Feature Extraction and Selection for Speech Recognition

by

Bahman Mashhadi-Farahani

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical Engineering

Waterloo, Ontario, Canada, 1998

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

The major contributions of this thesis are: a new discriminative training algorithm, new discriminative feature selection and extraction algorithms, and a new image segmentation algorithm used for feature extraction from speech spectrogram.

In the first part of this thesis, a new misclassification measure and a discriminative training algorithm are proposed. The misclassification measure is a smooth representation of classification probability of error and can be made as close as possible to this probability by varying its parameters. The training algorithm indirectly minimizes the probability of error by minimizing the misclassification measure. A new discriminative training algorithm for speech segmentation based on another misclassification measure is also introduced.

In the second part of this thesis, a feature selection and a feature extraction algorithm are proposed. The proposed algorithms allow the dimensionality of feature space to be decreased, while trying to maintain a class separability measure. This measure is the misclassification measure of a classifier built in the higher dimensional space. The feature selection and extraction algorithms determine the maximum change in the misclassification measure (or indirectly the maximum loss in probability of correct classification) for the feature vectors presented in the lower dimensional space. The algorithms find the best subset of features and an optimum orthogonal linear mapping before applying feature selection that minimizes the maximum change in the misclassification measure.

In the third part of this thesis, several algorithms for feature extraction from speech spectrograms are proposed. Some of these algorithms first segment the spec-

trogram using a new self-organizing image segmentation algorithm. This algorithm segments the spectrogram into two classes of object and background, where pixels of each class have common characteristics. The algorithm iteratively minimizes a defined segmentation measure in the spectrogram image. Moreover, pixels with lower likelihood of belonging to object or background classes are adjusted less in each iteration, delaying their segmentation until more image information is available. The resulting features are the inputs to the proposed feature selection and extraction algorithms.

Some speaker independent isolated word speech recognition experiments are also carried out in this thesis which validate the proposed algorithms.

Acknowledgments

I wish to thank my supervisor, Professor Li Deng, for his support throughout this work. I would also like to thank the scholarship department of Iran ministry of culture and higher education for the financial support of this work.

To my fiance, Najmeh

and

my mother, Parvaneh

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The object of speech recognition is to provide a system that enables a human to communicate to a computer via speech signal. Fig. 1.1 shows a block diagram of a typical speech recognition system. The input speech signal is first passed through a feature selection and extraction block to reduce its redundant information for the purpose of speech recognition. Ideally, the representation of the signal should be minimal while containing all the sufficient information for recognition purpose. The reduced feature set is then passed through a classifier whose output is either a

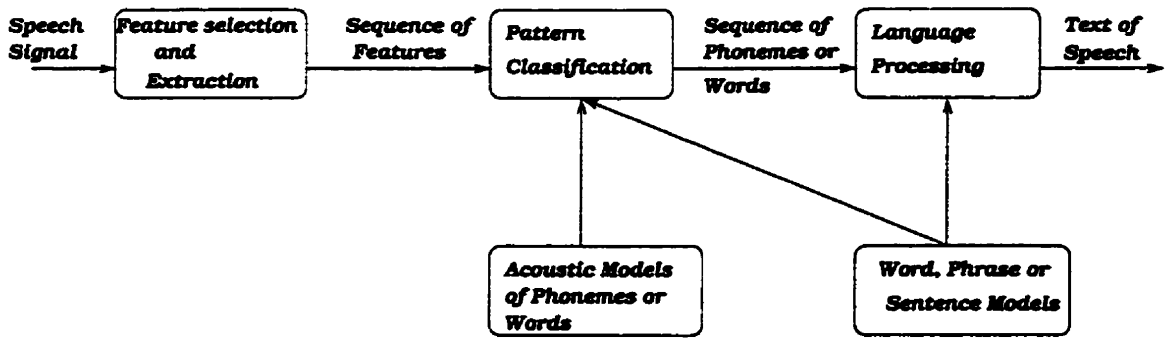


Figure 1.1: A speech recognition system

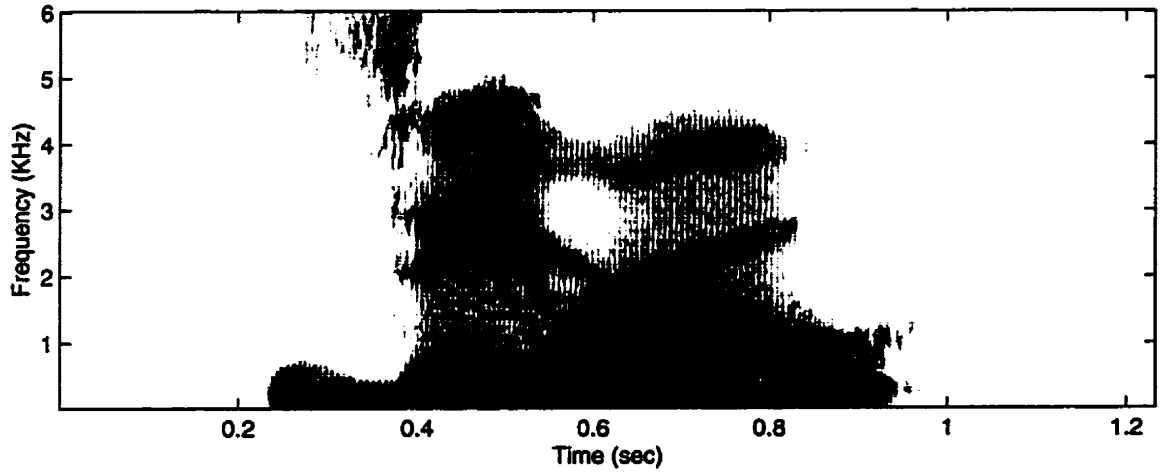
sequence of phonemes or words. Finally, at the language processing stage, sentences corresponding to the speech utterances are provided using word, phrase or sentence models.

The fundamental difficulty of speech recognition is the high variability of speech signals. Based on *a priori* knowledge, the speech waveform is a band-limited signal. The bandwidth of speech can be limited to $4 - 6$kHz without reducing its perceptual characteristics significantly. Even by only considering this limited a priori knowledge, a large number of samples is still required to represent speech signal.

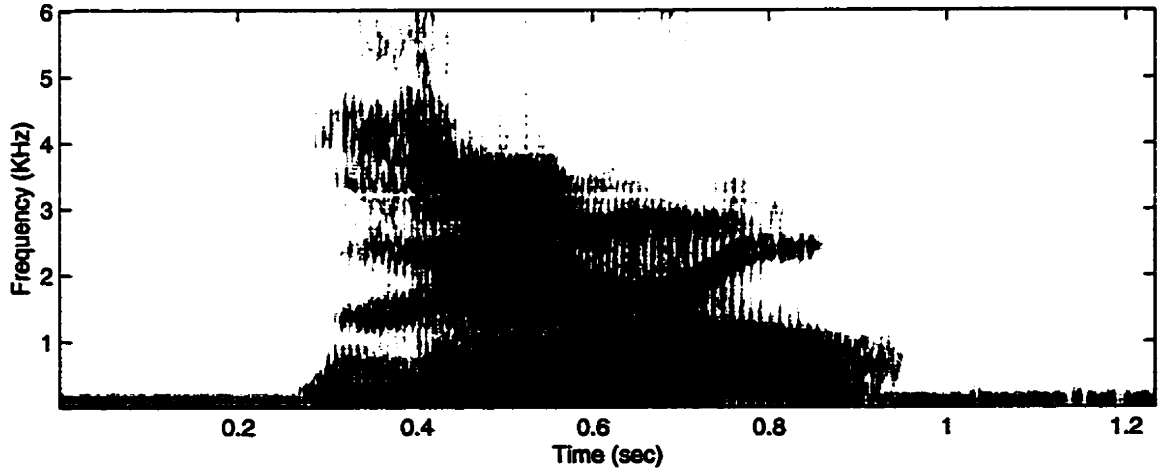
The speech signal is generated by the movement of the articulatory apparatus which modulates the air pressure to generate speech. Although speech signal has energy and information in the frequency domain up to several kHz, its pattern does not significantly change in intervals of more than 10ms because of the slow movement of articulators. The speech signal is also produced by humans in a way that is easily recognizable by the human recognition system.

The speech signal contains information that is not useful for recognition purposes such as the identity of the speaker, the speaker's emotional state, speaking rate, etc. Also, the characteristics of a given utterance can differ significantly for different occurrence of the utterance. Such differences are also recognizable for human listeners, but they are not useful for the purpose of recognition. Fig. 1.2 shows the spectrograms (short time Fourier transformations) [40] of the word zero produced by a female and a male speaker, respectively. There is an evident similarity in the appearance of the patterns. However, the patterns have distinctive differences as it may also be heard in the sound of these utterances.

In general, if the performance of classifiers is inadequate, new features should be added. Increasing the number of features requires an increase in the number of



(a)



(b)

Figure 1.2: Word zero uttered by (a) a female speaker (b) a male speaker

model parameters. Such an increase will reduce the performance of the classifier beyond a certain limit due to three factors: lack of enough training data, improper choice of models, and lack of an appropriate training algorithm. Feature selection and extraction can alleviate these design problems. Also, feature selection and extraction are appealing for real time speech recognition, as they can highly reduce the dimensionality of the feature space, and thereby reduce the computational costs. Moreover, each feature can usually be evaluated independently of others using parallel processors.

Ideally, the design of feature selection and extraction algorithms should be based on minimizing the probability of classification error. In this case, the design of feature selection and extraction cannot be separated from the design of the classifier. An ideal feature extractor is nothing other than an ideal classifier. The design difficulties of feature selection and extraction based on minimizing the probability of error are the same as the aforementioned difficulties for the design of classifiers. Therefore, a suboptimal solution for the design of feature selection or extraction is usually selected.

There are two practical ways to carry out the feature extraction and selection tasks. One is based on using human judgment to rely on his/her *a priori* knowledge of the classification problem to extract or select features. The other one is to define a class separability measure in lieu of probability of error. For feature extraction, a parametric structure can be defined (again based on *a priori* knowledge of the problem), and the parameters can be found to maximize the class separability measure. For feature selection in this case, a set of features should be selected that maximizes the class separability measure. Both of these approaches are exercised in this thesis.

The overall proposed recognition system is shown in Fig. 1.3. Here, a broad cat-

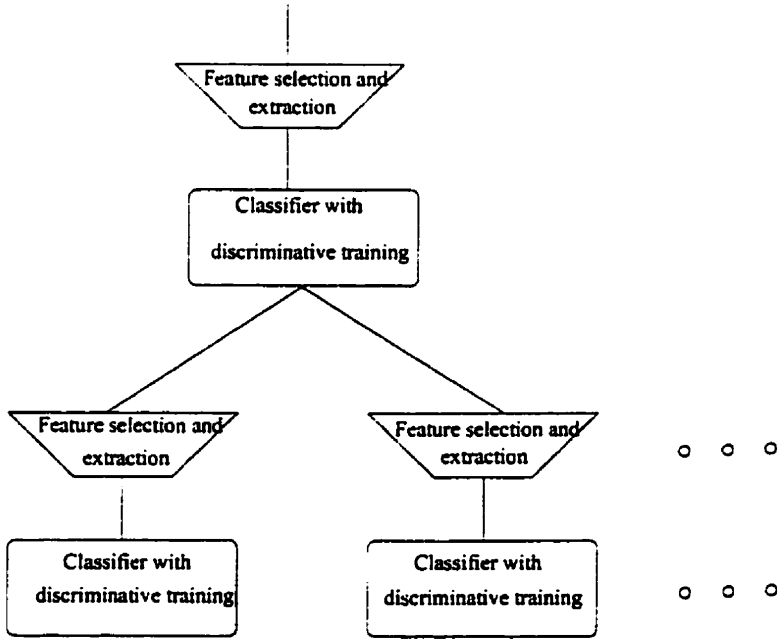


Figure 1.3: The overall classification system

egory of speech classes is first identified at the root of the tree, and then a detailed classification is carried out, depending on the results of the broad classification. In each category, the speech events are represented in a more discriminative nature using the feature selection and extraction blocks. Moreover, greater emphasis is placed to find classifier parameters that reduce the classification error using a discriminative training algorithm.

The organization of this thesis is as follows. In Chapter 2, a new misclassification measure and a new discriminative training algorithm along with a new form of classifier for speech recognition are introduced. The training algorithm finds the model parameters that reduce the misclassification measure. This algorithm has a lower complexity compared to other similar discriminative training algorithms such as the one introduced by Juang *et al.* [23]. The misclassification measure

is also used as a measure of performance for the feature selection and extraction algorithms presented later in Chapter 3. Also, in this Chapter, a new discriminative segmentation algorithm is introduced. In this algorithm, model parameters are trained for the purpose of segmentation to find the distinctive characteristics of speech utterances within a word. The segmentation algorithm can also use the feature extraction and selection algorithms. This greatly reduces the computational cost of segmentation and improves its performance.

In Chapter 3, the proposed feature selection and extraction algorithms are described. The feature selection and extraction techniques allow the dimensionality of data to be decreased, while trying to keep the discriminative information content of the remaining features for the purpose of classification task. In the proposed feature selection and extraction processes, a classifier is first designed in the higher dimensional space. The form of this classifier and its misclassification measure are adapted to make the feature extraction algorithm feasible. The feature selection and extraction algorithms can provide the maximum change in the misclassification measure if the classifier is built in a lower dimensional space for the extracted or selected features.

The preliminary selection of acoustic measurements is made based on the *a priori* knowledge of speech by extracting features from speech spectrograms. In practice, an expert spectrogram reader can classify speech utterances with a high accuracy rate. In Chapter 4, the information that is usually used in spectrogram reading experiments is measured using a new image segmentation technique. In this algorithm, the spectrogram is first segmented into two classes of object and background. The object and background classes consist of pixels having common characteristics such as regions that can be associated to formants. For each pixel, the *a posteriori* probabilities of belonging to each class are estimated based on the

knowledge about the shape and the intensity characteristics of each class. Using such knowledge, a new self-organizing image segmentation algorithm is introduced. This algorithm minimizes a defined segmentation measure in the image. The segmentation measure is a function of the *a posteriori* probability of pixels, and is minimum for any segmented image (having zero or one for the *a posteriori* probabilities). The algorithm iteratively adjusts the probabilities of pixels to reduce this measure. Pixels that are less likely to belong to object or background classes are adjusted less in each iteration, delaying their segmentation until more image information is available.

In Chapter 5, some speaker independent isolated word speech recognition experiments are provided. The experimental results validate the proposed algorithms.

Finally, in Chapter 6, summary and conclusion of the thesis is provided.

Chapter 2

Training criteria

2.1 Introduction

Bayes decision theory [17] [15] is a fundamental theory in classification. It is optimum in terms of minimizing the probability of classification error. In Bayes theory, class C_i for an input sample X is selected , if the *a posteriori* probability of that class, $P(C_i|X)$, is maximum among all the possible classes. Therefore, classifiers can be built by estimating the *a posteriori* probability of each class on the input domain. However, we have

$$P(C_i|X) = \frac{P(C_i)P(X|C_i)}{P(X)}. \tag{2.1}$$

Considering Bayes decision rule, we only need to estimate the *a priori* probability $P(C_i)$, and the conditional probability $P(X|C_i)$, of each class C_i since $P(X)$ is the same among all classes. Estimates of the conditional probabilities are usually provided by assuming that each class C_i is generated by a parametric probability model with the parameter set of λ_i. That means, we choose $P(X|\lambda_i)$ to model $P(X|C_i)$. In

this case. model parameters λ_i can be estimated based on the maximum likelihood criterion. where we maximize

$$\Pi_{u=1}^{U} P(X^u|\lambda_i). \tag{2.2}$$

over all the samples $X^1, X^2, \ldots, X^U$ of the training set of class C_i (assuming independence between training samples).

Considering the Bayes decision theory. only an accurate estimate of the *a posteriori* probabilities around the decision surface is required to build an optimum classifier. Therefore, more emphasis can be placed for estimating the parameters of the models to estimate the *a posteriori* probabilities accurately in more critical regions of the feature space (around the decision boundaries). Since the exact decision boundaries are not known, approximate boundaries are first estimated using estimates of *a posteriori* probabilities (based on maximum likelihood training), and then misclassification measures representing the degree of ambiguity about the classification of each input are defined and minimized. The misclassification measure can place more emphasis on regions closer to decision boundaries. This approach for finding the model parameters, is referred to as *discriminative training.* Compared to the maximum likelihood approach, the cost of training is usually higher. However, the model parameters are usually better estimated for the objective of minimizing the error rate.

When the assumptions about the parametric probability models are correct or classes are distinctively separated in the feature space, maximum likelihood training can perform well. On the other hand, if we have poor models for the underlying probability distributions and the classes are not distinctively separated, the discriminative training should be selected when the computational costs can be afforded.

In this chapter, different classifier design approaches are reviewed and a new approach for the design of classifiers is introduced with an emphasis on models suited for speech recognition. Also, a new segmentation algorithm for speech segmentation is introduced. This algorithm can find model parameters to distinctively characterize the differences within a speech unit such as word.

2.2 Minimum risk and error criteria

Suppose, for an input X, class C_i is selected. If X indeed belongs to class C_j and if a loss of $l(C_i|C_j, X)$ can be associated to such a decision, the expected loss associated with selecting class C_i can be written as [15]

$$R(C_i|X) = \sum_j l(C_i|C_j, X)P(C_j|X). \tag{2.3}$$

This expected loss is usually referred to as *conditional risk*. The overall *risk* can then be defined as

$$R = \int_X R(C(X)|X)P(X)dX, \tag{2.4}$$

where $C(X)$ is the class selected for a given observation X. It is clear from the above definition that for minimizing the overall risk, one should select the class with the minimum conditional risk for a given observation X.

If the conditional risk for a correct selection is assumed to be *one* and for an incorrect selection *zero*, the overall risk can be interpreted as the probability of error and the optimum selection is the class with the highest *a posteriori* probability, since the conditional risk can be written as

$$R(C_i|X) = \sum_j l(C_i|C_j, X)P(C_j|X) = \sum_{j,j\neq i} P(C_j|X) = 1 - P(C_i|X). \tag{2.5}$$

To build up a classifier, one may estimate the conditional risks or the *a posteriori* probabilities of different classes depending upon having minimum risk or minimum error rate as the criterion, respectively.

Another common approach to build a classifier is to use discriminant functions in lieu of the *a posteriori* probabilities. In this case, a discriminant function $g_i(X;\lambda_i)$, with free parameters λ_i, is associated to each class C_i. The class C_i is selected (in a similar way as in Bayes classifiers) when it has the highest $g_i(X;\lambda_i)$ among all classes, *i.e.*, when

$$g_i(X;\lambda_i) > g_j(X;\lambda_j) \quad \forall j \neq i. \tag{2.6}$$

The optimum Bayes classifier can be achieved by selecting $g_i(X;\lambda_i) = -P(C_i|X)$. However, the choice of optimum discriminant functions is not unique. For example, they can be multiplied by a positive constant, added by a constant or even replaced by $f(g_i(X;\lambda))$, where $f(\cdot)$ is a monotonic function, without changing the probability of error. Considering Eq. (2.1), the following discriminant function also results in the optimum classifier for minimum error rate criterion:

$$g_i(X;\lambda_i) = logP(X|C_i) + logP(C_i). \tag{2.7}$$

The discriminant functions do not necessarily need to be monotonic functions of a posteriori probabilities to result in optimum classification. The only requirement is that the inequality (2.6) results in the same classification as Bayes inequality for every point in space. The approach that is usually selected is to make the discriminant functions be close to monotonic functions of the a posteriori probabilities only in the regions close to estimates of decision boundaries.

There are different ways to estimate the parameters of the discriminant functions. Assume the discriminant functions are limited between zero and one. One

criterion is based on minimizing a *mean square error* function over the feature space defined as

$$E_m = \sum_{k=1}^{M} \int_X P(X, C_k) \sum_{i=1}^{M} [\delta(C_k = C_i) - g_i(X; \lambda_i)]^2 dX, \tag{2.8}$$

where $\delta(s)$ is *one* if the statement "s" is true and *zero* otherwise, $P(X, C_k)$ is the joint probability of class C_k and input X, and M is the number of classes. It can be shown that the above error function can also be written as (see [36] for more detail)

$$E_m = \sum_i \int_X P(X)[P(C_i|X) - g_i(X; \lambda_i)]^2 + P(C_i|X)[1 - P(C_i|X)]dX. \tag{2.9}$$

As can be seen from the above equation, the global minimum of the error is achieved when

$$g_i(X; \lambda_i) = P(C_i|X). \tag{2.10}$$

That means, if the parametric form of the discriminant functions are consistent with the true *a posteriori* probability functions of different classes, this criterion can result in the minimum error rate. In this training algorithm, the error may decrease during the course of training, but the probability of error may increase at the same time. Note that only the global minimum of the error function is an optimum solution. This is an important drawback as the training algorithm is usually trapped in the local minimum of the error function.

The above shortcoming of training algorithm is addressed by *Minimum Classification Error* algorithm (MCE) proposed by Juang *et al.*, [23], [26], [46]. In this algorithm, a misclassification measure is first defined for any input sample X to measure the degree of performance of the classifier. A common form of misclassification measure is defined as

$$h_i(X; \lambda) = -g_i(X; \lambda_i) + [\frac{1}{M-1} \sum_{j, j \neq i} g_j(X; \lambda_j)^\eta]^{1/\eta}, \tag{2.11}$$

where $g_i(\cdot)$ is the discriminant function of the correct class. λ is the set of model parameters and η is a positive number. A cost function (a monotonic function of $h(\cdot)$) is defined as $\gamma(\xi h_i(X;\lambda))$, where ξ is a positive number and $\gamma(x)$ is a sigmoid function defined as

$$\gamma(x) = \frac{1}{1+e^{-x}}. \tag{2.12}$$

This cost function is defined to place less emphasis on regions having high misclassification measure, that is, regions having less degree of ambiguity about their class membership. Moreover, it can make the training algorithm feasible as the misclassification measure is limited for any input sample. We can minimize the expected cost in the domain X in a similar way as risk theory to find the model parameters. That is to minimize

$$E_c = \sum_i \int_X P(X, C_i)\delta(C(X) = C_i)\gamma(\xi h_i(X;\lambda))dX. \tag{2.13}$$

However, the classification error rate can be written as

$$E = \sum_i \int_X P(X, C_i)\delta(C(X) = C_i)\delta(P(C_i|X) \neq \max_j P(C_j|X))dX. \tag{2.14}$$

Comparing Eq. (2.13) and Eq. (2.14), we can see that the difference is only in the second $\delta(\cdot)$ term of Eq. (2.14) and the $\gamma(\cdot)$ term of Eq. (2.13). One can make these two terms as close as possible to each other by varying the value of parameters ξ and η [23]. By increasing η, we approach to $\max_j g_j(X;\lambda_j)$ for the second term in Eq. (2.11), and then by increasing ξ, we approach to *one* for a wrong classification and to *zero* for a correct classification. Therefore, the minimization of the overall cost function (Eq. (2.13)) is consistent with minimization of error rate, as the defined error function is indeed a smooth representation of the error rate.

The minimization of the defined error function is usually carried out using a steepest descent algorithm which finds a local minimum of the error function. As

was discussed earlier, the defined cost function places a greater emphasis on more accurately estimating the *a posteriori* probabilities (or their monotonic functions) around the decision boundaries. This emphasis can only be meaningful if a good approximation of initial model parameters or decision boundaries exists. If this is not the case, the chance of being trapped in an undesirable local minimum may increase since a wrong cost has been initially associated with regions of space. The initial approximation of model parameters is usually provided by initializing the discriminant functions $g_i(X;\lambda_i)$'s, by $\log(P(X|\lambda_i))$ (assuming equal *a priori* probability for different classes), where λ_i's have already been estimated based on maximum likelihood criterion.

2.3 A new misclassification measure

As mentioned before, the essence of discriminative training can be interpreted as properly modeling the *a posteriori* probability functions (or their monotonic functions) in the neighborhood of decision surface. The maximum likelihood criterion can provide estimates of the decision boundaries. Therefore, one can use such boundaries, and place a proper emphasis on different regions of space. In the MCE training algorithm proposed in [23] , the parametric form of discriminant functions of each class is defined as $\log(P(C_i|X))$, and its parameters are initialized using maximum likelihood training algorithm. Here, the parametric form of discriminant functions is defined as a function of the parametric form of the *a posteriori* probabilities of all classes. The form of discriminant functions is defined to be

$$g_i(X;\lambda) = \frac{1}{M-1} \sum_{j=1, j\neq i}^{M} \gamma(\xi_2 \pi_{ij}(X;\lambda)), \tag{2.15}$$

where M is the number of classes, ξ_2 is a positive constant, and $\pi_{ij}(X;\lambda)$ is defined as

$$\pi_{ij}(X;\lambda) = \log P(C_i|X) - \log P(C_j|X), \tag{2.16}$$

and $\gamma(\cdot)$ is a sigmoid function and is defined as in Eq. (2.12). Note that the use of $\gamma(\cdot)$ function place less emphasis on regions of space that are far from the decision boundaries. If we assume equal *a priori* probabilities, we have

$$\pi_{ij}(X;\lambda) = \log P(X|\lambda_i) - \log P(X|\lambda_j). \tag{2.17}$$

If $P(C_i|X)$ is maximum among all classes, then $g_i(X;\lambda)$ is maximum for all classes and vice versa, since each term of Eq. (2.15) is greater than a corresponding term of the discriminant functions of other classes, i.e., if $P(C_i|X)$ is maximum among all classes, then

$$\gamma(\xi_2(\log P(C_i|X) - \log P(C_k|X))) > \gamma(\xi_2(\log P(C_j|X) - \log P(C_k|X))), \tag{2.18}$$

where C_k is a class other than C_i and C_j. Also, we have

$$\gamma(\xi_2(\log P(C_i|X) - \log P(C_j|X))) > \gamma(\xi_2(\log P(C_j|X) - \log P(C_i|X))). \tag{2.19}$$

Therefore, we have

$$g_i(X;\lambda) > g_j(X;\lambda). \tag{2.20}$$

Clearly, for the minimum error rate criterion, one should select the class having the highest $g_i(\cdot)$ or $P(C_i|X)$. The motivation behind the selection of this form of discriminant functions will be more clarified when state models and feature extraction and selection algorithms are presented in this thesis.

In a similar way as in the MCE algorithm, a misclassification measure $h_i(X;\lambda)$ for each class i can be defined as

$$h_i(X;\lambda) = -g_i(X;\lambda) + \theta_M, \tag{2.21}$$

where

$$\theta_M = \frac{1}{2} + \frac{M-2}{2(M-1)}. \tag{2.22}$$

A cost function l_i can also be defined as

$$l_i = \gamma(\xi_3 h_i). \tag{2.23}$$

The above cost function is also a smooth representation of probability of error. If $\xi_3 >> 0$ and $\xi_2 >> 0$, for a correct classification, l_i is close to zero and for a wrong classification, l_i is close to one. If a correct classification is made, g_i is maximum and is close to 1. That implies h_i is negative since $1/2+(M-2)/2(M-1) < 1$. Negative h_i means l_i approaches to zero as ξ_3 becomes large. If a wrong classification is made, $g_k(X;\lambda)$ of a class k other than class i is maximum. It implies that $P(C_k|X)$ is maximum, and it can be concluded that

$$0 \leq g_i(X;\lambda) < \frac{M-2}{M-1}, \tag{2.24}$$

and from here h_i becomes positive and therefore l_i approaches to one if ξ_3 is large.

