

Evaluation Of Unsupervised Models Using Minimal Pair ABX Measure

Y Satya Dheeraj

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology



भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department of Electrical Engineering

July 2015

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

Dheeraj

(Signature)

Y. Satya Dheeraj

(Y Satya Dheeraj)

ee13m1009

(Roll No.)

Approval Sheet

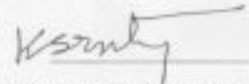
This Thesis entitled Evaluation Of Unsupervised Models Using Minimal Pair ABX Measure by Y Satya Dhara is approved for the degree of Master of Technology from IIT Hyderabad



(Dr. Ketan P. Detroja) Examiner
Dept. of Electrical Eng
IITH



(Dr. Sumohana S. Channappayya) Examiner
Dept. of Electrical Eng
IITH



(Dr. K Sri Rama Murty) Adviser
Dept. of Electrical Eng
IITH



(Dr. C Krishna Mohan) Chairman
Dept. of Computer Science
IITH

Acknowledgements

I would like to thank my guide Dr. K Sriram Murthy for his aspiring guidance and valuable suggestions and strong moral support. I would like to thank my labmates P Raghavedhra, Swathi J, Shekar Nayak and Karthika Vijayan for their valuable suggestions. I am also thankful to all my friends at IIT Hyderabad. Especially, Appina Balasubramanyam and B Naresh Reddy deserve a special mention for the enormous amount of help they have provided.

Abstract

The Minimal-Pair ABX (MP-ABX) task has been proposed as a method for evaluating speech features for zero resource (i.e only limited amount of labelled data) unsupervised speech technologies. MP-ABX task is an alternative to the phoneme word error rate, it is necessary to discriminate between the minimal pair of words from a language. We compared Mel Frequency Cepstral Coefficients (MFCC) with modelling parameters of these MFCC's by using unsupervised generative models like Gaussian Mixture Model (GMM) and Gaussian-Bernoulli Restricted Boltzmann Machine (GBRBM).

In an MP-ABX task, the features (MFCC) a, b and x associated to three speech sounds, A, B and X are computed, where A and B are chosen to be minimally different words (e.g. dog vs doll) and X is linguistically identical to either A or B, although it can be indexically different (different talker or added noise). Then, one determines whether x is closer to a or b by computing Distance Time Wrapping algorithm (DTW) of the evaluated features. By repeating this on a representative set of A, B, X triplets, a measure of the discriminability of minimal pairs when coded with the tested featural representation is obtained. This evaluation metric is especially suitable for zero-resource settings.

It is noticed that modelled MFCC's performed better in the case of PaT(Phoneme across Talker) and its performance decreases when the dimension of the modelled parameters increased. And in the case of PaC(Phoneme across Context), the performance of modelled parameters degraded. In case of PaC, RBM works better where as in PaT, GMM gives better results.

Contents

Declaration	ii
Approval Sheet	iii
Acknowledgements	iv
Abstract	v
Nomenclature	vii
1 Introduction	1
1.1 Introduction	1
1.2 MP-ABX Measure Discrimination Tasks	2
1.2.1 Phoneme across Context (PaC)	2
1.2.2 Phoneme across Talker (PaT)	2
1.2.3 Talker across Phoneme (TaP)	3
1.3 Model of MP-ABX tasks	3
1.3.1 DTW algorithm	3
2 Spectral Feature Extraction	6
2.1 Mel Frequency Cepstral Coefficients (MFCC)	6
2.1.1 Procedure	6
2.1.2 Fast Fourier Transform	7
2.1.3 Log-MEL Scale	8
2.1.4 Discrete Cosine Transform (DCT)	9
2.1.5 Noise Sensitivity	9
2.1.6 Applications	9
2.2 Instantaneous Frequency Cepstral Coefficients (IFCC)	9
2.2.1 Feature extraction from Instantaneous Frequency	10
3 Posterior Features Extraction using Unsupervised Models	13
3.1 Generative and Discriminative Models	13
3.1.1 Generative Models	13
3.1.2 Discriminative Models	14
3.2 Gaussian Mixture Models	14
3.2.1 EM Algorithm	14
3.2.2 Posterior extraction using GMM	15
3.3 Posterior Extraction using GBRBM	17

4 Results and Discussions	23
5 Conclusion	25
References	26

Chapter 1

Introduction

1.1 Introduction

Speech features are generally evaluated based on their outcome on an entire speech recognition system through phoneme error rates or word error rates. These metrics are precarious to unsupervised or Zero resource (i.e only limited amount of labelled data). First, it is very expensive because they need large amount of transcription speech data to be trained. Secondly, they lack intuition in that supervised training of the speech recognition system might compensate for future defects of the speech features (such as noisy or unreliable channels), even though such defects can be very harmful to zero-resource applications.

We proposed an alternative to phoneme error rates for evaluating speech features ,the error rate in a minimal pair ABX task (MP-ABX task). MP-ABX task venture the simple idea that in order to understand a language, it is necessary to discriminate between minimal-pairs of words from this language.

In an MP-ABX task,the features a, b and x accomlice to three speech sounds A, B and X are computed where A and B are choswen to be minimally different words (e.g,dog vs doll) and X is linguistically identical to either A or B ,although it can be indexically different(different talker or added noise). Then, one resolves whether x is closer to a or b according to a metric defined on the space of the evaluated features, and the result is compared to the expected answer. By repeating this on a representative set of A, B, X triplets, a measure of the discriminability of minimal pairs when coded with the tested featural representation is obtained. This evaluation metric is especially suitable for zero-resource settings as it doesnt unduly correct defects in the speech features and it

encapsulates all modelling assumptions in the choice of a metric on the space of the features, an object conceptually much simpler than a typical speech recognition pipeline.

1.2 MP-ABX Measure Discrimination Tasks

ABX tasks consist in presenting three stimuli A,B and X. A and B differ by some minimal contrast (differ only one phoneme) , and X is matched to either A or B. We use three variants of the task [1].

- i. Phoneme across Context (PaC)
- ii. Phoneme across Talker (PaT)
- iii. Talker across Phoneme (TaP)

1.2.1 Phoneme across Context (PaC)

- * In PaC ,A and B differ by only one phoneme either in a consonant or in a Vowel.
- * Both A and B are spoken by the same speaker.
- * X is also spoken by the same speaker ,matches to either A or B in one phoneme and differs from both in the other phoneme.
- * It measures context invariance in phoneme discrimination.

