Cognitive Component Analysis on Phonemes

Jinbo Li

Kongens Lyngby 2008

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Abstract

Cognitive component analysis(COCA) is defined as unsupervised grouping of data leading to a group structure well aligned with that resulting from human cognitive activity[16].

The thesis describes the *Cognitive Components Analysis* processes on the low-level cognitive components(phonemes)[4]. We try to prove that an information optimal statistical regularity resembles the human's cognition activity.

Preface

This thesis was prepared at Informatics Mathematical Modelling, the Technical University of Denmark in partial fulfillment of the requirements for acquiring the Msc. degree in engineering.

Lyngby, August 2007

Jinbo LI

Acknowledgements

I thank my supervisors Ling and Lars Kai Hansen who lead me to explore this machine learning world.Thanks for useful advice and ideas from Lars and Ling. Thanks for Ling's generous answers to all my questions.

Contents

[1]This chapter is based on the [1]
[2]This section is based on the [9]
[3]This section is based on the [9]A detailed introdution was given,here I only extract some key steps

[4]This section is based on [11]

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Cognitive Component Analysis

Cognitive component analysis(COCA) is defined as unsupervised grouping of data leading to a group structure well aligned with that resulting from human cognitive activity[16]. In this thesis, we focus on the *COCA* on phoneme which is the smallest distinguishable unit in the speech perception.

Human beings need deal with huge amount of information from the surroundings. During the evolution process, the human perception and cognition system may figure out a regulation or rule to efficiently represent and process the information from the outside world. We envision that this regulation may consist of the sparse representation of the "real world data" and statistical process regularities.

MFCC(Mel Frequency Cepstral Coefficient) is a very successful feature for speech modelling. There are very rich documents about the application of this feature. It is thought to be a very good representation of the outside acoustical signal in the auditory system. An energy based sparsification on the MFCCs was proposed as the sparse coding step for our *COCA*. *PCA(Principal component analysis)* is commonly used as a dimension reduction technique for the *ICA(Independent Component Analysis)*. It also reveals a cognitive structure on

the sparse MFCCs(section. 3.1).

So *COCA* is not limited to the *ICA* which is only one of the possible statistical regularities for the human brain. But *independence* is a very optimal information processing solution. More over, *ICA* has been found to be a more appropriated model when it is used to group some abstract data[16],[6]. ICA is also found to be a resemble representation in the perceptual system[2],[3]. In this thesis, *ICA* is used to identify the *cognitive component* and further to prove that human'cognition activity may resemble an information optimal mechanism(independence)[4].

Another unsupervised learning technique-*Soft-Lost* has been used in this thesis as a comparison to the *ICA*. The *Soft-Lost* is based on the covariance structure of the data set while *ICA* take the higher order statistics information into account. By comparing the *Soft-Lost* and *ICA* in *COCA*, we can know if the cognition activity need dealing with the high order statistics information, that is to say, more approximated to an information optimal regularity-independence.

We also use some supervised learning method to better understand the data set and provide some ideas for the unsupervised learning method in *COCA*

1.2 Thesis Outline

The chapter2 **Introduction to the Algorithms**, we give an introduction about the main algorithms used in this project.

PCA is an unsupervised learning algorithm to transfer the data set to a new coordinate system in which the projections of the data set on each new coordinate are ordered in the sequence of variance. *PCA* on the sparsified *MFCC* feature of speech reveals a linear cloud structure. The low level *cognitive components*(phonemes) of speech are then found aligned with the rays.

ICA is a method for finding underlying factors or components from multivariate (multidimensional) statistical data. It can estimate the independent components from linear mixture data and identify these line vectors in the ray structure.

Kernel Principal Component Analysis is the principal component analysis used on a *feature space*. It was proposed to extract the *high-order statistics* information for the *Independent Component Analysis*.

Soft-Lost is another unsupervised learning technique to find the orientations of

the ray structure. It is used in this project as a comparison to the *ICA*.

Fisher Linear Discriminant Analysis is a supervised learning method to provide a linear separation solution for a classification task.It also provides us a technique to know what the speaker difference and phoneme difference could be.

In the chapter 3 **Experiments**, we describe four experiments about the process of the *COCA* on the low level speech signals and further discuss the generality of the model on different speakers.

In section *SOFA letter utterance experiment*[4], we describe the process of the *Cognitive Component Analysis* on the low level cognitive component by an example on the 'SOFA' which is short for the four letters 's','o','f' and 'a'. The *cognitive component*(phoneme), /e/ in 's' and 'f' is found aligned a ray structure in a linear cloud of sparsified data.

In section *Cognitive Analysis on the phoneme data sets*, we extend the *COCA* to serval speakers and analyze a series of figures.

In section *Similarity measurement & Invariant Cue:*, we hope the *COCA* can be used to figure out the *Invariant Cue* phenomenon in speech perception.

In section *Unsupervised Classification*, we use the *ICA* on a two phonemes classification task.

Chapter 2

Introduction to the Algorithms

2.1 ICA(Independent component analysis)[1]

ICA is a method for finding underlying factors or components from multivariate (multidimensional) statistical data. It can estimate the independent components from linear mixture data (observation data). ICA is the technique to find a linear non-orthogonal coordinate system in multivariate data. The directions of the axes of this coordinate system are determined by the data's second and higher-order statistics. The goal of the ICA is to linearly transform the data such that the transformed variables are statistically independent from each other.

Independent component analysis was originally developed to solve the cocktail-party problem. It has been found that the ICA remarkably resembles the mechanism in the perceptual system of multimedia data in both human and animals[16].

The ICA algorithm decomposes a data set or observation data into a mixing matrix and a source component matrix. It reveals that the observation data is a linear mixture of some statistically independent components. We also neglect

[1]This chapter is based on the [1]

any time delays that may occur in the mixing. So this model is often called the instantaneous mixing model. A noise free ICA can be written as:

$$\mathrm{X} = \mathrm{AS} = \sum_{i=n} a_i s_i \tag{2.1}$$

Some Restriction in ICA

1. The independent components are assumed statistically independent.
2. The independent components must have non Gaussian distribution
3. We assume the unknown mixing matrix is square

2.1.1 What is statistical independence?

Mathematically, statistical independence is defined in terms of probability densities. The random variables x and y are said to be independent if and only if:

$$p_{x,y}(x, y) = p_x p_y \tag{2.2}$$

In words, the joint density $p_{x,y}(x, y) = p_x p_y$ could be factorized into product of their marginal densities. Equivalently, independence could be defined by replacing the probability density functions in the definition by the respective cumulative distribution functions, which must also be factorizable

$$E\{g(x)h(y)\} = E\{g(x)\}E\{h(x)\} \tag{2.3}$$

where $g(x)$ and $h(y)$ are any absolutely integrable function of x and y. This is because:

$$E\{g(x)h(y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x)h(y)p_{x,y}dydx = \int_{-\infty}^{\infty} g(x)p_x(x)dx \int_{-\infty}^{\infty} h(y)p_y(y)dy$$

$$= E\{g(x)\}E\{h(x)\} \tag{2.4}$$

Eq. 2.4 shows that *uncorrelatedness* is only a special case of independence when both $h(y)$ and $g(x)$ are linear functions, and will only calculate the second-order statistics.

2.1.2 ICA by Maximum Likelihood Estimation[2]

The ICA model :

$$\mathrm{X} = \mathrm{AS}$$

can be reformulated as:

$$p_x(x) = |\det B| p_s(s) = |\det B| \prod_i p_i(s_i) \tag{2.5}$$

where$B = A^{-1}$, and the p_i denotes the densities of the independent components.The s_i can be replaced by $s_i = b_i^T X$. (b_i is a column vector in the matrix B)

$$p_x(x) = |\det B| p_s(s) = |\det B| \prod_i p_i(b_i^T X) \tag{2.6}$$

X is made up by T observations $x(1), x(2), \ldots, x(T)$ then the likelihood can be constructed as the product of the estimated density function at T points:

$$L(B) = \prod_{t=1}^{T} \prod_{i=1}^{n} p_i(b_i^T x(t)) |\det B| \tag{2.7}$$

The log-likelihood is given by:

$$\log L(B) = \sum_{t=1}^{T} \sum_{i=1}^{n} \log p_i(s_i(t)) + T \log |\det B| \tag{2.8}$$

We can devide both sides by the observation number T and the average value over the observations T can be replaced by the expection operator:

$$\frac{1}{T} \log L(B) = E\{\sum_{i=1}^{n} \log p_i(b_i^T x(t))\} + \log |\det B|$$

$$= E\{\sum_{i=1}^{n} \log p_i(s_i)\} + \log |\det B| \tag{2.9}$$

Then the objective is to maximize this log-likelihood function. The maximization algorithm used is called *UCMINF* given in [8].

To maximize the objective function by a gradient method, we get:

$$g_i(s_i) = \frac{\partial \log p_i(s_i)}{\partial s_i} = \frac{p_i'(s_i)}{p_i(s_i)} \tag{2.10}$$

[2]This section is based on the [1]

The ML estimator Eq. 2.9 will be locally consistent, if the assumed densities p_i fulfill[1]:

$$E\{s_i g_i(s_i) - g'(s_i)\} > 0 \tag{2.11}$$

We can see from the above formulations, the densities of the independent components are needed to be estimated. Because non-parametric estimation of the densities are normally difficult. In some case, we can know the density of the source components in advance. A second way, we can estimate the density of the independent components by a family of densities that are specified by a limited number of parameters[1].
In the *COCA*, the high level representation of the speech is a sparse coded data. We can hypothesis the source components is the sparse response in the human brain. We thus use the density estimation prior[3]:

$$p(s_i) = \pi^{-1} / \cosh(s_i) \tag{2.12}$$

and then

$$\log p(s_i) = -\log \pi - \log \cosh(s_i) \tag{2.13}$$

$p(s_i)$ is the density funciton of the *ith* source component. This density function will give us a super Gaussian density (Fig. 2.1). This approximation will greatly simplify the formulations and the Eq. 2.11 is fufilled.

$$g_i(s_i) = \tanh(s_i) = s_i - \frac{1}{3}s_i^3 + \frac{2}{15}s_i^5 - \dots \tag{2.14}$$

Equation . 2.14 shows how the estimation prior bring the *high-order statistics* information and non-linearity to the maximum likelihood function.

2.1.3 ICA Ambiguities

In the ICA model in equation. 2.15, it is easy to see that the following ambiguities will necessarily hold:

1. We can not determinate the variances of the independent components.
   This can be easily found in the ICA formulation

   $$\mathrm{X} = \sum_i (\frac{1}{a}\mathrm{A}_i)(\mathrm{S}_i a) \tag{2.15}$$

   The formulation shows that scaling the S source component by any factor$a$ could be compensated by multiplying the mixing matrix with that factor.[5]

[3]This estimation prior is used in the DTU ICA toolbox
[5]in the ML algorithm, only the final result the variance of the s was unified.

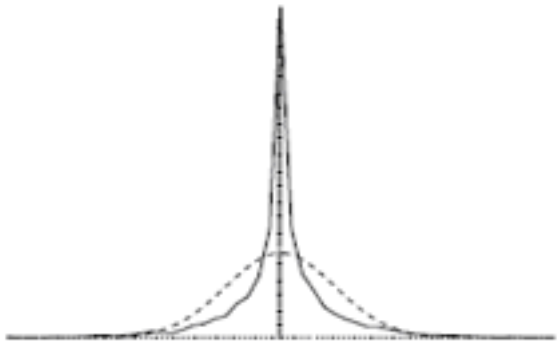


Figure 2.1: The dash line is the Gaussian density, the solid line is the super Gaussian density

In the consequence, we can constrain the result by unifying the variance of the source component $E(s_i^2) = 1$, then the magnitude of the A will change accordingly. With this modification, in our experiment, the length of the ray could represents the strength(energy) of that source component. Another ambiguity is the sign of the A and S. We could multiply an independent component by-1 without affecting the model. This ambiguity is resolved by a step to look for the positive direction and component in the source component and mixing matrix(section. 3.1.5).

2. We can not determinate the order of the independent component
   This could also be found in the formulation of *ICA*:

$$\mathrm{X} = \mathrm{A}_1\mathrm{S}_1 + \ldots\ldots + \mathrm{A}_n\mathrm{S}_n \tag{2.16}$$

Since both S and A are being unknown, we can freely change the order of the terms in polyminal in the equation above. There is some way to order the components and the columns of the mixing matrix. An application of the importance of the column vector has been discussed in[7].The L^2norm and the *variance* of the column vectors are calculated and ordered. The L^2norm of a vector $X = (x_1, x_2, x_3)$ is given by:

$$|x| = \sqrt{x_1^2 + x_2^2 + x_3^2} \tag{2.17}$$

Fig. 2.2 shows that the L^2norm is well aligned with the order of the variance of the vectors. Thus we order the columns of the mixing matrix in the order of the L^2norm[6]

[6]In the ML of the DTU toolbox the mixing matrix are ordered according to the energy of each princinpal component

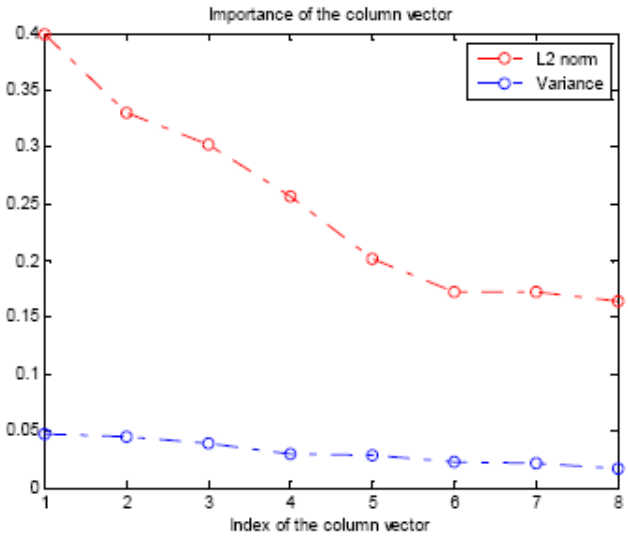


Figure 2.2: Energy based Sparsification revealing the ray structure

2.2 PCA(Principal Component Analysis)[7]

PCA is dimension reduction technique. With the dimension reduction, the ICA computational cost is reduced and noise is also reduced by the PCA.