Now, consider the overall loss

$$E_n = \sum_i \int_X P(X, C_i)\delta(C(X) = C_i) l_i(X;\lambda) dX, \tag{2.25}$$

Comparing the above cost function and the probability of error as in Eq. (2.14), it can be concluded that the overall cost can be made as close as possible to the probability of error, and minimization of overall cost is consistent with minimization of error probability.

Since the distribution of training data is usually unknown, the overall misclassification measure defined in Eq. (2.25) is estimated by an empirical average cost. Consider the set of input samples is $X^1, X^2, \cdots, X^u, \cdots, X^U$. We can minimize

$$S_r = \sum_u \sum_i \delta(C(X^u) = C_i) l_i(X^u;\lambda), \tag{2.26}$$

where $C(X^u)$ gives the class to which the input sample X^u belongs. The model parameters can also be found to minimize the overall misclassification measure over the space, i.e., the following cost function can also be selected

$$l_i(X^u;\lambda) = -g_i(X^u,\lambda), \tag{2.27}$$

and the following overall cost should be minimized

$$S_m = \sum_u \sum_i \delta(C(X^u) = C_i) g_i(X^u;\lambda). \tag{2.28}$$

One common approach for minimizing the overall misclassification is to use the steepest gradient descent algorithm. According to this algorithm, the parameters are adjusted in proportion to the negative gradient of the misclassification measure. The model parameters can be updated using

$$\Delta\lambda_i^{\zeta+1} = -\alpha \sum_u \sum_i \delta(C(X^u) = C_i) \frac{\partial l_i(X;\lambda)}{\partial \lambda_i}|_{\lambda_i=\lambda_i^\zeta}, \tag{2.29}$$

where α is the learning rate and ζ is the iteration number.

In the following, we try to partly explain the motivation behind the selection of the form of discriminant functions. Consider two classes that can be modeled by Gaussian probability distribution with tied diagonal covariance matrix ($\Sigma_i = \Sigma_j = \Sigma$) and the mean M_i and M_j for class i and j respectively. Also, assume equal *a priori* probabilities for the two classes. Therefore, for a point $X^0 = [x_1^0, \ldots, x_n^0]$, we have

$$\begin{aligned} P(X^0|\lambda_i) &= N(X^0, M_i, \Sigma_i) \\ &= \frac{1}{(2\pi)^{n/2}(\Pi_{k=1}^n \sigma_k)^{1/2}} \exp(-\frac{1}{2}\sum_{k=1}^n \frac{(x_k - m_{ik})^2}{\sigma_k^2}), \end{aligned} \tag{2.30}$$

where σ_k is the kth diagonal element of Σ, m_{ik} is the kth element of M_i, and x_k is the kth element of X^0.

The decision boundary between the two classes is a hyper-plane. Fig. 2.1 shows such a boundary. In this figure, d_0 is the distance of an input sample X^0 from the

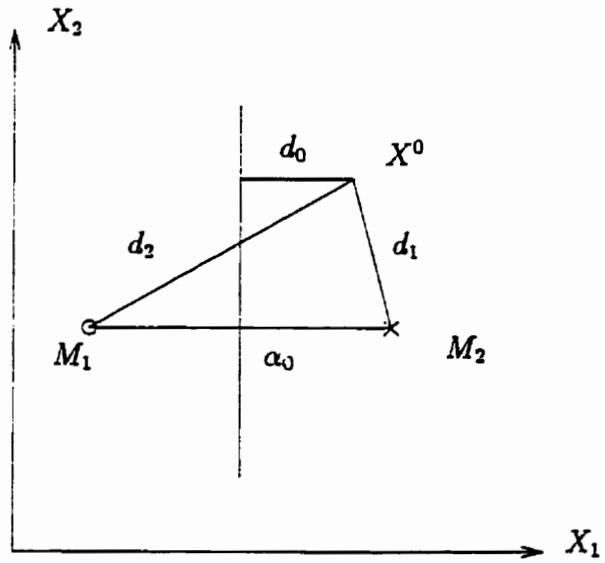


Figure 2.1: Two typical classes and their corresponding hyper-plane decision boundary

decision boundary, d_1 and d_2 are the distances of the input sample from the mean of different classes M_1 and M_2 respectively, and α_0 is the distance of the two means. The decision boundary has the following equation (see Fig. 2.1):

$$\pi_{ij}(X^0;\lambda) = 0, \tag{2.31}$$

where

$$\begin{aligned} \pi_{ij}(X^0;\lambda) &= \log P(X^0|\lambda_i) - \log P(X^0|\lambda_j) \\ &= -\tfrac{1}{2}\textstyle\sum_{k=1}^{n} \frac{(x_k-m_{ik})^2-(x_k-m_{jk})^2}{\sigma_k^2} \\ &= \textstyle\sum_{k=1}^{n} \frac{m_{ik}-m_{jk}}{\sigma_k^2}\left[x_k - \frac{m_{jk}+m_{ik}}{2}\right], \end{aligned} \tag{2.32}$$

and the associated misclassification measure would be

$$h_i(X^0;\lambda) = -\gamma(\xi_2\pi_{ij}(X^0;\lambda)) + \frac{1}{2}. \tag{2.33}$$

However, we can easily see that

$$\pi_{ij}(X^0;\lambda) = \alpha_0 d_0 q_0, \tag{2.34}$$

Figure 2.2: State model

where α_0 is the distance between the mean of the two classes normalized by covariance values, d_0 is the distance of X^0 from the decision boundary, and q_0 is 1 if X^0 belongs to the correct class and -1 otherwise. These values, d_0 and α_0, can be calculated as follows:

$$d_0 = \frac{\left|\sum_{k=1}^{n} \frac{m_{ik}-m_{jk}}{\sigma_k^2}(x_k^0 - \frac{m_{ik}+m_{jk}}{2})\right|}{[\sum_{k=1}^{n}(\frac{m_{ik}-m_{jk}}{\sigma_k^2})^2]^{1/2}}, \tag{2.35}$$

and

$$\alpha_0 = [\sum_{k=1}^{n}(\frac{m_{ik}-m_{jk}}{\sigma_k^2})^2]^{1/2}. \tag{2.36}$$

Note that the direction perpendicular to the decision hyper-plane is only important for classification. d_0 and α_0 (and consequently $\pi_{ij}(X^0;\lambda)$) and the misclassification measure h_i remain the same if this direction is preserved. In practice, decision boundaries can be estimated by several hyper-planes. As it will be seen in the next chapter, the directions perpendicular to these hyper-planes can eventually define the feature space for classification.

2.3.1 State models and discriminative training

The speech signal is not a memoryless signal. This is mainly a result of the articulatory constraints imposed in generating speech sounds and the phonetic constraints imposed by language models. Hidden Markov Models (HMM's) [42], [41], [19] are simple yet efficient in modeling this characteristic of speech signal.

Consider a Markov model (see Fig. 2.2) with a state transition probability matrix $A = [a_{sr}]$, where a_{sr} is the probability of making a transition from state s to state r. Also, consider the initial state probability matrix $\Pi = [\pi_s]$, where π_s is the initial state probability of state s. Let $b_s(X_t)$ be the probability density function governing the observation X_t (each observation is actually a feature vector extracted from speech frame) produced by state s. Let $q_m = s_1, \ldots, s_T$ be a possible sequence of states. The density function of an input X produced by a Markov model can be defined as the following equation

$$P(X|\lambda_i) = \sum_{\text{all}_{q_m}} \pi_{s_1} \Pi_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(X_t), \tag{2.37}$$

or

$$P(X|\lambda_i) = \max_{\text{all}_{q_m}} \pi_{s_1} \Pi_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(X_t). \tag{2.38}$$

The optimum state durations for Eq. (2.38) can be found using Viterbi algorithm as described in [42].

To estimate the model parameters λ, the model likelihood score should be maximized. This can be achieved based on the Baum-Welch algorithm [42] or the segmental k-means algorithm [22]. The second approach uses Eq. (2.38) for the likelihood score. This algorithm is more attractive as it has a lower computational cost while having similar performances as Baum-Welch algorithm. This algorithm has two steps: the segmentation step and estimation step. In the segmentation step, the state durations that maximize the overall likelihood are found. In the estimation step, the model parameters (λ) are estimated using statistical characteristics of the cluster sets found in segmentation. These two steps of segmentation and estimation are repeated for several iterations until the change in the average likelihood probabilities of input samples in the training set becomes small. For more details of the algorithm, the reader is referred to [22]. Several algorithms

have also been proposed in literature for finding the parameters of hidden Markov models based on discriminant training criterion. In [1], Bahl *et al.* use maximum mutual information as the criteria. Juang *et al.* [38], [23], [21], [4], [26] minimize a misclassification measure to find the model parameters.

Another practical issue in HMM's is that the initial state probabilities and the transitional state probabilities are not important factors in classification decisions or in segmentation decisions of the Viterbi algorithm. This is due to lack of discriminative capability of such parameters [24]. The transitional probabilities also inherently model the duration of stay in each state of the model by

$$P_s(d) = (a_{ss})^{d-1}(1 - a_{ss}). \tag{2.39}$$

This assumption about the state durations imposed by transitional probabilities is inappropriate for almost any speech event [24].

Semi-Markov models [43] try to address this shortcoming of Markov models by introducing a distribution model for probability of stay in each state. The models that are adopted for discriminant functions of speech classes are based on similar models to semi-Markov models with Gaussian state distributions. In the following, these models are introduced.

Consider a semi-Markov model with S states (Fig. 2.2), where the states take the duration sequence $d = d_1, ..., d_S$, respectively, that is the model stays in state 1 for d_1 frames, etc. Let $X = X_1, ..., X_T$ be the observed frames for the modeled token having total length $T = d_1 + ... + d_S$. The free parameters used to train the model are collected as $\lambda = (M, \Sigma)$, where M is the mean and Σ is covariance matrix for the continuous output densities associated with the states.

Let $b_s(X_t)$ be a Gaussian probability density governing the observation X_t produced while the model is in state s. Observations are modeled as conditionally

independent and identically distributed given the state. Duration of stay in each state is modeled by a uniform distribution having a minimum duration of d^{min} and a maximum of d^{max}. Let $D^m = \{d_1^m, \ldots, d_t^m, \ldots, d_s^m\}$ be a possible set for duration of stay in state 1 to s, where d_t^m be the duration of stay in state t. The density of observation can be expressed as the following equation:

$$P(X) = \sum_{\text{all} D^m} [b_1(X_1) \cdots b_1(X_{d_1^m})] \cdots [b_S(X_{d_1^m + \cdots + d_{S-1}^m + 1}) \cdots b_S(X_T)], \qquad (2.40)$$

or

$$P(X) = \max_{\text{max all} D^m} [b_1(X_1) \cdots b_1(X_{d_1^m})] \cdots [b_S(X_{d_1^m + \cdots + d_{S-1}^m + 1}) \cdots b_S(X_T)]. \qquad (2.41)$$

Viterbi beam algorithm [34] is a practical way to calculate Eq. (2.41). In this algorithm, only a maximum of N best path up to each frame of time is considered. N is referred to beam size. In HMM's, calculating the best state sequence has a problem size of $\mathcal{O}(S\ X\ T)$, where S is the number of states in the model and T is the duration of input sample. In the Viterbi beam algorithm used for calculation of Eq. (2.41), the problem size is of the order of $\mathcal{O}(\bar{S}_b\ X\ T)$, where $\bar{S}_b$ is the average beam size. This increased computational cost is a disadvantage of semi-Markov models over HMM's. However, as it will be seen in the next section, model parameters can be trained to properly model the differences within states of the model. As a result, the maximum size of beam can be reduced in this algorithm.

In the proposed algorithm, the state durations are found using a discriminant segmentation algorithm described in the following section. Therefore, the model parameters only need to be estimated using the statistical characteristics of the segmented input tokens. After the models are trained, they are used to initialize the discriminant functions before applying the discriminative training. The

discriminant function of each class is defined as

$$g_i(X;\lambda) = \sum_{j,j\neq i} \gamma(\xi_2(log(P(X|\lambda_i)) - log(P(X|\lambda_j)))), \tag{2.42}$$

where ξ_2 is a positive constant. It is also known that

$$log(P(X|\lambda_i)) = \frac{1}{T}\sum_t N(X_t, M_i^t, \Sigma_i^t), \tag{2.43}$$

and

$$log(P(X|\lambda_j)) = \frac{1}{T}\sum_t N(X_t, M_j^t, \Sigma_j^t), \tag{2.44}$$

where $N(\cdot)$ is defined as in Eq. (2.30), and M_i^t and Σ_i^t are the mean and covariance of the state selected at time t for input vector X_t in the model of class i.

Here, diagonal shared covariance matrices are used and the following approximation is applied

$$log(P(X|\lambda_i)) - log(P(X|\lambda_j))) = \frac{1}{T}\sum_{t=1}^{T} \gamma(\xi_1 \pi_{ij}^t) - 0.5, \tag{2.45}$$

where π_{ij}^t is

$$\pi_{ij}^t = \sum_{k=1}^{n} \frac{m_{ik}^t - m_{jk}^t}{{\sigma_{ijk}^t}^2}[x_k^t - \frac{m_{jk}^t + m_{ik}^t}{2}], \tag{2.46}$$

for the state parameters selected for class i and j at time frame t (σ_{ijk}^t is the kth element of the shared covariance between state i and j). Please note that if ξ_1 is small enough, we are in the linear portion of the sigmoid function, and therefore by applying the nonlinearity imposed by $\gamma(\cdot)$, we will not have a different classification result other than that of maximum likelihood. In practice, usually the performance is slightly better, as the nonlinearity of sigmoid function can limit the effect of an input vector X_t in the overall decision making. Although initially the class with maximum $P(X|\lambda_i)$ results in maximum $g_i(X;\lambda)$, this may not be the case after training of model parameters using discriminant training algorithm.

The empirical overall cost function that should be minimized for a set of input samples $X^1, \ldots . X^U$ is defined as follows:

$$E = \sum_u \sum_i \delta(C_i = C(X^u)) \gamma(\frac{-\xi_3}{M-1} \sum_{j,j \neq i} \gamma(\frac{\xi_2}{T_u} \sum_{t=1}^{T_u} \gamma(\xi_1 \pi_{ij}^t) - 0.5) + \theta_M). \quad (2.47)$$

Using steepest descent algorithm, the adjustment of mean parameters is as follows:

$$\Delta m_i^t = -\frac{\alpha \xi_1 \xi_2 \xi_3}{M-1} \sum_u \sum_i \delta(C_i = C(X^u)) y(1-y) \sum_j \frac{\omega(1-\omega)}{T_u} \sum_{t=1}^{T_u} \nu(1-\nu) \frac{\partial \pi_{ij}^t}{\partial m_j^t}, \quad (2.48)$$

where

$$y = \gamma(\frac{-\xi_3}{M-1} \sum_{j,j \neq i} \gamma(\frac{\xi_2}{T_u} \sum_{t=1}^{T_u} \gamma(\xi_1 \pi_{ij}^t) - 0.5) + \theta_M), \quad (2.49)$$

and

$$\omega = \gamma(\frac{\xi_2}{T_u} \sum_{t=1}^{T_u} \gamma(\xi_2 \pi_{ij}^t) - 0.5), \quad (2.50)$$

and

$$\nu = \gamma(\xi_1 \pi_{ij}^t), \quad (2.51)$$

and

$$\frac{\partial \pi_{ij}(X^0; \lambda_i)}{\partial m_i} = \frac{1}{\sigma_k^2}(m_{ik} - x_k). \quad (2.52)$$

In the experiments reported in this thesis, the mean parameters are only adjusted. As can be seen from the above equations, the training algorithm is very similar to the training algorithm of LVQ2.1 [29] [28] [30]. Here, the terms $y(1-y)$, $\omega(1-\omega)$. $\nu(1-\nu)$ play the role of the window region in LVQ2.1 algorithm. Compared to MCE algorithm [21], the cost of calculating the derivative of the overall cost function is lower resulting in faster training time.

2.4 Discriminant segmentation

For traditional speech recognition systems using Hidden Markov Models, the optimum segmentation is to find the sequence of states that maximizes the overall likelihood of an input sample produced by the Markov model. This strategy has some advantages and disadvantages. The main advantage is that the same model is used for segmentation and classification. The disadvantage of this strategy appears when we study the differences of more confusing classes of speech. Such classes of speech often differ in a limited number of features within a limited number of frames. These frames usually are not assigned to a separate state using the maximum likelihood segmentation, as they cannot produce a significant change in the overall likelihood. As a result, the statistical characteristics of such regions do not change the statistics of their associated states significantly. As a result, the overall classification is not affected by these regions.

Another problem appears when model parameters are trained using discriminant training. In this case, the model parameters are trained to minimize the defined overall cost functions. Therefore, such parameters are not valid to estimate the likelihood models anymore. Since in the Viterbi segmentation, the optimum choice of state durations that maximizes the overall likelihood are selected, the newly trained parameters cannot be used to do the segmentation task. Therefore, the task of segmentation and classification should be carried out by different models.

If we carefully examine the segmentation task, we notice that it is nothing more than a classification problem. In segmentation, we have to classify or select the best path from a set of possible paths. The maximum likelihood segmentation selects the path that has the highest likelihood. Similarly, we can associate discriminant functions to each path and select the path having the highest discriminant function

value. Also, similar to discriminant training algorithm described in the previous sections, misclassification measures can be defined and discriminant training algorithm can be applied for training of segmentation models. Discriminant training can only focus on the acoustic differences within a class which is a reduced classification problem. As it will be seen in the next chapter, a feature extraction algorithm can also be applied for segmentation purpose that can greatly reduce the computational cost of discriminative segmentation.

Here, we define a discriminant function for any given path. The path is selected such that its discriminant function is maximum. To initialize model parameters, each input sample is first hand segmented and the parameters of the state models are initialized using the statistical characteristics of their associated segmented regions.

Consider a path $q^u = s_1^u, \ldots, s_t^u, \ldots, s_T^u$, where s_t^u is the state that input X^u has stayed in at time frame t. The discriminant function of this path is defined as

$$g_u(X^u; q^u) = \frac{1}{T_u(J-1)} \sum_{t=1}^{T_u} \sum_{j=1, j\neq i}^{J} \gamma(\xi_1 \pi^t_{s_t^u s_j}) - 0.5, \tag{2.53}$$

where J is the number of states in the model and,

$$\pi^t_{s_t^u s_j} = log(P(X_t^u | s_t^u)) - log(P(X_t^u | s_j)). \tag{2.54}$$

If $q^* = \{s_1^*, \ldots, s_t^*, \ldots, s_T^*\}$ is a path that maximizes $P(X^u|\lambda)$ over all possible choices of q, and if ξ_1 is small enough that we are in the linear portion of the sigmoid function, then q^* also maximizes $g(X^u; q)$ over all possible choices of q. This can be proved as follows: Let assume that $q^m = \{s_1^m, \ldots, s_t^m, \ldots, s_T^m\}$ is a path other than q^*, then we have

$$\sum_{t=1}^{T_u} log(P(X_t^u | s_t^*)) > \sum_{t=1}^{T_u} log(P(X_t^u | s_t^m)), \tag{2.55}$$

then, we have

$$\sum_{t=1}^{T_u}(Jlog(P(X_t^u|s_t^*))-\sum_{j=1}^{J}log(P(X_t^u|s_j))) > \sum_{t=1}^{T_u}(Jlog(P(X_t^u|s_t^m))-\sum_{j=1}^{J}log(P(X_t^u|s_j))), \tag{2.56}$$

and we have

$$\sum_{t=1}^{T_u}\sum_{j=1,s_j\neq s_t^*}^{J} log(P(X_t^u|s_t^*)) - log(P(X_t^u|s_j)) > \sum_{t=1}^{T_u}\sum_{j=1,s_j\neq s_t^m}^{J} log(P(X_t^u|s_t^m)) - log(P(X_t^u|s_j)), \tag{2.57}$$

and by considering that $\gamma(\cdot)$ is a monotonic function and ξ_1 is small enough, we have

$$g(X^u;q^*) > g(X^u;q^m). \tag{2.58}$$

Here, for the training of model parameters, we minimize the misclassification measure of the best paths in the training set. If we have a set of input training tokens $X^1,\ldots,X^U$ with their corresponding best paths $q^1,\ldots,q^U$, we minimize the following cost function

$$G = -\sum_{u=1}^{U} g(X^u;q^u). \tag{2.59}$$

Note that the optimum state durations should be found using the Viterbi beam search to reduce the computational cost. However, due to discriminant ability of the model, small sizes of beam results in almost perfect segmentation. In my experiments, the maximum beam size was selected to be 50. Also, note that the computational costs can be reduced using the feature selection and extraction algorithms proposed in the following chapter.

2.5 Summary

A new discriminant training algorithm along with a new form of state models for the design of speech classifiers was introduced in this chapter. The training algorithm first initializes the model parameters using statistical characteristics of the training set. Then, model parameters are adjusted using a discriminative training algorithm by minimizing a defined misclassification measure. The misclassification measure is a smooth version of probability of error. Therefore, the probability of error is indirectly minimized. A new discriminant segmentation algorithm was also introduced. Discriminant functions were associated to each possible path of the state model and the model parameters were trained to emphasis on the differences of states within speech units such as word.

Chapter 3

Feature selection and extraction

3.1 Introduction

Feature extraction is a preprocessing mechanism to reduce the dimensionality of data by mapping the original measurements into more discriminative features. The proper choice of the mapping functions depends on *a priori* knowledge of data and in practice, usually heuristic techniques are used for this selection. A completely optimal feature extractor can never be anything but an optimal classifier. In other words, if the minimum error rate criterion is our objective, the design of mapping functions cannot be separated from the design of the classifier. Examples of such systems can be seen in literature in [4], [45], [35], [31], [5], [33]. It is usually hard to achieve the objective of minimum error rate criterion, mainly due to high dimensionality of feature space. Therefore, class separability measures are defined and optimized instead of error rate criterion to measure the discriminative power of new feature set.

Generally, if the performance of a classifier is inadequate, we would like to add

new features, in particular those features that are more effective for the classification of more confusing classes. The performance of the classifier should increase if an optimum classifier can be built up in the new feature space. The worse thing is that the optimum classifier ignores the new features. In practice, beyond a certain point, inclusion of new features leads to worse performance of classifier. The basic source of the problem can be traced to the fact that we usually use a parametric classifier to model the optimum classifier. Such a classifier requires more free parameters to model the higher dimensional input space. When the models are estimating a posteriori probabilities (at least around the decision boundaries), the increase in the number of model parameters usually results in an increase in the mismatch of models and true a posteriori probabilities. When the number of parameters is increased, the chance of finding the optimum set of parameters is also decreased. Moreover, we require a higher number of sample data to estimate the increased number of parameters, resulting in another possible source of error.

Feature selection is an approach to alleviate these design problems. Here, we select a subset of k features from n possible candidate features. In practice, a linear transformation of the feature space before or after feature selection can usually improve the performance of the classifier considerably. This method is an example of *feature extraction*. The optimum solutions (reducing the error rate) for feature selection or extraction are hard to find. Therefore, instead of reducing the error rate, some predefined discriminative measure in data is maximized.

There are several attempts in literature to design a discriminative feature extraction [4], [5], [31]. Various techniques have also been used to jointly train the parametric form feature extraction and classifier [45], [3], [12], [35], [33], [8], [7], [9].

My approach to the problem is to use the classifier built in the higher dimensional space, and use its misclassification measure as an estimator of the discrimi-

nation ability of different directions in the feature space. Although such a classifier cannot be a solution for the classification problem, it can provide certain useful information on the informativeness of different directions in the space. In this chapter, we first review some related work for feature selection and extraction, and then describe our proposed algorithm.

3.2 Related works

A classical method to reduce the dimensionality of data is a technique known as *Karhunen-loeve expansion* (KL-expansion) or *principal component analysis* [15]. Consider an n-dimensional random vector $\vec{X}$. This vector can be represented by n orthogonal basis vector $\vec{\Phi}_i$ without any error. In KL-expansion, we represent the n-dimensional input vector $\vec{X}$ by an m-dimensional estimate $\hat{\vec{X}}_m$. Here, the estimate is found such that the mean-squared of the magnitude difference of these two vectors is minimized. That is, to minimize

$$\bar{\epsilon^2} = E\{||\vec{X} - \hat{\vec{X}}_m||^2\}. \tag{3.1}$$

It can be shown that this estimate should be evaluated as follows (see [17] for more detail)

$$\hat{\vec{X}}_m = \sum_{i=1}^{m} y_i \vec{\Phi}_i + \sum_{i=m+1}^{n} b_i \vec{\Phi}_i, \tag{3.2}$$

where

$$b_i = E\{y_i\} = \vec{\Phi}_i^T E\{\vec{X}\}. \tag{3.3}$$

and $\vec{\Phi}_i$'s are the eigenvectors of the covariance matrix of $\vec{X}$, Σ_X, sorted to the order of their corresponding eigenvalues λ_i, where the largest one is $\vec{\Phi}_1$. In this case, the

error can be shown to be (see [15])

$$\bar{\epsilon^2} = \sum_{i=m+1}^{n} \lambda_i. \tag{3.4}$$

The normalized value of eigenvalues can be defined as

$$\mu_i = \frac{\lambda_i}{\sum_j \lambda_i}. \tag{3.5}$$

Based on the acceptable level of error or the normalized value of eigenvalues, one can select the dimensionality of the new feature space. The KL-transform has three attractive properties. First, it can order the importance of each direction in representing $\vec{X}$ using the value of its corresponding eigenvalues. Second, the covariance matrix of $\vec{\bar{X}}$ is diagonal. Third, the transformation is optimum in terms of minimizing the mean-squared error defined in (3.1) over all choices of orthogonal transformations.

The disadvantage of using the KL-expansion in pattern recognition is that this transformation is only appropriate in terms of representing the data not in terms of minimizing the classification error or maximizing a class separability measure. Another disadvantage of this algorithm may appear in calculating the covariance matrix, since we require an increased number of sample data when the dimensionality of the input space is increased.

Another common approach for feature extraction is based on finding a linear mapping A of the n dimensional measurement space, and then select m features such that a measure of discrimination ability is maximized. Fisher's linear discriminant method [15] is an example of this approach, where we try to maximize

$$\frac{A^t \Sigma_b A}{A^t \Sigma_w A}, \tag{3.6}$$

where Σ_b and Σ_w are the between-class and within-class scatter matrices, respectively, and are defined as

$$\Sigma_b = \sum_i P(C_i)(\vec{M}_i - \vec{M}_o)(\vec{M}_i - \vec{M}_o)^T, \tag{3.7}$$

$$\Sigma_w = \sum_i P(C_i)\Sigma_i, \tag{3.8}$$

where $P(C_i)$, $\vec{M}_i$ and Σ_i are the *a priori* probability, mean and covariance matrix of class C_i, respectively. $\vec{M}_o$ is the mean of all classes, *i.e.*,

$$\vec{M}_o = \sum_i P(C_i)\vec{M}_i. \tag{3.9}$$

Fisher's algorithm finds the linear mapping A from the n dimensional space to the $K-1$ dimensional space where K is the number of classes. It can be shown that the optimum solution for this criterion can be obtained by solving a generalized eigenvector problem

$$\Sigma_B.\vec{a}_i = \lambda_i \Sigma_w \vec{a}_i, \tag{3.10}$$

where $\vec{a}_i$ are columns of A corresponding to $k-1$ nonzero eigenvalues of λ_i. The reader is referred to [15] for more detail. The basic shortcoming in using such criterion is that the class separability measure used in Eq. (3.6) is not optimum for all classification problems. There are certain practical cases where this criterion does not perform well. For example, when the mean of a class is very different from the mean of other classes, that class will then be dominant in calculating the between class scatter matrix, and therefore undermining the feature extraction method. Moreover, the true estimation of the criterion may not be very accurate when the dimensionality of the feature space is increased due to possible lack of enough training data.

