THESIS

Kernel Methods and Frequency Domain Independent Component Analysis for Robust Speaker Identification

Submitted by

Makoto Yamada
Department of Statistical Science

In partial fulfillment of the requirements
For the Degree of Doctor of Philosophy
The Graduate University for Advanced Studies
Hayama, Japan
Mar 2010

THE GRADUATE UNIVERSITY FOR ADVANCED STUDIES

Mar, 2010

WE HEREBY RECOMMEND THAT THE **THESIS** PROPOSAL PREPARED UNDER OUR SUPERVISION BY **MAKOTO YAMADA** ENTITLED **KERNEL METHODS AND FREQUENCY DOMAIN INDEPENDENT COMPONENT ANALYSIS FOR ROBUST SPEAKER IDENTIFICATION** BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

Prof. Kenji Fukumizu

Prof. Shiro Ikeda

Prof. Masashi Sugiyama

Prof. Tomoko Matsui
Adviser

Prof. Takashi Tsuchiya
Department Head/Director

ABSTRACT OF THESIS

Kernel Methods and Frequency Domain Independent Component Analysis for Robust Speaker Identification

The speaker identification is one of the key technologies for person identification in humanoid robots. Especially, when the face information is not available, the speaker identification is the only way to identify person, thus, to improve the speaker identification performance is an important issue for person identification tasks.

There are four major issues in speaker identification for humanoid robots in practice. First, the humanoid robots should identify the speaker in real-time with high identification rates. In these days, the kernel methods such as the support vector machine (SVM) and kernel logistic regression (KLR) are popular for speaker identification tasks, and the kernel based systems outperform the conventional Gaussian Mixture Model (GMM) based system. However, the kernel based speaker identification systems are usually computationally intensive, and this is of course not preferable for real-time implementation. To deal with the computational issue, we propose a method of approximating the sequence kernel that is shown to be computationally very efficient in Chapter 3. More specifically, we formulate the problem of approximating the sequence kernel as the problem of obtaining a *pre-image* in a reproducing kernel Hilbert space. The effectiveness of the proposed approximation is demonstrated in text-independent speaker identification experiments with 10 male speakers—our approach provides significant reduction in computation time while performance degradation is kept moderately. Based on the proposed method, we develop

a real-time kernel-based speaker identification system using the Virtual Studio Technology (VST).

Second, the speech features vary over time due to session dependent variation, the recording environment change, and physical conditions/emotions. However, conventional kernel based systems implicitly ignore these facts, and they just simply assume that the training and test input probability distributions of the training and test datasets are same at any time. To alleviate the influence of session dependent variation, it is popular to use several sessions of speaker utterance samples or to use *cepstral mean normalization* (CMN). However, gathering several sessions of speaker utterance data and assigning the speaker ID to the collected data are expensive both in time and cost and therefore not realistic in practice. Moreover, it is not possible to perfectly remove the session dependent variation by CMN alone. Thus, in Chapter 4, we propose a novel semi-supervised speaker identification method that can alleviate the influence of non-stationarity such as session dependent variation, the recording environment change, and physical conditions/emotions. We assume that the voice quality variants follow the *covariate shift* model, where only the voice feature distribution changes in the training and test phases. Our method consists of weighted versions of kernel logistic regression and cross validation and is theoretically shown to have the capability of alleviating the influence of covariate shift, where the weight (a.k.a importance) is estimated from the training and test distribution using the Kullback-Leibler Importance Estimation Procedure (KLIEP). We experimentally show through text-independent/dependent speaker identification simulations that the proposed method is promising in dealing with variations in voice quality.

Third, the humanoid robots are desired to automatically detect the unknown speaker and add the unknown speaker into the dictionary. Thus, the speaker detection task can be formulated as the outlier detection problem (i.e., outliers can be the unknown speakers). Since the outlier detection problem can be solved through the comparison between the log likelihoods of the unknown speaker and the speakers, the estimation accuracy of the log likelihoods is an important issue to improve the speaker detection performance. Thus, in Chapter 5, we propose a new importance (a.k.a likelihood) estimation method using Gaussian mixture models (GMMs) and principal component analyzers (PPCAs) mixture, where the proposed approach estimates the importance without going through the density estimation. An advantage of the proposed methods is that covariance matrices or projection matrices can also be learned through an expectation-maximization procedure, so the proposed method expected to work well when the true importance function has high correlation. Through experiments of outlier detection, we show the validity of the proposed approaches.

Forth, the humanoid robots move throughout the world, and the surrounding environment, source positions, and source mixtures are constantly changing. In addition, the speech overlaps are frequently occurred during conversation. Thus, the source separation techniques are useful for improving the speaker identification performance. To deal with those problems, in Chapter 6, we consider the problem of two-source signal separation from a two-microphone array, where a point source such as a speech signal is placed in front of a two-microphone array, while no information is available about another *interference* signal. We propose a simple and computationally efficient method for estimating the geometry and source type (a point or diffuse) of the interference signal, which allows us to adaptively choose a suitable unmixing matrix initialization scheme. Our proposed method, *noise adaptive optimization of matrix initialization* (NAOMI), is shown to be effective through source separation and speaker identification simulations.

Makoto Yamada
Department of Statistical Science
The Graduate Universities for Advanced Studies
Hayama, Kanagawa, Japan
Mar. 2010

ACKNOWLEDGEMENTS

I would like to express my gratitude to my adviser, Dr. Tomoko Matsui, for her helpful support. She has lead me to challenge the real-world problems and made me question things to allow me to obtain a deeper understanding. In addition, I fairly appreciate that she held the research meeting on weekends, because of my inconvenience. Her support has allowed me to be here and has forced me to use all my skills to solve problems.

I also would like to express my gratitude to Dr. Masashi Sugiyama for his supervision in Tokyo Institute of Technology. With his valuable assistance and heartwarming encouragement in spite of his extremely busy schedule, I could enjoy doing research and find many interesting research topics. In addition, I would like to thank Dr. Peter J. Brockwell for encouragement to obtain Ph.D. Without meeting him, I would never challenge to obtain a Ph.D.

I would like to thank my committee members, Dr. Kenji Fukumizu and Dr. Shiro Ikeda, for their patience and cooperation. I also would like to thank my co-authors, Dr. Ali Pezeshki, Dr. Mahmood Azimi-Sadjadi, for valuable discussions, brilliant ideas, and helpful support.

I must deeply thank Hitachi Corporation and Yamaha Corporation for the approval of my study in Matsui laboratory. I would like to express my gratitude to Mr. Furugaki, Mr. Shigeki Fujii, Mr. Hideki Kenmochi, managing director, Mr. Kazunobu Kondo, senior engineer. I am obliged to all the other people working for Yamaha as well. I also would like to thank my friends Juan Jesus Cabrera, Sungyoung

Kim, Gordon Wichern, Fernando V. Marquez, Vincent Wong, Yuu Takahashi, and Yoshikazu Yokotani for encouraging me to obtain Ph.D.

Last but not least, I would like to express my gratitude to my family. I am very grateful to my parents, uncle, and grandparents, for the education they gave me. Special thanks go to my wife, Noriko, for her support, encouragement, and patience during my Ph.D. period.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This dissertation is devoted to developing a useful speaker identification system for humanoid robots. In this chapter, we state the motivation and objective of our work.

1.1 Four issues in Speaker Identification for humanoid robots

The speaker identification is one of the key technologies for person identification in humanoid robots. Especially, when the face information is not available, the speaker identification is the only way to identify person. Therefore, to improve the speaker identification performance is an important issue.

There are four major issues in speaker identification for humanoid robots. First, the humanoid robots should identify the speaker in real-time with high identification rates. Second, since the speech features vary over time due to session dependent variation, the recording environment change, and physical conditions/emotions, the robust speaker identification system under the feature changes is required. Third, the humanoid robots should automatically add the speakers in dictionary, when the unknown speaker talks to it. Forth, since humanoid robots move throughout the world, the surrounding environment, source positions, and source mixtures are constantly changing.

To cope with these issues, we address the following topics in this dissertation:

1. Kernel based real-time speaker identification with acceleration of Mean Operator Sequence Kernel Computation

2. Semi-supervised Speaker Identification under Covariate Shift

3. Direct Importance Estimation with Gaussian Mixture Models and Probabilistic Principal Component Analyzers

4. Noise Adaptive Optimization of Matrix Initialization

In what follows, we present a brief introduction to each of these topics.

1.2 Kernel based real-time speaker identification with Acceleration of Mean Operator Sequence Kernel Computation

The humanoid robots should identify the speaker in real-time with high identification rates. In these days, the kernel methods such as the support vector machine (SVM) and kernel logistic regression (KLR) are popular for speaker identification tasks, and the kernel based system outperforms the conventional Gaussian Mixture Model (GMM) based system. However, the kernel based speaker identification system is originally expensive than GMM based speaker identification system, thus, the current version of kernel based speaker identification system is not suited for implementing on the humanoid robot. In addition, the kernel based speaker identification system usually uses the vectorial data, even though the sequential data is useful.

To cope with sequential speech data, a *sequence kernel* has been introduced for speaker identification [2], which utilizes a sequence of frame-level features for capturing long-term structure in phones, syllables, words, and the whole utterances. This sequence kernel is also called the Generalized Linear Discriminant Sequence (GLDS) kernel. While the GLDS kernel produced rather good performance in practice, it is not computationally efficient when the dimension of the feature space is very large;

this is because the GLDS kernel explicitly computes the projection of sequence samples in the feature space. Due to this explicit computation, the GLDS kernel only allows us to employ *finite*-dimensional feature spaces such as the polynomial reproducing kernel Hilbert space (RHKS); infinite-dimensional feature spaces such as the Gaussian RKHS are not allowed to use.

To overcome this limitation, mean operator sequence kernel was introduced [1]. The mean operator sequence kernel measures similarity between two sequences by *implicitly* computing the inner product between the means of the sequences in the feature space. Therefore, it can deal with infinite-dimensional feature spaces. The mean operator sequence kernel based speaker verification system was shown to significantly outperform other methods such as GMM and SVM with finite-dimensional kernels.

However, the mean operator sequence kernel still has a weakness. The mean operator sequence kernel is often computationally more efficient than the GLDS kernel, but the mean operator sequence kernel is still computationally intensive; it requires NN' vectorial kernel computations for measuring the similarity between sequential data of length N and N'.

The goal of this dissertation is to overcome this problem and develop a computationally efficient alternative to the original mean operator sequence kernel for real-time speaker identification system. Our basic idea is to approximate the mean operator sequence kernel using k-means algorithm. Then, we formulate the problem of approximating the sequence kernel as the problem of obtaining a *pre-image* in an RKHS [3], where pre-image $\widehat{\boldsymbol{x}}$ is the vector in input space which corresponds to the vector in feature space.

1.3 Semi-supervised Speaker Identification under Covariate Shift

In conventional methods, it is popular to assume that training and test speech data follow the same probability distribution. However, since the speech features vary over time due to session dependent variation, the recording environment change, and physical conditions/emotions, the training and test distributions are not necessarily the same in practice. In addition, the influence of the session dependent variation of voice quality in speaker identification problems has been investigated and the identification performance was shown to decrease significantly over 3 months—the major cause for the performance degradation was the voice source characteristic variations [4].

To alleviate the influence of session dependent variation, it is popular to use several sessions of speaker utterance samples [5, 6] or to use *cepstral mean normalization* (CMN) [7]. However, gathering several sessions of speaker utterance data and assigning the speaker ID to the collected data are expensive both in time and cost and therefore not realistic in practice. Moreover, it is not possible to perfectly remove the session dependent variation by CMN alone.

A practical setup for compensating the session dependent variation would be *semi-supervised learning*, where unlabeled samples are additionally given from the testing environment. In semi-supervised learning, it is required that the training and test distributions are related in some sense; otherwise we may not be able to learn anything about the test distribution from the training samples. A common modeling is called *covariate shift*, where the input distributions are different in the training and test phases but the conditional distribution of labels remains unchanged. In many real-world applications such as robot control [8, 9], bioinformatics [10, 11], spam filtering [12], natural language processing [13], brain-computer interfacing [14, 15], and econometrics [16], the covariate shift model has been shown to be useful. Covariate shift is also naturally induced in selective sampling or active learning scenarios [17, 18, 19, 20, 21]. For this reason, learning under covariate shift is receiving a great deal of attention these days in the machine learning community [22].

In this dissertation, we formulate the semi-supervised speaker identification problem in the covariate shift framework and propose a method that can cope with voice quality variants. Under covariate shift, standard maximum likelihood estimation is no longer consistent. The influence of covariate shift can be asymptotically canceled by weighting the log-likelihood terms according to the *importance*[23]:

$$w(\mathrm{X}) = \frac{p_{te}(\mathrm{X})}{p_{tr}(\mathrm{X})},$$

where $p_{te}(\mathrm{X})$ and $p_{tr}(\mathrm{X})$ are test and training input densities. We apply this weighting idea in KLR. The importance weight $w(\mathrm{X})$ is unknown in practice and needs to be estimated from data. For weight estimation, we utilize the Kullback-Leibler importance estimation procedure (KLIEP) since it is equipped with a built-in model selection procedure [24]. The (regularized) kernel logistic regression model contain two tuning parameters: the kernel width and the regularization parameter. Usually those tuning parameters are optimized based on cross validation (CV). However, CV is no longer unbiased due to covariate shift and therefore is not reliable as a model selection method. To cope with this problem, we use importance weighted CV [15] for unbiased model selection. The validity of our approach is experimentally shown through text-independent speaker identification simulations.

1.4 Direct Importance Estimation using Gaussian Mixture Models and Probabilistic Principal Component Analysis

Humanoid robots are desired to automatically add the unknown speakers into dictionary, and it can be formulated as the outlier detection problem (i.e., outlier can be the unknown speakers). Since the outlier detection problem can be solved via the log likelihood between the unknown speaker and the speakers in the dictionary, to improve the estimation accuracy of log likelihood is an important issue to for outlier detection problems.

Recently, the problem of estimating the ratio of two probability density functions (a.k.a. the *importance* or likelihood ratio) has received a great deal of attention since it can be used for various data processing purposes. *Covariate shift adaptation* would be a typical example [22]. Covariate shift is a situation in supervised learning where the training and test input distributions are different while the conditional distribution of output remains unchanged [23]. Another example in which the importance is useful is outlier detection [25]. The outlier detection task addressed in that paper is to identify irregular samples (i.e., outliers) in an evaluation dataset based on a model dataset that only contains regular samples (i.e., inliers). If the density ratio of two datasets is considered, the importance values for regular samples are close to one, while those for outliers tend to be significantly deviated from one. Thus the values of the importance could be used as an index of the degree of outlyingness. A similar idea can also be applied to change detection in time series [26].

A naive approach to approximating the importance function is to estimate training and test probability densities separately and then take the ratio of the estimated densities. However, density estimation itself is a difficult problem and taking the ratio of estimated densities can magnify the estimation error. In order to avoid density estimation, a semi-parametric approach called the *Kullback-Leibler Importance Estimation Procedure* (KLIEP) was proposed [27]. KLIEP does not involve density estimation but directly models the importance function. The parameters in the importance model is learned so that the Kullback-Leibler divergence from the true test distribution to the estimated test distribution is minimized without going through density estimation. KLIEP was shown to be useful in covariate shift adaptation [27] and outlier detection [25]. A typical implementation of KLIEP employs a spherical Gaussian kernel model and the Gaussian width is chosen by cross validation. This means that when the true importance function is correlated, the performance of KLIEP is expected to be degraded.

To cope with this problem, we propose to use a Gaussian mixture model (GMM) in the KLIEP algorithm and learn the covariance matrix of the Gaussian components at the same time. This will allow us to learn the importance function more adaptively even when the true importance function contains high correlation. We develop an expectation-maximization procedure for learning the parameters in the Gaussian mixture model. The effectiveness of the proposed method—which we call the Gaussian mixture KLIEP (GM-KLIEP)—is shown through experiments.

In addition, since we need to estimate the inverse of covariance matrices for GM-KLIEP, it tends to be unstable when the rank-deficient input vectors are observed. To deal with the rank deficient data, it is popular to use the dimensionality reduction method such as principal component analysis (PCA) as a pre-processing tool. Thus, in this dissertation, we propose the mixture of probabilistic PCA model based importance estimation, and we call the method as PPCA mixture KLIEP (PM-KLIEP).

1.5 Noise Adaptive Optimization of Matrix Initialization

In practice, the speech overlaps are frequently occurred during conversation, and it causes the serious degradation of speaker identification performance in humanoid robots. Thus, the source separation technique is preferred to use as the pre-processing of speaker identification system to improve the speaker identification performance. Therefore, in this dissertation, we propose the source separation method to improve the speaker identification performance in humanoid robot.

Implementing real-time frequency domain independent component analysis(FDICA) [28, 29, 30] has recently received much attention from the audio industry, c.f. [31]. This is due to the many potential source separation applications (e.g. speech enhancement, speaker separation), and the recent technological advancements that enable the implementation of FDICA on humanoid robots. However, since the humanoid robots move throughout the world, the surrounding environment, source positions, and source mixtures are constantly changing. Thus, it is quite difficult to implement FDICA in humanoid robots for real-world usage.

Many effective approaches have been proposed for improving FDICA performance by exploiting: knowledge regarding room and sensor geometry [32], geometric information of sound sources [33, 34], and a sophisticated prior model of speech [35]. However, these approaches implicitly assume knowledge of the sound source geometry, the source type (point source, diffuse source, etc.), and are valid only in a specific surrounding environmental condition. In addition, since the cost function of FDICA is *non-convex* in nature, FDICA is not guaranteed to converge to the optimal solution, when the initial unmixing matrix is incorrectly chosen. Thus, unmixing matrix initialization is a key factor for humanoid robot implementation of FDICA.

A popular unmixing matrix initialization technique is the combination of *delay-and-sum (DS)* and *null* beamformers (NBF) [30, 36], which are known to be robust to the well-known FDICA permutation problem [30]. However, beamformer-based initialization heavily depends on the sound source geometry and the source mixture type. Thus, beamformer-based initialization itself is not suited for mobile usage, without a reasonable estimator of the source geometry and the source types.

In this paper, we propose a Noise Adaptive Optimization of Matrix Initialization Algorithm (NAOMI). We assume a two source separation problem, where a point source, e.g., speech signal, is placed in front of a two microphone array, while a second *interfering* source should be separated and removed using FDICA. The interfering source is either another point source that is not located directly in front of the microphones (e.g., a speech signal that is not intended to be captured by the microphones) or a diffuse source (e.g., loud background music or airplane engine rumble). To estimate the type of interfering source, we first estimate its direction of arrival (DOA) at each frequency bin using *covariance fitting* [37], and then use the statistics of the estimated DOAs to classify the interfering source. The initial unmixing matrix is then selected based on the estimated source type. The effectiveness of the proposed method for speech de-noising is evaluated via a source separation simulations in anechoic and reverberant rooms.

1.6 Organization of this dissertation

This dissertation consists of seven chapters. In this section, we show the organization of this dissertation.

Chapter 2 formulates the speaker kernel based identification problem and review existing methods such as KLR and CV. Then, we propose kernel based real-time speaker identification with acceleration of mean operator sequence kernel computation in Chapter 3. Chapter 4 is devoted to the semi-supervised speaker identification under covariate shift framework for alleviating the session dependent variation. In Chapter 5, importance weighting techniques using GMM and PPCA are introduced. In Chapter 6, we introduce the noise adaptive optimization of matrix initialization for practical source separation problem. Finally, Chapter 7 contains the conclusions and a section about future work.

CHAPTER 2

KERNEL-BASED SPEAKER IDENTIFICATION

In this chapter, we formulate the kernel based speaker identification approach and its model selection.

2.1 Problem formulation

An utterance feature X pronounced by a speaker is expressed as a set of N mel-frequency cepstrum coefficient (MFCC) [38] vectors of d dimensions:

$$\mathrm{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{d \times N}. \tag{2.1}$$

For training, we are given n_{tr} labeled utterance samples:

$$\mathcal{Z}^{tr} = \{\mathrm{X}_i, y_i\}_{i=1}^{n_{tr}}, \tag{2.2}$$

where $y_i \in \{1, \ldots, K\}$ denotes the index of the speaker who pronounced X_i. The goal of speaker identification is to predict the speaker index of a test utterance sample X based on the training samples. We predict the speaker index c of the test sample X following the Bayes decision rule:

$$P(y = c|\mathrm{X}) > P(y = i|\mathrm{X}) \quad \forall\, i \neq c. \tag{2.3}$$

For approximating the class-posterior probability, we use the following parametric model $p(y = c|\mathrm{X}, \mathrm{V})$:

$$p(y = c|\mathrm{X}, \mathrm{V}) = \frac{\exp f_{\boldsymbol{v}_c}(\mathrm{X})}{\sum_{l=1}^{K} \exp f_{\boldsymbol{v}_l}(\mathrm{X})}, \tag{2.4}$$

where $\mathrm{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K]^\top \in \mathbb{R}^{K \times n_{tr}}$ is the parameter, $^\top$ denotes the transpose, and $f_{\boldsymbol{v}_l}$ is a discriminant function corresponding to the speaker l. This model is known as the softmax function and widely used in multiclass logistic regression. We use the following kernel regression model as the discriminant function $f_{\boldsymbol{v}_l}$ [6]:

$$f_{\boldsymbol{v}_l}(\mathrm{X}) = \sum_{i=1}^{n_{tr}} v_{l,i} \mathcal{K}(\mathrm{X}, \mathrm{X}_i) \quad l = 1, \ldots, K, \tag{2.5}$$

where $\boldsymbol{v}_l = (v_{l,1}, \ldots, v_{l,n_{tr}})^\top \in \mathbb{R}^{n_{tr}}$ are parameters corresponding the speaker l and $\mathcal{K}(\mathrm{X}, \mathrm{X}')$ is a kernel function.

2.2 Feature Extraction

In speaker identification, it is common to extract a set of features from each speech signal, and we use the extracted feature for classification instead of the speech signals themselves. A good set of features should include discriminative information, and the feature set should be small enough to allow fast processing and robust.

A speech signal can be assumed as a *stationary* stochastic process within small time intervals (20-30ms). From this fact, the major discriminative information between speech signals appear in the frequency domain, and we usually use the sequence of short time spectral feature vectors which extracted from the speech signal. Figure 2.1 illustrates the extraction of a feature vector $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ from a speech signal. A window function of fixed width such as Hamming window is used to extract a short-time segment of the speech signal in order to convert to the spectral feature vector. Then, the window function is shifted with 5-10 ms to the right for further extraction of feature vectors until the end of the speech signal is reached. Note that, since different speech signals have different durations, feature extraction with a fixed window shift leads to time series with different number of vectors.

Mel-frequency cepstral coefficients (MFCC) [38] are both popular feature extraction methods. Often, the energy of the windowed speech signal is appended to the MFCC feature vectors. The total dimension of these feature vectors is usually 13.

