KERNEL EIGENVOICE SPEAKER ADAPTATION

by

HO KA-LUNG

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Master of Philosophy
in Computer Science

August 2003, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

HO KA-LUNG

KERNEL EIGENVOICE SPEAKER ADAPTATION

by

HO KA-LUNG

This is to certify that I have examined the above M.Phil. thesis

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by

the thesis examination committee have been made.

DR. BRIAN MAK, THESIS SUPERVISOR

PROF. LIONEL NI, HEAD OF DEPARTMENT

Department of Computer Science

18 August 2003

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to Dr. Brian Mak for his supervision throughout my MPhil study and to Dr. James Kwok for his valuable suggestions in the kernel methods.

I am also very grateful to the LASTRE group. With the guidance of Dr. Brian Mak and Dr. Manhung Siu in the QEF and ASTRI projects, I was opened to the field of speech recognition and gained a great deal of hands-on experience. It helps me a lot on my research.

Finally, I would like to express my thanks to my colleagues of LASTRE group including Wilson Tam, Arthur Chan, Ivan Chan, Franco Chong, Karen Leung, Jimmy Wong and Mimi Ng, who teach me a lot in these two years.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

KERNEL EIGENVOICE SPEAKER ADAPTATION

by

HO KA-LUNG

Department of Computer Science

The Hong Kong University of Science and Technology

ABSTRACT

Speech recognition is a powerful and widely used technology nowadays. However, its performance is not robust enough due to variations in speech introduced by the operating environment, noises (their type and energy) and inter-speaker differences.

Speaker adaptation is an important technology to fine-tune either features or speech models for the mis-match due to inter-speaker variation. In the last decade, eigenvoice (EV) speaker adaptation has been developed. It makes use of the prior knowledge of training speakers to provide a fast adaptation algorithm (in other words, only a small amount of adaptation data is needed). Inspired by the kernel eigenface idea in face recognition, kernel eigenvoice (KEV) is proposed. KEV is a non-linear generalization to EV. This incorporates Kernel Principal Component Analysis (KPCA), a non-linear version of Principal Component Analysis (PCA), to capture the higher order correlations in order to further explore the speaker space and enhance recognition performance. The major difficulty is that through KEV adaptation, the adapted speaker model is estimated in the kernel feature space which may not have an exact pre-image in the input speaker-supervector space, yet observation likelihoods are computed in the acoustic observation space for both adaptation and recognition. Composite kernel is proposed to solve the problem.

Experimental investigation on TIDIGITS corpus, an English digits recognition task, using 4 seconds of adaptation data shows that KEV adaptation gives a 21% relative improvement over the speaker-independent (SI) model, a 25% relative improvement over MLLR adaptation and a 32% relative improvement over EV adaptation. When the speaker-adapted models from KEV are interpolated with the SI model, the relative improvements increase to 32% over SI model, 35% over MLLR adaptation, and 31% over similarly interpolated EV adaptation.

CHAPTER 1

INTRODUCTION

1.1 Background

Speech recognition is a very powerful technology that is widely used nowadays. Examples include voice-activated phone-dialing (VAD) by AT&T Wireless, Nokia and Motorola, voice-controlled personal digital assistant (PDA) by Palm, voice-controlled in-car music system by Sony, voice-operated light switch by VOS Systems, computer assisted language learning (CALL) and so on. In addition, a call center with interactive voice response (IVR) is an important application of speech recognition. This is used in various domains including credit/debit card enquiries, international travel bookings and processing insurance details and it showed huge business values. All these examples show the potential and importance of speech recognition technology.

However, inter-speaker differences is an important bottleneck to further improvement on the accuracy of speech recognition. To counter these problems, various kinds of speaker normalization and speaker adaptation methods have been proposed. Feature-based adaptation (or normalization) aims to reduce the undesired variations in the features while model-based adaptation aims to modify the acoustic models to optimize on a certain amount of data of a given speaker.

Among the various adaptation methods, eigenvoice adaptation (EV) is a well-known method to extract inter-speaker variations such as gender, age and accent from a set of training speakers by Principal Component Analysis (PCA). By assuming any speakers to be a linear combination of eigenvectors with a set of weights. A speaker-adapted model is obtained by finding the weights by maximizing the expected log likelihood of the given adaptation data.

In this thesis, we propose a novel non-linear extension to EV, which we call **Kernel Eigenvoice** (KEV) by utilizing kernel methods. The hypothesis is that the use of linear PCA in EV may not be best to capture the inter-speaker variations. In fact, EV is a special case of KEV using a linear kernel. By using the kernel trick, KEV uses KPCA, performing linear PCA in the high dimensional feature space, to enhances its capability in non-linearity without an explicit non-linear optimization. The main difficulty is how to express the adaptation algorithm in the observation space using the non-linear information in the feature space. Our solution is to compute kernel PCA using composite kernels.

1.2 Outline of the thesis

In chapter 2, the idea of speaker-dependent (SD) and speaker-independent (SI) modeling are discussed. It is followed by the evolution of KEV from speaker adaptation, eigenface, eigenvoice and kernel methods.