C. Lee *et al.* use decision boundaries estimated in the higher dimensional space to find the importance of different directions in the feature space [32]. They first

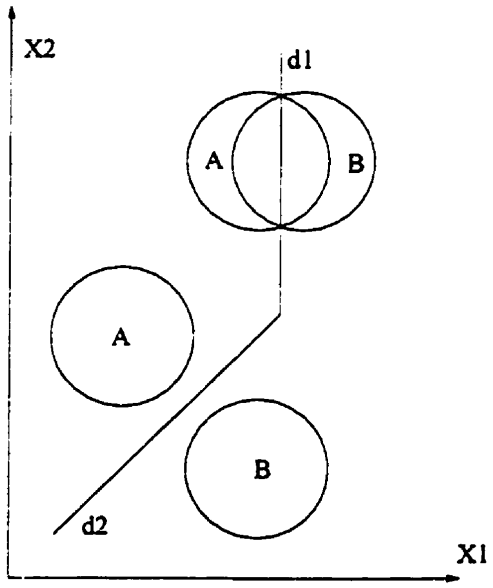


Figure 3.1: A simple counter example for Lee's algorithm

define the following feature matrix

$$\Sigma_D = \frac{1}{K} \int_S \vec{N}(x)\vec{N}^T(x)p(x)dx, \tag{3.11}$$

where $\vec{N}(x)$ is the unit normal vector to the decision boundary at point x, $p(x)$ is the probability density function of x, $K = \int_S p(x)dx$, and S is the decision boundary. They claim that if this matrix has zero eigenvalues, the direction of their corresponding eigenvectors do not have any useful information in the classification task. They further generalize the algorithm and try to sort different direction based on the eigenvalues of the feature matrix.

It can be seen that the above claim is not valid in some practical cases where $p(x) = 0$ on the decision boundary. Fig. 3.1 is such an example. Here, we assume the distribution of data is uniform within each circle for each class and it is zero elsewhere. Following the above algorithm, we can see that the feature matrix has a zero eigenvalue and that the zero eigenvalue corresponds to the direction

parallel to line d_1. That means the direction perpendicular to such a line can only be considered for classification without having any increase in error rate. As can be seen from the figure, this direction will result in an increase in error rate. Moreover, $p(x)$ is introduced in Eq. (3.11) to place a weight on regions having a higher concentration of data. Such emphasis is simply not valid. In fact, every point in space should provide a share in feature selection or extraction as each point has a share in the probability of error.

My proposed approach for feature extraction and selection that will be presented in the next section was motivated by the above approach. In the proposed approach, decision boundaries are approximated by several hyper-planes. The difference in calculating the similarity matrix is in the weighting factors $p(x)$. These weighting factors are replaced by the misclassification measure of each point. Also the integration is carried out over the whole space not just over the decision boundaries.

As mentioned before, one can directly select a subset of features without doing any transformation. This approach is rather interesting mainly because of its reduced computational cost in building the classifier after the selection of appropriate features. Moreover, feature selection can eliminate irrelevant information (*i.e.* noise in general). Here, some measures of class separability are also used instead of probability of error to evaluate the importance of different features. Probability distances are examples of such measures. These measures allow evaluation of discriminative power of each feature between two classes only. The followings are some examples of them [15]:

- Bhattacharyya's distance

$$J_b = -\ln \int_X [P(X|C_1) - P(X|C_2)]^{1/2} dX. \tag{3.12}$$

- Divergence

$$J_d = \int_X [P(X|C_1) - P(X|C_2)] \ln \frac{P(X|C_1)}{P(X|C_2)} dX. \tag{3.13}$$

In another algorithm, mutual information between features and the classes is used to order different feature. This mutual information is defined as [11][2]

$$I(C;X) = \sum_i \int_X P(C_i;X) \log[\frac{P(X,C_i)}{P(X)P(C_i)}] dX. \tag{3.14}$$

Mutual information measures the amount by which the knowledge provided by a feature decreases the uncertainty about that class. Therefore, the most informative feature can be found using the above measure. Generally, the nth selected feature, should be the one maximizing

$$I(C;X_n|X_1,\ldots,X_{n-1}) = \sum_i \int_X P(C_i;X_n|X_1,\ldots,X_{n-1}) \log[\frac{P(X_n,C_i)}{P(X_n)P(C_i)}] dX. \tag{3.15}$$

The computational cost and the increased number of required sample data to calculate the above mutual information makes the above algorithm practically infeasible, when the dimensionality increases.

3.3 The proposed algorithm

If the decision boundary is a hyper-plane, the directions parallel to such a hyper-plane do not contain any useful information for the purpose of classification. Therefore, we can use the direction perpendicular to such hyper-plane only, without changing the recognition error rate. However, the decision boundaries in practice are not usually hyper-planes and such absolutely redundant directions do not exist. However, decision boundaries can be estimated using a collection of hyper-planes. As we will see in this section, the perpendicular directions to such hyper-planes,

if properly weighted, can provide us useful information about the importance of different directions in the space.

Here, linear orthogonal transforms are found that can map the higher dimensional space before applying feature extraction. It is shown that in the lower dimensional space, at least a model can be build that its misclassification measure (or indirectly the probability of error) is close to that of the model in the higher dimensional space. Indeed, an upper-bound can be found in the maximum change of misclassification measure. The desired transform is the one that minimizes this upper-bound. It is also shown that a subset of features can be selected that minimizes the upper-bound.

Assume that the model is first trained in the higher dimensional space based on the discriminative training criterion explained in Section 2.3. Also assume that after the linear transformation Φ, only the first m-dimension of a vector $\vec{V}$ are used in the new space for its representation. Such a vector is shown by $\tilde{\vec{V}}$. That is

$$\vec{V} = \sum_{i=1}^{n} y_i \vec{\Phi}_i, \tag{3.16}$$

and,

$$\tilde{\vec{V}} = \sum_{i=1}^{m} y_i \vec{\Phi}_i. \tag{3.17}$$

The squared error in representing $\vec{V}$ can be defined as

$$\epsilon^2(\vec{V}) = \sum_{i=m+1}^{n} y_i^2. \tag{3.18}$$

Here, it is shown that if the model and input are transformed to a lower dimensional space, the change in the overall misclassification measure is bounded.

The cost function of an input X^u for class i was defined as (see Chapter 2)

$$l_i = \gamma\left(\frac{\xi_3}{M-1}\left(\sum_{j,j\neq i}\gamma\left(\frac{\xi_2}{T_u}\sum_{t}\gamma(\xi_1\pi_{ij}^t) - 0.5\right)\right) + \theta_M\right). \tag{3.19}$$

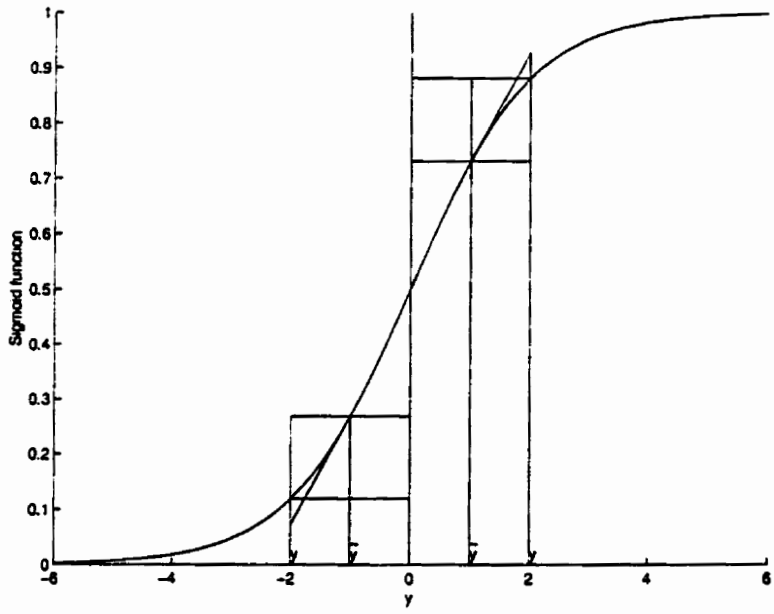


Figure 3.2: Sigmoid function

Let us define,

$$y = \frac{\xi_3}{M-1}\left(\sum_{j,j\neq i}\gamma\left(\frac{\xi_2}{T_u}\sum_t \gamma(\xi_1 \pi_{ij}^t) - 0.5\right)\right) + \theta_M), \tag{3.20}$$

and,

$$\omega = \frac{\xi_2}{T_u}\sum_t \gamma(\xi_1 \pi_{ij}^t) - 0.5. \tag{3.21}$$

Also consider $\tilde{y}$, $\tilde{\omega}$ and $\tilde{\pi}_{ij}^t$ as the estimation of y, ω and π_{ij}^t in the lower dimensional space, respectively. Considering Fig. 3.2, and the convexity of the sigmoid function, it can be written that

$$\begin{aligned} |\gamma(y) - \gamma(\tilde{y})| &\leq |y - \tilde{y}|\frac{\partial\gamma(x)}{\partial x}|_{x=x_0} \\ &\leq |y - \tilde{y}|\gamma(x_0)(1-\gamma(x_0)), \end{aligned} \tag{3.22}$$

where x_0 is

$$x_0 = \begin{cases} \min(y,\tilde{y}) & \text{if} \min(y,\tilde{y}) > 0 \\ \max(y,\tilde{y}) & \text{else,} \end{cases} \tag{3.23}$$

However, it can easily be shown that

$$\gamma(x)(1-\gamma(x)) < 0.25 \quad \forall x. \tag{3.24}$$

Therefore,

$$|\gamma(y)-\gamma(\tilde{y})| \leq 0.25|y-\tilde{y}|, \tag{3.25}$$

$$|y-\tilde{y}| < \frac{\xi_3}{M-1}(\sum_{j,j\neq i}|\gamma(\omega)-\gamma(\tilde{\omega})|). \tag{3.26}$$

In a similar way

$$|\gamma(\omega)-\gamma(\tilde{\omega})| \leq 0.25|\omega-\tilde{\omega}|, \tag{3.27}$$

and

$$|\omega-\tilde{\omega}| \leq 0.25\frac{\xi_2}{T_u}\sum_t|\gamma(\xi_1\pi_{ij}^t)-\gamma(\xi_1\tilde{\pi}_{ij}^t)|. \tag{3.28}$$

In the following, an upper bound for $|\gamma(\xi_1\pi_{ij}^t)-\gamma(\xi_1\tilde{\pi}_{ij}^t)|$ is found.

As it was seen in Section 2.3.1. $\pi_{ij}^t = d_t\alpha_t q_t$, where d_t is the distance of input vector X_t from its corresponding boundary hyper-plane, α_t is the distance of of the pair of code-books selected for input X_t, q_t is 1 if the input vector is correctly classified and -1 otherwise.

Consider the parallel vectors $d_t\vec{N}_t$, $\alpha_t\vec{N}_t$, and $\gamma(\xi_1 d_t\alpha_t q_t)\vec{N}_t$ in the feature space, where $\vec{N}_t$ is the unit normal vector of the decision hyper-plane in the direction of correct class associated to input vector X_t. The lengths of the first m-dimension of $d_t\vec{N}_t$ and $\alpha_t\vec{N}_t$ after the transformation are shown by $\tilde{d}_t$ and $\tilde{\alpha}_t$, respectively. Since the vectors $d_t\vec{N}_t$, $\alpha_t\vec{N}_t$ and $\gamma(q_t d_t)\vec{N}_t$ are parallel to each other, we can have

$$\epsilon^2(\gamma(\xi_1 d_t\alpha_t q_t)\vec{N}_t) = \epsilon^2(d_t\vec{N}_t)\frac{\gamma^2(\xi_1 d_t\alpha_t q_t)}{d_t^2}, \tag{3.29}$$

and

$$d_t^2 = \tilde{d}_t^2 + \eta d_t^2, \tag{3.30}$$

where

$$\eta = \frac{\epsilon^2(\gamma(\xi_1 d_t \alpha_t q_t)\vec{N}_t)}{\gamma^2(\xi_1 d_t \alpha_t q_t)}. \tag{3.31}$$

Considering $\tilde{d}_t > 0$ and $0 \leq \eta \leq 1$

$$\tilde{d}_t = (1-\eta)^{1/2} d_t. \tag{3.32}$$

In a same way

$$\tilde{\alpha}_t = (1-\eta)^{1/2} \alpha_t. \tag{3.33}$$

Therefore

$$(1-\eta) d_t \alpha_t = \tilde{d}_t \tilde{\alpha}_t. \tag{3.34}$$

Considering the convexity of $\gamma(\xi_1 d_t \alpha_t q_t)$ when $q_t = 1$,

$$\gamma(\xi_1 d_t \alpha_t) - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t) \leq \eta \xi_1 d_t \alpha_t \frac{\partial(\gamma(x))}{\partial x}|_{x=\xi_1 \tilde{d}_t \tilde{\alpha}_t}. \tag{3.35}$$

or

$$\gamma(\xi_1 d_t \alpha_t) - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t) \leq \eta \xi_1 d_t \alpha_t \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t)(1 - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t)). \tag{3.36}$$

Considering Eq. (3.34),

$$\gamma(\xi_1 d_t \alpha_t) - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t) \leq \frac{\eta}{(1-\eta)} \xi_1 \tilde{d}_t \tilde{\alpha}_t \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t)(1 - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t)). \tag{3.37}$$

It can easily be shown that

$$x\gamma(x)(1-\gamma(x)) < 0.23 \quad \forall x. \tag{3.38}$$

Therefore

$$\gamma(\xi_1 d_t \alpha_t) - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t) \leq 0.23 \frac{\eta}{(1-\eta)}, \tag{3.39}$$

or

$$\gamma(\xi_1 d_t \alpha_t) - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t) \leq 0.23 \frac{\epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t)}{\gamma^2(\xi_1 d_t \alpha_t) - \epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t)}. \tag{3.40}$$

Considering $\xi_1 d_t \alpha_t$ is positive,

$$0.25 \leq \gamma^2(\xi_1 d_t \alpha_t) \leq 1., \tag{3.41}$$

and

$$\gamma(\xi_1 d_t \alpha_t) - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t) \leq 0.5, \tag{3.42}$$

which results

$$\gamma(\xi_1 d_t \alpha_t) - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t) \leq \min(0.23 \frac{\epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t)}{0.25 - \epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t)}, 0.5). \tag{3.43}$$

The first term is dominant when $\epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t) < 0.18$. By evaluating the denominator with the highest value of $\epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t)$, we can write:

$$\gamma(\xi_1 d_t \alpha_t) - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t) \leq 0.28\epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t) \quad \text{if } \epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t) < 0.18, \tag{3.44}$$

But

$$0.28\epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t) > 0.5 \quad \text{if } \epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t) > 0.18. \tag{3.45}$$

Therefore

$$\gamma(\xi_1 d_t \alpha_t) - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t) \leq 0.28\epsilon^2(\gamma(\xi_1 d_t \alpha_t)\vec{N}_t). \tag{3.46}$$

Following similar steps for $q_t = -1$,

$$\gamma(-\xi_1 \tilde{d}_t \tilde{\alpha}_t) - \gamma(-\xi_1 d_t \alpha_t) \leq 0.28\epsilon^2(\gamma(-\xi_1 d_t \alpha_t)\vec{N}_t). \tag{3.47}$$

Combining the above two inequalities,

$$|\gamma(\xi_1 d_t \alpha_t q_t) - \gamma(\xi_1 \tilde{d}_t \tilde{\alpha}_t q_t)| \leq 0.28\epsilon^2(\gamma(\xi_1 d_t \alpha_t q_t)\vec{N}_t). \tag{3.48}$$

Combining the inequalities (3.25)-(3.28) with the above inequality, it can be concluded that the change in misclassification measure can be written as

$$|l_i - \tilde{l}_i| \leq 0.28 * 0.25^2 \frac{\xi_3 \xi_2}{M-1} \sum_{j, j \neq i} \sum_t \epsilon^2(\frac{\gamma(\xi_1 d_t \alpha_t q_t)}{T_u}\vec{N}_t). \tag{3.49}$$

The above bound shows that the loss in misclassification measure is bounded by the loss in representing the vectors perpendicular to decision hyper-planes weighted by $(\frac{\gamma(\xi_1 d_t \alpha_t q_t)}{T_u})$. Based on K-L expansion technique, such vectors can be best presented if the columns of transformation are the eigenvectors of the covariance matrix of these vectors sorted by the value of their corresponding eigenvalues.

Note that one can also select a subset of features without having any transformation by minimizing the upper bound over all possible subset of features. This can be done by sorting different directions by maximizing the average length of the projection of vectors $(\frac{\gamma(\xi_1 d_t \alpha_t q_t)}{T_u}\vec{N}_t)$ in different direction.

As it was shown above, minimization of the maximum change in overall misclassification measure using feature selection or extraction requires proper representation of a set of vectors $\gamma(\xi_1 d_t \alpha_t q_t)\vec{N}_t$ calculated for every frame of the input samples of the training set. Therfore, one can partition this set depending on selection of states in different models and find a proper transformation for each partition.

The above feature selection and extraction strategy can also be applied for the segmentation algorithm described in the previous chapter. Considering Eq. (2.53), and by going through similar steps as finding the bound in Eq. (3.49), it can be seen that the maximum change in evaluating discriminative functions of each path is bounded

$$|g(X^u;q^u) - g(\tilde{X}^u;q^u)| < 0.28 * 0.25\frac{\xi_2}{(J-1)}\sum_{t=1}^{T_u}\sum_{j=1,j\neq i}^{J}\epsilon^2(\frac{\gamma(\xi_1\pi^t_{s^u_t s_j})}{T_u}\vec{N}_t). \quad (3.50)$$

where $\vec{N}_t$ is the unit normal vector for the hyper-plane associated to state s^u_t and s_j. By considering such a bound, feature selection or extraction can be done by properly presenting the vectors $\frac{\gamma(\xi_1\pi^t_{s^u_t s_j})}{T_u}\vec{N}_t$ in the new space. Such representation can also be state dependent. Feature selection and extraction is appealing for the

segmentation algorithm to reduce the additional cost of segmentation imposed by using semi-Markov models.

3.4 Summary

It was shown in this chapter that for the given state model described in previous chapter, after an orthogonal transform, the dimensionality of the space can be reduced. It was also shown that by using such transform, the maximum change in misclassification measure of the models trained in the higher dimensional space has an upper-bound. The upper bound can be minimized by properly representing the vectors perpendicular to decision hyper-planes weighted by the share of the input vectors in the overall misclassification measure. The proper representation of such vectors is an easy task using KL-expansion algorithm. Feature selection is also possible by projecting such vectors in different directions of space and selecting the directions having a higher accumulated projection. The proposed algorithm can also be applied to the discriminant segmentation algorithm described in the previous chapter. Since the differences within speech units such as word is less complicated, the required number of features in practice is very small (5 features in the experiments reported in chapter 5). It was also discussed that feature selection and extraction can be implemented depending on state of the model.

Chapter 4

Feature extraction using spectrogram

4.1 Introduction

The speech spectrogram is a time-dependent Fourier representation of speech signal. To calculate the speech spectrogram, the speech signal is first Hamming-windowed with a window size of about 40ms. The resulting signal is then zero-padded and its Fast Fourier Transform is taken. The Hamming-window is then moved forward (about 10ms) and the same process is repeated.

Experts can use spectrogram and classify words or phonemes from a spoken sentences with a high accuracy rate [40], [27], [16], [18], [37], [48]. It is one of the objectives of this thesis to properly measure the features that are used by these experts. One of the most important features is the position of resonant frequencies or formants in the spectrogram and their relative movements, the existence of voicing information in the signal, and the distribution of energy patterns in the

spectrogram.

To extract features from the spectrogram, the image is first segmented into two homogeneous regions of object and background, each having similar characteristics. The object class is associated to regions having the desired features. For example, the object class can be the regions associated to resonance frequencies.

Existing segmentation algorithms include amplitude thresholding, component labeling, boundary-based approaches, region-based approaches, template matching and texture segmentation. An overview of the existing algorithms can be found in [6] and [20]. In the following section, a new self-organizing image segmentation algorithm is introduced.

4.2 A new self-organizing image segmentation algorithm

The overall goal of this algorithm is to segment images consisting of an object and a background. In particular, a self-organizing segmentation algorithm is presented that can segment the image based on *a priori* knowledge of object and background characteristics. These characteristics include the knowledge about the intensity of pixels in object and background classes and the shape of these classes. Based on Bayes decision theory, the optimum segmentation, in terms of minimizing the probability of segmentation error, is to decide in favor of object for each image pixel (j) if the *a posteriori* probability of that pixel belonging to object (O) is greater than that of background class (B) given the input image (I). That is if

$$P(j \in O|I) > P(j \in B|I). \tag{4.1}$$

A common segmentation strategy is then to estimate the above *a posteriori* probabilities using parametric stochastic models for object and background classes [10], [13], [39], [47], and [25]. However, such methods rely on unrealistic assumptions made in the selection of their stochastic models. Moreover, it is usually hard to find the optimum set of model parameters. As a result, the parametric estimation of the *a posteriori* probabilities does not usually result in accurate segmentation. However, these estimates can provide some information on the degree of confidence in making segmentation decisions by considering their closeness to one or zero. The proposed algorithm uses such confidence information provided by estimates of *a posteriori* probabilities along with a priori knowledge about the object shape. Here, the value of each pixel's *a posteriori* probabilities is iteratively adjusted where less ambiguous ones are adjusted more in each iteration in hope of finding a better estimate of the *a posteriori* probabilities for other pixels in the next iterations. Note that the *a posteriori* probabilities are functions of the value of other pixel's *a posteriori* probabilities in the image (usually the neighboring pixels). To avoid instability in such adjustments, an error function is also defined as a measure of overall segmentation of output image pixels. This error function is minimum for any binary image (with pixels having probabilities of zero or one). The adjustment strategy also reduces this error function, thereby leading to a higher degree of segmentation. The *a posteriori* probabilities are initially estimated using the intensities of image pixels and the knowledge of the object shape. After this initial estimation is found, the *a posteriori* probabilities are adjusted using the *a posteriori* probabilities of other pixels and the knowledge of the object shape.

The initial estimate of $P(j \in O|I)$ can be provided as

$$p_j = P(j \in O|I) = \frac{\sum_{k \in D} h(j,k) o_k}{\sum_{k \in D} h(j,k)} \tag{4.2}$$

where o_j is the intensity of pixel j, $h(j,k)$ is a constant depending on pixels j and

k. D is a set of neighboring pixels (including node j), O is the object class and I is the given image. After this initial estimate of p_j, these probabilities are adjusted during the segmentation phase using

$$\Delta p_j^{t+1} = \alpha \sum_{k \in D} d(j,k)(p_k^t - 0.5), \tag{4.3}$$

where $d(j,k)$ is a weighting factor that should be selected based on the shape of objects, p_k is the estimate of *a posteriori* probability of pixel k, t is the iteration number, and α is a positive constant. Here, it is assumed that $d(j,k) = d(k,j)$. This adjustment strategy adjusts image pixel's probabilities iteratively, where less ambiguous pixels are adjusted more in each iteration. The above adjustments will also result in a more segmented image after each iteration. If we carefully examine the above training strategy we notice that it also minimizes the following error function for which any binary image (having *a posteriori* probabilities of zero or one) is a global minimum:

$$E = \sum_{j=1}^{n} \sum_{k \in D} d(j,k) p_j (1 - p_k). \tag{4.4}$$

This can be shown by taking the derivative of E with respect to p_j and assuming that $d(j,k) = d(k,j)$.

$$\begin{aligned} \frac{\partial E}{\partial p_j} &= \textstyle\sum_{k \in D} d(j,k)(1 - p_k) - d(k,j) p_k \\ &= -\tfrac{1}{2} \textstyle\sum_{k \in D} d(j,k)(p_k - 0.5). \end{aligned} \tag{4.5}$$

Considering that α is a positive constant, and by comparing Eq. (4.5) and (4.3), we can conclude that the adjustment algorithm reduces the error function defined in Eq. (4.4), as the adjustment is in the negative direction of the derivative of the defined error function. In summary, the end product is a self-organizing algorithm described as follows:

- Calculate an estimate of *a posteriori* probability of image pixels belonging to object or background classes based on their intensity and the shape of object.
- Select a proper $d(j,k)$ (weighting coefficients of the defined measure) based on the shape of the objects. Also, select a learning rate (α) and adjust the estimate of probabilities iteratively using the following equations until a desired segmentation level is achieved at the output image

$$\Delta p_j^{t+1} = \begin{cases} \alpha\left[\sum_{k\in D} d(j,k)(p_k^t - 0.5)\right] & \text{if } p_j^t \neq 0 \text{ or } 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

and keep $0 \leq p_j^t \leq 1$.

Also, one should note that the change in the probability of node j, caused by a neighboring pixel k is limited to $\frac{\alpha}{2}d(j,k)$, prohibiting the system from excessive smoothing.

4.2.1 Formant segmentation

Here, formant regions are referred to regions that have high energy and are close to the resonant frequencies of speech signal. Correct identification of such regions can play an essential role in any speech recognition system. To segment the image into these two regions, the following approximation was made

$$p_{ij} = o_{ij}, \tag{4.7}$$

where i refers to horizontal position of a pixel in the image and j refers to its vertical position. After this initial estimation, the probability of each pixel is estimated using

$$\Delta p_{ij}^{t+1} = \begin{cases} \alpha\left[\sum_{i=-I}^{I}\sum_{j=-J}^{J} d(j)(p_{ij}^t - 0.5)\right] & \text{if } p_{ij}^t \neq 0 \text{ or } 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

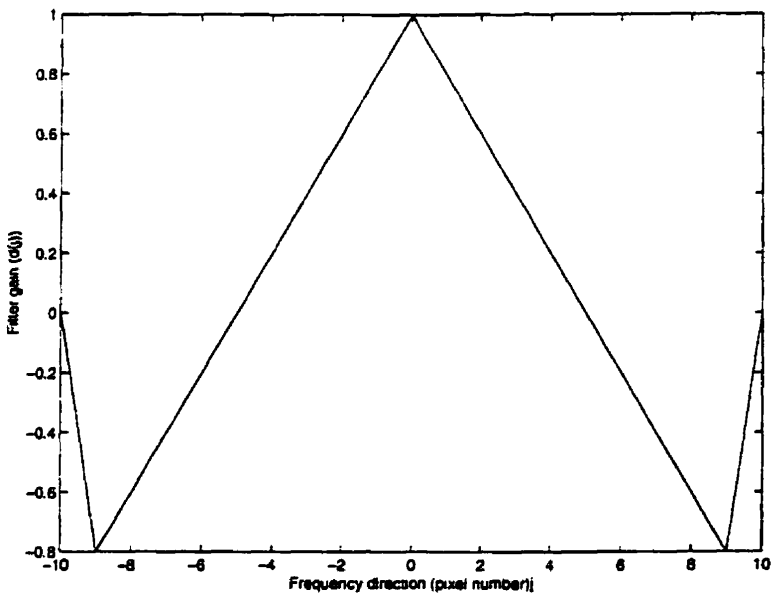


Figure 4.1: The filter that is used for format features (frequency range of each spectrogram contains 200 pixels for spectrograms up to 6KHz)

where $d(j)$ is given as in Fig. 4.1, I and J define the size of neighboring pixels. In these experiments, spectrogram images have 200 pixels in the frequency direction and 1 pixel every $1ms$ of time domain. For these images, $I = 11$ and $J = 9$. α was selected 0.2 and the segmentation was carried out for 10 iterations. The resulting segmented images of the words /nine/, /one/, /zero/, and their corresponding segmented images for different iterations are shown in Fig. 4.2- 4.10. Please compare the difference of the words /one/ and /nine/ in the time domain between $0.3sec$ and $0.4sec$ of both words for the segmented images and the original spectrogram images. It is indeed these regions that can play a significant role in classification of the two words.

