Probabilistic Modelling of Sequences: Learning

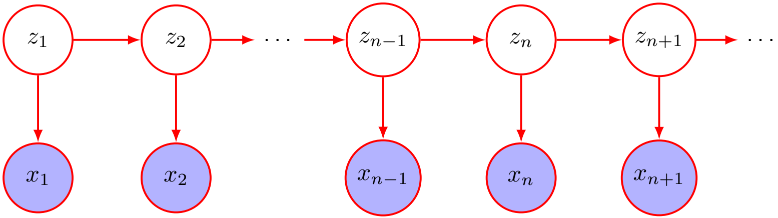
TTT4185 Machine Learning for Signal Processing

Giampiero Salvi

Department of Electronic Systems
NTNU

HT2020

HMM Inference: Learning

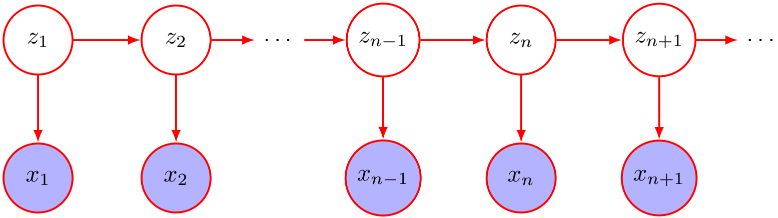


- Given observations X update model parameters

$$\theta = \{\pi, A, \phi\}$$

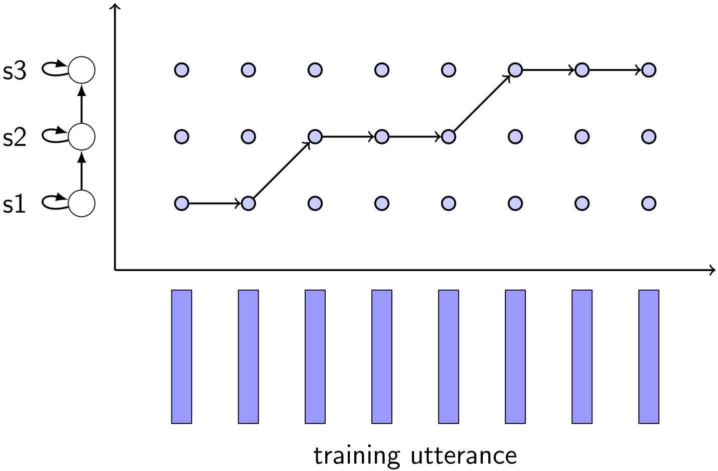
- to maximise either:
  - model fit to data (e.g. likelihood, posterior)
  - classification performance (discriminative training)

HMM Inference: Learning

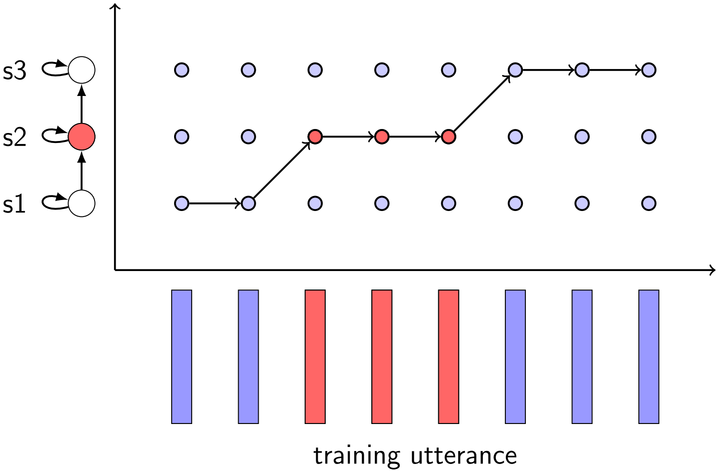


- problem: incomplete data, state sequence Z
- there is no closed-form solution
- only iterative procedures: given θ^{old} how to estimate θ^{new}