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on[13].

PCA by covariance method:

In the dataset we can find the directions with the most variance, these directions are the eigenvectors of the covariance matrix and the variances are the eigenvalues of the covariance matrix. A typical covariance method is performed by SVD (Singular Value Decomposition)

[7]This section is based on the [9]

we use SVD to perform PCA.We decompose X using SVD then

$$\mathrm{X} = \mathrm{U}\Gamma\mathrm{U}^T \tag{2.18}$$

The covariance matrix C can be written as

$$\sum X = \lim_{T\to\infty} \frac{1}{T}\mathrm{X}\mathrm{X}^T = \lim_{T\to\infty} \frac{1}{T}\mathrm{U}\Gamma^2\mathrm{U}^T \tag{2.19}$$

In which, U is a $(n \times m)$ matrix. The SVD organizes the singular values according to the size. If $n < m$, the first n columns in U corresponds to the sorted eigenvalues of C and if $m > n$, the first m corresponds to the sorted non-zero eigenvalues of C. Then the data in the tranfered coordinate system can be written as:

$$Y = \widetilde{\mathrm{U}}^T\mathrm{X} = \widetilde{\mathrm{U}}^T\mathrm{U}\Gamma V^T \tag{2.20}$$

Where $\tilde{\mathrm{U}}^T\mathrm{U}$ is a simple $n \times m$ matrix which is one on the diagonal and zero everywhere else.

2.2.1 Some information theory related property of the PCA:

1. The first$q(q \in \{1, \dots M\}$ principal components, i.e. projection on Eigenvectors carry the more variance than the left components
2. The mean-squared approximation error in representing the observatios by the first q principal is minimal.
3. The principal components are uncorrelated.
4. The first q principal components have the maximal mutual information with the original dataset

2.2.2 PCA relation with ICA:

What distinguishes ICA from other methods is that it looks for components that are both statistically independent and non Gaussian. ICA is different from the PCA. Because the non Gaussian structure of the data is taken into account and the higher order statistical information are considered in the ICA algorithms. PCA removes correlations between some variables or signals, at the same time finding directions with maximal variance. Thus, for gaussian data, PCA produces independence. For nongaussian data, PCA does not produce independence.

2.3 Kernel PCA[9]

According to the VC(Vapnik-chervonenkis) theory, a learning machine with more free parameters are generally expected to model more complex decision boundary and thus better achieve a result in the classification task. The input space, in our classification task is MFCC. But in order to extract higher order statistics information, we can map the input space (MFCC) into a *feature space*.

$$\Phi : R^N \rightarrow F, x \tag{2.21}$$

The *feature space* could have an arbitrarily large dimensionality.
The traditional *PCA* is a dimension reduction technique on the input space(MFCC). In order to do a better classification, we can map the input space data to a *feature space* which most often is non-linear with the *input variables*. Then we hope to generalize the *PCA* on the *feature space*.
The covariance of the input variables can be denoted as:

$$\tilde{C} = \frac{1}{M}\sum_{j=1}^{M} x_i x_j^T \tag{2.22}$$

Where M is the number of observation data. Then for the *feature space*, we have:

$$C = \frac{1}{M}\sum_{j=1}^{M} \Phi(x_j)\Phi(x_j)^T \tag{2.23}$$

We define an $M \times M$ matrix K by

$$K_{ij} = (\Phi(x_i) \cdot \Phi(x_j)) \tag{2.24}$$

Compute the dot products $(\Phi(x_i) \cdot \Phi(x_j))$ could be extremly high cost if we map the x to the higher dimension space $\Phi(x)$. For instance[9], if a vector $x = (x_1, x_2)$ to the vector which we extract the 2-th order of the products of the entries in x, then the new vector is $x_{new} = (x_1^2, x_2^2, x_1x_2, x_2x_1)$. Generally, mapping a N dimension input vector to the d_{th} order products will have a *feature space* with the dimension $\frac{(N+p-1)!}{p!(N-1)!}$. Then we use a technique called *Kernel Trick* to compute the dot products.

[9]This section is based on the [9]A detailed introdution was given,here I only extract some key steps

2.3.1 Kernel Trick

With the kernel trick, explicitly computing the dot products $(\Phi(x_i) \cdot \Phi(x_j))$ is not needed, we use a function ,$K(x_i, x_j)$.

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{2.25}$$

Kernel Function has to satisfy two conditions:

1. The kernel function must be symmetric
2. It must satisfy Mercer's Theorem[10]

Some kernel functions:

The polynomial kernel:

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{2.26}$$

In which, if we replace the 1 with 0, it is a homogenous polynomial kernel.

Gaussian kernel:

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2} \tag{2.27}$$

2.3.1.1 Kernel PCA relation with the ICA[9]

Linear PCA is an orthogonal transformation of the coordiinate of the system where we can get the components with the maximal variance. ICA is also a coordinate transformation in which we are looking for a directions(column vectors in A) in the dataset so that the projections on the directions are maximally independent, i.e. "non-Gaussian". We use linear PCA as a preprocess to reduct the dimenionality and in the mean while and hope to keep the most of the information in terms of second-order statics information, i.e. *variance*. In our *ICA* algorithem, (section. 2.1.2), the higher order-statistics information was computed. *Kernel PCA* provide a way to extract these high-order statics information: using polynomial kernels of degree d we are taking into the account d_{th} order statistics.

2.4 Fisher Linear Discriminant Analysis[11]

The objective of the Fisher Linear Discriminant Analysis is to use the label information of the data set to find a linear separation solution for the classification task. This separation direction is different with the PCA in which the separation is found by transforming the coordinate system.

The Fisher LDA is objected to find this separation by maximizing the objective:

$$J_(w) = \frac{w^T S_B w}{w^T S_w w} \tag{2.28}$$

$$S_B = \sum N_c(\mu_c - \bar{x})(\mu_c - \bar{x})^T \tag{2.29}$$

$$S_w = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T \tag{2.30}$$

In which, S_B is the "between classes scatter matrix" and S_w is the "within class scatter matrix". N_c is the number of cases in the class c. μ_c is the mean(center) of the class C. X is the overall mean of the dataset. This objective means the classification is well done when the means of the classes are well separated, measured relative to the sum of the variance of the data assigned to one prarticular class. This measurement also provides us a technique to know how much the classes can be separated in the data set by LDA. In the fellowing experiments, we use this to know the *separatability* of speaker and phonemes denoted as J_s and J_p. The solution in our program w is constrained[11] by:

$$w^T S_w w = 1 \tag{2.31}$$

2.4.1 Kernel Fisher Linear Discriminant Analysis

The general linear direction from the LDA are not sophisticated enough to provide a good solution for a complex problem. We can find some non linear directions by mapping the data (MFCC) non-linearly to a feature spaces. We can use different kernel functions and compared the result. The kernel function was introduced in the section. 2.3. We map the x to the new *feature space* $\Phi(x)$ and then the objection fuction became:

$$J(\alpha) = \frac{\alpha^T S_B^\Phi \alpha}{\alpha^T S_w^\Phi \alpha} \tag{2.32}$$

[11] This section is based on [11]

In the new space,the scatter matrix became:

$$S_B^{\Phi} = \sum_c [k_c k_c^T - k k^T] \tag{2.33}$$

$$S_w^{\Phi} = K^2 - \sum_c N_c k_c k_c^T \tag{2.34}$$

$$k_c = \frac{1}{Nc} \sum_{i \in c} K_{ij} \tag{2.35}$$

$$k = \frac{1}{N} \sum_i K_{ij} \tag{2.36}$$

2.5 Lost(Line Orientation Separation Technique)

The sparsified data set shows a linear cloud structure (ray structure) in the scatter plot. The *PCA* algorithm is not able to find the orientation of the line, that to say, to find the exact A. That is because the PCA algorithm is based on the analysis of covariance of the dataset.

$$\sum X = \lim_{T \to \infty} \frac{1}{T} \mathrm{X}\mathrm{X}^T \tag{2.37}$$

Clearly the information in AA^T is not enough to uniquely identify A, since if one solution A is found, any(row)rotated matrix $\tilde{\mathrm{A}} = \mathrm{AU}, \mathrm{UU}^T = \mathrm{I}$ is also a solution, because $\tilde{A}$ has the same outer product as A. ICA is performed to find the mixing matrix A and independent sources with the independent source component assumption.

Lost is another technique to locate the A and obtain the source components. But it doesn't assume the source component to be independent. The ICA is a more information theoretical algorithm. It provides us a better answer that if the human's cognition process has the similar process as an information optimal regularity like independence.

The LOST falls into two categories according to how the data is assigned to the Line. One is called *Soft Lost* in which the data is "softly" assigned to the M classes, the another one is called *hard lost* in which the data "hard" assigned to the M classes. In the equation, we measure the data point d_j, to each line orientation vector v_i and then soft assign the data to each line (M *class*) by a *soft-max* like function.

2.5.1 Soft-Lost[13]

The orientation of the linear cloud is responding to the principle eigenvector of the covariance matrix . The soft lost is a Expectation Maximization (EM) algorithm. The expectation step firstly soft assigns the data into M classes (source component numbers). The covariance matrix is then calculated for the data associated with each class and the principal eigenvector of the matrix is used as the new line orientation vector estimate.
The algorithm summary: Soft data assignment:

$$z_{ij} = ||d_j - (v_i \bullet d_j)v_i||^2 \tag{2.38}$$

$$\widetilde{z}_{ij} = \frac{e - \beta z_{ij}}{\sum_{i'} e^{-\beta z_{i'j}}} \tag{2.39}$$

In which β control the softness of the boundaries between the region atributed to each line and $\widetilde{z}_{ij}$ are the computed weights of data point$j$ for each line$i$.
Then determine the new line orientation estimated by calculating the principal eigenvector of the covariance matrix. The covariance matrix expression(with zero mean) and assignment weights are combined as fellows:

$$\sum\nolimits_i = \frac{\sum_j z_{ij} d_j d_j^T}{\sum_j z_{ij}} \tag{2.40}$$

where the $\sum_i$ is the covariance of weighted data associated with line i. The eigenvector decomposition of $\sum_i$ is expressed as :

$$\sum\nolimits_i = U_i \Lambda_i U_i^{-1} \tag{2.41}$$

The matrix U_i contains the eigenvectors of$\sum_i$ and the diagonal matrix Λ_i contains its associated eigenvalues $\lambda_i \ldots\ldots \lambda_N$. The new line orientation vector estimate is the principal eigenvector of $\sum_i$ which is expressed as:-

$$v_i = u_{\max} \tag{2.42}$$

where u_{max} is the largest principal eigenvector, the eigenvector whose eigenvalue is the largest. These steps are repeated until the v_i converged.

[13]This section is based on the [15]

Chapter 3

Experiments

3.1 SOFA letter utterance experiment

The four letters 'SOFA' utterances are from the *TIMIT* database. These four letters are separately pronounced.

In the 'SOFA' demo, we want to indicate that generalizable cognitive components corresponding to phonemes, e.g. /e/ from utterance 's' and 'f', can be identified using linear component analysis- *ICA* and *Soft-Lost*. We use this experiment to describe the process of *COCA* on the low level cognitive component(phoneme)[4].

3.1.1 Feature Extraction

MFCC (Mel Frequency Cepstral Coefficient)

MFCC has been the most successful feature for the speech recognition due to its ability to represent the speech amplitude spectrum in a compact form. It has shown the similar way that the human ears respond to the speech with a

logarithmically higher response to lower frequency ranges. The evidence is that the human cochlea is able to react to sound more accurately at lower frequencies. In this experiment, the time scale of window is 40ms, 16 MFCCs are used and 95 % overlapping [4].

MFCC are commonly derived as follows [13]:

1. Take the Fourier transform of (a windowed excerpt of) a signal
2. Map the log amplitudes of the spectrum obtained above onto the Mel scale, using triangular overlapping windows.
3. Take the Discrete Cosine Transform of the list of Mel log-amplitudes, as if it was a signal.
4. The MFCCs are the amplitudes of the resulting spectrum.

The Mel scale is based on a mapping between actual frequency and perceived pith. It is interesting to show the relation of the properties of the MFCC with the motivation of our experiments.

The MFCC could be thought as an accurate model of the low level interpretation of the speech information in the human auditory system. Unsupervised learning algorithm, ICA can be derived to estimate the functionality of the higher level cortex layer.

The use of the MFCC and the independent component analysis can potentially produce an acceptable model of the upper levels of the human auditory system.

3.1.2 Dynamic speech feature—Delta MFCC

$$\Delta MFCC_i = MFCC_{i+1} - MFCC_i \tag{3.1}$$

Fig. 3.2shows that the boundries between the different letters are located. Because delta-mfcc is dynamical features, it captures the change in the speech.

3.1.3 Data Normalization

These four letters are recorded in different conditions. In order to eliminate the loudness difference. More importantly, in our ICA , the high order statistics information are extracted to find the independent component. Data normalization consists of two steps:centering the variable and then unifying the variance.

