# *COURSE 02457*

# Signal Processing in Non-linear Systems:

# Lecture 11

- Review of sounds and features
- The HMM generative model vs the mixture model
- Estimating probabilities of sequences in HMM's
- The Viterbi algorithm
- Estimating parameters in the HMM's
- The forward-backward algorithm for calculation of updates.

# Speech signals

- Speech signals are composed of sound sequences
- The sounds and the transitions between them serve as symbolic representation of information
- Speech sounds are produced and shaped by the human vocal tract
- The vocal tract is excited either by short burst of periodic stimulus or white noise. Voiced sounds are produce by airflow through tight vocal cords. Unvoiced sounds (or fricatives) are produced by turbulence.

*After L.R. Rabiner and R.W. Schaefer: Digital Processing of Speech Signals*

# Linear predictive modeling

- Linear system model (IIR model)

$$x(n) \;=\; \sum_{j=1}^{p} w_j x(n-j) + \epsilon(n)$$

- System parameter are estimated from short sequences (20-30 msec)in which the signal is quasi-stationary by the Levinson-Durbin algorithm. The autocorrelation function is defined

$$R(m) \;=\; 1/N \sum_{n=m+1}^{N} x(n)x(n-m) + \epsilon(n)$$

- and the least squares estimates of the parameters satisfy

$$R(m) \;=\; \sum_{j=1}^{p} \widehat{w}(j) R(m-j)$$

# Cepstral coefficients

- The cepstrum is defined as the inverse DFT of the log of the measured signal

$$C(m) = IDFT\,(log(|X(k))|)$$
$$X(k) = DFT(|x(m)|)$$

- Can be used to separate a slowly varying (in frequency space) envelope from a rapidly varying excitation (e.g., a periodic component with higher harmonics).
- The observation sequence consists of cepstral coefficients and their time derivatives.
- Observation sequences are grouped in symbols (numbered $k = 1, ..., K$) by a vector quantizer, e.g., k-means.

# The K-means algorithm

- The K-means algorithm is a simple $clustering$ algorithm aimed at minimizing the cost function for $K$ clusters,

$$E = \sum_{j=1}^{K} \sum_{n \in S_j} (\mathbf{x}_n - \boldsymbol{\mu}_j)^2$$

- ...where $\boldsymbol{\mu}_j$ is the mean of the data points associated most with the $j'th$ component $S_j$ (i.e. closest to)

$$\boldsymbol{\mu}_j = \frac{\sum_{n \in S_j} \mathbf{x}_n}{\sum_{n \in S_j} 1}$$

- Initialization is rather important, e.g., a cluster component which is never assigned any points will not be updated

# Simple Markov models

- Let $y^n$ be a sequence of symbols with $K$ states
- Let $a_{j,j'}$ be the probability of going from $j$ to $j'$.
- $a_{j,j'}$ is a stochastic matrix $\sum_{j'} a_{j,j'} = 1$
- $a$ can be estimated by maximum likelihood.

# Math trick: Lagrange multipliers

- Constrained optimization problem:
  Minimize $f(\mathbf{u})$ subject to $g(\mathbf{u}) = 0$ .
- The contraint defines a (hyper-) surface in $(\mathbf{u})$ space. The partial derivative in any point on this surface can be written: $\nabla f = \nabla_{||} f + \nabla_{\perp} f$
- For a move $\boldsymbol{\epsilon}$ inside the surface $g(\mathbf{u}+\boldsymbol{\epsilon}) = g(\mathbf{u})+\nabla g \cdot \boldsymbol{\epsilon} = 0$, hence $\nabla g \cdot \boldsymbol{\epsilon} = 0$, and the derivative of $g$ is orthogonal to the surface.
- This means that $on\ the\ surface$ we have : $\nabla_{\perp} f \propto \nabla g$. This means that there is a $\lambda$ so that $\nabla_{\perp} f = -\lambda^* \nabla g$
- In other words we can minimize $f$ inside the surface ($\nabla_{||} f = 0$), by solving for all $\lambda$'s.

