Using Gaussian Mixtures for Hindi Speech Recognition System

R. K. Aggarwal and M. Dave

Associate Professor, Department of Computer Engineering,
NIT, Kurukshetra, (Haryana), India
rka15969@gmail.com, mdave67@gmail.com

Abstract

The goal of automatic speech recognition (ASR) system is to accurately and efficiently convert a speech signal into a text message independent of the device, speaker or the environment. In general the speech signal is captured and pre-processed at front-end for feature extraction and evaluated at back-end using the Gaussian mixture hidden Markov model. In this statistical approach since the evaluation of Gaussian likelihoods dominate the total computational load, the appropriate selection of Gaussian mixtures is very important depending upon the amount of training data. As the small databases are available to train the Indian languages ASR system, the higher range of Gaussian mixtures (i.e. 64 and above), normally used for European languages, cannot be applied for them. This paper reviews the statistical framework and presents an iterative procedure to select an optimum number of Gaussian mixtures that exhibits maximum accuracy in the context of Hindi speech recognition system.

Keywords: *ASR, HMM, MFCC, HLDA, Gaussian mixture, Hindi, Feature extraction, Acoustic modelling, MLE, MPE*

1. Introduction

The speech recognition problem is the task of taking an utterance of speech signal as input, captured by a microphone (or a microphone array), a telephone or other transducers, and converting it into a text sequence as close as possible to what was represented by the acoustic data. To make such a system ubiquitous, it is important that the system should be independent of speaker and language characteristics such as accents, speaking styles, disfluencies (particularly important in spontaneous speech), syntax and grammar, along with the capability of handling a large vocabulary [1].

Although, ASR technology has made remarkable progress over the last 50 years, there still exist a large number of problems that need to be solved. Gaussian mixture evaluation of acoustic signals is one such problem which is a computationally expensive task. In such systems, calculation of the state likelihoods makes a significant proportion (between 30-70%) of the total computational load [2]. A range of 8 to 64 mixture components per state have been found useful depending on the amount of training data. It is a tedious, time consuming and expensive process if in Gaussian mixture model we gradually increase the number of mixtures and then optimize it. In this paper we present a novel approach to speedup statistical pattern classification by reducing the time consumed in likelihood evaluations of feature vectors by using optimal number of Gaussian mixture components selected on the basis of empirical observations.

Various experiments were conducted using hidden Markov model (HMM) by varying number of mixtures at back-end and by using MFCC and its extension for feature extraction at front-end. Analysis was carried out to select the parameters giving the best results at both ends.

All the investigations are based on the experiments conducted in typical field condition and in the context of databases available for Indian languages. Rest of the paper is organized as follows: Section 2 describes the architecture and working of ASR with the issues related to data preparation for Indian languages. Feature extraction techniques are given in section 3. Section 4 presents the use of HMM with mixture of multivariate Gaussians. In section 5, an experimental comparison of ASR performance with various mixtures and training methods is presented. Finally, the paper concludes with a brief discussion of the experimental results.

2. Design and Modeling of ASR

2.1 Structure and Working

State-of-the-art ASR systems consist of four basic modules: the signal processing components (i.e., pre-processing and feature extraction), the set of acoustic models (i.e. HMM), the language model (i.e. N-gram estimation) and search engine for final decoding as shown in Figure 1. First signal processing module generates features from given speech signal and then pattern classifier module evaluates these features to produce the most likely word sequence as output, with the help of available statistical models and lexicon. The acoustic model typically consists of two parts. The first is to describe how a word sequence can be represented by sub-word units, often known as pronunciation modelling. The second is the mapping from each sub word units to acoustic observations [3]. Language model works using the results of the acoustic models. It accepts the various competitive hypotheses of words from the acoustic models and generates a probability for each sequence of word. This probability is combined with the acoustic models' likelihood assigned to the respective sequence, providing the overall probability of pronouncing sequence words with the given acoustics.