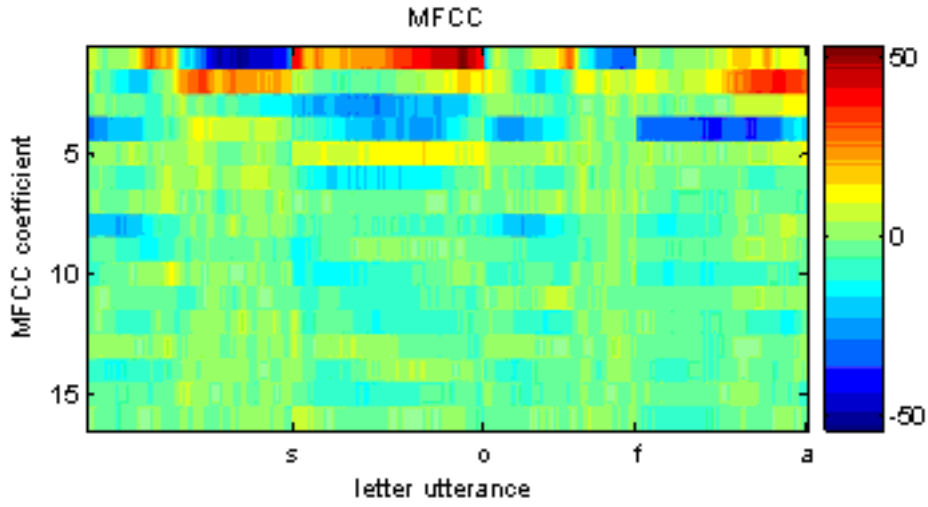


Figure 3.1: MFCCs of the 'sofa' utterance show clear boundaries between different utterance

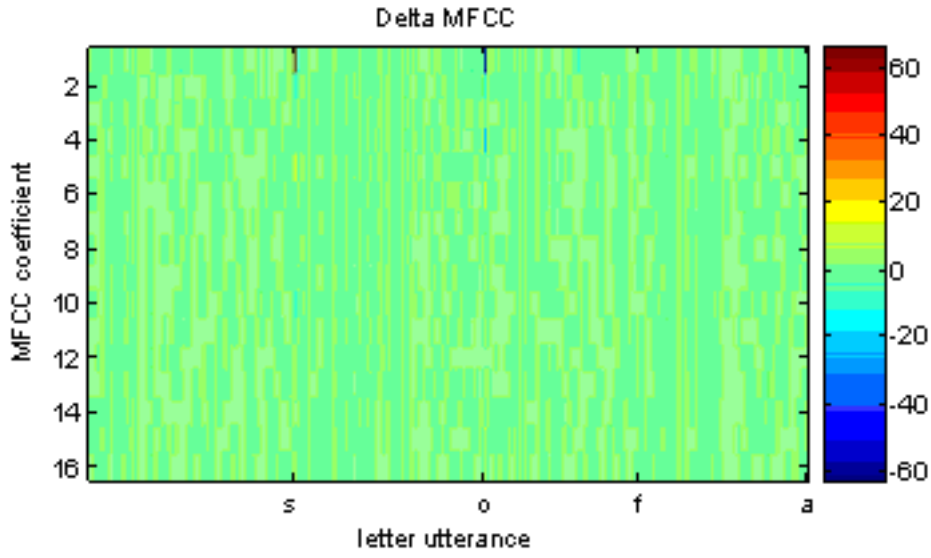


Figure 3.2: Delta MFCC of the 'sofa' utterance

Subtract the mean :

$$X_i^{'} = X_i - E\{X_i\} \tag{3.2}$$

Then in the ICA:

$$E\{S_i\} = \mathrm{A}^{-1} E\{X_i^{'}\} \tag{3.3}$$

This equation shows the independent component have zero mean as well.

Unify The Variance :

$$X_i^{'} = \frac{X_i}{std(X_i)} \tag{3.4}$$

Data Normalization combined with the PCA(section. 2.2) whitten the data set. It can simplify the *ICA* problem and provides a fast convergence in our Maximal likelihood ICA algorithm.

In Eq. 3.4 *std* is the standard deviation.

3.1.4 Data set sparsification

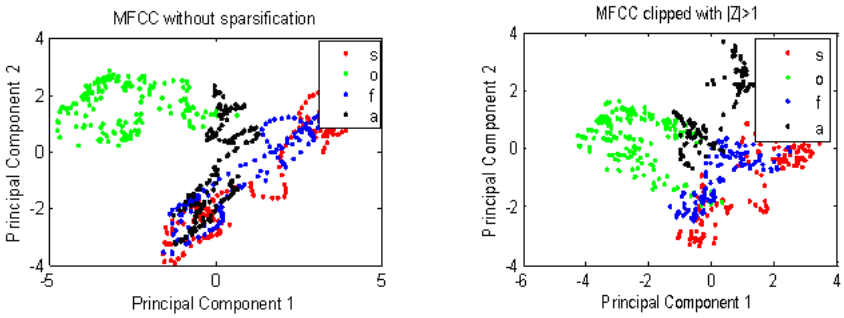
We sparsify the features based on the energy as a pre-process to reduce the intrinsic noise[4]. The sparsification unveils the "ray structure" in the dataset. With the sparsification, the data become a sparse matrix, and we hypothesis this sparse data is higher level cognition representation of the sound. We then do a linear component analysis on that sparsified data set.

Sparsification is done by resetting the entries to be "0" when the magnitudes (MFCC coefficients) are lower than a certain threshold. The threshold is based on the energy. 55 % of energy is retained in[4].

The following figures based on the SOFA dataset show the distribution gradually changes with the degree of the sparsification:

3.1.5 How to relabel the samples

ICA is an unsupervised learning method. This generative model doesn't come with the label information. But each column of the component matrix obtained through ICA. The source component matrix is a sparse matrix which means many of the entries are around zero. The entries with a higher value contain much of the information in the source matrix. Firstly, due to the ambiguity of the ICA (see section. 2.1.3), we can not make sure the sign of the source



(a) Without Sparsification (b) Weak Sparsification

Figure 3.3: With and without Sparsification

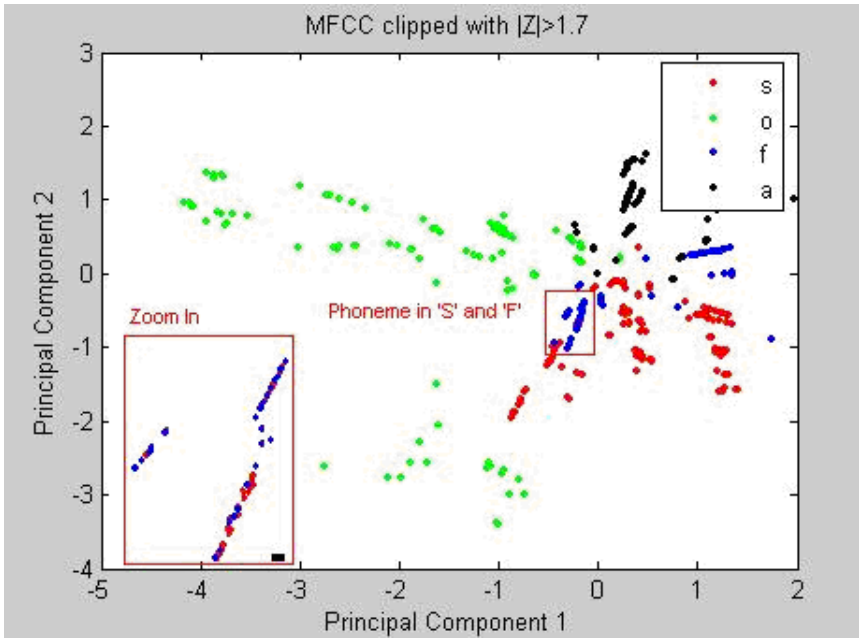


Figure 3.4: Energy based Sparsification revealing the ray structure

component and the corresponding column vector. We use a technique to find the positive direction and negative components[1].

$$A_i = (-1)^n A_i; s_i = (-1)^n s_i; n = \left\{ \begin{array}{l} 1 \; if \; abs(s_i) > s_i \\ 2 \; if \; abs(s_i) \leq s_i \end{array} \right\} \tag{3.5}$$

In words, these formulations find out in each source component if the maximum absolute value is negative or positive. If it is negative, we change the both the sign of the corresponding column in the mixing matrix and the sign of the source component. Fig. 3.5(b)shows that the color of the source component1,2,4 have been changed and accordingly the directions of the line vectosr have been reversed.
The next step, in each column of the source component matrix, we find out the position with *maximum value*, that to say, to find out which source component have the largest positive value in each column. We highlight this source component.

$$S'_{ij} = \left\{ \begin{array}{l} 0 \; else \\ 1 \; if \; s_{ij} = \max(s_{ij}) j = 1, \ldots\ldots n) \end{array} \right\} \tag{3.6}$$

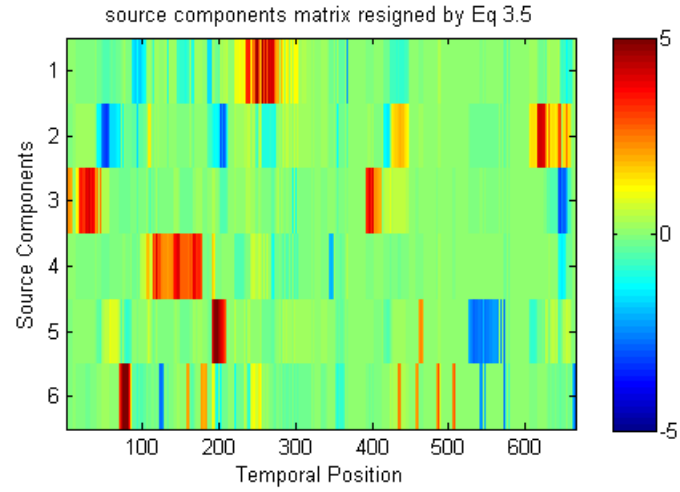
In Fig. 3.6,those highlighted vertical bar shows the source component is active(large value). We can label the column of MFCC by the source component.

Fig. 3.8 and Fig. 3.7shows that the similar /e/ sound in S and F can be reprensented by one component both in *ICA* and *Soft-Lost*. But the *Soft-Lost* is initialized with some random vectors. We need run serveral times to get a result more approximated to *ICA*. We replot the MFCCs in the time domain:

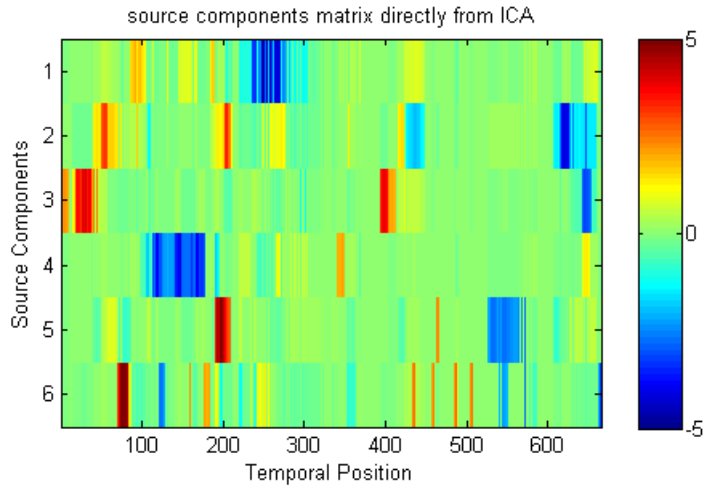
3.1.6 Test the generality of the Model

In this part, we use the mixing matrix A, which is $m \times m$(m is the number of source components) square matrix obtained by the *ICA* and *Soft-Lost* on the test data set which is another 'SOFA' letter utterance from the same speaker. We decompose the test data set with demixing matrix $B = A^{-1}$ and get the source components matrix by $S_{test} = BX_{test}$.
When we use the mixing matrix from the train set, we use a technique here to preserve more information related to the dataset. This is the same procedure

[1]This method is found in the DTU toolbook demo for the text classification and in my experiments it also helps to get a better result in classifications task, similarity measurement and in the 'sofa' demo