$$\begin{aligned} \nabla \mathcal{L}(\mathbf{u}, \lambda) &= 0 \\ \nabla f + \lambda \nabla g &= 0 \end{aligned}$$

  ... and then solve for $\mathbf{u}(\lambda)$ to make sure the constraint is fulfilled (i.e. make sure that $\nabla_{||} f = 0$).
- For multiple constraints $\nabla \mathcal{L}(\mathbf{u}, \{\lambda_j\}) = f(\mathbf{u}) + \sum_j \lambda_j g_j(\mathbf{u})$

# Simple Markov models

- $a$ can be estimated by maximum likelihood.

$$\begin{aligned} P(\{y_t\}|a) &= P(y_1) \prod_{t=2}^{N} P(y_t|y_{t-1}, a) \\ &= P(y_1) \prod_{<j,j'>} (a_{j,j'})^{\mathrm{n_{j,j'}}} \end{aligned}$$

- $n_{j,j'}$ is the occurrence of the transition.
- The (neg) likelihood is minimized subject to the condition $\sum_{j'} a_{j,j'} = 1$ leads to the solution,

$$\widehat{a}_{j,j'} = \frac{n_{j,j'}}{\sum_{j'} n_{j,j'}}$$

# Why Hidden Markov Models?

- Some distributions over sequences cannot be modeled with short range Markov chains. A more flexible model which is inherently short range is obtained by considering hidden Markov models.
- We need a model that is a mixture of two distributions
- The simplest HMM:
  - Discrete hidden states: $x_t \in \{1, ..., S\}$.
  - Discrete observations: $y_t \in \{1, ..., K\}$.
  - Probability of observation if state is known: $b_{k,j} = P(y_t = k | x_t = j)$
  - Probability of state transition: $a_{j,j'} = P(x_{t+1} = j' | x_t = j)$
  - Probability of starting symbols: $\pi(i) = P(y_1 = i)$
  - Parameters $\theta = \{a, b, \pi\}$
- Simultaneous probability of an observed sequence $and$ states

$$P(y_{1:T}, x_{1:T}|\theta) = P(x_1|\pi)P(y_1|x_1, b) \prod_{t=2}^{T} P(y_t|x_t, b)P(x_t|x_{t-1}, a)$$

$$P(y_{1:T}, x_{1:T}|\theta) = \pi_{x_1} b_{y_1,x_1} \prod_{t=2}^{T} b_{y_t,x_t} a_{x_{t-1},x_t}$$

# Hidden markov models vs. mixture model

- Simulating these processes are two-step procedures: First generate set of hidden variables, then generate observations.
- Probability of observations

$$p(\mathbf{x}) = \sum_{k=1}^{K} P(k) p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Probability of a sequence with known states

$$P(y_{1:T}, x_{1:T}|\theta) = P(x_1|\pi) P(y_1|x_1, b) \prod_{t=2}^{T} P(y_t|x_t, b) P(x_t|x_{t-1}, a)$$

- Graphical representation

# Estimating the most likely hidden state sequence

- Option: Search through all possible ($K^T$) hidden state combinations?
- The Viterbi algorithm makes use of dynamic programming for given observation sequence $y_{1:T}$
  - Define

$$\delta_t(i) = \max_{x_1,..,x_{t-1}} P(x_1, ..., x_{t-1}, x_t = i, y_1, ..., y_t|\theta)$$