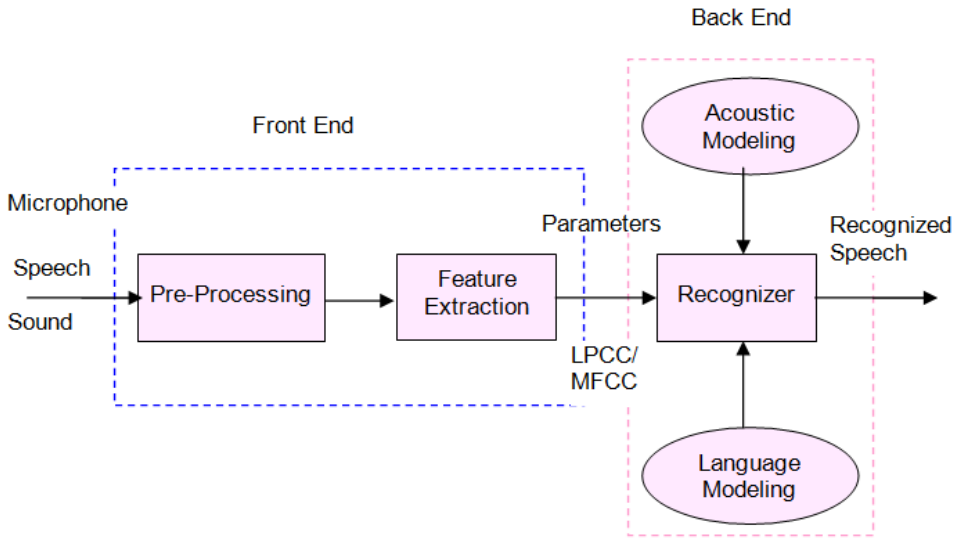

Figure 1. ASR Architecture

ASR is a pattern classification approach divided into training and decoding (i.e. testing) parts. The training task consists of taking a collection of utterances with associated word label, and learning an association between the specified word models and observed acoustics. For this it requires various information sources (i.e. databases and corpus) that include waveforms of isolated words or of phonetically labelled phrases. During recognition, the sequence of symbols generated by the acoustic components is compared with the set of words present in the lexicon to produce optimal sequence of words that compose the system's final output. In order to cover the words that are not seen in the acoustic training data, it is necessary to have a grapheme-to-phoneme (G2P) system that uses the word orthography to guess the pronunciation of the word [4].

2.2 Data Preparation for Indian Languages

There are 22 officially recognized languages among the 200 or so different written languages used in the Indian subcontinent. Apart from few Perso-Arabic scripts (i.e. Kasmiri, Sindhi and Urdu), all the other scripts (i.e. Assamese, Bengali, Devanagari, Gujarati, Kannada, Oriya, Punjabi, and Telugu etc.) used for Indian languages have evolved from the ancient Brahmi script and have a common phonetic structure. Brahmi-derived scripts are further subdivided into northern and southern groups. The northern group (of which Devanagari is a derivative) extends from northwestern India to Nepal and Tibet in the north, across the subcontinent to Bengal and Bangladesh and further east to southeast Asia (including Thailand, Indonesia and Korea). The other group, also known as Dravidian scripts (i.e., Tamil, Telugu, Kannada and Malayalam) is used predominantly in south India. Devanagari, as the script of Sanskrit literature, became the most widely used script in India by the 11th century. Languages written in Devanagari include Nepali, Marathi, Bengali, Gujarati and Hindi as well as Tibetan and Burmese.

Table 1: Hindi Character Set

Vowels	अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ	अं	अः
	a	ā	i	ī	u	ū	r	e	ai	o	au	aṅ	aḥ
Gutturals (कवर्ग)	क	ख	ग	घ	ङ								
	ka	kha	ga	gha	ṅa								
Palatals (चवर्ग)	च	छ	ज	झ	ञ								
	ca	cha	ja	jha	ña								
Cerebrals (टवर्ग)	ट	ठ	ड	ढ	ण								
	ṭa	ṭha	da	dha	ṇa								
Dentals (तवर्ग)	त	थ	द	ध	न								
	ta	tha	da	dha	na								
Labials (पवर्ग)	प	फ	ब	भ	म								
	pa	pha	ba	bha	ma								
Semi-Vowels	य	र	ल	व									
	ya	ra	la	va									
Sibilants	श	ष	स										
	sa	pa	sa										
Aspirate	ह												
	Ha												

The broad division of all sounds in human languages can be classified into two categories viz. vowels (v) and consonants(c). The vowels in Indian languages include short and long versions of the same sound. There are 12 basic vowels in Hindi languages which are called Barakhadi. The basic set of consonants has been categorized according to the place and manner of articulation as given in Table 1. Besides this, in Hindi language there are some graphemes which do not have atomic sound. They correspond to two or more concatenated phoneme sound, for example, AUM [ॐ] and RI [ऋ]. These can be mapped to a string of unit phonemes.

3. Feature Extraction and Reduction

This phase covers two steps: In the first step, cepstrum coefficients are extracted by applying non uniform filters on Fourier spectrum of speech signals, with the discrete cosine transform. The second step is aimed at incorporating the techniques which project the features into low dimensional subspace, while preserving discriminative information. These techniques are based on linear transformation schemes like principal component analysis (PCA) [5], linear discriminant analysis (LDA) [6] and Heteroscedastic linear discriminant analysis (HLDA) [7].