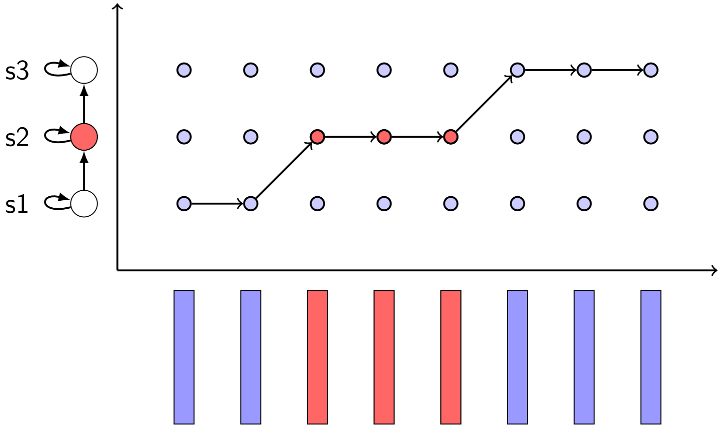
Viterbi training (simple approach)



Viterbi training (simple approach)



Viterbi training (simple approach)



training utterance

problem: sensitive to misalignments

but still used for ANN/DNN training

HMM Inference: Learning

Latent variables $\rightarrow$ Expectation Maximisation

- locally maximise the likelihood of the complete data X, Z
- efficient solution with froward-backward or Baum-Welch algorithm[1]
- general idea: sum over all possible paths weighted by posterior probability of the path
- also: every observation vector contributes to all parameter updates

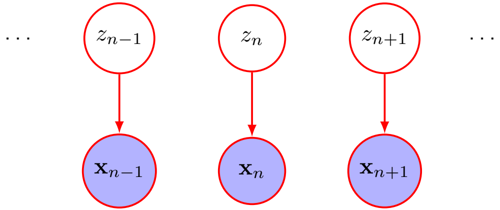
[1] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains". In: *Ann. Math. Statist.* 41.1 (1970), pp. 164–171.

Expectation Maximization for Mixture Models

$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(\mathbf{x}|\theta_k),$$

$$\theta = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_K\},$$

$$\sum_{k=1}^{K} \pi_k = 1$$

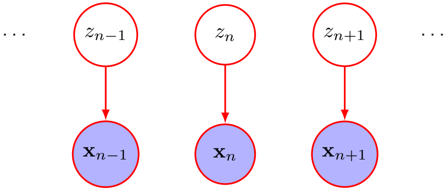


- augment the data with the latent variables: $z_i \in \{1, \ldots, K\}$ assignment of each data point x_i to a component of the mixture
- interpret the mixture as marginal of the joint

$$P(\mathbf{x}|\theta) = \sum_{z} P(\mathbf{x}, \mathbf{z}|\theta)$$

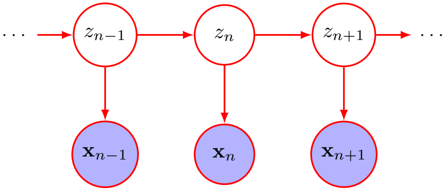
Mixture Models vs Hidden Markov Models

Mixture Model



EM Algorithm

Hidden Markov Model



Baum-Welch Algorithm (EM)

Expectation Maximization: Idea

Ideally we would like to maximize:

$$\log p(X|\theta) = \log \left\{ \sum_Z p(X, Z|\theta) \right\}$$

with $X = \{x_1, \ldots, x_N\}, \quad Z = \{z_1, \ldots, z_N\}$
... but log of sum hard to optimize
Instead optimize likelihood of complete data:

$$\log p(X, Z|\theta)$$

Z not known, but we can compute posterior given current model $p(Z|X, \theta^{\text{old}})$
Optimize the expected value of the likelihood:

$$\mathcal{Q}(\theta, \theta^{\text{old}}) = \sum_Z p(Z|X, \theta^{\text{old}}) \log p(X, Z|\theta)$$