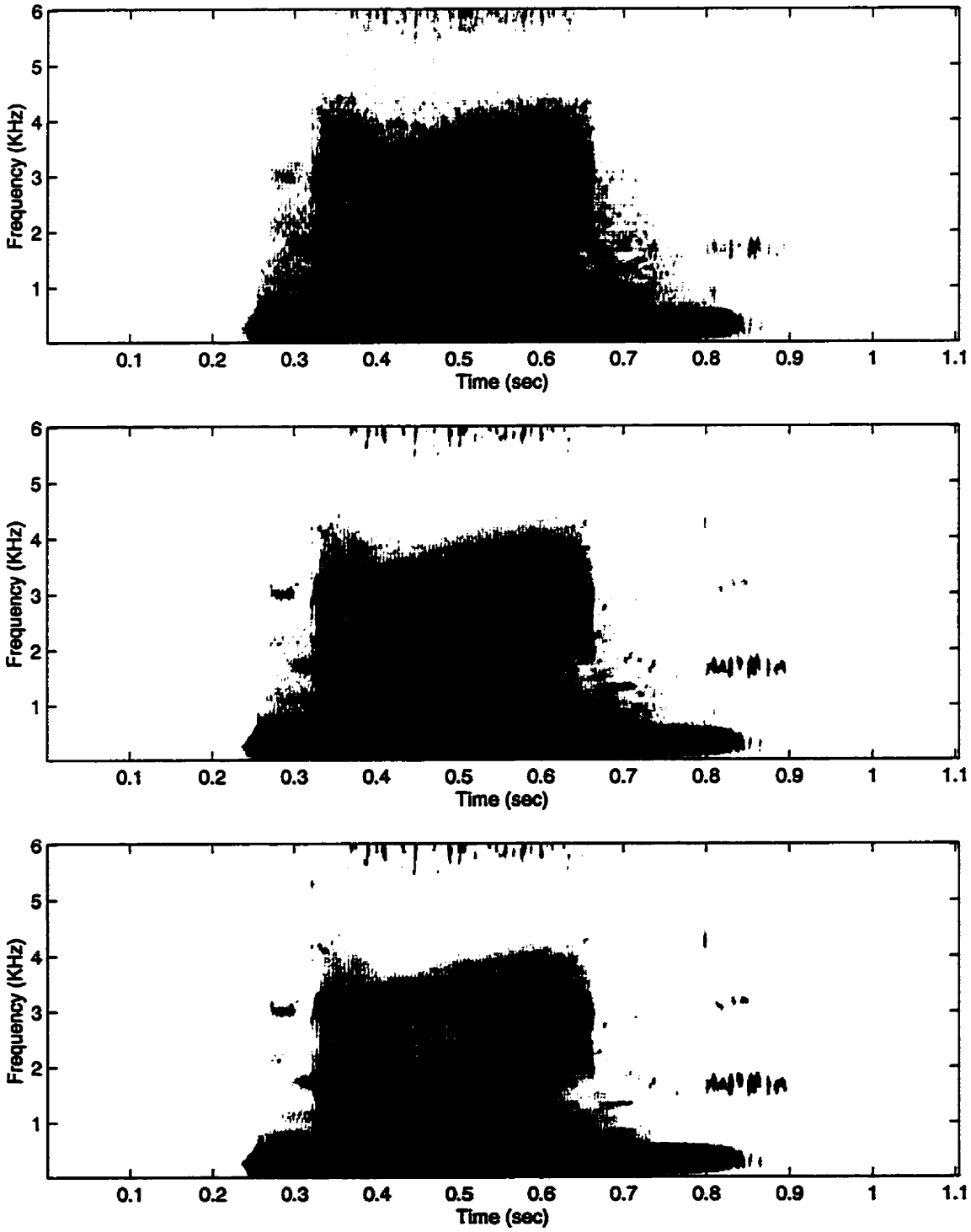


Figure 4.2: Progress of segmentation of word nine in different iterations, from top to button: origin-gal spectrum, iteration number 1,2

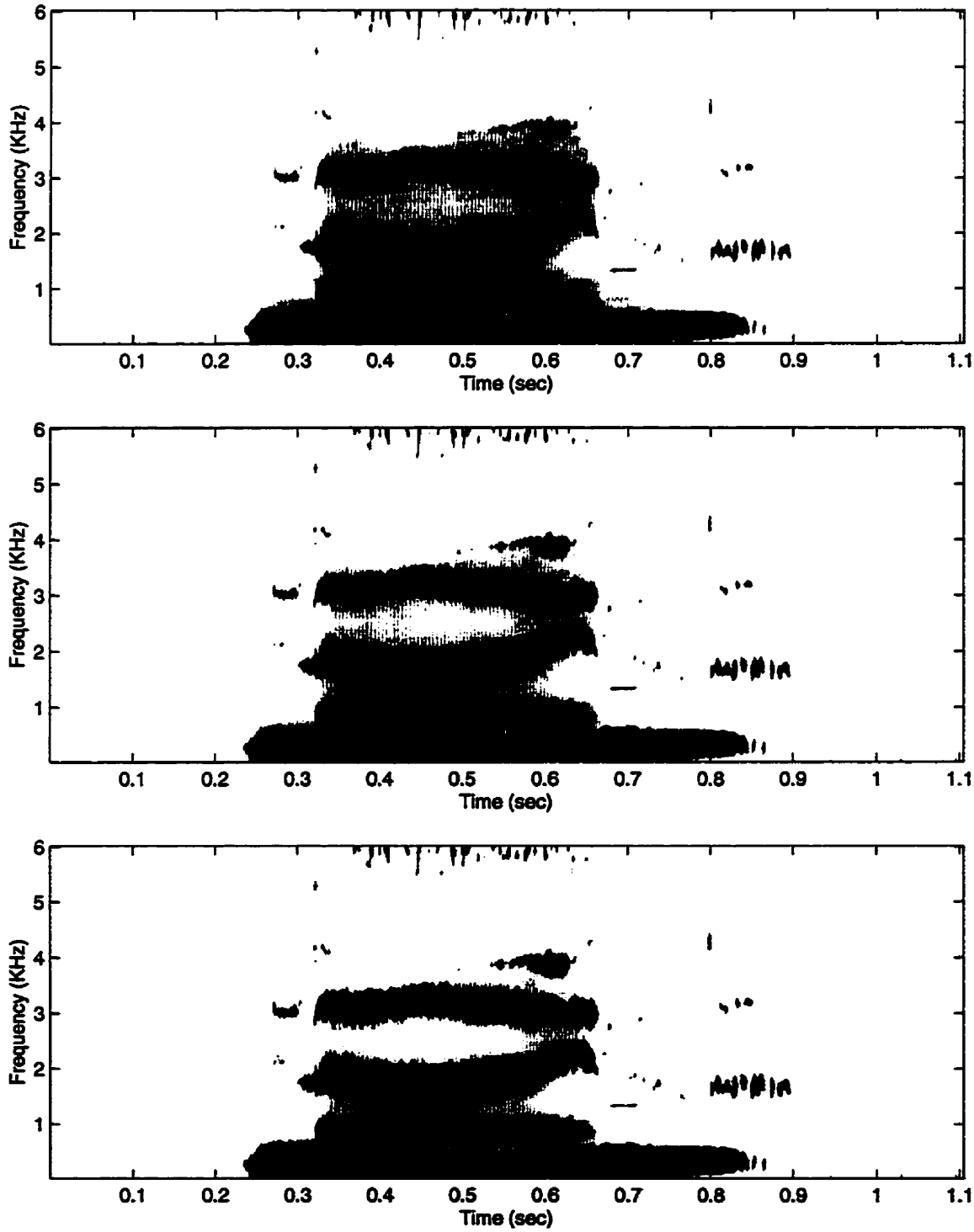


Figure 4.3: Progress of segmentation of word nine in different iterations, from top to button, iteration number: 3, 4, 5

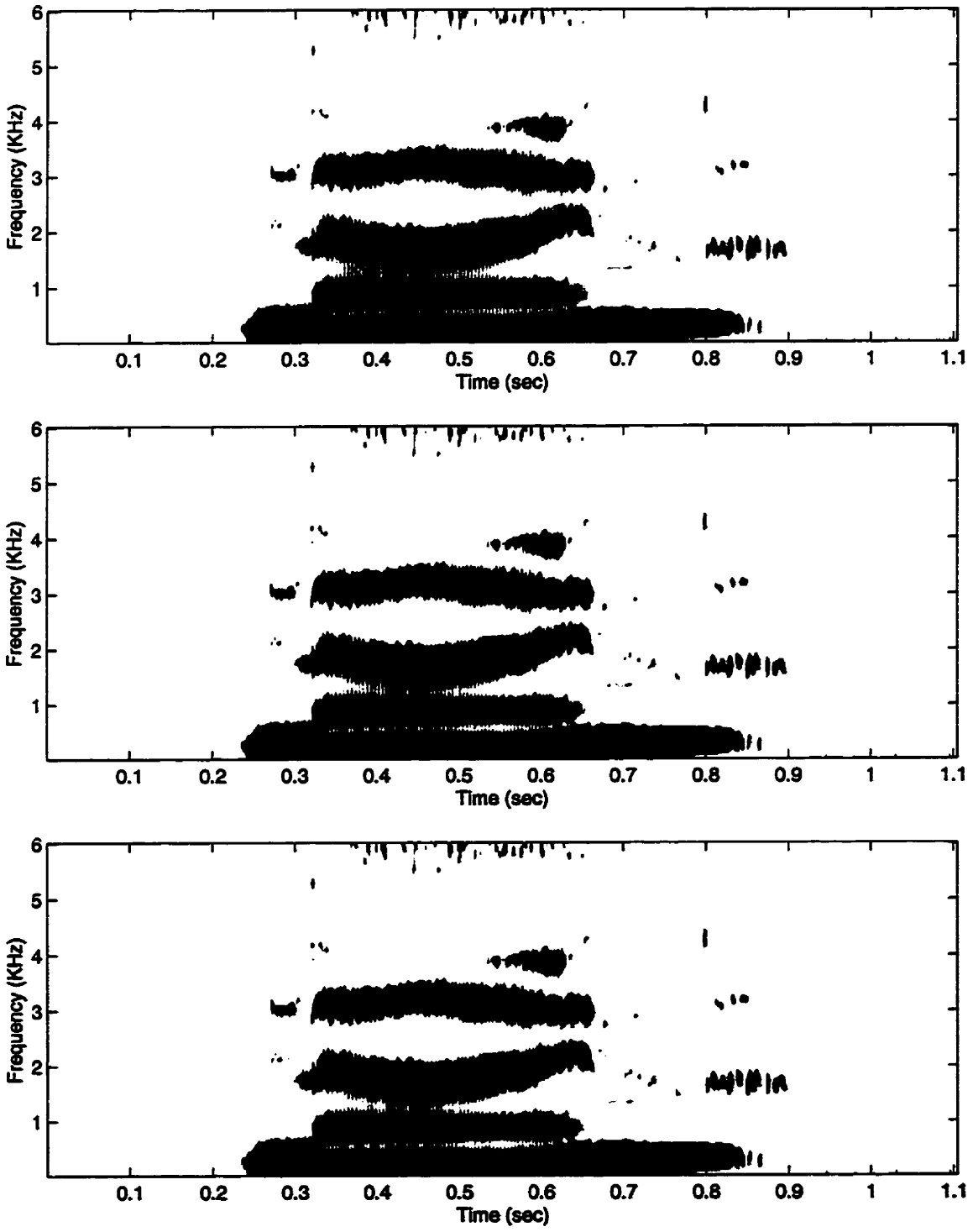


Figure 4.4: Progress of segmentation of word nine in different iterations, from top to button, iteration number: 6,7,8

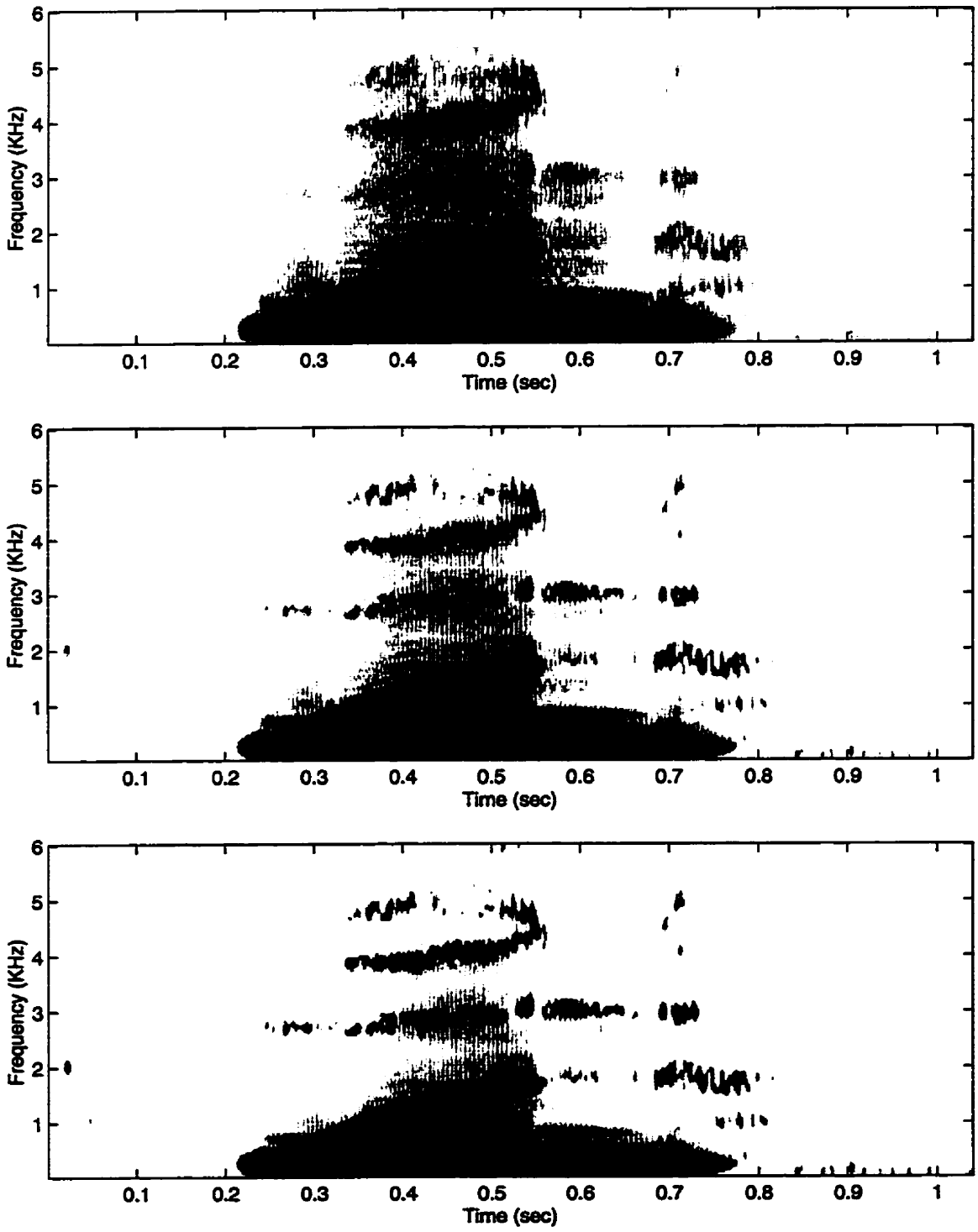


Figure 4.5: Progress of segmentation of word one in different iterations, from top to button: original spectrum, iteration number 1,2

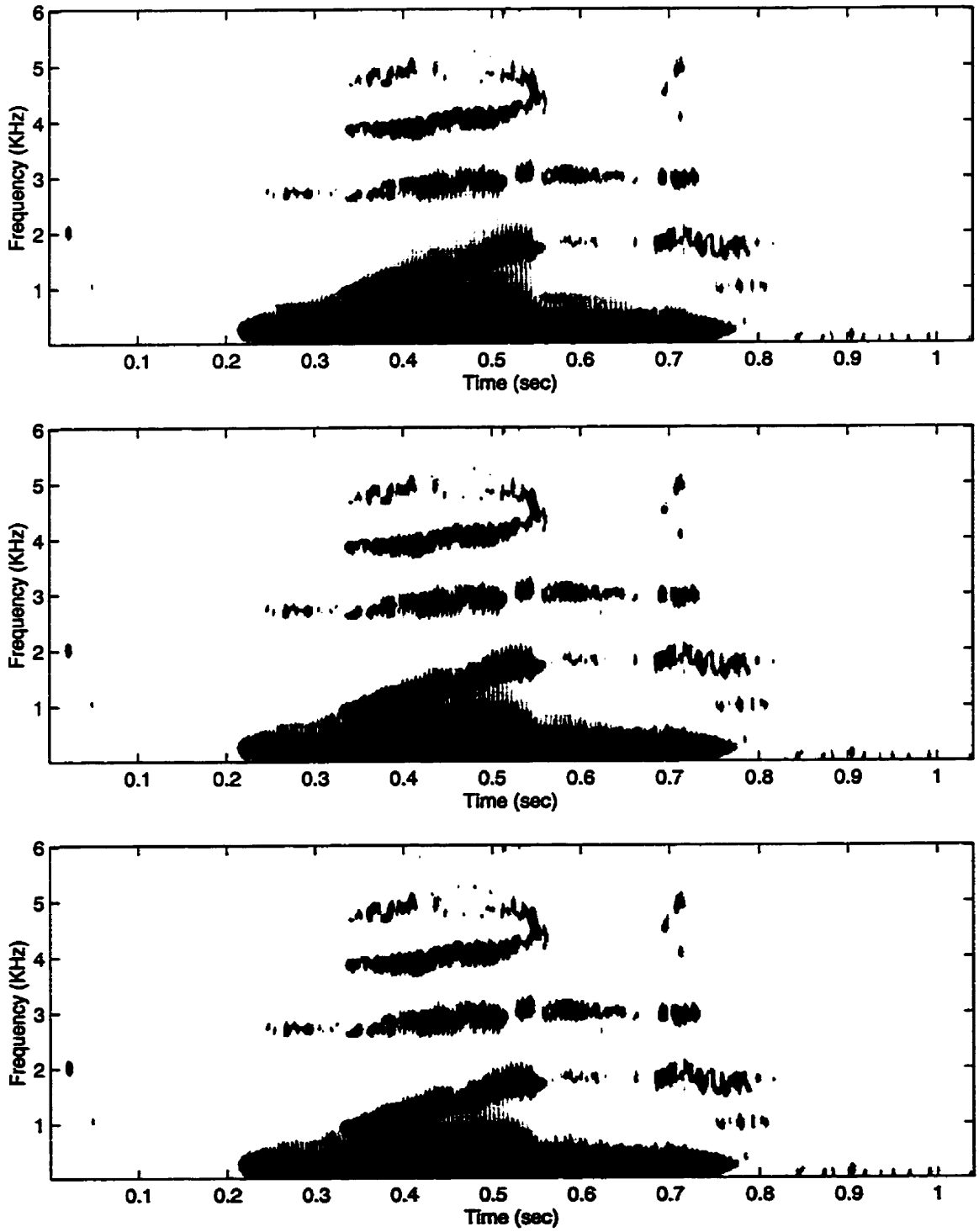


Figure 4.6: Progress of segmentation of word one in different iterations, from top to button, iteration number: 3, 4, 5

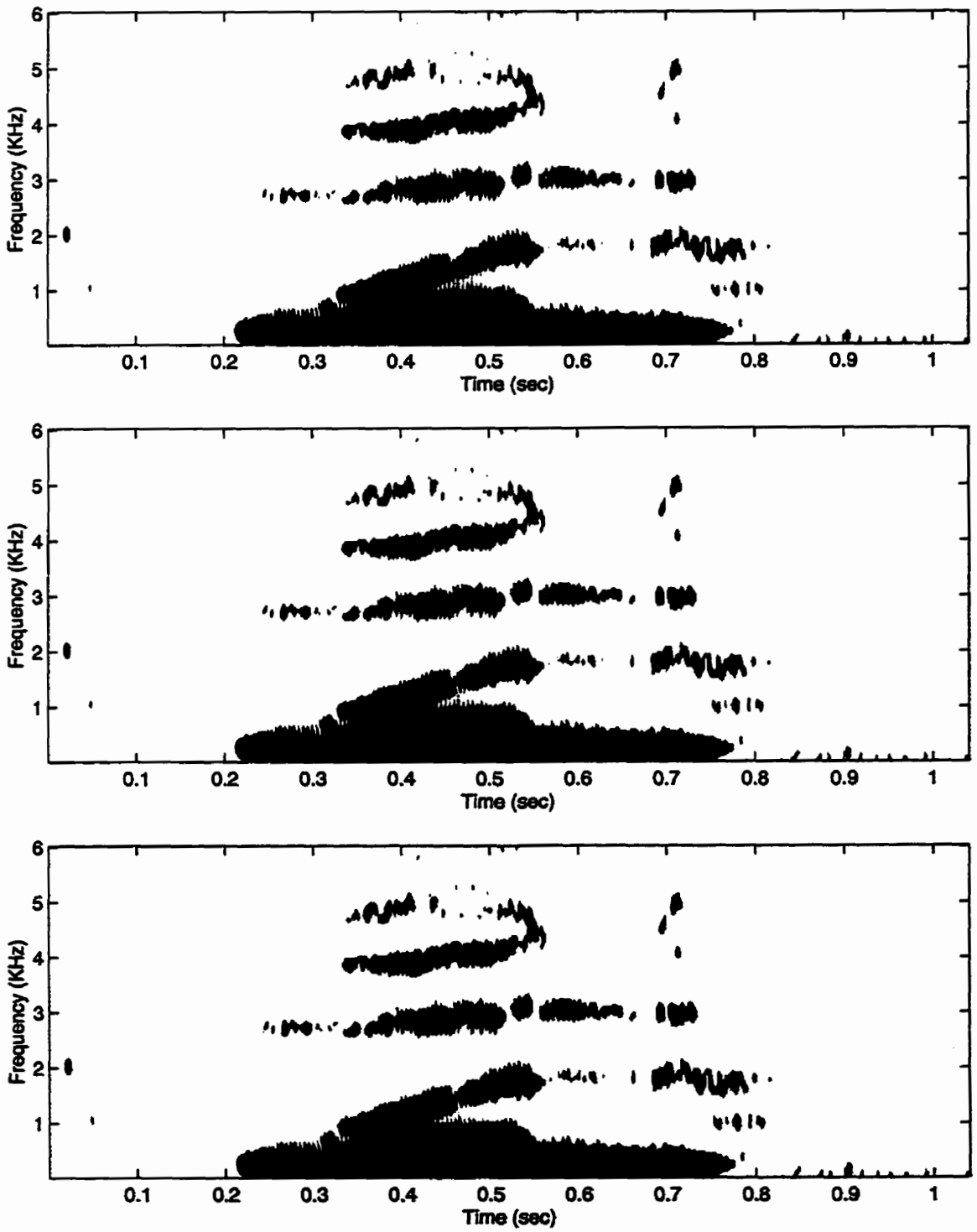


Figure 4.7: Progress of segmentation of word one in different iterations, from top to button, iteration number: 6,7,8

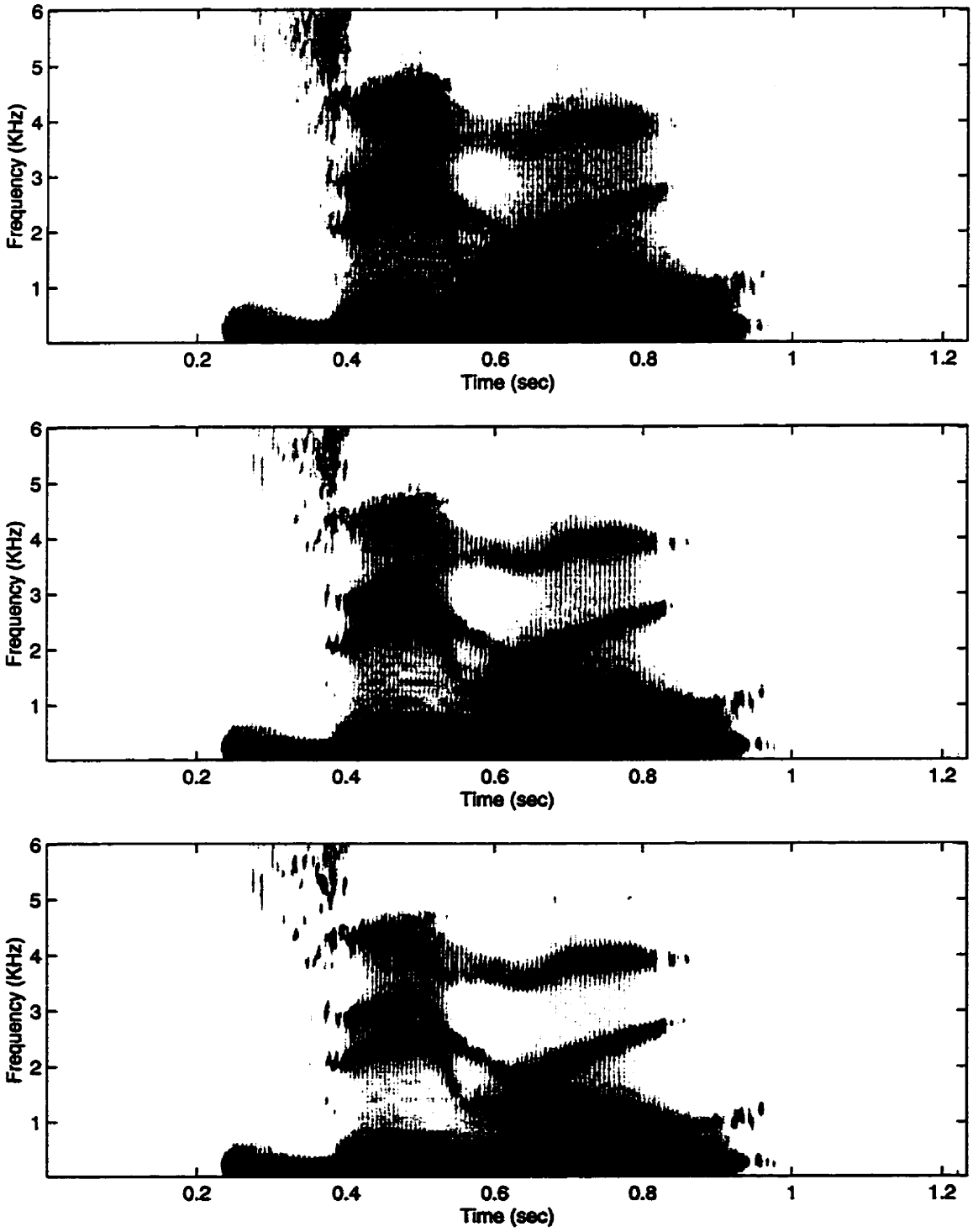


Figure 4.8: Progress of segmentation of word zero in different iterations, from top to button: original spectrogram, iteration number 1, and 2