3.1 Standard MFCC

Mel cepstral feature extraction is used in some form or another in virtually every state of the art speech and speaker recognition system. First, speech samples are divided into overlapping frames. The usual frame length is 25 ms and the frame rate is 10 ms. Each frame is usually processed by pre-emphasis filter to amplify higher frequencies. In the next step Hamming window is applied and Fourier spectrum is computed for the windowed frame signal. A Mel spaced bank of filters is then applied to obtain a vector of log energies. Usually 20 to 40 filters are used depending on application. The output of the filter-bank is then converted to cepstral coefficients by using discrete cosine transform (DCT), where only the first 12 coefficients are retained for computing the feature vector. Finally the feature vector consists of 39 values including the 12 cepstral coefficients with one energy, 13 delta cepstral coefficients and 13 delta delta coefficients [8].

3.2 Extended MFCC

Thirteen extra triple delta features are added in standard 39 MFCC features forming a feature vector of 52 values. These 52 values are then reduced to 39 by applying any feature reduction technique. These techniques are based on linear transformation schemes like principal component analysis (PCA), linear discriminant analysis (LDA) and Hetroscedastic linear discriminant analysis (HLDA). HLDA, first proposed by N. Kumar [7] has been widely used for various feature combination techniques. It maximizes the likelihood of all the training data in the transformed space and each training sample contributes equally to the objective function. We have used HLDA for feature reduction and this procedure is named extended MFCC as shown in Figure 2.

3.3 Robust Features

In noisy environments when training and testing conditions are severely mismatched, these features cannot work well. Therefore, feature domain signal processing methods are applied

to enhance the distorted speech. Spectral subtraction is widely used as a simple technique to reduce additive noise in the spectral domain [9]. In order to eliminate the convolutive channel effect, Cepstral Mean Normalization (CMN) is applied which removes mean vector in the acoustic features of the utterance. An extension of CMN, Cepstral Variance Normalization (CVN) also adjusts feature variance to improve ASR robustness [10]. Relative spectra (RASTA) processing and its variants such as J-RASTA, phase corrected RASTA have also been used to reduce both communication channel effects and noise distortion [11].

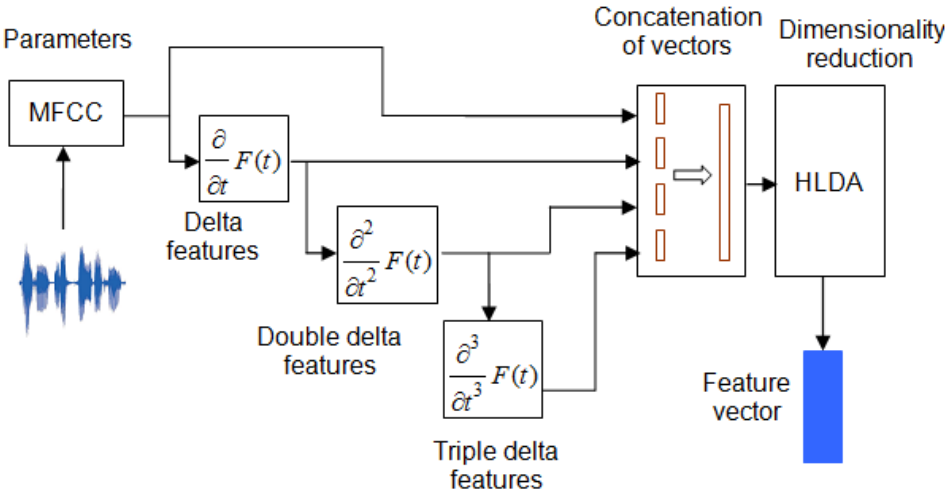

Figure 2. Extended MFCC

4. Gaussian Mixture HMM

In this method continuous density hidden Markov models are used to match the phonetic information of speech signal with the feature vectors derived at front end. Multivariate Gaussian mixtures are used to calculate the likelihood of observation vectors (i.e. spectral features). Representation of phonetic information, HMM topology and number of Gaussian mixtures are the key issues for the implementation of these statistical techniques [12].

