Appendix A
LPCC Features

The cepstral coefficients derived from either linear prediction (LP) analysis or a filter bank approach are almost treated as standard front end features [1, 2]. Speech systems developed based on these features have achieved a very high level of accuracy, for speech recorded in a clean environment. Basically, spectral features represent phonetic information, as they are derived directly from spectra. The features extracted from spectra, using the energy values of linearly arranged filter banks, equally emphasize the contribution of all frequency components of a speech signal. In this context, LPCCs are used to capture emotion-specific information manifested through vocal tract features. In this work, the 10th order LP analysis has been performed, on the speech signal, to obtain 13 LPCCs per speech frame of 20 ms using a frame shift of 10 ms. The human way of emotion recognition depends equally on two factors, namely: its expression by the speaker as well as its perception by a listener. The purpose of using LPCCs is to consider vocal tract characteristics of the speaker, while performing automatic emotion recognition.

Cepstrum may be obtained using linear prediction analysis of a speech signal. The basic idea behind linear predictive analysis is that the nth speech sample can be estimated by a linear combination of its previous p samples as shown in the following equation.

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + a_3 s(n-3) + \cdots + a_p s(n-p)$$

where a_1, a_2, a_3, ... are assumed to be constants over a speech analysis frame. These are known as predictor coefficients or linear predictive coefficients. These coefficients are used to predict the speech samples. The difference of actual and predicted speech samples is known as an error. It is given by

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k)$$

K.S. Rao et al., *Language Identification Using Spectral and Prosodic Features*,
SpringerBriefs in Speech Technology, DOI 10.1007/978-3-319-17163-0

where $e(n)$ is the error in prediction, $s(n)$ is the original speech signal, $\hat{s}(n)$ is a predicted speech signal, $a_k s$ are the predictor coefficients.

To compute a unique set of predictor coefficients, the sum of squared differences between the actual and predicted speech samples has been minimized (error minimization) as shown in the equation below

$$E_n = \sum_m \left[s_n(m) - \sum_{k=1}^{p} a_k s_n(m-k) \right]^2$$

where m is the number of samples in an analysis frame. To solve the above equation for LP coefficients, E_n has to be differentiated with respect to each a_k and the result is equated to zero as shown below

$$\frac{\partial E_n}{\partial a_k} = 0, \qquad \text{for } k = 1, 2, 3, \ldots, p$$

After finding the $a_k s$, one may find cepstral coefficients using the following recursion.

$$C_0 = \log_e p$$

$$C_m = a_m + \sum_{k=1}^{m-1} \frac{k}{m} C_k a_{m-k}, \qquad \text{for } 1 < m < p \text{ and}$$

$$C_m = \sum_{k=m-p}^{m-1} \frac{k}{m} C_k a_{m-k}, \qquad \text{for } m > p$$

References

1. Makhoul J (1975) Linear prediction: a tutorial review. In: Proceedings of IEEE, vol 63, pp 561–580
2. Rabiner L, Juang B (1993) Fundamentals of speech recognition. Prentice Hall, New Jersey

Appendix B
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

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

   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 et al., *Language Identification Using Spectral and Prosodic Features*,
SpringerBriefs in Speech Technology, DOI 10.1007/978-3-319-17163-0

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.

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

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

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

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 non-linear function given in Eq. (B.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. B.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.

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

where M is total number of triangular mel weighting filters [5, 6]. $H_m(k)$ is the weight given to the kth energy spectrum bin contributing to the mth output band and is expressed as:

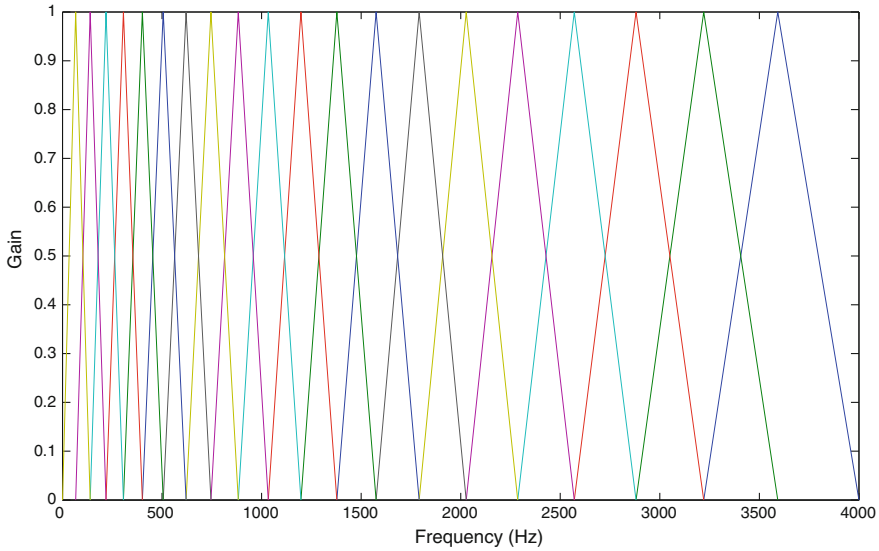

Fig. B.1 Mel-filter bank

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

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 que-frequency 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]

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

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, 8]. 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

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

where $c_m(n)$ denotes the mth feature for the nth time frame, k_i is the ith 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. Picone JW (1993) Signal modeling techniques in speech recognition. In: Proceedings of IEEE, vol 81, pp 1215–1247, Sept 1993
2. Deller JR, Hansen JH, a202 JG (1993) Discrete time processing of speech signals, 1st edn. Prentice Hall PTR, Upper Saddle River
3. Benesty J, Sondhi MM, Huang YA (2008) Springer handbook of speech processing. Springer, New York
4. Volkmann J, Stevens S, Newman E (1937) A scale for the measurement of the psychological magnitude pitch. J Acoust Soc Am 8:185–190
5. Fang Z, Guoliang Z, Zhanjiang S (2001) Comparison of different implementations of MFCC. J Comput Sci Technol 16(6):582–589
6. Ganchev GKT, Fakotakis N (2005) Comparative evaluation of various MFCC implementations on the speaker verification task. In: Proceedings of international conference on speech and computer, Patras, Greece, pp 191–194
7. Furui S (1981) Comparison of speaker recognition methods using statistical features and dynamic features. IEEE Trans Acoust Speech Signal Process 29(3):342–350
8. Mason JS, Zhang X (1991) Velocity and acceleration features in speaker recognition. In: Proceedings IEEE international conference acoustics speech signal processing, Toronto, Canada, pp 3673–3676, Apr 1991

Appendix C
Gaussian Mixture Model (GMM)

In the speech and speaker recognition the acoustic events are usually modeled by Gaussian probability density functions (PDFs), described by the mean vector and the covariance matrix. However unimodel PDF with only one mean and covariance are unsuitable to model all variations of a single event in speech signals. Therefore, a mixture of single densities is used to model the complex structure of the density probability. For a D-dimensional feature vector denoted as x_t, the mixture density for speaker Ω is defined as weighted sum of M component Gaussian densities as given by the following [1]

$$P(x_t|\Omega) = \sum_{i=1}^{M} w_i P_i(x_t) \tag{C.1}$$

where w_i are the weights and $P_i(x_t)$ are the component densities. Each component density is a D-variate Gaussian function of the form

$$P_i(x_t) = \frac{1}{(2\pi)^{D/2}\,|\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}\left[(x_t-\mu_i)'\,\Sigma_i^{-1}(x_t-\mu_i)\right]} \tag{C.2}$$

where μ_i is a mean vector and Σ_i covariance matrix for ith component. The mixture weights have to satisfy the constraint [1]

$$\sum_{i=1}^{M} w_i = 1. \tag{C.3}$$

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

$$\Omega = \{w_i, \mu_i, \Sigma_i\}; \qquad i = 1, 2, ...M. \tag{C.4}$$

K.S. Rao et al., *Language Identification Using Spectral and Prosodic Features*,
SpringerBriefs in Speech Technology, DOI 10.1007/978-3-319-17163-0

C.1 Training the GMMs

To determine the model parameters of GMM of the speaker, the GMM has to be trained. In the training process, the maximum likelihood (ML) procedure is adopted to estimate model parameters. For a sequence of training vectors $X = \{x_1, x_2, .., x_T\}$, the GMM likelihood can be written as (assuming observations independence) [1]

$$P(X|\Omega) = \prod_{t=1}^{T} P(x_t|\Omega). \tag{C.5}$$

Usually this is done by taking the logarithm and is commonly named as log-likelihood function. From Eqs. (C.1) and (C.5), the log-likelihood function can be written as

$$\log\left[P(X|\Omega)\right] = \sum_{t=1}^{T} \log\left[\sum_{i=1}^{M} w_i P_i(x_t)\right]. \tag{C.6}$$