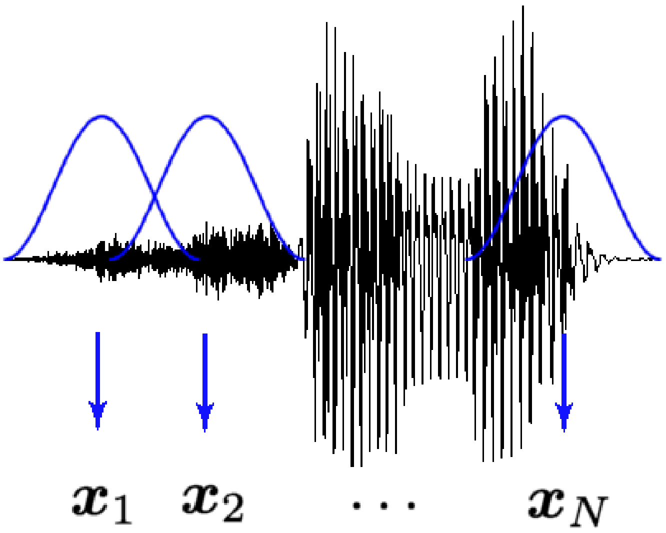


Figure 2.1: Feature extraction from a speech signal.

We also use the *delta* and $acceleration coefficients$ of the feature vectors for improving recognition performance. Delta coefficients (Δ MFCC) are the first order time derivatives of a feature vector sequence, and contain information on the rate of change of the vectors in the sequence. Similarly, acceleration coefficients (ΔΔ MFCC) are the approximations to the second order time derivatives, and contain information on the rate of the rate of change. We usually concatenate the MFCC coefficients, Δ MFCC, and ΔΔ MFCC, and we use the vector for speaker identification.

2.3 Kernel Logistic Regression

Logistic regression is a one of the popular statistical method for estimating the conditional probability distribution $p(y|\mathrm{X})$ of a class label $y \in \mathcal{Y}$ given an observation $\mathrm{X} \in \mathcal{X}$. Classification is accomplished by selecting the class label $\hat{y}$ given the largest conditional probability:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} p(y|\mathrm{X}). \tag{2.6}$$

For approximating the class-posterior probability, we use the following parametric model $p(y = c|\mathrm{X}, \mathrm{V})$:

$$p(y = c|\mathrm{X}, \mathrm{V}) = \frac{\exp f_{\boldsymbol{v}_c}(\mathrm{X})}{\sum_{l=1}^{K} \exp f_{\boldsymbol{v}_l}(\mathrm{X})}, \tag{2.7}$$

where $\mathrm{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K]^\top \in \mathbb{R}^{K \times n_{tr}}$ is the parameter, $^\top$ denotes the transpose, and $f_{\boldsymbol{v}_l}$ is a discriminant function corresponding to the class $y = l$. This function is known as *$softmax$* function. *Kernel logistic regression* (KLR) is a kernelized variant of *logistic regression*. In KLR, we map the input vector to a high-dimensional space (feature space) and solve the logistic regression problem in the feature space; the similarity in feature space can be implicitly computed via the *kernel trick*. The kernel trick allows one to non-linearize a linear algorithm without sacrificing computational simplicity of the linear algorithm. Below, we briefly review KLR following the papers [39, 40].

We employ maximum likelihood estimation for learning the parameter V. The negative log-likelihood function $\mathcal{P}_\delta^{\log}(\mathrm{V}; \mathcal{Z}^{tr})$ for the kernel logistic regression model is given by

$$\mathcal{P}_\delta^{\log}(\mathrm{V}; \mathcal{Z}^{tr}) = -\sum_{i=1}^{n_{tr}} \log P(y_i|\mathrm{X}_i, \mathrm{V}) + \frac{\delta}{2}\mathrm{trace}(\mathrm{V}\mathrm{K}\mathrm{V}^\top), \tag{2.8}$$

where $\mathrm{trace}(\mathrm{VKV}^\top)$ is a regularizer to avoid overfitting, δ is the regularization parameter that controls strength of regularization, and $\mathrm{K} = [\mathcal{K}(\mathrm{X}_i, \mathrm{X}_j)]_{i,j=1}^{n_{tr}}$ is the kernel Gram matrix. The negative log-likelihood function is convex and the unique minimizer can be obtained by, e.g., the Newton method. In the Newton method, the parameter matrix V is updated iteratively as

$$\mathrm{V} \leftarrow \mathrm{V} - \epsilon \Delta \mathrm{V}, \tag{2.9}$$

where ϵ is the step size and $\Delta\mathrm{V}$ is defined as

$$\mathrm{vec}\Delta\mathrm{V} = [\nabla^2 \mathcal{P}_\delta^{\log}(\mathrm{V}; \mathcal{Z})]^{-1} \mathrm{vec}\nabla\mathcal{P}_\delta^{\log}(\mathrm{V}; \mathcal{Z}). \tag{2.10}$$

'vec' denotes the vectorization operator, $\nabla\mathcal{P}_\delta^{\log}(\mathrm{V}; \mathcal{Z})$ is the gradient of Eq.(2.8) with respect to V, and $\nabla^2\mathcal{P}_\delta^{\log}(\mathrm{V}; \mathcal{Z})$ is the Hessian of Eq.(2.8) with respect to V. The gradient and Hessian are given as

$$\begin{aligned}
\nabla \mathcal{P}_{\delta}^{\log}(\mathrm{V};\mathcal{Z}) &= (\mathrm{P}(\mathrm{V}) - \mathrm{Y} + \delta \mathrm{V})\mathrm{K}, & (2.11)\\
\nabla^2 \mathcal{P}_{\delta}^{\log}(\mathrm{V};\mathcal{Z}) &= \sum_{i=1}^{n_{tr}} (\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top) \otimes \boldsymbol{k}(\mathrm{X}_i)\boldsymbol{k}(\mathrm{X}_i)^\top \\
&\quad +(\mathrm{K}^\top \otimes \mathrm{I}), & (2.12)
\end{aligned}$$

where

$$\mathrm{P}(\mathrm{V}) = [\boldsymbol{p}(\mathrm{X}_1), \ldots, \boldsymbol{p}(\mathrm{X}_{n_{tr}})] \in \mathbb{R}^{K \times n_{tr}} \tag{2.13}$$

is a matrix whose n-th column is a vector of the class-posterior probabilities $\boldsymbol{p}(\mathrm{X}_n)$,

$$\boldsymbol{p}(\mathrm{X}) = [p(y=1|\mathrm{X},\mathrm{V}), \ldots, p(y=K|\mathrm{X},\mathrm{V})]^\top \in \mathbb{R}^K \tag{2.14}$$

denotes the class-posterior probabilities for all classes given X,

$$\mathrm{Y} = [\boldsymbol{e}_{y^1}, \ldots, \boldsymbol{e}_{y^N}] \in \mathbb{R}^{K \times n_{tr}}, \tag{2.15}$$

whose n-th column $\boldsymbol{e}_{y^n}$ is a unit vector with all zeros except for element y^n being 1, $\mathrm{diag}(a, \ldots, b)$ denotes the diagonal matrix with diagonal elements $a, \ldots, b$,

$$\boldsymbol{k}(\mathrm{X}) = [\mathcal{K}(\mathrm{X},\mathrm{X}_1), \ldots, \mathcal{K}(\mathrm{X},\mathrm{X}_{n_{tr}})]^\top \in \mathbb{R}^{n_{tr}} \tag{2.16}$$

is a vector whose elements are given by the mean operator sequence kernel, $\otimes$ denotes the Kronecker product, and I denotes the identity matrix.

In order to estimate the update matrix $\Delta\mathrm{V}$, the inverse of the Hessian needs to be computed at every iteration. This is computationally expensive so we approximate $\Delta\mathrm{V}$ by the conjugate gradient method; an approximation $\widehat{\Delta\mathrm{V}}$ can be estimated by solving the following linear equation [39, 40]:

$$\nabla^2 \mathcal{P}_{\delta}^{\log}(\mathrm{V};\mathcal{Z}) \mathrm{vec}\widehat{\Delta\mathrm{V}} = \mathrm{vec}\nabla \mathcal{P}_{\delta}^{\log}(\mathrm{V};\mathcal{Z}). \tag{2.17}$$

Substituting Eqs.(2.11) and (2.12) into Eq.(2.17) and using the transformation

$$\mathrm{vec}(\mathrm{ABC}) = (\mathrm{C}^\top \otimes \mathrm{A})\mathrm{vec}(\mathrm{B}), \tag{2.18}$$

we have

$$\sum_{i=1}^{n_{tr}}(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top)\widehat{\Delta \mathrm{V}}\boldsymbol{k}(\mathrm{X}_i)\boldsymbol{k}(\mathrm{X}_i)^\top = (\mathrm{P}(\mathrm{V}) - \mathrm{Y} + \delta\mathrm{V})\mathrm{K}. \tag{2.19}$$

1. Initialize: Start with an initial matrix $\Delta\mathrm{V}_0^i$ and compute the matrices R_0 and Q_0:

$$\begin{aligned} \mathrm{R}_0 &= \mathrm{P}(\mathrm{V}^i) - \mathrm{Y} + \delta\mathrm{V} \\ &- \sum_{j=1}^{N} \boldsymbol{e}_j\boldsymbol{k}(\mathrm{X}_i)\Delta\mathrm{V}_0^i(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top) - \delta\Delta\mathrm{V}_0^i, \qquad (2.20) \\ \mathrm{Q}_0 &= \boldsymbol{k}(\mathrm{X}_i)\boldsymbol{e}_j^\top\mathrm{R}_0(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top) - \mathrm{R}_0. \qquad (2.21) \end{aligned}$$

2. Iterate: Generate a sequence $(\Delta\mathrm{V}_1^i, \Delta\mathrm{V}_2^i, \ldots)$ according to

$$\begin{aligned} \alpha_k &= \frac{\boldsymbol{k}(\mathrm{X}_i)\boldsymbol{e}_j^\top\mathrm{R}_0(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top) + \mathrm{R}_k}{\boldsymbol{e}_j\boldsymbol{k}(\mathrm{X}_i)^\top\mathrm{Q}_k(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top) + \mathrm{Q}_k}, \qquad (2.22) \\ \Delta\mathrm{V}_{k+1}^i &= \Delta_k^i\mathrm{V}_k^i + \alpha_k\mathrm{R}_k, \qquad (2.23) \\ \mathrm{R}_{k+1} &= \mathrm{P}(\mathrm{V}^i) - \mathrm{Y} + \delta\mathrm{V} \\ &- \sum_{j=1}^{N} \boldsymbol{e}_k\boldsymbol{k}(\mathrm{X}_i)\Delta\mathrm{V}_{k+1}^i(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top) - \delta\Delta\mathrm{V}_{k+1}^i, \quad (2.24) \\ \beta_k &= \frac{\boldsymbol{k}(\mathrm{X}_i)\boldsymbol{e}_j^\top\mathrm{R}_{k+1}(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top) + \mathrm{R}_{k+1}}{\boldsymbol{k}(\mathrm{X}_i)\boldsymbol{e}_j^\top\mathrm{R}_k(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top) + \mathrm{R}_k}, \qquad (2.25) \\ \Delta\mathrm{V}_{k+1}^i &= \Delta_k^i\mathrm{V}_k^i + \alpha_k\mathrm{R}_k, \qquad (2.26) \\ \mathrm{Q}_{k+1} &= \boldsymbol{k}(\mathrm{X}_i)\boldsymbol{e}_j^\top\mathrm{R}_{k+1}(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top) - \mathrm{R}_{k+1}. \qquad (2.27) \end{aligned}$$

2.4 Mean Operator Sequence Kernel

The performance of KLR depends on the choice of the kernel function. A popular choice for speaker identification is the *mean operator sequence kernel*, which is defined as follows [1]:

$$\begin{aligned} \mathcal{K}(\mathrm{X}, \mathrm{X}') &= \frac{1}{N}\sum_{p=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_p)^\top\frac{1}{N'}\sum_{p'=1}^{N'}\boldsymbol{\phi}(\boldsymbol{x}'_{p'}), \\ &= \frac{1}{NN'}\sum_{p=1}^{N}\sum_{p'=1}^{N'}k(\boldsymbol{x}_p, \boldsymbol{x}'_{p'}) \end{aligned}$$

where

$$\mathrm{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{d \times N},$$
$$\mathrm{X}' = [\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{N'}] \in \mathbb{R}^{d \times N'},$$

are sequences of d-dimensional features of length N and N' and

$$k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}')$$

is a 'base' vectorial kernel function.

In this dissertation, we use the Gaussian kernel for 'base' vectorial kernel function:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right). \tag{2.28}$$

2.5 Model selection in Kernel Logistic Regression

The above KLR method includes two tuning parameters: the Gaussian width σ and the regularization parameter δ. KLR is shown to be *consistent*, i.e., the learned parameter converges to the optimal value as the number of training samples tends to be infinity:

$$\lim_{N\to\infty} \widehat{\mathrm{V}} = \mathrm{V}^*,$$

where $\widehat{\mathrm{V}}$ is the parameter learned by KLR and V^* is the optimal parameter that minimizes the expected prediction error for test samples:

$$\mathrm{V}^* = \underset{\mathrm{V}}{\mathrm{argmin}} \iint I(y = \widehat{y}(\mathrm{X} \,|\, \mathrm{V}))p(y \,|\, \mathrm{X})p(\mathrm{X})\mathrm{d}y\mathrm{dX}.$$

$\widehat{y}(\mathrm{X} \,|\, \mathrm{V})$ is an estimate of speaker of an utterance feature X for parameter V. Also, when $p(\mathrm{X})$ and $p(y \,|\, \mathrm{X})$ are common in the training and test phases, cross-validation (CV) is (almost) *unbiased* [3]:

$$\mathrm{E}_{\mathcal{Z}^{tr}}\left[\widehat{R}_{CV}^{\mathcal{Z}^{tr}} - R^{\mathcal{Z}^{tr}}\right] \approx 0,$$

where $\mathrm{E}_{\mathcal{Z}^{tr}}$ is the expectation over the training set $\mathcal{Z}^{tr}$ and $R^{\mathcal{Z}^{tr}}$ is the expected prediction error defined by

$$R^{\mathcal{Z}^{tr}} = \iint I(y = \widehat{y}(\mathrm{X}; \mathcal{Z}^{tr}))p(y \,|\, \mathrm{X})p(\mathrm{X})\mathrm{d}y\mathrm{d}\mathrm{X}.$$

$\widehat{y}(\mathrm{X}; \mathcal{Z}^{tr})$ is a learned function from the training set $\mathcal{Z}^{tr}$.

One of the popular approaches to model selection is k-fold cross validation (kCV). Let us divide the training set $\mathcal{Z}^{tr} = \{(\mathrm{X}_i, y_i)\}_{i=1}^{n_{tr}}$ into k disjoint non-empty subsets $\{\mathcal{Z}_i^{tr}\}_{i=1}^{k}$. Let $\widehat{y}_{\mathcal{Z}_j^{tr}}(\mathrm{X})$ be an estimate of a speaker of a test utterance sample X obtained from $\{\mathcal{Z}_i^{tr}\}_{i \neq j}$ (i.e., without $\mathcal{Z}_j^{tr}$). Then the score is given by

$$\widehat{R}_{kCV}^{\mathcal{Z}^{tr}} = \frac{1}{k}\sum_{j=1}^{k}\frac{1}{|\mathcal{Z}_j^{tr}|}\sum_{(\mathrm{X},y)\in\mathcal{Z}_j^{tr}} I(y = \widehat{y}_{\mathcal{Z}_j^{tr}}(\mathrm{X})), \tag{2.29}$$

where $|\mathcal{Z}_j^{tr}|$ is the number of samples in the subset $\mathcal{Z}_j^{tr}$ and $I(\cdot)$ denotes the indicator function.

CHAPTER 3

KERNEL BASED REAL-TIME SPEAKER IDENTIFICATION WITH ACCELERATING SEQUENCE KERNEL COMPUTATION

This chapter is devoted to developing a kernel based real-time speaker identification.

3.1 Introduction

Kernel methods such as the support vector machine (SVM) [41] and kernel logistic regression (KLR) [42] are successful approaches in speaker identification, given that the kernel functions are designed appropriately. Recently, a *mean operator sequence kernel* (MOSK) has been introduced for speaker identification [1], which utilizes a sequence of frame-level features for capturing long-term structure in phones, syllables, words, and entire utterances. MOSK measures the similarity between two sequences by computing the inner product between the means of the sequences *implicitly* in the feature space. The MOSK based speaker verification system was shown to significantly outperform other methods such as the Gaussian mixture model (GMM) and the SVM with finite-dimensional kernels.

Although MOSK performs well in the speaker verification task, its computational complexity limits its use in applications where real time processing is required. Specifically, MOSK requires NN' vector kernel computations for measuring the similarity between two data sequences of length N and N', respectively. The goal of this paper is to develop a computationally efficient alternative to the MOSK for real time speaker identification. The first step in our approach is to approximate the MOSK using k-means clustering. Then, we formulate the problem of approximating the sequence kernel as the problem of obtaining a *pre-image* in a reproducing kernel Hilbert space (RKHS) [41]. A pre-image is a vector in the input space mapped to the target feature vector in the RKHS.

The practical effectiveness of the proposed method is investigated in text-independent speaker identification experiments with 10 male speakers. Results demonstrate that the proposed method provides significant reduction in computation time while speaker identification accuracy is only moderately degraded. Furthermore, using the pre-image approximation we develop a real-time speaker identification system using Virtual Studio Technology (VST).

3.2 Problem Formulation

In this section, we formulate the speaker identification problem based on the kernel logistic regression (KLR) model.

3.2.1 Kernel-based Text-independent Speaker Identification

An utterance sample X pronounced by a speaker is expressed as a set of N *mel-frequency cepstrum coefficient (MFCC)* [38] vectors of dimension d:

$$\mathrm{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{d \times N}.$$

For training, we are given n labeled utterance samples:

$$\mathcal{Z} = \{(\mathrm{X}_i, y_i)\}_{i=1}^{n},$$

where $y_i \in \{1, \ldots, K\}$ denotes the index of the speaker who pronounced X_i. The goal of speaker identification is to predict the speaker index of a test utterance sample X based on the training samples. We predict the speaker index c of the test sample X following *Bayes decision rule*:

$$\max_{c} p(y = c \,|\, \mathrm{X}).$$

For approximating the class-posterior probability, we use

$$p(y = c \,|\, \mathrm{X}; \mathrm{V}) = \frac{\exp f_{\boldsymbol{v}_c}(\mathrm{X})}{\sum_{l=1}^{K} \exp f_{\boldsymbol{v}_l}(\mathrm{X})},$$

where $\mathrm{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K]^\top \in \mathbb{R}^{K \times n}$ is the parameter, $^\top$ denotes the transpose, and $f_{\boldsymbol{v}_l}$ is a discriminant function corresponding to speaker l. This form is known as the *softmax* function and widely used in multiclass logistic regression. We use the following kernel regression model as the discriminant function $f_{\boldsymbol{v}_l}$:

$$f_{\boldsymbol{v}_l}(\mathrm{X}) = \sum_{i=1}^{n} v_{l,i} \mathcal{K}(\mathrm{X}, \mathrm{X}_i) \quad l = 1, \ldots, K,$$

where $\boldsymbol{v}_l = (v_{l,1}, \ldots, v_{l,n})^\top \in \mathbb{R}^n$ are parameters corresponding to speaker l and $\mathcal{K}(\mathrm{X}, \mathrm{X}')$ is a kernel function.

We employ maximum likelihood estimation for learning the parameter V. The negative log-likelihood function $\mathcal{P}^{\log}(\mathrm{V}; \mathcal{Z})$ for the kernel logistic regression model is given by

$$\mathcal{P}^{\log}(\mathrm{V}; \mathcal{Z}) = -\sum_{i=1}^{n} \log P(y_i \,|\, \mathrm{X}_i; \mathrm{V}),$$

where $\mathrm{K} = [\mathcal{K}(\mathrm{X}_i, \mathrm{X}_j)]_{i,j=1}^{n}$ is the kernel Gram matrix. $\mathcal{P}^{\log}(\mathrm{V}; \mathcal{Z})$ is a convex function with respect to V and therefore its unique minimizer can be obtained using, e.g., the Newton method [39].

3.2.2 Mean Operator Sequence Kernel [1]

The performance of KLR depends on the choice of the kernel function. In this chapter, we use the *mean operator sequence kernel* (MOSK) [1] as the kernel function since it allows us to handle feature sequences of different length. For sequences of d-dimensional feature vectors of length N and N',

$$\mathrm{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{d \times N},$$
$$\mathrm{X}' = [\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{N'}] \in \mathbb{R}^{d \times N'},$$

MOSK is defined as

$$\begin{aligned} \mathcal{K}(\mathrm{X}, \mathrm{X}') &= \frac{1}{N} \sum_{p=1}^{N} \boldsymbol{\phi}(\boldsymbol{x}_p)^\top \frac{1}{N'} \sum_{p'=1}^{N'} \boldsymbol{\phi}(\boldsymbol{x}'_{p'}), \\ &= \frac{1}{NN'} \sum_{p=1}^{N} \sum_{p'=1}^{N'} k(\boldsymbol{x}_p, \boldsymbol{x}'_{p'}), \end{aligned}$$

where

$$k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}')$$

is a ‘base’ vector kernel function.

MOSK requires NN' vector kernel computations for calculating the similarity between utterances X and X′. Therefore, the MOSK computation is not suited for real-time application when NN' is very large.

3.3 Approximation of MOSK

In this section, we provide an approximation method of the MOSK computation. Below, we focus on the Gaussian kernel as the base kernel function:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right).$$

3.3.1 Approximating Mean Operator Sequence Kernel by Parts

For $D \ll N$, let us divide the samples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ into D clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_D\}$ such that

$$\begin{aligned} \mathcal{C}_i \cap \mathcal{C}_j &= \emptyset \quad \text{for } i \neq j, \\ \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_D &= \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}. \end{aligned}$$

We may use the k-means clustering algorithm for this purpose. Then, $\frac{1}{N}\sum_{p=1}^{N}\phi(\boldsymbol{x}_p)$ can be expressed as

$$\frac{1}{N}\sum_{p=1}^{N}\phi(\boldsymbol{x}_p) = \frac{1}{N}\left\{\sum_{\boldsymbol{x}\in\mathcal{C}_1}\phi(\boldsymbol{x}) + \cdots + \sum_{\boldsymbol{x}\in\mathcal{C}_D}\phi(\boldsymbol{x})\right\}.$$
$$= \frac{\pi_1}{N_1}\sum_{\boldsymbol{x}\in\mathcal{C}_1}\phi(\boldsymbol{x}) + \cdots + \frac{\pi_D}{N_D}\sum_{\boldsymbol{x}\in\mathcal{C}_D}\phi(\boldsymbol{x}), \qquad (3.1)$$

where N_i is the number of samples in cluster $\mathcal{C}_i$ and $\pi_i = N_i/N$.