4.1 Phonetic Representation of Speech Signals

To decide how speech and non speech units should be represented is essential to build an ASR system. In the limited domain applications where only small vocabulary is needed, the whole word as linguistic unit is the natural choice and exhibit good results. This method is rather unpractical for open systems where new words can be tested. It is also totally infeasible for large vocabulary continuous speech recognition system to use whole word model where several dozens of realizations for every word are required. The opposite extreme is to construct models only for single phonemes which would solve the flexibility and feasibility problems. However, the acoustic realization of a phoneme may heavily depend on the context in which it occurs. This effect is usually called coarticulation which causes a sudden drop in the accuracy. In order to account for the acoustic variability and coarticulation effects, context-based subunits are used, for instance triphones and quinphones. Triphone based system are common, in which subword units are phonemes and each HMM represents a phoneme in the context of a distinct preceding and following phoneme. Context dependent

models like triphones can be constructed in two ways: either word internal or cross word. When constructing word internal models, context beyond the word borders are not considered. On the other hand, for cross word triphones, the phonemes at the end or beginning of neighbouring words are considered to affect the phonology used for modeling [13]. Context dependent modelling significantly increases the number of model parameters to be estimated. The most common solutions used some form of parameter or distribution tying, in which equivalence classes are defined between model constructs (e.g. HMM states) and then constructs in the same class share the same parameters for associated distributions [14].

4.2 Hidden Markov Model

Each subword unit is realized by a hidden Markov model in most state-of-the-art LVCSR systems. HMM is a statistical model [3] for an ordered sequence of symbols, acting as a stochastic finite state machine which is assumed to be built up from a finite set of possible states, each of those states being associated with a specific probability distribution or probability density function (pdf). Three fundamental problems of HMMs are probability evaluation, determination of the best sequence, and parameter estimation. The probability evaluation can be realized easily with the forward algorithm [15]. The determination of the best state sequence is often referred as a decoding or search process. Viterbi search [16] and A* search [17] are two major search algorithms. The parameter estimation in ASR is solved with the well-known maximum likelihood estimation (MLE) using a forward-backward procedure [18]. Several discriminative training methods have been proposed in recent years to boost ASR system accuracy like maximum mutual information estimation (MMIE); minimum classification error (MCE); and minimum word error/minimum phone error (MWE/MPE) [19].

In MLE based HMM we find those HMM model parameters, λ, which maximize the likelihood of the HMMs having generated with training data. Thus, given training data $Y^{(1)}.....Y^{(r)}$ the maximum likelihood (ML) training criterion may be expressed as:

$$F_{MLE}(\lambda) = \frac{1}{R}\sum_{r=1}^{R}\log(p(Y^{(r)} \mid w_{ref}^{(r)};\lambda)) \qquad (1)$$

where $Y^{(r)}$ is the r^{th} training utterance with transcription $w_{ref}^{(r)}$. This optimization is normally performed using EM [20]. However, for ML to be the best training criterion, the data and models would need to satisfy a number of requirements, in particular, training data sufficiency and model-correctness [21].

The MPE criterion is a smoothed approximation to the phone transcription accuracy measured on the output of a word recognition system given the training data. The objective function in MPE, which is to be maximized, is:

$$F_{MPE}(\lambda) = \sum_{r=1}^{R}\sum_{S} P_{\lambda}^{k}(S \mid O_r)A(S, S_r) \qquad (2)$$

where λ represents the HMM parameters; $P_{\lambda}^{k}(S \mid O_r)$ is defined as the scaled posterior probability of the sentence S being the correct one (given the model) and formulated by:

$$P_{\lambda}^{k}(S \mid O_r) = \frac{P_{\lambda}(O_r \mid S)^k P(S)^k}{\sum_u P_{\lambda}(O_r \mid u)^k P(u)^k} \qquad (3)$$

Where K is the scaling factor typically less than one, O_r is the speech data for r^{th} training sentence; and $A(S, S_r)$ is the raw phone transcription accuracy of the sentence S given the

reference, S_r, which equals the number of reference phones minus the number of errors [22, 23].

4.3 Database for Speech Recognition

For the estimation of acoustic model parameters λ, and evaluation of ASR performance a corpus of training utterances is required, which is also known as speech database. Ideally, the databases of speech are labelled with textual transcriptions and each speech signal is aligned with its words and phones, so that word-based and phone-based models could be trained automatically.

For the design and development of European languages ASR systems, large and standard databases are available which were prepared by various agencies. For example, TIMIT and ATIS are two of the most important databases that are used to build acoustic models of American English in ASRs [24]. But to prepare such kind of standard databases for Indian languages, no much effort has been done so far. The few databases available for Indian languages are relatively small and phonetically not very rich, as these were prepared by various research groups especially for their own use.