1.2.2 Phoneme across Talker (PaT)

- * In PaT, A and B differ by only one phoneme either in a consonant or in a Vowel.
- * Both A and B are spoken by the same speaker.
- * X is spoken by different speaker and matches to either A or B in both phonemes.
- * It measures talker discrimination in phoneme discrimination.

1.2.3 Talker across Phoneme (TaP)

- * A and B are spoken by two different talkers and are phonetically identical.
- * X is spoken by the same speaker as either A or B, but differs by them by one phoneme , enabling the measurement of talker discrimination.

Table 1.1: Example of a Possible choice of A, B, and X sounds for a single MP-ABX task. **ta** stands for talker.

Task	A	B	X	Answer
PaT	/ba/ ta1	/ga/ ta1	/ba/ ta2	A
PaT	/ba/ ta1	/ga/ ta1	/gu/ ta1	B
PaT	/ba/ ta1	/ga/ ta2	/ba/ ta1	A

1.3 Model of MP-ABX tasks

To perform these tasks on the speech representations a, b and x of the sounds A ,B and X (here we are using MFCCS and IFCCs) buy computing the Dynamic time wrapping (DTW) distances $d(a, x)$ and $d(b, x)$ between A ,X and B, X on the basis of an underlying frame based distance metrics.

Then the sign of $d(a, x)-d(b, x)$ is used to determine the response of the model (B or A for a positive and negative sign respectively) and error rate is compute for a representative set of stimuli. The choice of the underlying frame based metrics is important and may bounce the results. Here we use the cosine distances.

1.3.1 DTW algorithm

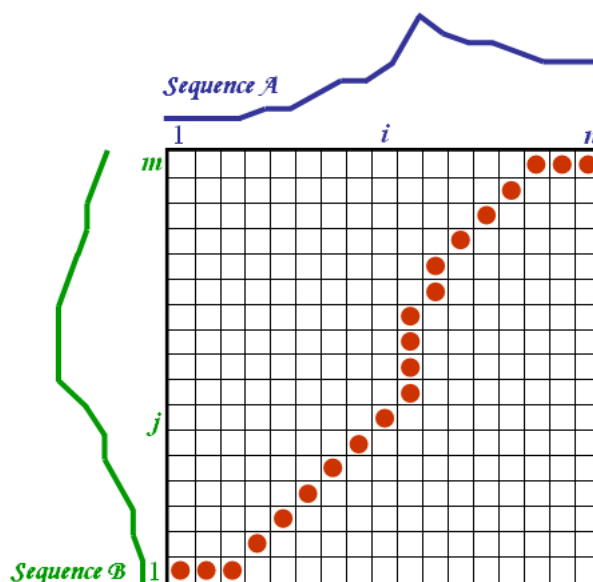
Dynamic time warping (DTW) is a time series alignment algorithm developed originally for speech recognition [2]. It aims at aligning two sequences of feature vectors by warping the time axis iteratively until an optimal match (according to a suitable metrics) between the two sequences is found.

Consider two sequences of feature vectors:

$$\mathbf{A} = a_1, a_2, a_3, \dots, a_n$$

$$\mathbf{B} = b_1, b_2, b_3, \dots, b_n$$

The two sequences can be arranged on the sides of a grid, with one on the top and the other up the left hand side. Both sequences start on the bottom left of the grid.



Dynamic time warping (DTW) is a time series alignment algorithm developed originally for speech recognition(1). It aims at aligning two sequences of feature vectors by warping the time axis iteratively until an optimal match (according to a suitable metrics) between the two sequences is found.

Consider two sequences of feature vectors:

The two sequences can be arranged on the sides of a grid, with one on the top and the other up the left hand side. Both sequences start on the bottom left of the grid.

Inside each cell a distance measure can be placed, comparing the corresponding elements of the two sequences. To find the best match or alignment between these two sequences one need to find a path through the grid which minimizes the total distance between them. The procedure for computing this overall distance involves finding all possible routes through the grid and for each one compute the overall distance. The overall distance is the minimum of the sum of the distances between the individual elements on the path divided by the sum of the weighting function. The weighting function is used to normalize for the path length. It is apparent that for any considerably long sequences the number of possible paths through the grid will be very large. The major optimisations or constraints of the DTW algorithm arise from the observations on the nature of acceptable paths through the grid:

Monotonic condition: the path will not turn back on itself, both the i and j indexes either stay the same or increase, they never decrease.

Continuity condition: the path advances one step at a time. Both i and j can only increase by at most 1 on each step along the path.

Boundary condition: the path starts at the bottom left and ends at the top right.

The foregoing constraints allow to restrict the moves that can be made from any point in the path and so limit the number of paths that need to be considered. The power of the DTW algorithm is in the fact that instead finding all possible routes through the grid which satisfy the above conditions, the DTW algorithm works by keeping track of the cost of the best path to each point in the grid. During the calculation process of the DTW grid it is not known which path is minimum overall distance path, but this can be traced back when the end point is reached.

Chapter 2

Spectral Feature Extraction

2.1 Mel Frequency Cepstral Coefficients (MFCC)

Mel frequency cepstral coefficient is very popular and efficient technique for speech signal processing. Basically MFCC [3] [4] is very common and one of the best methods for feature extraction and commonly used in speech recognition for speaker identification.

The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum.

2.1.1 Procedure

MFCCs are commonly derived as follows:

- * Take the fourier transform of a signal.
- * Raise the powers of the spectrum obtained above onto the mel scale , using triangular overlapping windows.
- * Take the logs of the powers at each of the mel frequencies.
- * Take the discrete cosine transform of the list of mel log powers to reduce the redundancy.
- * The MFCCs are the amplitudes of the resulting spectrum.