Expectation Maximization in Practice

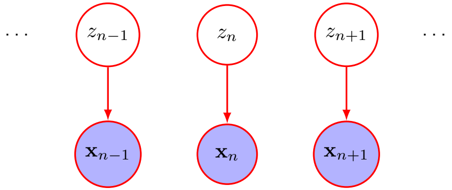
1. Initialization: choose initial value of θ^{old}
2. Expectation step: evaluate posterior $p(Z|X, \theta^{\text{old}})$
3. Maximization step: evaluate θ^{new} with:

$$\begin{aligned}\theta^{\text{new}} &= \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{\text{old}}) \\ &= \arg\max_{\theta} \sum_{Z} p(Z|X, \theta^{\text{old}}) \log p(X, Z|\theta)\end{aligned}$$

4. Check for convergence, otherwise $\theta^{\text{old}} \leftarrow \theta^{\text{new}}$ and go to step 2.

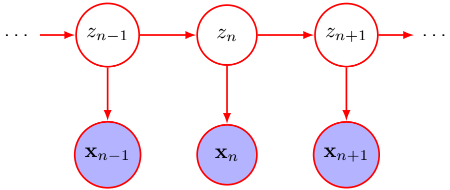
Mixture Models vs Hidden Markov Models

Mixture Model



EM Algorithm

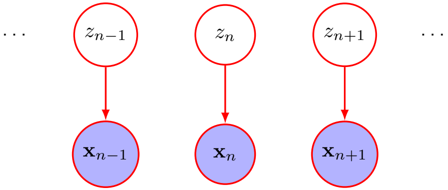
Hidden Markov Model



Baum-Welch Algorithm (EM)

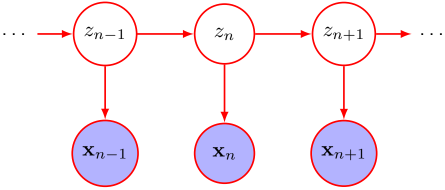
Mixture Models vs Hidden Markov Models

Mixture Model



EM Algorithm

Hidden Markov Model



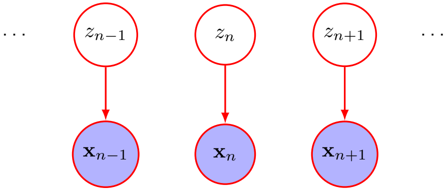
Baum-Welch Algorithm (EM)

1. Posterior of latent variable z_n depends on full sequence X

$$\gamma_n(k) = P(z_n = k \mid X, \theta^{(t)}) \neq P(z_n = k \mid x_n, \theta^{(t)})$$

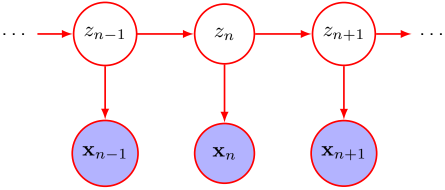
Mixture Models vs Hidden Markov Models

Mixture Model



EM Algorithm

Hidden Markov Model



Baum-Welch Algorithm (EM)

1. Posterior of latent variable z_n depends on full sequence X

$$\gamma_n(k) = P(z_n = k \mid X, \theta^{(t)}) \neq P(z_n = k \mid x_n, \theta^{(t)})$$

2. We also need the posterior of two subsequent latent variables

$$\xi_n(i,j) = P(z_{n-1} = s_i, z_n = s_j | X, \theta)$$

Baum-Welch E-Step

Estimate posterior $P(Z|X, \theta^{\text{old}})$ or, at least the sufficient statistics given:

- current model parameters
- full sequence of observations X

1. Posterior of latent variable z_n

$$\gamma_n(i) = P(z_n = s_i | X, \theta)$$

2. Posterior of two subsequent latent variables

$$\xi_n(i, j) = P(z_{n-1} = s_i, z_n = s_j | X, \theta)$$

Baum-Welch E-Step

Estimate posterior $P(Z|X, \theta^{\text{old}})$ or, at least the sufficient statistics given:

- current model parameters
- full sequence of observations X

1. Posterior of latent variable z_n

$$\gamma_n(i) = P(z_n = s_i | X, \theta)$$

2. Posterior of two subsequent latent variables

$$\xi_n(i,j) = P(z_{n-1} = s_i, z_n = s_j | X, \theta)$$

Note: Huang, Acero and Hon call this $\gamma_n(i,j)$

Note: posteriors conditioned to full sequence X not only x_n

Baum-Welch M-Step

Weighted Maximum Likelihood estimates:

Emission probabilities:

Same as mixture model given $\gamma_n(i) = P(z_n = s_j | X, \theta)$

Transition probabilities

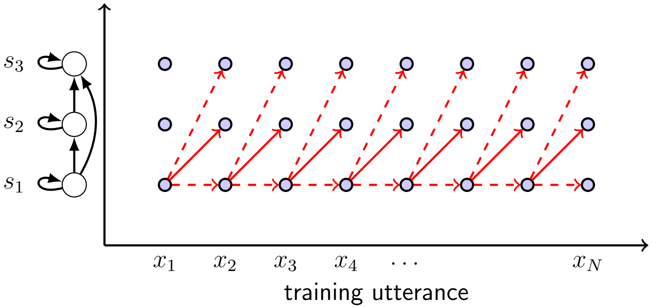
$$a_{ij}^{\text{new}} = \frac{E\left[s_i \to s_j | X, \theta^{\text{old}}\right]}{E\left[s_i \to s_{\text{any}} | X, \theta^{\text{old}}\right]} = \frac{\sum_{n=2}^{N} \xi_n(i,j)}{\sum_{n=2}^{N} \sum_{k=1}^{M} \xi_n(i,k)}$$

or, equivalently,

$$= \frac{E\left[s_i \to s_j | X, \theta^{\text{old}}\right]}{E\left[s_i | X, \theta^{\text{old}}\right]} = \frac{\sum_{n=2}^{N} \xi_n(i,j)}{\sum_{n=1}^{N-1} \gamma_n(i)}$$

Expectations are over the posteriors $P(Z | X, \theta^{\text{old}})$.

Example: Transition Probability



$$a_{12}^{\text{new}} = \frac{E\left[s_1 \rightarrow s_2 | X, \theta^{\text{old}}\right]}{E\left[s_1 \rightarrow s_{\text{any}} | X, \theta^{\text{old}}\right]} = \frac{\sum_{n=2}^{N} \xi_n(1,2)}{\sum_{n=2}^{N} \sum_{k=1}^{3} \xi_n(1,k)}$$

$$= \frac{E\left[s_1 \rightarrow s_2 | X, \theta^{\text{old}}\right]}{E\left[s_1 | X, \theta^{\text{old}}\right]} = \frac{\sum_{n=2}^{N} \xi_n(1,2)}{\sum_{n=1}^{N-1} \gamma_n(1)}$$

Example: Transition Probability

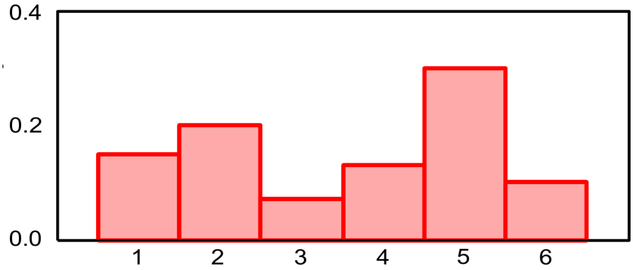
- $\sum_{n=2}^{N} \xi_n(i,j)$ is the expected number of transitions between state s_i and s_j (given X and θ^{old})
- $\sum_{n=1}^{N-1} \gamma_n(i)$ is the expected number of times we are in state s_i (given X and θ^{old})
- we never take a hard decision on when the transition happened