(a) The original source component matrix



(b) The resigned source component matrix by eq. 3.5

Figure 3.5: Illustration about the Equation. 3.5

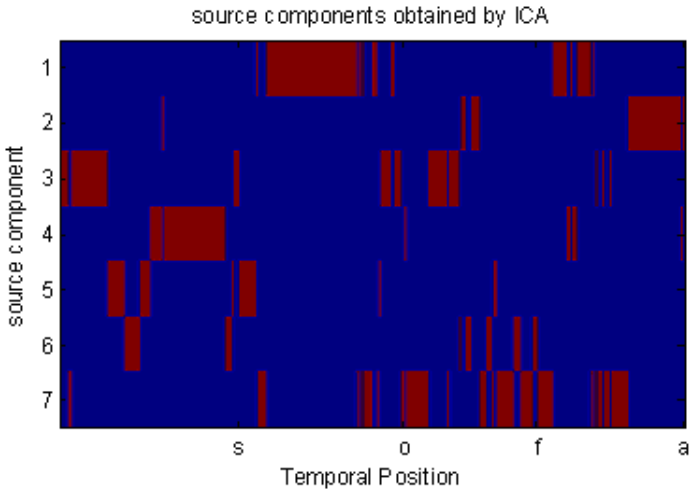


Figure 3.6: Translate the Fig. 3.5(b) by the eq. 3.6

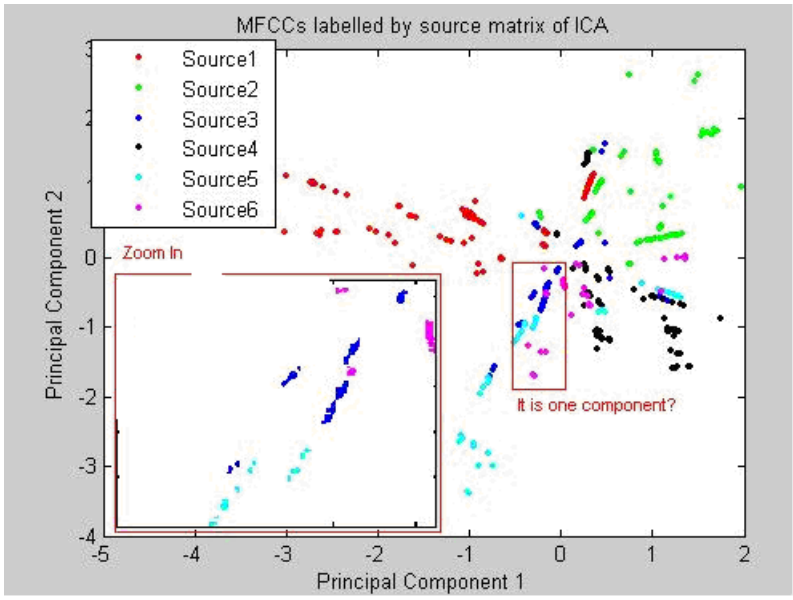


Figure 3.7: mfccs relabelled by the source matrix obtained by ICA

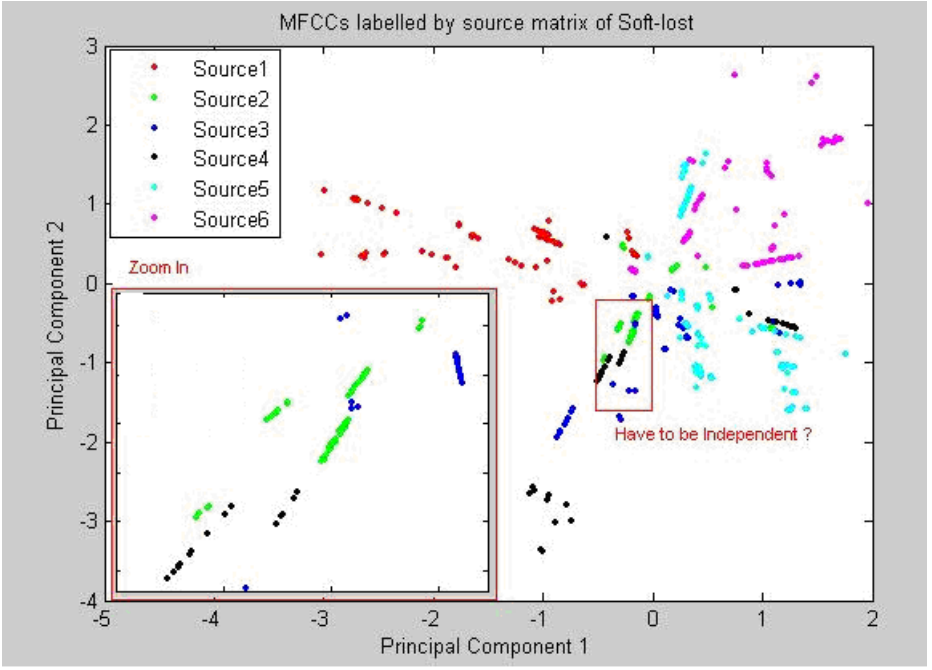
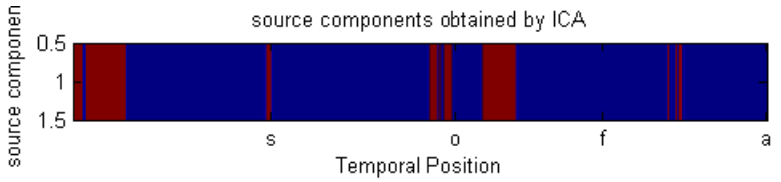
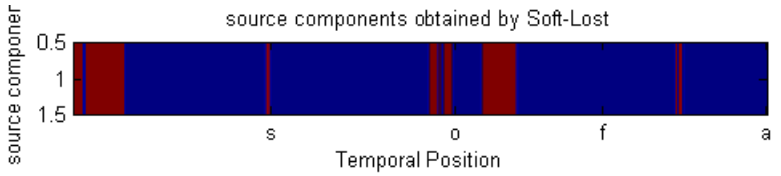


Figure 3.8: mfccs relabelled by the source matrix obtained by Soft-Lost



(a) Source component from ICA



(b) Source component from Soft-Lost

Figure 3.9: The temporal positon of the component indicates that this component is the phoneme /e/

we use on another experiment. 3.3:

After we perform SVD on the training data set, $\mathrm{X} = \mathrm{U}\Gamma\mathrm{U}^T$, we transfer the data to the new coordinate system by $Y = \widetilde{\mathrm{U}}^T_{training}X$ (see details on section. 2.2) and then we can transfer the A back to the MFCC domain:

$$\mathrm{A}' = \widetilde{\mathrm{U}}_{training}\mathrm{A} \tag{3.7}$$

When we perform SVD on the test data set $X_{test} = \mathrm{U}_{test}\Gamma\mathrm{U}^T_{test}$.we transfer the$\mathrm{A}'$ to the new coordinate system by :

$$\mathrm{A}'' = \mathrm{U}_{test}\mathrm{A}' \tag{3.8}$$

A is data set dependent, but in this way we can use the A cross on different data set.

We plot the same resulting figures as those in the train set:

Firsty, we plot the first two principal MFCCs labelled by the letters in Fig. 3.10.[2]: Then we plot the figures labeled by the source componens: Fig. 3.11 shows that those line vectors(Columns of A) from the training set are well aligned with the ray structure of the test data set.Some source components(Second and Third component in Fig. 3.11)have very few samples becasue the sparification is very high here.
Similarly ,we give the temporal position of one component at the locations of phoneme /e/:

From this experiment, we make some conclusions and interpretations about the procedures:

1. The energy sparsification threshold reveals the ray structure. But we find that some phonemes vanish faster as the sparsification threshold increases.(Fig. 3.11)

2. The MFCC length and frame length play a critical role to find the cognitive component. MFCC dimension higher than 16 can't genearate an accoutable result. The last step of MFCC-*Discrete Cosine Transform* is an approximation of the *Karhunen-Loeve transform*. It decorrelates the cofficients and reprensent them in a compact form. So from an informatioin point of view, 6 princinpal components of a 16 demensional data set will lose more information than that 6 princinpal components of 12 demensional data set.

[2]The fig. 3.10 and the fig. 3.12 are obtained with different sparsification threshold

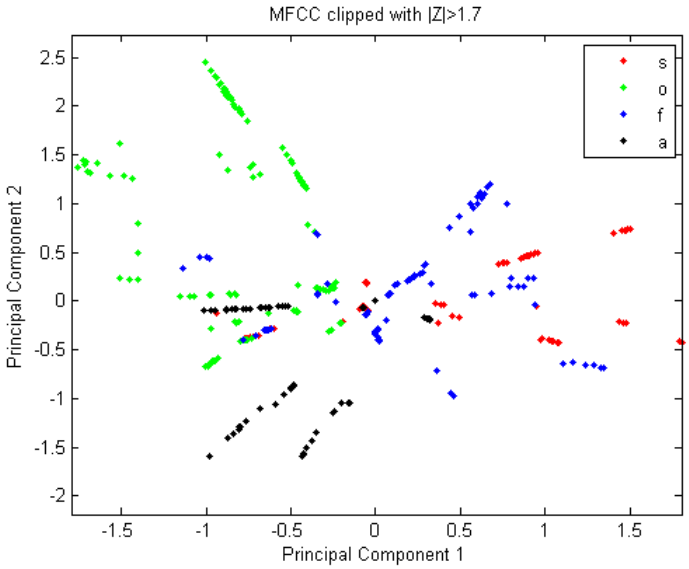


Figure 3.10: First two principal components of sparsified MFCCs on test set

3. The step to find the positive direction by Eq. 3.5 is very critical and helpful step. By this method, the error rate(section. 3.4) is reduced and it helps us to measure the distance in section. 3.3.

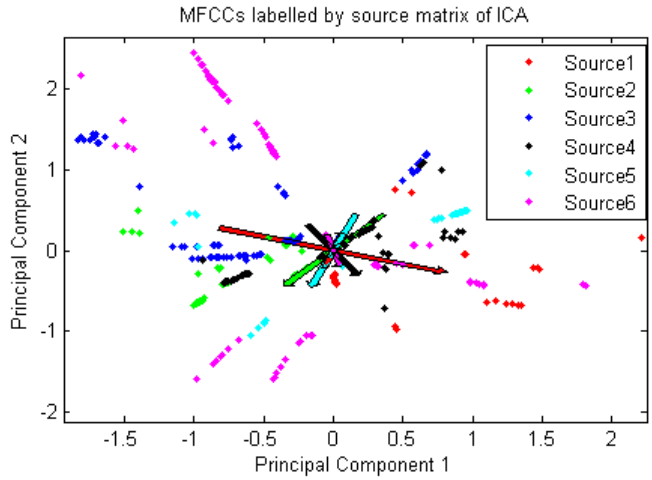
4. Transfering the mixing matrix A_{traing} from training data set to the new coordinate of test data set by making use of the $\widetilde{\mathrm{U}}$ in Eq. 3.7 provides a method to better use the data set information.

5. The *Soft-Lost* gives us a similar result in this experiment. But it is initialized by random line vectors and final results are different every time.

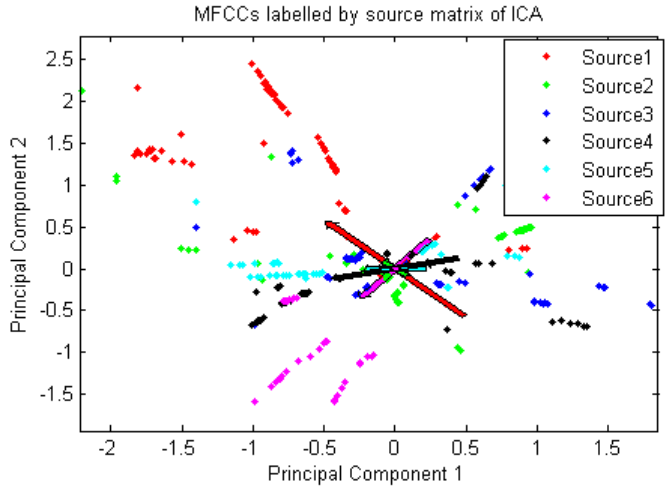
3.2 Cognitive Components Analysis on phonemes data set

3.2.1 TIMIT Speech Corpus:[12]

This is a corpus of high-quality recordings of read continuous speech from North American speakers. The entire corpus is reliably transcribed at the word and



(a) MFCCs of test data set labelled by the source components decomposed by the ICA without the step in Eq. 3.7 and Eq. 3.8



(b) MFCCs of test data set labelled by the source components decomposed by the ICA with the step in Eq. 3.7 and Eq. 3.8

Figure 3.11: With the step in Eq. 3.7 and Eq. 3.8, the line vectors in mixing matrix A of the training data set appear to be more aligned with the rays of test data set

surface phonetic levels. The TIMIT corpus contains the same 10 sentences from 630 speakers and these speakers fall into 8 different dialects. Those speakers have different gender, height, age, and race and education level[3] The TIMIT database is a very large database. In this experiments, I use only 4 female speakers falling into the first dialect.

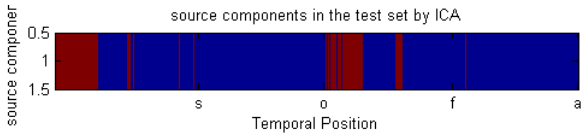
With the help of the TIMIT database, we can label the MFCC feature by the phonemes. In TIMIT database, these phonemes are already given. These following analysis are titled with the dataset composition. Data set are composed by different number of speakers and thus display the analysis result of *COCA* in different scenarios.

3.2.2 How the Dataset Composed:

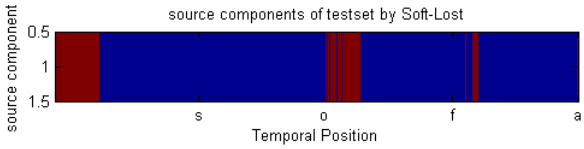
Based on the feature selection, the original speech wave from format files are extracted to feature files. The feature files are MFCCs and labelled by the phonemes.

We selected three vowels in these experiments. They are 'ao', 'ix' and 'ay'. In or-

[3]I extract only the first four attributes. I use the speaker ID as the name of the mat file to differ from each other. A sample of the filename looks like: dr1f23 (5.5) dml0.mat. Reversely, we can know which speaker used in the TIMIT database)



(a) One source component of test data given by the ICA



(b) One source component of test data given by the Soft-Lost

Figure 3.12: The source component decomposed by the mixing matrix indicates the same phoneme /e/ found in letter S and F

der to know how they sound like, we give words which contain these phonemes[4] :

ao bought bcl b AO tcl t
ix debit dcl d eh bcl b IX tcl t
ay bite bcl b AY tcl t

These phonemes are selected because they are vowels and show up more frequently in the TIMIT data base. So we can get more samples. These three phonemes are rarely adjected to each other in English. After we select the phonemes and speakers, we align the data (MFCC) in a regular format. The format of the data alignment[5]:

Figure 3.13: Dataset Format

3.2.3 Phonemes from one speaker

In the following experiment, we compose a dataset with those three phonemes from one speaker. The scatter plot Fig. 3.14 shows the linear ray structure of three phonemes. In this scatter plot, these phonemes are in different rays and well separated from each other. The scatters outside of the rays can be considered as noise.

Fig. 3.15 shows the scatter plot labelled by the source components.

Some comments and conclusions:

1. In the first figures. 3.14 we can see that the phonemes *congnitive structure* could be well aligned with the linear rays in the sparsified PCA dataset. That means a column in the mixing matrix. The directions of the ray structure means a cognitive stucture in *COCA*. But some phonemes('iy'and 'ay') may have different directions and some phonemes('ao') may have opposite direction.