If we can approximate the mean $\frac{1}{N_i}\sum_{\boldsymbol{x}\in\mathcal{C}_i}\phi(\boldsymbol{x})$ by a single point $\phi(\boldsymbol{m}_i)$, the computational cost of the mean in the feature space is reduced from $\mathcal{O}(N)$ to $\mathcal{O}(D)$. To obtain a good approximation point $\boldsymbol{m}_i$, we minimize the following criterion:

$$J_i(\boldsymbol{m}_i) = \|\phi(\boldsymbol{m}_i) - \frac{1}{N_i}\sum_{\boldsymbol{x}\in\mathcal{C}_i}\phi(\boldsymbol{x})\|^2.$$

This is often called the *pre-image* problem in the context of kernel methods [41]. For the Gaussian kernel, the above criterion can be written as

$$J_i(\boldsymbol{m}_i) = 1 - \frac{2}{N_i}\sum_{\boldsymbol{x}\in\mathcal{C}_i}k(\boldsymbol{m}_i,\boldsymbol{x}) + \frac{1}{N_i^2}\sum_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{C}_i}k(\boldsymbol{x},\boldsymbol{x}'), \qquad (3.2)$$

where we used

$$k(\boldsymbol{m}_i,\boldsymbol{m}_i) = \exp\left(-\frac{\|\boldsymbol{m}_i-\boldsymbol{m}_i\|^2}{2\sigma^2}\right) = 1.$$

Taking the derivative of Eq.(3.2) with respect to $\boldsymbol{m}$, we have

$$\frac{\partial J_i(\boldsymbol{m}_i)}{\partial \boldsymbol{m}_i} = \frac{\partial}{\partial \boldsymbol{m}_i}\left[-\frac{2}{N_i}\sum_{\boldsymbol{x}\in\mathcal{C}_i}\exp\left(-\frac{\|\boldsymbol{m}_i-\boldsymbol{x}\|^2}{2\sigma^2}\right)\right]$$
$$= \frac{1}{\sigma^2 N_i}\sum_{\boldsymbol{x}\in\mathcal{C}_i}\exp\left(-\frac{\|\boldsymbol{m}_i-\boldsymbol{x}\|^2}{2\sigma^2}\right)(\boldsymbol{m}_i-\boldsymbol{x}). \qquad (3.3)$$

Equating Eq.(3.3) to zero, we have

$$\widehat{\boldsymbol{m}}_i = \frac{\sum_{\boldsymbol{x}\in\mathcal{C}_i}\exp\left(-\frac{\|\boldsymbol{m}_i-\boldsymbol{x}\|^2}{2\sigma^2}\right)\boldsymbol{x}}{\sum_{\boldsymbol{x}'\in\mathcal{C}_i}\exp\left(-\frac{\|\boldsymbol{m}_i-\boldsymbol{x}'\|^2}{2\sigma^2}\right)}. \qquad (3.4)$$

We use Eq.(3.4) as a re-estimation formula, i.e., $\widehat{\boldsymbol{m}}_i$ is updated by Eq.(3.4) with $\boldsymbol{m}_i$ in the right-hand side replaced by the current estimate $\widehat{\boldsymbol{m}}_i$ and this is repeated until convergence.

Then Eq.(3.1) yields

$$\frac{1}{N}\sum_{p=1}^{N}\phi(\boldsymbol{x}_p) \approx \sum_{i=1}^{D}\pi_i\phi(\widehat{\boldsymbol{m}}_i). \tag{3.5}$$

Based on Eq.(3.5), MOSK can be approximated by

$$\begin{aligned}\mathcal{K}(\mathrm{X},\mathrm{X}') &\approx \sum_{i=1}^{D}\pi_i\phi(\widehat{\boldsymbol{m}}_i)^\top\sum_{i'=1}^{D'}\pi'_{i'}\phi(\widehat{\boldsymbol{m}}_{i'}) \\ &= \sum_{i=1}^{D}\sum_{i'=1}^{D'}\pi_i\pi'_{i'}k(\widehat{\boldsymbol{m}}_i,\widehat{\boldsymbol{m}}_{i'}). \qquad (3.6)\end{aligned}$$

Following the k-means clustering algorithm, we call the proposed method the *k-means operator sequence kernel* (k-MOSK). The number of vectorial kernel computations in the original MOSK is NN', while that in k-MOSK is DD'. Thus k-MOSK would be computationally much more efficient than MOSK given that D and D' are much smaller than N and N'. It is clear that k-MOSK satisfies positive definiteness; thus it is a valid kernel function.

The computation of the k-means clustering algorithm for every utterance in the test phase is expensive. So we compute the kernel between a training sample X and a test sample $\mathrm{X}' = \{x'_1, \ldots, x'_{N'}\}$ as

$$\mathcal{K}(\mathrm{X},\mathrm{X}') = \frac{1}{N'}\sum_{i=1}^{D}\sum_{p=1}^{N'}\pi_i k(\widehat{\boldsymbol{m}}_i,\boldsymbol{x}'_p). \tag{3.7}$$

3.4 Experiments

In this section, we compare the performance of MOSK and k-MOSK with different numbers of clusters D in a speaker identification task.

3.4.1 System and Data Acquisition

The data for training and testing were collected from 10 male speakers, where each speaker uttered several different words as listed in Table 3.1.

The duration of an utterance for each training sentence was approximately four seconds. Thus, the total duration of utterances over three training sentences was

Table 3.1: Training sentences and test words (in Japanese, written using the Hepburn system of Romanization).

	Contents
Training sentences:	1. seno takasawa hyakunanajusseNchi hodode mega ookiku yaya futotteiru 2. oogoeo dashisugite kasuregoeni natte shimau 3. tashizaN hikizaNwa dekinakutemo eha kakeru
Testing words:	1. mouichido 2. torikaeshi 3. teisei 4. horyuu 5. shoukai

approximately 12 seconds per speaker. For testing purposes, we use utterances of 5 words recorded in three sessions over six months with no time overlap to the training session. Thus the total number of test words was 150 (10 speakers × 5 words × 3 sessions).

A feature vector of 26 dimensions, consisting of 12 MFCCs, normalized log energy, and their first derivatives, is derived once every 10ms over a 25.6ms Hamming-windowed speech segment. We divide each training utterance into 300ms disjoint segments, each of which corresponds to a set of features of size 26×30. On the other hand, for testing, we use the whole utterance of each word consisting of approximately 1000ms duration for computing MOSK and k-MOSK since each word is treated as a single test sample.

3.4.2 Results

We evaluate the proposed k-MOSK with the several different numbers of clusters D. The Gaussian width σ in the base Gaussian kernel is chosen from

$$\{8, 10, 12, 14, 16\}$$

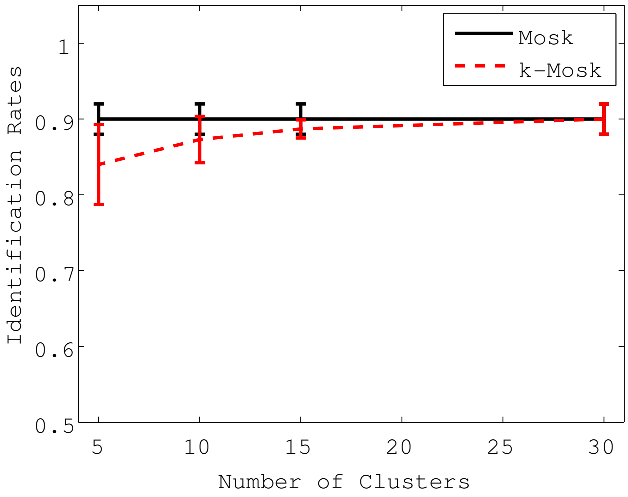


Figure 3.1: Speaker identification rates obtained using 30, 15, 10, and 5 clusters, with selected kernel widths of 12, 14, 14, and 16, respectively.

by 10-fold *cross-validation* (CV). In our preliminary experiments, we observed that the 10-fold CV scores tend to be heavily affected by the random split of the training samples. We conjecture that this is due to non-i.i.d. nature of the MFCC features, which is different from the theoretical assumptions of CV. In order to obtain reliable experimental results, we repeat the CV procedure 50 times with different random data splits and use the average score for model selection.

Figure 3.1 depicts the speaker identification rates for the test words using MOSK and k-MOSK with different numbers of clusters D. In Figure 3.2, we plot the computation time of MOSK and k-MOSK in training and testing using a standard personal computer with a Quad Core 2.0GHz processor and 2GB memory. The computation time for MOSK is normalized to one. These results demonstrate that k-MOSK is computationally more efficient than the original MOSK with mild degradation in identification accuracy.

Based on k-MOSK, we have developed a real-time kernel-based speaker identification system using a Virtual Studio Technology (VST) plugin (see Figure 3.3). A demo movie is available at *http://dsp.syuriken.jp/demo/sid.html*.

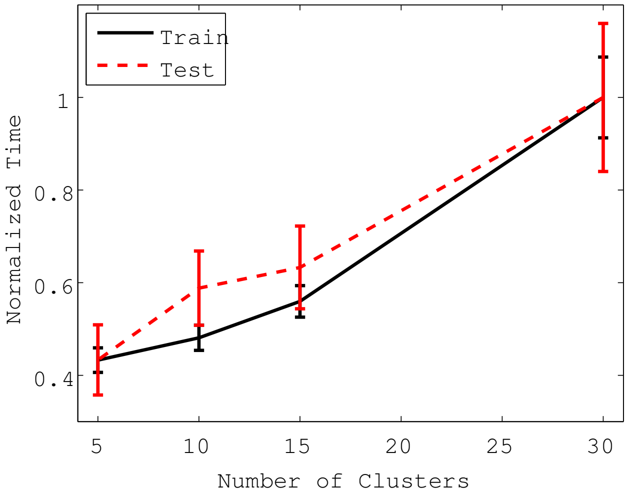


Figure 3.2: The normalized computation time of MOSK and k-MOSK in training and testing using a standard personal computer with Quad Core 2.0GHz processor and 2GB memory.

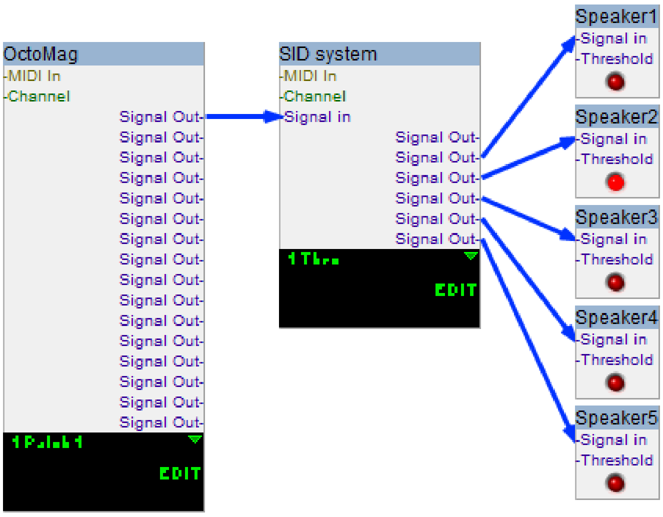


Figure 3.3: Five-speaker identification system implemented with the VST plugin, where OctoMag is the waveplayer and the SID system is the kernel-based speaker identification module. Each LED lights when the corresponding speaker is speaking.

CHAPTER 4

SEMI-SUPERVISED SPEAKER IDENTIFICATION UNDER COVARIATE SHIFT

This chapter is devoted to developing a semi-supervised speaker identification method under covariate shift.

4.1 Introduction

Popular methods of text-independent speaker identification are based on the Gaussian mixture model (GMM) [43] or kernel methods such as the support vector machine (SVM) [44, 45]. In these supervised learning methods, it is implicitly assumed that training and test data follow the same probability distribution. However, since the speech features vary over time due to session dependent variation, the recording environment change, and physical conditions/emotions, the training and test distributions are not necessarily the same in practice. In the paper [46], the influence of the session dependent variation of voice quality in speaker identification problems has been investigated and the identification performance was shown to decrease significantly over 3 months—the major cause for the performance degradation was the voice source characteristic variations.

To alleviate the influence of session dependent variation, it is popular to use several sessions of speaker utterance samples [6, 5] or to use *cepstral mean normalization*

(CMN) [7]. However, gathering several sessions of speaker utterance data and assigning the speaker ID to the collected data are expensive both in time and cost and therefore not realistic in practice. Moreover, it is not possible to perfectly remove the session dependent variation by CMN alone.

A more practical/effective setup would be *semi-supervised learning*, where unlabeled samples are additionally given from the testing environment. In semi-supervised learning, it is required that the probability distributions of training and test are related to each other in some sense; otherwise we may not be able to learn anything about the test probability distribution from the training samples. A common modeling assumption is called *covariate shift*, where the input (feature) probability distributions are different in the training and test phases but the conditional probability distribution of labels remains unchanged. In many real-world applications such as robot control [9, 47, 48], bioinformatics [49, 50], spam filtering [51], natural language processing [52, 53], brain-computer interfacing [54, 55], and econometrics [56], the covariate shift model has been shown to be useful. Covariate shift is also naturally induced in selective sampling or active learning scenarios [57, 58, 59, 60, 61]. For this reason, learning under covariate shift is receiving a great deal of attention these days in the machine learning community [22].

In this chapter, we formulate the semi-supervised speaker identification problem in the covariate shift framework and propose a method that can cope with voice quality variants. Under covariate shift, standard maximum likelihood estimation is no longer consistent. The influence of covariate shift can be asymptotically canceled by weighting the log-likelihood terms according to the *importance* [62]:

$$w(\mathrm{X}) = \frac{p_{te}(\mathrm{X})}{p_{tr}(\mathrm{X})},$$

where $p_{te}(\mathrm{X})$ and $p_{tr}(\mathrm{X})$ are test and training input densities. We apply this weighting idea in kernel logistic regression (KLR). The importance weight $w(\mathrm{X})$ is unknown in practice and needs to be estimated from data. For weight estimation, we utilize the Kullback-Leibler importance estimation procedure (KLIEP) [27, 63] since it is equipped with a built-in model selection procedure. The (regularized) kernel logistic regression model contain two tuning parameters: the kernel width and the regularization parameter. Usually those tuning parameters are optimized based on cross validation (CV). However, ordinary CV is no longer unbiased due to covariate shift and therefore is not reliable as a model selection method. To cope with this problem, we use importance weighted CV [55] for unbiased model selection. The validity of our approach is experimentally shown through text-independent/dependent speaker identification simulations.

The rest of this chapter is structured as follows. Section 4.2 formulates the semi-supervised speaker identification problem and review existing methods such as KLR and CV. In Section 4.3, importance weighting techniques for covariate shift adaptation are introduced. Experimental results are reported in Section 4.4.

4.2 Problem Formulation

In this section, we formulate the speaker identification problem from a machine learning point of view.

4.2.1 Kernel-based Speaker Identification

An utterance feature X pronounced by a speaker is expressed as a set of N mel-frequency cepstrum coefficient (MFCC) [38] vectors of d dimensions:

$$\mathrm{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{d \times N}. \tag{4.1}$$

For training, we are given n_{tr} labeled utterance samples:

$$\mathcal{Z}^{tr} = \{\mathrm{X}_i, y_i\}_{i=1}^{n_{tr}}, \tag{4.2}$$

where $y_i \in \{1, \ldots, K\}$ denotes the index of the speaker who pronounced X_i. The goal of speaker identification is to predict the speaker index of a test utterance sample X based on the training samples. We predict the speaker index c of the test sample X following the Bayes decision rule:

$$P(y = c|\mathrm{X}) > P(y = i|\mathrm{X}) \quad \forall\, i \neq c. \tag{4.3}$$

For approximating the class-posterior probability, we use the following parametric model $p(y = c|\mathrm{X}, \mathrm{V})$:

$$p(y = c|\mathrm{X}, \mathrm{V}) = \frac{\exp f_{\boldsymbol{v}_c}(\mathrm{X})}{\sum_{l=1}^{K} \exp f_{\boldsymbol{v}_l}(\mathrm{X})}, \tag{4.4}$$

where $\mathrm{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K]^\top \in \mathbb{R}^{K \times n_{tr}}$ is the parameter, $^\top$ denotes the transpose, and $f_{\boldsymbol{v}_l}$ is a discriminant function corresponding to the speaker l. This model is known as the softmax function and widely used in multiclass logistic regression. We use the following kernel regression model as the discriminant function $f_{\boldsymbol{v}_l}$ [6]:

$$f_{\boldsymbol{v}_l}(\mathrm{X}) = \sum_{i=1}^{n_{tr}} v_{l,i} \mathcal{K}(\mathrm{X}, \mathrm{X}_i) \quad l = 1, \ldots, K, \tag{4.5}$$

where $\boldsymbol{v}_l = (v_{l,1}, \ldots, v_{l,n_{tr}})^\top \in \mathbb{R}^{n_{tr}}$ are parameters corresponding the speaker l and $\mathcal{K}(\mathrm{X}, \mathrm{X}')$ is a kernel function. In this chapter, we use the *sequence kernel* [45] as the kernel function since it allows us to handle features with different size; for two utterance samples $\mathrm{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{d \times N}$ and $\mathrm{X}' = [\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{N'}] \in \mathbb{R}^{d \times N'}$ (generally $N \neq N'$), the sequence kernel is defined as

$$\mathcal{K}(\mathrm{X}, \mathrm{X}') = \frac{1}{NN'} \sum_{i=1}^{N} \sum_{i'=1}^{N'} k(\boldsymbol{x}_i, \boldsymbol{x}'_{i'}), \tag{4.6}$$

where $k(\boldsymbol{x}, \boldsymbol{x}')$ is a vectorial kernel; we use the Gaussian kernel:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right). \tag{4.7}$$

Note that kernel logistic regression is a modeling assumption, thus the true class-conditional probability may not be exactly realized by the kernel logistic regression model. This implies that there exists a model error, i.e., even when the parameter is chosen optimally, there remains an approximation error. This setup is not of course preferable, but more or less there exists a model error in practice since it is not generally possible to have an exact model in reality. Traditional machine learning theories often assume that the model at hand is correct (i.e., no model error exists). However, this is not realistic and not useful in practice, so in this chapter we explicitly take into account *model misspecification.*

4.2.2 KLR, CV, and Covariate Shift

Here, we show potential limitations of KLR and CV in the light of model misspecification.

The use of KLR and CV could be theoretically justified when the training utterance features and the test utterance features independently follow the *same* probability distribution with density $p(\mathrm{X})$ and the class label y follows the *common* conditional probability distribution $p(y|\mathrm{X})$ in the training and test phases. Indeed, if the above conditions are met, KLR is shown to be *consistent*, i.e., the learned parameter converges to the optimal value:

$$\lim_{n_{tr}\to\infty} \widehat{\mathrm{V}} = \mathrm{V}^*, \tag{4.8}$$

where $\widehat{\mathrm{V}}$ is the parameter learned by KLR and V^* is the optimal parameter that minimizes the expected prediction error for test samples:

$$\mathrm{V}^* = \operatorname*{argmin}_{\mathrm{V}} \iint I(y = \widehat{y}(\mathrm{X}|\mathrm{V}))p(y|\mathrm{X})p(\mathrm{X})\mathrm{d}y\mathrm{d}\mathrm{X}. \tag{4.9}$$

$\widehat{y}(\mathrm{X}|\mathrm{V})$ is an estimate of speaker of an utterance feature X for parameter V. Also, when $p(\mathrm{X})$ and $p(y|\mathrm{X})$ are common in the training and test phases, kCV is (almost) *unbiased* [3]:

$$\mathrm{E}_{\mathcal{Z}^{tr}}\left[\widehat{R}^{\mathcal{Z}^{tr}}_{kCV} - R^{\mathcal{Z}^{tr}}\right] \approx 0, \tag{4.10}$$

where $\mathrm{E}_{\mathcal{Z}^{tr}}$ is the expectation over the training set $\mathcal{Z}^{tr}$ and $R^{\mathcal{Z}^{tr}}$ is the expected prediction error defined by

$$R^{\mathcal{Z}^{tr}} = \iint I(y = \widehat{y}(\mathrm{X}; \mathcal{Z}^{tr}))p(y|\mathrm{X})p(\mathrm{X})\mathrm{d}y\mathrm{d}\mathrm{X}. \tag{4.11}$$

$\widehat{y}(\mathrm{X}; \mathcal{Z}^{tr})$ is a learned function from the training set $\mathcal{Z}^{tr}$.

However, in practical speaker identification, speech features are not stationary due to time-dependent voice variation, the recording environment change, and physical conditions/emotion. Thus, the training and test feature distributions are not the same. Then, the above good theoretical properties are no longer true[1].

In this chapter, we explicitly deal with such changing environment via the *covariate shift* model [62]—the input distributions change between the training and test phases, $p_{tr}(\mathrm{X}) \neq p_{te}(\mathrm{X})$, but the conditional distribution $p(y|\mathrm{X})$ remains unchanged.

4.3 Importance Weighting Techniques for Covariate Shift Adaptation

In this section, we show how to cope with covariate shift.

4.3.1 Parameter Learning and Model Selection under Covariate Shift

Here we show how KLR and CV could be extended and justified even under covariate shift.