Emission probabilities

- Discrete HMMs (DHMMs)
  - vector quantisation
- Continuous HMMs
  - Single Gaussian $\phi_j(x_n) = N(x_n|\mu_j, \Sigma_j)$
  - Gaussian Mixture
- Semi-continuous HMMs (SCHMMs)

Discrete HMMs

- quantise feature vectors
- observation: sequence of discrete symbols
- $\phi_j(x_n)$ simple discrete probability distribution
- problem: quantisation error



Discrete HMMs: learn $\phi_j(x_n)$

Remember that

$$\gamma_n(j) = P(z_n = s_j | X, \theta)$$

are the posteriors of the latent variable

Update rule:

$$\phi_j(x_n = k) = p(x_n = k | z_n = s_j) = \frac{E[x_n = k, z_n = s_j]}{E[z_n = s_j]} = \frac{\sum_{n:(x_n=k)} \gamma_n(j)}{\sum_{n=1}^{N} \gamma_n(j)}$$

HMMs with Gaussian Emission Probability

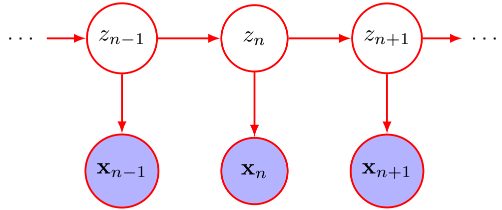
$$\phi_j(x_n) = N(x_n|\mu_j, \Sigma_j)$$

Update rules:

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_n(j) x_n}{\sum_{n=1}^{N} \gamma_n(j)}$$

$$\Sigma_j = \frac{\sum_{n=1}^{N} \gamma_n(j) \left(x_n - \mu_j\right) \left(x_n - \mu_j\right)^T}{\sum_{n=1}^{N} \gamma_n(j)}$$

Calculate sufficient statistics



$$\begin{aligned}\gamma_n(i) &= P(z_n = s_i | X, \theta) \\ \xi_n(i,j) &= P(z_{n-1} = s_i, z_n = s_j | X, \theta)\end{aligned}$$

We can do this with the help of the forward and backward variables:

$$\begin{aligned}\alpha_n(i) &= P(x_1, \ldots, x_n, z_n = s_i | \theta) \\ \beta_n(i) &= P(x_{n+1}, \ldots, x_N | z_n = s_i, \theta)\end{aligned}$$

Calculate γ (forward-backward)

$$\gamma_n(i) = P(z_n = s_i | X, \theta)$$

Calculate γ (forward-backward)

$$
\begin{aligned}
\gamma_n(i) &= P(z_n = s_i | X, \theta) \\
&= \frac{p(X, z_n = s_j | \theta)}{p(X|\theta)}
\end{aligned}
$$

Calculate γ (forward-backward)

$$\begin{aligned}
\gamma_n(i) &= P(z_n = s_i | X, \theta) \\
&= \frac{p(X, z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \ldots, x_n, x_{n+1}, \ldots, x_N, z_n = s_j | \theta)}{p(X|\theta)}
\end{aligned}$$

Calculate γ (forward-backward)

$$
\begin{aligned}
\gamma_n(i) &= P(z_n = s_i|X, \theta) \\
&= \frac{p(X, z_n = s_j|\theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \ldots, x_n, x_{n+1}, \ldots, x_N, z_n = s_j|\theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \ldots, x_n, x_{n+1}, \ldots, x_N|z_n = s_j, \theta)P(z_n = s_j|\theta)}{p(X|\theta)}
\end{aligned}
$$