[4]bcl and dcl are the closures of b and d,we can know how it sounds like by the words
[5]The sequence of the speakers and phonemes won't affect the result at all.We just make a short description here

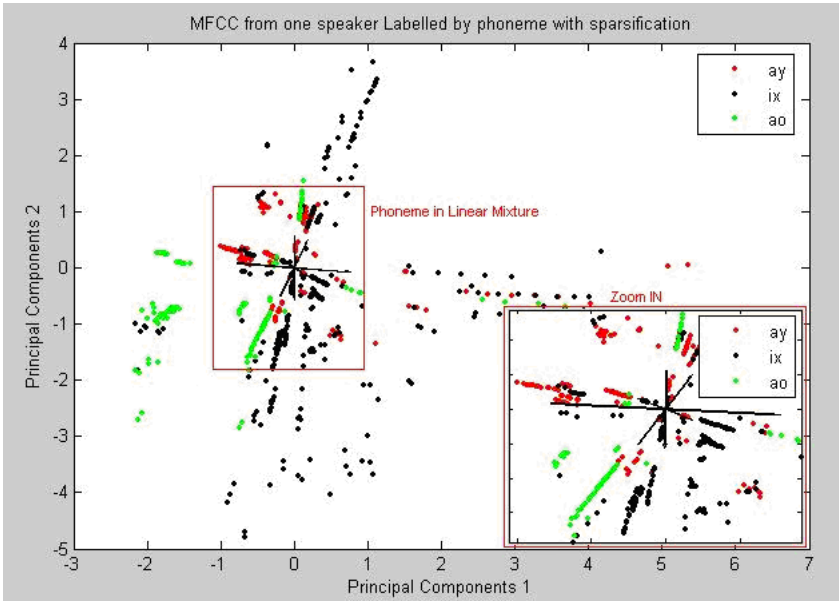


Figure 3.14: Mfccs of three phonemes from one speaker with sparsification

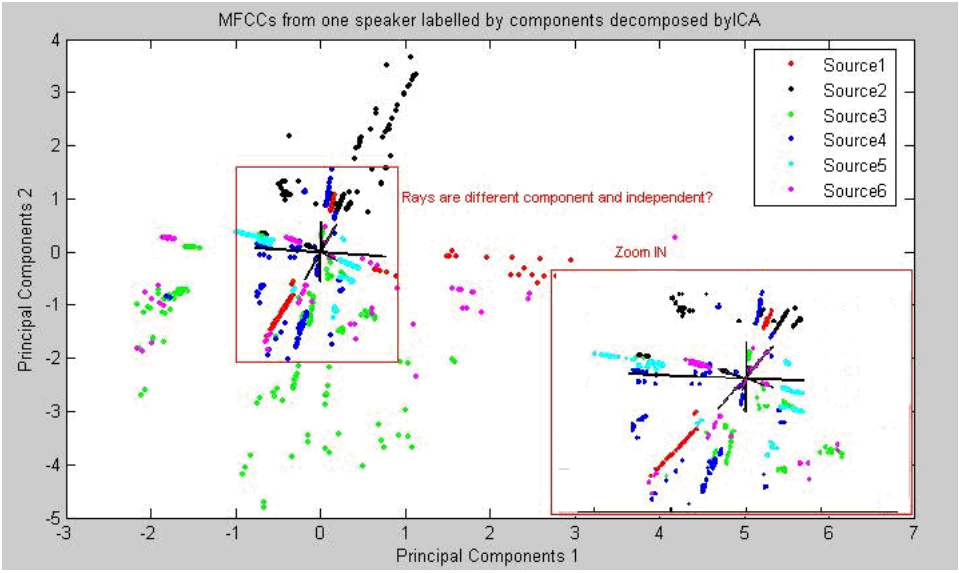


Figure 3.15: MFCCs relabelled by the source component matrix

2. In the second figure. 3.15 and the zoom-in Fig. 3.16(b), we can see that the ray structures can be well reprensented by the source components. But if the rays are too close, they are easy to confuse. Two phonemes, 'ay' and 'ao', partially lie in about the same ray, and then may have the same source component(fifth). It is a problem we have to solve.

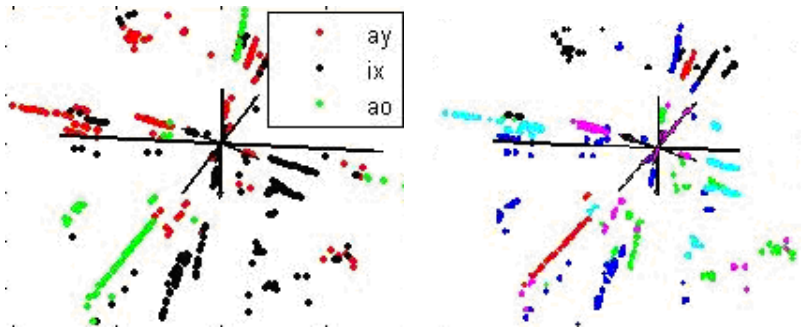
3.2.3.1 Test the generality on another speaker

In this experiment, we extract four phonemes of two speakers from the *TIMIT* database.One speaker is useed as the training data set, and another speaker is used as the test data set. Both of them are female and from the first dialect. These four phonemes are 's','aa','iy' and 'ae'.
We give the words containing these phonemes

aa bott bcl b AA tcl t
ae bat bcl b AE tcl t
s sea S iy
iy beet bcl b IY tcl t

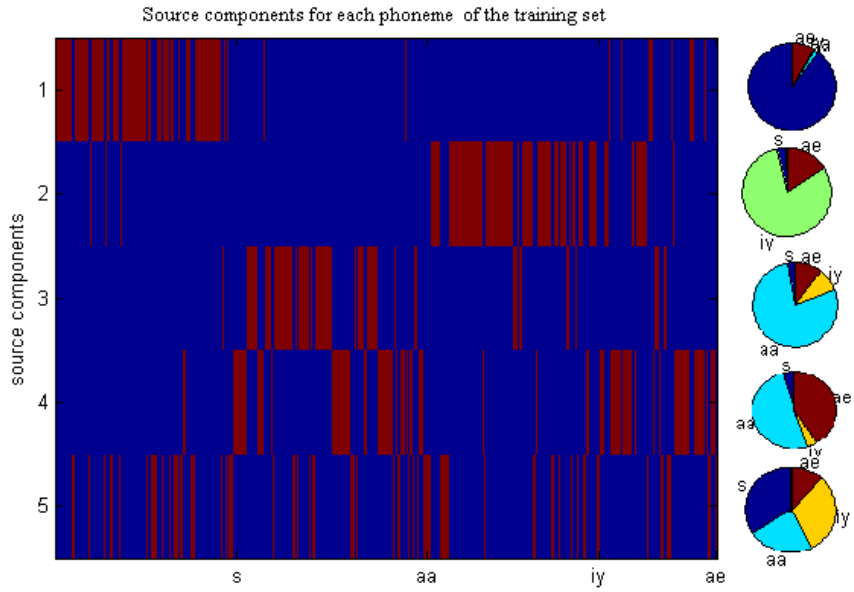
we decompose the training set by *ICA* to obtain the mixing matrix A, and then decompose the test set with A^{-1}.We plot the source components in the time domain to indicate which source component is associated with which phoneme as the Fig. 3.12 and Fig. 3.9 in the *SOFA* experiment.

The pies in the Fig. 3.17 give us a percentage about contributions of source

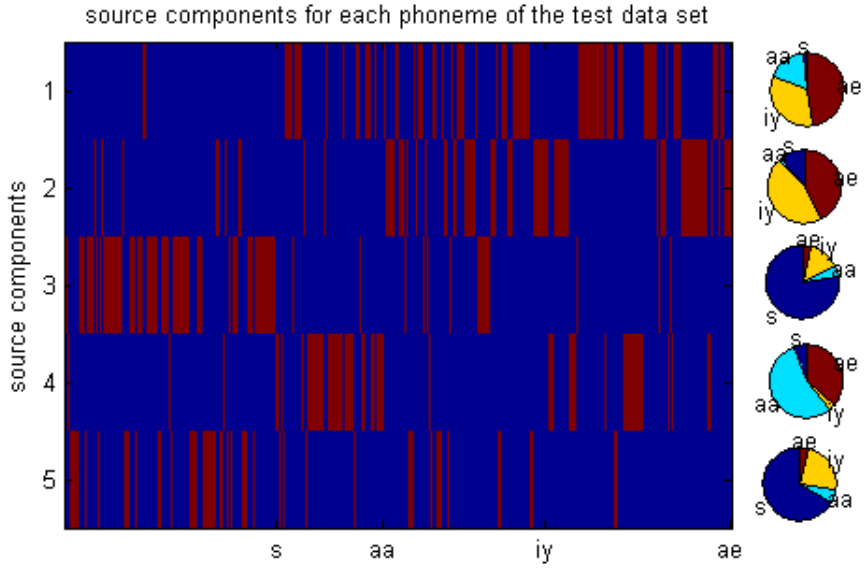


(a) Zoom-in of the Fig. 3.14 (b) Zoom-in of the Fig. 3.15

Figure 3.16: A zoom-in comparison



(a) The source components of the training set



(b) The source components of the test set

Figure 3.17: These four phonemes, 's','iy','aa' and 'ae', in this experiment, only hree phonemes 's','iy'and 'aa' in the training set can be repensented by source components, but only one phoneme 's' can be indicated in the test source components

component to each phonemes. Sparsification threshold in this experiment is very low which leave a lot of noise in the data set. The source component are sorted by the Energy,so the last component(with the least Energy) is more interfered by the noise.

Some conclusions:

1. The Cognitive components(phonemes)from one speaker can be well aligned with ray structures in the sparsified data set and repesented by the independent components given by the *ICA*.
2. Find the cognitive component with the model $A_{training}$ on another speaker is not easy. Result of the test set given in this experiment is not comparable with the result of training set.Only one of the four phoneme 's' can be indicated in the souce components. That is one reason why we make a classification task with only two phonemes in the section. 3.4.
3. The sparsification threshold is supposed to remove the instrinsic noise of the phonemes. But we find out the phonemes vanlish in different speed as the the threshold increases.

3.2.4 3 Phonemes from 4 speakers:

In this section, we extend our *COCA* to the phonemes samples from several speakers. We want to show the how the ray structures differ among speakers. Firstly, we give detailed plots of the principal MFCCs of different phonemes from several speakers, and then we show how one phoneme varies in differnt speakers in the sparsified MFCCs. Fig. 3.18 shows the ray structure still exists in the dataset made up by 4 speakers. The ray structures in Fig. 3.19 are more diverse and have more line vectors compared to the Fig. 3.18. It seems that the speaker difference are more obvious in the less important principal components.

In this part, we extract only one phoneme 'ao' and show how it varies in different speakers.In the following graphs, we could see if we can find any independent component to reprensent this phoneme.

We plot the source component in time domain in Fig. 3.24:

Some conclusions:

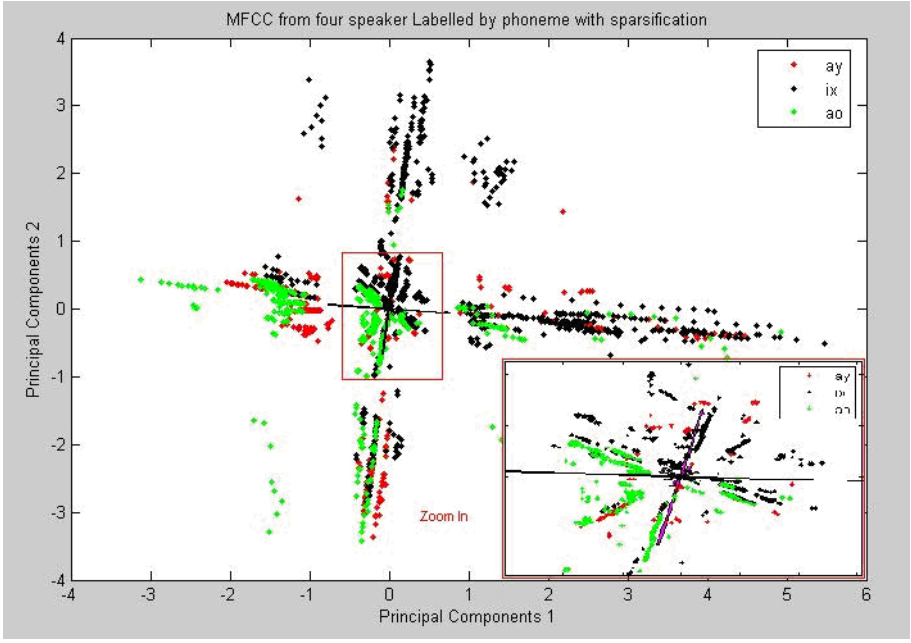
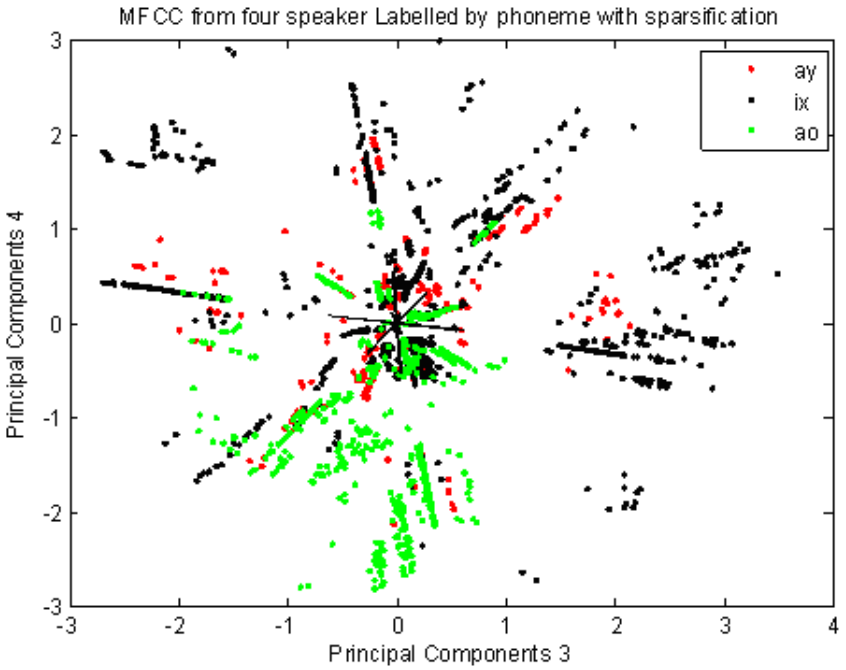
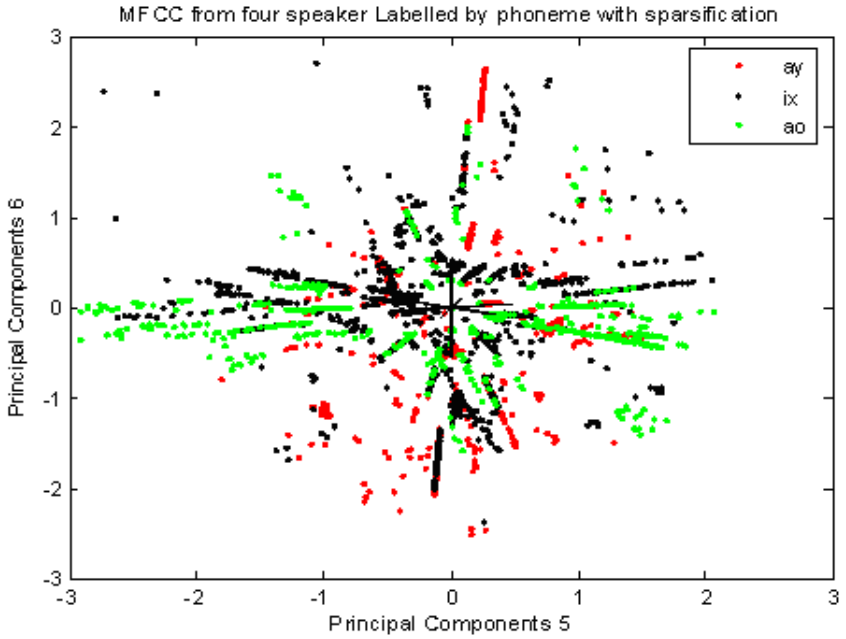