4.3.1.1 Importance Sampling

In the absence of covariate shift, the expectation over test samples can be consistently estimated by the expectation over training samples since they are drawn from the same distribution. However, under covariate shift, the difference of input distributions should be explicitly taken into account. A basic technique for compensating for the distribution change is *importance sampling* [64], i.e., the expectation over training samples is weighted according to their importance in the test distribution. Indeed,

[1] If the KLR model is exactly correct, consistency of KLR and almost unbiasedness of CV still holds even when the feature distributions change between the training and test stages. However, the correct model assumption is not satisfied in reality.

for the importance weight

$$w(\mathrm{X}) = \frac{p_{te}(\mathrm{X})}{p_{tr}(\mathrm{X})}, \tag{4.12}$$

the expectation of some function $F(\mathrm{X})$ over the probability density $p_{te}(\mathrm{X})$ can be computed by

$$\begin{aligned} \mathrm{E}_{p_{te}(\mathrm{X})}[F(\mathrm{X})] &= \int F(\mathrm{X}) p_{te}(\mathrm{X}) \mathrm{dX} \\ &= \int F(\mathrm{X}) w(\mathrm{X}) p_{tr}(\mathrm{X}) \mathrm{dX} = \mathrm{E}_{p_{tr}(\mathrm{X})}[F(\mathrm{X}) w(\mathrm{X})]. \end{aligned} \tag{4.13}$$

4.3.1.2 Importance Weighted Kernel Logistic Regression

If the importance sampling technique is applied to KLR, we have the following importance weighted KLR (IWKLR) [62]:

$$\widetilde{\mathcal{P}}^{\log}(\mathrm{V}; \mathcal{Z}^{tr}) = -\sum_{i=1}^{n_{tr}} w(\mathrm{X}_i) \log P(y_i | \mathrm{X}_i, \mathrm{V}). \tag{4.14}$$

IWKLR is consistent even under covariate shift:

$$\lim_{n_{tr} \to \infty} \widetilde{\mathrm{V}} = \mathrm{V}^*, \tag{4.15}$$

where $\widetilde{\mathrm{V}}$ is the parameter learned by IWKLR and V^* is the optimal parameter that minimizes the expected prediction error for test samples:

$$\mathrm{V}^* = \underset{\mathrm{V}}{\operatorname{argmin}} \iint I(y = \widehat{y}(\mathrm{X}|\mathrm{V})) p(y|\mathrm{X}) p_{te}(\mathrm{X}) \mathrm{d}y \mathrm{dX}. \tag{4.16}$$

In practice, we may include a regularizer:

$$\widetilde{\mathcal{P}}_{\delta}^{\log}(\mathrm{V}; \mathcal{Z}^{tr}) = -\sum_{i=1}^{n_{tr}} w(\mathrm{X}_i) \log P(y_i | \mathrm{X}_i, \mathrm{V}) + \frac{\delta}{2} \mathrm{trace}(\mathrm{VKV}^\top), \tag{4.17}$$

where δ is the *regularization parameter.*

The Newton update rule for IWKLR is given by the same form as Eq.(2.9); the gradient and Hessian of (4.17) are given by

$$\nabla \mathcal{P}_{\delta}^{\log}(\mathrm{V}; \mathcal{Z}) = \{(\mathrm{P}(\mathrm{V}) - \mathrm{Y})\mathrm{W} + \delta \mathrm{V}\}\mathrm{K}, \tag{4.18}$$

$$\begin{aligned} \nabla^2 \mathcal{P}_{\delta}^{\log}(\mathrm{V}; \mathcal{Z}) &= \sum_{i=1}^{n_{tr}} w(\mathrm{X}_i)(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top) \otimes \boldsymbol{k}(\mathrm{X}_i)\boldsymbol{k}(\mathrm{X}_i)^\top \\ &\quad +(\mathrm{K}^\top \otimes \mathrm{I}), \end{aligned} \tag{4.19}$$

where

$$\mathrm{W} = \mathrm{diag}(w(\mathrm{X}_1), \ldots, w(\mathrm{X}_{n_{tr}})) \in \mathbb{R}^{n_{tr} \times n_{tr}}. \tag{4.20}$$

An approximation $\widetilde{\Delta \mathrm{V}}$ of the update factor is given as the solution of the following linear equation:

$$\sum_{i=1}^{n_{tr}} w(\mathrm{X}_i)(\mathrm{diag}(\boldsymbol{p}(\mathrm{X}_i)) - \boldsymbol{p}(\mathrm{X}_i)\boldsymbol{p}(\mathrm{X}_i)^\top)\widetilde{\Delta \mathrm{V}}\boldsymbol{k}(\mathrm{X}_i)\boldsymbol{k}(\mathrm{X}_i)^\top = \{(\mathrm{P}(\mathrm{V}) - \mathrm{Y})\mathrm{W} + \delta \mathrm{V}\}\mathrm{K}. \tag{4.21}$$

4.3.1.3 Importance Weighted Cross Validation

In a similar way as IWKLR, CV could also be enhanced based on the importance weighting technique: [55].

$$\widetilde{R}^{\mathcal{Z}^{tr}}_{kIWCV} = \frac{1}{k}\sum_{j=1}^{k} \frac{1}{|\mathcal{Z}^{tr}_j|} \sum_{(\mathrm{X},y)\in\mathcal{Z}^{tr}_j} w(\mathrm{X}) I(y = \widetilde{y}_{\mathcal{Z}^{tr}_i}(\mathrm{X})). \tag{4.22}$$

We refer to this method as k-fold importance-weighted CV (kIWCV). Even under covariate shift, kIWCV is almost unbiased:

$$\mathrm{E}_{\mathcal{Z}^{tr}}\left[\widetilde{R}^{\mathcal{Z}^{tr}}_{kIWCV} - R^{\mathcal{Z}^{tr}}\right] \approx 0. \tag{4.23}$$

4.3.2 Importance Weight Estimation

As shown above, the importance weight $w(\mathrm{X})$ plays a central role in covariate shift adaptation. However, the importance weight is usually unknown, thus it needs to be estimated from samples. Here, we assume that in addition to the training input samples $\mathcal{X}^{tr} = \{\mathrm{X}_i\}_{i=1}^{n_{tr}}$, we are given (unlabeled) test samples $\mathcal{X}^{te} = \{\mathrm{X}_i\}_{i=1}^{n_{te}}$ drawn independently from $p_{te}(\mathrm{X})$ (i.e., the semi-supervised setup).

Under this setup, the importance weight may be simply approximated by estimating $p_{tr}(\mathrm{X})$ and $p_{te}(\mathrm{X})$ from training and test samples separately and then taking their ratio. However, density estimation is known to be a hard problem and taking the ratio of estimated quantities tends to magnify the estimation error. Thus this two-shot process is not reliable in practice. In this chapter, we use a method that allows us to directly learn the importance weight function without going through density estimation. The method is called the *Kullback Leibler Importance Estimation Procedure (KLIEP)* [27, 63], and its derivation is described in Chapter 5.

4.3.3 Illustrative Examples

Here, we illustrate the behavior of IWKLR, IWCV, and KLIEP in covariate shift adaptation.

Figure 4.1 illustrates a two-dimensional binary classification problem under covariate shift. In this experiment, we define the optimal class posterior probability as follows:

$$\begin{aligned} p(y=+1|\boldsymbol{x}) &= \frac{1+\tanh(x^{(1)}-\min(0,x^{(2)}))}{2}, && (4.24)\\ p(y=-1|\boldsymbol{x}) &= 1-p(y=+1|\boldsymbol{x}), && (4.25)\end{aligned}$$

where $\boldsymbol{x}=[x^{(1)},x^{(2)}]^\top \in \mathbb{R}^2$ is the input vector. Data samples were generated from mixtures of Gaussian distributions as follows:

$$\begin{aligned} p_{tr}(\boldsymbol{x}) &= \sum_{k=1}^{2} \pi_k^{tr}\mathcal{N}(\mathrm{X}|\boldsymbol{\mu}_k^{tr},\Sigma_k^{tr}),\\ p_{te}(\boldsymbol{x}) &= \sum_{k=1}^{2} \pi_k^{te}\mathcal{N}(\mathrm{X}|\boldsymbol{\mu}_k^{te},\Sigma_k^{te}),\end{aligned}$$

where π_k^{tr} and π_k^{te} are mixing coefficients of training and test distributions, and $\mathcal{N}(\mathrm{X}|\boldsymbol{\mu},\Sigma)$ denotes the Gaussian density with mean $\boldsymbol{\mu}\in\mathbb{R}^2$ and covariance matrix $\Sigma\in\mathbb{R}^{2\times 2}$. In this experiment, we set the mixing coefficients, means, and covariances as described in Table 4.1.

Let the number of training and test samples be $n_{tr}=1000$ and $n_{te}=2000$. We use KLR/IWKLR with the linear kernel and employ CV/IWCV for tuning the regularization parameter δ. The value δ chosen by CV and IWCV for KLR and IWKLR were 10^{-6} and 1, respectively. The importance weights used in IWKLR and IWCV

Table 4.1: Setup of illustrative examples.

	$p_{tr}(\boldsymbol{x})$		$p_{te}(\boldsymbol{x})$	
	Mixture 1	Mixture 2	Mixture 1	Mixture 2
π	0.5	0.5	0.5	0.5
$\boldsymbol{\mu}$	$(-2, 2.5)$	$(2, 2.5)$	$(-3.5, -0.5)$	$(0.5, -0.5)$
Σ	$\begin{pmatrix} 0.5 & 0 \\ 0 & 2.5 \end{pmatrix}$	$\begin{pmatrix} 0.5 & 0 \\ 0 & 2.5 \end{pmatrix}$	$\begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$	$\begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$

are learned by KLIEP and LCV is used for choosing the Gaussian width τ in KLIEP. Figure 4.1 shows the decision boundaries obtained by KLR+CV and IWKLR+IWCV. For references, we also showed 'OPT', which is the optimal decision boundary given by Eqs.(4.24) and (4.25). As the figure clearly shows, IWKLR+IWCV gives the decision boundary that is closer to OPT for the test samples than plain KLR+CV. The correct classification rate of KLR+CV is 93.6%, while that of IWKLR+IWCV is 96.1%. This illustrates that, under covariate shift, the prediction performance can be improved by employing the importance weighting techniques.

4.4 Experiments

In this section, we report the results of speaker identification in the light of covariate shift adaptation.

4.4.1 Data and System Description

Training and test samples were collected from 10 male speakers, and we have conducted two types of experiment—text-dependent and text-independent speaker identification. In text-dependent speaker identification, the training and test sentences are common to all speakers. On the other hand, in text-independent speaker identification, the training sentences are common to all speakers, but the test sentences are different from training sentences.

Each speaker uttered several Japanese sentences for text-dependent and text-independent speaker identification evaluation. The following three sentences are used

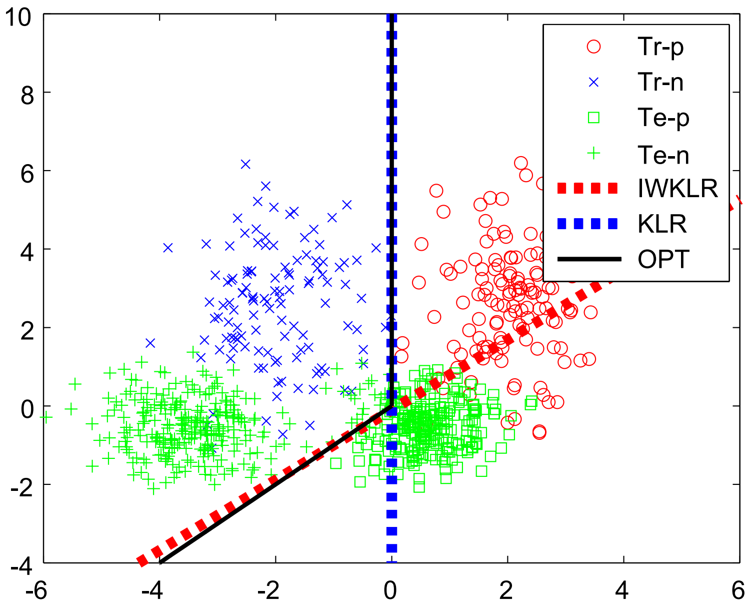


Figure 4.1: Decision boundaries obtained by IWKLR+IWCV and KLR+CV (red and blue dashed lines) and the optimal decision boundary (black solid line). '$\circ$' and '$\times$' are positive and negative training samples, while '$\Box$' and '$+$' are positive and negative test samples. Note that the input-output test samples are not used in the training of KLR and the output test samples are not used in the training of IWKLR—they are plotted in the figure for illustration purposes.

as training and test samples in the text-dependent speaker identification experiments (Japanese sentences written using the Hepburn system of Romanization):

- seno takasawa hyakunanajusseNchi hodode mega ookiku yaya futotteiru,
- oogoeo dashisugite kasuregoeni natte shimau,
- tashizaN hikizaNwa dekinakutemo eha kakeru.

In the text-independent speaker identification experiments, the following three sentences are used as training samples:

- seno takasawa hyakunanajusseNchi hodode mega ookiku yaya futotteiru,
- oogoeo dashisugite kasuregoeni natte shimau,
- tashizaN hikizaNwa dekinakutemo eha kakeru,

and the following five sentences are used as test samples:

- tobujiyuuwo eru kotowa jiNruino yume datta,
- hajimete ruuburubijutsukaNe haittanowa juuyoneNmaeno kotoda,
- jibuNno jitsuryokuwa jibuNga ichibaN yoku shitteiru hazuda,
- koremade shouneNyakyuu mamasaN bareenado chiikisupootsuo sasae shimiNni micchakushite kitanowamusuuno boraNtiadatta,
- giNzakeno tamagoo yunyuushite fukasase kaichuude sodateru youshokumo hajimatteiru.

The utterance samples for training were recorded in 1990/12, while the utterance samples for testing were recorded in 1991/3, 1991/6, and 1991/9, respectively. Since the recording time is different between training and test utterance samples, the voice quality variation is expected to be included. Thus, the target speaker identification problem is a challenging task.

The total duration of the training sentences is about 9 sec. The durations of the test sentences for text-dependent and text-independent speaker identifications are 9 sec and 24 sec, respectively. There are approximately 10 vowels in the sentences for every 1.5 sec.

The input utterance is sampled at 16kHz. A feature vector consists of 26 components: 12 MFCCs, the normalized log energy, and their first derivatives. Feature vectors are derived at every 10 ms over the 25.6-ms Hamming-windowed speech segment, and the *cepstral mean normalization* (CMN) is applied over the features to remove channel effects. We divide each utterance into 300-ms disjoint segments, each of which corresponds to a set of features of size 26 × 30. Thus the training set is given as $\mathcal{X}^{tr} = \{\mathrm{X}_i\}_{i=1}^{411}$ for text-independent and text-dependent speaker identification evaluations. For text-independent speaker identification, the sets of test samples for 1991/3, 1991/6, and 1991/9 are given as $\mathcal{X}_1^{te1} = \{\mathrm{X}_i\}_{i=1}^{907}$, $\mathcal{X}_1^{te2} = \{\mathrm{X}_i\}_{i=1}^{919}$, and $\mathcal{X}_1^{te3} = \{\mathrm{X}_i\}_{i=1}^{906}$, respectively. For text-dependent speaker identification, the sets of test data are given as $\mathcal{X}_2^{te1} = \{\mathrm{X}_i\}_{i=1}^{407}$, $\mathcal{X}_2^{te2} = \{\mathrm{X}_i\}_{i=1}^{407}$, and $\mathcal{X}_2^{te3} = \{\mathrm{X}_i\}_{i=1}^{412}$, respectively.

We compute the speaker identification rate at every 1.5s, 3.0s, and 4.5s and identify the speaker from the average posterior probability:

$$p(\mathrm{X}_t|\mathrm{V}) = \frac{1}{m}\sum_{i=1}^{m} p(\mathrm{X}_{t-i}|V), \tag{4.26}$$

where $m = 5, 10,$ and 15, respectively.

4.4.2 The Results of Speaker Identification under Covariate Shift

We compared GMM, KLR, and IWKLR by computing the speaker identification rates on the 1991/3, 1991/6, and 1991/9 datasets, respectively. For GMM and KLR training, we only use the 1990/12 dataset (inputs $\mathcal{X}^{tr}$ and their labels).

For GMM training, the means, diagonal covariance matrices, and mixing coefficients are initialized by the results of k-means clustering on all training sentences for all speakers; then these parameters are estimated via the EM algorithm [65] for each speaker. The number of mixtures is determined by 5-fold CV. In the test phase of GMM, we compare the probability $p(\mathrm{X}_t|\boldsymbol{\mu}_k, \Sigma_k) = \prod_{j=1}^{p} p(\boldsymbol{x}_{t-j}|\boldsymbol{\mu}_k, \Sigma_k), k = 1, \ldots, 10,$ where $\boldsymbol{\mu}_k$ and Σ_k are the means and covariance matrices for speaker k.

For IWKLR training, we use unlabeled samples $\mathcal{X}^{te1}$, $\mathcal{X}^{te2}$, and $\mathcal{X}^{te3}$ in addition to the training inputs $\mathcal{X}^{tr}$ and their labels (i.e., semi-supervised). We first estimate the importance weight from the training and test dataset pairs ($\mathcal{X}^{tr}$, $\mathcal{X}^{te1}$), ($\mathcal{X}^{tr}$, $\mathcal{X}^{te2}$), or ($\mathcal{X}^{tr}$, $\mathcal{X}^{te3}$) by KLIEP with 5-fold LCV, and we use 5-fold IWCV to decide the kernel band width σ and regularization parameter δ.

In our preliminary experiments, we observed that the kCV and kIWCV scores tend to be heavily affected by the way the data samples are split into k disjoint subsets (we used $k = 5$). We conjecture that this is due to non-i.i.d. nature of the

MFCC features, which is different from the theory. To obtain reliable experimental results, we decided to repeat the CV procedure 50 times with different random data splits and use the highest score for model selection.

Table 4.2 shows the text-independent speaker identification rates in percent for 1991/3, 1991/6, and 1991/9. IWKLR refers to IWKLR with σ and δ chosen by 5-fold IWCV, KLR refers to KLR with σ and δ chosen by 5-fold CV, and GMM refers to GMM with the number of mixtures chosen by 5-fold CV. The chosen values of these hyper-parameters are described in the bracket. 'Std' in the bottom line refers to the standard deviation of the estimated importance weights $\{w(\mathrm{X}_i)\}_{i=1}^{n_{tr}}$; the smaller the standard deviation is, the 'flatter' the importance weights are. Flat importance weights imply that there is no significant distribution change between the training and test phases. Thus, the standard deviation of the estimated importance weights may be regarded as a rough indicator of the degree of distribution change.

As can be seen from the table, IWKLR+IWCV outperforms GMM+CV and KLR+CV for all sessions. This result implies that importance weighting is useful in coping with the influence of non-stationarity in practical speaker identification such as utterance variation, the recording environment change, and physical conditions/emotions.

Table 4.3 summarizes the text-dependent speaker identification rates in percent for 1991/3, 1991/6, and 1991/9, showing that IWKLR+IWCV and KLR+CV slightly outperform GMM and are highly comparable to each other. The result that IWKLR+IWCV and KLR+CV are comparable in this experiment would be a reasonable consequence since the standard deviation of the estimated importance weights is very small in all three cases—implying that there is no significant distribution change and therefore no adaptation is necessary. This result indicates that the proposed method does not degrade the performance when there is no significant distribution change.

Overall, the proposed method tends to improve the performance when there exists a significant distribution change and it tends to maintain the good performance of the baseline method when no distribution change exists. Based on these experimental results, we conclude that the proposed method is a promising approach to handling session dependent variation.

Table 4.2: Correct classification rates for text-independent speaker identification. All values are in percent. IWKLR refers to IWKLR with σ and δ chosen by 5-fold IWCV, KLR refers to KLR with σ and δ chosen by 5-fold CV, and GMM refers to GMM with the number of mixtures chosen by 5-fold CV. The chosen values of these hyper-parameters are described in the bracket. 'Std' refers to the standard deviation of estimated importance weights $\{w(\mathrm{X}_i)\}_{i=1}^{n_{tr}}$, roughly indicating the degree of distribution change.

	1991/3			1991/6			1991/9		
	IWKLR	KLR	GMM	IWKLR	KLR	GMM	IWKLR	KLR	GMM
Time	(1.4, 10^{-2})	(1.0, 10^{-2})	(16)	(1.3, 10^{-4})	(1.0, 10^{-2})	(16)	(1.2, 10^{-4})	(1.0, 10^{-2})	(16)
1.5s	**91.0**	88.2	89.7	**91.0**	87.7	90.2	**94.8**	91.7	92.1
3.0s	**95.0**	92.9	94.4	**95.3**	91.1	94.0	**97.9**	96.3	95.0
4.5s	**97.7**	96.1	94.6	**97.4**	93.4	96.1	**98.8**	98.3	95.8
Std	0.34	n/a	n/a	0.37	n/a	n/a	0.35	n/a	n/a

Table 4.3: Correct classification rates for text-dependent speaker identification. All values are in percent. IWKLR refers to IWKLR with σ and δ chosen by 5-fold IWCV, KLR refers to KLR with σ and δ chosen by 5-fold CV, and GMM refers to GMM with the number of mixtures chosen by 5-fold CV. The chosen values of these hyper-parameters are described in the bracket. 'Std' refers to the standard deviation of estimated importance weights $\{w(\mathrm{X}_i)\}_{i=1}^{n_{tr}}$, roughly indicating the degree of distribution change.

	1991/3			1991/6			1991/9		
	IWKLR	KLR	GMM	IWKLR	KLR	GMM	IWKLR	KLR	GMM
Time	(1.2, 10^{-4})	(1.0, 10^{-2})	(16)	(1.2, 10^{-4})	(1.0, 10^{-2})	(16)	(1.2, 10^{-4})	(1.0, 10^{-2})	(16)
1.5s	**100.0**	98.9	96.8	97.5	96.2	**97.8**	**100.0**	**100.0**	98.2
3.0s	**100.0**	**100.0**	97.7	97.5	97.2	**98.1**	**100.0**	**100.0**	98.4
4.5s	**100.0**	**100.0**	97.9	**98.9**	97.4	98.3	**100.0**	**100.0**	98.5
Std	0.05	n/a	n/a	0.05	n/a	n/a	0.05	n/a	n/a

CHAPTER 5

DIRECT IMPORTANCE ESTIMATION WITH GAUSSIAN MIXTURE MODEL AND PRINCIPAL COMPONENT ANALYZERS

This chapter is devoted to developing a direct importance estimation method for outlier detection problem.

5.1 Introduction

Humanoid robots are desired to automatically add the unknown speakers into dictionary, and it can be formulated as the outlier detection problem (i.e., outlier can be the unknown speakers). Since the outlier detection problem can be solved via the log likelihood between the unknown speaker and the known speakers in the dictionary, to improve the estimation accuracy of log likelihood is an important issue to for outlier detection problems.

Recently, the problem of estimating the ratio of two probability density functions (a.k.a. the *importance*) has received a great deal of attention since it can be used for various data processing purposes.

Covariate shift adaptation would be a typical example [22]. Covariate shift is a situation in supervised learning where the training and test input distributions are different while the conditional distribution of output remains unchanged [23]. In many real-world applications such as robot control [48], bioinformatics [50], spam filtering [51], natural language processing [53], brain-computer interfacing [55], and speaker identification [66], covariate shift adaptation has been shown to be useful. Covariate shift is also naturally induced in selective sampling or active learning scenarios and adaptation improves the generalization performance [59, 60, 61, 67].

Another example in which the importance is useful is outlier detection [25]. The outlier detection task addressed in that chapter is to identify irregular samples (i.e., outliers) in an evaluation dataset based on a model dataset that only contains regular samples (i.e., inliers). If the density ratio of two datasets is considered, the importance values for regular samples are close to one, while those for outliers tend to be significantly deviated from one. Thus the values of the importance could be used as an index of the degree of outlyingness. A similar idea can also be applied to change detection in time series [26].

A naive approach to approximating the importance function is to estimate training and test probability densities separately and then take the ratio of the estimated densities. However, density estimation itself is a difficult problem and taking the ratio of estimated densities can magnify the estimation error. In order to avoid density estimation, a semi-parametric approach called the *Kullback-Leibler Importance Estimation Procedure* (KLIEP) was proposed [27]. KLIEP does not involve density estimation but directly models the importance function. The parameters in the importance model is learned so that the Kullback-Leibler divergence from the true test distribution to the estimated test distribution is minimized without going through density estimation. KLIEP was shown to be useful in covariate shift adaptation [27] and outlier detection [25]. A typical implementation of KLIEP employs a spherical Gaussian kernel model and the Gaussian width is chosen by cross validation. This means that when the true importance function is correlated, the performance of KLIEP is expected to be degraded (see Figs.5.1-(b) and 5.1-(c))

To cope with this problem, we propose to use a Gaussian mixture model (GMM) [68, 69] in the KLIEP algorithm and learn the covariance matrix of the Gaussian components at the same time. This will allow us to learn the importance function more adaptively even when the true importance function contains high correlation (see Fig.5.1-(d)). We develop an expectation-maximization procedure for learning the parameters in the Gaussian mixture model. The effectiveness of the proposed method—which we call the Gaussian mixture KLIEP (GM-KLIEP)—is shown through experiments.