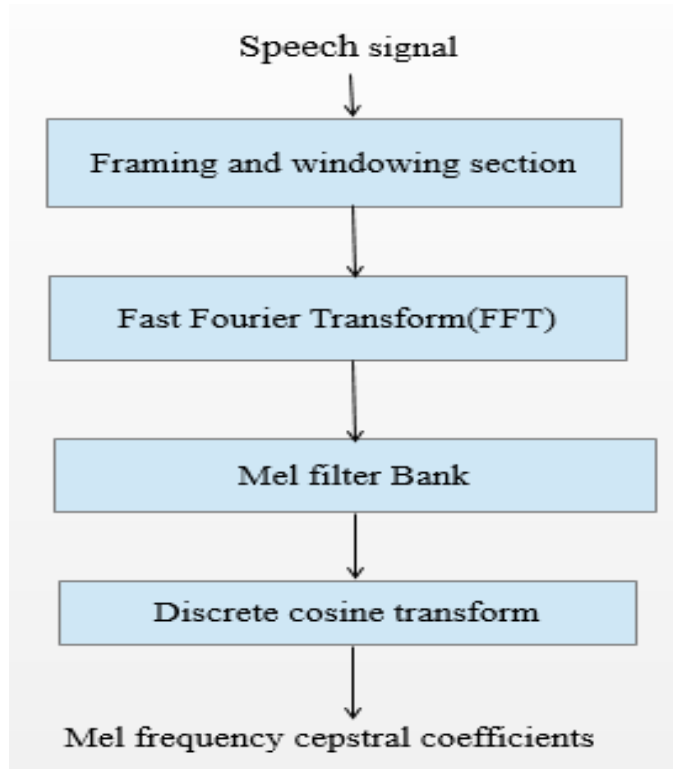


Figure 2.1: Block diagram of MFCC

In the figure 2.1 shows the block diagram of MFCC. The next following text contain the clear explanation of each block of MFCC.

2.1.2 Fast Fourier Transform

FFT is used to convert spatial(time)domain to the frequency domain. Each frame having N samples are converted into frequency domain. Fast Fourier transformation is a fast algorithm to apply Discrete Fourier Transform (DFT), on the given set of N samples shown below:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}; 0 \leq k \leq N - 1$$

$$W_N = e^{-\frac{j2\pi}{N}}$$

The above equations are representing the 1-D FFT equations.

2.1.3 Log-MEL Scale

In this step, the above calculated spectrums are mapped on Mel scale to know the approximation about the existing energy at each spot with the help of Triangular overlapping window also known as triangular filter bank. These filter bank is a set of band pass filters having spacing along with bandwidth decided by steady Mel frequency time [5]. Thus, Mel scale helps how to space the given filter and to calculate how much wider it should be because, as the frequency gets higher these filters are also get wider. For Mel- scaling mapping is need to done among the given real frequency scales (Hz) and the perceived frequency scale (Mels). During the mapping, when a given frequency value is up to 1000Hz the Mel-frequency scaling is linear frequency spacing, but after 1000Hz the spacing is logarithmic as shown in Figure . The formula to convert frequency f hertz into Mel m_f is given

$$m = 2595 \log_{10} \left(1 + \frac{f}{10} \right)$$

The variation of frequency (Hz) and Mel scale as shown in the below figure 2.2.

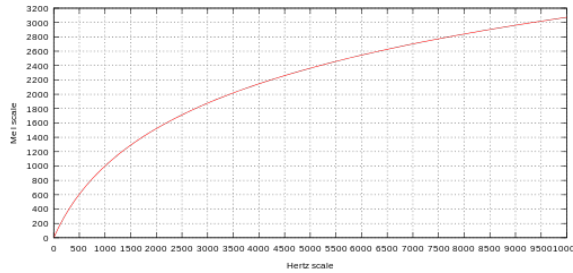


Figure 2.2: Plots of pitch mel scale versus Hertz scale.

Thus, with the help of Filter bank with proper spacing done by Mel scaling it becomes easy to get the estimation about the energies at each spot and once this energies are estimated then the log of these energies also known as Mel spectrum can be used for calculating first 13 coefficients using DCT. Since, the increasing numbers of coefficients represent faster change in the estimated energies and thus have less information to be used for classifying the given signals. Hence, first 13 coefficients are calculated using DCT and higher are discarded. To get better discrimination between the signals we can increase the dimensionality of mfcc by adding acceleration coefficients(delta,delta-delta).

2.1.4 Discrete Cosine Transform (DCT)

DCT is done in order to convert the log Mel spectrum back into the spatial domain. For this transformation we can use inverse DFT or DCT as they divide finite sequence data into discrete vector. But we consider DCT because it compresses data and reduces redundancy and has more information in a small number of coefficients so it is easy and requires less storage to represent the mel spectrum in a less number of coefficients. The output after applying DCT is called MFCC. The single variable DCT equation is shown below.

$$F(u) = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \Delta(i) \cos\left[\frac{\pi u}{2N}(2i+1)\right] * f(i)$$

2.1.5 Noise Sensitivity

MFCC values are not very robust in the presence of additive noise, and so it is common to normalize their values in speech recognition systems to reduce the influence of noise. Some modifications to the basic MFCC algorithm to improve robustness, such as by raising the Log-MEL-amplitudes to a suitable power (around 2 or 3) before taking the DCT, which reduces the influence of low-energy components.

2.1.6 Applications

- * MFCC are commonly used in speech recognition systems.
- * They are also common in speaker recognition which is a task of recognizing people from their voices.
- * MFCCs are also increasingly finding uses in music information retrieval applications such as genre classification, audio similarity measures, etc . . .

2.2 Instantaneous Frequency Cepstral Coefficients (IFCC)

Instantaneous Frequency Cepstral Coefficients (IFCC) are the cepstral coefficients extracted from the smoothed subband instantaneous frequency [6]. The performance of IFCC features are comparable with MFCC features in terms of equal error rates and minimum detection of cost function values. As we know most widely used features of speech like MFCC, LPC, FDLP deal with only magnitude information from speech signals. Hence phase information is also significant as the mag-

nitude. IFCC deals with the phase characteristics of speech signals.

Since computation of analytic phase suffers from phase wrapping problems, here we considered its derivative i.e. instantaneous frequency. Instantaneous frequency (IF) values computed over frames of speech are smoothed using many filters to nullify the sharp spurious peaks in it. DCT is applied on smoothed IF values to obtain stabilized cepstral features.