In chapter 3, conventional eigenvoice is introduced. A discussion of its objectives and a brief comparison between EV and Cluster Adaptive Training (CAT) [17] will be given. The outline of the EV algorithm follows. The general experimental setup is stated and two variations of speech model training methods are introduced and discussed. This chapter ends with a comparison of the recognition results on EV.

In chapter 4, the KPCA algorithm and the kernel eigenvoice adaptation are developed. The challenge of KEV and its proposed solution are investigated. The KEV algorithm for Gaussian kernel and polynomial kernel are presented. The time complexity of the algorithm as well as the recognition results on KEV are discussed.

Robust EV and KEV are introduced in chapter 5. The motivation, reformulation and the experimental results are also presented. In chapter 6, a comparison among EV, KEV and conventional adaptation techniques including

MLLR and MAP is presented. A brief discussion of the significance tests are included. Then, the relationship between the eigenvectors and speakers' characteristics is analyzed. The conclusion and future work are discussed in the last chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 SD modeling versus SI modeling

In speech recognition, acoustic modeling can be divided into two methods, that is, speaker-dependent (SD) modeling and speaker-independent (SI) modeling. SD modeling means that an acoustic model is trained by data from a specific speaker while SI modeling means that an acoustic model is trained by data from all speakers.

There are pros and cons in both SD and SI modeling. For SD modeling, the advantage is that a well-trained SD model is usually better than a well-trained SI model for the training speaker. [29] states that the error rate of an SD model is about one third of the error rate of an SI model. In [29], the author explains that ‘phonemes do not occupy absolute positions in acoustic space, but are perceived relative to each other‘. As all speakers are used to train an SI model, its probability distribution of phonemes in an SI model spreads out. In technical words, if Gaussian is used to model the distribution, it becomes flatter with larger variance. An illustrative example is that ‘One person's ”ow” in ”about” may sound like another person's ”oo” in ”room”.‘ In other words, the relative position of phonemes in acoustic space is weakened in an SI model. The disadvantage is that an SD model gives a very tough constraint on the application since it is usable by the training speaker only. This means that each user needs to have his/her own SD model. The amount of data for training a robust SD model is more than 5 minutes of speech data (depending on the domain and the complexity required). It is a completely user-unfriendly idea.

For SI modeling, its importance is that a fairly good acoustic model can be achieved for all people in general. Any user can utilize this model immediately. Recording speech and training an acoustic model for each new user is no longer necessary. Although it makes speech recognition more user-friendly, there are two drawbacks. Firstly, the distribution of data could affect or be biased toward some groups of people. So, it is important to keep everything as balanced as possible in the training-set and the evaluation-set, such as gender, accent and age group in order to ensure the performance is not biased on some factors. Secondly, since the modeling technique has to deal with the variations among speakers, complexity of the acoustic model should be higher. For example, the number of mixtures of HMMs in SI modeling should be higher than that in SD in order to achieve the same accuracy. This means that the decoding speed in SI is usually slower than the in SD. Although the complexity of HMMs partially gives the capacity for describing the variations in speakers, the mixture design does not directly account for speaker variation.

It is true that the SI approach is dominant in acoustic modeling. However, speaker variations cause a bottleneck in the recognition accuracy. Therefore, if a certain amount of speaker-specific data (adaptation data) is available, can we make use of it to improve acoustic models? This leads to speaker adaptation research.

2.2 Speaker adaptation

As introduced in Section 1.1, speaker adaptation can be divided into two classes, which are feature-based adaptation and model-based adaptation. Vocal-tract normalization (VTLN) [13] is one feature-based example, which is a parametric method used to normalize the effect introduced by the variations of the vocal tract length of speakers. As stated in [49], its major limitation is that it is inefficient to have phone-level or word-level control in a feature-based adaptation. For example, if the adaptation is realized by a transformation, this transformation has to be applied to all observable frames. In contrast, a model-based adaptation allows a transformation to be applied to observable frames which belong to vowels while another transformation is applied to observable frames which belong to fricatives.

The three most common model-based adaptations are the Maximum Likelihood Linear Regression (MLLR) [34], the Maximum a Posterior (MAP) adaptation [33] and eigenvoice (EV) adaptation [29, 30, 28, 27].

Instead of giving the details of the adaptation algorithms, the comparison is highlighted. In MAP adaptation, large amounts of adaptation data as well as the coverage of the parameters are important. Rarely seen parameters could result in poor performance. The rate of convergence to an SD model is slow. In MLLR adaptation, using block diagonal or full transformation with a regression class tree makes it flexible and tunable. However, insufficient adaptation data could result in a poorly estimated transformation matrix leading to poor recognition accuracy. In eigenvoice adaptation, the major idea is to make use of a priori knowledge of speaker information. By applying PCA on training speakers, eigenvoices are obtained. They describe inter-speaker variations. Speaker space is spanned by the first few eigenvoices. There is a set of weights for each unseen speaker and each weight corresponds to each eigenvoice. Speaker-adapted model is found within the speaker space by obtaining the set of weights in the adaptation process. Further discussion of EV continues in chapter 3.

