
  - Let

$$V_t^k = \max_{\{y_1, \ldots, y_{t-1}\}} P(x_1, \ldots, x_{t-1}, y_1, \ldots, y_{t-1}, x_t, y_t^k = 1)$$

  = Probability of most likely **sequence of states** ending at state $y_t = k$

  - The recursion:

$$V_t^k = p(x_t \mid y_t^k = 1) \max_i a_{i,k} V_{t-1}^i$$



  - Underflows are a significant problem

$$p(x_1, \ldots, x_t, y_1, \ldots, y_t) = \pi_{y_1} a_{y_1, y_2} \cdots a_{y_{t-1}, y_t} b_{y_1, x_1} \cdots b_{y_t, x_t}$$

    - These numbers become extremely small – underflow
    - Solution: Take the logs of all values: $V_t^k = \log p(x_t \mid y_t^k = 1) + \max_i \left(\log(a_{i,k}) + V_{t-1}^i \right)$

Computational Complexity and implementation details

- What is the running time, and space required, for Forward, and Backward?

$$\alpha_t^k = p(x_t \mid y_t^k = 1) \sum_i \alpha_{t-1}^i a_{i,k}$$

$$\beta_t^k = \sum_i a_{k,i} p(x_{t+1} \mid y_{t+1}^i = 1) \beta_{t+1}^i$$

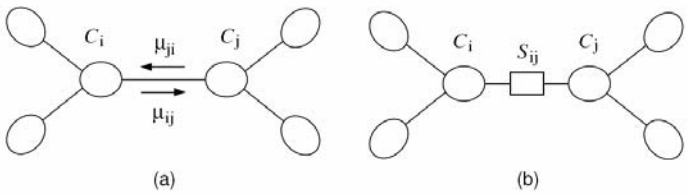
$$V_t^k = p(x_t \mid y_t^k = 1) \max_i a_{i,k} V_{t-1}^i$$

Time: $O(K^2 N)$; Space: $O(KN)$.

- Useful implementation technique to avoid underflows
  - Viterbi: sum of logs
  - Forward/Backward: rescaling at each position by multiplying by a constant

Shafer Shenoy for HMMs

- Recap: Shafer-Shenoy algorithm



  - Message from clique i to clique j:

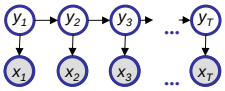
$$\mu_{i \to j} = \sum_{C_i \setminus S_{ij}} \psi_{C_i} \prod_{k \neq i} \mu_{k \to i}(S_{ki})$$

  - Clique marginal

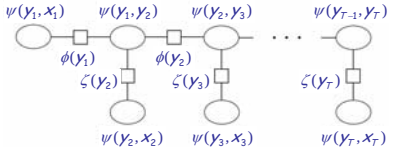
$$p(C_i) \propto \mu_{i \to j} = \psi_{C_i} \prod_k \mu_{k \to i}(S_{ki})$$

Shafer Shenoy for HMMs (cont.)

- A junction tree for the HMM

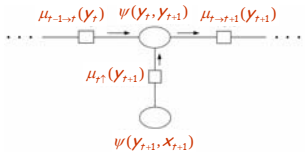


$\Rightarrow$



- Rightward pass

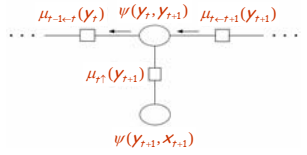
$$\mu_{t\to t+1}(y_{t+1}) = \sum_{y_t} \psi(y_t, y_{t+1})\mu_{t-1\to t}(y_t)\mu_{t\uparrow}(y_{t+1})$$
$$= \sum_{y_t} p(y_{t+1}\mid y_t)\mu_{t-1\to t}(y_t)p(x_{t+1}\mid y_{t+1})$$
$$= p(x_{t+1}\mid y_{t+1})\sum_{y_t} a_{y_t,y_{t+1}}\mu_{t-1\to t}(y_t)$$



  - This is exactly the *forward algorithm*!

- Leftward pass …

$$\mu_{t-1\leftarrow t}(y_t) = \sum_{y_{t+1}} \psi(y_t, y_{t+1})\mu_{t\leftarrow t+1}(y_{t+1})\mu_{t\uparrow}(y_{t+1})$$
$$= \sum_{y_{t+1}} p(y_{t+1}\mid y_t)\mu_{t\leftarrow t+1}(y_{t+1})p(x_{t+1}\mid y_{t+1})$$



  - This is exactly the *backward algorithm*!

