Appendix A. Continuous Density Hidden Markov Model (CD-HMM)

Hidden Markov Model (HMM) is a very well known and established modeling method for many years now. It is well documented in books and tutorial papers. A short summary of the method is brought here to help the un-informed reader. Some practical implementation information is included in this appendix.

HMM is a stochastic function of a Markov chain. As such, it is composed of two elements: a Markov process and a set of stochastic functions, or output probabilities [Rabiner et al., 1993], [Dugad et al., 1996]. The name HMM stands from the fact that the model assumes the process to be Markovian and the fact that there is no direct way of determining the state the model is in at any given time. There are two kinds of output probabilities to consider: discrete probability - in Discrete Density HMM (DD-HMM), and continuous probability - in Continuous Density HMM (CD-HMM).

Two kinds of model architectures are often related with speaker recognition systems: Left-to-Right model, where the transitions between states are constrained to left-to-right direction (suitable for text-dependent tasks). The other architecture is Ergodic, where all the state transitions are allowed (suitable for text-independent tasks). In our text-dependent speaker verification system, we employed a left-to-right CD-HMM, which trains models using *multiple observations* training (enrollment of several sequences/utterances).

A CD-HMM is characterized by the following parameters:

(1) N, the number of states in the model. We label the individual states as $\{1,2,...,N\}$, and denote the state at time t as q_t.

(2) The state-transition probability distribution matrix $\mathbf{A} = \{a_{ij}\}$ where

$$a_{ij} = P[q_{t+1} = j \mid q_t = i]; \quad 1 \le i, j \le N. \tag{A.1}$$

(3) The initial state distribution $\pi = \{\pi_i\}$ in which

$$\pi_i = P[q_1 = i]; \quad 1 \le i \le N. \tag{A.2}$$

(4) When the observations are continuous, the output of state j when the observation is $\mathbf{o}$ is represented by means of the Probability Density Function (PDF): $b_j(\mathbf{o})$; $1 \le j \le N$

The PDF is estimated here by a linear combination of M Gaussians.

$$b_j(\mathbf{o}) = \sum_{k=1}^{M} c_{jk} N(\mathbf{o}, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})$$

$$N(\mathbf{o}, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_{jk}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{o} - \boldsymbol{\mu}_{jk})^T \Sigma_{jk}^{-1} (\mathbf{o} - \boldsymbol{\mu}_{jk})\right\}$$

where, for every state j $(j = 1,2,\ldots,N)$ and mixture k ($k = 1,2,\ldots,M$), the output model parameters are:

- c_{jk} - the mixture coefficients of the linear combination ($\sum_{k=1}^{M} c_{jk} = 1$),
- $\boldsymbol{\mu}_{jk}$ - the expectation vector, and
- $\boldsymbol{\Sigma}_{jk}$ - the covariance matrix.

Finally, the CD-HMM model λ is represented by the parameters:

$$\lambda = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{c}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$$

In general, the covariance matrix $\boldsymbol{\Sigma}_{jk}$ is a full matrix. In many practical implementations, the covariance matrix $\boldsymbol{\Sigma}_{jk}$ is assumed diagonal. This does not imply feature independence.

A.1. Estimation of Model Parameters - The Training Problem

Given a training database that contains H sequences, (of duration: T_h ; h = 1,2,…,H), it is required to estimate the model's parameters:

$$\hat{\lambda} = \{\hat{\boldsymbol{\pi}}, \hat{\mathbf{A}}, \hat{\mathbf{c}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\}$$

For training the model, we use some probability variables:

(1) Forward variable: $\alpha_t(i)$; $\alpha_t(i) = p(\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_t, q_t = i \mid \lambda)$, and

(2) Backward variable: $\beta_t(i)$; $\beta_t(i) = p(\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, ..., \mathbf{o}_T \mid q_t = i, \lambda)$.