	MAP	**MLLR**	**eigenvoice**
Amount of adaptation data	Large	Medium	Small
Convergence to SD model	Yes	Yes	No
Rate of saturation	Slow	Fast	Fast
Others	dependent on on the distribution of data	flexible: regression, class tree, block diagonal transformation	model speaker variations directly

Table 2.1: Comparison of the three main model-based adaptation methods

EV is especially suitable for small amounts of adaptation data. It models the speaker variations directly, but it does not necessarily converge to an SD model. Empirical results show that improvement saturates quickly, meaning that beyond a certain limit, more adaptation data would not give further improvement. The comparison is summarized as in Table 2.1.

2.3 From PCA to eigenface and eigenvoice

The story of kernel eigenvoice starts from one of the most famous linear transformation methods which is the PCA [23]. It is a simple but powerful method that can be used for dimensionality reduction or redundancy reduction, de-correlation of data, feature extraction and so on. PCA guarantees that the mean square of reconstruction error is minimized. It is a second order method that only makes use of information in correlation or covariance of multi-dimensional data.

In conventional face recognition methods, facial features including eyeballs, nose, mouth and head shape are detected for face identification. In 1992, Turk and Pentland [46] first proposed the eigenface. It is a novel unsupervised way to decouple faces into basis-faces by PCA. Any face is then expressed as a linear combination of the eigenfaces so that the dimension is reduced substantially. The detection and identification of human faces becomes a simple pattern recognition task in the eigenface space.

Two main streams of extension to the eigenfaces are available. The first stream is the work on statistical analysis methods other than PCA. In [19], instead of using PCA, it was proposed to use Fisher representation to enhance the discrimination power; this is called fisherface. Other variations such as the use of Independent Component Analysis (ICA) on face recognition was investigated in [3]. The second stream of extension is that instead of applying the statistical analysis methods on the pixels of the image directly, other spaces are explored. Eigenhill and eigenedge was investigated in [54] while eigenmotion was investigated in [55].

In the speech domain, speaker identification and recognition is a direct analogy to face recognition tasks while speaker adaptation is a closely related problem. Speaker adaptation using an eigen-decomposition technique, called eigenvoice, was first proposed in [29]. In [30, 28, 27], the maximum-likelihood eigen-decomposition (MLED) estimator for Gaussian mean adaptation was outlined. Experiments on isolated English letter recognition showed encouraging results. Later, the use of eigenvoice in speaker identification and recognition was also explored in [44].

Similar to eigenfaces, the extension of eigenvoice can be divided into four streams. The first stream is an extension of the statistical analysis. In [38], the PCA-based eigenvoice adaptation was extended to the Linear Discriminant Analysis (LDA) transformation and piecewise linear constraints. In [22], both PCA and ICA were used to analyze the speaker variability. It was found that the first two ICA components corresponded to gender and accent respectively while the first PCA component corresponded to gender only. In [15], instead of using the maximum likelihood for eigenvoice adaptation, eigenvoice was used for speaker clustering. HMM sets were trained for each speaker cluster and a parallel recognition scheme for choosing the maximum HMM score was adopted. The second stream extends the scope of eigenvoice. It means that instead of applying statistical analysis on covariance or correlation of the means of HMM sets, other targets are explored. In [8] and [48], the eigenspace-based MLLR approach was introduced. PCA was applied to the MLLR transformation matrix. In [9], the eigenspace-based MAP linear regression approach was proposed. In [10], the idea of eigenroom was introduced. Adaptation was used to deal with the mismatch mostly due to room reverberation. The third stream extends the eigenvoice family technique suitable for the migration from small vocabulary tasks to large vocabulary continuous speech recognition (LVCSR). In [38], the first experiments on relatively large corpus Wall Street Journal dictation tasks were done, which achieved a 15% relative improvement. In [31], the use of eigen-centroid plus delta tree (EDT) for a compact context-dependent eigenvoice modeling was proposed. The fourth stream investigates the combination of the eigenvoice approach with other conventional adaptation approaches. This is due to the fact that eigenvoice is only good at a small amounts of adaptation data. When the amount of adaptation data increases, conventional approaches such as MLLR and MAP are more advantageous. Related discussions were presented in [7] and [9].

2.4 Kernel methods

On top of the various statistical analysis methods such as PCA, LDA, ICA, kernel methods have been developing at a fast pace in the last decade. The idea of kernel methods was discussed thoroughly in [3]. A simple example borrowed from it (the example is the same although the figures are re-generated) to show the power of high dimension in Figures 2.1 and 2.2.