Often, the average log-likelihood is used value is used by dividing $\log\left[P(X|\Omega)\right]$ by T. This is done to normalize out duration effects from the log-likelihood value. Also, since the incorrect assumption of independence is underestimating the actual likelihood value with dependencies, scaling by T can be considered a rough compensation factor [2]. The parameters of a GMM model can be estimated using maximum likelihood (ML) estimation. The main objective of the ML estimation is to derive the optimum model parameters that can maximize the likelihood of GMM. The likelihood value is, however, a highly nonlinear function in the model parameters and direct maximization is not possible. Instead, maximization is done through iterative procedures. Of the many techniques developed to maximize the likelihood value, the most popular is the iterative expectation maximization (EM) algorithm [3].

C.1.1 Expectation Maximization (EM) Algorithm

The EM algorithm begins with an initial model Ω and tends to estimate a new model such that the likelihood of the model increasing with each iteration. This new model is considered to be an initial model in the next iteration and the entire process is repeated until a certain convergence threshold is obtained or a certain predetermined number of iterations have been made. A summary of the various steps followed in the EM algorithm are described below.

1. **Initialization**: In this step an initial estimate of the parameters is obtained. The performance of the EM algorithm depends on this initialization. Generally, LBG [4] or K-means algorithm [5, 6] is used to initialize the GMM parameters.

2. **Likelihood computation**: In each iteration the posterior probabilities for the ith mixture is computed as [1]:

$$\Pr(i|x_t) = \frac{w_i P_i(x_t)}{\sum_{j=1}^{M} w_j P_j(x_t)}. \tag{C.7}$$

3. **Parameter update**: Having the posterior probabilities, the model parameters are updated according to the following expressions [1].
Mixture weight update:

$$\overline{w_i} = \frac{\sum_{i=1}^{T} \Pr(i|x_t)}{T}. \tag{C.8}$$

Mean vector update:

$$\overline{\mu_i} = \frac{\sum_{i=1}^{T} \Pr(i|x_t) x_t}{\sum_{i=1}^{T} \Pr(i|x_t)}. \tag{C.9}$$

Covariance matrix update:

$$\overline{\sigma}_i^2 = \frac{\sum_{i=1}^{T} \Pr(i|x_t)\, |x_t - \overline{\mu_i}|^2}{\sum_{i=1}^{T} \Pr(i|x_t)}. \tag{C.10}$$

In the estimation of the model parameters, it is possible to choose, either full covariance matrices or diagonal covariance matrices. It is more common to use diagonal covariance matrices for GMM, since linear combination of diagonal covariance Gaussians has the same model capability with full matrices [7]. Another reason is that speech utterances are usually parameterized with cepstral features. Cepstral features are more compactable, discriminative, and most important, they are nearly uncorrelated, which allows diagonal covariance to be used by the GMMs [1, 8]. The iterative process is normally carried out 10 times, at which point the model is assumed to converge to a local maximum [1].

C.1.2 Maximum a Posteriori (MAP) Adaptation

Gaussian mixture models for a speaker can be trained using the modeling described earlier. For this, it is necessary that sufficient training data is available in order to

create a model of the speaker. Another way of estimating a statistical model, which is especially useful when the training data available is of short duration, is by using maximum a posteriori adaptation (MAP) of a background model trained on the speech data of several other speakers [9]. This background model is a large GMM that is trained with a large amount of data which encompasses the different kinds of speech that may be encountered by the system during training. These different kinds may include different channel conditions, composition of speakers, acoustic conditions, etc. A summary of MAP adaptation steps are given below.