  - Recursion

$$\begin{aligned}\delta_{t+1}(j) &= \left(\max_i \delta_t(i)a_{i,j}\right) b_{y_{t+1},j} \\ \psi_{t+1}(j) &= \arg\max_i \delta_t(i)a_{i,j}\end{aligned}$$

  is obtained by considering

$$\begin{aligned}&P(x_{1:t}, x_{t+1}, y_{1:t+1}|\theta) = \\ &P(x_{1:t-1}, x_t, y_{1:t}|\theta)P(x_{t+1}|x_t, \theta)P(y_{t+1}|x_{t+1}, \theta)\end{aligned}$$

  and using

$$\max_{a,b} f(a, b)g(b) = \max_b(\max_a f(a, b) \cdot g(b))$$

- termination and 'backtrack' to find the best sequence

$$\begin{aligned}x_T^* &= \arg\max_i \delta_t(i)a_{i,j} \\ x_t^* &= \psi_{t+1}(x_{t+1}^*)\end{aligned}$$

# Probability of an observation sequence

- Could we use the Viterbi algorithm to find the most likely state sequence ($x^*_{1:T} = f(y_{1:T}, \theta)$) and use probability of the sequence with this hidden sequence?:

$$P(y_{1:T}, x^*_{1:T}|\theta) \;=\; P(y_1, \pi) \prod_{t=2}^{T} P(y_t|x^*_t, b) P(x^*_t|x^*_{t-1}, a)$$

$$P(y_{1:T}, x^*_{1:T}|\theta) \;=\; \pi_{y_1} \prod_{t=2}^{T} b_{y_t, x^*_t} a_{x^*_{t-1}, x^*_t}$$

- No, the probability is given by the sum over all hidden states

$$P(y_{1:T}|\theta) \;=\; \sum_{x_{1:T}} P(y_{1:T}, x_{1:T}|\theta)$$

- How many operations are need for such a calculation?. If the number of hidden states is $K$, there are $K^T$ possible configurations.

# Probability of an observation sequence con't.

- The forward summation, by definition

$$\begin{aligned} P(y_{1:t+1}, x_{t+1}|\theta) &= \sum_{x_t} P(y_{1:t}, y_{t+1}, x_t, x_{t+1}) \\ &= \sum_{x_t} P(y_{1:t}, x_t, x_{t+1}) P(y_{t+1}|x_{t+1}) \\ &= \sum_{x_t} P(y_{1:t}, x_t) P(x_{t+1}|x_t) P(y_{t+1}|x_{t+1}) \end{aligned}$$

- Hence we have a recursion again, define

$$\begin{aligned} \alpha_{t+1}(x_{t+1}) &= (\sum_{x_t} \alpha_t(x_t) a_{x_t, x_{t+1}}) b_{x_{t+1}}(y_{t+1}) \\ \alpha_{t+1}(j) &= \sum_i \alpha_t(i) a_{i,j} b_j(y_{t+1}) \end{aligned}$$

- with the termination

$$\begin{aligned} P(y_{1:T}|\theta) &= \sum_{x_T} P(y_{1:T}, x_T|\theta) \\ &= \sum_j \alpha_t(j) \end{aligned}$$

- The number of operations is roughly $T * K^2$, since we have a sum over the $K$ symbols for each $\alpha_{t+1}(j)$

# Estimation of HMM parameters

- The log-likelihood function reads

$$\begin{aligned}\log P(y_{1:T}|\theta) &= \log \sum_{x_{1:T}} P(y_{1:T}, x_{1:T}|\theta) \\ &= \log \sum_{x_{1:T}} \pi(x_1) \prod_{j,j'} a_{j,j'}^{n_{j,j'}(x)} \prod_{k,j} b_{k,j}^{m_{k,j}(x,y)}\end{aligned}$$

  where $n_{j,j'}(x)$ is the number of transitions $j \longrightarrow j'$, and $m_{k,j}(x)$ is the number of times the symbol $k$ is emitted in the state $j$.