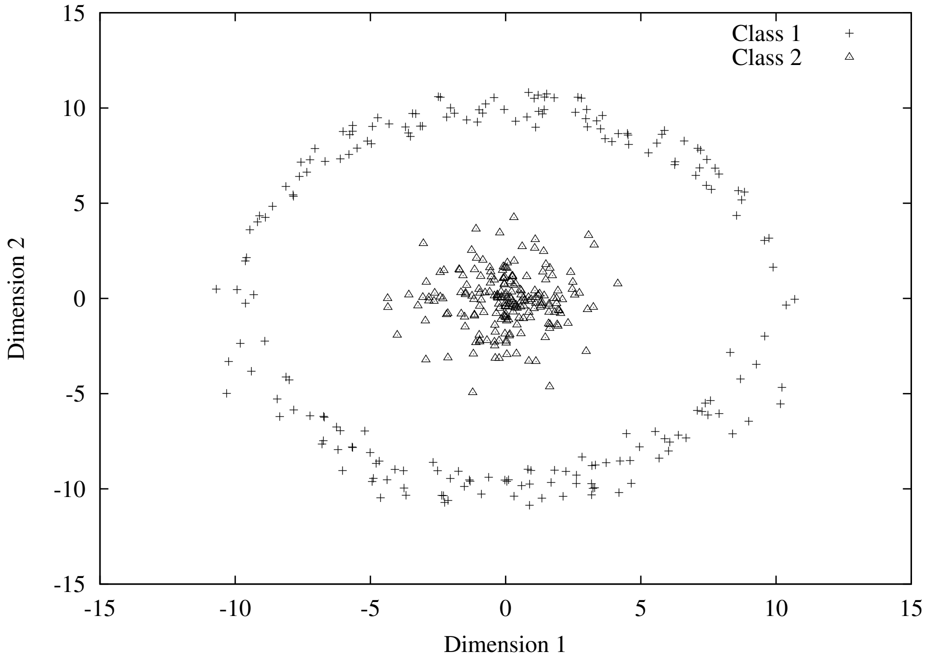

Figure 2.1: Input space of the toy problem (Dimension 1 and 2 correspond to x_1 and x_2 respectively)

In Figure 2.1, there are some data points with two dimensions in two classes in the input space which is not linearly separable. If there is a mapping $\varphi : (x_1, x_2) \to (x_1, x_2, x_1^2 + x_2^2)$, data points from input space can map to the feature space as shown in Figure 2.2 where class 1 and class 2 are linearly separable.

However, as the observation dimension increases, the possible combinations of high dimension representation increase exponentially. It is not a good idea to have an explicit form. Therefore, if the dot product in the feature space is given by $k(x_1, x_2) = \Big\langle \varphi(x_1), \varphi(x_2) \Big\rangle$ and the algorithm is expressed in terms of dot product, then, we can perform the algorithm in high dimensional feature space using dot products without knowing the explicit form of the mapping.

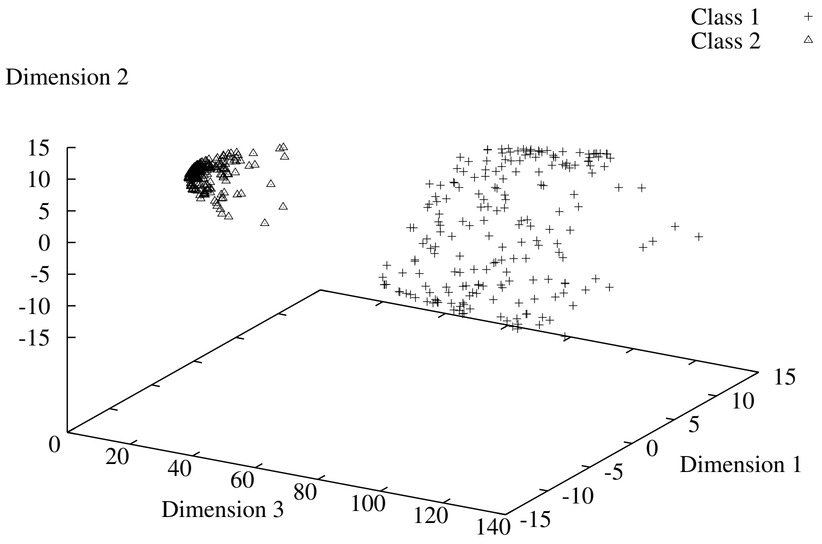

Figure 2.2: Feature space of the toy problem (Dimension 1, 2 and 3 correspond to x_1, x_2 and $x_1^2 + x_2^2$ respectively)

In [3], the Kernel Principal Component Analysis (KPCA) was introduced. The main concept is to map the input space to a feature space of higher dimension and linear PCA is performed in the feature space. Recently, KPCA is applied to face recognition to take into account higher order correlations [53, 26] and the method is called kernel eigenface. Later, the Fisher Linear Discriminant (FLD) was explored in the work of [51].

2.5 Summary of the evolution

The summary of the evolution is shown in Figure 2.3. In the party of linear algorithms, it starts from PCA, following the development of eigenface and eigenvoice for face recognition and speaker adaptation respectively. Similarly, in the party of non-linear algorithms, KPCA first evolved from PCA. It was followed by the study of kernel eigenface and currently proposed kernel eigenvoice in this thesis.