2.2.1 Feature extraction from Instantaneous Frequency

A continuous time signal $s(t)$ can be represented in the complex analytic domain

$$s_a(t) = s(t) + js_h(t)$$

where $s_h(t)$ is the Hilbert transform of the real signal $s(t)$ and is given by

$$s_h(t) = \mathcal{F}^{-1}\{S_h(j\Omega)\}$$

where \mathcal{F}^{-1} denotes inverse Fourier transform and $S_h(j\Omega)$ is given by

$$S_h(j\Omega) = \begin{cases} +S(j\Omega) & \Omega < 0 \\ -S(j\Omega) & \Omega > 0 \end{cases}$$

where $S(j\Omega)$ is the Fourier transform of $s(t)$. The analytic signal $s_a(t)$ contains only positive frequency components. It can be expressed in polar form as

$$s_a(t) = a(t)e^{j\phi(t)} \tag{2.1}$$

where $a(t)$ and $\phi(t)$ are the time-varying magnitude and phase of the analytic signal, respectively. If $s(t)$ is a narrow band signal, then $a(t)$ and $\phi(t)$ can be interpreted as amplitude modulated (AM) and frequency modulated (FM) components of $s(t)$. The computation of FM component $\phi(t)$, the analytic phase of $s(t)$, suffers from phase-wrapping problem and hence cannot be determined unambiguously. But its derivative with respect to time can be computed unambiguously, which is defined as instantaneous frequency (IF) and is given by

$$\phi'(t) = \frac{d\phi(t)}{dt}$$

Computation of IF does not require analytic phase. It can be computed by taking derivative on logarithm of the analytic signal, and is given by

$$\phi'(t) = \mathcal{I}\left\{\frac{s'_a(t)}{s_a(t)}\right\}$$

where $\mathcal{I}\{\cdot\}$ denotes imaginary part of a complex quantity and $s_a(t)$ is the time derivative of the analytic signal $s_a(t)$. The time derivative of the analytic signal can be computed using the Fourier transform relation as follows:

$$s'_a(t) = j\mathcal{F}^{-1}\{\Omega S_a(j\Omega)\}$$

where $S_a(j\Omega)$ is the Fourier transform of the analytic signal $s_a(t)$. IF can be interpreted as the frequency of a sinusoid which locally fits the signal. It has physical significance only when the signal is narrowband. Notice that a wideband signal cannot be approximated locally with a single sinusoid.

We can't directly compute IF from speech signal because speech is a wideband signal. We have to pass through narrowband filters in order to compute IFCC. IF is a representative of analytic phase of speech signals and hold information about formants. Speech signal is passed through L narrow band filters to get narrow band signals and then compute IF components. Since their is overlapping of filters, so there must be redundancy among IF coefficients. So DCT is applied and retain only first few DCT coefficients ($< L$). The low dimensional features obtained after applying DCT on IF coefficients are called IFCC.

The steps involved in feature extraction from IF is illustrated using a block diagram in Figure 2.3. The speech signal is passed through a bank of L filters centered around Ω_i , $i = 1, 2, \dots, L$, to obtain narrowband components $s_i(t)$. IF for each of these narrowband components ($\phi'_i(t)$) is computed, and is smoothed in order to remove impulse-like discontinuities at glottal closure instants. The center frequency of the narrowband signal is subtracted from its corresponding IF in order to bring it to zero mean.

The mean subtracted and smoothed IF are segmented into short frames to extract short-time IF features for speaker verification. The IF values within each frame are averaged over time to obtain

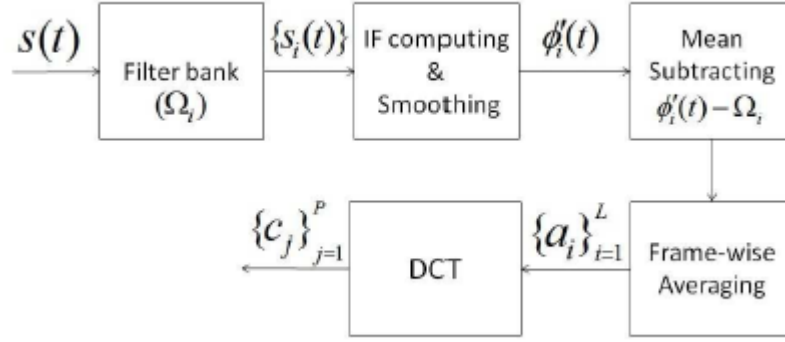


Figure 2.3: Block diagram of IFCC extraction from speech signal.

a set of L coefficients. Since the IF coefficients are obtained by employing overlapping filters in the frequency domain, they carry redundant information. The redundancy among IF coefficients is exploited to obtain a low-dimensional representation by employing discrete cosine transform (DCT) and retaining first few DCT coefficients ($j \leq L$). The low dimensional features obtained by applying DCT on IF coefficients, from here onwards is referred to as instantaneous frequency cepstral coefficients (IFCC). The IFCC along with their first and second order time derivatives are used for building a speaker verification system.

Chapter 3

Posterior Features Extraction using Unsupervised Models

Main objective of modelling techniques is to build a system which can discriminate different classes of inputs. In unsupervised approaches, this goal is accomplished without using labelled data. In the case of low resource languages, where less or no labelled data is available, unsupervised approaches can be a promising solution. In this study, generative models, namely GMM and GBRBM, are employed for unsupervised posterior feature extraction.

3.1 Generative and Discriminative Models

3.1.1 Generative Models

Generative models is a probabilistic model of all variables,where as discriminative models provides a model only for the target variables conditional on the observed variables.So, generative models can be used to generate values of any variable in the model.generative models perform better than discriminative variables at classification and regression tasks.

Examples:

- 1) Gaussian Mixture Models
- 2) Hidden Morkov Models
- 3)Restricted Boltzmann Machine

Generative models train a model of the joint probability $p(x, y)$, of the inputs x and the label y ,and find by using Bayes rule to calculate $P(y|x)$, then picking the most likely label y . Generative

models are typically more flexible than discriminative models in expressing dependencies in complex learning tasks.

3.1.2 Discriminative Models

Discriminative models are also called conditional models used in machine learning to get the dependence of a target variable Y on an observed variable X. Discriminative models allows only sampling of the target variables conditional on the observed quantities.

Examples:

- 1) Support Vector Machine
- 2) Linear Regression

3.2 Gaussian Mixture Models

A Gaussian mixture model is a generative model and probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters [7]. We estimate all the unknown parameters through expectation-maxmisation[EM] algorithm. Here unknown parameters are mean vector, variance vector and weight matrix. Initially we start with random values to those unknown parameters and compute log likelihood function and here we consider all data points are independent and identically distributed (*iid*) random variables.

3.2.1 EM Algorithm

We randomly initialize some values to unknown parameters like mean variance and weights of the Gaussian and we update these values through Expectation-Maximization[EM]Algorithm.