These forward and backward variables are calculated iteratively [Rabiner et al., 1993]:

$$\alpha_1(j) = \pi_j b_j(\mathbf{o}_1) ; \quad 1 \le j \le N$$

$$\alpha_t(i) = [\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}] b_i(\mathbf{o}_t) ; \quad 2 \le t \le T ; 1 \le i \le N \tag{A.3}$$

The training of the model is performed by the *Baum-Welch* algorithm, which is an iterative Expectation-Maximization (EM) algorithm [Dempster et al., 1977], [Moon, 1996]. With this algorithm, we start with an initial model, $\hat{\lambda}_0 = \{\hat{\boldsymbol{\pi}}_0, \hat{\mathbf{A}}_0, \hat{\mathbf{c}}_0, \hat{\boldsymbol{\mu}}_0, \hat{\boldsymbol{\Sigma}}_0\}$, usually using k-means algorithm [Rabiner et al., 1993]. In every iteration, the model is re-estimated, by, first calculating the forward and backward probabilities (for every sequence), and calculating variable $\gamma_t(i,k)$: $\gamma_t(i,k) = p(q_t = i, \text{ using the } k\text{-th component} \mid \mathbf{O}, \lambda)$,

$$\gamma_t(i,k) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \cdot \frac{c_{ik} N(\mathbf{o}_t, \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})}{\sum_{m=1}^{M} c_{im} N(\mathbf{o}_t, \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im})} \tag{A.4}$$

And then, re-estimating the model parameters: $\hat{\boldsymbol{\pi}}, \hat{\mathbf{A}}, \hat{\mathbf{c}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}$, by

$$\hat{a}_{ij} = \frac{\sum_{h=1}^{H}\sum_{t=1}^{T_h-1}\alpha_t^h(i)a_{ij}b_j\left(\mathbf{o}_{t+1}^h\right)\beta_{t+1}^h(j)}{\sum_{h=1}^{H}\sum_{t=1}^{T_h-1}\alpha_t^h(i)\beta_t^h(i)\frac{1}{C_t}} \tag{A.5}$$

where $C_t = \frac{1}{\sum_{i=1}^{N}\alpha_t(i)}$

$$\hat{c}_{ik} = \frac{\sum_{h=1}^{H}\sum_{t=1}^{T_h}\gamma_t^h(i,k)}{\sum_{h=1}^{H}\sum_{k=1}^{M}\sum_{t=1}^{T_h}\gamma_t^h(i,k)} \tag{A.6}$$

$$\hat{\boldsymbol{\mu}}_{ik} = \frac{\sum_{h=1}^{H}\sum_{t=1}^{T_h}\gamma_t^h(i,k)\,\mathbf{o}_t^h}{\sum_{h=1}^{H}\sum_{t=1}^{T_h}\gamma_t^h(i,k)} \tag{A.7}$$

$$\hat{\boldsymbol{\Sigma}}_{ik} = \frac{\sum_{h=1}^{H}\sum_{t=1}^{T_h}\gamma_t^h(i,k)(\mathbf{o}_t^h-\boldsymbol{\mu}_{ik})(\mathbf{o}_t^h-\boldsymbol{\mu}_{ik})^{\mathrm{T}}}{\sum_{h=1}^{H}\sum_{t=1}^{T_h}\gamma_t^h(i,k)} \tag{A.8}$$

Note that in left-to-right HMM the initial vector is set to $\boldsymbol{\pi} = [1,0,0,\ldots,0]^T$ and is not estimated.

A.2. The Identification Problem - Output Probability

The identification algorithm is very simple as compared with the training process. Given a model $\lambda = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{c}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, and an observation sequence, $\mathbf{O}$, we want to calculate the probability of having the sequence from the model: $p(\mathbf{O}\,|\,\lambda)$. For this, the forward variables are calculated using (A.3). Then, the probability $p(\mathbf{O}\,|\,\lambda)$ is the calculation:

$$p(\mathbf{O}\,|\,\lambda) = \sum_{i=1}^{N}\alpha_T(i) \tag{A.9}$$

There is a third HMM problem – the "optimal" state sequence, which is not discussed here.