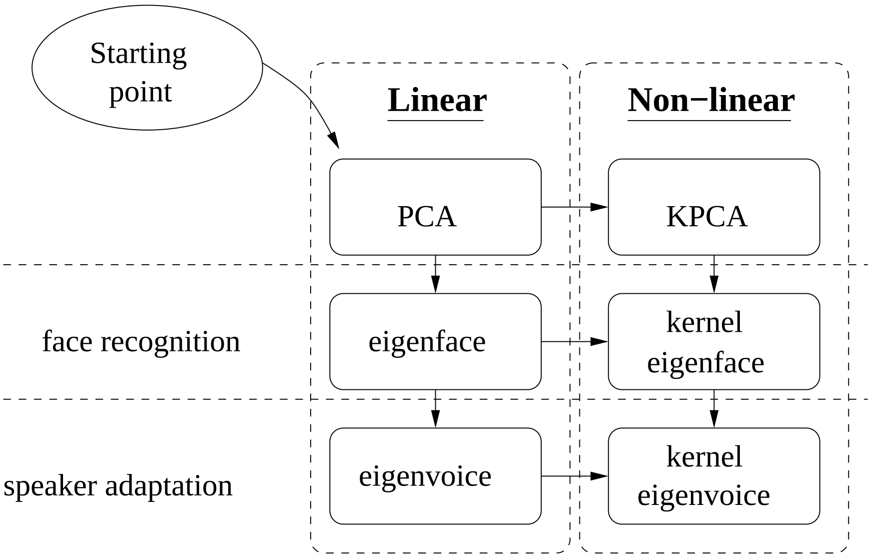

Figure 2.3: Summary of the evolution

CHAPTER 3

CONVENTIONAL EIGENVOICE

3.1 Idea of eigenvoice

Following the discussion in Section 2.1, one may wonder if it is possible to estimate an SD model with a very small amount of data. The idea of EV is seeded from this question. One trivial but important observation is that some speakers are similar. An unseen speaker can be inferred from a similar one from training speakers. This situation exists in the eigenface research too. As discussed in [46], it is true that humans' faces usually have two eyes, two ears, a nose and a mouth. They are common in many aspects although they may differ in face shape or their relative positions. It inspires researchers to try to reduce free parameters from all pixels of faces to the weight parameters on eigenfaces.

If we have many speakers in the training set, we can pre-train a lot of SD models from various kinds of reference speakers. A simple method is to use the adaptation data of a new speaker to pick the closest SD model as the adapted model. The main shortcoming of the method is that it demands a huge amount of speakers. In addition, an SD model from a similar speaker in training-set is usually not good enough for speakers in an unseen test-set.

A modification of the last method is to assume that any speaker model is a weighted sum of the training speaker model. This increases the speaker space so that it is more likely that a good model exists in this search space. However, as the number of training speakers increase, the number of parameters increases and more adaptation data is required.

Thus, there is a need to reduce the number of parameters so as to reduce the requirement of adaptation data. One way to do this is through clustering of speakers, as in CAT [17]. Another way is to perform eigen-decomposition on the data to extract the principal components, which is the eigenvoice. Any speaker model is represented as a linear combination of the eigenvoices in the eigenspace.

In short, eigenvoice adaptation can be divided into two main steps, which are defining the speaker space and searching for a good speaker model. This is given in Table 3.1.

	step 1 – defining speaker space	step 2 – searching for a good speaker model
CAT	by clustering	maximum likelihood
eigenvoice	PCA	maximum likelihood

Table 3.1: Summary of the two steps in eigenvoice-family adaptation

3.2 Introduction of parameter spaces

Several parameter spaces are used in EV at different stages. Three of them are introduced for clarity in this section, which are the observation space, supervector space and eigenspace. The idea is summarized in Figure 3.1 and they are elaborated below.

1. **Observation space**
   This is the acoustic feature space after feature extraction in step one of Figure 3.1. For example, an acoustic observation vector used in this thesis consists of 12 mel-frequency cepstral coefficients (MFCC) and the normalized energy from each speech frame. It is a 13-dimensional space.

2. **Supervector space** (or speaker input space)
   A supervector is formed by concatenating the means of HMM states as shown in step two of Figure 3.1. Supervectors define the input space in step three with dimension of:

$$\begin{aligned} & \text{dim(supervector space)} \\ = \; & \text{(number of HMMs) x (number of states per HMM) *} \\ & \text{(dimension of the observation space)} \end{aligned}$$