4.4 Mixtures of Multivariate Gaussian

To model the complex speech signal, mixtures of Gaussian have been used as emission pdfs in the hidden Markov models. In such systems, the output likelihood of a HMM state S for a given observation vector, X_n can be represented as a weighted sum of probabilities:

$$p(X_n \mid S) = \sum_{k=1}^{K} w_k p_k(X_n) \qquad (4)$$

Where, parameters of the state pdf are number of mixture components K; their weighing factors, w_k, which satisfies $w_k > 0$ and $\sum_{k=1}^{K} w_k = 1$; mean vector μ_k and the variance-covariance matrix $\sum_k$ of the k^{th} mixture component. Each mixture component belongs to a D-dimensional multivariate Gaussian density function defined as:

$$p_k(X_n) = \frac{1}{(2\pi)^{D/2} \mid \sum_k \mid^{1/2}} \exp\left[-\frac{(x_n - \mu_k)^T \sum_k^{-1} (x_n - \mu_k)}{2} \right] \qquad (5)$$

In practice the full covariance matrices are reduced to diagonal covariance due to computational and data sparseness reasons. By substituting the values of probabilities defined in Equation (5), the state model in Equation (4) becomes the Gaussian mixture model (GMM) defined as:

$$p(X_n \mid S) = \sum_{k=1}^{K} Z_k \exp\left[-\frac{1}{2} \sum_{q=1}^{D} \frac{(x_{nq} - \mu_{kq})^2}{\sigma_{kq}^2} \right] \qquad (6)$$

where Z_k is a constant for each Gaussian i.e.,

$$Z_k = \frac{w_k}{(2\pi)^{D/2} (\Pi_{q=1}^{D} \sigma_{kq}^2)^{1/2}}$$

In order to compute efficiently and to avoid underflow, probabilities are computed in log domain. Therefore the log likelihood can be expressed as:

$$\log p(X_n \mid S) = \log add_{k=1}^{K} \left[\log(Z_k) - \frac{1}{2} \sum_{q=1}^{D} \frac{(x_{nq} - \mu_{kq})^z}{\sigma_{kq}^2} \right] \quad (7)$$

Here, the function $\log add[\bullet]$ is defined as follows:

$$\log add_{k=1}^{K}[y_k] = \log\left[\sum_{k=1}^{K} \exp(y_k) \right] \quad (8)$$

In a typical HMM-based LVCSR system, the number of model states ranges from 2000 to 6000, each of which is a weighted sum of typically 8–64 multidimensional Gaussian distributions as in Equation (6). For each input frame, the output likelihoods should be evaluated against each active state. Therefore, the state likelihoods estimation is computationally intensive and takes about 30–70% of the total recognition time [25]. This kind of likelihood-based statistical acoustic decoding is so time consuming that it is one of the most important reasons why the recognition is slow. Some LVCSR systems might even decode speech several times slower than real time; that is to say, these systems are not practical for most spontaneous applications, such as man-machine dialogue. Therefore, it is necessary to develop efficient techniques in order to reduce the time consumption of likelihood computation without a significant degradation of recognition accuracy [2].

4.5 Large Margin Training of GMM

Inspired by support vector machines (SVMs), a new algorithm known as large margin Gaussian mixture model (GMM), was proposed in literature for multiway classification [26]. SVM provides state-of-the-art performance for many applications in pattern recognition but it is challenging to apply the same in large applications like ASR, which does not require binary classification. Another limitation of SVM is the high computation and memory requirements at the time of training and testing [27]. The parameters of large margin GMMs are trained as in SVMs, by a convex optimization that focuses on examples near the decision boundaries. For complex applications like ASR, large margin GMM has certain advantages over SVM. The reason is that large margin GMMs use ellipsoids to model classes, which induce non-linear decision boundaries in the input space, in place of half-spaces and hyperplanes used by SVMs. One potential weakness of LME is that it updates models only with accurately classified samples. However, it is well known that misclassified samples are also critical for classifier learning. Consequently, LME often needs a very good preliminary estimate from the training set to make the influence of ignoring misclassified samples small [28].

5. Experimental Results

The input speech was sampled at 12 kHz and then processed at 10 ms frame rate with a Hamming window of 25 ms to obtain the feature vectors. The CDHMM with linear left-right topology is used to compute the score against a sequence of features for their phonetic transcription. To compute the likelihood of a given word, the word is broken into sub words or its constituent phones, and the likelihood of the phones is computed from the HMMs. Three differerent HMMs based on word, context independent (CI) phones, and triphones

modeling units, were implemented. In phoneme based HMM, total 48 CI phone models were used. We used word internal triphone models for our experiments.

