Appendix A
MFCC Features

The MFCC feature extraction technique basically includes windowing the signal, applying the DFT, taking the log of the magnitude, and then warping the frequencies on a Mel scale, followed by applying the inverse DCT. The detailed description of various steps involved in the MFCC feature extraction is explained below.

1. **Pre-emphasis**: Pre-emphasis refers to filtering that emphasizes the higher frequencies. Its purpose is to balance the spectrum of voiced sounds that have a steep roll-off in the high-frequency region. For voiced sounds, the glottal source has an approximately -12 dB/octave slope [1]. However, when the acoustic energy radiates from the lips, this causes a roughly $+6$ dB/octave boost to the spectrum. As a result, a speech signal when recorded with a microphone from a distance has approximately a -6 dB/octave slope downward compared to the true spectrum of the vocal tract. Therefore, pre-emphasis removes some of the glottal effects from the vocal tract parameters. The most commonly used pre-emphasis filter is given by the following transfer function

$$H(z) = 1 - bz^{-1} \tag{A.1}$$

   where the value of b controls the slope of the filter and is usually between 0.4 and 1.0 [1].
2. **Frame blocking and windowing**: The speech signal is a slowly time-varying or quasi-stationary signal. For stable acoustic characteristics, speech needs to be examined over a sufficiently short period of time. Therefore, speech analysis must always be carried out on short segments across which the speech signal is assumed to be stationary. Short-term spectral measurements are typically carried out over 20 ms windows, and advanced every 10 ms [2, 3]. Advancing the time window every 10 ms enables the temporal characteristics of individual speech sounds to be tracked, and the 20 ms analysis window is usually sufficient to provide good spectral resolution of these sounds, and at the same time short enough to resolve significant temporal characteristics. The purpose of the overlapping analysis is that each speech sound of the input sequence would be approximately centered

K.S. Rao and Manjunath K.E., *Speech Recognition Using Articulatory and Excitation Source Features*, SpringerBriefs in Speech Technology,
DOI 10.1007/978-3-319-49220-9

at some frame. On each frame, a window is applied to taper the signal towards the frame boundaries. Generally, Hanning or Hamming windows are used [1]. This is done to enhance the harmonics, smooth the edges, and to reduce the edge effect while taking the DFT on the signal.

3. **DFT spectrum**: Each windowed frame is converted into magnitude spectrum by applying DFT.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}}; \qquad 0 \leq k \leq N-1 \tag{A.2}$$

where N is the number of points used to compute the DFT.

4. **Mel spectrum**: Mel spectrum is computed by passing the Fourier transformed signal through a set of band-pass filters known as Mel-filter bank. A Mel is a unit of measure based on the human ears perceived frequency. It does not correspond linearly to the physical frequency of the tone, as the human auditory system apparently does not perceive pitch linearly. The Mel scale is approximately a linear frequency spacing below 1 kHz and a logarithmic spacing above 1 kHz [4]. The approximation of Mel from physical frequency can be expressed as

$$f_{Mel} = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{A.3}$$

where f denotes the physical frequency in Hz, and f_{Mel} denotes the perceived frequency [2].

Filter banks can be implemented in both time domain and frequency domain. For MFCC computation, filter banks are generally implemented in frequency domain. The center frequencies of the filters are normally evenly spaced on the frequency axis. However, in order to mimic the human ears perception, the warped axis, according to the nonlinear function given in Eq. (A.3), is implemented. The most commonly used filter shaper is triangular, and in some cases the Hanning filter can be found [1]. The triangular filter banks with Mel frequency warping is given in Fig. A.1.