- Introducing Lagrange multipliers to cope with the contraints $\sum_{j'} a_{j,j'} = 1$ and $\sum_k b_{k,j} = 1$, we will minimize

$$\begin{aligned}\mathcal{L} &= \log \sum_{x_{1:T}} \pi(x_1) \prod_{j,j'} a_{j,j'}^{n_{j,j'}(x)} \prod_{k,j} b_{k,j}^{m_{k,j}(x,y)} \\ &+ \sum_j \lambda_j (\sum_{j'} a_{j,j'} - 1) \\ &+ \sum_j \kappa_j (\sum_k b_{k,j} - 1)\end{aligned}$$

# Estimation of HMM parameters

- Minimizing $\mathcal{L}$ produces the result

$$\widehat{a}_{j,j'} = \frac{\sum_{x_{1:T}} n_{j,j'}(x) P(x_{1:T}|y_{1:T},\theta)}{\sum_{j''}\sum_{x_{1:T}} n_{j,j''}(x) P(x_{1:T}|y_{1:T},\theta))}$$

$$\widehat{b}_{k,j} = \frac{\sum_{x_{1:T}} m_{k,j}(x) P(x_{1:T}|y_{1:T},\theta)}{\sum_{k'}\sum_{x_{1:T}} m_{k,j}(x) P(x_{1:T}|y_{1:T},\theta))}$$

- or ...

$$\widehat{a}_{j,j'} = \frac{\langle n_{j,j'}(x)|y,\theta\rangle_x}{\sum_{j''}\langle n_{j,j''}(x)|y,\theta\rangle_x}$$

$$\widehat{b}_{k,j} = \frac{\langle m_{k,j}(x)|y,\theta\rangle_x}{\sum_{k}\langle m_{k,j}(x)|y,\theta\rangle_x}$$

# Computing the HMM expectations

- How to calculate the expectation?

$$\begin{aligned}\langle n_{j,j'}|y,\theta\rangle &= \sum_{x_{1:T}} n_{j,j'}(x)P(x|y,\theta) \\ &= \sum_{t=1}^{T-1} P(x_t = j, x_{t+1} = j'|y,\theta)\end{aligned}$$

- we can use the forward calculation (the $\alpha$'s) in combination with a backward recursion, based on

$$\begin{aligned}P(y_{t:T}|x_{t-1}) &= \sum_{x_t} P(y_{t:T}|x_t)P(x_t|x_{t-1}) \\ &= \sum_{x_t} P(y_t|x_t)P(y_{t+1:T}|x_t)P(x_t|x_{t-1})\end{aligned}$$

- if we define $\beta_t(j) = P(y_{t:T}|x_{t-1} = j)$, this implies

$$\beta_t(j) = \sum_i b(y_{t:T}|i)a(i,j)\beta_{t+1}(i)$$

# Computing the HMM expectations

- How to calculate the expectation?

$$\begin{aligned}\langle n_{j,j'}|y,\theta\rangle &= \sum_{x_{1:T}} n_{j,j'}(x)P(x|y,\theta) \\ &= \sum_{t=1}^{T-1} P(x_t=j,x_{t+1}=j'|y,\theta)\end{aligned}$$

- we can use the forward calculation (the $\alpha$'s) in combination with the backward recursion, $\beta_t(j) = P(y_{t:T}|x_{t-1} = j)$,

$$P(x_t=j,x_{t+1}=j'|y,\theta) = \sum_i b(y_{t:T}|i)a(i,j)\beta_{t+1}(i)$$

- we note that

$$\begin{aligned}& P(x_t=j,x_{t+1}=j'|y,\theta) \\ =\ & P(x_t=j,x_{t+1}=j',y|\theta)/P(y|\theta) \\ =\ & P(y|x_t,x_{t+1})P(x_{t+1}|x_t)P(x_t)/P(y|\theta) \\ =\ & P(y_{1:t}|x_t)P(y_{t+1}|x_{t+1})P(y_{t+2:T}|x_{t+1})P(x_{t+1}|x_t)\frac{P(x_t)}{P(y|\theta)} \\ =\ & \alpha_t(j)b_{y_{t+1},j'}\beta_{t+1}(j')a_{j,j'}/P(y|\theta)\end{aligned}$$