However, since we need to estimate the inverse of covariance matrices for GM-KLIEP, it fails to estimate the covariance matrices when the rank-deficient input vectors are observed. To deal with the rank deficient data, it is popular to use the dimensionality reduction method such as principal component analysis (PCA) as a pre-processing tool. Thus, in this chapter, we employ the mixture of probabilistic PCA model instead of GMM for importance estimation, and we call the method as PPCA mixture KLIEP (PM-KLIEP).

5.2 Background

In this section, we formulate the importance estimation problem and briefly review the KLIEP method.

5.2.1 Problem Formulation

Let $\mathcal{D} \in \mathbb{R}^d$ be the data domain and suppose we are given i.i.d. training samples $\{\boldsymbol{x}_i^{tr}\}_{i=1}^{n_{tr}}$ from a training data distribution with density $p_{tr}(\boldsymbol{x})$ and i.i.d. test samples $\{\boldsymbol{x}_j^{te}\}_{j=1}^{n_{te}}$ from a test data distribution with density $p_{te}(\boldsymbol{x})$. We assume that $p_{tr}(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathcal{D}$. The goal of this chapter is to develop a method of estimating the *importance* $w(\boldsymbol{x})$ from $\{\boldsymbol{x}_i^{tr}\}_{i=1}^{n_{tr}}$ and $\{\boldsymbol{x}_j^{te}\}_{j=1}^{n_{te}}$:

$$w(\boldsymbol{x}) = \frac{p_{te}(\boldsymbol{x})}{p_{tr}(\boldsymbol{x})}.$$

Our key restriction is that we avoid estimating densities $p_{te}(\boldsymbol{x})$ and $p_{tr}(\boldsymbol{x})$ when estimating the importance $w(\boldsymbol{x})$.

5.2.2 Kullback-Leibler Importance Estimation Procedure

Kullback-Leibler Importance Estimation Procedure (KLIEP) allows one to directly estimate $w(\boldsymbol{x})$ without going through density estimation [27]. In KLIEP, the following linear importance model is used:

$$\widehat{w}(\boldsymbol{x}) = \sum_{l=1}^{b} \alpha_l \varphi_l(\boldsymbol{x}), \tag{5.1}$$

where $\{\alpha_l\}_{l=1}^{b}$ are parameters, b is the number of parameters, and $\varphi_l(\boldsymbol{x})$ is a basis function. In the original KLIEP paper [27], the Gaussian kernel was chosen as the basis functions:

$$\varphi_l(\boldsymbol{x}) = \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{c}_l\|^2}{2\tau^2}\right),$$

where τ^2 is the Gaussian width and $\boldsymbol{c}_l$ is a template point randomly chosen from the test set $\{\boldsymbol{x}_i\}_{i=1}^{n_{te}}$. Using the model $\widehat{w}(\boldsymbol{x})$, one can estimate the test data density $p_{te}(\boldsymbol{x})$ as

$$\widehat{p}_{te}(\boldsymbol{x}) = \widehat{w}(\boldsymbol{x}) p_{tr}(\boldsymbol{x}).$$

Based on this, $\{\alpha_l\}_{l=1}^{b}$ is determined so that the Kullback-Leibler divergence from $p_{te}(\boldsymbol{x})$ to $\widehat{p}_{te}(\boldsymbol{x})$ minimized:

$$\begin{aligned} KL[p_{te}(\boldsymbol{x}) \| \widehat{p}_{te}(\boldsymbol{x})] &= \int p_{te}(\boldsymbol{x}) \ln \frac{p_{te}(\boldsymbol{x})}{p_{tr}(\boldsymbol{x})\widehat{w}(\boldsymbol{x})} d\boldsymbol{x} \\ &= \int p_{te}(\boldsymbol{x}) \ln \frac{p_{te}(\boldsymbol{x})}{p_{tr}(\boldsymbol{x})} d\boldsymbol{x} - \int p_{te}(\boldsymbol{x}) \ln \widehat{w}(\boldsymbol{x}) d\boldsymbol{x}. \end{aligned}$$

The first term in the above equation is independent of $\{\alpha_l\}_{l=1}^{b}$, so it can be ignored. Let us define the second term as J:

$$J = \int p_{te}(\boldsymbol{x}) \ln \widehat{w}(\boldsymbol{x}) d\boldsymbol{x} \approx \frac{1}{n_{te}} \sum_{j=1}^{n_{te}} \ln \widehat{w}(\boldsymbol{x}_j^{te}),$$

where the expectation over the test distribution is approximated by the test sample average. Since $\widehat{p}_{te}(\boldsymbol{x})$ is a probability density, the following equation should hold:

$$1 = \int \widehat{p}_{te}(\boldsymbol{x})d\boldsymbol{x} = \int p_{tr}(\boldsymbol{x})\widehat{w}(\boldsymbol{x})d\boldsymbol{x} \approx \frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}} \widehat{w}(\boldsymbol{x}_i^{tr}),$$

where the expectation over the training distribution is approximated by the training sample average. Then the KLIEP optimization problem is given as follows:

$$\max_{\{\alpha_l\}_{l=1}^{b}} \left[\sum_{j=1}^{n_{te}} \ln\left(\sum_{l=1}^{b} \alpha_l \varphi_l(\boldsymbol{x}_j^{te})\right)\right]$$

$$\text{s.t. } \frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}}\sum_{l=1}^{b} \alpha_l \varphi_l(\boldsymbol{x}_i^{tr}) = 1 \text{ and } \alpha_1, \ldots, \alpha_b \geq 0.$$

5.2.3 Model Selection by Likelihood Cross Validation

The choice of the Gaussian width τ in KLIEP heavily affects the performance of importance estimation. Since KLIEP is based on the maximization of the score J, it is natural to determine τ so that J is maximized.

The expectation over $p_{te}(\boldsymbol{x})$ involved in J can be numerically approximated by *likelihood cross validation* (LCV) as follows [27]: First divide the test samples $\{\boldsymbol{x}_j^{te}\}_{j=1}^{n_{te}}$ into K disjoint subsets $\{\mathcal{X}_i^{te}\}_{i=1}^{K}$ of approximately the same size. Then obtain an importance estimate $\widehat{w}_k(\boldsymbol{x})$ from $\{\mathcal{X}_j^{te}\}_{j\neq k}$ (i.e., without $\mathcal{X}_k^{te}$) and approximate the score J using $\mathcal{X}_k^{te}$ as

$$\widehat{J}_k = \frac{1}{|\mathcal{X}_k^{te}|}\sum_{\boldsymbol{x}\in\mathcal{X}_k^{te}} \ln \widehat{w}_k(\boldsymbol{x}).$$

This procedure is repeated for $k = 1, \ldots, K$ and the average of $\widehat{J}_k$ over all k is used as an estimate of J:

$$\widehat{J} = \frac{1}{K}\sum_{k=1}^{K} \widehat{J}_k.$$

For model selection, $\widehat{J}$ is computed for all model candidates (the Gaussian width τ in the current setting) and choose the one that maximizes $\widehat{J}$.

5.3 KLIEP with Gaussian Mixture Models

In this section, we propose our new method, *the Gaussian mixture KLIEP* (GM-KLIEP).

Instead of the linear model (5.1), we use a Gaussian mixture model as an importance model:

$$w(\boldsymbol{x}) = \sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}|\boldsymbol{m}_l, \boldsymbol{\Sigma}_l),$$

where π_l are mixing coefficients, $\mathcal{N}(\boldsymbol{x}|\boldsymbol{m}_l, \boldsymbol{\Sigma}_l)$ is the Gaussian density with mean vector $\boldsymbol{m}_l$ and covariance matrix $\boldsymbol{\Sigma}_l$, and b is the number of mixture components. Then the KLIEP optimization problem becomes

$$\begin{aligned}
&\max_{\{\pi_l, \boldsymbol{m}_l, \boldsymbol{\Sigma}_l\}_{l=1}^{b}} \left[\sum_{j=1}^{n_{te}} \ln \left(\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_j^{te}|\boldsymbol{m}_l, \boldsymbol{\Sigma}_l) \right) \right] \\
&\text{s.t. } \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_i^{tr}|\boldsymbol{m}_l, \boldsymbol{\Sigma}_l) = 1, \\
&\pi_1, \ldots, \pi_b \geq 0.
\end{aligned} \tag{5.2}$$

Here, we employ an expectation-maximization (EM) algorithm [68] for optimization:

Initialization step: Initialize the means $\boldsymbol{m}_k$, the covariances $\boldsymbol{\Sigma}_k$, and the mixing coefficients π_k.

E-step: Evaluate the responsibility values γ_{kj} using the current parameters:

$$\gamma_{kj} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_j^{te}|\boldsymbol{m}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_j^{te}|\boldsymbol{m}_l, \boldsymbol{\Sigma}_l)}.$$

M-step: Re-estimate the parameters using the current responsibility values:

$$\begin{aligned}
\boldsymbol{m}_k^{new} &= \frac{\sum_{j=1}^{n_{te}} \gamma_{kj} \boldsymbol{x}_j^{te}}{\sum_{j=1}^{n_{te}} \gamma_{kj}}, \\
\boldsymbol{\Sigma}_k^{new} &= \frac{\sum_{j=1}^{n_{te}} \gamma_{kj} (\boldsymbol{x}_j^{te} - \boldsymbol{m}_k^{new})(\boldsymbol{x}_i^{te} - \boldsymbol{m}_k^{new})^\top}{\sum_{j=1}^{n_{te}} \gamma_{kj}} + \delta \mathrm{I}, \\
\pi_k^{new} &= \frac{n_{tr} \sum_{j=1}^{n_{te}} \gamma_{kj}}{n_{te} \sum_{i=1}^{n_{tr}} \mathcal{N}(\boldsymbol{x}_i^{tr}|\boldsymbol{m}_k, \boldsymbol{\Sigma}_k)},
\end{aligned}$$

where δ is the regularization parameter and I the identity matrix.

Evaluation step: Evaluate the log-likelihood:

$$\ln p(\boldsymbol{x}|\boldsymbol{m}, \boldsymbol{\Sigma}, \pi) = \sum_{j=1}^{n_{te}} \ln \left(\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_j^{te} | \boldsymbol{m}_l^{new}, \boldsymbol{\Sigma}_l^{new}) \right).$$

Repeat the E- and M-steps until the log-likelihood converges.

Practically, we may use the k-means clustering algorithm for parameter initialization [68] and LCV is used for tuning the number of mixtures b and the regularization parameter δ.

5.3.1 Derivation of the EM Algorithm

Here, we show the derivation of the EM algorithm.

The cost function of GM-KLIEP is given by

$$J(\boldsymbol{\pi}, \mathrm{M}, \boldsymbol{\Sigma}) = \sum_{j=1}^{n_{te}} \ln \left(\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_j^{te} | \boldsymbol{m}_l, \boldsymbol{\Sigma}_l) \right). \qquad (5.3)$$

Differentiating Eq.(5.3) with respect to $\boldsymbol{m}_k$, we have

$$\frac{\partial J(\boldsymbol{\pi}, \mathrm{M}, \boldsymbol{\Sigma})}{\partial \boldsymbol{m}_k} = \sum_{j=1}^{n_{te}} \frac{\pi_k \mathcal{N}(\boldsymbol{x}_j^{te} | \boldsymbol{m}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_j^{te} | \boldsymbol{m}_l, \boldsymbol{\Sigma}_l)} \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_j^{te} - \boldsymbol{m}_k).$$

Equating this to zero, we have

$$\boldsymbol{m}_k = \frac{\sum_{j=1}^{n_{te}} \gamma_{kj} \boldsymbol{x}_j^{te}}{\sum_{j=1}^{n_{te}} \gamma_{kj}},$$

where

$$\gamma_{kj} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_j^{te} | \boldsymbol{m}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_j^{te} | \boldsymbol{m}_l, \boldsymbol{\Sigma}_l)}.$$

Similarly, differentiating Eq.(5.3) with respect to $\boldsymbol{\Sigma}_k$, we have

$$\begin{aligned}\frac{\partial J(\boldsymbol{\pi}, \mathrm{M}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}_k} &= \sum_{j=1}^{n_{te}} \frac{\pi_k \mathcal{N}(\boldsymbol{x}_j^{te} | \boldsymbol{m}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_j^{te} | \boldsymbol{m}_l, \boldsymbol{\Sigma}_l)} \\ &\quad \times \left(-\frac{1}{2} (\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_j^{te} - \boldsymbol{m}_k)(\boldsymbol{x}_j^{te} - \boldsymbol{m}_k)^\top \boldsymbol{\Sigma}_k^{-1}) \right) \\ &= \sum_{j=1}^{n_{te}} \gamma_{kj} \left(-\frac{1}{2} (\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_j^{te} - \boldsymbol{m}_k)(\boldsymbol{x}_j^{te} - \boldsymbol{m}_k)^\top \boldsymbol{\Sigma}_k^{-1}) \right).\end{aligned}$$

Equating this to zero, we have

$$\boldsymbol{\Sigma}_k \;=\; \frac{\sum_{j=1}^{n_{te}} \gamma_{kj}(\boldsymbol{x}_j^{te} - \boldsymbol{m}_k)(\boldsymbol{x}_i^{te} - \boldsymbol{m}_k)^\top}{\sum_{j=1}^{n_{te}} \gamma_{kj}}.$$

Finally, in order to satisfy the constraint (5.2), we introduce the Lagrange multiplier λ as

$$\begin{aligned} J(\boldsymbol{\pi}, \mathrm{M}, \boldsymbol{\Sigma}) &= \sum_{j=1}^{n_{te}} \ln\left(\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_j^{te}|\boldsymbol{m}_l, \boldsymbol{\Sigma}_l)\right) \\ &+ \lambda\left(\sum_{i=1}^{n_{tr}}\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_i^{tr}|\boldsymbol{m}_l, \boldsymbol{\Sigma}_l) - n_{tr}\right). \end{aligned}$$

Differentiating this with respect to π_k and equating it to zero, we have

$$\begin{aligned} \frac{\partial J(\boldsymbol{\pi}, \mathrm{M}, \boldsymbol{\Sigma})}{\partial \pi_k} &= \sum_{j=1}^{n_{te}} \frac{\mathcal{N}(\boldsymbol{x}_j^{te}|\boldsymbol{m}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_j^{te}|\boldsymbol{m}_l, \boldsymbol{\Sigma}_l)} + \lambda \sum_{i=1}^{n_{tr}} \mathcal{N}(\boldsymbol{x}_i^{tr}|\boldsymbol{m}_l, \boldsymbol{\Sigma}_l) \\ &= 0. \end{aligned} \tag{5.4}$$

Summing up this for all $k = 1, \ldots, b$, we have

$$\lambda \sum_{k=1}^{b}\sum_{i=1}^{n_{tr}} \pi_k \mathcal{N}(\boldsymbol{x}_i^{tr}|\boldsymbol{m}_k, \boldsymbol{\Sigma}_k) = -\sum_{k=1}^{b}\sum_{j=1}^{n_{te}} \frac{\pi_k \mathcal{N}(\boldsymbol{x}_j^{te}|\boldsymbol{m}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{b} \pi_l \mathcal{N}(\boldsymbol{x}_j^{te}|\boldsymbol{m}_l, \boldsymbol{\Sigma}_l)}.$$

Solving this in terms of λ, we have

$$\lambda = -\frac{n_{te}}{n_{tr}}.$$

Inserting this back into Eq.(5.4), we have

$$\pi_k = \frac{n_{tr} \sum_{j=1}^{n_{te}} \gamma_{kj}}{n_{te} \sum_{i=1}^{n_{tr}} N(\boldsymbol{x}_i^{tr}|\boldsymbol{m}_k, \boldsymbol{\Sigma}_k)}.$$

5.4 KLIEP with Mixture of Probabilistic Principal Component Analyzers

In this section, we propose our new method, *PPCA Mixture KLIEP* (PM-KLIEP).

Instead of the linear model (5.1), we use a probabilistic PCA (PPCA) mixture as the importance model:

$$\begin{aligned} w(\boldsymbol{x}) &= \sum_{l=1}^{b} \pi_l p(\boldsymbol{x}|\Theta_l), \\ p(\boldsymbol{x}|\Theta_l) &= (2\pi\sigma_l^2)^{-d/2} \exp\left\{-\frac{1}{2\sigma_l^2} \parallel \boldsymbol{x} - \mathrm{W}_l \boldsymbol{z}_l - \boldsymbol{m}_l \parallel^2\right\}, \end{aligned}$$

where π_l are mixing coefficients, $p(\boldsymbol{x}|\Theta_l)$ is the probability density function with $\Theta_l = \{\mathrm{W}_l \in \mathbb{R}^{d\times m}, \boldsymbol{m}_l \in \mathbb{R}^d, \sigma_l^2 \in \mathbb{R}\}$, $\boldsymbol{z}_l$ is a latent indicator variable, d is dimensionality of $\boldsymbol{x}$, $m \leq d$ is the dimensionality of the latent space, and b is the number of mixture components. Then the KLIEP optimization problem becomes

$$\max_{\{\pi_l,\Theta_l\}_{l=1}^b} \left[\sum_{j=1}^{n_{\mathrm{te}}} \ln\left(\sum_{l=1}^{b} \pi_l p(\boldsymbol{x}|\Theta_l)\right)\right]$$

$$\text{s.t. } \frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}}\sum_{l=1}^{b} \pi_l p(\boldsymbol{x}|\Theta_l) = 1, \text{ and } \pi_1, \ldots, \pi_b \geq 0.$$

Here, we employ the expectation-maximization (EM) algorithm [68] for optimization:

Initialization step: Initialize the mapping function W_k, the mean $\boldsymbol{m}_k$, the variance σ_k, and the mixing coefficients π_k.

E-step: Evaluate the responsibility values γ_{kj} using the current parameters:

$$\gamma_{kj} = \frac{\pi_k p(\boldsymbol{x}_j^{\mathrm{te}}|\Theta_k)}{\sum_{l=1}^{b} \pi_l p(\boldsymbol{x}_j^{\mathrm{te}}|\Theta_l)}.$$

M-step: Re-estimate the parameters using the current responsibility values:

$$\begin{aligned}
\boldsymbol{m}_k &= \frac{\sum_{j=1}^{n_{\mathrm{te}}} \gamma_{kj}(\boldsymbol{x}_j^{\mathrm{te}} - \mathrm{W}_k \boldsymbol{z}_{kj})}{\sum_{j=1}^{n_{\mathrm{te}}} \gamma_{kj}}, \\
\mathrm{W}_k &= \left(\sum_{j=1}^{n_{\mathrm{te}}} \gamma_{kj}(\boldsymbol{x}_j^{\mathrm{te}} - \boldsymbol{m}_k)\boldsymbol{z}_{kj}{}^\top\right)\left(\sum_{j=1}^{n_{\mathrm{te}}} \gamma_{kj}\mathrm{C}_{kj}\right)^{-1}, \\
\pi_k &= \frac{n_{\mathrm{tr}} \sum_{j=1}^{n_{\mathrm{te}}} \gamma_{kj}}{n_{\mathrm{te}} \sum_{i=1}^{n_{\mathrm{tr}}} p(\boldsymbol{x}_i^{\mathrm{tr}}|\Theta_k)}, \\
\sigma_k^2 &= \frac{1}{d\sum_{j=1}^{n_{\mathrm{te}}} \gamma_{kj}}\left(\sum_{j=1}^{n_{\mathrm{te}}} \gamma_{kj} \parallel \boldsymbol{x}_j^{\mathrm{te}} - \boldsymbol{m}_k \parallel^2 - 2\sum_{j=1}^{n_{\mathrm{te}}} \gamma_{kj}\boldsymbol{z}_{kj}{}^\top \mathrm{W}_k{}^\top (\boldsymbol{x}_j^{\mathrm{te}} - \boldsymbol{m}_k)\right. \\
&\quad \left. + \sum_{j=1}^{n_{\mathrm{te}}} \gamma_{kj}\mathrm{tr}(\mathrm{C}_{kj}\mathrm{W}_k{}^\top \mathrm{W}_k)\right), \\
\boldsymbol{z}_{kj} &= \mathrm{M}_k^{-1}\mathrm{W}_k(\boldsymbol{x}_j^{\mathrm{te}} - \boldsymbol{m}_k), \\
\mathrm{C}_{kj} &= \sigma_i^2\mathrm{M}_k^{-1} + \boldsymbol{z}_{kj}\boldsymbol{z}_{kj}{}^\top, \\
\mathrm{M}_k &= \sigma_i^2\mathrm{I} + \mathrm{W}_k{}^\top \mathrm{W}_k.
\end{aligned}$$

where I is the identity matrix.

Evaluation step: Evaluate the log-likelihood:

$$\ln p(\boldsymbol{x}|\boldsymbol{\pi}, \Theta) = \sum_{j=1}^{n_{\mathrm{te}}} \ln \left(\sum_{l=1}^{b} \pi_l p(\boldsymbol{x}_i^{\mathrm{te}}|\Theta_l) \right).$$

Repeat the E- and M-steps until the log-likelihood converges.

Note that, we use LCV for tuning the number of mixtures b and the dimensionality of the latent space m.

5.5 Experiments

In this section, we compare the performance of GM-KLIEP with the original KLIEP.