At front end, two feature extraction methods standard MFCC and extended form of MFCC were investigated. At back end, two training methods MLE and MPE were used. The experiment was performed on a set of speech data consisting of four hundred words of Hindi language recorded by 10 male and 10 female speakers. Since databases from non-Indian languages cannot be used for Hindi (owing to the language specific effects), we have developed our own corpus which includes documents from EMILLE text corpus [29], and popular Hindi news papers. Testing of randomly chosen fifty sentences spoken by different speakers was performed and recognition rate (i.e. accuracy) was calculated as:

$$Recognition\ rate = \frac{Successfully\ detected\ words}{Number\ of\ words\ in\ test\ set}$$

The decoding was performed in two phases:

- In the first phase, standard 39 MFCC features (i.e. MFCC +Δ+ΔΔ) were used at front-end and models were trained using MLE, MPE and LME.
- In the second phase, 3rd order MFCC (i.e. MFCC + Δ+ΔΔ+ΔΔΔ) features were used, they were processed by HLDA, and models were trained using MLE, MPE and LME.

Many public domain software tools are available for the research work in the field of ASR such as Sphinx from Carnegie Mellon University [30], hidden Markov model toolkit (HTK) from Cambridge University [31] and LVCSR engine Julius from Japan [32]. We have used HTK-3.4.1 in LINUX environment for our experimental work. Further the experiment consists of an evaluation of the system using the room condition and standard speech capturing hardware such as sound card and a head set microphone.

Using the frame synchronous CDHMM statistical model for training and testing the following results were analyzed:

- Variation in the recognition rate with the number of Gaussian mixtures for two different feature extraction techniques.
- Variation in the recognition rate with the number of vocabulary sizes and with the different word representation models.
- Variation in accuracy with different types of training methods as MLE and MPE.
- Variation in the accuracy with the number of Gaussian mixtures for different modeling units.

5.1 Experiment with Different Mixtures

Experiments were performed six times with different number of Gaussians along with both standard and extended MFCC features as shown in Figure 3. Further, triphone model as fundamental speech unit and LME technique for parameter estimation, in HMM, were used. Mixtures are in multiples of 2 as it is convenient to split then in this form. According to results, there is an improvement in performance up to 4 mixtures components, but after that no improvement was observed. Extended MFCC gained around 2% improvement over standard MFCC.

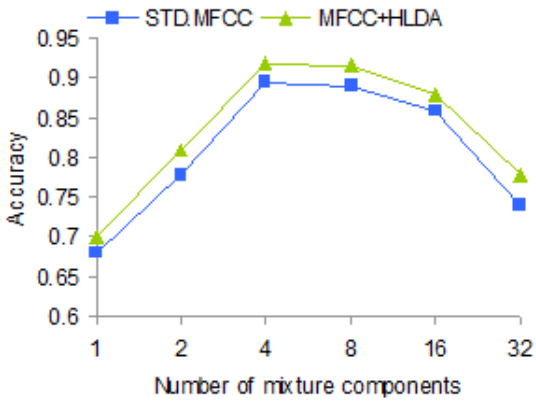

Figure 3: Accuracy versus Gaussian Mixtures

5.2 Experiment with Different Vocabulary Sizes

In this experiment, accuracy of the system was observed by varying the size of vocabulary (100 words, 200 words, 300 words and 400 words). Smaller the size of vocabulary, lesser the chances of confusion and hence better should be the accuracy. This fact is supported by results as shown in Figure 4. Four Gaussian mixtures and standard MFCC were used in training of the model to get best results. At back end two models, whole word model and sub word triphone model were investigated with various vocabulary sizes using standard MLE training method. It is not preferable to use whole word model beyond 200 vocabulary size. Only domain specific and small speech system recognition systems can use word models.

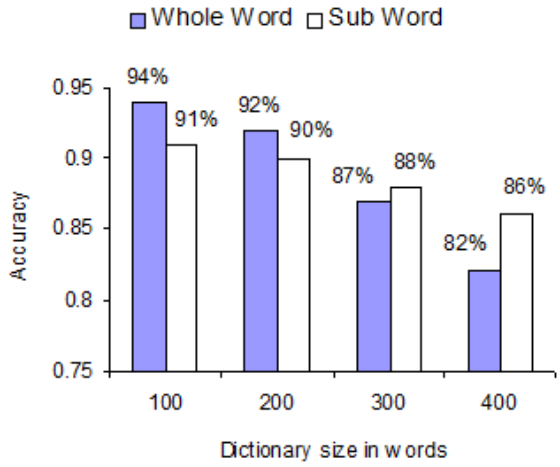

Figure 4: Accuracy versus Dictionary Size

5.3 Experiment with Different Training Methods

Experiments were performed by using MLE and MPE training methods. At front-end standard MFCC and 3rd order MFCC integrated with HLDA, the two separate feature extraction techniques were used for speech signal parameterization. The results are shown in Table 2. Four mixture components were used to train HMM in various experiments. Discriminative MPE technique is far better then MLE as it has shown about 2 to 3% improvement in performance. Margin based LME supersedes the standard MLE and MPE.