Expectation Step:

We initialize parameter values randomly and then try to maximize log likelihood function function to build a model in a maximization step.

$$r_{nk} = \frac{w_k \mathcal{N}(x_n / \mu_k, \sigma_k)}{\sum_{m=1}^M w_m \mathcal{N}(x_n / \mu_m, \sigma_m)}$$

$$N_k = \sum_{n=1}^N r_{nk}$$

Maximization Step

After expectation step, we maximize the log likelihood function by updating the parameters as shown below. After some iterations we compute log likelihood function $L(\theta)$, if it is under threshold or if it does not change then we will stop the procedure ,otherwise repeat these steps again.

$$\begin{aligned}\mu_k^{new} &= \frac{1}{N_k} \sum_{n=1}^N r_{nk} x_n \\ \sigma_k^{new} &= \frac{1}{N_k} \sum_{n=1}^N r_{nk} (x_n - \mu_k^{new})^2 \\ w_k^{new} &= \frac{N_k}{N}\end{aligned}$$

3.2.2 Posterior extraction using GMM

Mixture models capture the underlying statistical properties of data. In particular, GMM models the probability distribution of the data as a linear weighted combination of Gaussian densities. That is, given a data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, the probability of data X drawn from GMM is

$$p(X) = \sum_{i=1}^M w_i \mathcal{N}(X/\mu_i, \Sigma_i) \quad (3.1)$$

where $\mathcal{N}(\cdot)$ is Gaussian distribution, M is number of mixtures, w_i is the weight of the i^{th} Gaussian component, μ_i is its mean vector and Σ_i is its covariance matrix. The parameters of the GMM $\theta_i = \{w_i, \mu_i, \Sigma_i\}$ for $i = 1, 2, \dots, M$, can be estimated using Expectation Maximization (EM) algorithm [8]. Fig. 3.1 illustrates the joint density capturing capabilities of GMM, using 2-dimensional data uniformly disturbed along a circular ring. The red ellipses, superimposed on the data blue points, correspond to the locations and shapes of the estimated Gaussian mixtures.

In the case of 4-mixture GMM, with diagonal covariance matrices, the density was poorly estimated at odd multiples of 45° , as shown in Fig. 3.1(a). As the number of mixtures increases, the density is better captured as shown in Fig. 3.1(b). Since the diagonal matrices cannot capture correlations between dimensions, the curvature of the circular ring is not captured well.

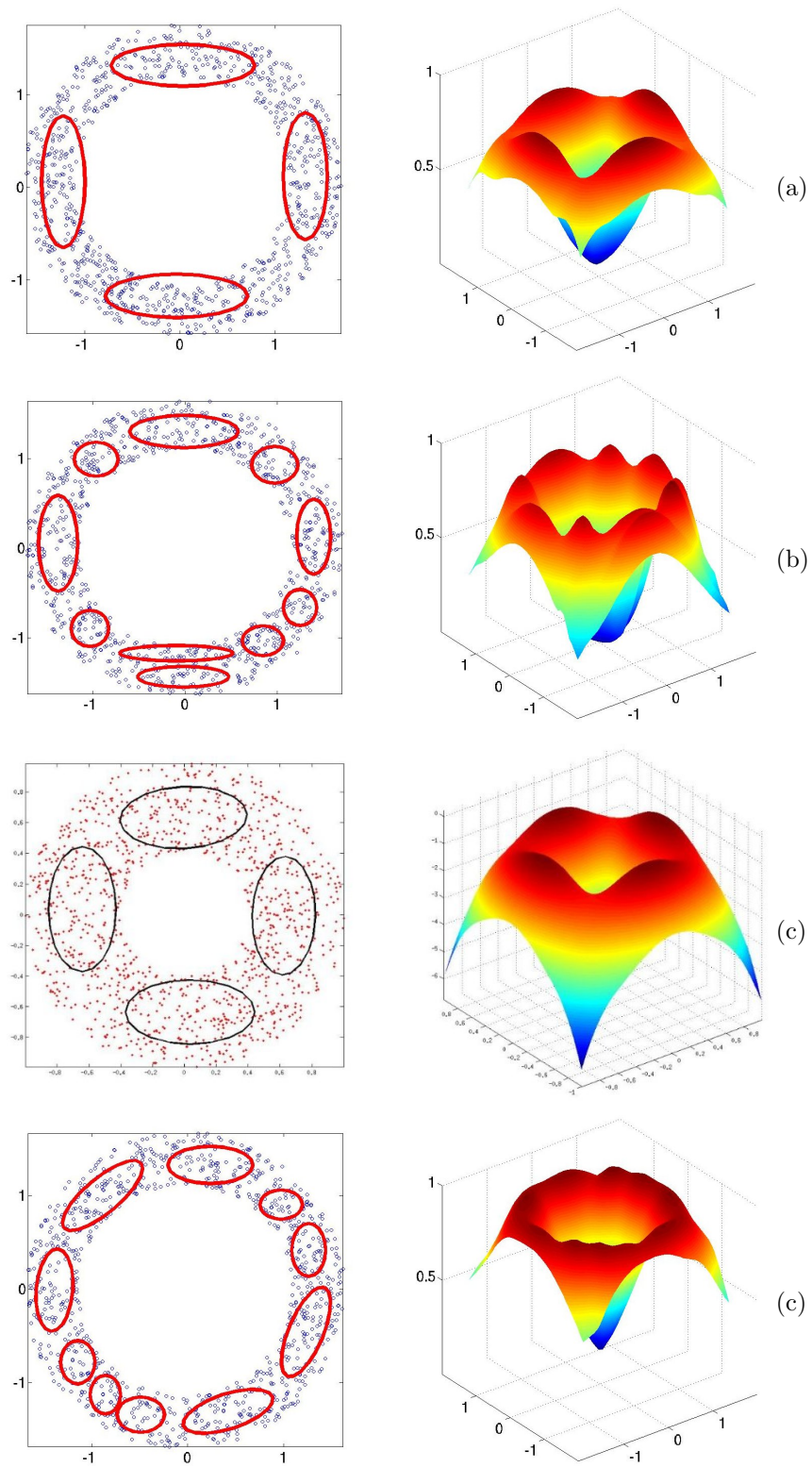


Figure 3.1: Illustration of distribution capturing capability of GMM. GMM trained with diagonal covariance matrices (a) 4-mixtures (b) 10-mixtures and (c) 10-mixture GMM trained with full covariance matrices

In the case of diagonal covariance matrices, the ellipses are aligned with the xy-axes as shown in Fig. 3.1(a) and Fig. 3.1(b). The density estimation can be improved using full covariance matrices, as shown in Fig. 3.1(c).