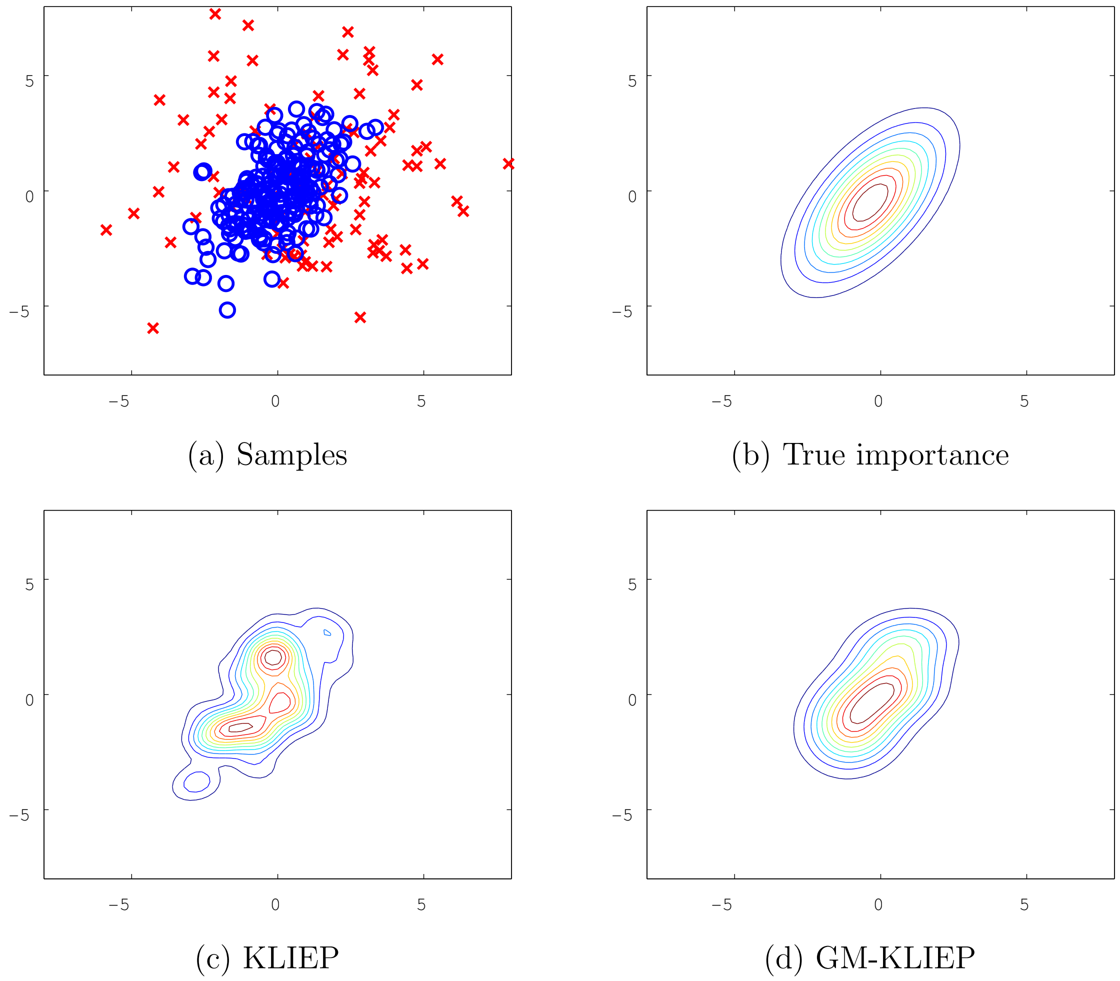
5.5.1 Illustrative Example for GM-KLIEP

Let us consider a toy two-dimensional importance estimation problem, where the true training and test density functions are defined as

$$\begin{aligned} p_{tr}(\boldsymbol{x}) &= \mathcal{N}\left(\boldsymbol{x} \,\middle|\, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}\right), \\ p_{te}(\boldsymbol{x}) &= \mathcal{N}\left(\boldsymbol{x} \,\middle|\, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1.5 & 1 \\ 1 & 2.5 \end{bmatrix}\right). \end{aligned}$$

In KLIEP, we set $b = 100$ and use the Gaussian kernel as the basis function; the kernel width is chosen based on 5-fold LCV. In GM-KLIEP, we use the k-means clustering algorithm for parameter initialization [68], and choose the number of mixtures and the regularization parameter based on 5-fold LCV.

We draw $n_{tr} = 100$ training samples and $n_{te} = 1000$ test samples following the above densities, which are depicted in Fig.5.1-(a). Figures 5.1-(b), 5.1-(c), and 5.1-(d) are the contour plots of the true importance function, the estimated importance function by KLIEP, and an estimated importance function by GM-KLIEP, respectively. The results show that GM-KLIEP can capture the correlated profile of the



(a) Samples

(b) True importance

(c) KLIEP

(d) GM-KLIEP

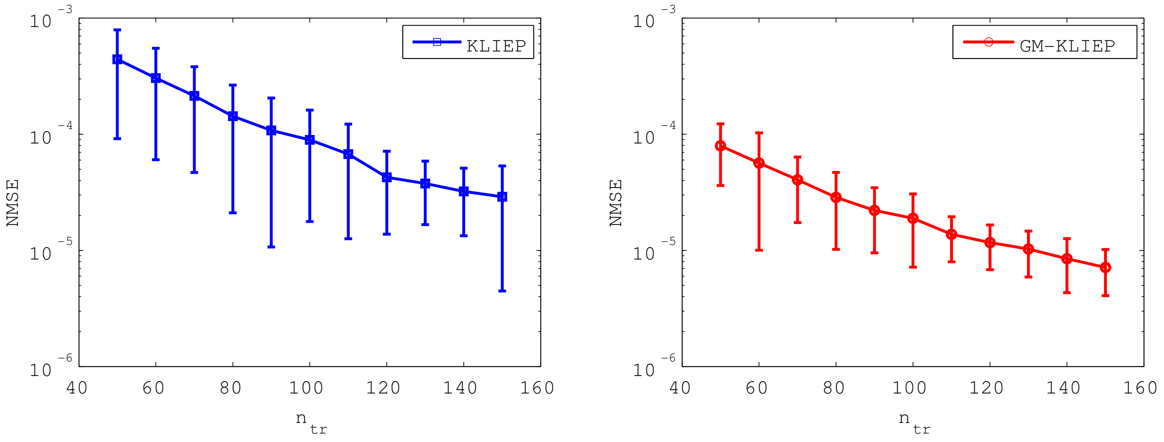
Figure 5.1: Samples and contour plots of the true importance function, the estimated importance function by KLIEP, and an estimated importance function by GM-KLIEP in the illustrative example.

true importance function better than the original KLIEP. The result of KLIEP seems to be rather overfitted due to high flexibility of the kernel model.

Next, we vary the number of training samples as $n_{tr} = 50, 60, \ldots, 150$ and quantitatively compare the performance of KLIEP and GM-KLIEP. We run the experiments 100 times for each n_{tr}, and evaluate the quality of an importance estimate $\widehat{w}(\boldsymbol{x})$ by the *normalized mean squared error* (NMSE) [27]:

$$\mathrm{NMSE} = \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \left(w(\boldsymbol{x}_i^{tr}) - \widehat{w}(\boldsymbol{x}_i^{tr}) \right)^2,$$

where $\Sigma_{i=1}^{n_{tr}} \widehat{w}(\boldsymbol{x}_i^{tr})$ and $\Sigma_{i=1}^{n_{tr}} w(\boldsymbol{x}_i^{tr})$ are normalized to be one, respectively.



(a) KLIEP (b) GM-KLIEP

Figure 5.2: NMSEs averaged over 100 trials (log scale) in the illustrative examples.

NMSEs averaged over 100 trials are plotted in Figs.2-(a) and 2-(b), showing that the errors of both methods tend to decrease as the number of training samples grows. GM-KLIEP tends to outperform the plain KLIEP, especially when the number of training samples is small; indeed, GM-KLIEP is shown to be significantly better than KLIEP by the *t-test* at the significance level 5%.

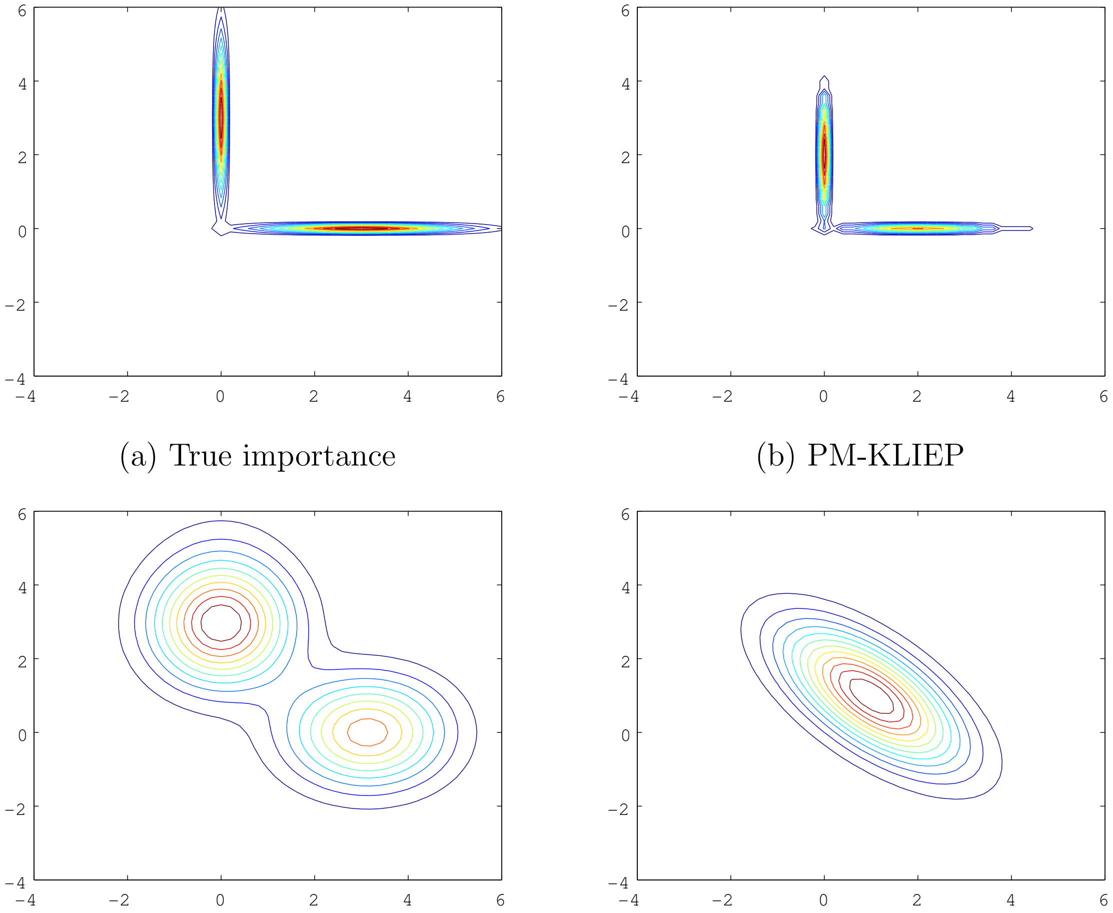
5.5.2 Illustrative Example

Let us first consider a rank-deficient two-dimensional importance estimation problem. The true training and test density functions are defined as

$$p_{\mathrm{tr}}(\boldsymbol{x}) = \frac{1}{2}\mathcal{N}\left(\boldsymbol{x} \;\middle|\; \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & \epsilon \end{bmatrix}\right) + \frac{1}{2}\mathcal{N}\left(\boldsymbol{x} \;\middle|\; \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \epsilon & 0 \\ 0 & 2 \end{bmatrix}\right),$$

$$p_{\mathrm{tr}}(\boldsymbol{x}) = \frac{1}{2}\mathcal{N}\left(\boldsymbol{x} \;\middle|\; \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix}\right) + \frac{1}{2}\mathcal{N}\left(\boldsymbol{x} \;\middle|\; \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} \epsilon & 0 \\ 0 & 1 \end{bmatrix}\right),$$

where $\epsilon = 2.22 \times 10^{-16}$. In this experiment, we draw $n_{\mathrm{tr}} = 100$ training samples and $n_{\mathrm{te}} = 1000$ test samples. In KLIEP, we set $b = 100$ and use the Gaussian kernel as the basis function; the kernel width is chosen based on 5-fold LCV. In GM-KLIEP, we use the k-means clustering algorithm for parameter initialization [68], and we choose the number of mixtures based on 5-fold LCV. In PM-KLIEP, we choose the number of mixtures and the dimension of the latent space via 5-fold LCV.



(a) True importance

(b) PM-KLIEP

(c) KLIEP

(d) GM-KLIEP

Figure 5.3: Contour plots of the true importance function, and the importance functions estimated by KLIEP, GM-KLIEP, and PM-KLIEP for the illustrative example.

Figure 5.3 depicts the true importance, along with the importance functions estimated by KLIEP, PM-KLIEP, and GM-KLIEP, respectively. As can be seen, PM-KLIEP accurately estimates the importance from the rank-deficient data, while KLIEP and GM-KLIEP do not.

5.5.3 Application to Inlier-based Outlier Detection

Next, we compare the performance of KLIEP and GM-KLIEP with the proposed PM-KLIEP method for inlier-based outlier detection.

Datasets provided by IDA [70] are used for performance evaluation; we exclude the "splice" dataset since it is discrete. The datasets are binary classification and each set consists of positive/negative and training/test samples. We use all positive test samples as inliers and the first 5% of negative test samples as outliers in the "evaluation" set; we use positive training samples as inliers in the "model" set. Thus, the positive samples are treated as inliers and the negative samples are treated as outliers. We assign the evaluation set to $p_{\mathrm{tr}}(\boldsymbol{x})$ and the model set to $p_{\mathrm{te}}(\boldsymbol{x})$. Thus, a sample with a small importance value is likely an outlier.

In the evaluation of outlier detection performance, it is important to take into account both the *detection rate* (the amount of true outliers an outlier detection algorithm can find) and the *detection accuracy* (the amount of true inliers that an outlier detection algorithm misjudges as outliers). Since there is a trade-off between the detection rate and the detection accuracy, we adopt the *area under the ROC curve* (AUC) as our error metric.

The results are summarized in Tab.5.1, showing that GM-KLIEP and PM-KLIEP compare favorably with KLIEP.

Table 5.1: Mean AUC values (with their standard deviation in brackets) over 20 trials in the outlier detection experiments. If the performance of one of three methods is significantly different by the *t-test* at a significance level of 5%, we use '$\circ$' as the case where GM-KLIEP or PM-KLIEP outperforms KLIEP, '$+$' as the case where KLIEP or PM-KLIEP outperforms GM-KLIEP, and '$\star$' as the case where KLIEP or GM-KLIEP outperforms PM-KLIEP.

Datasets	KLIEP	GM-KLIEP	PM-KLIEP
banana	55.9 (5.0)	$^{\circ\star}$70.6 (2.3)	$^{\circ}$60.3 (1.2)
brestcancer	71.1 (8.8)	69.7(13.0)	65.0(11.8)
diabetes	$^{+\star}$63.0 (9.0)	53.1 (7.3)	55.6 (4.0)
flaresolar	57.5 (6.7)	60.1 (6.4)	59.4 (6.7)
german	58.8 (7.9)	56.2 (7.8)	56.9 (6.7)
heart	69.0(15.1)	73.1(15.6)	73.6(15.0)
image	55.1 (6.6)	$^{\circ}$69.8(14.3)	$^{\circ}$72.7 (6.5)
thyroid	57.8 (9.8)	$^{\circ}$78.0 (9.1)	$^{\circ}$76.9(14.8)
titanic	63.7 (9.1)	63.2 (2.1)	63.6 (2.4)
twonorm	70.1 (8.4)	70.6 (3.3)	$^{\circ+}$85.1 (1.5)
waveform	63.5 (8.0)	$^{\circ}$76.1 (2.7)	$^{\circ+}$81.6 (1.3)
Average	62.3 —	67.3 —	68.2 —

CHAPTER 6

NOISE ADAPTIVE OPTIMIZATION OF MATRIX INITIALIZATION

In this chapter, we formulate the frequency domain independent component analysis for pre-processing of speaker identification.

6.1 Introduction

Implementing frequency domain independent component analysis (FDICA)[28, 29, 30] has recently received much attention from the audio industry, c.f. [31]. This is due to the many potential source separation applications (e.g. speech enhancement, speaker separation), and the recent technological advancements that enable the implementation of FDICA on humanoid robots. However, since humanoid robots move throughout the world, the surrounding environment, source positions, and source mixtures are constantly changing. Thus, it is quite difficult to implement FDICA in humanoid robots for real-world usage.

Many effective approaches have been proposed for improving FDICA performance by exploiting: knowledge regarding room and sensor geometry [32], geometric information of sound sources [33, 34], and a sophisticated prior model of speech [35]. However, these approaches implicitly assume knowledge of the sound source geometry, the source type (point source, diffuse source, etc.), and are valid only in a specific surrounding environmental condition. In addition, since the cost function of FDICA is *non-convex* in nature, FDICA is not guaranteed to converge to the optimal solution, when the initial unmixing matrix is incorrectly chosen. Thus, unmixing matrix initialization is a key factor for implementing FDICA to humanoid robots.

A popular unmixing matrix initialization technique is the combination of *delay-and-sum (DS)* and *null* beamformers (NBF) [30, 36], which are known to be robust to the well-known FDICA permutation problem [30]. However, beamformer-based initialization heavily depends on the sound source geometry and the source mixture type. Thus, beamformer-based initialization itself is not suited for humanoid robot usage, without a reasonable estimator of the source geometry and the source types.

In this chapter, we propose a Noise Adaptive Optimization of Matrix Initialization (NAOMI). We assume a two source separation problem, where a point source, e.g., speech signal, is placed in front of a two microphone array, while a second *interfering* source should be separated and removed using FDICA. The interfering source is either another point source that is not located directly in front of the microphones (e.g., a speech signal that is not intended to be captured by the microphones) or a diffuse source (e.g., loud background music or airplane engine rumble). To estimate the type of interfering source, we first estimate its direction of arrival (DOA) at each frequency bin using *covariance fitting* [37], and then use the statistics of the estimated DOAs to classify the interfering source. The initial unmixing matrix is then selected based on the estimated source type. The effectiveness of the proposed method for speech denoising is evaluated via a source separation simulations in anechoic and reverberant rooms.

6.2 Problem formulation

In this section, we briefly explain FDICA and beamformer-based unmixing matrix initialization.

6.2.1 Frequency Domain Independent Component Analysis

The K observed signal by N microphones in natural environments can be modeled as convolutive mixtures:

$$x_j(n) = \sum_{i=1}^{K}\sum_{k=1}^{P} a_{ji}(k)s_i(n-k+1),\ (j = 1, \ldots, N), \tag{6.1}$$

where s_i is the signal from a source i, x_j is the observed signal at microphone j, and a_{ji} is the P-taps impulse response from a source i to a microphone j. In this chapter, we assume the number of observed signal and microphones are $N = M = 2$.

Converting the time-domain convolutive mixtures into the frequency domain by Short-Time Fourier Transform (STFT), the convolutive mixture can be expressed as

$$\boldsymbol{x}(f,\tau) = \mathrm{A}(f)\boldsymbol{s}(f,\tau), \tag{6.2}$$

where τ is the frame index. The observed signal vector $\boldsymbol{x}(f,\tau) \in \mathbb{C}^N$ and $\boldsymbol{s}(f,\tau) \in \mathbb{C}^K$ are

$$\begin{aligned} \boldsymbol{x}(f,\tau) &= [x_1(f,\tau), \cdots, x_N(f,\tau)]^\top, & (6.3)\\ \boldsymbol{s}(f,\tau) &= [s_1(f,\tau), \cdots, s_K(f,\tau)]^\top, & (6.4) \end{aligned}$$

where $^\top$ is the transpose of a matrix. The mixing matrix $\mathrm{A} \in \mathbb{C}^{N\times K}$ is

$$\begin{aligned} \mathrm{A}(f) &= \begin{bmatrix} a_{11}(f) & \cdots & a_{1K}(f) \\ \vdots & \ddots & \vdots \\ a_{N1}(f) & \cdots & a_{NK}(f) \end{bmatrix}, \\ &= [\boldsymbol{a}_1(f), \ldots, \boldsymbol{a}_K(f)], \end{aligned} \tag{6.5}$$

where $\boldsymbol{a}_i(f) = [a_{1i}(f), \ldots, a_{Ni}]^\top \in \mathbb{C}^N$.

The goal of FDICA is to estimate the unmixing matrix $\mathrm{W}(f) \in \mathbb{C}^{N\times N}$ that satisfies $\mathrm{W}(f)\mathrm{A}(f) = \mathrm{I}$, where I is the identity matrix. In this chapter, we employ the information theoretic principles to find an unmixing matrix [71] and estimate the unmixing matrix by following the iterative formula [28]:

$$\mathrm{W}^{l+1}(f) \ = \mathrm{W}^{l}(f) + \eta\big\{\mathrm{I} - E[\boldsymbol{\phi}(f,\tau)\mathbf{y}^*(f,\tau)]\mathrm{W}^{l}(f)\big\}, \tag{6.6}$$

where * is the complex conjugate of a matrix, $\mathrm{W}^l(f)$ is lth iteration of $\mathrm{W}(f)$, η is a step size parameter, $E[\boldsymbol{x}(f,\tau)]$ is the expectation of $\boldsymbol{x}(f,\tau)$ with respect to τ. The nonlinear function $\boldsymbol{\phi}(\cdot)$ is defined as

$$\boldsymbol{\phi}(f,\tau) = [\phi_1(f,\tau)\cdots\phi_K(f,\tau)]^\top, \tag{6.7}$$

$$\phi_k(f,\tau) = \mathrm{sgn}(\mathrm{Re}\{y_k(f,\tau)\}) + j\mathrm{sgn}(\mathrm{Im}\{y_k(f,\tau)\}), \tag{6.8}$$

where sgn$(\cdot)$ is sign function and Re$\{\cdot\}$ and Im$\{\cdot\}$ are real and imaginary part of a complex number, respectively.

The segregated signal vector $\mathbf{y}(f,\tau) = [y_1(f,\tau),\cdots,y_N(f,\tau)]^\top] \in \mathbb{C}^N$ can then be represented in matrix form as:

$$\mathbf{y}(f,\tau) = \mathrm{W}(f)\boldsymbol{x}(f,\tau). \tag{6.9}$$

Note that, since the cost function of FDICA is *non-convex*[71], $W^\infty(f)$ may not converge to the true solution when the initial unmixing matrix $\mathrm{W}^0(f)$ is set incorrectly. This implies that when source positions or source types are changed, we may need to re-initialize the unmixing matrix to obtain good separation results.

6.2.2 Beamformer based unmixing matrix initialization

In this chapter, we concentrate on the two source separation problem, i.e., $K = N = 2$, where one source is assumed to be a *point source* located in front of the array. In such a case, the possible combinations of sound source types are *point source* + *point source* or *point source* + *diffuse source*, where we define a *point source* as a speech signal located near the microphone array, while a *diffuse source* is defined as a widely spread source located far from the microphone array. In the following, we introduce two popular beamformer-based unmixing matrix initialization techniques for the *point source* + *point source* and *point source* + *diffuse source* cases, respectively. Note that, in this chapter, we focus only on beamformer-based matrix initialization techniques since they are known to be robust to the FDICA permutation problem [30].

6.2.2.1 Null beamformer based initialization

Null beamformer (NBF)-based initialization is often used for separating mixtures of two point sources, i.e., speech + speech, and is given by [30]:

$$\mathrm{W}^0(f) \;=\; \begin{bmatrix} 1 & -e^{i2\pi f d\sin(\theta_2)/V_c} \\ 1 & -e^{i2\pi f d\sin(\theta_1)/V_c} \end{bmatrix}, \tag{6.10}$$

where θ_1 and θ_2 are the DOAs of the point sources, d is the microphone distance, and V_c is the speed of sound. The first row of Eq.(6.10) cancels the signal from direction θ_2 and enhances the signal from θ_1, while the second row cancels the signal from direction θ_1 and enhances the signal from θ_2. By using the same θ_1 and θ_2 values to initialize the unmixing matrix for every frequency, NBF-based initialization is considered robust to the permutation problem. However, if the DOAs of point sources θ_1 and θ_2 are incorrectly set, $\mathrm{W}^\infty(f)$ might fail to converge to an acceptable solution. In addition, if the observed signal is *point source + diffuse source*, NBF-based initialization is not a reasonable choice, since the DOA of a diffuse source is usually not well defined.

6.2.2.2 delay sum + null beamformer based initialization

In the *point source + diffuse source* case, we can not estimate the DOA of the diffuse source, therefore, the following initial unmixing matrix [36] is used:

$$\mathrm{W}^0(f) \;=\; \begin{bmatrix} \frac{1}{2} & \frac{1}{2}e^{i2\pi f d\sin(\theta_1)/V_c} \\ 1 & -e^{i2\pi f d\sin(\theta_1)/V_c} \end{bmatrix}, \tag{6.11}$$

where the first row of Eq.(6.11) enhances the signal from direction θ_1, while the second row cancels the signal from direction θ_1, i.e., enhances the signal at all other directions. Hereinafter, we refer to this method as *DS-NBF* initialization. With this technique, we do not need the DOA of diffuse source, thus, DS-NBF initialization is well suited for the *point source + diffuse source* separation problem. However, it may not work well for the separation of two point sources, necessitating an automatic method for classifying the type of interfering source.