Calculate γ (forward-backward)

$$
\begin{aligned}
\gamma_n(i) &= P(z_n = s_i | X, \theta) \\
&= \frac{p(X, z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \ldots, x_n, x_{n+1}, \ldots, x_N, z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \ldots, x_n, x_{n+1}, \ldots, x_N | z_n = s_j, \theta) P(z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \ldots, x_n | z_n = s_j, \theta) p(x_{n+1}, \ldots, x_N | z_n = s_j, \theta) P(z_n = s_j | \theta)}{p(X|\theta)}
\end{aligned}
$$

Calculate γ (forward-backward)

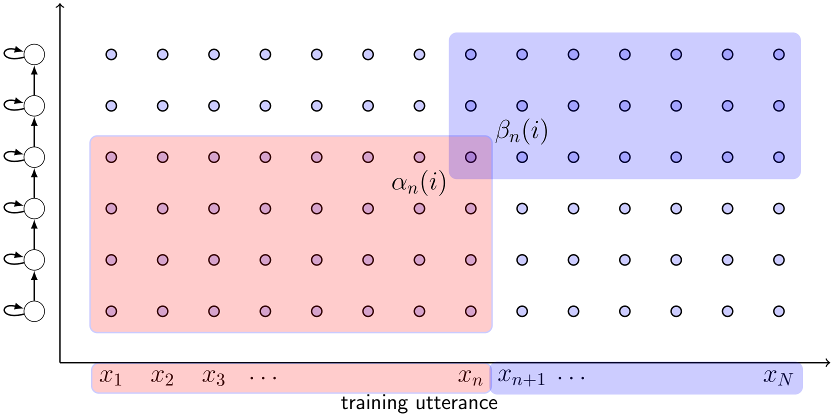
$$
\begin{aligned}
\gamma_n(i) &= P(z_n = s_i | X, \theta) \\
&= \frac{p(X, z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \dots, x_n, x_{n+1}, \dots, x_N, z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \dots, x_n, x_{n+1}, \dots, x_N | z_n = s_j, \theta) P(z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \dots, x_n | z_n = s_j, \theta) p(x_{n+1}, \dots, x_N | z_n = s_j, \theta) P(z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \dots, x_n, z_n = s_j | \theta) p(x_{n+1}, \dots, x_N | z_n = s_j, \theta)}{p(X|\theta)}
\end{aligned}
$$

Calculate γ (forward-backward)

$$
\begin{aligned}
\gamma_n(i) &= P(z_n = s_i | X, \theta) \\
&= \frac{p(X, z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \ldots, x_n, x_{n+1}, \ldots, x_N, z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \ldots, x_n, x_{n+1}, \ldots, x_N | z_n = s_j, \theta) P(z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \ldots, x_n | z_n = s_j, \theta) p(x_{n+1}, \ldots, x_N | z_n = s_j, \theta) P(z_n = s_j | \theta)}{p(X|\theta)} \\
&= \frac{p(x_1, \ldots, x_n, z_n = s_j | \theta) p(x_{n+1}, \ldots, x_N | z_n = s_j, \theta)}{p(X|\theta)} \\
&= \frac{\alpha_n(i) \beta_n(i)}{\sum_i \alpha_N(i)}
\end{aligned}
$$

Calculate γ: Illustration

$$\gamma_n(i) = P(z_n = s_i | X, \theta) = \frac{\alpha_n(i)\beta_n(i)}{\sum_i \alpha_N(i)}$$



Calculate ξ (forward-backward)