The Mel spectrum of the magnitude spectrum $X(k)$ is computed by multiplying the magnitude spectrum by each of the of the triangular Mel weighting filters.

$$s(m) = \sum_{k=0}^{N-1} \left[|X(k)|^2 H_m(k) \right]; \qquad 0 \leq m \leq M-1 \tag{A.4}$$

where M is total number of triangular Mel weighting filters [5, 6]. $H_m(k)$ is the weight given to the k^{th} energy spectrum bin contributing to the m^{th} output band and is expressed as:

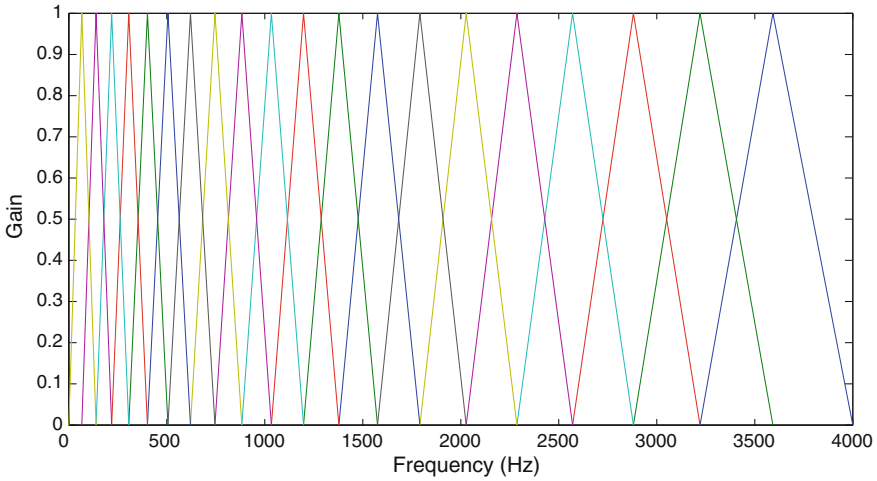

Fig. A.1 Mel-filter bank

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \frac{2(k-f(m-1))}{f(m)-f(m-1)}, & f(m-1) \le k \le f(m) \\ \frac{2(f(m+1)-k)}{f(m+1)-f(m)}, & f(m) < k \le f(m+1) \\ 0, & k > f(m+1) \end{cases} \tag{A.5}$$

with m ranging from 0 to $M-1$.

5. **Discrete cosine transform (DCT)**: Since the vocal tract is smooth, the energy levels in adjacent bands tend to be correlated. The DCT is applied to the transformed Mel frequency coefficients produces a set of cepstral coefficients. Prior to computing DCT, the Mel spectrum is usually represented on a log scale. This results in a signal in the cepstral domain with a quefrequency peak corresponding to the pitch of the signal and a number of formants representing low quefrequency peaks. Since most of the signal information is represented by the first few MFCC coefficients, the system can be made robust by extracting only those coefficients ignoring or truncating higher order DCT components [1]. Finally, MFCC is calculated as [1]

$$c(n) = \sum_{m=0}^{M-1} \log_{10}(s(m)) \cos\left(\frac{\pi n(m-0.5)}{M}\right); \qquad n = 0, 1, 2, \ldots, C-1 \tag{A.6}$$

where $c(n)$ are the cepstral coefficients, and C is the number of MFCCs. Traditional MFCC systems use only 8–13 cepstral coefficients. The zeroth coefficient is often excluded since it represents the average log-energy of the input signal, which only carries little speaker-specific information.

6. **Dynamic MFCC features**: The cepstral coefficients are usually referred to as static features, since they only contain information from a given frame. The extra information about the temporal dynamics of the signal is obtained by computing first and second derivatives of cepstral coefficients [7–9]. The first-order derivative is called delta coefficients, and the second-order derivative is called delta–delta coefficients. Delta coefficients tell about the speech rate, and delta–delta coefficients provide information similar to acceleration of speech. The commonly used definition for computing dynamic parameter is [7]

$$\Delta c_m(n) = \frac{\sum_{i=-T}^{T} k_i c_m(n+i)}{\sum_{i=-T}^{T} |i|} \tag{A.7}$$

where $c_m(n)$ denotes the m^{th} feature for the n^{th} time frame, k_i is the i^{th} weight, and T is the number of successive frames used for computation. Generally T is taken as 2. The delta–delta coefficients are computed by taking the first-order derivative of the delta coefficients.