6.3 Noise Adaptive Optimization of Matrix Initialization

In this section, we propose the *noise adaptive optimization of matrix initialization* (NAOMI) algorithm, which consists of two parts: estimating the geometry of the interfering sound source, and classifying the type of interfering source.

6.3.1 Estimating the geometry of the interfering sound source

Let $\boldsymbol{d}_f(\theta) \in \mathbb{C}^N$ denote the response of the array to a plane wave of unit amplitude arriving from direction θ at frequency f; we will refer to $\boldsymbol{d}_f(\cdot)$ as the *array manifold* or *steering vector*. If we assume that narrowband sources $s_1(f,\tau)$ and $s_2(f,\tau)$ are impinging on the array at angles θ_1 and θ_2, respectively, then the vector array input $\boldsymbol{x}(f,\tau) \in \mathbb{C}^N$ can be represented as

$$\begin{aligned}\boldsymbol{x}(f,\tau) &= s_1(f,\tau)\boldsymbol{a}_1(f) + s_2(f,\tau)\boldsymbol{a}_2(f,\tau),\\ &= s_1(f,\tau)\boldsymbol{d}_f(\theta_1) + s_2(f,\tau)\boldsymbol{d}_f(\theta_2), \qquad (6.12)\end{aligned}$$

where we assume that $\boldsymbol{a}_1(f,\tau) = \boldsymbol{d}_f(\theta_1)$ and $\boldsymbol{a}_2(f,\tau) = \boldsymbol{d}_f(\theta_2)$, respectively.

If $s_1(f,\tau)$ and $s_2(f,\tau)$ are independent, zero-mean signals, i.e., $E[s_1(f,\tau)s_2^*(f,\tau)] = E[s_1(f,\tau)]E[s_2^*(f,\tau)] = 0$, we can write the covariance matrix of the observed signals as

$$\begin{aligned}\mathrm{R}_{xx}(f) &= E[s_1^2(f,\tau)\boldsymbol{d}_f(\theta_1)\boldsymbol{d}_f(\theta_1)^*] + E[s_2^2(f,\tau)\boldsymbol{d}_f(\theta_2)\boldsymbol{d}_f(\theta_2)^*], \qquad (6.13)\\ &= \sigma_1^2(f)\boldsymbol{d}_f(\theta_1)\boldsymbol{d}_f(\theta_1)^* + \sigma_2^2(f)\boldsymbol{d}_f(\theta_2)\boldsymbol{d}_f(\theta_2)^*, \qquad (6.14)\\ &= \mathrm{R}_{x_1x_1}(f) + \mathrm{R}_{x_2x_2}(f), \qquad (6.15)\end{aligned}$$

where $\sigma_1^2(f) = E[s_1^2(f,\tau)]$, $\sigma_2^2(f) = E[s_2^2(f,\tau)]$, and $\mathrm{R}_{x_ix_i} = \sigma_i^2\boldsymbol{d}_f(\theta_i)\boldsymbol{d}_f(\theta_i)^*$. If the signal power and array manifold $\boldsymbol{d}_f(\theta_1)$ corresponding to source $s_1(f,\tau)$ are known, the covariance matrix of the unknown source signal $\mathrm{R}_{x_2x_2}(f)$ can be written as

$$\begin{aligned}\mathrm{R}_{x_2x_2}(f) &= \mathrm{R}_{xx}(f) - \mathrm{R}_{x_1x_1}(f), \qquad (6.16)\\ &= \sigma_2^2(f)\boldsymbol{d}_f(\theta_2)\boldsymbol{d}_f(\theta_2)^*. \qquad (6.17)\end{aligned}$$

Here, we want to estimate θ_2, which is equivalent to estimating the array manifold $\boldsymbol{d}_f(\theta_2)$ by solving the following eigenvalue problem,

$$\begin{aligned} \max \quad & \boldsymbol{w}^*(f)\mathrm{R}_{x_2x_2}(f)\boldsymbol{w}(f), \\ \text{s.t.} \quad & \boldsymbol{w}^*(f)\boldsymbol{w}(f) = 1. \end{aligned} \tag{6.18}$$

Since $\mathrm{R}_{x_2x_2}(f)$ is spanned by the array manifold $\boldsymbol{d}_f(\theta_2)$, it is clear that we have $\boldsymbol{w}(f) = \boldsymbol{d}_f(\theta_2)$. Therefore, we can estimate the *interfering* point source DOA $\theta_2(f)$ from

$$\widehat{\theta_2}(f) = \arg\max_{\theta} |\boldsymbol{w}(f)^*\boldsymbol{d}_f(\theta)|. \tag{6.19}$$

In practice, it is difficult to estimate $\widehat{\theta_2}(f)$ at low frequencies due to the small time difference between the observed signals at the two microphones. Additionally, spatial aliasing occurs if $f > \frac{V_c}{2d}$. Therefore, we use information only from a certain range of frequencies to estimate $\widehat{\theta_2}$ as

$$\widehat{\theta_2} = \frac{1}{f_e - f_s}\sum_{f=f_s}^{f_e}\widehat{\theta_2}(f), \tag{6.20}$$

where f_s is the low frequency cutoff and the high frequency cutoff is $f_e \leq \frac{V_c}{2d}$.

In the above derivation of $\widehat{\theta_2}$, we assume that $\mathrm{R}_{x_2x_2}(f)$ is known. However, since $\mathrm{R}_{x_2x_2}(f)$ is not available in practice, we need to estimate $\mathrm{R}_{x_2x_2}(f)$ from the observed signals.

Since $\mathrm{R}_{x_2x_2}(f)$ can be written as

$$\mathrm{R}_{x_2x_2}(f) = \mathrm{R}_{xx}(f) - \sigma_1^2(f)\boldsymbol{d}_f(\theta_1)\boldsymbol{d}_f^*(\theta_1), \tag{6.21}$$

and we want to remove the $\boldsymbol{d}_f(\theta_1)$ component from $\mathrm{R}_{xx}(f)$ as much as possible, the estimation problem of $\mathrm{R}_{x_2x_2}(f)$ can be formulated as

$$\begin{aligned} \max \quad & \sigma_1^2 \\ \text{s.t.} \quad & \mathrm{R}_{xx}(f) - \sigma_1^2(f)\boldsymbol{d}_f(\theta_1)\boldsymbol{d}_f^*(\theta_1) \succeq 0, \end{aligned} \tag{6.22}$$

where $\succeq 0$ means that the matrix to the right of the inequality is positive semidefinite. This formulation is known as *covariance fitting*, and is often used for robust beamformer estimation [37].

Multiplying both sides of Eq.(6.22) by $\mathrm{R}_{xx}^{-1/2}$, we have

$$\mathrm{I} - \sigma_1^2(f)\mathrm{R}_{xx}^{-1/2}(f)\boldsymbol{d}_f(\theta_1)\boldsymbol{d}_f^H(\theta_1)\mathrm{R}_{xx}^{-*/2}(f) \succeq 0. \tag{6.23}$$

Since Eq.(6.23) should be positive semidefinite and $\mathrm{R}_{xx}^{-1/2}\boldsymbol{d}_f(\theta_1)\boldsymbol{d}_f^H(\theta_1)\mathrm{R}_{xx}^{-*/2}$ is a rank one matrix, we have

$$\sigma_1^2(f) \leq \frac{1}{\boldsymbol{d}_f(\theta_1)\mathrm{R}_{xx}^{-1}(f)\boldsymbol{d}_f^*(\theta_1)}. \tag{6.24}$$

Thus, the optimal $\sigma_1^2(f)$ is given by

$$\widehat{\sigma}_1^2(f) = \frac{1}{\boldsymbol{d}_f^*(\theta_1)\mathrm{R}_{xx}^{-1}(f)\boldsymbol{d}_f(\theta_1)}, \tag{6.25}$$

and the estimate of $\mathrm{R}_{x_2x_2}(f)$ is

$$\widehat{\mathrm{R}}_{x_2x_2}(f) = \mathrm{R}_{xx}(f) - \widehat{\sigma}_1^2(f)\boldsymbol{d}_f(\theta_1)\boldsymbol{d}_f^*(\theta_1). \tag{6.26}$$

It is important to note that the computational cost in the proposed method is very small. In fact, the entire computational cost of the proposed method is almost identical to that of a single update iteration of Eq.(6.6) for all frequencies. Moreover, the proposed method provides a good initial unmixing matrix, which is expected to decrease the number of iterations necessary for FDICA to converge. Therefore, we can reduce the computational complexity without deteriorating the separation performance.

6.3.2 Source type classification

In the two point source mixture case, the estimated DOA of the interfering source at each frequency is close to the true DOA. On the other hand, since a diffuse source consists of the reverberation or mixture of many sound sources, the estimated DOAs of diffusion source tend to be random across frequencies. Thus, we use the variance of estimated DOAs to decide whether the source mixture type is *point source* + *point source* or *point source* + *diffuse source*. The variance of estimated DOAs is given by

$$\widehat{\sigma}^2 = \frac{1}{f_e - f_s} \sum_{f=f_s}^{f_e} (\theta_2(f) - \widehat{\theta_2})^2, \tag{6.27}$$

where f_s and f_e are the low and high cutoff frequencies, respectively. Finally, we select the initial unmixing matrix using NBF as follows: if $\widehat{\sigma}^2 < \rho$ (two point source case) or DS-NBF if $\widehat{\sigma}^2 \geq \rho$ (point source + diffuse source). In addition, if the estimated DOA $\widehat{\theta_2}$ is close to θ_1, the separation performance is degraded. Therefore, we heuristically choose DS-NBF beamformer, when $|\widehat{\theta_2} - \theta_1| < \epsilon$, where ϵ is an arbitrary threshold parameter. A pseudo code for the proposed algorithm is described in Algorithm 1.

Algorithm 1 Noise Adaptive Optimization of Matrix Initialization

```
$\theta_1 = 0$;
compute $\widehat{\theta_2}$;
for f = 0; f $\leq \frac{FFTSize}{2} + 1$; f++ do
    if $\widehat{\sigma}^2 < \rho$ or $|\widehat{\theta_2} - \theta_1| \geq \epsilon$ then
        $W^0(f) = \begin{bmatrix} 1 & -e^{i2\pi f d \sin(\widehat{\theta_2})/V_c} \\ 1 & -e^{i2\pi f d \sin(\theta_1)/V_c} \end{bmatrix}$;
    else
        $W^0(f) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}e^{i2\pi f d \sin(\theta_1)/V_c} \\ 1 & -e^{i2\pi f d \sin(\theta_1)/V_c} \end{bmatrix}$;
    end if
    $W^0(f) = (W^0(f) R_{xx}(f) {W^0(f)}^H)^{-\frac{1}{2}} W^0(f)$;
    for l = 0; l < L; l++ do
        $W^{l+1}(f) = W^l(f) + \eta\{I - E[\boldsymbol{\phi}(f,\tau)\mathbf{y}^*(f,\tau)]W^l(f)\}$;
    end for
end for
```

6.4 Source Separation Experiments

We assess the effectiveness of the proposed method by performing a simulation experiment for blind source separation with the proposed method, and then evaluate the performance of the proposed method in speaker identification. As previously discussed, we assume the two source separation problem, where one of the sources is a point source located in front of microphones, i.e., $\theta_1 = 0$.

6.4.1 *point source* + *point source* separation in an anechoic chamber

In this experiments, we use speech signals from 2 male and 4 female speakers, recorded in anechoic chamber. By choosing one speaker as the point source located in front of the array, and a different speaker as the interfering source, 30 speaker combinations are used in the experiment. The interfering point source is placed at either -90, -45, 45, or 90 degrees respectively, while the target point source is placed at 0 degrees. Figure 6.1 shows the placement of sound sources and microphones, while the detailed recording conditions are described in Table 6.1.

We compare the proposed method to a NBF with nulls at 0 and 90 degrees, a NBF with nulls at 0 and -90 degrees, and a DS-NBF combination consisting of a DS at 0 degrees and a NBF at 0 degrees [30, 36]. Note that, to evaluate the robustness to permutation of the separated sources, we do not explicitly solve the permutation problem via post-processing methods. For evaluation, we compute the average noise reduction rate (NRR) [30] for 30 combinations of speakers at each unknown point source angle. Figure 6.2 shows the NRR as a function of the FDICA iteration number in Eq.(6.6). As can be seen, the proposed method gives high NRR for every interfering point source position, while the other initialization methods work only if the DOA of the interfering point source and the DOA used to initialize the unmixing matrix are identical. In Table 6.2, we show the results of DOA estimation using covariance fitting, and the estimated DOA is likely to be biased, since we estimate the DOA from mixture of sound sources. However, we have observed that highly accurate DOA estimates are not critically important in the FDICA initialization phase, so we conclude that the proposed method outperforms conventional methods in an anechoic chamber.

Table 6.1: Recording conditions in anechoic chamber

Sampling rate	8 [kHz]
FFT Size	1024 [sample]
FFT Shift	512 [sample]
Signal Len.	3 [s]
Microphone	OMNI, SHURE SM93
Num. of Mic.	2
Interval of Mic.	5.4 [cm]

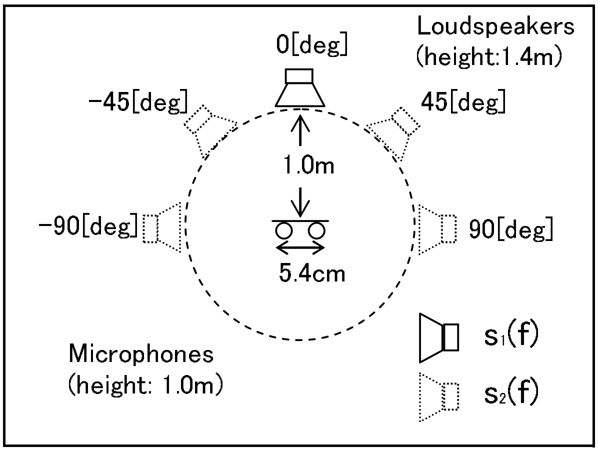


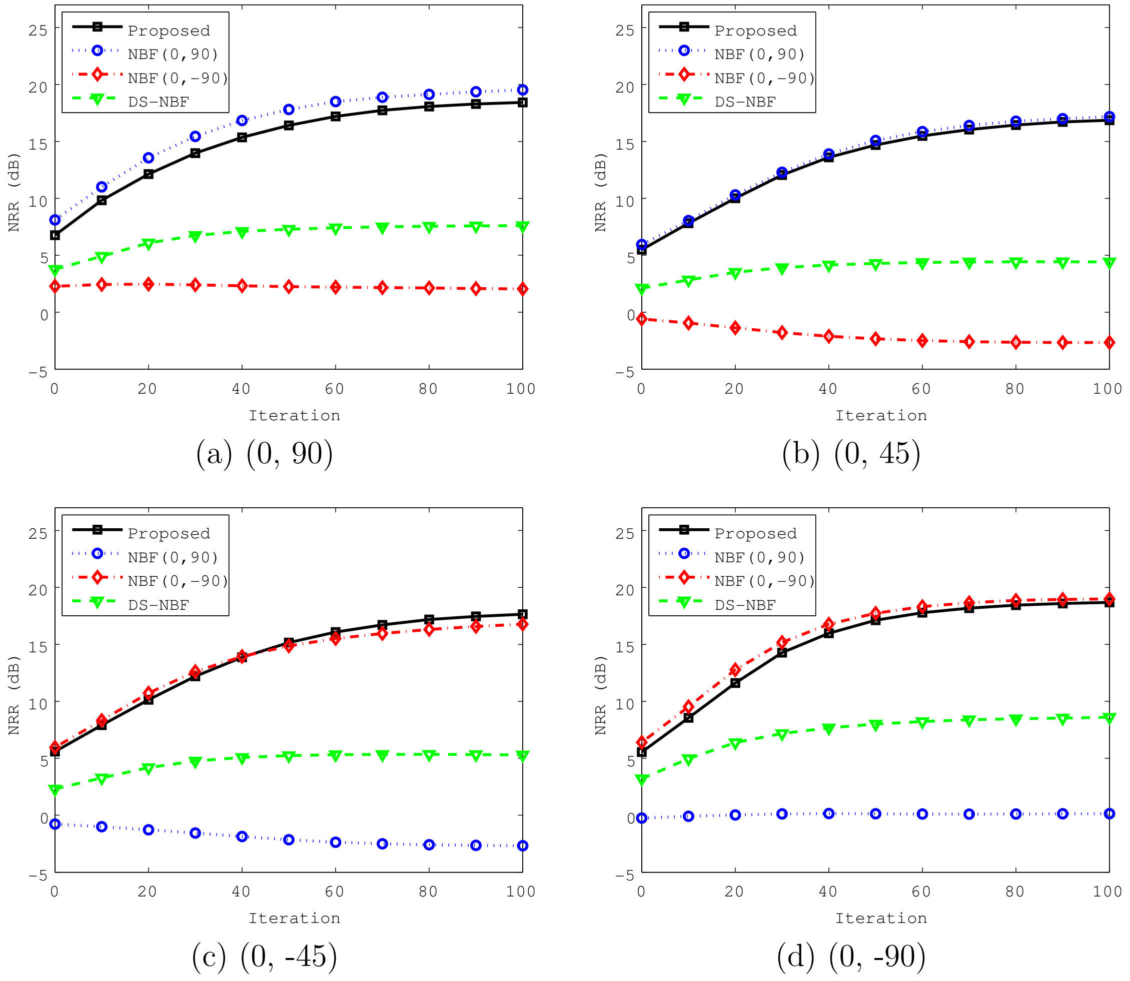
Figure 6.1: Recording Environment in anechoic chamber.

6.4.2 *point source* + *point source* separation in reverberant room

In this experiment, we use speech signals from 2 male and 4 female speakers, which are recorded in a reverberant room with a 400ms reverberation time. If each speaker can be either the target or interfering source, we have 30 possible combinations, where the interfering point source is placed as −90, −45, 45, or 90 degrees, respectively. Figure

Table 6.2: DOA result when the unknown source is directional signal. The mean and standard deviation of $\widehat{\theta_2}$ are from 30 combinations of speakers.

θ_2	90.0	45.0	-45.0	-90.0
$\widehat{\theta_2}$	50.2 ± 7.4	24.2 ± 5.0	-30.5 ± 9.2	-53.2 ± 12.5



(a) (0, 90)

(b) (0, 45)

(c) (0, -45)

(d) (0, -90)

Figure 6.2: Noise reduction rate as a function of FDICA iteration for the two point source case in an anechoic chamber. The DOA of true sources are shown in the bracket (θ_1, θ_2).

6.3 shows the placement of sound sources and microphones, while the microphone distance and FFT size are the same as in the anechoic chamber case.

We again compare the proposed method to a NBF with nulls at 0 and 90 degrees, a NBF with nulls at 0 and -90 degrees, and DS-NBF as shown in Figure 6.4. As can be seen, the proposed method provides good NRR for every point source position, even in the presence of heavy reverberation. In Table 6.3, it is observed that the DOA estimates of the proposed method in highly reverberant environment are similar to those reported in Table 6.2 for the anechoic case, meaning the method should work well in real-world conditions.

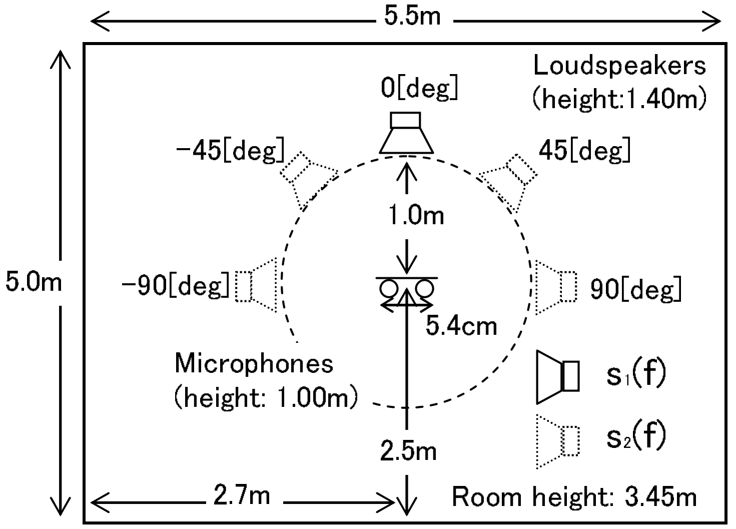


Figure 6.3: Recording Environment in reverberant room.

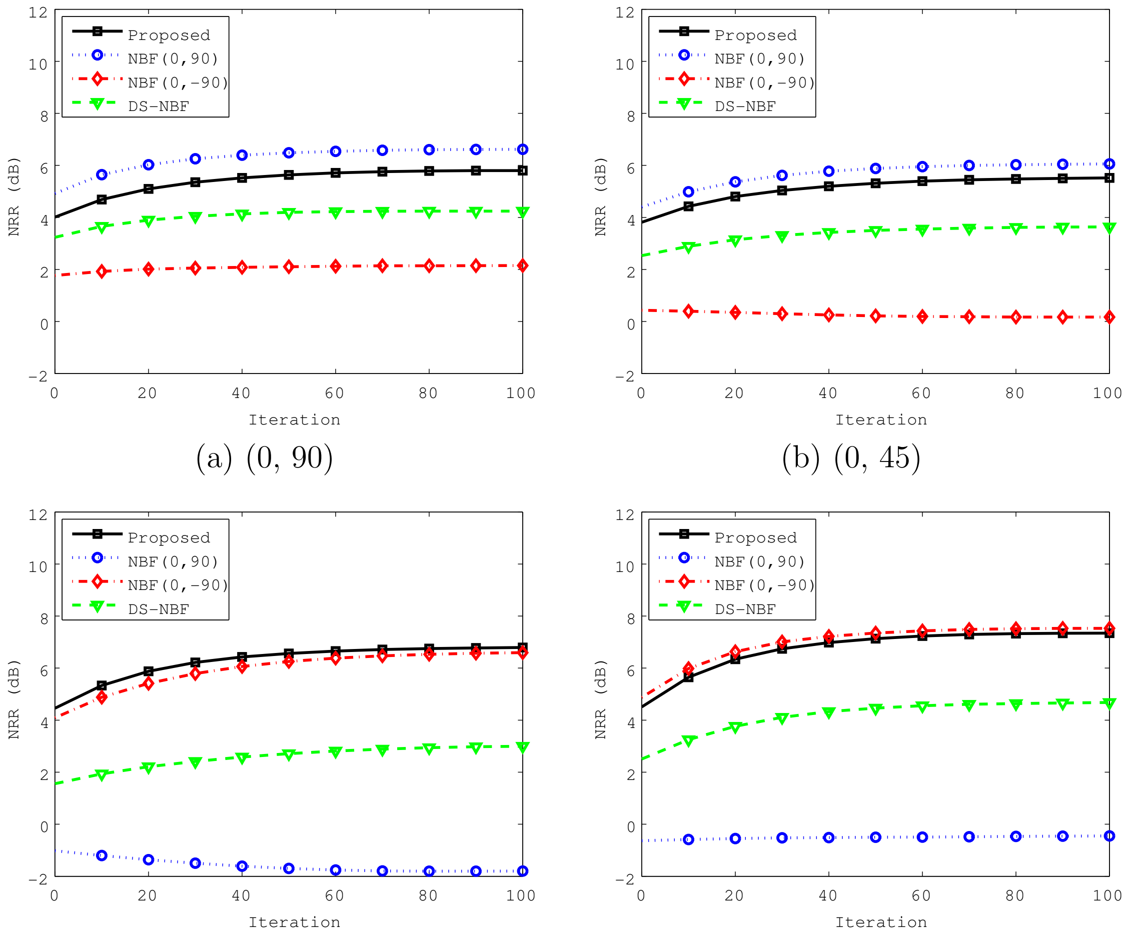
Table 6.3: DOA result when the unknown source is directional signal. The mean and standard deviation of $\widehat{\theta_2}$ are from 30 combinations of speakers.