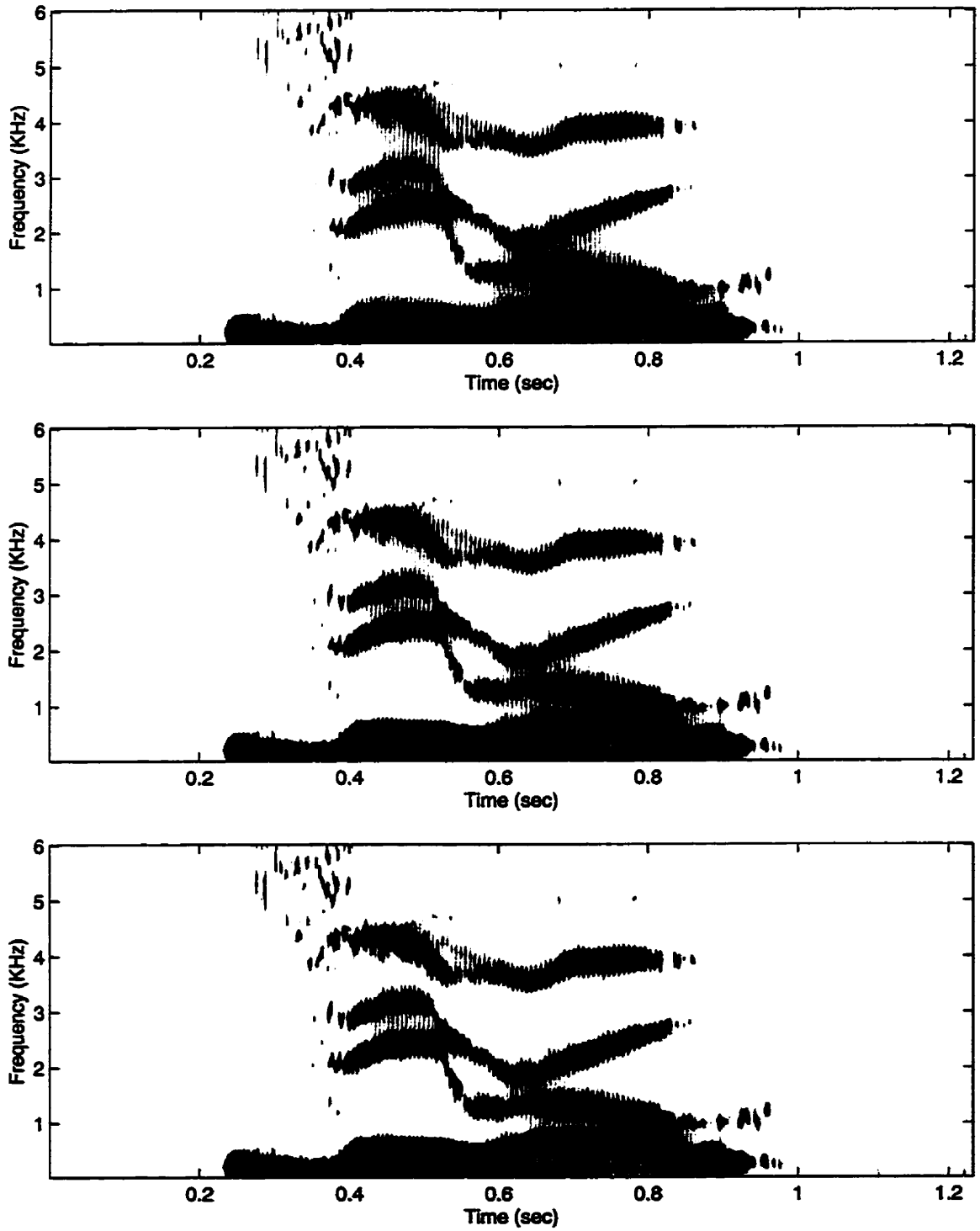


Figure 4.9: Progress of segmentation of word zero in different iterations, from top to button: iteration number 3, 4, and 5

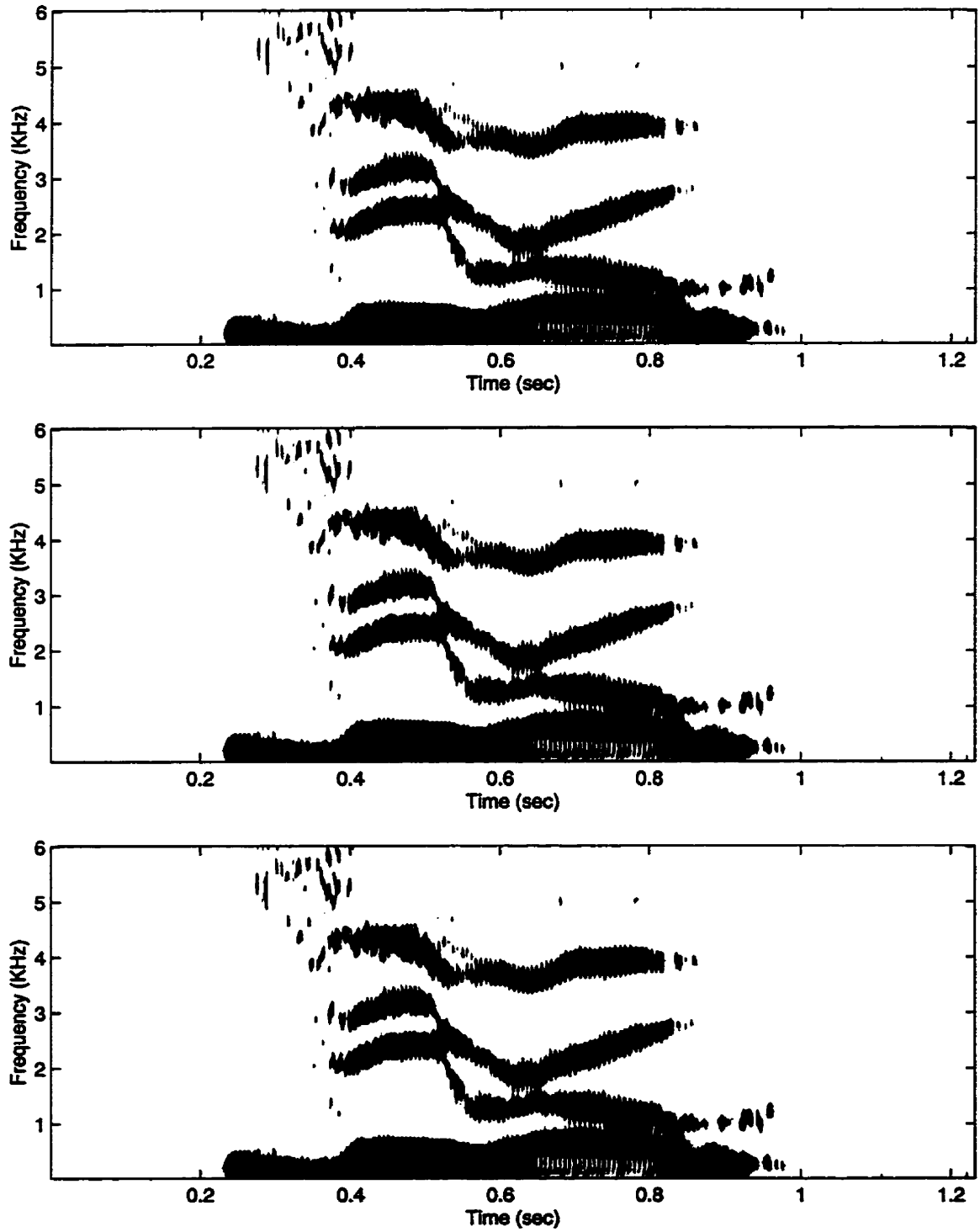


Figure 4.10: Progress of segmentation of word zero in different iterations, from top to button: iteration number 6, 7, and 8

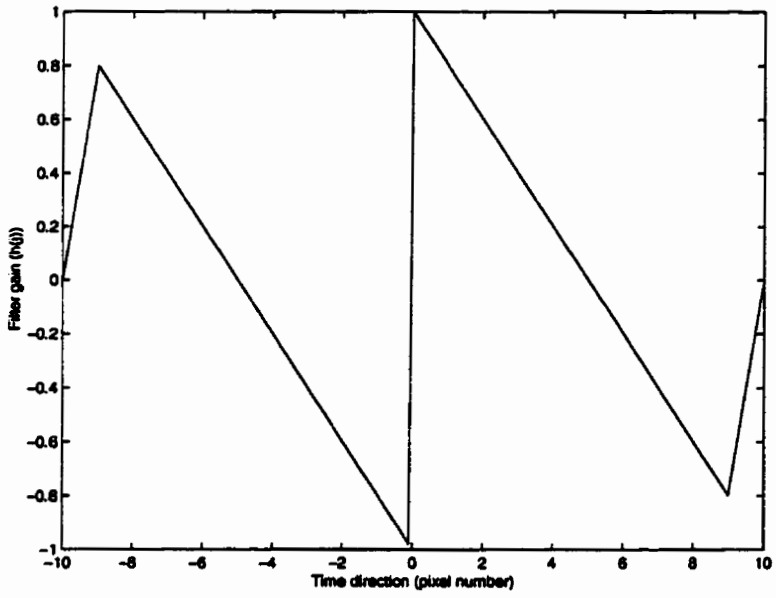


Figure 4.11: The filter that is used for voicing features (each frame represent 10ms)

4.2.2 Voicing features

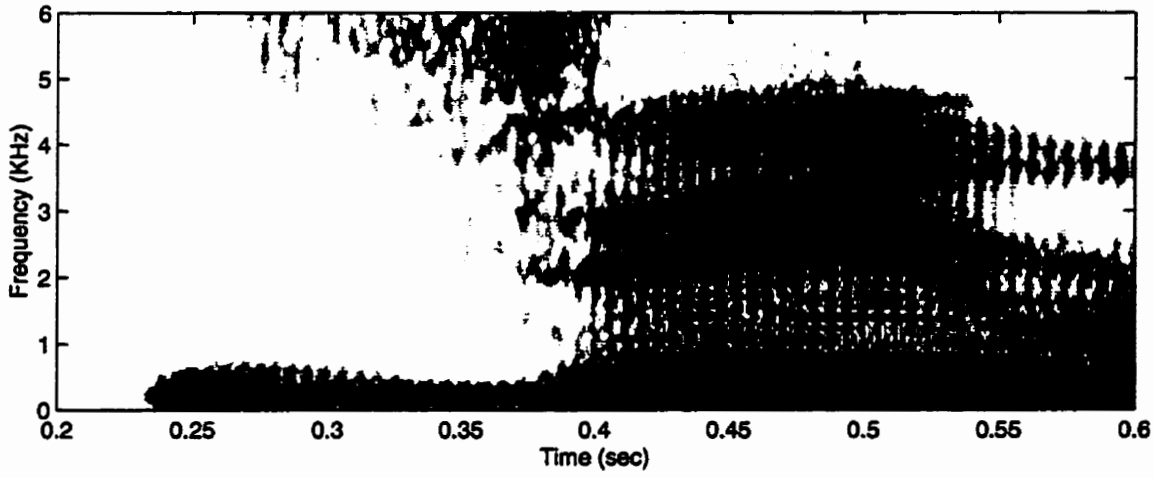
By voicing regions, we refer to regions of spectrogram that show a pencil line pattern having higher energy in a large portion of frequencies for a short period of time. To extract voicing information, we estimate the approximate probabilities as follows:

$$p_{ij} = \sum_{i=-I}^{I} \sum_{j=-J}^{J} h(i) o_{ij}, \tag{4.9}$$

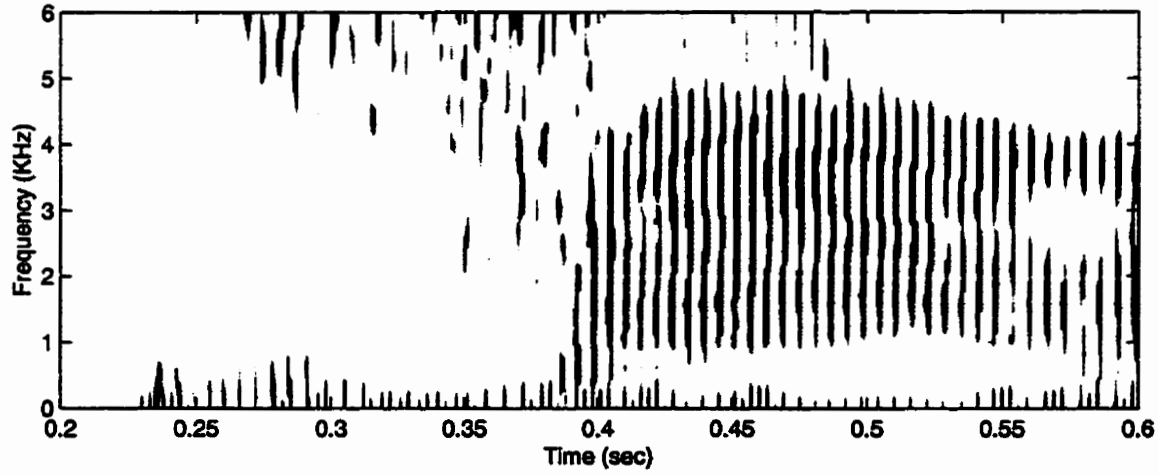
where $h(i)$ is defined as in Fig. 4.11. The adjustment of the probabilities are done using:

$$\Delta p_{ij}^{t+1} = \begin{cases} \alpha \left[\sum_{i=-I}^{I} \sum_{j=-J}^{J} (p_{ij}^{t} - 0.5) \right] & \text{if } p_{ij}^{t} \neq 0 \text{ or } 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

In our experiment $I = 1$ and $J = 11$. Fig. 4.12 shows the resulting images for part of the spectrogram image of word /zero/. Fig. 4.13 shows the resulting segmented image for part of the word /ti/.

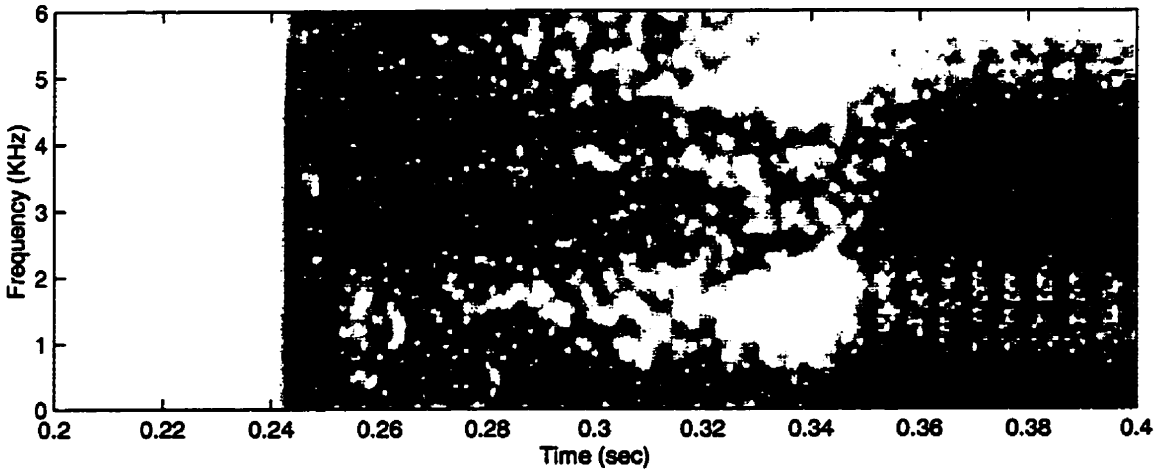


(a)

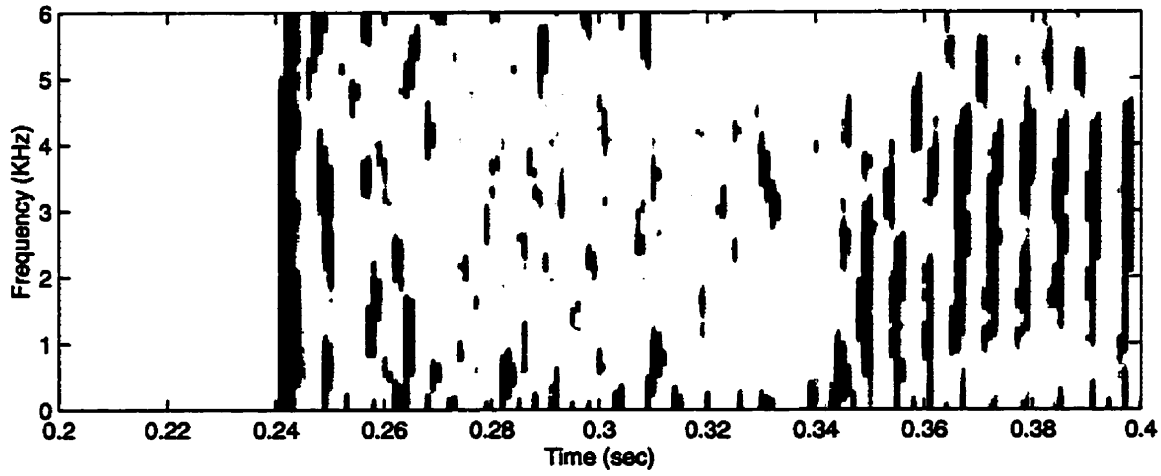


(b)

Figure 4.12: Segmented voicing regions found using part of the spectrogram of the word /zero/: (a) original spectrogram of part of the word /zero/ (b) segmented regions



(a)



(b)

Figure 4.13: Segmented voicing regions found using part of the spectrogram of the word /ti/: (a) part of the original spectrogram (b) the corresponding segmented regions

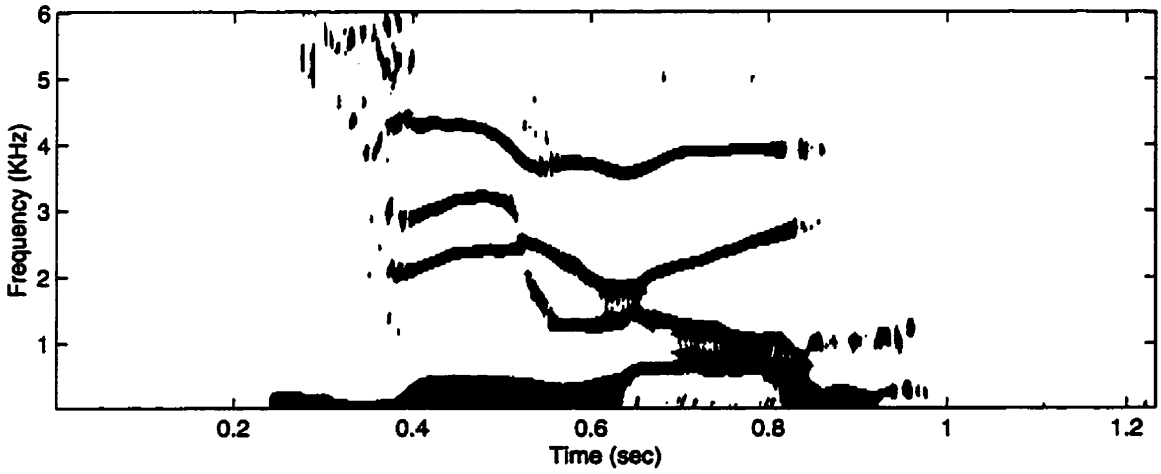


Figure 4.14: Center of gravity of the objects of the segmented image of Fig. 4.10

4.2.3 Rising and falling formats

One of the features that is very important for speech recognition is to know if the formant frequencies are increasing or decreasing. To find that, first the center of gravity for a window of size IxJ was calculated as follows:

$$M_j = \sum_{i=-I}^{I} \sum_{j=-J}^{J} j \cdot p_{i-m,j-n}, \tag{4.11}$$

and

$$M_i = \sum_{i=-I}^{I} \sum_{j=-J}^{J} i \cdot p_{i-m,j-n}. \tag{4.12}$$

In these experiments, $I = 4$ and $J = 8$. Then, the points that their center of gravity are closer to their position were selected. Fig. 4.14 shows the resulting image from the segmented image of Fig. 4.10. To find if the formants are uprising or down-falling, the best regression line that can be passed through each point in a window of size 21x21 was calculated. We separate the points with positive and negative slopes of the regression line into different images. Fig. 4.15 and 4.16 show the resulting images for the segment image of Fig. 4.10.

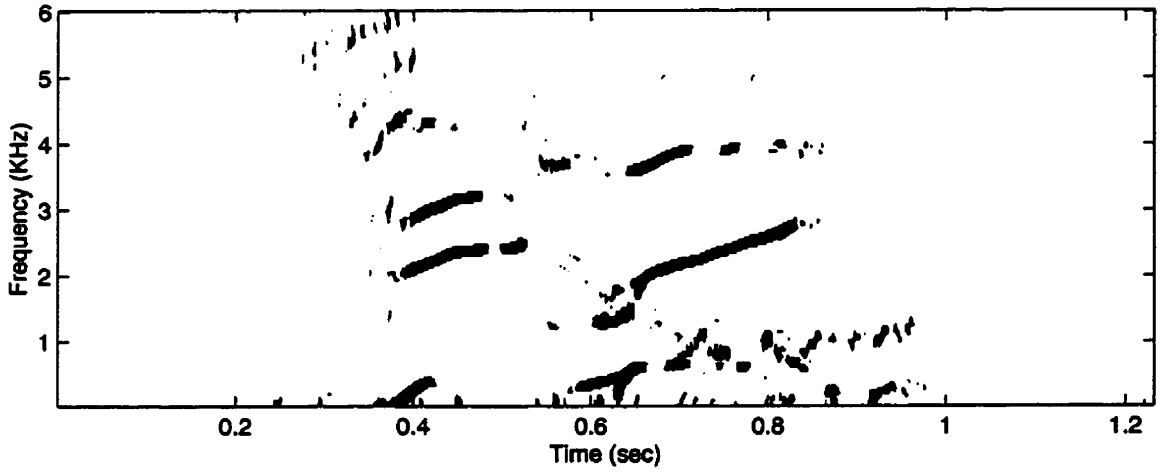


Figure 4.15: Uprising features found from the segmented image in Fig. 4.14

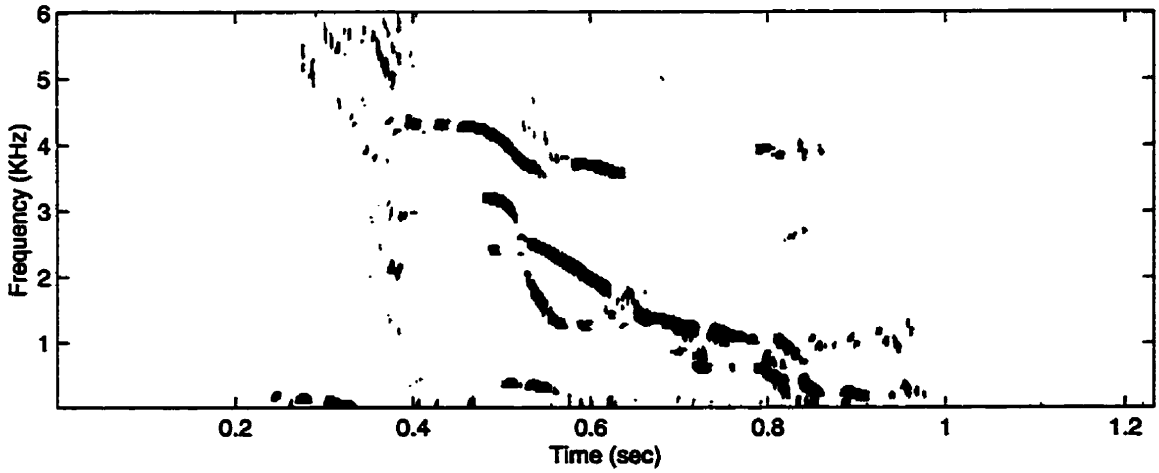


Figure 4.16: Falling formant found from the segmented image in Fig. 4.15

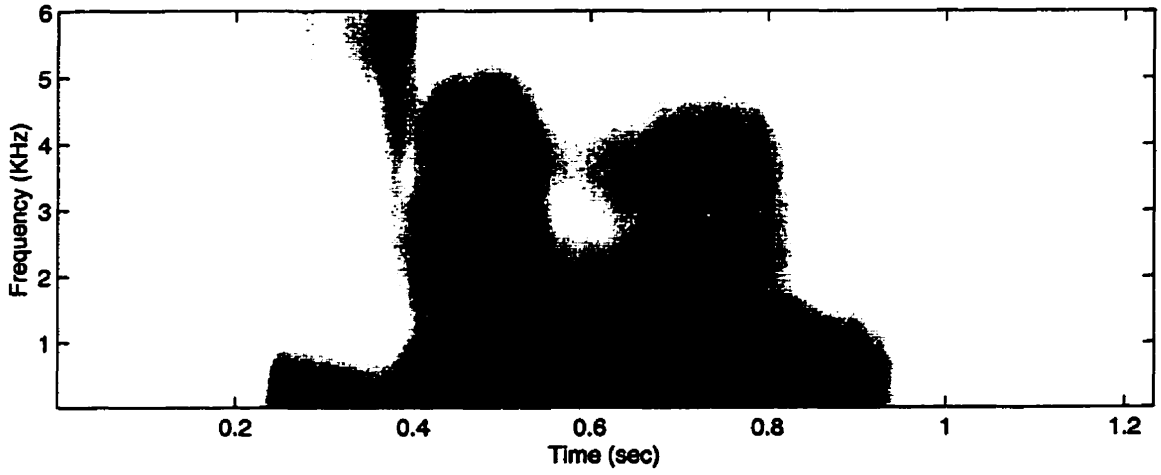


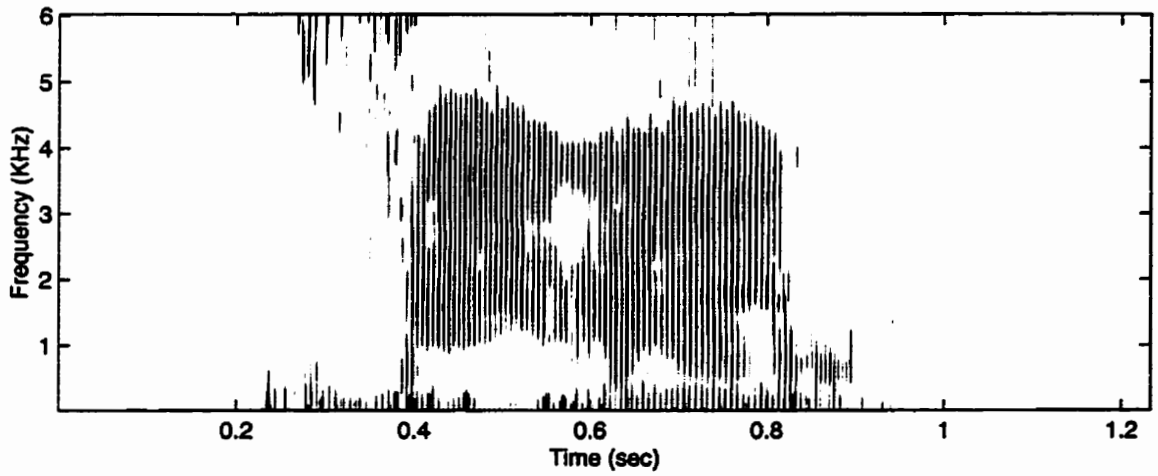
Figure 4.17: Local energy found from the spectrogram in Fig. 4.8

4.2.4 Energy features

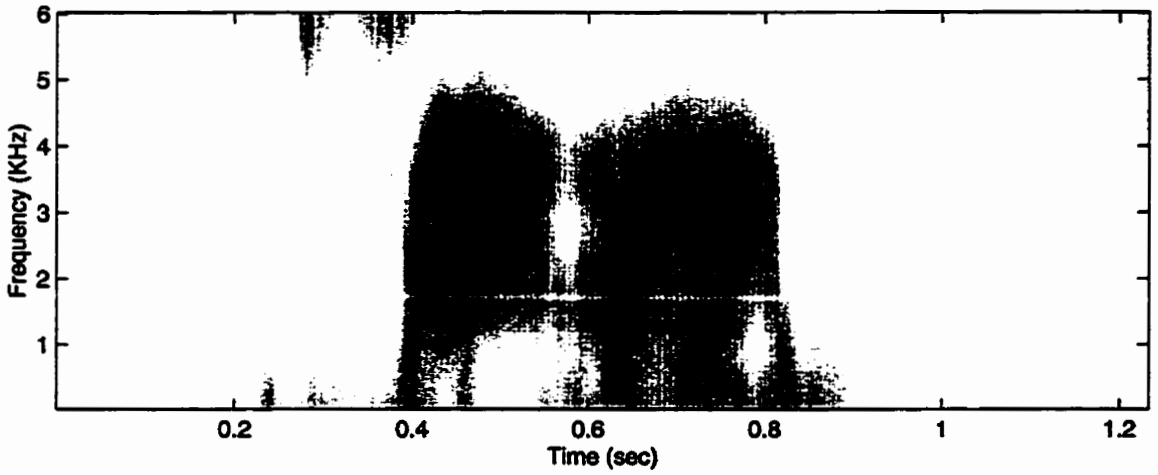
Another set of features are the smooth version of spectrogram. The spectrogram image is averaged over a window of 15x5. Fig. 4.17 shows the resulting image of the spectrogram image of Fig. 4.8. A smooth version of voicing images over a window of 20x8 was also calculated. The resulting image for the spectrogram shown in 4.17 is shown in Fig. 4.18.