References

1. J.W. Picone, Signal modeling techniques in speech recognition. Proc. IEEE **81**, 1215–1247 (1993)
2. J.R. Deller, J.H. Hansen, J.G. Proakis, *Discrete Time Processing of Speech Signals* (Prentice Hall, NJ, 1993)
3. J. Benesty, M.M. Sondhi, Y.A. Huang, *Handbook of Speech Processing* (Springer, New York, 2008)
4. J. Volkmann, S. Stevens, E. Newman, A scale for the measurement of the psychological magnitude pitch. J. Acoust. Soc. Am. **8**, 185–190 (1937)
5. Z. Fang, Z. Guoliang, S. Zhanjiang, Comparison of different implementations of MFCC. J. Comput. Sci. Technol. **16**, 582–589 (2000)
6. G.K.T. Ganchev, N. Fakotakis, Comparative evaluation of various MFCC implementations on the speaker verification task, in *Proceedings of International Conference on Speech and Computer (SPECOM)* (2005), pp. 191–194
7. L. Rabiner, B.-H. Juang, B. Yegnanarayana, *Fundamentals of Speech Recognition* (Pearson Education, London, 2008)
8. S. Furui, Comparison of speaker recognition methods using statistical features and dynamic features. IEEE Trans. Acoust. Speech Sig. Proc. **29**, 342–350 (1981)
9. J.S. Mason, X. Zhang, Velocity and acceleration features in speaker recognition, in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (1991), pp. 3673–3676

Appendix B
Pattern Recognition Models

In this work, hidden Markov model (HMM), support vector machine (SVM), and auto-associative neural network (AANN) models are used to capture the pattern present in features. HMMs are used to capture the sequential information present in feature vectors for CV recognition. SVMs are used to capture the discriminative information present in the feature vectors for CV recognition. AANN models are used to capture the nonlinear relations among the feature vectors for speaker identification. The following sections briefly describe the pattern recognition models used in this study.

B.1 Hidden Markov Models

Hidden Markov models (HMMs) are the commonly used classification models in speech recognition [1]. HMMs are used to capture the sequential information present in feature vectors for developing PRSs. HMM is a stochastic signal model which is referred to as Markov sources or probabilistic functions of Markov chains. This model is an extension to the concept of Markov model which includes the case where the observation is a probabilistic function of the state. HMM is a finite set of states, each of which is associated with a probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state, an outcome or observation can be generated, according to the associated probability distribution. Here, only the outcome is known and the underlying state sequence is hidden. Hence, it is called a hidden Markov model.

Following are the basic elements that define HMM:

1. N, Number of states in the model,
   $s = \{s_1, s_2, ...s_N\}$
2. M, Number of distinct observation symbol per state,
   $v = \{v_1, v_2, ...v_M\}$
3. State transition probability distribution $A = \{a_{ij}\}$, where

K.S. Rao and Manjunath K.E., *Speech Recognition Using Articulatory and Excitation Source Features*, SpringerBriefs in Speech Technology,
DOI 10.1007/978-3-319-49220-9

$$a_{ij} = P\left[q_{t+1} = s_j | q_t = s_i\right], 1 \leq i, j \leq N \tag{B.1}$$

4. Observation symbol probability distribution in state j,
   B = $\{b_j(k)\}$, where

$$b_j(k) = P\left[v_k \; at \; t | q_t = s_j\right] \qquad 1 \leq j \leq N, 1 \leq k \leq M \tag{B.2}$$

5. Initial state distribution $\Pi = \{\Pi_j\}$, where

$$\Pi_j = P\left[q_1 = s_i\right] \qquad 1 \leq i \leq N \tag{B.3}$$

So, a complete specification of an HMM requires specification of two model parameters (N and M), specification of observation symbols, and the specification of three probability measures A, B, Π. Therefore, HMM is indicated by the compact notation

$$\lambda = (A, B, \Pi)$$

Given that state sequence $q = (q1q2...q_T)$ is unknown, the probability of observation sequence $O = (o1o2...o_T)$, given the model λ, is obtained by summing the probability of over all possible state sequences q as follows:

$$P(o|\lambda) = \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) ... a_{q_{T-1} q_T} b_{q_T}(o_T) \tag{B.4}$$

where π_{q_1} is the initial state probability of q_1, and T is length of observation sequence.