θ_2	90.0	45.0	-45.0	-90.0
$\widehat{\theta_2}$	41.8 ± 14.2	28.1 ± 6.8	-24.1 ± 11.9	-41.2 ± 17.3

6.4.3 *point source* + *diffuse source* separation in reverberant room

In this experiments, we use speech signals from 2 male and 4 female speakers, where the target point source is located in front of the microphone array. For the interfering diffuse source, we use the ambient sound of a *shinkansen* (bullet train). Thus, six total sound mixtures are used in this experiment. The microphone distance and FFT size are same as those given in Table 6.1, and a diagram of the recording setup is shown in Figure 6.5.

We compare the proposed method to a NBF with nulls at 0 and 90 degrees, a NBF with nulls at 0 and -90 degrees, and a DS-NBF. For evaluation, we compute the average of the NRR for six speaker combinations. Figure 6.6 shows the NRR



(a) (0, 90)

(b) (0, 45)

(c) (0, -45)

(d) (0, -90)

Figure 6.4: Noise reduction rate as a function of FDICA iteration for the two point source case in a reverberant room. The DOA of true sources are shown in the bracket (θ_1, θ_2).

as a function of the FDICA iteration number given in Eq.(6.6). As can be seen, the proposed method always chooses the DS-NBF and gives good NRR performance, while NBF-based initialization fails to converge to a reasonable solution.

6.4.4 Environmental adaptation in reverberant room

In this experiment, we evaluate the proposed system in a changing environment. The total duration of the signal used in this experiment is 30s, where the source signal consists of three parts: speech (0 deg) + speech (-45 deg) (0s - 10s), speech (0 deg) + speech (45 deg) (10s - 20s), and speech (0 deg) + ambient noise (20s - 30s). We

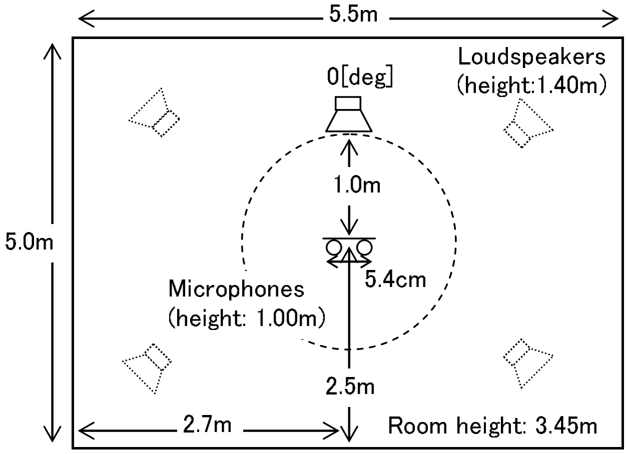


Figure 6.5: Recording Environment in diffuse interfering noise case.

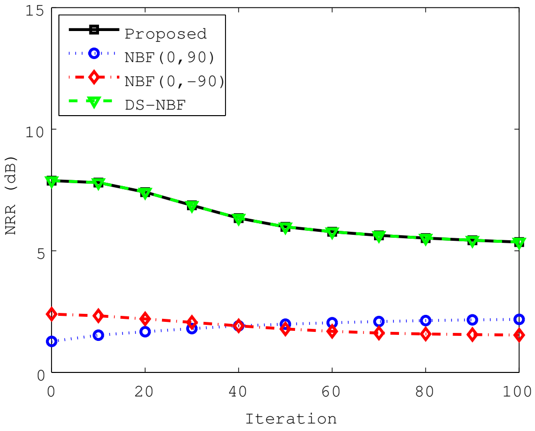


Figure 6.6: Source separation result in interfering diffuse source case.

use a two second block of the signal for estimating the unmixing matrix, and then filter the next non-overlapping two second block using the unmixing matrix estimated from the previous block. The parameters ρ, η and ϵ are experimentally set 0.7, 0.01 and 20, respectively, and the number of FDICA iterations is fixed to 100 for the two point source separation case and 10 for the point source + diffuse source case.

We compare the proposed method to a NBF with nulls at 0 and 45 degrees, a NBF with nulls at 0 and -45 degrees, and a DS-NBF. In this experiment, we fix the number of FDICA iterations for these conventional methods to 100. The NRR as a function of time is compared among the four methods in Figure 6.7. As can be seen, the proposed method gives high NRR even if the source mixture types are changed (with a lag equal to the block size), while the other initialization methods work only if the DOA of the interfering point source and the DOA used to initialize the unmixing matrix are identical.

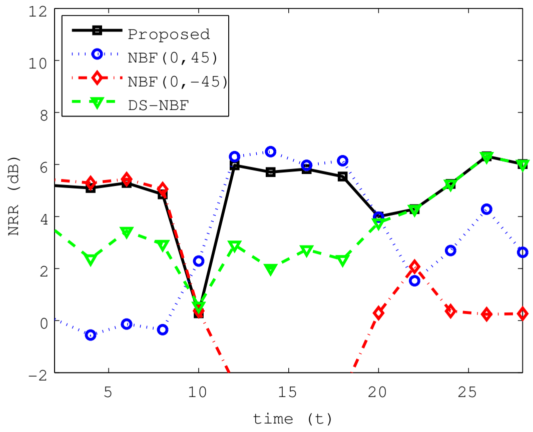


Figure 6.7: Source separation results in varying environmental conditions.

6.4.5 Speaker Identification Experiments

In this experiment, we evaluate speaker identification with the proposed source separation algorithm in a changing environment, where we use the kernel logistic regression (KLR) based speaker identification with sequence kernel (see Chapter 2 for detail). We use speech signals from 10 male speakers recorded in anechoic chamber. By choosing one speaker as the point source located in front of the array, and a different speaker as the interfering source. The total duration of the signal used in this experiment is 22s, where the source signal consists of three parts: speech (0 deg) + speech (-45 deg) (0s - 10s), speech (0 deg) + speech (45 deg) (10s - 22s). We choose the kernel width for sequence kernel as 1.0 and the regularization parameter 0.01, where these parameters are selected by 5-fold cross validation (CV).

We compare the proposed method to a NBF based source separation system with nulls at 0 and 45 degrees [30, 36], and no source separation system. Table 6.4 shows the mean speaker identification rate over the experiments. As can be seen, the proposed method gives high identification rate compare to the other methods.

Table 6.4: Speaker identification under environmental change.

	NAOMI	NBF (0,45)	No ICA
Identification ratio	58.2	54.9	29.6

CHAPTER 7

CONCLUSION

This dissertation was devoted to propose the speaker identification methods for humanoid robots. Especially, we focused on four major issues in speaker identification in this dissertation.

First, since the humanoid robots should identify the speaker in real-time with high identification rates, thus, we developed the kernel-based real-time speaker identification system in chapter 3. In this chapter, we gave approximation schemes of the mean operator sequence kernel (MOSK) based on pre-images in RKHSs for real-time speaker identification purpose. Through numerical experiments, the proposed methods were shown to be useful in text-independent speaker identification when they are combined with kernel logistic regression (KLR) and cross validation (CV). In addition, we implemented the proposed algorithm with Virtual Studio Technology (VST) plugin.

Second, the speech features vary over time due to session dependent variation, the recording environment change, and physical conditions/emotions. To deal with the problem, we proposed a novel semi-supervised speaker identification method that can alleviate the influence of non-stationarity such as session dependent variation, the recording environment change, and physical conditions/emotions in Chapter 4. Under such non-stationary environment, standard machine learning techniques such as KLR and CV or Gaussian mixture models (GMM) and CV do not work properly due to changing environment.

Our assumption was that voice quality variants follow the *covariate shift* model—the voice feature distribution changes between the training and test phases, but the conditional distribution of the speaker index given voice features is unchanged. Under this covariate shift model, we employed the importance weighted KLR (IWKLR) method, where the importance weights are estimated by using the Kullback-Leibler importance estimation procedure (KLIEP) with likelihood CV (LCV). By combining IWKLR and KLIEP, classification accuracy under covariate shift is highly improved. Moreover, the kernel width and the regularization parameter of IWKLR are tuned based on importance weighted CV (IWCV), which is guaranteed to be almost unbiased even under covariate shift. To verify the validity of our approach, we conducted text-independent/dependent speaker identification simulations and experimentally found that the covariate shift formulation with IWKLR, IWCV, and KLIEP is a promising approach.

Third, the humanoid robots are desired to automatically detect the unknown speakers, and the unknown speakers information should be automatically included into the dictionary. Indeed, the speaker detection task can be formulated as the outlier detection problem (i.e., outliers can be the unknown speakers), where it can be solved through the comparison between the log likelihoods of the unknown speaker and the speakers. Thus, to improve the estimation accuracy of the log likelihoods is an important issue to have better speaker detection performance. To deal with the problem, in Chapter 5, we proposed a new importance estimation method using the Gaussian Mixture Model (GMM) and mixture of probabilistic principal component analyzers (PPCAs). Optimization of the proposed algorithm, GM-KLIEP and PM-KLIEP, can be efficiently carried out by the EM algorithm. The usefulness of the proposed approach was evaluated through experiments.

Forth, the humanoid robots move throughout the world, and the surrounding environment, source positions, and source mixtures are constantly changing. In addition, the speech overlaps are frequently occurred during conversation. Thus, the source separation techniques are useful for improving the speaker identification performance. To deal with those problems, in Chapter 6, we proposed the noise adaptive optimization of matrix initialization (NAOMI) for frequency domain independent component analysis (FDICA), and used it for pre-processing of speaker identification. The experimental results showed the effectiveness of the proposed method in a realistic environment when compared with conventional beamformer-based initialization methods.

7.1 Future works

In this section we give some suggestions for future work.

7.1.1 Mean Operator Sequence kernel based speaker identification

We will implement the kernel-based speaker identification system for small devices such as DSP, and we will use it for robotics, conference systems, or human interfaces. Moreover, using the developed speaker identification system for multi-modal person identification is also an interesting topic. The general person identification systems mainly use the image processing to identify the person. However, when the image input is not reliable (e.g., in night), using the speech based speaker identification could be the reasonable solution.

7.1.2 Semi-supervised speaker identification

There are several remaining issues to be pursued for further improving the identification performance in the semi-supervised framework. For example, the IWCV method appeared to be rather unstable in experiments when the degree of distribution shift is very high. In such cases, further regularization of the IWCV method is expected to be useful, e.g., following the line of the paper [72]. Another challenging issue is to weaken the covariate shift assumption. The covariate shift model where only the input distribution changes could be rather restrictive in practice—the conditional distribution may also change in speaker identification tasks. In such cases, however, it is not possible to learn well in principle in the semi-supervised setup since there is no information on the test output distribution. To cope with this situation, we need to change the problem setup from semi-supervised learning to *transfer learning* where a small number of test output samples are also available. We expect that a similar weighting approach is still useful even in the transfer learning scenarios.

7.1.3 Direct Importance Estimation

We presented GMM and PPCA based direct importance estimation methods for outlier detection, and verified the performance of the proposed methods based on the outlier detection problems. The future work includes the evaluation of the unknown speaker detection problem. In addition, since the GM-KLIEP and PM-KLIEP employ the EM-algorithm, the expansion of the GM-KLIEP and PM-KLIEP to online EM-algorithm is an interesting future work. Moreover, in speech processing area, there are many possible applications of proposed methods such as speaker identification/verification and voice activity detection (VAD).

7.1.4 Noise Adaptive Unmixing Matrix Initialization

There are several remaining issues to be pursued for further improving the source separation performance. For example, in this paper, we assumed that the number of sources is two, however, there may exist more than two sources in the real world. Thus, we will work for the source separation problem with more than three sound sources in future. Also, implementing the proposed system for as the interface of speech recognition or speaker identification is the future work.

REFERENCES

[1] J. Mariethoz and S. Bengio, "A kernel trick for sequences applied to text-independent speaker verification systems," *Pattern Recognition*, vol. 40, no. 8, pp. 2315–2324, 2007.

[2] W. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of the IEEE International Conference on Audio Speech and Signal Processing*, Orland, Florida, USA, 2002, pp. 161–164.

[3] B. Schölkopf and A. J. Smola, *Learning with Kernels.* Cambridge, MA: MIT Press, 2002.

[4] S. Furui, "Comparison of speaker recognition methods using statistical features and dynamic features," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 29, no. 3, pp. 342–350, 1986.

[5] T. Matsui and K. Aikawa, "Robust model for speaker verification against session-dependent utterance variation," *IEICE Transactions on Information and Systems*, vol. E86-D, no. 4, pp. 712–718, 2003.

[6] T. Matsui and K. Tanabe, "Comparative study of speaker identification methods: dPLRM, SVM, and GMM," *IEICE Transactions on Information and Systems*, vol. E89-D, no. 3, pp. 1066–1073, 2006.

[7] S. Furui, "Cepstral analysis technique for automatic speaker verification," *Journal of Acoustical Society of America*, vol. 55, pp. 1204–1312, June, 1974.

[8] R. S. Sutton and G. A. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 1998.

[9] H. Hachiya, T. Akiyama, M. Sugiyama, and J. Peters, "Adaptive importance sampling with automatic model selection in value function approximation," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI2008)*, Chicago, USA, 2008, pp. 1351–1356.

[10] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach.* Cambridge, MA: MIT Press, 1998.

[11] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Interesting structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[12] S. Bickel and T. Scheffer, "Dirichlet-enhanced spam filtering based on biased samples," in *Advances in Neural Information Processing Systems.* Cambridge, MA: MIT Press, 2007, pp. 161–168.

[13] J. Jing and Z. ChengXiang, "Instance weighting for domain adaptation in NLP," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics.* Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 264–271.

[14] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[15] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *Journal of Machine Learning Research*, vol. 8, pp. 985–1005, 2007.

[16] J. J. Heckman, "Sample selection bias as a specification error," *Econometrica*, vol. 47, no. 1, pp. 153–162, 1979.

[17] D. A. Chon, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[18] V. V. Fedorov, *Theory of Optimal Experiments.* New York: Academic Press, 1972.

[19] D. P. Wiens, "Robust weights and designs for biased regression models: Least squares and generalized M-estimation," *Journal of Statistical Planning and Inference*, vol. 83, no. 2, pp. 395–412, 2000.

[20] T. Kanamori and H. Shimodaira, "Active learning algorithm using the maximum weighted log-likelihood estimator," *Journal of Statistical Planning and Inference*, vol. 116, no. 1, pp. 149–162, 2003.

[21] M. Sugiyama, "Active learning in approximately linear regression based on conditional expectation of generalization error," *Journal of Machine Learning Research*, vol. 7, pp. 141–166, 2006.

[22] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning.* Cambridge, MA: MIT Press, 2008.

[23] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[24] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Advances in Neural Information Processing Systems.* Cambridge, MA: MIT Press, 2008, pp. 1433–1440.

[25] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, “Inlier-based outlier detection via direct density ratio estimation,” in *Proceedings of IEEE International Conference on Data Mining (ICDM2008)*, Pisa, Italy, Dec. 15–19 2008, pp. 223–232.

[26] Y. Kawahara and M. Sugiyama, “Change-point detection in time-series data by direct density-ratio estimation,” in *Proceedings of 2009 SIAM International Conference on Data Mining (SDM2009)*, Sparks, Nevada, USA, Apr. 30–May 2 2009, pp. 389–400.

[27] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, “Direct importance estimation for covariate shift adaptation,” *Annals of the Institute of Statistical Mathematics*, vol. 60, no. 4, pp. 699–746, 2008.

[28] P. Smaragdis, “Blind separation of convolved mixtures in the frequency domain,” *Neurocomputing*, vol. 22, no. 1-3, pp. 21–34, 1998.

[29] S. Ikeda and N. Murata, “A method of ica in time-frequency domain,” in *Proceedings of International Workshop on Independent Component Analysis and Blind Signal Separation*, Aussions, France, 1999, pp. 365–371.

[30] H. Saruwatari, S. Kurita, K. Takeda, F. Itakura, T. Nishikawa, and K. Shikano, “Blind source separation combining independent component analysis and beamforming,” *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 11, pp. 1135–1146, 2003.

[31] Y. Mori, H. Saruwatari, T. Takatani, S. Ukai, K. Shikano, T. Hiekata, Y. Ikeda, H. Hashimoto, and T. Morita, “Blind separation of acoustic signals combining simo-model-based independent component analysis and binary masking,” *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–17, 2006.

[32] H. Sawada, R. Mukai, S. Araki, and S. Makino, *Frequency domain blind source separation.* Springer, 2005.

[33] L. Parra and C. Alvino, "Geometric source separation: merging convolutive source separation with geometric beamforming," in *Proceedings of the IEEE Signal Processing Society Workshop*, 2001, pp. 273–282.

[34] G. W. Taylor, M. L. Seltzer, and A. Acero, "Maximum a posteriori ica:applying prior knowledge to the separation of acoustic sources," in *Proceedings of the IEEE International Conference on Audio Speech and Signal Processing*, Las Vegas, Nevada, 2008, pp. 1821–1824.

[35] H. Attias, *Source separation with a sensor array using graphical models and subband filtering.* Cambridge, MA: MIT Press, 2003.

[36] Y. Takahashi, T. Takatani, H. Saruwatari, and K. Shikano, "Blind spatial subtraction array with independent component analysis for hands-free speech recognition," in *Proceedings of the IEEE International Workshop on Acoustic Echo and Noise Control*, Paris, France, 2006.

[37] J. Li, P. Stoica, and Z. Wang, "On robust capon beamforming and diagonal loading," *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1702–1715, 2003.

[38] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[39] K. Tanabe, "Penalized logistic regression machines: New methods for statistical prediction 1," Institute of Statistical Mathematics, Tech. Rep. 143, 2001.

[40] O. Birkenes, "A framework for speech recognition using logistic regression," Ph.D. dissertation, Norwegian University of Science and Technology, 2007.

[41] B. Schölkopf and A. J. Smola, *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

[42] M. Yamada, M. Sugiyama, and T. Matsui, "Semi-supervised speaker identification under covariate shift," *Signal Processing*, vol. 90, no. 8, pp. 2353–2361, 2010.

[43] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, 2000.

[44] W. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of the IEEE International Conference on Audio Speech and Signal Processing*, Orland, Florida, USA, 2002, pp. 161–164.

[45] J. Mariethoz and S. Bengio, "A kernel trick for sequences applied to text-independent speaker verification systems," *Pattern Recognition*, vol. 40, no. 8, pp. 2315–2324, 2007.

[46] S. Furui, "Comparison of speaker recognition methods using statistical features and dynamic features," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 29, no. 3, pp. 342–350, 1986.

[47] R. S. Sutton and G. A. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 1998.

[48] H. Hachiya, T. Akiyama, M. Sugiyama, and J. Peters, "Adaptive importance sampling for value function approximation in off-policy reinforcement learning," *Neural Networks*, vol. 23, no. 1, pp. 44–59, 2010.

[49] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach.* Cambridge, MA: MIT Press, 1998.

[50] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Interesting structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[51] S. Bickel and T. Scheffer, *Dirichlet-enhanced spam filtering based on biased samples.* Cambridge, MA: MIT Press, 2007.

[52] J. Jing and Z. ChengXiang, "Instance weighting for domain adaptation in nlp," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics.* Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 264–271.

[53] Y. Tsuboi, H. Kashima, S. Hido, S. Bickel, and M. Sugiyama, "Direct density ratio estimation for large-scale covariate shift adaptation," *IPSJ Journal*, vol. 50, no. 4, pp. 1–19, 2009.

[54] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[55] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *Journal of Machine Learning Research*, vol. 8, pp. 985–1005, 2007.

[56] J. J. Heckman, "Sample selection bias as a specification error," *Econometrica*, vol. 47, no. 1, pp. 153–162, 1979.

[57] D. A. Chon, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[58] V. V. Fedorov, *Theory of Optimal Experiments.* New York: Academic Press, 1972.

[59] D. P. Wiens, “Robust weights and designs for biased regression models: Least squares and generalized m-estimation,” *Journal of Statistical Planning and Inference*, vol. 83, no. 2, pp. 395–412, 2000.

[60] T. Kanamori and H. Shimodaira, “Active learning algorithm using the maximum weighted log-likelihood estimator,” *Journal of Statistical Planning and Inference*, vol. 116, no. 1, pp. 149–162, 2003.

[61] M. Sugiyama, “Active learning in approximately linear regression based on conditional expectation of generalization error,” *Journal of Machine Learning Research*, vol. 7, pp. 141–166, 2006.

[62] H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[63] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, *Direct importance estimation with model selection and its application to covariate shift adaptation*. Cambridge, MA: MIT Press, 2008.

[64] G. S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*. Berlin: Springer-Verlag, 1996.

[65] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[66] M. Yamada, M. Sugiyama, and T. Matsui, “Covariate shift adaptation for semi-supervised speaker identification,” in *Proceedings of 2009 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2009)*, Taipei, Taiwan, Apr. 19–24 2009, pp. 1661–1664.

[67] M. Sugiyama and S. Nakajima, “Pool-based active learning in approximate linear regression,” *Machine Learning*, vol. 75, no. 3, pp. 249–274, 2009.

[68] C. M. Bishop, *Pattern recognition and machine learning.* New York: Springer-Verlag, 2006.

[69] S. Calinon, F. Guenter, and A. Billard, “On learning, representing and generalizing a task in a humanoid robot,” *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, vol. 36, no. 5, 2006.

[70] G. Rätsch, T. Onoda, and K.-R. Müller, “Soft margins for adaboost,” *Machine Learning*, vol. 42, pp. 287–320, 2001.

[71] S. Amari, A. Chichocki, and H. H. Yang, *A new learning algorithm for blind signal separation.* Cambridge, MA: MIT Press, 1996.

[72] M. Sugiyama, M. Kawanabe, and K.-R. Müller, “Trading variance reduction with unbiasedness: The regularized subspace information criterion for robust model selection in kernel regression,” *Neural Computation*, vol. 16, no. 5, pp. 1077–1104, 2004.