Table 2. Accuracy with Training Methods

	Standard MFCC (Non-HLDA)	Extended MFCC (HLDA)
MLE	86%	87.6%
MPE	88.5%	91%
LME	89.6%	91.8%

5.4 Experiment with Different Modelling Units

Experiments were performed by using three standard modelling units viz. word model, phoneme model, triphone model in the context of Hindi language using discriminative training technique [33, 34]. All the models were processed and evaluated with each category of Gaussian mixtures as shown in Table 3. Context independent (CI) phoneme model is better than word model. However, triphone model outperforms both phoneme model as well as word model.

Table 3. Model Accuracy versus Gaussian Mixtures

No. of Gaussian	% Accuracy of Different Models		
	Word Model	Phoneme Model	Triphone Model
1	62	63	66
2	74	75	78
4	82	84	88
8	79	80	84
16	76	78	80

6. Conclusion

Recognition of human speech is a problem with many solutions, but still open because none of the current methods are fast and precise enough to be comparable with recognition capability of human beings. Although there are various methods but among all these methods, very little are used in real automatic speech recognition systems. Actually, most of the methods are still at the stage of experimental research and do not provide enough convincing results to be integrated. In this paper we have proposed a novel approach to develop speaker independent speech recognition system for Hindi language using MFCC and its extensions for feature extraction and HMM with Gaussian mixtures to generate acoustic models.

In our approach the numbers of mixture components are kept fixed while mean and variances may be varying from state to state. Experimental results have shown that only 4 Gaussian mixtures, applied for discriminative and margin based techniques, yield optimal performance in the context of small databases available for Indian languages which have been used to train the hidden Markov model. Results also illustrate that for small vocabulary up to 200, whole word model gives maximum accuracy, and beyond that triphone model must be used for better accuracy. At front- end if 3rd order MFCC combined with HLDA (i.e. extended MFCC) is used for feature extraction and at back-end if discriminative minimum

phone error (MPE) or margin based techniques are applied in ASR, accuracy can be improved by 5-6%.

References

[1] D. O'Shaughnessy, "Interacting with Computers by Voice-Automatic Speech Recognitions and Synthesis", (Invited Paper), Proceedings of the IEEE, Vol. 91, No. 9, 2003, pp. 1272-1305.

[2] Jun Cai, Ghazi Bouselmi, Yves Laprie, and Jean- Paul Haton, "Efficient Likelihood Evaluation and Dynamic Gaussian Selection for HMM-Based Speech Recognition", Computer Speech and Language, Vol.23, 2009, pp.147–164.

[3] F. Jelinek, Statistical Methods for Speech Recognition, MIT press, 1997.

[4] N. Goel, S.Thomas, M. Agarwal et al. "Approaches to Automatic Lexicon Learning with Limited Training Examples", Proc. of IEEE Conference on Acoustic Speech and Signal Processing, 2010.

[5] O. Duda and P. E. Hart Pattern, Classification and Scene Analysis, Wiley, New York, 1973.

[6] R. Haeb-Umbach and H. Ney, "Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition", in Proceedings of ICASSP, 1992, pp13-16.

[7] N. Kumar and A. G. Andreou, "Heteroscedastic Disciminant Analysis and Reduced Rank HMMs for Improved Speech Recognition", Speech Communication, Vol.26, 1998, pp. 283-297.

[8] S. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.28, 1980, pp.357-366.

[9] S. F. Boll, "Suppression of Acoustic Noise in Speech using Spectral Subtraction", IEEE Transaction of Acoustic, Speech and Signal Processing, Vol.27, No. 2, 1979, pp. 113-120.

[10] S. Molau, F. Hilger and H. Ney, "Feature Space Normalization in Adverse Acoustic Conditions", Proc. of ICASSP, 2003, pp. 656-659.

[11] H. Hermansky and N. Morgan, "RASTA Processing of Speech", IEEE Transaction on Speech and Audio Processing, Vol.2, No. 4, 1994, pp. 578-589.

[12] S. Young, "A Review of Large Vocabulary Continuous Speech Recognition", IEEE Signal Processing Mag., Vol.13, 1996, pp. 45-57.

[13] C.H. Lee, J. L. Gauvain, R. Pieraccini, and L. R. Rabiner, "Large Vocabulary Speech Recognition using Subword Units", Speech Communication, Vol.13, 1993, pp. 263-279.

[14] W. Reichl and W. Chou, "Robust Decision Tree State Tying for Continuous Speech Recognition", IEEE Transaction on Speech and Audio Processing, Vol. 8, No. 5, 2000, pp. 555-566.