B.2 FeedForward Neural Networks

FFNNs are the artificial neural networks, where the information moves from the input layer to output layer through the hidden layer in forward direction with no loops in the network. FFNNs are used to capture the nonlinear relationship between the feature vectors and the phonetic sound units. FFNNs map an input feature vector into one of the phonetic units, among the set of phonetic sound units used for training the FFNN models. Each unit in one layer of the FFNN has directed connections to the units in the subsequent layer. FFNNs consist of an input layer, an output layer, and one or more hidden layers. The number of units in the input is equal to the dimension of feature vectors, while the number of units in output layer is equal to the number of phonetic sound units being modeled. The hidden and output layers are nonlinear, whereas the input layer is linear. The nonlinearity is achieved using activation functions such as sigmoid, softmax. The general structure of three-layered FFNN is as shown in Fig. B.1. A three-layered FFNN has one input layer, one hidden layer, and one output layer.

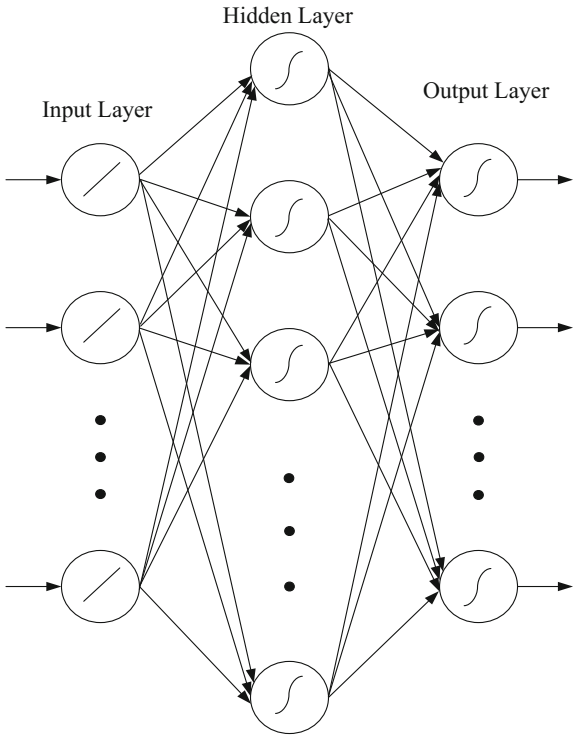

Fig. B.1 General structure of three-layered FeedForward neural networks

The feature vectors are fed to the input layer, and the corresponding phone labels are fed to the output layer of the FFNN. FFNNs are trained using a learning algorithm such as back-propagation algorithm [2, 3]. The back-propagation algorithm is most commonly used in the development of speech recognition applications using FFNNs. In back-propagation algorithm, the calculated output is compared with the correct output, and the error between them is computed using a predefined error function. The error is then back-propagated through the network, and the weights of the network are adjusted based on the computed error. The weights are adjusted using a nonlinear optimization method such as gradient descent method. This process is repeated for sufficiently large number of training examples till the network converges. After the completion of training phase, the weights of the network are used for decoding the phonetic sound units in the spoken utterances. Determining the network structure is an optimization problem. At present, there are no formal methods for determining the optimal structure of a neural network. The key factors that influence the neural network structure are amount of training data, learning ability of the network, and capacity to generalize the acquired knowledge.

References

1. L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**, 257–286 (1989)
2. R. Rojas, *Neural Networks - A Systematic Introduction* (Springer, Berlin, 1996)
3. M. Nielsen, Neural Networks and Deep Learning. http://neuralnetworksanddeeplearning.com.