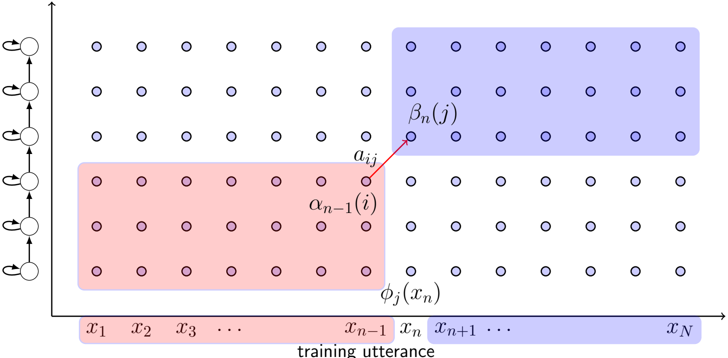
$$\xi_n(i,j) \quad = \quad P(z_n = s_j, z_{n-1} = s_i | X, \theta)$$

Calculate ξ (forward-backward)

$$
\begin{aligned}
\xi_n(i,j) &= P(z_n = s_j, z_{n-1} = s_i | X, \theta) \\
&= \frac{P(z_n = s_j, z_{n-1} = s_i, X | \theta)}{P(X|\theta)}
\end{aligned}
$$

Calculate ξ (forward-backward)

$$\begin{aligned}\xi_n(i,j) &= P(z_n = s_j, z_{n-1} = s_i | X, \theta) \\ &= \frac{P(z_n = s_j, z_{n-1} = s_i, X|\theta)}{P(X|\theta)} \\ \cdots &= \frac{\alpha_{n-1}(i)\ a_{ij}\ \phi_j(x_n)\ \beta_n(j)}{\sum_{k=1}^{M} \alpha_N(k)}\end{aligned}$$



Baum-Welch: Properties

instance of Expectation Maximisation:

- iterative procedure
- guaranteed to convert to local maximum of the likelihood $P(X|\theta^{\text{new}})$
- sensitive to initialisation
- update formulae for emission probability model $\phi_j(x_n)$ same as for mixture models (with new version of posteriors)

Numerical Problems

Product of many probabilities.
Solution: work in log domain

linear domain

$$\alpha_1(j) = \pi_j \phi_j(x_1)$$

$$\alpha_n(j) = \phi_j(x_n) \sum_{i=1}^{M} \alpha_{n-1}(i) a_{ij}$$

$$\beta_N(i) = 1$$

$$\beta_n(i) = \sum_{j=1}^{M} a_{ij} \phi_j(x_{n+1}) \beta_{n+1}(j)$$

log domain

$$\alpha'_1(j) = \pi'_j + \phi'_j(x_1)$$

$$\alpha'_n(j) = \log \sum_{i=1}^{M} e^{\left(\alpha'_{n-1}(i) + a'_{ij}\right)} + \phi'_j(x_n)$$

$$\beta'_N(i) = 0$$

$$\beta'_n(i) = \log \sum_{j=1}^{M} e^{\left(a'_{ij} + \phi'_j(x_{n+1}) + \beta'_{n+1}(j)\right)}$$

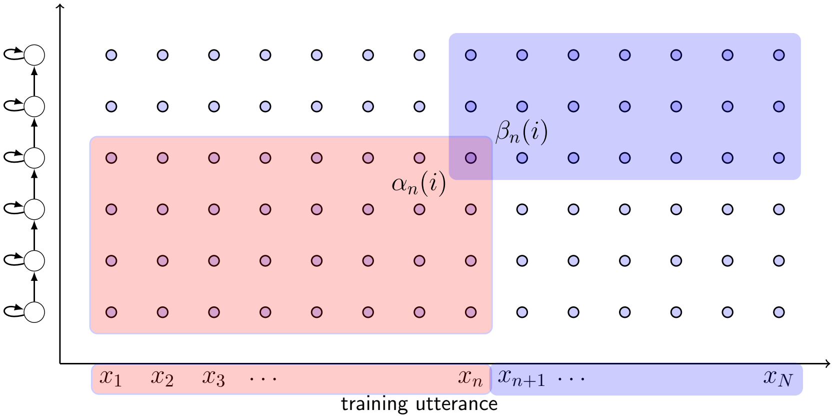
Train on several utterances

Set of utterances $X^1, X^2, \dots, X^U$

- cannot concatenate them: need to calculate α, β, γ and ξ each time

Each utterance corresponds to several models

- reuse model states (sentence $\rightarrow$ words $\rightarrow$ phonemes)