[15] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, Vol. 77, No. 2, 1989, pp. 257-286.

[16] H. Ney, "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 32, No. 2, 1984, pp. 263-271.

[17] D. B. Paul, "Algorithms for an Optimal A* Search and Linearizing the Search in the Stack Decoder", Proc. ICASSP, Vol. 1, 1991, pp. 693-696.

[18] X. D. Huang , Y. Ariki and Jack, M.A., Hidden Markov Models for Speech Recognition, Edinburg University Press 1990.

[19] H. Jiang, Discriminative Training of HMM for Automatic Speech Recognition: A Survey, Computer Speech and Language, Vol. 24, 2010, 589-608.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithms", Journal of the Royal Statistical Society Series B, Vol.39, 1977, pp. 1-38.

[21] M. Gale and S. Young, "The Application of Hidden Markov Models in Speech Recognition", Foundations and Trends in Signal Processing, Vol.1, No. 3, 2007, pp. 195-304.

[22] Sim Khe Chai and M. J. F. Gales, "Minimum Phone Error Training of Precision Matrix Models", IEEE Transaction on Acoustic Speech and Signal Processing 2006.

[23] Xu Haihua, Povey Daniel, Zhu Jie and Wu Guanyong, "Minimum Hypothesis Phone Error as a Decoding Method for Speech Recognition", Interspeech (ISCA), 2009, pp. 76-79.

[24] C. Becchati and K. Ricotti Speech Recognition Theory and C++ Implementation, John Wiley & Sons 2004.

[25] M. J. F. Gales, K. M. Knill and S. J. Young, "State-Based Gaussian Selection in Large Vocabulary Continuous Speech Recognition using HMM's", IEEE Transactions on Speech and Audio Processing, Vol.7, No. 2, 1999, pp.152–161.

[26] F. Sha, and L. K. Saul, Large Margin Hidden Markov Models for Automatic Speech Recognition, in B. Scholkopf, J. Platt, and T. Hoffman (Eds.), Advances in Neural Information Processing Systems, 19 MIT Press, 2007, 1249–1256.

[27] C. M. Bishop, Pattern Recognition and Machine Learning (Springer, 2006).

[28] H. Jiang, and X. Li, "Incorporating Training Errors for Large Margin HMMs under Semi Definite Programming Framework", Proc. ICASSP, 2007.

[29] ELRA Catalogue, The EMILLE/CIIL Corpus, Catalogue Reference: ELRA-W0037, http://catalog.elra.info

[30] SPHINX: An Open Source at CMU: http://cmusphinx.sourceforge.net/html/cmusphinx.php.

[31] Hidden Markov Model Toolkit (HTK-3.4.1): http://htk.eng.cam.ac.uk.

[32] Julius: An Open Source for LVCSR Engine: http://julius.sourceforge.jp.

[33] M. Kumar, A. Verma, and N. Rajput, "A Large Vocabulary Speech Recognition System for Hindi," Journal of IBM Research, Vol. 48, 2004, pp. 703-715.

[34] R.K. Aggarwal and M. Dave, "Discriminative Techniques for Hindi Speech Recognition System", Communication in Computer and Information Science (Information Systems for Indian Languages), Springer-Verlag Berlin Heidelberg, Vol. 139, 2011, pp. 261-266.

Authors

R. K. Aggarwal received his M. Tech. degree in 2006 and pursuing PhD from National Institute of Technology, Kurukshetra, INDIA. Currently he is also working as an Associate Professor in the Department of Computer Engineering of the same Institute. He has published more than 24 research papers in various International/National journals and conferences and also worked as an active reviewer in many of them. He has delivered several invited talks, keynote addresses and also chaired the sessions in reputed conferences. His research interests include speech processing, soft computing, statistical modeling and science and spirituality. He is a life member of Computer Society of India (CSI) and Indian Society for Technical Education (ISTE). He has been involved in various academic, administrative and social affairs of many organizations having more than 20 years of experience in this field.

Mayank Dave obtained the M. Tech. degree in Computer Science and Technology from IIT Roorkee, INDIA in 1991 and PhD from the same institute in 2002. He is presently working as Associate Professor in Department of Computer Engineering at NIT Kurukshetra, INDIA with more than 19 years experience of academic and administrative affairs in the institute. He is presently heading Department of Computer Engineering and Department of Computer Applications. He has published approximately 60 research papers in various International / National Journals and Conferences. He has coordinated several projects and training programs for students and faculty. He has delivered number of expert lectures and keynote addresses on different topics. He has guided four PhDs and several M. Tech. dissertations. His research interests include Peer-to-Peer Computing, Pervasive Computing, Wireless Sensor Networks and Database Systems.