However, this improvement comes at the expense of increased number of parameters and computation. We need to estimate $M(2D + 1)$ parameters for an M-mixture GMM, with diagonal covariances, where D is the dimension of the data. For a GMM with full covariance matrices, we need to estimate $M(0.5D^2 + 1.5D + 1)$ parameters, which in turn requires large amount of data.

Given a trained GMM and a data point \mathbf{x} , the posterior probability that it is generated by the i^{th} Gaussian component c_i can be computed using the Bayes' rule as follows:

$$P(c_i/\mathbf{x}) = \frac{w_i \mathcal{N}(\mathbf{x}/\mu_i, \Sigma_i)}{p(\mathbf{x})} \quad (3.2)$$

The vector of posterior probabilities for $i = 1, 2, \dots, M$ is called Gaussian posterior vector. Gaussian posterior representation was found be better suited for Phoneme across talker(PaT) task than the MFCC coefficients and the performance of Phoneme across Context is degraded. [9], [10].

3.3 Posterior Extraction using GBRBM

A Restricted Boltzmann machine (RBM) is an undirected bipartite graphical model with visible and hidden layers [11]. In contrast to a Boltzmann machine, intra-layer connections do not exist in RBM, and hence the word *restricted*. Example architecture of RBM is shown in Fig.???. In an RBM, the output of a visible unit is conditionally Bernoulli given the state of hidden units. Hence the RBM can model only binary valued data. On the other hand in a GBRBM, the output of a visible unit is conditionally Gaussian given the state of hidden units, and hence it can model real valued data. Both in RBM and GBRBM, the output of a hidden unit is conditionally Bernoulli, given the state of visible units, and hence can assume only binary hidden states. Since the same binary hidden state is used to sample all the dimensions of the visible layer, GBRBM are capable of modelling correlated data. A GBRBM can be completely characterized by its parameters, i.e., weights, hidden biases, visual biases and variances of the visible units. The GBRBM associates an energy for every configuration of visible and hidden states. The parameters of the GBRBM are estimated such that the overall energy of GBRBM, over the ensemble of training data, reaches a minima on the energy landscape.

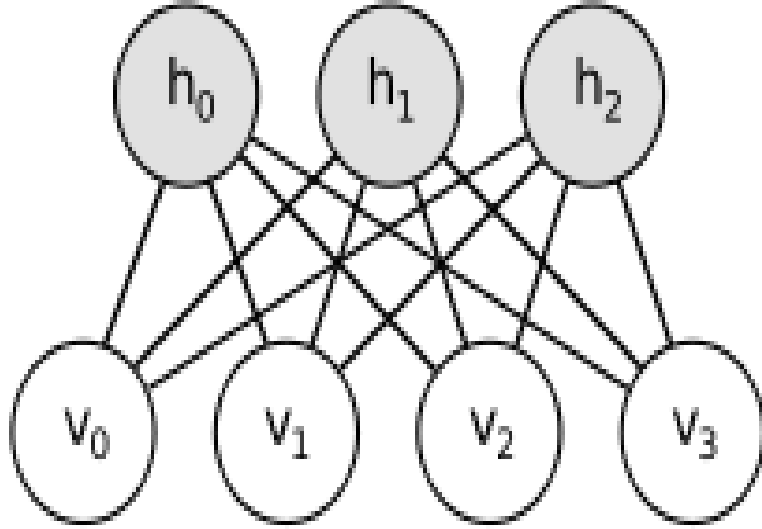


Figure 3.2: Network architecture of a Restricted Boltzmann Machine

The energy function for GBRBM, for a particular configuration of real-valued visible state vector \mathbf{v} and binary hidden state vector \mathbf{h} , is defined as [12]

$$E(\mathbf{v}, \mathbf{h}) = \sum_{i=1}^V \frac{(v_i - b_i^v)^2}{2\sigma_i^2} - \sum_{j=1}^H b_j^h h_j - \sum_{i=1}^V \sum_{j=1}^H \frac{v_i}{\sigma_i} h_j w_{ij}, \quad (3.3)$$

where V and H are total number of visible and hidden units, v_i is the state of i^{th} visible unit, h_j is the state of j^{th} hidden unit, w_{ij} is the weight connecting the i^{th} visible unit to the j^{th} hidden unit, b_i^v is the bias on the i^{th} visible unit, b_j^h is the bias on the j^{th} hidden unit, σ_i is the variance of the i^{th} visible unit.

The joint density of the visible and hidden unit states is related to the energy of the network as

$$p(\mathbf{v}, \mathbf{h}) \propto e^{-E(\mathbf{v}, \mathbf{h})} \quad (3.4)$$

The parameters of the GBRBM are estimated by maximizing the likelihood of the data. Because of the issues in tractability of true gradient of the likelihood, Markov Chain Monte Carlo (MCMC) approximation methods were used to train RBMs. Contrastive Divergence (CD) [13] is one such technique which is proven to work well in practice. Energy of the system which is directly related to likelihood, is minimized in CD algorithm. Variants of CD include Persistent CD (PCD), Fast PCD, Tempered Transitions and Parallel Tempering. In this work, we use CD algorithm. The updates for

the parameters can be estimated using Contrastive Divergence (CD) algorithm, as follows:

$$\begin{aligned}\Delta w_{ij} &\propto \left\langle \frac{v_i h_j}{\sigma_i} \right\rangle_{data} - \left\langle \frac{v_i h_j}{\sigma_i} \right\rangle_{recall} \\ \Delta b_i^v &\propto \left\langle \frac{v_i}{\sigma_i^2} \right\rangle_{data} - \left\langle \frac{v_i}{\sigma_i^2} \right\rangle_{recall} \\ \Delta b_j^h &\propto \langle h_j \rangle_{data} - \langle h_j \rangle_{recall} \\ \Delta \sigma_i &\propto \langle \gamma \rangle_{data} - \langle \gamma \rangle_{recall}\end{aligned}$$

where

$$\gamma = \frac{(v_i - b_i^v)^2}{\sigma_i^3} - \sum_{j=1}^H \frac{h_j w_{ij} v_i}{\sigma_i^2}$$

and $\langle \cdot \rangle_{data}$ denotes expectation over the input data, and $\langle \cdot \rangle_{recall}$ denotes expectation over its reconstruction.