4.2.5 Overall energy

Overall energy of signal over a window of 25 frames were also calculated as a single feature.



(a)



(b)

Figure 4.18: (a)Voicing image of the spectrogram in Fig. 4.8(b) its corresponding local energy image

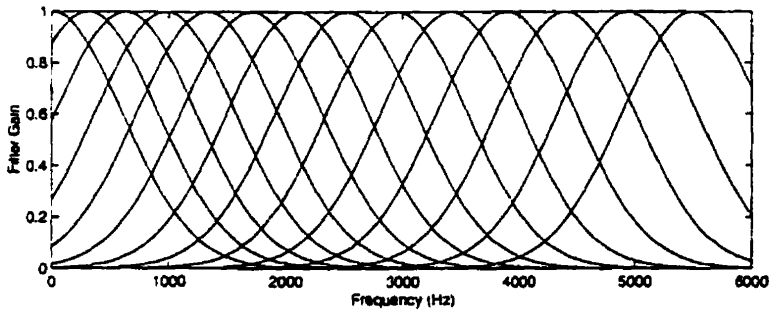


Figure 4.19: The filter that is used for extracting features from segmented images

4.2.6 Filter-banking

After the feature images are calculated, 14 features per frame were extracted using overlapping filters shown in Fig. 4.19.

4.3 Summary

A self-organizing algorithm for segmentation of spectrogram images was proposed. The algorithm first calculates the estimates of the *a posteriori* probabilities of each pixel for object and background classes, and then iteratively adjusts these probabilities to reduce a defined segmentation measure. Pixels that are less likely to belong to object or background classes are adjusted less in each iteration, delaying their segmentation until more image information is available. Experiments showed that the algorithm can be applied to successfully segment formant and voicing regions in the spectrogram image. Other sets of features were also calculated which include: uprising and down-falling formant features, local energy of spectrogram and local energy of segmented voicing images. From each image, and for each frame 14 features were calculated using overlapping filters. These features are the primary set of features as the input to our recognition system.

Chapter 5

Experimental results

5.1 Data base

For the experimental tests, a corpus of isolated spoken words was selected. This corpus was designed and collected at Texas Instruments (TI) in 1980 called TI46 [14], [44]. The material contained on this data base was recorded in a low noise environment.

The TI46 corpus contains 16 speakers: 8 males and 8 females. There are 46 words per speaker. They are: numbers: ZERO to NINE, English letters: A to Z and the words: ENTER, ERASE, GO, HELP, NO, RUBOUT, REPEAT, STOP, START, YES. In all the experiments reported here, 4 samples were used per each speaker in the training set resulting in a total of 64 samples for each word, and 12 samples per speaker were selected for each word of test set resulting in 192 samples per word.

5.2 Segmentation experiments

For any input sample, all the set of images discussed in the previous chapter are computed. From each set of images, 14 features are extracted using the filter banks shown in Fig. 4.19. There are 6 images resulting in 84 features and 1 total energy feature, which results in 85 feature in total for each input.

Based on a priori knowledge of important regions of speech words using spectrogram reading experiences, a state model for each word was designed. The training set of corpus was then hand segmented. Appendix A shows an example of each word and its corresponding state model and segmented regions.

For each word, three models are used for training and testing. These models are called: statistical model, segmentation model, and discriminant classification model. For the statistical model, the parameters are trained using the statistical characteristics of segmented regions of each word. The segmentation model is used for the segmentation of each word and the discriminant classification model is used for discriminant training of model parameters.

After the statistical model parameters are estimated, the segmentation model parameters are initialized with the parameters of the statistical models. There are two phases for the discriminant segmentation training algorithm. In the first phase, the model parameters were trained for 10 iterations. In these experiments, $\xi_1 = 0.01$, $\xi_2 = 0.5$, and the learning rate $\alpha = 0.1$. After this training phase, only 10 directions were selected using our proposed feature selection algorithm. Fig. 5.1 shows a typical progress of overall cost function for the word /zero/ during this training phase. Fig. 5.2 shows the accumulative projection of $\gamma(\xi_1 d_t \alpha_t q_t)\vec{N}_t$ for a typical state of the model versus the number of directions that can be used. Note that the direction numbers are sorted based on their importance. In the second

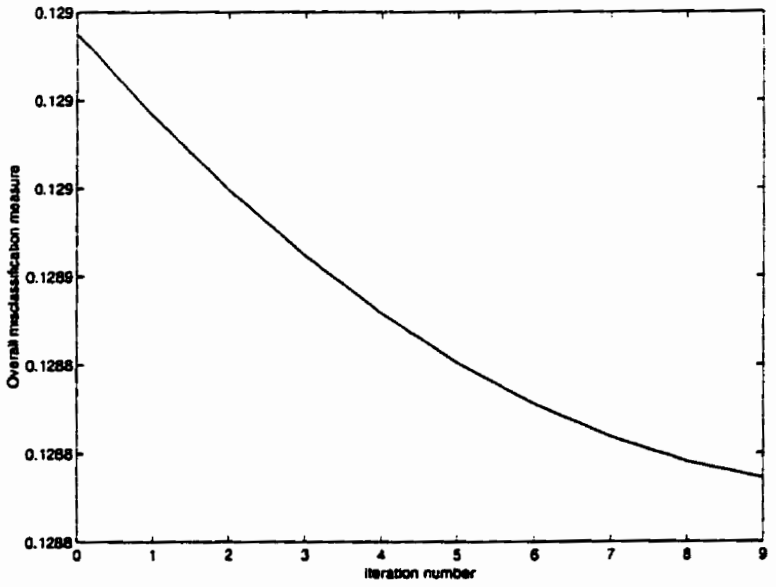


Figure 5.1: The progress of overall misclassification measure during the first phase of segmentation training for word /zero/

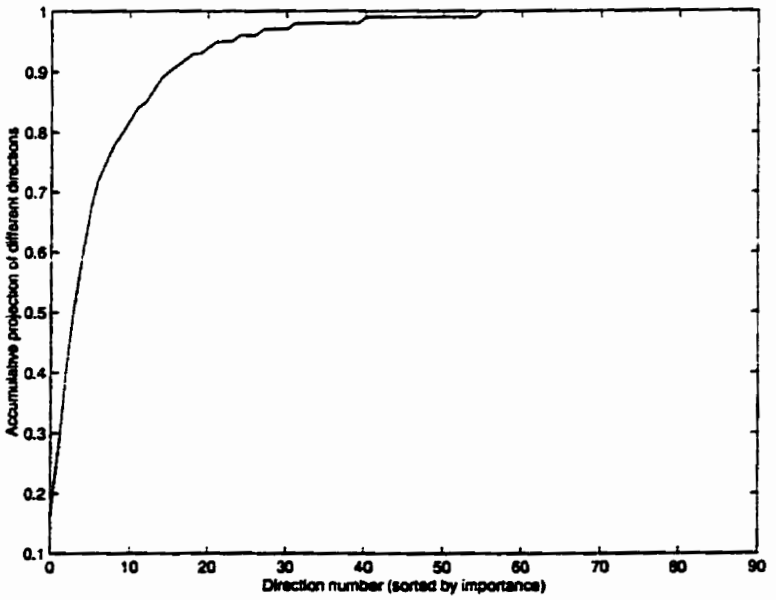


Figure 5.2: The accumulative projection of $\gamma(\xi_1 d_t \alpha_t q_t)\bar{N}_t$ for a typical state of the model after the first phase of segmentation training for word /zero/

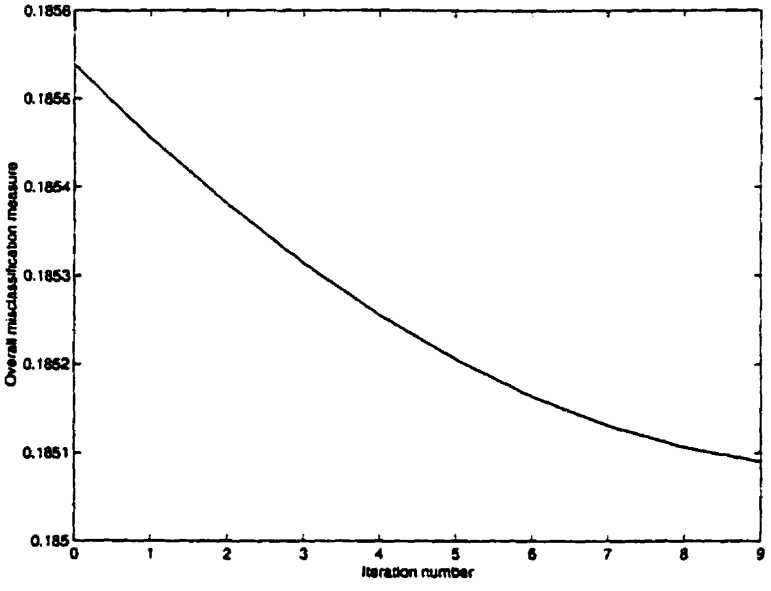
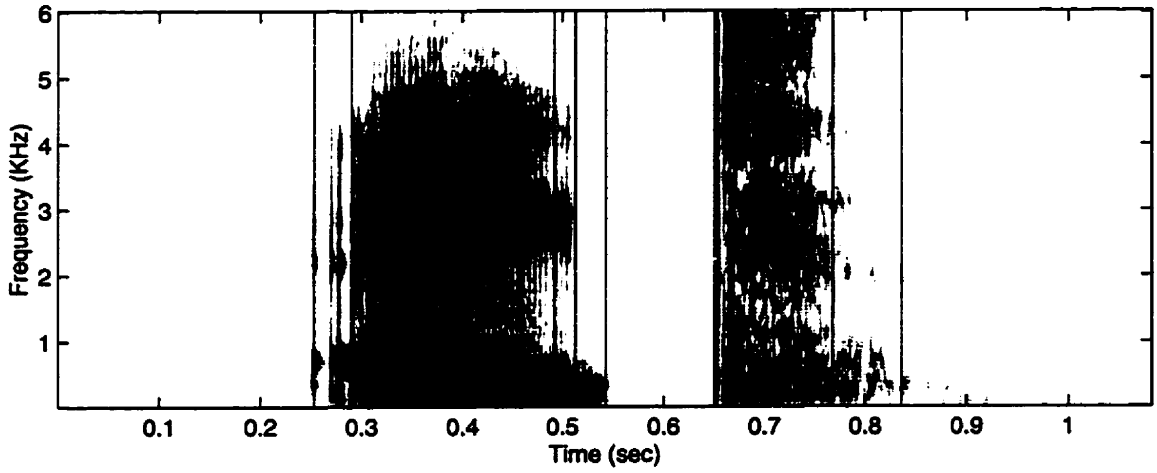


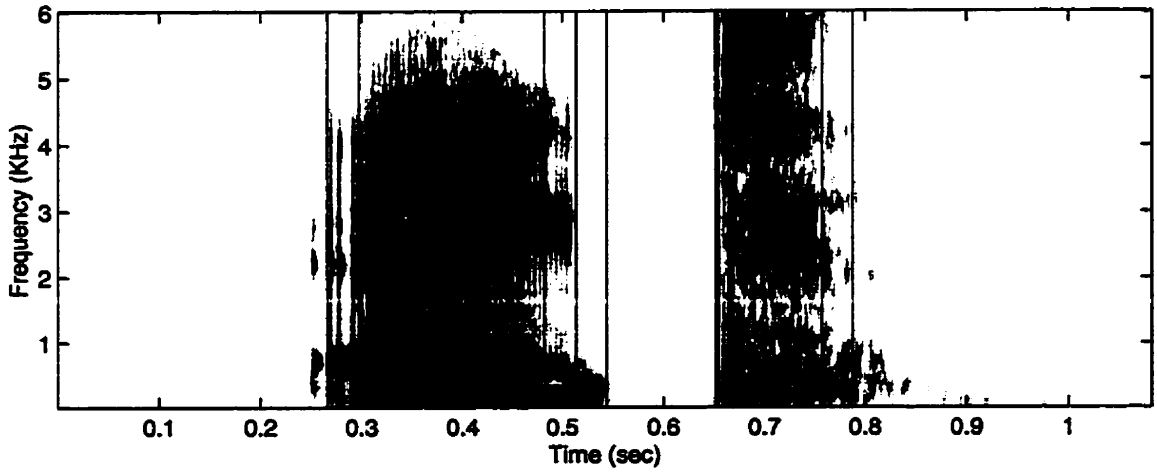
Figure 5.3: The progress of overall misclassification measure during the second phase of segmentation training for word /zero/

phase of segmentation training, the training of model parameters was continued, but this time only 10 directions per state was used. The training continued for 10 more iterations. Fig. 5.3 shows the progress of overall cost during training for word /zero/. After the second training phase, an orthogonal transforms for each state is calculated and the dimensionality of input is mapped to only 5 directions. The resulting models were used for segmentation of input samples. Fig. 5.4 compares the segmentation of the word /eight/ done before training with hand and the segmentation results of segmentation model of /eight/. In practice, the segmentation result by models usually outperform that of hand segmentation, as the models have a better estimate of overall statistical characteristics of each state.

After the segmentation models are trained, all the words in training set are segmented by the segmentation models of all classes using Viterbi beam algorithm.



(a)



(b)

Figure 5.4: (a)Hand segmented regions of word /eight/ before training(b) the segmented regions resulted from the segmentation model

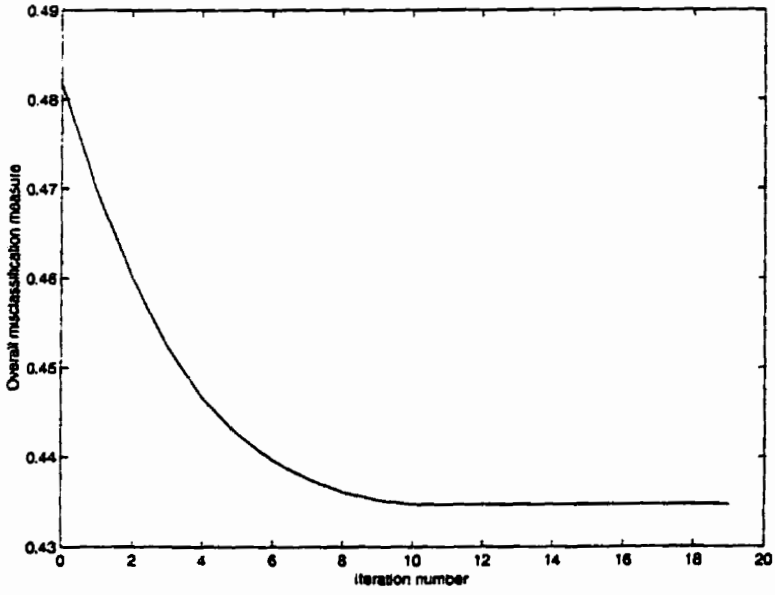


Figure 5.5: The progress of overall cost function during the first phase of discriminant training for word /B/

The maximum beam size of 50 was selected in this algorithm. The resulting sequences was then recored in the data base for later references.

5.3 Training of discriminant classification models

The training phase of discriminant classification models starts by initializing the models by the statistical models. Again, a similar procedure as in segmentation training phase was carried out. First, the model parameters were trained for 50 iterations using the discriminant training algorithm, and then the feature selection algorithm was applied. At this stage only 20 features out of 85 features were selected for each state. Again the training was continued for 20 iterations and then the feature extraction algorithm was applied to transform the 20 features to extract

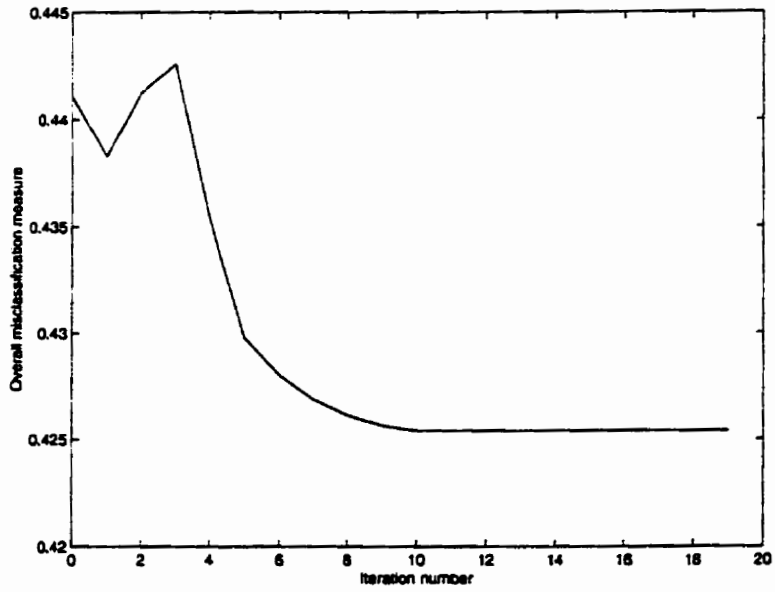


Figure 5.6: The progress of overall cost function during the second phase of discriminant training for word /B/

	b	d	g	p	t	recognition rate
b	60	2	0	2	0	class0=93.75%
d	9	52	0	3	0	class1=81.25%
g	0	0	59	3	2	class2=92.19%
p	0	0	1	62	1	class3=96.88%
t	0	0	4	5	55	class4=85.94%
average recognition rate						90.00%

Table 5.1: Recognition result on training set using the statistical model in the higher dimensional space

10 features per state. Fig. 5.5 and Fig. 5.6 show the progress of overall cost during each phase of training. For comparison, the input features were mapped using KL-expansion algorithm to 20 dimensions and the statistical model and discriminative classification model were trained. The following reports on the results.

Here, the result of classification on the confusing set /bi/, /di/, /gi/, /pi/, and /ti/ are reported. Table 5.1 and 5.2 show the recognition results for the statistical model (in the higher dimensional space) on training and test sets, respectively. Table 5.3 and 5.4 shows the results for the discriminant training algorithm and after reduction of dimensionality on the training and test sets, respectively. Table 5.5 and 5.6 show the results of statistical model in the reduced feature set on training and test sets, respectively. Table 5.7 and 5.8 show the results of discriminative classification model in the reduced feature set on training and test sets, respectively. As can be verified by the results, discriminant training and feature extraction methods can improve the recognition rate of these confusing classes while having a reduced set of parameters for the classifiers and much lower computational cost.

	b	d	g	p	t	recognition rate
b	97	27	0	4	0	class0=75.78%
d	16	90	7	15	0	class1=70.31%
g	0	0	113	5	10	class2=88.28%
p	1	1	8	106	12	class3=82.81%
t	0	1	28	14	85	class4=66.41%
average recognition rate						76.72%

Table 5.2: Recognition result on test set using the statistical model in the higher dimensional space

	b	d	g	p	t	recognition rate
b	60	4	0	0	0	class0=93.75%
d	5	58	0	1	0	class1=90.62%
g	0	0	62	0	2	class2=96.88%
p	7	2	0	52	3	class3=81.25%
t	0	0	0	1	63	class4=98.44%
average recognition rate						92.19%

Table 5.3: Recognition result on train set after discriminant training and reduction of dimensionality (the proposed approach)

	b	d	g	p	t	recognition rate
b	105	22	0	1	0	class0=82.03%
d	9	114	2	2	1	class1=89.06%
g	0	0	125	1	2	class2=97.66%
p	10	5	1	93	19	class3=72.66%
t	0	1	10	2	115	class4=89.84%
average recognition rate						86.25%

Table 5.4: Recognition result on test set after discriminant training and reduction of dimensionality (the proposed approach)

	b	d	g	p	t	recognition rate
b	52	10	0	2	0	class0=81.25%
d	12	50	2	0	0	class1=78.12%
g	0	2	57	1	4	class2=89.06%
p	2	0	1	58	3	class3=90.62%
t	0	0	1	5	58	class4=90.62%
average recognition rate						85.94%

Table 5.5: Recognition result on training set for the reduced dimensionality found by KL-expansion algorithm and by training the statistical model

	b	d	g	p	t	recognition rate
b	92	33	0	3	0	class0=71.88%
d	18	105	4	0	1	class1=82.03%
g	1	4	112	3	8	class2=87.50%
p	12	0	3	100	13	class3=78.12%
t	2	1	23	6	96	class4=75.00%
average recognition rate						78.91%

Table 5.6: Recognition result on test set for the reduced dimensionality found by KL-expansion algorithm and by training the statistical model

	b	d	g	p	t	recognition rate
b	59	4	0	1	0	class0=92.19%
d	7	56	1	0	0	class1=87.50%
g	0	1	61	1	1	class2=95.31%
p	0	0	1	62	1	class3=96.88%
t	0	0	1	0	63	class4=98.44%
average recognition rate						94.06%

Table 5.7: Recognition result on training set for the reduced dimensionality found by KL-expansion algorithm and by training the discriminative model

	b	d	g	p	t	recognition rate
b	102	24	0	2	0	class0=79.69%
d	10	112	1	3	2	class1=87.50%
g	1	6	116	1	4	class2=90.62%
p	8	4	1	97	18	class3=75.78%
t	0	2	9	3	114	class4=89.06%
average recognition rate						84.53%

Table 5.8: Recognition result on test set for the reduced dimensionality found by KL-expansion algorithm and by training the discriminative model

The next set of experiments were carried out on digits /zero/ to /nine/. Table 5.9 and 5.10 show the recognition results for the statistical model (in the higher dimensional space) on training and test sets, respectively. Table 5.11 and 5.12 shows the results for the discriminant training algorithm and after reduction of dimensionality on the training and test sets, respectively. Table 5.13 and 5.14 show the results of statistical model in the reduced feature set on training and test sets, respectively. Table 5.15 and 5.16 show the results of discriminative classification model in the reduced feature set on training and test sets, respectively. This set of experiments also validate the improved performance of the proposed algorithm.

	zero	one	two	three	four	five	six	seven	eight	nine	recognition rate
zero	59	0	0	0	0	5	0	0	0	0	class0=92.19%
one	1	58	0	0	0	3	0	0	0	2	class1=90.62%
two	4	0	50	0	3	4	0	3	0	0	class2=78.12%
three	0	0	0	63	0	0	0	0	1	0	class3=98.44%
four	0	0	0	0	64	0	0	0	0	0	class4=100.00%
five	0	0	0	0	0	64	0	0	0	0	class5=100.00%
six	0	0	0	0	0	4	60	0	0	0	class6=93.75%
seven	1	0	0	0	0	10	0	53	0	0	class7=82.81%
eight	0	0	0	0	0	6	3	0	55	0	class8=85.94%
nine	0	1	0	0	0	12	0	0	0	51	class9=79.69%
average recognition rate											90.16%

Table 5.9: Recognition result on training set using the statistical model in the higher dimensional space

	zero	one	two	three	four	five	six	seven	eight	nine	recognition rate
zero	171	0	0	1	8	12	0	0	0	0	class0=89.06%
one	0	170	0	0	0	20	0	0	0	2	class1=88.54%
two	4	0	158	3	2	18	0	7	0	0	class2=82.29%
three	0	0	0	185	0	7	0	0	0	0	class3=96.35%
four	0	0	0	0	191	1	0	0	0	0	class4=99.48%
five	0	0	0	0	0	190	0	0	0	2	class5=98.96%
six	0	0	0	0	0	10	182	0	0	0	class6=94.79%
seven	2	0	0	1	1	19	0	169	0	0	class7=88.02%
eight	0	0	0	0	0	25	8	0	159	0	class8=82.81%
nine	0	4	0	3	0	61	0	0	0	124	class9=64.58%
average recognition rate											88.49%

Table 5.10: Recognition result on test set using the statistical model in the higher dimensional space

	zero	one	two	three	four	five	six	seven	eight	nine	recognition rate
zero	64	0	0	0	0	0	0	0	0	0	class0=100.00%
one	0	64	0	0	0	0	0	0	0	0	class1=100.00%
two	0	0	64	0	0	0	0	0	0	0	class2=100.00%
three	0	0	0	64	0	0	0	0	0	0	class3=100.00%
four	0	0	0	0	64	0	0	0	0	0	class4=100.00%
five	0	3	0	0	0	61	0	0	0	0	class5=95.31%
six	0	0	0	0	0	0	64	0	0	0	class6=100.00%
seven	0	0	0	0	0	0	0	64	0	0	class7=100.00%
eight	0	0	0	0	0	0	0	0	64	0	class8=100.00%
nine	0	0	0	0	0	1	0	0	0	63	class9=98.44%
average recognition rate											99.38%

Table 5.11: Recognition result on train set after discriminant training and reduction of dimensionality (the proposed approach)

	zero	one	two	three	four	five	six	seven	eight	nine	recognition rate
zero	190	0	0	0	1	0	0	1	0	0	class0=98.96%
one	0	191	0	0	0	1	0	0	0	0	class1=99.48%
two	1	0	191	0	0	0	0	0	0	0	class2=99.48%
three	1	0	0	187	0	0	0	0	4	0	class3=97.40%
four	0	1	0	0	191	0	0	0	0	0	class4=99.48%
five	0	0	0	0	0	187	0	0	0	5	class5=97.40%
six	0	0	0	0	0	0	190	0	2	0	class6=98.96%
seven	2	0	0	0	0	2	0	188	0	0	class7=97.92%
eight	0	0	0	1	0	0	3	0	188	0	class8=97.92%
nine	0	6	0	2	0	3	0	1	0	180	class9=93.75%
average recognition rate											98.07%