Figure 3.18: The first two principal MFCC from 4 different speakers



(a) The third and forth principal MFCCs



(b) The fifth and sixth principal MFCCs

Figure 3.19: The higher(less important) principal mfccs show more rays among speakers

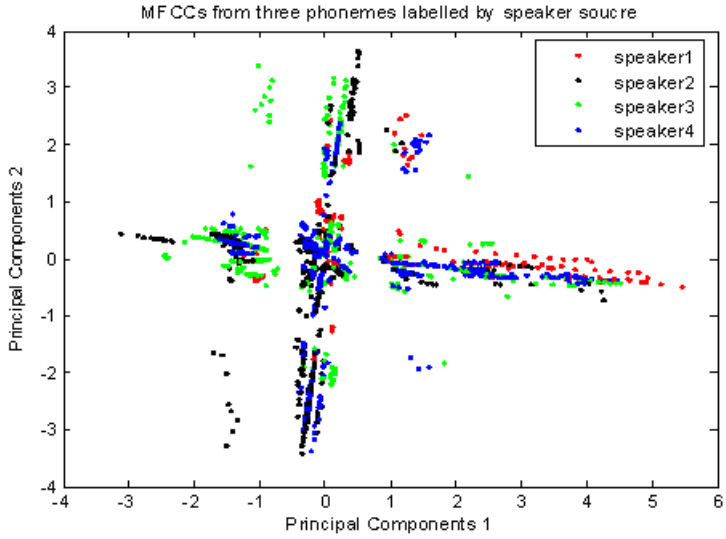


Figure 3.20: The MFCCs labeled by the speaker

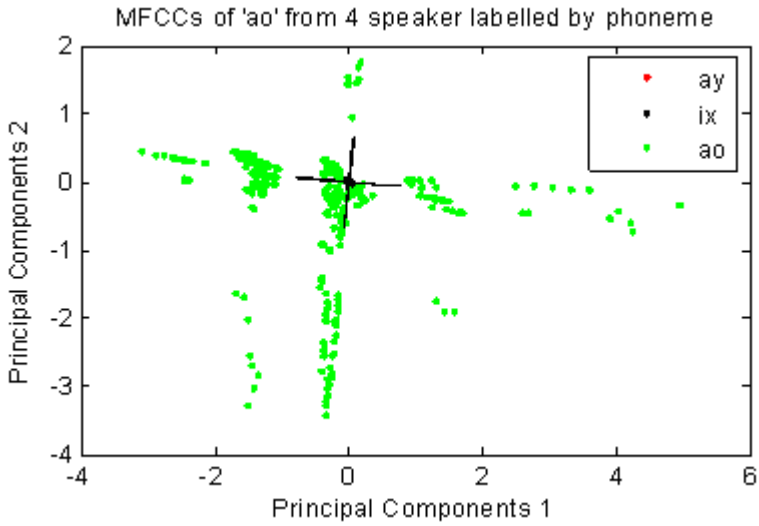


Figure 3.21: One phoneme 'ao' from 4 different speakers.Notice that the other two phonemes are removed compared with the Fig. 3.18

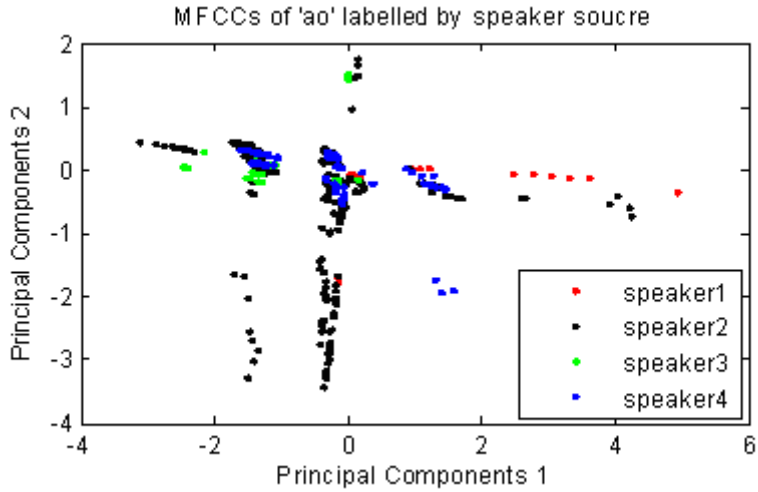


Figure 3.22: The mfccs of ao labelled by the speaker

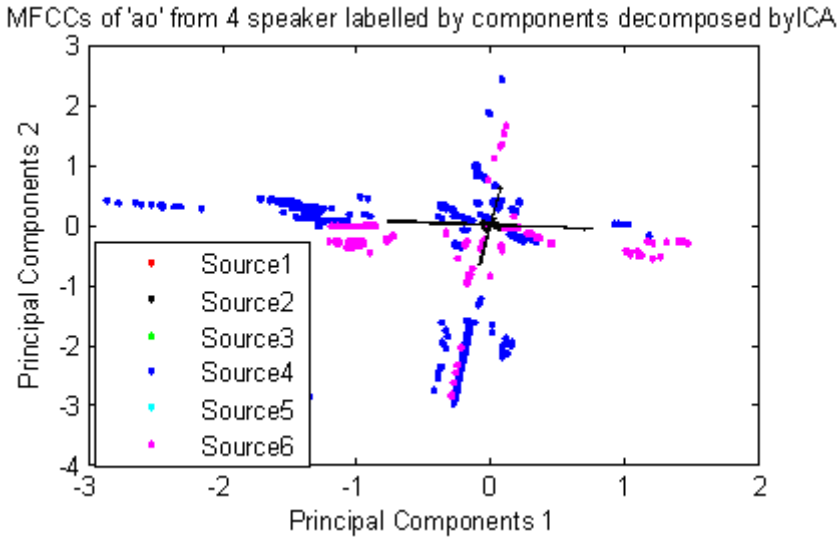


Figure 3.23: 'ao' is reprensented by two independent source components

1. Fig. 3.21 shows that the phoneme 'ao' have different ray directions in the sparsified data set.But Fig. 3.24 and Fig. 3.23 show the phoneme 'ao' from four speakers can be indentified by two source components.

2. The speaker1 in the Fig. 3.22 has very few samples left. Because of the sparsifiatioin threshold, we lost some information from "weak" speaker. These samples removed by sparsification may from a ray structure. So the two source components in the Fig. 3.23 may only partially reprensent the *Cognitive components* from 4 speakers.

3.2.5 Fisher Linear Discriminant Analysis on two phonemes from two speakers

This step is closly related to the section. 3.3. From the previous figures, we have known the MFCCs of one phoneme from several speakers are very different. Phonemes from different speakers tend to lie in different rays in the sparsified data set. If we have two phonemes and two speakers, we can do a phoneme classification task and also a speaker classification task. With the label information given by the *TIMIT* database, we thought that we could use a supervised leaning step to know what the speaker difference and phoneme difference are. It might provide us some clues for the unsupervised learning.

The *Fisher Linear Discriminant Analysis* provides a technique to know a *separatability* of two classes(section. 2.4). This linear solution of the classfication is different from the *ICA*. Becasue *ICA* was a transferation of the coordinate system. Our idea is that we use this technique to know which kernel function increase the *separatability* of phonemes more than the *separatability* of speakers.Then we can use a *kernel PCA* before the *ICA*. But this idea is not succeful in our experiment, the ratio of the *separatability* between the phoneme and speaker does't show an increase after we use a *Kernel Fisher Linear Discriminant Analysis*. Here we give a plot from the experiments we did on the phonemes 'ao' and 'ix' from 12 speakers. This analysis was done on the sparsified dataset after PCA, which means it is already been decorelated. In this Fig. 3.25 shows the

Figure 3.24: 'ao' from 4 speakers can be indicated by one source component

separatability given by LDA. The phoneme separatabilities are slightly larger than the speaker separatabilities among the 6 pairs of speakers. More over, these two phonemes 'ao' and 'ix' have a low error rate in our unspervised classification task.

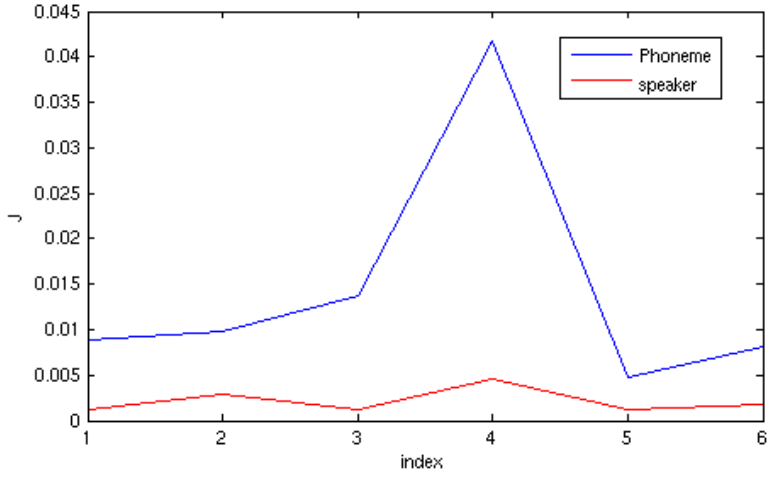


Figure 3.25: Separatability in LDA on two phonemes and speakers

3.3 Similarity measurement & Invariant Cue:

[6] ICA decomposes the MFCC dataset to a mixing matrix and source component matrix. Source components are sparse matrix with few information. Figure . 3.14 shows that the phoneme can be represented in a ray structure in the sparsified dataset. This ray direction corresponds one column vector in the mixing matrix.
Invariant cue is a well known phenomenon in the human speech perception. It describes that even the the accoustics of the phoneme are very different among speakers(fig. 3.26). Human auditory can still perceive them as one phoneme. The mixing matrix of ICA trained by the different phoneme samples and different speaker contains the phoneme and speaker information. If these rays in our *COCA* can align with the cognition of phoneme. We believe that it should be able to show these invariant cues of phoneme and reveal that how we can perceive the utterance from different speaker as the same phoneme. The mean-

[6] In this experiment, the MFCC dimension is 8. MFCC Length of 12 can also show a similar result. Other feature extraction parameters are the same as in the SOFA experiment.

ings of the dimensions in the source component matrix are determined by the new coordinate system. The data in the new coordiantes $Y = \widetilde{\mathrm{U}}^T X$ need to be transfered back to the MFCC domain. Then we reconstruct this mixing matrix in the original MFCC domain by the equation:

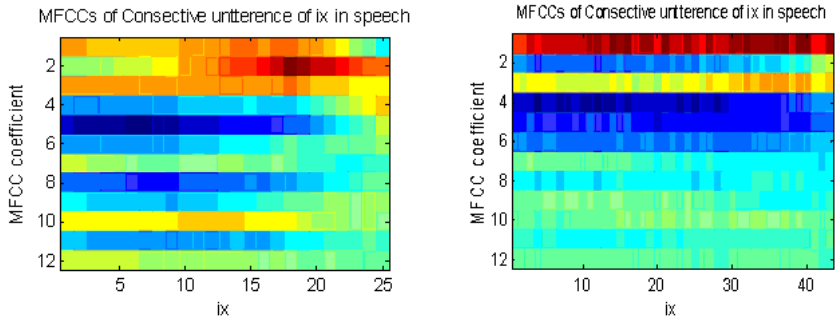
$$\mathrm{A}^{'} = \widetilde{\mathrm{U}}\mathrm{A} \tag{3.9}$$

In Eq. 3.9, A is a $m \times m$ square matrix and the $A^{'}$ is a $M \times m$ matrix. m is the dimension number of the source components and M is the dimension number of *MFCC* coefficients–the dimension number of the data set before dimension reduction.