Summary of the F-B algorithm

$$\alpha_t^k \overset{\text{def}}{=} \mu_{t-1\to t}(k) = P(x_1,\ldots,x_{t-1},x_t,y_t^k=1)$$
$$\alpha_t^k = p(x_t\mid y_t^k=1)\sum_i \alpha_{t-1}^i a_{i,k}$$
$$\beta_t^k \overset{\text{def}}{=} \mu_{t-1\leftarrow t}(k) = P(x_{t+1},\ldots,x_T\mid y_t^k=1)$$
$$\beta_t^k = \sum_i a_{k,i}p(x_{t+1}\mid y_{t+1}^i=1)\beta_{t+1}^i$$
$$\xi_t^{i,j} \overset{\text{def}}{=} \mu_{t-1\to t}(y_t^i=1)\mu_{t\leftarrow t+1}(y_t^j=1)p(x_{t+1}\mid y_{t+1})p(y_{t+1}\mid y_t)$$
$$= p(y_t^i=1,y_t^j=1,x_{1:T})$$
$$\xi_t^{i,j} = \alpha_t^i\beta_{t+1}^j a_{i,j}p(x_{t+1}\mid y_{t+1}^i=1)$$
$$\gamma_t^i \overset{\text{def}}{=} p(y_t^i=1\mid x_{1:T}) \propto \alpha_t^i\beta_t^i\sum_j \xi_t^{i,j}$$

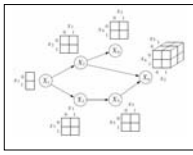
The matrix-vector form:

$$B_t(i) \overset{\text{def}}{=} p(x_t\mid y_t^i=1)$$
$$A(i,j) \overset{\text{def}}{=} p(y_{t+1}^j=1\mid y_t^i=1)$$

$$\alpha_t = \left(A^T\alpha_{t-1}\right).*B_t$$
$$\beta_t = A\left(\beta_{t+1}.*B_{t+1}\right)$$
$$\xi_t = \left(\alpha_t\left(\beta_{t+1}.*B_{t+1}\right)^T\right).*A$$
$$\gamma_t = \alpha_t.*\beta_t$$

Learning HMM: two scenarios

- Supervised learning: if only we knew the true state path then ML parameter estimation would be trivial
  - E.g., recall that for complete observed tabular BN:



$$\theta_{ijk}^{ML} = \frac{n_{ijk}}{\sum_{i,j',k} n_{ij'k}}$$

$$a_{ij}^{ML} = \frac{\#(i \to j)}{\#(i \to \bullet)} = \frac{\sum_n \sum_{t=2}^{T} y_{n,t-1}^i y_{n,t}^j}{\sum_n \sum_{t=2}^{T} y_{n,t-1}^i}$$

$$b_{ik}^{ML} = \frac{\#(i \to k)}{\#(i \to \bullet)} = \frac{\sum_n \sum_{t=1}^{T} y_{n,t}^i x_{n,t}^k}{\sum_n \sum_{t=1}^{T} y_{n,t}^i}$$

  - What if y is continuous? We can treat $\{(x_{n,t}, y_{n,t}) : t = 1:T, n = 1:N\}$ as $N \times T$ observations of, e.g., a GLIM, and apply learning rules for GLIM …
- Unsupervised learning: when the true state path is unknown, we can fill in the missing values using inference recursions.
  - The Baum Welch algorithm (i.e., EM)
    - Guaranteed to increase the log likelihood of the model after each iteration
    - Converges to local optimum, depending on initial conditions

The Baum Welch algorithm

- The complete log likelihood

$$\ell_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{y}) = \log p(\mathbf{x}, \mathbf{y}) = \log \prod_n \left(p(y_{n,1}) \prod_{t=2}^{T} p(y_{n,t} \mid y_{n,t-1}) \prod_{t=1}^{T} p(x_{n,t} \mid x_{n,t}) \right)$$

- The expected complete log likelihood

$$\langle \ell_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{y}) \rangle = \sum_n \left(\left\langle y_{n,1}^i \right\rangle_{p(y_{n,1} \mid \mathbf{x}_n)} \log \pi_i \right) + \sum_n \sum_{t=2}^{T} \left(\left\langle y_{n,t-1}^i y_{n,t}^j \right\rangle_{p(y_{n,t-1}, y_{n,t} \mid \mathbf{x}_n)} \log a_{i,j} \right) + \sum_n \sum_{t=1}^{T} \left(x_{n,t}^k \left\langle y_{n,t}^i \right\rangle_{p(y_{n,t} \mid \mathbf{x}_n)} \log b_{i,k} \right)$$