In all experiments in this thesis, the number of HMMs is 11 while the

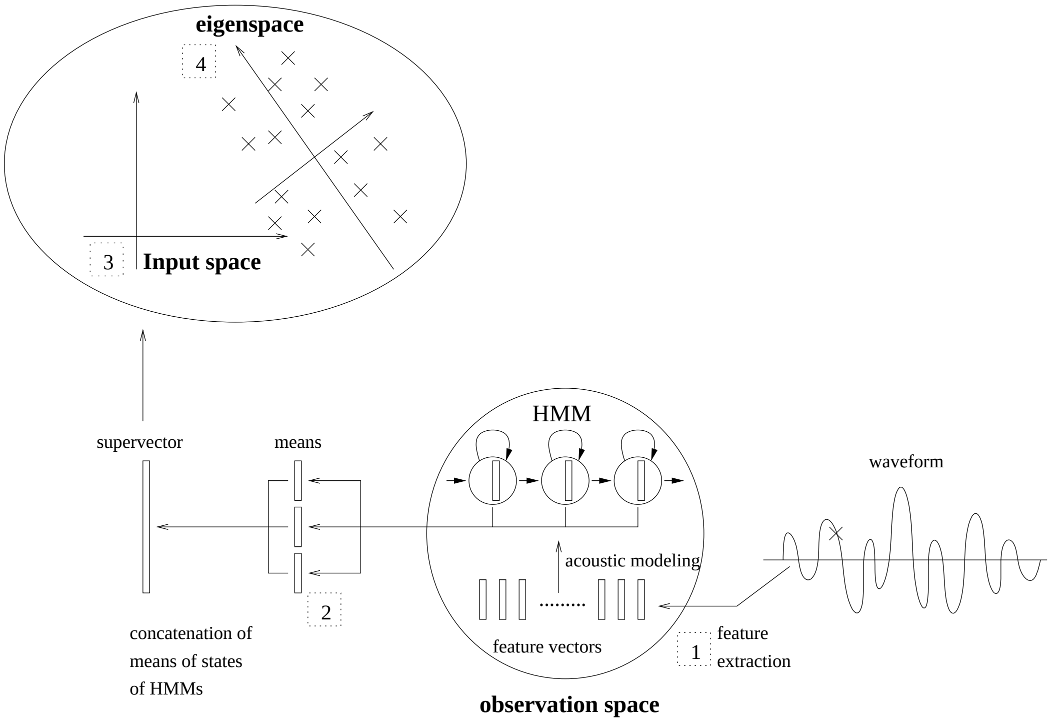

Figure 3.1: Illustration of various parameter spaces used in EV (1) Features are extracted from raw speech files which defines the input space. (2) Means of states of HMMs are concatenated. (3) Supervectors define the input space. (4) Eigenspace is obtained by applying PCA on the input space.

number of states per HMM is 16. Therefore, the dimension of supervector space is 2288.

3. **Eigenspace** (in order to distinguish the eigenspace found in KEV, it is called conventional eigenspace)
   It is the space after eigen-decomposition on the supervector space as shown in step four. Only the first few eigenvectors with the largest eigenvalues are chosen usually so that its dimension is much less than that of the supervector space. It is used in conventional eigenvoice.

3.3 Conventional eigenvoice adaptation

In the conventional eigenvoice, the Gaussian mean vectors of all HMM states of a speaker are concatenated in a given order to form the speaker supervector. Therefore, if μ_r is the r^{th} Gaussian mean vector, then μ is the concatenated speaker supervector in Equation 3.1.

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_r \\ \vdots \\ \mu_R \end{bmatrix} \tag{3.1}$$

PCA is performed on the covariance matrix or correlation matrix of the speaker supervectors to extract the eigenvectors. These eigenvectors are called eigenvoices. It is also in the form in Equation 3.1. Any speaker supervector is assumed to be a linear combination of the eigenvoices as in equation 3.2 and 3.3 for covariance and correlation approach respectively.

- When covariance matrix is used for eigen-decomposition, the unseen speaker supervector $\mathbf{s}$ is defined as

  $$\begin{aligned} \mathbf{s} - \bar{\mathbf{e}} &= \sum_{m=1}^{M} w_m \mathbf{e}_m \\ \mathbf{s} &= \bar{e} + \sum_{m=1}^{M} w_m \mathbf{e}_m \end{aligned} \tag{3.2}$$

  where $\bar{\mathbf{e}}$ is the mean of eigenvectors and w_m is the weight of the m^{th} eigenvector. The set of weights are unknown variables and each speaker has his own set of weights.

- When correlation matrix is used, the difference is that each dimension is normalized before eigen-decomposition. It becomes:

  $$\begin{aligned} \mathbf{Z}^{-1}(\mathbf{s} - \bar{\mathbf{e}}) &= \sum_{m=1}^{M} w_m \mathbf{e}_m \\ \mathbf{s} &= \bar{\mathbf{e}} + \sum_{m=1}^{M} w_m \mathbf{Z} \mathbf{e}_m \\ &= \bar{\mathbf{e}} + \sum_{m=1}^{M} w_m \tilde{\mathbf{e}}_m \end{aligned} \tag{3.3}$$

where $\mathbf{Z} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma_D \end{bmatrix}$

where σ_d is the standard deviation of the d^{th} component in the supervectors, and $\tilde{\mathbf{e}}_m = \mathbf{Z}\mathbf{e}_m$