For each mixture i from the background model, $Pr(i|x_t)$ is calculated as [10]

$$\Pr(i|x_t) = \frac{w_i P_i(x_t)}{\sum_{j=1}^{M} w_j P_j(x_t)}. \tag{C.11}$$

Using $Pr(i|x_t)$, the statistics of the weight, mean and variance are calculated as follows [10]

$$n_i = \sum_{i=1}^{T} \Pr(i|x_t) \tag{C.12}$$

$$E_i(x_t) = \frac{\sum_{i=1}^{T} \Pr(i|x_t) x_t}{n_i} \tag{C.13}$$

$$E_i(x_t^2) = \frac{\sum_{i=1}^{T} \Pr(i|x_t) x_t^2}{n_i}. \tag{C.14}$$

These new statistics calculated from the training data are then used adapt the background model, and the new weights ($\hat{w}_i$), means ($\hat{\mu}_i$) and variances ($\hat{\sigma}_i^2$) are given by [10]

$$\hat{w}_i = \left[\frac{\alpha_i n_i}{T} + (1 - \alpha_i) w_i\right] \gamma \tag{C.15}$$

$$\hat{\mu}_i = \alpha_i E_i(x_t) + (1 - \alpha_i)\mu_i \tag{C.16}$$

$$\hat{\sigma}_i^2 = \alpha_i E_i(x_t^2) + (1 - \alpha_i)(\sigma_i^2 + \mu_i^2) - \hat{\mu}_i^2. \tag{C.17}$$

A scale factor γ is used, which ensures that all the new mixture weights sum to 1. α_i is the adaptation coefficient which controls the balance between the old and new model parameter estimates. α_i is defined as [10]

$$\alpha_i = \frac{n_i}{n_i + r} \tag{C.18}$$

where r is a fixed relevance factor, which determines the extent of mixing of the old and new estimates of the parameters. Low values for α_i ($\alpha_i \rightarrow 0$), will result in new parameter estimates from the data to be de-emphasized, while higher values ($\alpha_i \rightarrow 1$) will emphasize the use of the new training data-dependent parameters. Generally only mean values are adapted [2]. It is experimentally shown that mean adaptation gives slightly higher performance than adapting all three parameters [10].

C.2 Testing

In identification phase, mixture densities are calculated for every feature vector for all speakers and speaker with maximum likelihood is selected as identified speaker. For example, if S speaker models $\{\Omega_1, \Omega_2, ..., \Omega_S\}$ are available after the training, speaker identification can be done based on a new speech data set. First, the sequence of feature vectors $X = \{x_1, x_2, .., x_T\}$ is calculated. Then the speaker model $\hat{s}$ is determined which maximizes the a posteriori probability $P(\Omega_S|X)$. That is, according to the Bayes rule [1]

$$\hat{s} = \max_{1 \leq s \leq S} P(\Omega_S|X) = \max_{1 \leq s \leq S} \frac{P(X|\Omega_S)}{P(X)} P(\Omega_S). \tag{C.19}$$

Assuming equal probability of all speakers and the statistical independence of the observations, the decision rule for the most probable speaker can be redefined as

$$\hat{s} = \max_{1 \leq s \leq S} \sum_{t=1}^{T} \log P(x_t|\Omega_s) \tag{C.20}$$

with T the number of feature vectors of the speech data set under test and $P(x_t|\Omega_s)$ given by Eq. (C.1).

Decision in verification is obtained by comparing the score computed using the model for the claimed speaker Ω_S given by $P(\Omega_S|X)$ to a predefined threshold θ. The claim is accepted if $P(\Omega_S|X) > \theta$, and rejected otherwise [2].

References

1. Reynolds DA (1995) Speaker identification and verification using Gaussian mixture speaker models. Speech Commun 17:91–108
2. Bimbot F, Bonastre J, Fredouille C, Gravier G, Chagnolleau MI, Meignier S, Merlin T, Garcia OJ, Delacretaz P, Reynolds DA (1997) A tutorial on text-independent speaker verification. EURASIP J Appl Sig Proc 2004(4):430–451
3. Dempster AP, Laird NM, Rubin D (1997) Maximum likelihood from incomplete data via the EM algorithm. J Roy Stat Soc 39(1):1–38

4. Linde Y, Buzo A, Gray R (1980) An algorithm for vector quantizer design. IEEE Trans Commun 28:84–95
5. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Cam LML, Neyman J (eds) Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol 1. University of California Press, pp 281–297
6. Hartigan JA, Wong MA (1979) K-means clustering algorithm. Appl Stat 28(1): 100–108
7. Hong QY, Kwong S (2005) A discriminative training approach for text-independent speaker recognition. Signal Process 85(7):1449–1463
8. Reynolds D, Rose R (1995) Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Trans Speech Audio Process 3:72–83
9. Gauvain J, Lee C-H (1994) Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Trans Speech Audio Process 2:291–298
10. Reynolds DA (2000) Speaker verification using adapted Gaussian mixture models. Digital Signal Process 10:19–41