Contrastive Divergence uses two tricks to speed up the sampling process.

- since we eventually want $p(v) \approx p_{train}(v)$ (the true, underlying distribution of the data), we initialize the Markov chain with a training example (i.e., from a distribution that is expected to be close to p , so that the chain will be already close to having converged to its final distribution p).
- CD does not wait for the chain to converge. Samples are obtained after only k -steps of Gibbs sampling. In practice, $k=1$ has been shown to work surprisingly well.

In one cycle of CD algorithm, CD_1 , the probability of firing of a hidden unit, j , is activation of sigmoid function for an input of weighted sum of previous layer (visible layer) activations as shown in the following equation. Since the hidden units are stochastic, output is forced to be 1 if output of activation function is greater than a random number sampled from uniform distribution $[0, 1]$. Binary activations are sent back to visible layer for reconstruction. Visible units are assumed to be Gaussian in GRBBM. Visible activations are sampled from Gaussian distribution with mean equal to weighted sum of inputs from hidden layer and learnt variance. For the second cycle of CD, these reconstructions are fed as input to visible units.

Reconstruction error can be used to monitor progress of learning but can not be relied entirely as it does not correlate with objective function, energy equation. It is the difference between input data point and reconstructed visible activations.

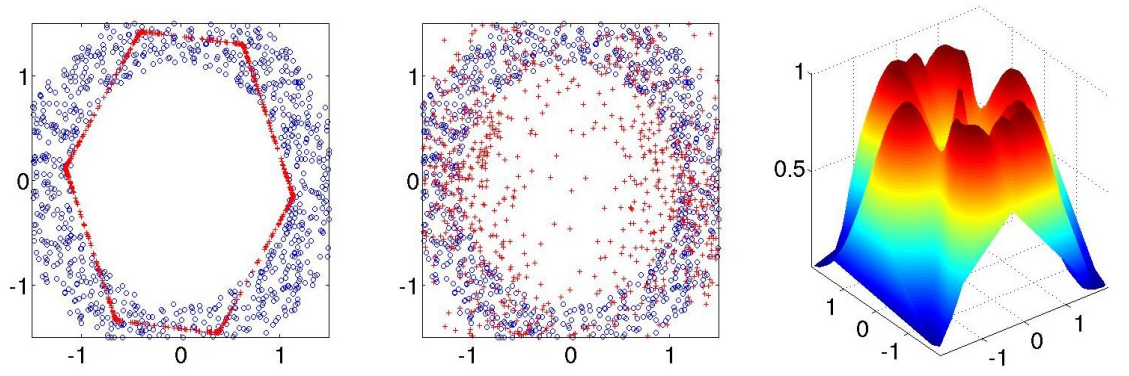
Generally, overfitting occurs when number of examples used for training are not sufficient to estimate model parameters. Generalizability is required for a model to be usable for a test data

point unseen in training data. For a test data point supplied to overfitted model, the outcome is erroneous which can not be expected before hand. For a well trained model, outcome for a test point can be expected. In the problem of learning underlying probability distribution, probability of a test data point drawn from current model gives an idea of usability of model. But it is difficult to compute probability in the case of GBRBM as calculation of partition function is computationally intensive. However, comparison of free energies of training data and validation data is enough as probability is directly related to free energy. Large positive difference between free energies of validation data and training data denotes model overfitting. It can be avoided by using several techniques: Cross-validation, Regularization, early stopping. In this work, regularization is used to avoid overfitting. Sparsity and weight-decay terms are added in update equations for regularization.

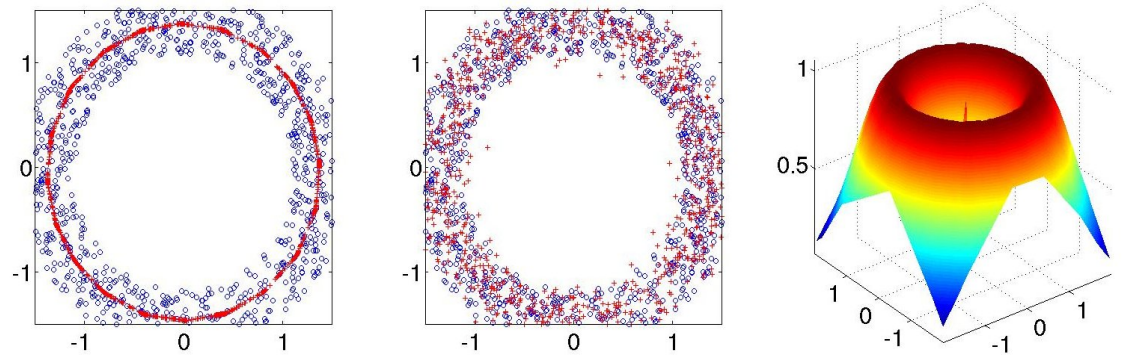
Since the hidden units activations are stochastic, any initialization works but badly initialized models take large number of iterations to get converged. Usually weights are initialized to small random values sampled from zero-mean Gaussian distribution.

During each cycle of CD, the energy associated with the joint configuration of visible and hidden states is supposed to decrease, although there is no theoretical guarantee. After a large number of iterations, the expectation of the energy does not change any more, indicating thermal equilibrium of the network. At thermal equilibrium, the GBRBM models the joint density of the training data. The trained GBRBM model is capable of generating the data points which resemble the training data.