Thus, determining the speaker-adapted model for a new speaker means finding his/her eigenvoices weights. This can be done by maximizing the likelihood of his/her adaptation data. Since the state sequence is a hidden variable, expectation maximization (EM) is used for optimization. The auxiliary function is defined as the expected log likelihood and is given by:

$$Q(\mathbf{w}) = Q_\pi + Q_a + Q_b(\mathbf{w}) \tag{3.4}$$

where

$$\begin{aligned} Q_\pi &= \sum_{r=1}^{R} \gamma_1(r) \log(\pi_r) \\ Q_a &= \sum_{p,r=1}^{R} \sum_{t=1}^{T-1} \xi_t(p,r) \log(a_{pr}) \\ Q_b(\mathbf{w}) &= \sum_{r=1}^{R} \sum_{t=1}^{T} \gamma_t(r) \log\Big(b_r(\mathbf{o}_t)\Big) \end{aligned} \tag{3.5}$$

Q_π, Q_a and $Q_b(\mathbf{w})$ corresponds to the initial probability, transition probability and observation probability; π_r is the initial probability of state r; $\gamma_t(r)$ is the posterior probability of observation $\mathbf{o}$ being at state r at time t; $\xi_t(p,r)$ is the posterior probability of observation $\mathbf{o}$ being at state p at time t and at state r at time $t+1$; b_r is the Gaussian pdf of the r^{th} state after re-estimation and $\mathbf{o}_t$ is an observation frame at time t.

Since Q_π and Q_a are independent of w_j, they can be ignored in the weights estimation. For simplicity, we only consider $Q_b(\mathbf{w})$ as the auxiliary function in the rest of the thesis. It is expanded as:

$$Q_b(\mathbf{w}) = \sum_{r=1}^{R}\sum_{t=1}^{T} \gamma_t(r)\Big[d_1 \log(2\pi) + \mathbf{C}_r + ||\mathbf{s}_r - \mathbf{o}_t||^2_{\mathbf{C}_r}\Big] \tag{3.6}$$

where $\mathbf{C}_r$ is the covariance matrix of the Gaussian at state r; $\mathbf{s}_r$ is the new speaker's mean vector defined in Equations 3.2 or 3.3.

In EV, $\mathbf{s}_r$ can be expressed in terms of weights w_m and they are unknown. By differentiating $Q_b(\mathbf{w})$ with respect to each w_j for $j = 1 \cdots M$, a set of M linear equations with M variables are obtained. This problem is analytically solvable. They are described as follows.

- For the covariance case:

$$\begin{aligned}\frac{\partial Q_b}{\partial w_j} &= -\sum_{r=1}^{R}\sum_{t=1}^{T} \gamma_{tr}\mathbf{e}'_{jr}\mathbf{C}_r^{-1}(\mathbf{s}_r - \mathbf{o}_t) \\ &= -\sum_{r=1}^{R}\sum_{t=1}^{T} \gamma_{tr}\mathbf{e}'_{jr}\mathbf{C}_r^{-1}\Big[\Big(\bar{\mathbf{e}}_r + \sum_{m=1}^{M} w_m \mathbf{e}_{mr}\Big) - \mathbf{o}_t\Big] \end{aligned} \tag{3.7}$$

Set $\frac{\partial Q_w}{\partial w_j} = 0$,

$$\sum_{r=1}^{R}\sum_{t=1}^{T} \gamma_{tr}\mathbf{e}'_{jr}\mathbf{C}_r^{-1}(\mathbf{o}_t - \bar{\mathbf{e}}_r) = \sum_{r=1}^{R}\sum_{t=1}^{T} \gamma_{tr}\Big(\sum_{m=1}^{M} w_m \mathbf{e}'_{jr}\mathbf{C}_r^{-1}\mathbf{e}_{mr}\Big) \tag{3.8}$$

- For the correlation case, the solution is the same as the one in covariance case except that $\mathbf{e}_{jr}$ is replaced by $\tilde{\mathbf{e}}_{jr}$ as follows:

$$\sum_{r=1}^{R}\sum_{t=1}^{T} \gamma_{tr}\tilde{\mathbf{e}}'_{jr}\mathbf{C}_r^{-1}(\mathbf{o}_t - \bar{\mathbf{e}}_r) = \sum_{r=1}^{R}\sum_{t=1}^{T} \gamma_{tr}\Big(\sum_{m=1}^{M} w_m \tilde{\mathbf{e}}'_{jr}\mathbf{C}_r^{-1}\tilde{\mathbf{e}}_{mr}\Big) \tag{3.9}$$

3.4 Experimental setup

TI-digits corpus [35] is the target corpus for investigation. It is a clean connected digit corpus sampled at 20KHz. There are 163 speakers for each of the standard training-set and test-set. There are about 77 utterances for each speaker. They are in various length ranging from one to seven digits. Speakers are from 22 dialect regions of USA with ages ranging from six to seventy. In the corpus it is, by default, divided into four main groups, which are girl, boy, woman and man.