Experiment:

First, we extract one phoneme samples of one speaker in the TIMIT database. Then we decompose the data set by the *ICA* and *Soft-Lost* to obtain the mixing matrix. Then we transfer the matrix A back to the MFCC domain section. 3.9.

Our experiments are based on the first dialect set. There are eight dialects in the TIMIT database. Each contains certain number of speakers of the TIMIT database. There are 48 speakers and thus we get 48 mixing matrices for each phoneme. We measure the similarities of the matrices two by two. The entire combinations are $(48 * 47)/2 = 1128$ for the speaker difference within phoneme and $(48 * 48)/2 = 1152$ for the phoneme difference between phonemes. Finally we get the mean value and standard deviation of the difference.



(a) Mfccs of one 'ix' from a consective speech (b) Mfccs of another 'ix' from a consective speech

Figure 3.26: 'ix' mfccs in two different unterance could be very different

3.3.1 Distance Measurements:

3.3.1.1 Euclidean distance

The distance measurements on these vectors:
The Euclidean Distance between two points$P = (p_1, p_2, \ldots\ldots, p_n)$ and $Q = (q_1, q_2, \ldots\ldots, q_n)$ in Euclidean *n-spaces*, is defined as:

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots\ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \qquad (3.10)$$

This euclidean distance in Eq. 3.10 can only measure the distance between two vectors.So we sum up the column vectors in the A and then measure the distance between the sum-up vectors. Fig. 3.28 gives an example about how to calculate the *Euclidean distance* of the sum-up vectors.

Figure 3.27: Decompose one phoneme data set to get mixing matrix to reprensent the phoneme

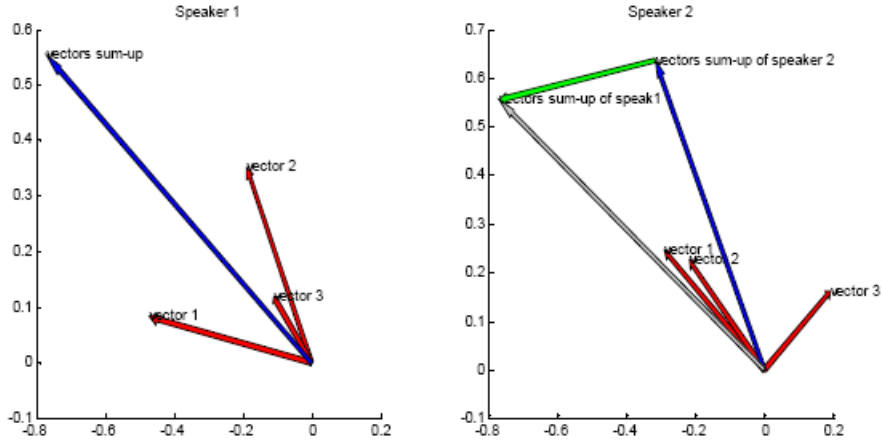


Figure 3.28: Decompose one phoneme data set to get mixing matrix

Hausdorff distance
Definition:

The Hausdorff distance, or Hausdorff metric, measures how far two compact non-empty subsets of a metric space are from each other. It is widely used in the pattern or shape matching in computer vision.

Motivation:

The motivation of using this new distance measure is because that the ray structure in the sparsified data set can be considered as a pattern. The Euclidean distance is very simple and can not take the pattern into consideration. This pattern reveals the linear mixtures of some sparse source components which represents high level cognition of sound in human brain.

The calculation steps of the Hausdorff distance are described as fellows:

If$x \in \chi$ the *distance* from x to B is $d(x, B) = \min b \in B\{d(x, b)\}$The distance from A to B is $d(A, B) = \max x \in A\{d(x, B)\}$.We can see from the figure bellowed that this *distance function* is not symmetrical.

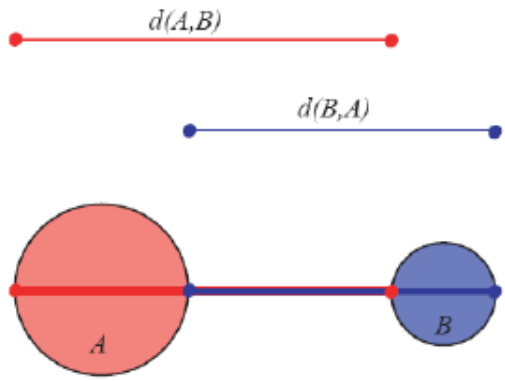


Figure 3.29: hausdorff distance

The Hausdorff is definded as:

$$H(A,B) = \max(\min(||a-b||\,\forall b \in B)\forall a \in A) \tag{3.11}$$

Modified Hausdorff Distance

For the extraction of descriptors which is invariant to translation, rotation and scale, an upgraded Hausdorff distance comes out.
In cases of translation, all the nearest neighbor distances are increased by the same amount:

$$H(\tilde{A},B) = \max(\min(||a-b||\,\forall b \in B)\forall a \in A) - \min(\min(||a-b||\,\forall b \in B)\forall a \in A) \tag{3.12}$$

$$H(\tilde{A},B) = 80th\%(\min(||a-b||\,\forall b \in B)\forall a \in A) - 20th\%(\min(||a-b||\,\forall b \in B)\forall a \in A) \tag{3.13}$$

To make the distance symmetrical, we sum the distance from A to B and the distance from B to A and then get the final one:

$$D(A,B) = H(A,B) + H(B,A) \tag{3.14}$$

3.3.2 Result Analysis

In this experiment, the phonemes are still grouped in pair. There are six pairs:

{'ix','ao'},{'ay','ao'},{'ix','ay'},{'ao','iy'},{'iy','ih'},{'ih','ix'};

iy beet bcl b IY tcl

ih bit bcl b IH tcl

The first four pairs sound very unsimilar but the last two are very similar phonemes.

In table. 3.1 , D_s denotes the distance between speakers. D_p denotes the distance between two phonemes in each pair. The first and second means the order of

Table 3.1: Euclidean distance measured within phonemes and between speakers

	ix&ao	ay &ao	ix&ay
D_s within first phoneme|std	3.15|0.8	3.27|1	3.15|0.8
D_s within second phoneme |std	3.2|1	3.20|1	3.27|1
$D_p$between two phonemes |std	3.71|0.8	3.52|0.95	3.4|0.87
	ao &iy	iy &ih	ih&ix
D_s within first phoneme|std	3.2|1	3.15|0.92	3.29|0.86
D_s within second phoneme |std	3.15|0.92	3.30|0.86197	3.15|0.84
$D_p$between two phonemes |std	4.16|0.87	3.4552|0.82	3.2|0.75

phoneme in the pairs. *Std* in the table is short for *standard deviation*. It is used for the furthur analysis:

The table. 3.1 gives us a mean and the standard deviation of the Euclidean distance. We suppose that both distance samplings are from a population of *Normal distribution*, [7]i.e.$D_s \sim N(\mu_s, \sigma_s^2)$ and$D_p \sim N(\mu_p, \sigma_p^2)$. In this way, we can get a distribution of their difference $D_s - D_p \sim N(\mu_s - \mu_p, \sigma_s^2 + \sigma_p^2)$. We can get a probability of $D_s > D_p$ by $1 - \Phi(0)$. Then we get a new table about the probability based on the table. 3.1:

Table 3.2: The probability of phoneme distance larger than the speaker difference within phonemes based on the Table. 3.1

	ix&ao	ay &ao	ix&ay	ao &iy	iy &ih	ih&ix
$P(D_p > D_s)_1$	0.69	0.58	0.59	0.77	0.59	0.48
$P(D_p > D_s)_2$	0.65	0.59	0.54	0.79	0.55	0.53

In the experiment. 3.2, we have found out in the less important principal components(Fig. 3.19and Fig. 3.19(a)) the ray structure is more diverse. We have sorted the column vectors in the A by the importance of the vectors in the section. 2.1.3. We made an envision that the speaker difference is more obvious in the less important column vectors. The main vectors are desicive in the Cognitive component. In the next part of the similarity measurement, we split the square matrix $A(8 \times 8)$ to two parts . Firstly we measure the first 4 column vectors and then measure the following 4 column vectors. Then we translate the table. 3.3and. 3.4 to the probability: In this experiment, the *MFCC* length is chosen to be 8 and the dimension of source component matrix is also 8. In the simulation, a result based on the 12 MFCC length and 6 components also generate a similar result. We choose 8 MFCC length and 8 components, beca-

[7]By ploting the histgram of the result,they really look like a normal distribution

Table 3.3: Euclidean distance measured within phonemes and between speakers on the last 4 column vectors of the A

	ix&ao	ay &ao	ix&ay
D_s within first phoneme|std	2.19|0.57	2.02|0.52	2.19|0.57
D_s within second phoneme |std	2.04|0.56	2.04|0.56	2.02|0.52
$D_p$between two phonemes |std	2.19|0.47	2.05|0.45	2.12|0.47
	ao &iy	iy &ih	ih&ix
D_s within first phoneme|std	2.04|0.56	2.22|0.57	2.15|0.55
D_s within second phoneme |std	2.22|0.57	2.15|0.55	2.2|0.57
$D_p$between two phonemes |std	2.27|0.49	2.2|0.48	2.18|0.56

Table 3.4: Euclidean distance measured within phonemes and between speakers on the first 4 column vectors of the A

	ix&ao	ay &ao	ix&ay
D_s within first phoneme|std	2.34|0.63	2.53|0.78	2.35|0.63
D_s within second phoneme |std	2.60|0.78	2.60|0.78	2.53|0.78
$D_p$between two phonemes |std	2.86|0.60	2.87|0.70	2.60|0.66
	ao &iy	iy &ih	ih&ix
D_s within first phoneme|std	2.60|0.78	2.52|0.73	2.52|0.69
D_s within second phoneme |std	2.52|0.73	2.52|0.69	2.35|0.63
$D_p$between two phonemes |std	3.22|0.65	2.69|0.65	2.47|0.57

Table 3.5: The probability of phoneme distance larger than the speaker difference within phonemes based on the Table. 3.3

	ix&ao	ay &ao	ix&ay	ao &iy	iy &ih	ih&ix
$P(D_p > D_s)_1$	0.50	0.52	0.46	0.62	0.49	0.52
$P(D_p > D_s)_2$	0.58	0.51	0.56	0.53	0.53	0.49

Table 3.6: The probability of phoneme distance larger than the speaker difference within phonemes based on the Table. 3.4

	ix&ao	ay &ao	ix&ay	ao &iy	iy &ih	ih&ix
$P(D_p > D_s)_1$	0.72	0.63	0.61	0.73	0.57	0.48
$P(D_p > D_s)_2$	0.61	0.60	0.53	0.77	0.57	0.56

sue in this way, we don't lose information by the PCA reduction. We hope the mixing matrix can reprensent a high level cognition of phonemes and help us to find the *Invariant cue*. We believe the similarity measured by the two *distance*

functions can explain the *Invariant cue* in our experiment. From the analysis of the result, we can make fellowing conclusions:

1. The mean value of distance between phonemes are found larger than the speaker distance within phonemes. A probability result was given in the tabel.3.2. The result are aligned.
2. By sorting the *importance* of the column vectors section. 2.1.3, we find that the more important column vectors is the desicive factor for the difference between phonemes . In the less important vectors, the two distances have no differences.
3. The *Hausdorff metric* don't give us a difference as the Euclidean distance. These *Hausdorff distances* is measuring the pattern shape.

Table 3.7: The Hausdorff distance doesn't show a difference

	ix&ao	ay &ao	ix&ay	ao &iy	iy &ih	ih&ix
$H(D_{s1})$	1.91	1.91	1.82	2.02	1.88	2.02
$H(D_{s2})$	2.02	1.82	2.02	1.91	2.02	1.88
$H(D_p)$	2.01	1.94	1.94	2.05	1.98	1.98

3.4 Unsupervised Classification

3.4.1 The experiment setup[8]

We construct two categories of phonemes both of which are made up of 6 groups of two phonemes.
In the first category, two phonemes in each group are similar from the perception point of view.

{'ih','ix'},{'s','sh'},{'f','v'},{'ae','eh'},{'n','l'},{'iy','ih'}

The second category consists of six groups with distinguishable pronouncing phonemes:

[8]In this experiment, the MFCC length is 12.Other feature extraction settings are the same as the 'sofa' experiment

{'ix','ao'},{'ay','ao'},{'ix','ay'},{'oy','ix'},{'ao','iy'},{'d','k'}

Before we start out the classificatioin task, we have to make sure the phonemes have the same number of samples. When one phoneme has samples more than other phonemes, due to our error rate calculation procedure(see eq. 3.15), the outnumbered phonemes will dominate in the source component assignment. So we have to make the phoneme even[9] by picking off some samples from the outnumbered phoneme. If the shortest phoneme has N samples, we randomly select N samples for other longer phonemes, in this way, we can keep the speaker information in the dataset shown in the Fig. 3.30.

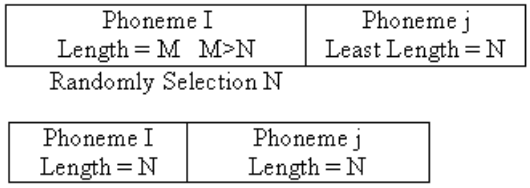


Figure 3.30: Randomly select the samples to make the phonemes even

Our task is to use unsupervised learning method to classify the phonemes under different conditions or settings(In Fig. 3.31). In the previous chapters, we have introduced two unsupervised learning methods, ICA (noiseless) and Soft-Lost. All of them have the same model:

$$X = AS \quad S = A^{-1}X$$

3.4.2 Some Instructions about the experiment

We use the unsupervised learning on one data set *training set* and use the obtained mixing matrix A to decompose another data set *test set*. In this way, we can test the generality of the unsupervised learning method like in the SOFA experiment.