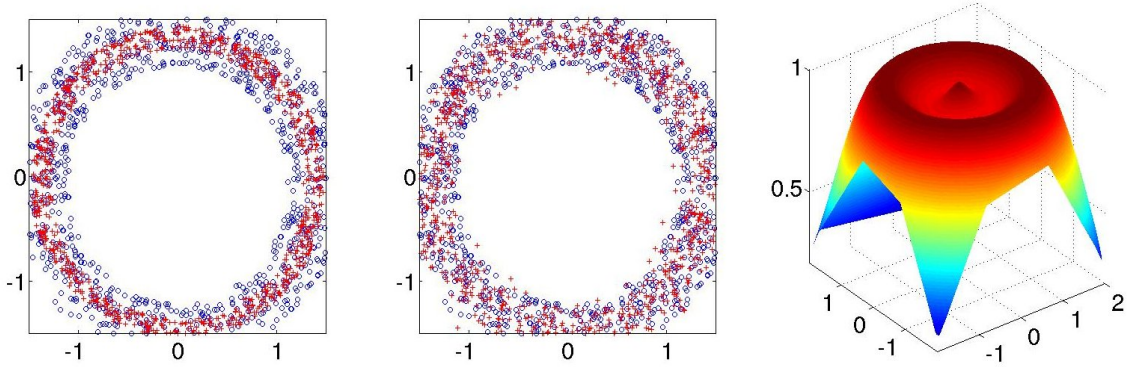
The distribution capturing capability of GBRBM is illustrated, in Fig.3.3, with 2-dimensional data uniformly distributed along a circular ring. First column shows the mean of the unbiased samples generated by the model, second column shows the reconstructed data points, and third column shows the estimated density function. Input data points are plotted as blue 'o' and reconstructions are plotted as red '+'. A GBRBM with different number of hidden units is trained, to capture the joint density of this data, for 200 cycles using CD. The number of hidden units plays an important role in capturing the distribution of input data. For a GBRBM with H hidden units, the hidden state vector can take at most 2^H binary configurations. However, only a few of those 2^H hidden states sustain at the thermal equilibrium of the network. Hence, the reconstructed visible state can take a maximum of 2^H different mean vectors. The mean visible layer activations, for a GBRBM with 3 hidden units is shown in Fig. 3.3(a). In this case the mean of the circular ring is approximated by a hexagon, i.e., with 6 (out of $< 2^3$ possible) stable states at thermal equilibrium.



(a)



(b)



(c)

Figure 3.3: Illustration of distribution capturing capability of GBRBM. Left: Original training data blue, and mean of the unbiased samples generated by trained GBRBM, Middle: Original training data blue, and unbiased samples generated by GBRBM, Right: captured density plot. GBRBM plots trained with (a) 3 (b) 10 (c) 200 hidden layer neurons

As the number of hidden units increase, the number of possible stable states also increase, leading to a better approximation of the mean of the input data. The mean of the unbiased samples generated by a GBRBM with 10 hidden units, in Fig. 3.3(b), faithfully estimated the mean of the circular ring. When the number of hidden units is further increased to 200, the mean activations are spread over input data leading to a overfit. The data distributions captured by GBRBMs, with different hidden units, are shown in the third column of Fig. 3.3. It is clear that the GBRBM with 10 hidden units has captured the input distribution better.

The variance parameters of the visible units also influence the distribution capturing capabilities of the GBRBM. Usually, when the GBRBM is used to pre-train the first layer of an MLP the variances are simply set to one. However, variance learning is necessary when GBRBM is used to capture the joint density of the input data.

Chapter 4

Results and Discussions

Table 4.1: Comparison of spectral features using MP-ABX tasks.

Features	PaC	PaT	TaP
MFCC	17.67	19.69	19.62
IFCC	20.99	25.95	15.78

In the table 4.1 MFCC performs better when compared to IFCC, hence we developed unsupervised models by using MFCC spectral features and we compared the results among the models by changing the dimensions of posterior features.

Table 4.2: Performance of GMM based Model.

Task	MFCC	m=32	m=64	m=128
PaC	15.77	18.59	17.89	18.75
PaT	30.93	26.33	26.65	29.69

m=number of Gaussian Mixtures.

In the table 4.2 contains the performance of PaT and PaC for MFCC and posteriors of GMM. From the table we can clearly say that the PaT task is improved for GMM model over MFCC features.

Table 4.3: Performance of RBM Model.

Task	MFCC	h=32	h=64	h=128
PaC	15.77	17.70	17.73	17.81
PaT	30.93	29.88	30.03	30.20

h=number of Hidden neurons In the table 4.3 contains the performance of PaT and PaC for MFCC and posteriors of RBM. From the table we can clearly say that the PaT task is improved

for RBM model over MFCC features. And the PaC task got better results compared to GMM posteriors.

Chapter 5

Conclusion

Analysis of features of speech extracted from magnitude and phase of the complex analytic representation was carried out using MP ABX tasks. Posterior features of Gaussian Mixture Model and Gaussian Bernoulli Restricted Boltzmann Machine were extracted . Their performances with respect to phoneme and speaker discriminative MP-ABX tasks based on CV pair speech stimuli were evaluated and compared with those of conventional MFCC features. It was observed that the GMM posterior features are efficient in talker discrimination task but its performance reduces with increase in number of Gaussian mixtures in the model. On the other hand, RBM posterior features gives better performance in speaker discrimination task compared to GMM posteriors and its performance in creases with increase in dimension and then decreases. This study suggests the importance of modeling of spectral features by using unsupervised models in revealing speaker characteristics especially in Zero resource settings.

Bibliography

- [1] T. Schatz, V. Peddinti, F. Bach, A. Jansen, H. Hermansky, and E. Dupoux. Evaluating speech features with the Minimal-Pair ABX task: Analysis of the classical MFC/PLP pipeline. In INTERSPEECH 2013: 14th Annual Conference of the International Speech Communication Association. 2013 1–5.
- [2] <http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWalgorithm.htm>.
- [3] <http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/>.
- [4] J.-J. Ding. Time Frequency Analysis Tutorial. *R99942057* .
- [5] B. Logan et al. Mel Frequency Cepstral Coefficients for Music Modeling. In ISMIR. 2000 .
- [6] K. Vijayan, V. Kumar, and K. S. R. Murty. Feature Extraction from Analytic Phase of Speech Signals for Speaker Verification. In Fifteenth Annual Conference of the International Speech Communication Association. 2014 .
- [7] D. Reynolds. Gaussian mixture models. In Encyclopedia of Biometrics, 659–663. Springer, 2009.
- [8] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern classification. John Wiley & Sons, 2012.
- [9] Y. Zhang and J. R. Glass. Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU Merano/Meran, Italy. 2009 398–403.
- [10] R. R. Pappagari, K. Rout, and K. S. R. Murty. Query word retrieval from continuous speech using GMM posteriorgrams. In International Conference on Signal Processing and Communications (SPCOM), 2014, Bangalore, India. 2014 1–6.
- [11] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In Proceedings of the 24th international conference on Machine learning. ACM, 2007 791–798.
- [12] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science* 313, (2006) 504–507.
- [13] M. A. Carreira-Perpinan and G. E. Hinton. On contrastive divergence learning. In Proceedings of the tenth international workshop on artificial intelligence and statistics. 2005 33–40.