There are some implementation issues for HMMs. Scaling of the forward and the backward variables is essential in order to keep their values within the dynamic range of the computer [Rabiner et al., 1993]. We have noticed that even doing so, the forward variable is sometimes lower than the smallest positive floating-point value of the computer (underflow). Therefore, we have limited the forward variable to the minimum value of 10^{-200} (in double-float precision). Diagonal covariance matrices have been used, limited in their minimum value to 0.001, to avoid singularity.

Appendix B. Gaussian Mixture Model (GMM)

GMM can be viewed as one component (state) of (continuous-density) HMM. Since the GMM is always at the (only) given state, it is no longer "hidden."

A Gaussian mixture density is a weighted sum of M component densities, as given by the following [Reynolds and Rose, 1995]

$$p(\mathbf{o}\,|\,\lambda)=\sum_{i=1}^{M}c_i b_i(\mathbf{o}) \tag{B.1}$$

where $\mathbf{o}$ is a D-dimensional random observation vector, $b_i(\mathbf{o})$, $i=1,...,M$ are the component densities, and c_i, $i=1,...,M$ are the mixture weights. Each component density is a D-variate Gaussian function of the form

$$b_i(\mathbf{o})=\frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_i|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{o}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{o}-\boldsymbol{\mu}_i)\right\} \tag{B.2}$$

with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. The mixture weights have to satisfy the constraint $\sum_{i=1}^{M}c_i=1$. The complete Gaussian mixture density is parameterized by the mean vector, the covariance matrix and the mixture weight from all component densities. These parameters are collectively represented by

$$\lambda=\{c_i,\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i\};\ i=1,...,M \tag{B.3}$$

As in the general case of HMM, the covariance matrices $\boldsymbol{\Sigma}_i$ are often restricted to be diagonal.

In the training process, the maximum likelihood (ML) procedure is adopted to estimate model parameters, by maximizing the likelihood of GMM given the training data. For a

sequence of T training vectors $\mathbf{O}=\{\mathbf{o}_1,...,\mathbf{o}_T\}$, the GMM likelihood can be written as (assuming observations independence)

$$p(\mathbf{O}\mid\lambda)=\prod_{t=1}^{T}p(\mathbf{o}_t\mid\lambda) \tag{B.4}$$

The ML parameter estimates are obtained iteratively using the expectation-maximization (EM) algorithm. At each iteration, the parameter update formulas are as below, which guarantee a monotonic increase in the likelihood value.

Mixture weight update:

$$\hat{c}_i=\frac{1}{T}\sum_{t=1}^{T}p(i\mid\mathbf{o}_t,\lambda) \tag{B.5}$$

Mean vector update:

$$\hat{\boldsymbol{\mu}}_i=\frac{\sum_{t=1}^{T}p(i\mid\mathbf{o}_t,\lambda)\mathbf{o}_t}{\sum_{t=1}^{T}p(i\mid\mathbf{o}_t,\lambda)} \tag{B.6}$$

Covariance matrix update:

$$\hat{\boldsymbol{\Sigma}}_i=\frac{\sum_{t=1}^{T}p(i\mid\mathbf{o}_t,\lambda)(\mathbf{o}_t-\hat{\boldsymbol{\mu}}_i)(\mathbf{o}_t-\hat{\boldsymbol{\mu}}_i)^T}{\sum_{t=1}^{T}p(i\mid\mathbf{o}_t,\lambda)} \tag{B.7}$$

The a posteriori for the ith mixture is given by

$$p(i\mid\mathbf{o}_t,\lambda)=\frac{c_i b_i(\mathbf{o}_t)}{\sum_{k=1}^{M}c_k b_k(\mathbf{o}_t)} \tag{B.8}$$