Based on the composition(section. 3.2.2) of the data set. The classification could be implemented in different conditions and with different settings.

[9]The side effect of the random selection is that it contributes a variation to the error rate. The randomly selected data can not represent the original dataset completely, that is to say, we may miss the some important information in the phoneme. But We may be able to eliminate this random variation by a mean value

These conditions and settings included in our experiment are speaker dialect; speaker gender of the training and test data set; speaker number of the trainging and test data set; component numbers (Model dimension) and sparsification threshold. We give an example of these settings in Fig. 3.31:

In this experiment, we choose sparsification threshold to be 0.8. Sparsification is supposed to remove the intrinsic noise[4]of speech. This sparsification threshold is two times lower than the previsous *SOFA* experiment and *Similarity* experiment. Becasue the dataset contains different phonemes from different speaker. Keeping the sparsification lower helps better keep the "*weak*" phoneme or speaker information in the data set. Since we already found out in the SOFA experiment that some phonemes are easier to be removed by the sparsification threshold. The dialect and sex of the training speakers are fixed to be "dialect one" and "Female". The dialects and sex of the test data set are chosen between "dialect one" and "dialect two" , "male" and "female" from *TIMIT* database. The model dimension*(source component number)* is constantly 6. 6 is choosen based on the simulation results. We sweep the source component from 2 to 8 and the error rate was lowest around 6.

we randomly select the speakers and run the experiment 10 times in each conditioins, therefore the error rate in the fellowing tables are an average of 10 trials.

Field	Value
trainerror	0.18312
trdialect	'dr1'
trainspeakernum...	1
trsex	'f'
testsex	'f'
test_dia	'dr1'
trainspar	0.8
testspar	0.8
testerror	0.20049
test_speaker_nu...	1
components	6
trainsamples	314
testsamples	138

Figure 3.31: One example of the parameter setting of the experiments

3.4.3 How to evaluate the result

3.4.3.1 Error Rate:

Based on the Eq. 3.6, we get the new source matrix $S_{ij}^{'}$, This new matrix indicates which phoneme one source component responses most to. We can assign the souce components to phonemes by counting the $S_{ij}^{'}$ in the range of each phoneme. This process can be described by the fellowing equation:

$$\begin{aligned} Error_i &= N - \max(\sum_{j \in C} S_{ij}^{'}) \\ Errorrate &= \frac{\sum\limits_{i=m} Error_i}{M \times N}\% \end{aligned} \tag{3.15}$$

In which, the N is the length of the samples, M is the number of the components which is 6 in our experiment. The "C" is the class number, in our experiment, it is 2.

3.4.3.2 Result Analysis on the Error Rate:[10]

1. In this part, we have four experiments, i.e table. 3.8, table. 3.9, table. 3.10 and table. 3.11. We get the mixing matrix A from 1 speaker and 3 speakers and then use this model on the test set with 1 and 3 speakers.
   (a) First, we find out that the results are aligned with the similarities of the phonemes. The similar phonemes(group one) have higher *error Rate* than the unsimilar group.
   (b) Secondly, with the increase of the speaker numbers, the error rates raise in both train set and test set, but the difference is lower than 0.1 in most cases.
   (c) When we use a model from training set with 3 speakers on a test set with 1 speakers. It neither improves or deteriorates the result. Vice visa, when we use a model from 1 speaker on a test set with 3 speakers.
   (d) There are some random factors in these experiments but their influences are lower than 0.05 in most cases.

[10]These Error rate are the average of 10 trials
[20]Means the number of the speaker in the training data set
[21]Means the the number of speaker in the test set and the corresponding training set is right above

Table 3.8: Error Rate

	ih&ix	s &sh	f&v	ae &eh	n &l	iy&ih
Train set with 1 speaker[20]	0.3581	0.2739	0.2757	0.3008	0.2559	0.3194
Test set with 1 speaker[21]	0.3611	0.248	0.2893	0.3465	0.2598	0.348
	ix&ao	ay &ao	ix&ay	oy &ix	ao &iy	d&k
Train set with 1 speaker	0.1831	0.1974	0.2819	0.2158	0.2113	0.2928
Test set with 1 speaker	0.2005	0.2517	0.3067	0.2925	0.1866	0.2898

Table 3.9: Error Rate

	ih&ix	s &sh	f&v	ae &eh	n &l	iy&ih
Train set with 1 speaker	0.3671	0.243	0.2549	0.3311	0.2597	0.2947
Test set with 3 speaker	0.4077	0.2782	0.3061	0.3849	0.2549	0.3479
	ix&ao	ay &ao	ix&ay	oy &ix	ao &iy	d&k
Train set with 1 speaker	0.1379	0.2103	0.2642	0.2423	0.1842	0.2998
Test set with 3 speaker	0.2151	0.3354	0.3062	0.2991	0.244	0.34

Table 3.10: Error Rate

	ih&ix	s &sh	f&v	ae &eh	n &l	iy&ih
Train set with 3 speaker	0.3821	0.2937	0.3234	0.3726	0.3079	0.3464
Test set with 3 speaker	0.4036	0.2842	0.3341	0.3977	0.303	0.317
	ix&ao	ay &ao	ix&ay	oy &ix	ao &iy	d&k
Train set with 3 speaker	0.2146	0.2702	0.2942	0.2565	0.1698	0.3548
Test set with 3 speaker	0.2234	0.2933	0.2937	0.2819	0.2422	0.3514

Table 3.11: Error Rate

	ih&ix	s &sh	f&v	ae &eh	n &l	iy&ih
Train set with 3 speaker	0.3891	0.2974	0.3254	0.3568	0.2515	0.342
Test set with 1 speaker	0.3564	0.2571	0.2737	0.3508	0.2204	0.33
	ix&ao	ay &ao	ix&ay	oy &ix	ao &iy	d&k
Train set with 3 speaker	0.2575	0.2859	0.2848	0.3065	0.284	0.32
Test set with 1 speaker	0.2111	0.2408	0.2794	0.2892	0.239	0.2745

2. In this part, we switch the test data set to "dialect two" and "male" which are contrat to the trainning set.By comparing the result in table . 3.12 with table. 3.8 and table . 3.13 with table. 3.10., we can conclue that the *high-level cognitive differences*(dialect and sex differences) does't

[22]Due to the sparsification reason, some phonemes sample is empty

contribute more difference in the low-level phoneme cognition.

Table 3.12: Error Rate

	ih&ix	s &sh	f&v	ae &eh	n &l	iy&ih
Train set with 1 speaker	0.3629	0.2844	0.2651	0.2818	0.2621	0.3056
Test set with 1 speaker	0.38	0.2475	0.296	0.3421	0.2605	0.3353
	ix&ao	ay &ao	ix&ay	oy &ix	ao &iy	d&k
Train set with 1 speaker	0.1507	0.2505	0.29	0.2611	0.22	0.282
Test set with 1 speaker	0.2242	0.2631	0.3305	NaN[20]	0.2077	0.3033

Table 3.13: Error Rate

	ih&ix	s &sh	f&v	ae &eh	n &l	iy&ih
Train set with 3 speaker	0.3899	0.2955	0.3191	0.3765	0.2771	0.3674
Test set with 3 speaker	0.4245	0.2917	0.2971	0.3979	0.2762	0.3382
	ix&ao	ay &ao	ix&ay	oy &ix	ao &iy	d&k
Train set with 3 speaker	0.2344	0.2406	0.2926	0.2818	0.2129	0.3639
Test set with 3 speaker	0.2196	0.2957	0.3717	0.3002	0.1958	0.3659

3. The *Soft-Lost* perform very poorly in this task. For one reason, the sparsification threshold is two times lower than the 'SOFA' experiment. A lot of noise will interfere the performance of the *Soft-Lost*. *Soft-Lost* only considers the covariance structure, but the *ICA* take the *high order statistics* into account. We give an example of the error rate result from the *Soft-Lost* method.

Table 3.14: Error Rate from Soft-Lost Decomposition

	ih&ix	s &sh	f&v	ae &eh	n &l	iy&ih
Train set with 1 speaker	0.3991	0.3829	0.3894	0.3859	0.3912	0.3952
Test set with 1 speaker	0.3897	0.3113	0.3223	0.3519	0.3886	0.3716
	ix&ao	ay &ao	ix&ay	oy &ix	ao &iy	d&k
Train set with 1 speaker	0.3882	0.3656	0.3523	0.3595	0.4022	0.3320
Test set with 1 speaker	0.3132	0.3267	0.3606	0.3061	0.2750	0.3200

Chapter 4

Conclusion and Future work

4.1 Conclusion

1. This thesis describes the process of the *COCA* on the speech signals. The phoneme *Cognitive Component* can be identified by *Independent Component Analysis* and *Soft-Lost* in the 'SOFA' experiment.

2. *Soft-Lost* is used as another unsupervised technique for our *COCA*. It provides a similar result as the *ICA* in the 'SOFA' experiment. But it works well only in a very linear cloud of data when the sparsification is enough.

3. Because some information was lost by the sparsification, we can only claim that phonemes from several speakers can also be partially identified by the *ICA* from the experiments in section. 3.2.4.

4. Testing the generality of the model cross different speakers is not successful in the experiment(section. 3.2.3.1). Only one of the four phonemes in the test set can be identified by a mixing matrix decomposed by ICA. By comparing the result in the error rate result(section. 3.4.3.2), we may conclude that our model obtained by *ICA* so far can only work well with two phonemes cross different speakers. The *Soft-Lost* is not successful in this experiment.

5. For the *COCA* on the *Invariant Cue* problem, the result is promising. The differences between phonemes are found larger than the speaker differences within phonemes. The result also are aligned with the perceptual similarities. The phonemes are from continuous speech. We think if we can deal with clearly pronounced phonemes. The result could be better.

6. We use the *Fisher Linear Discriminant Analysis* to analyze the speaker difference and phoneme difference. The kernel method to map the data to the feature space is not successful in increasing the ratio of the *separatability* between the phoneme and speaker.

7. In the two phoneme classification task, the *ICA* outperforms than the *Soft-Lost*. From this result, we can envision that higher order statistics and independence may resemble the cognition activity of the brain.

4.2 Future work

1. Better sparsification mechanism. When we work with phonemes from several speakers. We need a sparsification method which can remove the intrinsic noise equally.

2. ICA mixture method[14] could be used in an advanced analysis.

3. Better data base. A data base with clearly pronounced phonemes from different speakers could be more helpful in identifying the phonemes from different speakers.

4. New error rate result analysis method *(Confusion Matrix)* can be used for the unsupervised classification task.

Chapter 5

Appendix A

5.1 Confusion Matrix

[1]A *confusion matrix* (Kohavi and Provost, 1998) contains information about actual and predicted classifications made by a machine learning classifier. Performance of this classifier is commonly evaluated using the data in the matrix.It is another technique to evaluate the result of our unsupervised classification. The accuracy (AC) is the proportion of the total number of predictions that

Table 5.1: A typical the confusion matrix

		Predicted	
		Negative	Positive
actual	Negative	a	b
actual	Positive	c	d

were correct. It is determined using the equation [17]:

$$AC = \frac{a+d}{a+b+c+d} \tag{5.1}$$

[1]This section is based on the [17]

In our unsupervised classification, we have M source components,each source component is active according to one phoneme.we can make our confusion matrix :

Table 5.2: A Modified confusion matrix

	Phoneme1		Phoneme2	
$Component_{ith}$	Negative	Positive	Negative	Positive

Bibliography

[1] A. Hyvarinen, J. Karhunen, and E. Oja, Independent Component Analysis, John Wiley Sons, 2001.

[2] M.S. Lewicki,"Efficient coding of natural sounds," Nature Neuroscience, vol. 5, no. 4, pp. 356–363, 2002.

[3] Anthony J. Bell and Terrence J. Sejnowski, "The 'independent components of natural scenes are edge filters," Vision Research, vol. 37, no. 23, pp. 3327–3338, 1997.

[4] L. Feng and L. K. Hansen, "On low level cognitive components of speech" accepted in CIMCA'05 -International Conference on Computational Intelligence for Modelling, Nov 2005.

[5] L. K. Hansen, J. Larsen, and T. Kolenda, "On independent component analysis for multimedia signals," in Multimedia Image and Video Processing, pp. 175–199. CRC Press, Sep 2000

[6] T. Kolenda, L.K. Hansen, J. Larsen and O. Winther Independent Component Analysis for Understanding Multimedia Content in H. Bourlard, T. Adali, S. Bengio, J. Larsen, and S. Douglas (eds.) Proceedings of IEEE Workshop on Neural Networks for Signal Processing XII Matigny, Valais, Switzerland, Sept. 4–6, 2002, pp. 757–766.

[7] Logan, B. Mel Frequency Cepstral Coefficients for music modeling. Read at the first International Symposium on Music Information Retrieval..

[8] H.B. Nielsen,UCMINF - an Algorithm for Unconstrained, Nonlinear Optimization, IMM, Technical University of Denmark,IMM-TEC-0019,2001

[9] B. Scholkopf, A.J. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 10:1299–1319, 1998.

[10] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):955–974, 1998.

[11] Max Welling,Fisher Linear Discriminant Analysis.

[12] J. S. Garofolo et al., DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM, NIST, 1993.

[13] www.wikipedia.com

[14] Lee, T.-W., Lewicki, M. S., and Sejnowski, T. J. (1999c). ICA mixture models for unsupervised classification and automatic context switching. In International Workshop

[15] Paul D. O'Grady and Barak A. Pearlmutter.Soft-LOST: EM on a Mixture of Oriented Lines

[16] L. K. Hansen, P. Ahrendt, and J. Larsen, "Towards cognitive component analysis,"in AKRR'05 -International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning. Jun 2005, Pattern Recognition Society of Finland, Finnish Artificial Intelligence Society, Finnish Cognitive Linguistics Society

[17] www2.cs.uregina.ca