Table 5.12: Recognition result on test set after discriminant training and reduction of dimensionality (the proposed approach)

	zero	one	two	three	four	five	six	seven	eight	nine	recognition rate
zero	63	0	0	0	0	0	0	1	0	0	class0=98.44%
one	0	62	0	0	0	0	0	0	0	2	class1=96.88%
two	0	0	64	0	0	0	0	0	0	0	class2=100.00%
three	0	0	0	61	0	0	0	0	3	0	class3=95.31%
four	1	1	0	0	59	3	0	0	0	0	class4=92.19%
five	1	3	0	0	0	59	0	0	0	1	class5=92.19%
six	0	0	0	0	0	0	63	0	1	0	class6=98.44%
seven	2	0	0	0	0	0	0	62	0	0	class7=96.88%
eight	0	0	0	0	0	0	0	0	64	0	class8=100.00%
nine	1	13	0	0	0	1	0	0	0	49	class9=76.56%
average recognition rate											94.69%

Table 5.13: Recognition result on training set for the reduced dimensionality found by KL-expansion algorithm and by training the statistical model

	zero	one	two	three	four	five	six	seven	eight	nine	recognition rate
zero	185	0	2	0	3	1	0	1	0	0	class0=96.35%
one	0	184	0	0	0	1	0	0	0	7	class1=95.83%
two	0	0	192	0	0	0	0	0	0	0	class2=100.00%
three	0	1	3	184	0	2	0	0	2	0	class3=95.83%
four	4	5	1	0	181	1	0	0	0	0	class4=94.27%
five	1	1	0	0	2	176	0	1	0	11	class5=91.67%
six	1	0	0	0	0	2	186	0	3	0	class6=96.88%
seven	7	0	0	0	0	2	0	183	0	0	class7=95.31%
eight	0	0	0	0	0	5	3	0	184	0	class8=95.83%
nine	0	38	0	1	0	2	0	1	0	150	class9=78.12%
average recognition rate											94.01%

Table 5.14: Recognition result on test set for the reduced dimensionality found by KL-expansion algorithm and by training the statistical model

	zero	one	two	three	four	five	six	seven	eight	nine	recognition rate
zero	63	0	0	0	0	0	0	1	0	0	class0=98.44%
one	0	63	0	0	0	0	0	0	0	1	class1=98.44%
two	0	0	62	0	0	0	1	0	1	0	class2=96.88%
three	0	0	0	59	0	0	0	0	5	0	class3=92.19%
four	0	0	0	0	61	3	0	0	0	0	class4=95.31%
five	1	1	0	0	0	62	0	0	0	0	class5=96.88%
six	0	0	0	0	0	0	63	0	1	0	class6=98.44%
seven	2	0	0	0	0	0	0	62	0	0	class7=96.88%
eight	0	0	0	0	0	0	1	0	63	0	class8=98.44%
nine	0	4	0	0	0	0	0	0	0	60	class9=93.75%
average recognition rate											96.56%

Table 5.15: Recognition result on training set for the reduced dimensionality found by KL-expansion algorithm and by training the discriminative model

	zero	one	two	three	four	five	six	seven	eight	nine	recognition rate
zero	179	0	7	0	2	0	0	3	0	1	class0=93.23%
one	0	183	0	0	0	1	0	0	0	8	class1=95.31%
two	0	0	191	0	0	0	0	1	0	0	class2=99.48%
three	0	0	4	179	0	1	0	0	8	0	class3=93.23%
four	0	1	0	0	190	1	0	0	0	0	class4=98.96%
five	0	1	0	0	1	183	0	2	0	5	class5=95.31%
six	0	0	0	2	0	4	182	0	4	0	class6=94.79%
seven	1	0	0	0	0	1	0	190	0	0	class7=98.96%
eight	0	0	0	2	0	1	3	0	186	0	class8=96.88%
nine	0	18	0	2	0	4	0	0	0	168	class9=87.50%
average recognition rate											95.36%

Table 5.16: Recognition result on test set for the reduced dimensionality found by KL-expansion algorithm and by training the discriminative model

Chapter 6

Summary and conclusion

The motivations behind this thesis were inspired based on the following three facts. First, speech signal is produced by our articulatory apparatus which has inherent physical constraints in the production mechanism, and as a result statistical constraints are imposed in the pattern of speech. Second, speech units such as words or sentences are produced by human in a way that they can be recognizable by human recognition system. Third, although speech units may have a high degree of variability in their patterns, their differences are more easily measurable when compared with each other pairwise.

Most of the features extracted in this thesis use speech spectrogram. Based on speech spectrogram reading experiences, speech units that sound differently, have measurable differences in their spectrogram patterns. In this thesis, I tried to measure such differences with emphasis on those features that are more important in the classification of more confusing classes using image processing techniques.

Extracting discriminative features from spectrogram for different speech units results in a large dimensionality of input vector for each frame of speech. This

is due to the fact that features that may be good for a classification task, may not be good for other classification tasks. As a result of high dimensionality of input space, design of classifiers becomes difficult mainly due to three factors: lack of enough training data, improper choice of models, and lack of an appropriate training algorithm. It may also be computationally impractical for real time speech recognition to build classifiers in higher dimensional space (if the classifiers can not be implemented using parallel processors). Design of feature selection and feature extraction based on minimizing the probability of error is also a difficult problem for the same aforementioned reason for the design of classifiers.

Instead of minimizing the probability of error, the proposed feature selection and extraction algorithms use a classifier trained in the higher dimensional space as a measure of class separability. This was achieved by introduction of a new form of classifiers for speech recognition along with a new discriminative training algorithm. In the training algorithm, first a new form of misclassification measure was defined, and then this measure was minimized over the training set. The misclassification measure was shown to be a smooth version of probability of error. It was shown that the change in the misclassification measure (or indirectly the probability of correct classification) for the proposed feature selection and extraction algorithms was bounded. It was also shown that such a bound could be minimized. This was achieved by properly presenting vectors perpendicular to decision hyperplanes weighted by their share of misclassification measure in the lower dimensional space. This in turn resulted in the proposed state dependent feature selection and extraction algorithm.

It was also shown through speaker independent experiments that classification of confusing classes can be much better achieved by the proposed algorithm. Also, it was shown that classification can be achieved in a low dimensional space with

better classification rate.

This thesis also suggests that the problem of speech recognition should be partly addressed in the feature selection and extraction stages. The algorithms proposed here can be practically implemented using parallel processors and the computations left for classifiers are highly reduced. This suggests that increasing the parallel processing ability of computers is more important and economically cheaper than increasing their speed for the purpose of speech recognition.

6.0.1 Contributions

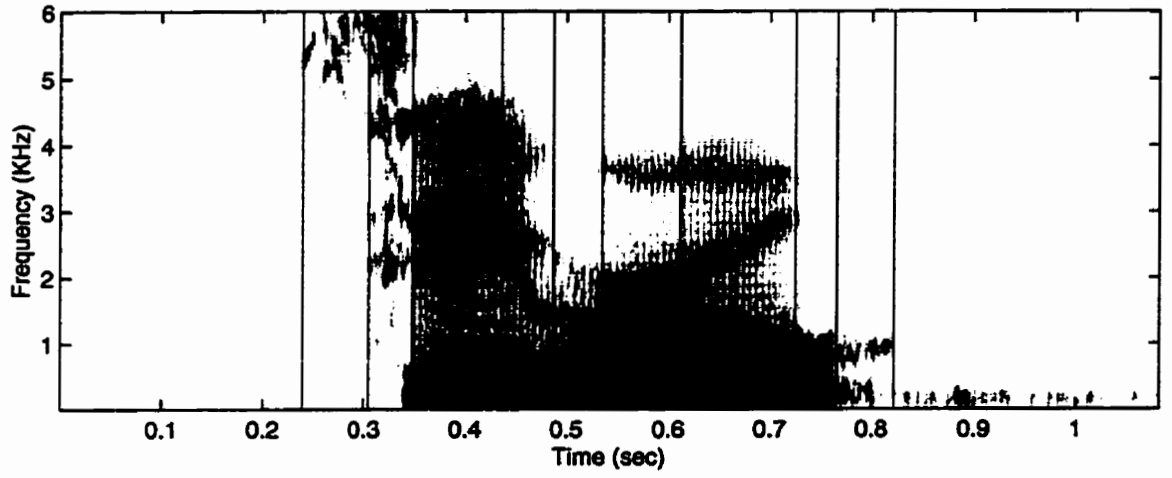
The contribution of this thesis are as follows:

- Introduction of a new discriminative training algorithm for the design of speech classifiers with emphasis on a specifically designed state model.
- Introduction of a new discriminative segmentation algorithm for segmentation of speech utterances to states of the proposed model.
- Introduction of a new feature selection and extraction algorithm based on minimizing the misclassification measure of classifiers built in higher dimensional space.
- Introduction of a new self-organizing image segmentation algorithm for feature extraction from speech spectrogram.

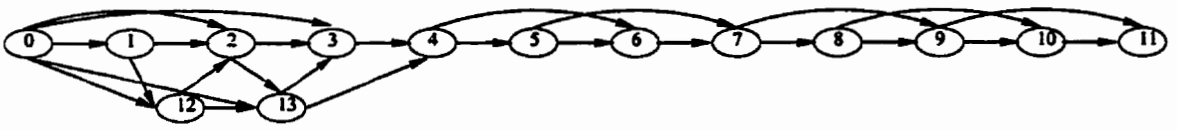
Appendix A

State models

In this appendix state models used in the experiments are shown. We also provide a sample of spectrogram of each word and its corresponding hand segmented regions.



(a)

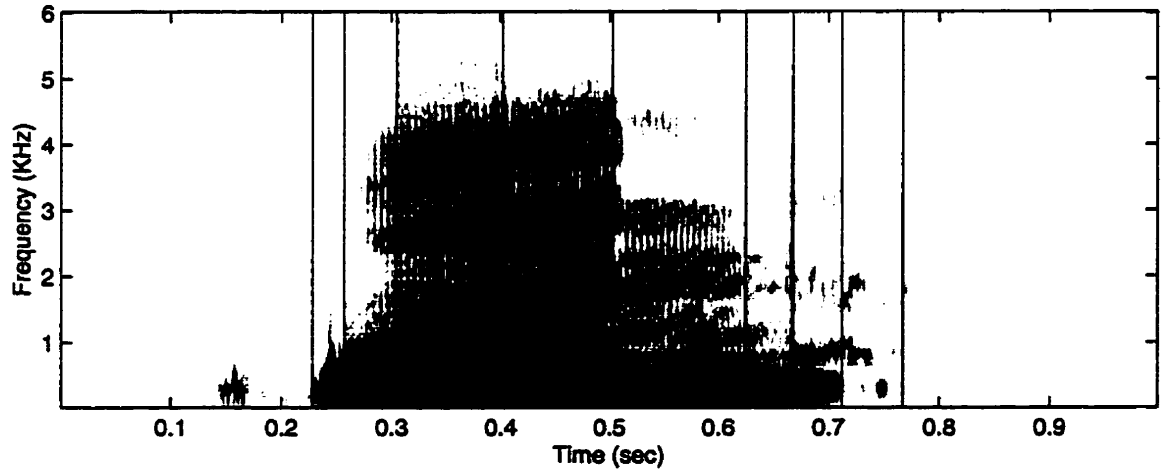


(b)

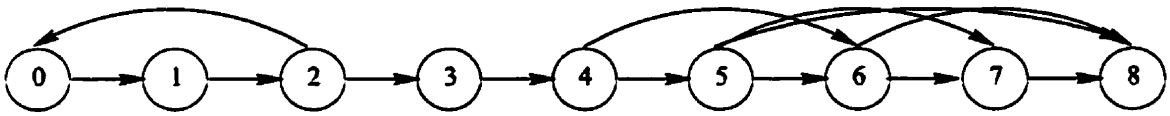
Figure A.1: (a) A sample of segmented word /zero/ (b) state model of word /zero/

0,11	silence
1	voice bar only
2	voice bar + friction of /z/
3	voice bar + friction of /z/ + F2 + F3
4	vowel /i/ as in /zero/
5	transition /i/ to /r/, F2 going down
6	/r/ F3 low
7	F3 going up, transition /r/ to /o/
8	F3 much higher than F2 in /o/
9	F1 and F2 in /o/ present
10	whisper mode of /o/
12	friction of /z/ without voice bar
13	friction of /z/ without voice bar + F2 + F3

Table A.1: Important features associated to states of model /zero/



(a)

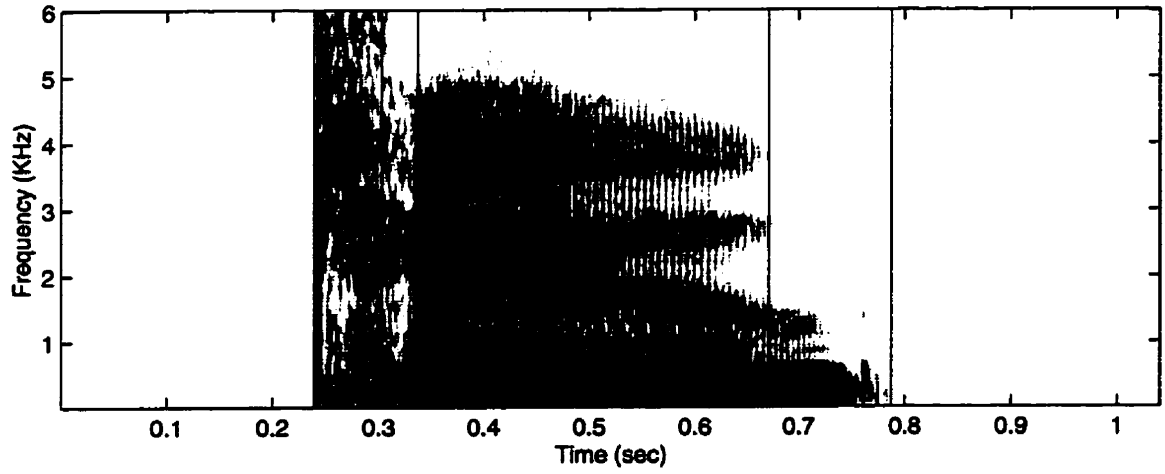


(b)

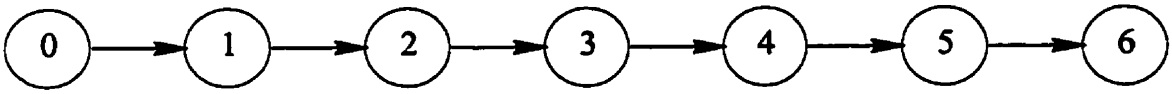
Figure A.2: (a) A sample of segmented word /one/ (b) state model of word /one/

0,8	silence
1	F1 +F2 low in /w/
2	F1 +F2 low + F3 in /w/
3	F1 low + F2 high + F3 (transition /w/ to /n/)
4	/n/
5	/n/ with voice bar only
6	/n/ + aspiration
7	/n/ in whisper mode
9	voice bar

Table A.2: Important features associated to states of model /one/



(a)

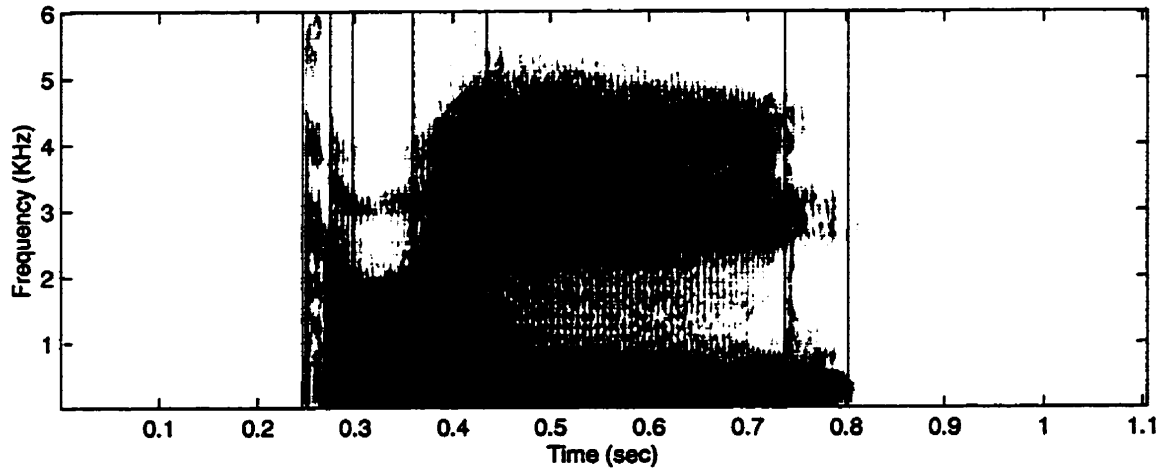


(b)

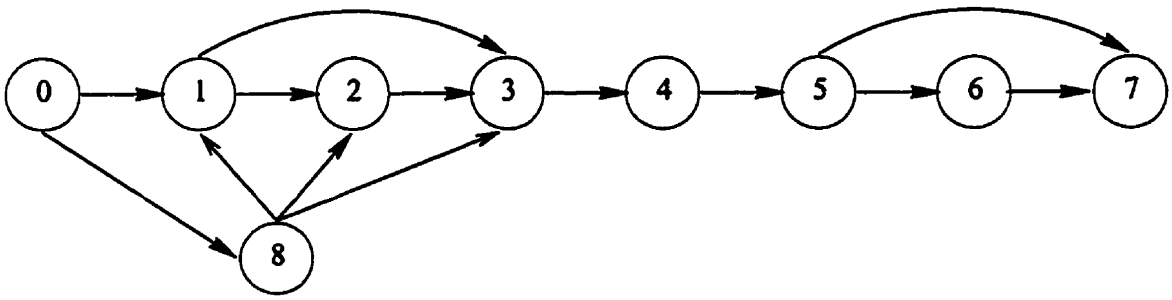
Figure A.3: (a) A sample of segmented word /two/ (b) state model of word /two/

0, 6	silence
1	pencil line
2	aspiration of /t/ with high energy
3	aspiration of /t/ with low energy
4	transition of /t/ to /o/ (F2 high)
5	/o/, F1 +F2, F3 not present

Table A.3: Important features associated to states of model /two/



(a)

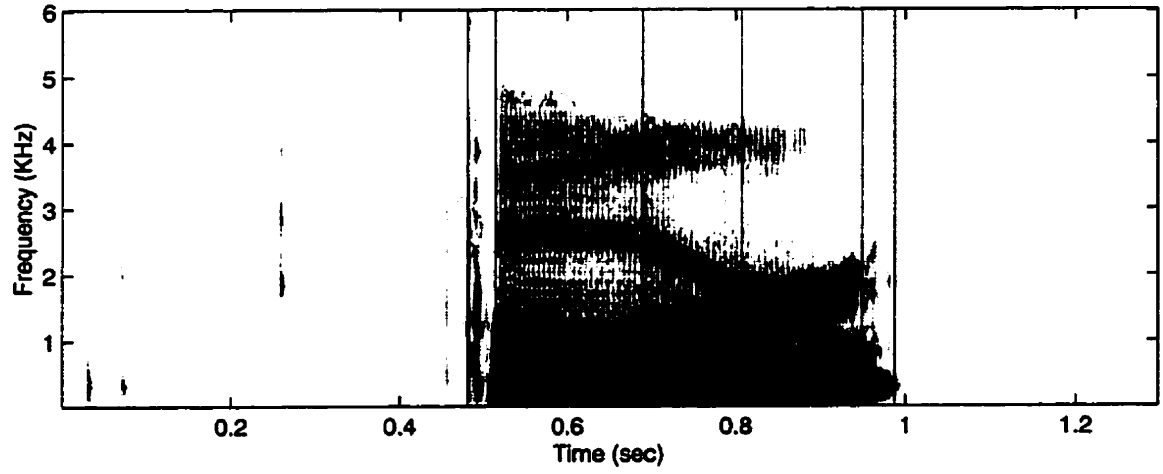


(b)

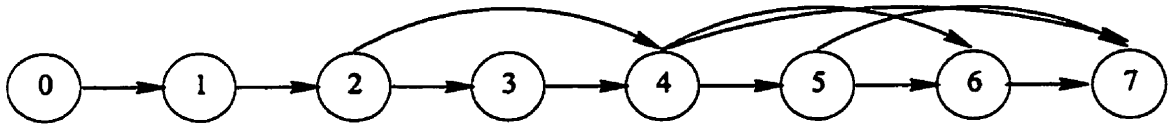
Figure A.4: (a) A sample of segmented word /three/ (b) state model of word /three/

0, 7	silence
1	/th/ aspiration +F2 + F3
2	F2 and F3 going down, transition of /th/ to /r/
3	/r/ F3 low
4	transition of /r/ to /i/, F2 and F3 going up
5	/i/ sound
6	voice bar
8	/th/ aspiration, F2 and F3 not present

Table A.4: Important features associated to states of model /three/



(a)

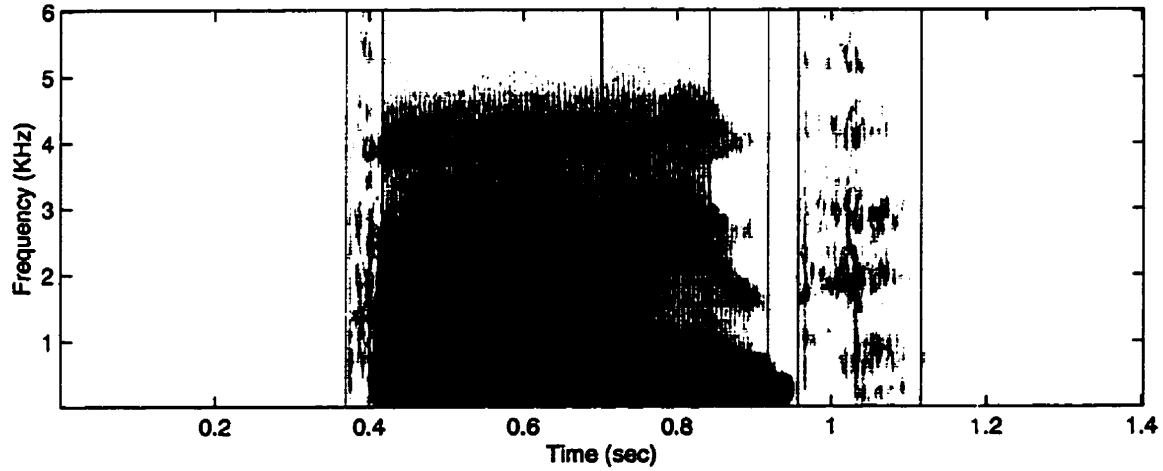


(b)

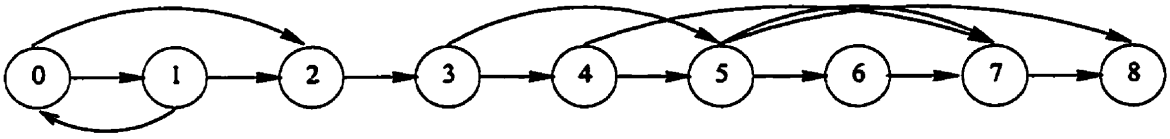
Figure A.5: (a) A sample of segmented word /four/ (b) state model of word /four/

0, 7	silence
1	friction of /f/
2	/o/, F2 low, F3 high
3	transition of /o/ to /r/, F3 going down
4	/r/ , F3 low
5	voice bar
6	whisper mode of voice bar

Table A.5: Important features associated to states of model /four/



(a)

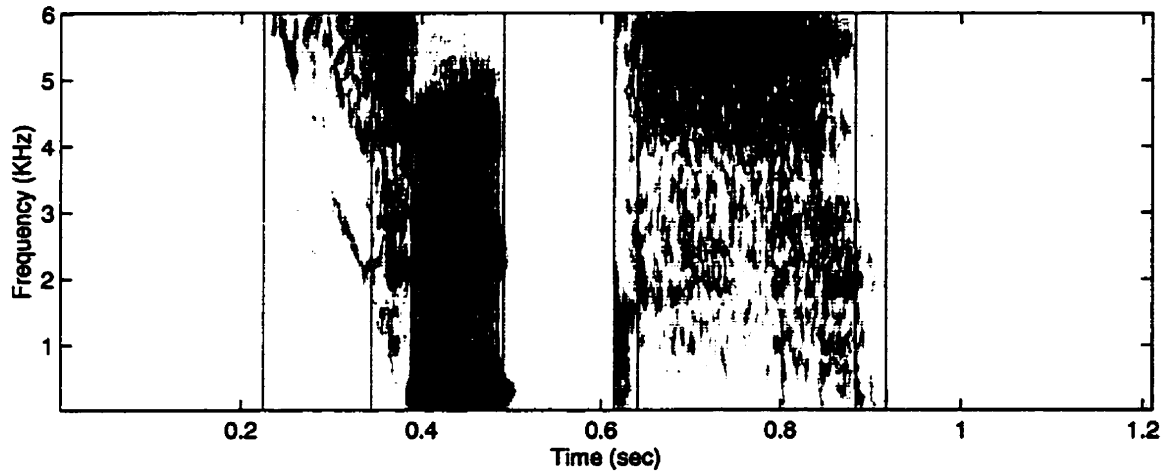


(b)

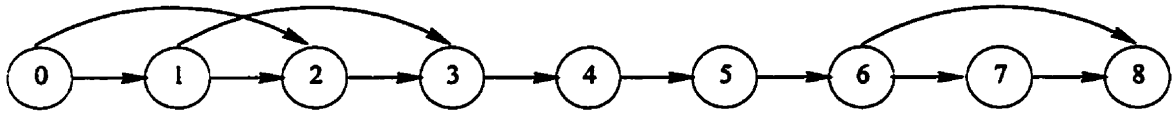
Figure A.6: (a) A sample of segmented word /five/ (b) state model of word /five/

0, 8	silence
1	friction of /f/
2	/aa/
3	transition of /aa/ to /ey/, F3 going up
4	transition of /ey to /v/, F3 going down
5	voice bar in /v/
6	silence
7	friction of /v/

Table A.6: Important features associated to states of model /five/



(a)

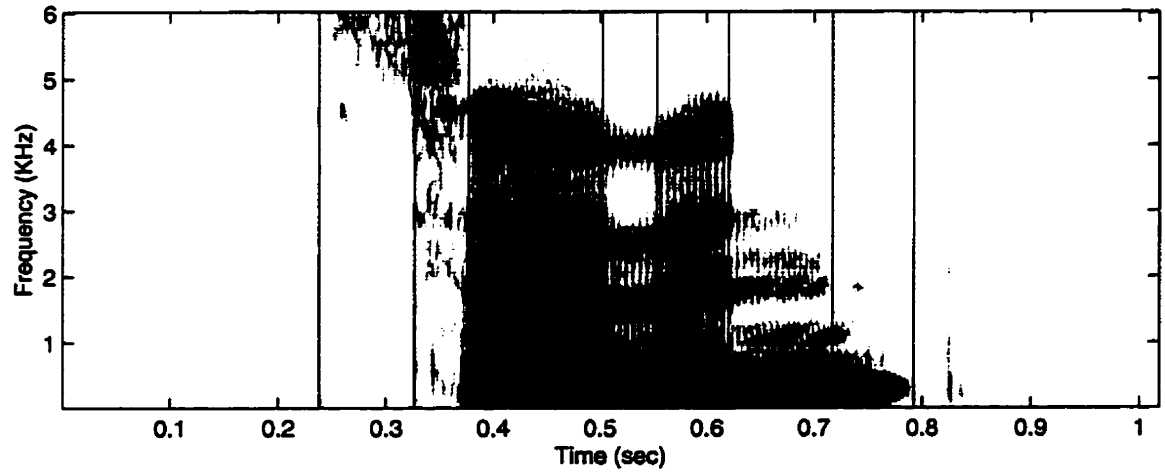


(b)