Concatenating HMMs

Utterance to words:

sil one zero one three sil

Words to phones

sil w ah n sp z iy r ow sp w ah n sp th r iy sp sil

Phones to states

sil0 sil1 sil2 w0 w1 w2 ah0 ah1 ah2 n0 n1 n2 sp0 z0 z1 z2 iy0 iy1 iy2 r0 r1 r2 ow0 ow1 ow2 sp0 w0 w1 w2 ah0 ah1 ah2 n0 n1 n2 sp0 th0 th1 th2 r0 r1 r2 iy0 iy1 iy2 sp0 sil0 sil1 sil2

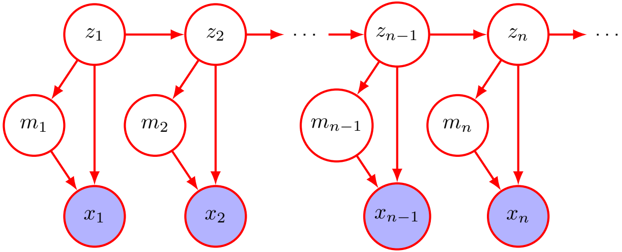
HMMs with Mixture Emission Probability

Often the Emission probability is modelled as a Mixture of Gaussians

$$\phi_j(x_n) = \sum_{k=1}^{K} w_{jk} N(x_n|\mu_{jk}, \Sigma_{jk})$$

$$\sum_{k=1}^{M} w_{jk} = 1$$

HMMs with Mixture Emission Probability



Emission:

$$
\begin{aligned}
p(x_n|z_n, m_n) &= \mathcal{N}(x_n; \mu_{z_n,m_n}, \Sigma_{z_n,m_n}) \\
p(m_n|z_n) &= W(m_n, z_n)
\end{aligned}
$$

Semi-Continuous HMMs

- All Gaussian distributions in a pool of pdfs
- each $\phi_j(x_n)$ is a discrete probability distribution over the pool of Gaussians
- similar to quantisation, but probabilistic
- used for sharing parameters

Hybrid HMM+Multi Layer Perceptron

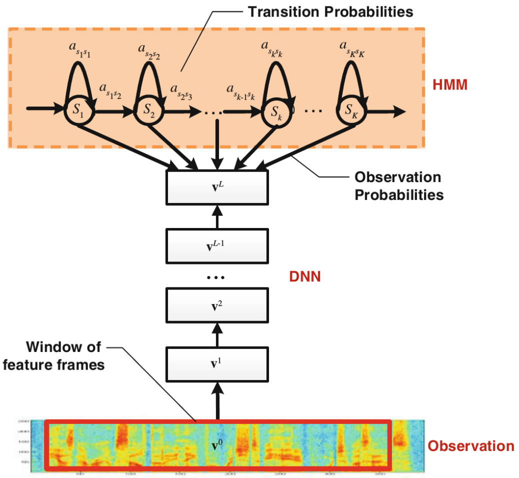


Figure from Yu and Deng

Combining probabilities

- HMMs use likelihoods $P(\text{sound}|\text{state})$
- MLPs and DNNs estimate posteriors $P(\text{state}|\text{sound})$

We can combine with Bayes:

$$P(\text{sound}|\text{state}) = \frac{P(\text{state}|\text{sound})P(\text{sound})}{P(\text{state})}$$

- $P(\text{state})$ can be estimated from the training set
- $P(\text{sound})$ is constant and can be ignored

Use scaled likelihoods:

$$\bar{P}(\text{sound}|\text{state}) = \frac{P(\text{state}|\text{sound})}{P(\text{state})}$$