- EM
  - The E step

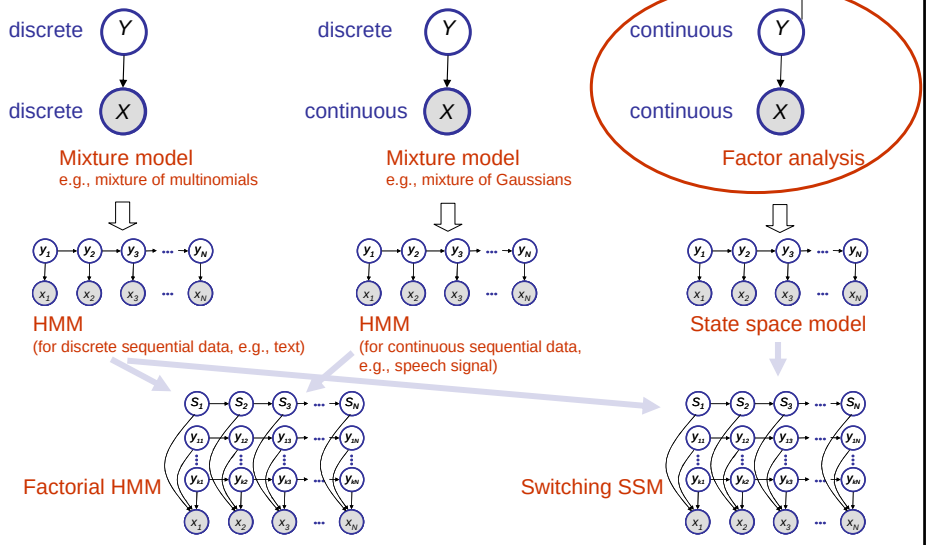
$$\gamma_{n,t}^i = \left\langle y_{n,t}^i \right\rangle = p(y_{n,t}^i = 1 \mid \mathbf{x}_n)$$

$$\xi_{n,t}^{i,j} = \left\langle y_{n,t-1}^i y_{n,t}^j \right\rangle = p(y_{n,t-1}^i = 1, y_{n,t}^j = 1 \mid \mathbf{x}_n)$$

  - The M step ("symbolically" identical to MLE)

$$\pi_i^{ML} = \frac{\sum_n \gamma_{n,1}^i}{N} \qquad a_{ij}^{ML} = \frac{\sum_n \sum_{t=2}^{T} \xi_{n,t}^{i,j}}{\sum_n \sum_{t=1}^{T-1} \gamma_{n,t}^i} \qquad b_{ik}^{ML} = \frac{\sum_n \sum_{t=1}^{T} \gamma_{n,t}^i x_{n,t}^k}{\sum_n \sum_{t=1}^{T-1} \gamma_{n,t}^i}$$

A road map to more complex dynamic models



Review: A primer to multivariate Gaussian

- Multivariate Gaussian density:

$$p(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\tfrac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

- A joint Gaussian:

$$p\left(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} \mid \mu, \Sigma\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} \middle| \begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right)$$

- How to write down $p(\mathbf{x}_1)$, $p(\mathbf{x}_1|\mathbf{x}_2)$ or $p(\mathbf{x}_2|\mathbf{x}_1)$ using the block elements in μ and Σ?
  - Formulas to remember:

$$p(\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2 \mid \mathbf{m}_2^m, \mathbf{V}_2^m) \qquad p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1 \mid \mathbf{m}_{1|2}, \mathbf{V}_{1|2})$$

$$\mathbf{m}_2^m = \mu_2 \qquad \mathbf{m}_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2)$$

$$\mathbf{V}_2^m = \Sigma_{22} \qquad \mathbf{V}_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

Review: The matrix inverse lemma

- Consider a block-partitioned matrix: $M = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$
- First we diagonalize M

$$\begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} \begin{bmatrix} I & 0 \\ -H^{-1}G & I \end{bmatrix} = \begin{bmatrix} E - FH^{-1}G & 0 \\ 0 & H \end{bmatrix}$$

  - Schur complement: $M/H = E - FH^{-1}G$
- Then we inverse, using this formula: $XYZ = W \Rightarrow Y^{-1} = ZW^{-1}X$

$$M^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -H^{-1}G & I \end{bmatrix} \begin{bmatrix} (M/H)^{-1} & 0 \\ 0 & H^{-1} \end{bmatrix} \begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} (M/H)^{-1} & -(M/H)^{-1}FH^{-1} \\ -H^{-1}G(M/H)^{-1} & H^{-1} + H^{-1}G(M/H)^{-1}FH^{-1} \end{bmatrix} = \begin{bmatrix} E^{-1} + E^{-1}F(M/E)^{-1}GE^{-1} & -E^{-1}F(M/E)^{-1} \\ -(M/E)^{-1}GE^{-1} & (M/E)^{-1} \end{bmatrix}$$

- Matrix inverse lemma

$$(E - FH^{-1}G)^{-1} = E^{-1} + E^{-1}F(H - GE^{-1}F)^{-1}GE^{-1}$$

Review: Some matrix algebra

- Trace and derivatives $\quad \mathrm{tr}[A] \stackrel{\mathrm{def}}{=} \sum_i a_{ii}$
  - Cyclical permutations

$$\mathrm{tr}[ABC] = \mathrm{tr}[CAB] = \mathrm{tr}[BCA]$$

  - Derivatives

$$\frac{\partial}{\partial A}\mathrm{tr}[BA] = B^T$$

$$\frac{\partial}{\partial A}\mathrm{tr}[x^T Ax] = \frac{\partial}{\partial A}\mathrm{tr}[xx^T A] = xx^T$$

- Determinants and derivatives

$$\frac{\partial}{\partial A}\log|A| = A^{-T}$$