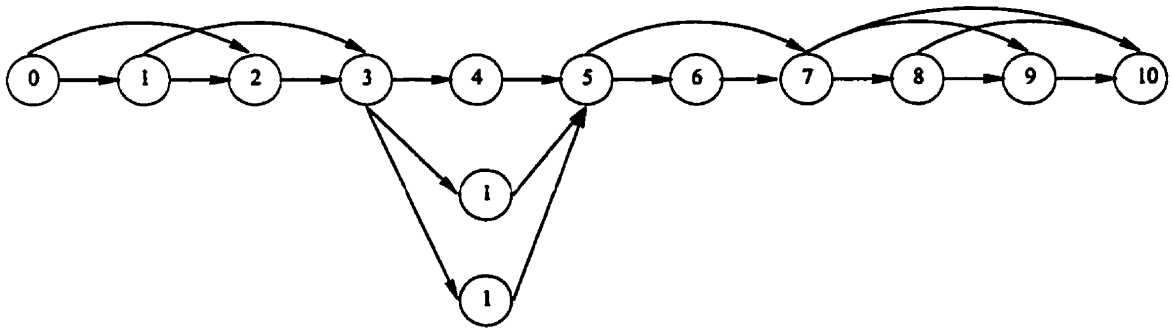
Figure A.7: (a) A sample of segmented word /six/ (b) state model of word /six/

0, 4, 8	silence
1	friction of /s/ high frequencies only
2	friction of /s/ high and low frequencies F2, F3 present
3	/i/ in six, F2 and F3 join
5	/k/ sound high energy around 2kHz
6	friction of /s/ high frequencies
7	friction of /s/ low frequencies

Table A.7: Important features associated to states of model /six/



(a)

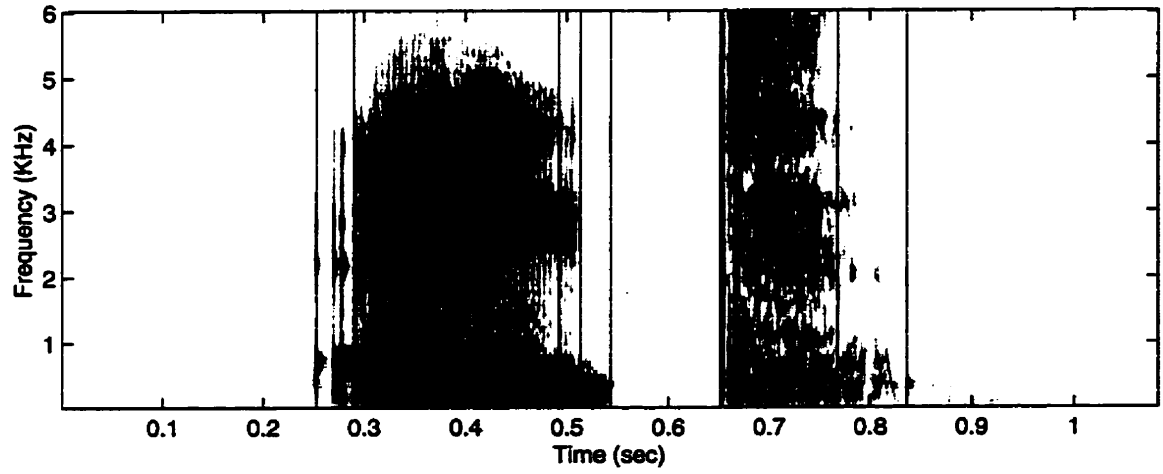


(b)

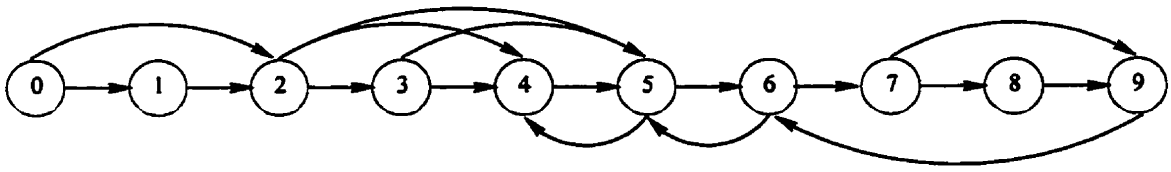
Figure A.8: (a) A sample of segmented word /seven/ (b) state model of word /seven/

0, 10	silence
1	friction of /s/ high frequencies
2	friction of /s/ high and low frequencies
3	/e/
4	/v/ with pattern as in low energy /e/
5	transition of /e/ to /v/
6	/n/
7	/n/ voice bar only
8	/n/ voice bar and aspiration
9	whisper mode of voice bar
11	/v/ with pattern of voice bar
12	/v/ with pattern of friction as in /f/

Table A.8: Important features associated to states of model /seven/



(a)

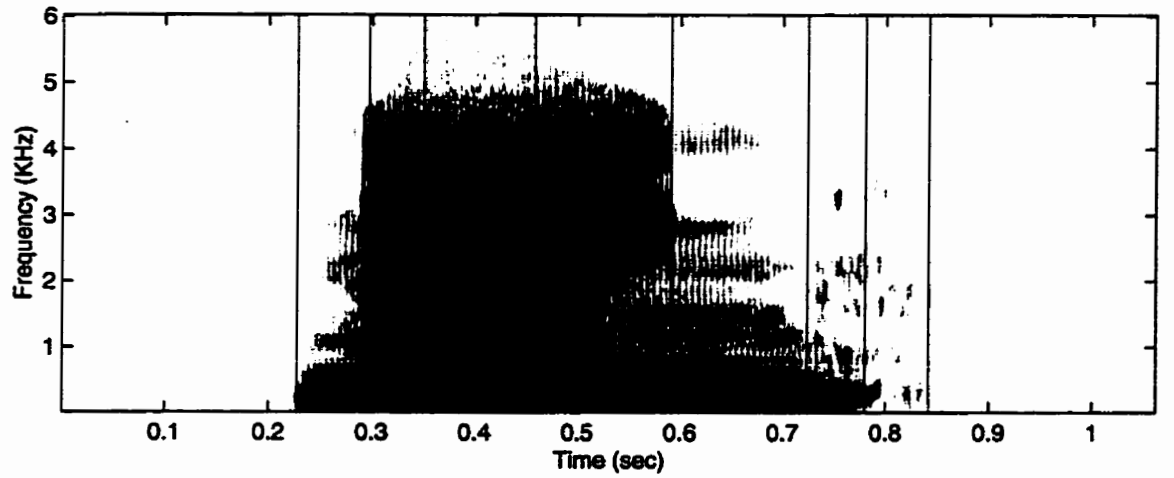


(b)

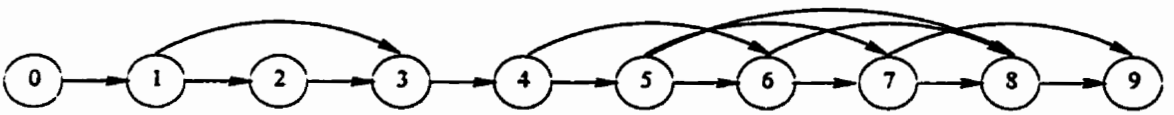
Figure A.9: (a) A sample of segmented word /eight/ (b) state model of word /eight/

0, 5, 9	silence
1	/e/ with low energy
2	/ey/ F2 going up
3	/ey/ F2 going down
4	voice bar
6	pencil line
7	aspiration of /t/ with high and low energy
8	aspiration of /t/ with low energy only

Table A.9: Important features associated to states of model /eight/



(a)

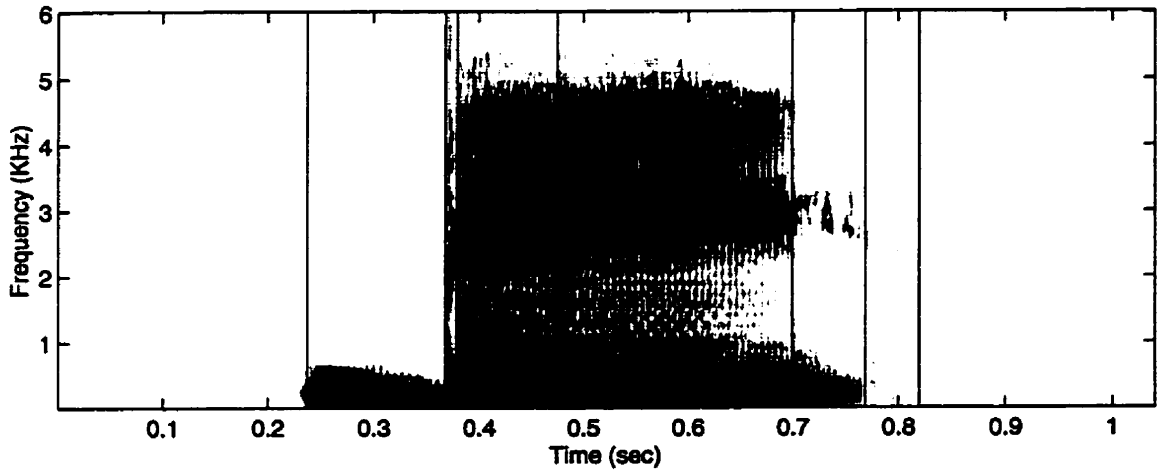


(b)

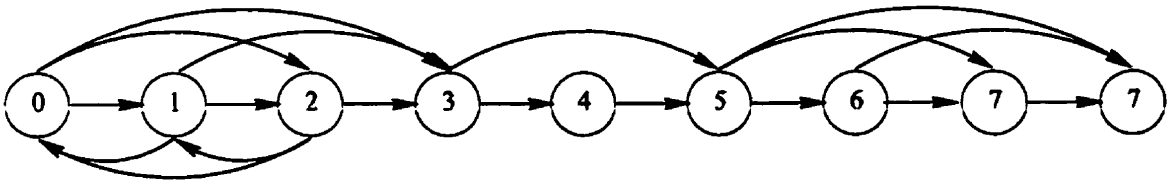
Figure A.10: (a) A sample of segmented word /nine/ (b) state model of word /nine/

0, 9	silence
1	/n/
2	transition of /n/ and /aa/, F1 and F2 are distanced
3	/aa/
4	/ey/ F2 going up
5	/n/
6	voice bar
7	/n/ with aspiration
8	/n/ aspiration only

Table A.10: Important features associated to states of model /nine/



(a)

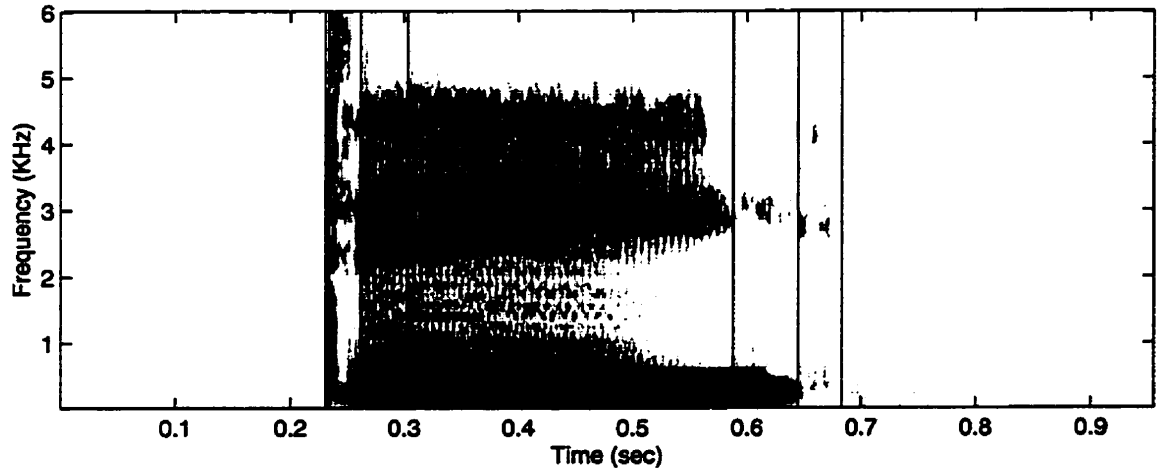


(b)

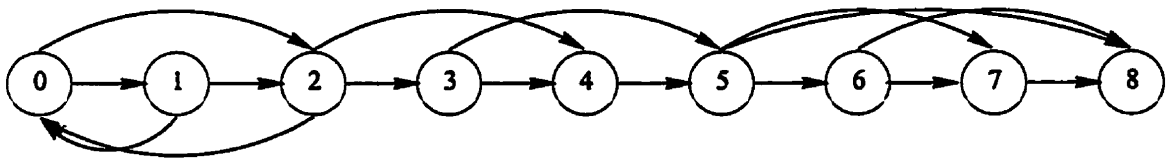
Figure A.11: (a) A sample of segmented word /bi/ (b) state model of word /bi/

0, 8	silence
1	voice bar
2	pencil line
3	aspiration of /b/
4	transition of /b/ to /i/, F2 and F3 moving up
5	/i/
6	/i/ with weak F2
7	/i/ in whisper mode

Table A.11: Important features associated to states of model /bi/



(a)

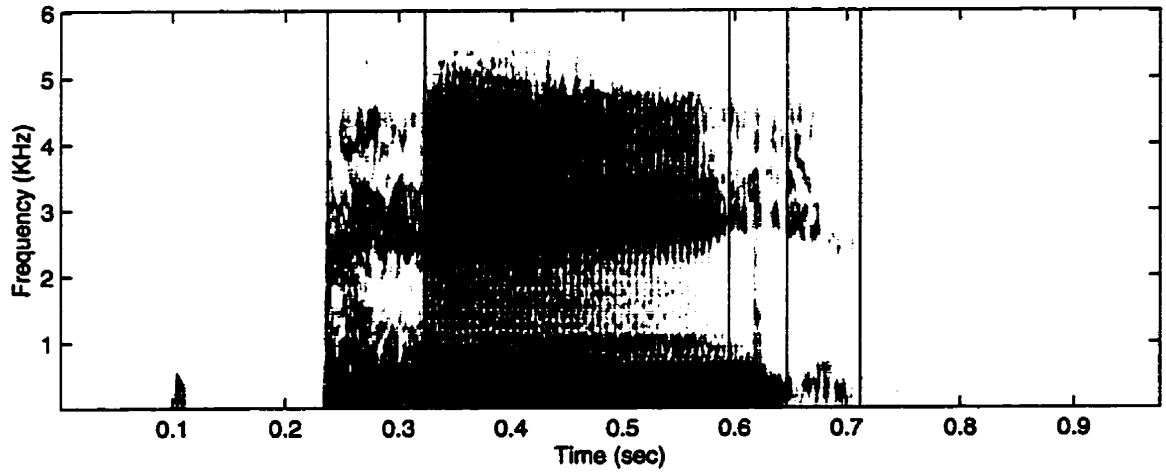


(b)

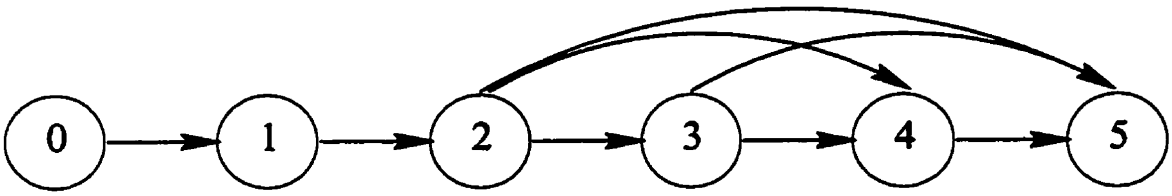
Figure A.12: (a) A sample of segmented word /di/ (b) state model of word /di/

0, 8	silence
1	voice bar
2	pencil line
3	aspiration of /d/
4	transition of /d/ to /i/, F2 and F3 moving up
5	/i/
6	/i/ with weak F2
7	/i/ in whisper mode

Table A.12: **Important features associated to states of model /di/**



(a)

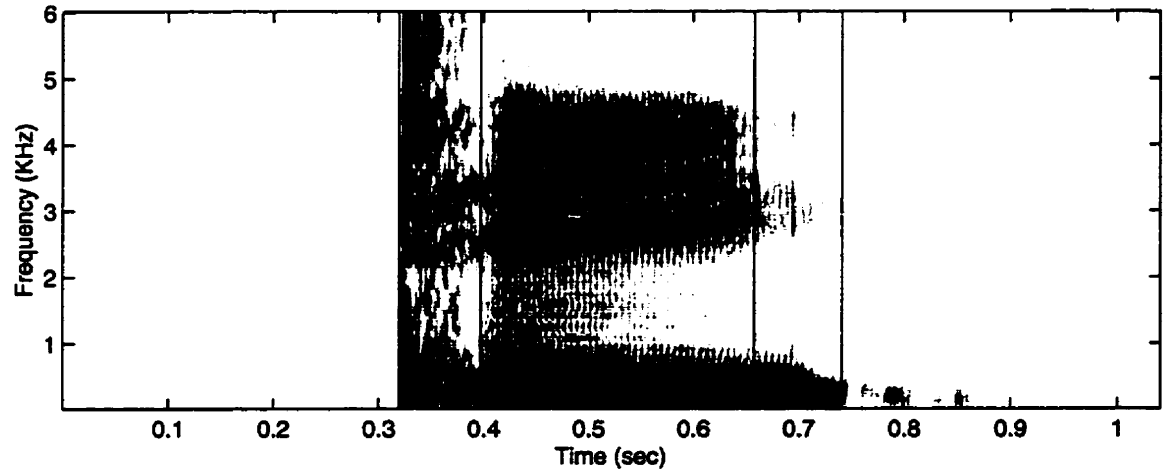


(b)

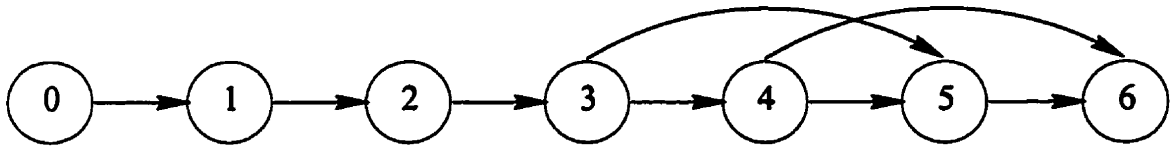
Figure A.13: (a) A sample of segmented word /pi/ (b) state model of word /pi/

0, 5	silence
1	aspiration of high energy in low frequencies
2	/i/
3	/i/ with weak F2
4	aspiration of /i/

Table A.13: Important features associated to states of model /pi/



(a)



(b)

Figure A.14: (a) A sample of segmented word /ti/ (b) state model of word /ti/

0, 6	silence
1	pencil line
2	aspiration of /t/
3	/i/
4	/i/ with weak F2
5	voice bar

Table A.14: Important features associated to states of model /ti/

Bibliography

[1] L. R. Bahl. P. F. Brown, P. V. de Souza, and R. L. Mercer. "Maximum mutual information estimation of hidden Markov model parameters for speech recognition ". *IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Tokyo, Japan*, pages 2.3.1–2.3.4, 1986.

[2] R. Battiti. "Using mutual information for selecting features in supervised neural net learning ". *IEEE Transactions on Neural Networks*, 5(4):537–550, July 1994.

[3] A. Biem and S. Katagiri. "Filter bank design based on discriminative feature extraction ". *IEEE proc. ICASSP*, pages 485–488, 1994.

[4] A. Biem, S. Katagiri, and B. Juang. "Pattern recognition using discriminative feature extraction". *IEEE Transactions on Signal Processing*, 45(2):500–504, Feb. 1997.

[5] E. L. Bocchieri and J. G. Wilpon. "Discriminative feature selection for speech recognition ". *Computer Speech and Language*, pages 229–246, 1993.

[6] V. I. Borisenko, A. A. Zlatopolskii, and I. B. Muchnik. "Image Segmentation (state of the art survey)". *Automat. Remote Contr.*, 48(7):837–879, July 1987.

[7] R. Chengalvarayan and L. Deng. "HMM-Based speech recognition using state-dependent, discriminatively derived transforms on Mel-warped DFT features ". *IEEE Transactions on Speech and Audio Processing*, 5(3):243–256, May 1997.

[8] R. Chengalvarayan and L. Deng. "Use of generalized dynamic feature parameters for speech recognition ". *IEEE Transactions on Speech and Audio Processing*, 5(3):232–242, May 1997.

[9] R. Chengalvarayan and L. Deng. " Speech trajectory discrimination using the Minimum Classification Error learning ". *IEEE Transactions on Speech and Audio Processing*, 6(6):505–515, Nov. 1998.

[10] F. S. Cohen. "Maximum likelihood unsupervised textured image segmentation". *Graphical models and image processing*, 54(3):239–251, May 1992.

[11] T. M. Cover and J. A. Thomas. *"Elements of Information Theory"*. John Wiley and Sons, New York, 1991.

[12] L. Deng. "Integrated optimization of dynamic feature parameters for hidden Markov modeling of speech". *IEEE Signal Processing Lett.*, 1(4):66–69, 1994.

[13] H. Derin and H. Elliott. "Modeling and segmentation of noisy and textured images using Gibbs Random Fields". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1):39–55, Jan. 1987.

[14] G. R. Doddington and T. B. Schalk. "Speech recognition: turning theory to practice ". *IEEE Spectrum*, pages 26–32, 1981.

[15] R. O. Duda and P. E. Hart. *"Pattern classification and scene analysis"*. John Wiley and Sons, New York, 1973.

[16] D. Fohr, J. Haton, and Y. Laprie. "Knowledge-Based techniques in acoustic-phonetic decoding of speech: interest and limitations". *International Journal of Pattern Recognition and Artificial Intelligence*, 8(1):133–153, 1994.

[17] K. Fukunaga. "*Introduction to statistical pattern recognition*". Academic Press, New York, 1972.

[18] J. Harrington. "*Acoustic cues for automatic recognition of English consonants*", chapter 2, pages 69–143. EDITS. Edinburgh University Press, 1985. Ed. J. Laver.

[19] X. D. Huang, Y. Ariki, and M. A. Jack. "*Hidden Markov models for speech recognition*". Redwood Press Limited, Melksham, Wilts, 1990.

[20] A.K. Jain. "*Fundamentals of digital image processing*". Prentice-Hall, Inc., Englewood Cliffs, 1989.

[21] B. Juang and C. Lee. Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 5(3):257–265, May 1997.

[22] B. Juang and L. R. Rabiner. "The segmental k-means algorithm for estimating parameters of hidden Markov models ". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(9):1639–1641, Sept. 1990.

[23] B.-H. Juang and S. Katagiri. "Discriminative learning for minimum error classification ". *IEEE Transactions on Signal Processing*, 40(12):3043–3054, Dec. 1992.

[24] B. H. Juang and L. R. Rabiner. "Issues in using Hidden Markov Models for speech recognition". In S. Furui and M. M. Sondhi, editors, *Advances in speech signal processing*, chapter 17, pages 509–553. Marcel Dekker, Inc., 1991.

[25] S. R. Kadaba, S. B. Gelfand, and R. L. Kashyap. "Bayesian Decision feedback for segmentation of binary images". *IEEE Transactions on Image Processing*, 5(7):1163–1178, July 1996.

[26] S. Katagiri, C. H. Lee, and B. H. Juang. "New discriminative training algorithms based on the generalized probabilistic descent method". *Proc. of IEEE Workshop Neural Networks for Signal Processing, Piscataway, NJ*, pages 299–308, Aug. 1991.

[27] R. D. Kent and C. Read. *"The acoustic analysis of speech"*. Singular Publishing Group Inc., San Diego, California, 1992.

[28] T. Kohonen. *"Self-Organization and associative memory"*. Springer-Verlag, Berlin, 3th edition, 1989.

[29] T. Kohonen. "Improved versions of learning vector quantization". *Proceedings of the International Joint Conference on Neural Networks, San Diego*, pages I545–550, June 1990.

[30] T. Kohonen, J. Kangas, and J. Laaksonen. "LVQ-PAK: A program package for the correct application of learning vector quantization algorithms". *Proceedings of the International Joint Conference on Neural Networks, baltimore*, pages I725–730, June 1992.

[31] S. Krishnan, k. Samudravijaya, and P.V.S Rao. "Feature selection for pattern classification with Gaussian mixture models: A new objective criterion". *Pattern Recognition letters*, 17:803–809, 1996.

[32] C. Lee and D. A. Landgrebe. "Feature extraction based on decision boundaries ". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):388–400, April 1993.

[33] D. Lowe and A. R. Webb. "Optimized feature extraction and the Bayes decision in feed-forward classifier networks ". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):355, April 1991.

[34] B. Lowerre and R. Reddy. "The HAPPY speech understanding system". In W. Lea, editor, *Trends in speech recognition*, pages 340–346. Prentice-Hall, Inc., 1980.

[35] S. Moon and J. Hwang. "Robust speech recognition based on joint model and feature space optimization of Hidden Markov Models". *IEEE Transactions on Neural Networks*, 8(2):194–204, March 1997.

[36] H. Ney. "On the probabilistic interpretation of neural network classifiers and discriminative training criteria". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):107–119, Feb. 1995.

[37] J. P. Olive, A. Greenwood, and J. Coleman. "*Acoustics of American English Speech*". Springer-Verlag, New York, 1993.

[38] K. K. Paliwal, M. Bacchiani, and Y. Sagisaka. "Minimum classification error training algorithm for feature extractor and pattern classifier in speech recognition ". *Proc. Europ. Conf. Speech Communication Technology*, 1:541–544, May 1995.

[39] T. N. Pappas. "An adaptive clustering algorithm for image segmentation". *IEEE Transactions on Signal Processing*, 40(4):901–914, April 1992.

[40] R. K. Potter. G. A. Kopp, and H. G. Kopp. *"Visible speech"*. Dover Publications Inc., New York, 1966.

[41] L. Rabiner and B. H. Juang. *"Fundamentals of speech recognition"*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1993.

[42] L. R. Rabiner. "A Tutorial on hidden Markov models and selected applications in speech recognition". *Proc. IEEE*, 77:257–285, Feb. 1989.

[43] M. J. Russell and R. K. Moore. "Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition ". *IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Tampa, Fla.*, pages 5–8, May 1985.

[44] T. B. Schalk. "The design and use of speech recognition data bases ". *Proceedings of the workshop on standardization for speech I/O technology*, pages 211–214, March 1982.

[45] A. Torre, A. M. Peinado, N. J Rubio, V. E. Sanchez, and J. e. Diaz. "An application of minimum classification error to feature space transformations for speech recognition". *Speech Communication*, 20(3-4):273–290, Dec. 1996.

[46] C. H. Lee W. Chou, B. H. Juang, and F. K. Soong. "A minimum error rate pattern recognition approach to speech recognition ". *International Journal of Pattern Recognition and Artificial Intelligence*, 8(1):5–31, 1994.

[47] J. Zhang, J. W. Modestino, and D. A. Langan. "Maximim-likelihood parameter estimation for unsupervised stochastic model-based image segmentation". *IEEE Transactions on Image Processing*, 3(4):404–420, July 1994.

[48] V. W. Zue. "The use of speech knowledge in automatic speech recognition ". *Proceedings of IEEE*, 73(11):1602–1611, Nov. 1985.