Adaptation experiments were done with different amounts of adaptation data. Three of them are investigated, which are 2-second, 4-second and 10-second adaptation-sets. The detailed information is provided by Table 3.2.

Name	**Number of digits**	**Duration**	**Duration** (without silence)
2-second	5	3.0 s	2.1 s
4-second	10	5.5 s	4.1 s
10-second	20	13.0 s	9.6 s

Table 3.2: Detailed information of the adaptation sets (The third column is the recorded duration and the fourth column is the speech duration without silence according to the force alignment by the SI model.)

For each testing speaker, their data are divided into five mutually exclusive sets (e.g., A, B, C, D, E) as in Figure 3.2. A random subset (depending on the amounts of adaptation data) of one set is used for adaptation while the remaining four sets are used for testing each time. Sets are rotated and tested repeatedly five times. (It means that, subset of ”A” is used for adaptation and ”B”, ”C”, ”D” and ”E” are used for testing for the first time. Subset of ”B” is used for adaptation and ”A”, ”C”, ”D” and ”E” are used for testing in the second time and so on.) In each subset, the length of the utterances is kept balanced. Supervised adaptation is adopted.

In the feature extraction, an acoustic vector consisting of 12 MFCCs and the normalized energy is extracted from each speech frame of 25ms at each 10ms. HMM is used for acoustic modeling. The prototypes of the HMMs are illustrated in Figure 3.3 Sixteen (real) states left-to-right HMMs are used for modeling eleven digits (including ”one”, ”two”, ..., ”nine”, ”oh” and ”zero”). Three (real) states left-to-right HMM (with a skip arc from state one to state three and a loop-back arc from state three to state one) is used for modeling silence. One (real) state HMM is used for modeling an optional short pause. For simplicity, only single mixture Gaussian is used for each state of HMMs. These settings are used throughout all the experiments. Since the dimension of observation space is 13

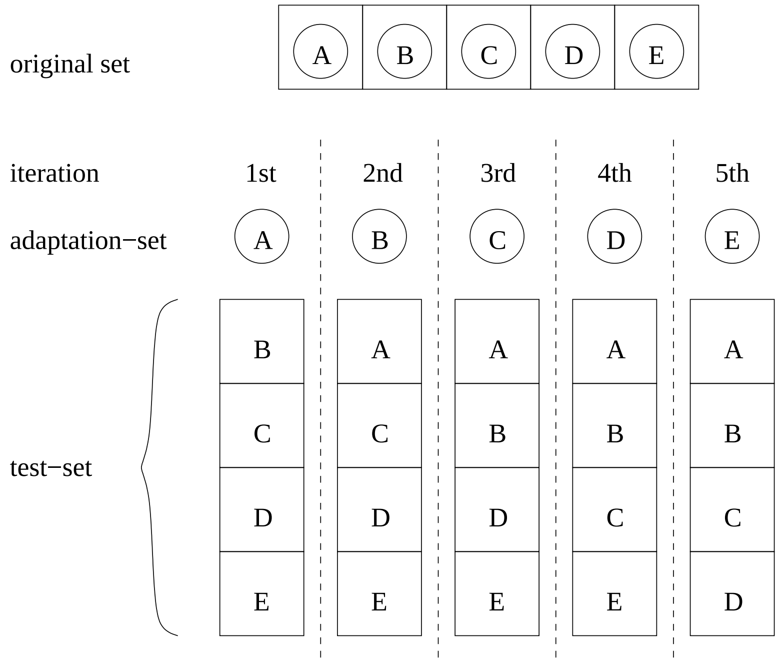

Figure 3.2: Defining adaptation-sets and test-sets (Original set is divided into 5 sets denoted by square. A subset is random sampled from each of the 5 sets denoted by circle.)

and there are 11 digits with 16 states, the resulted dimension in supervector is $13 * 11 * 16 = 2288$.

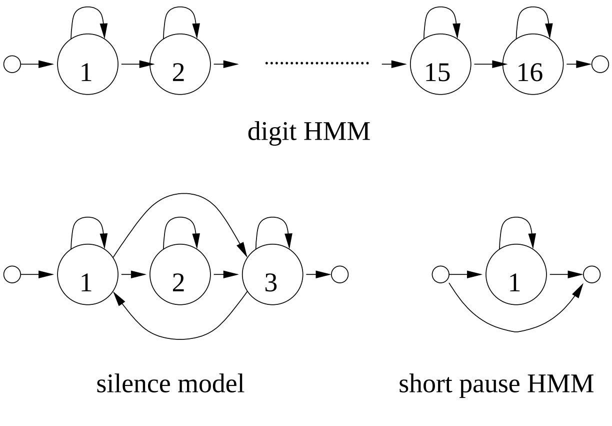

Figure 3.3: Illustration of prototype of the HMMs (the small circle represents a null state while the large circle represents a real state. There are 16 real states for each digit HMM.)

In training the SI model and the SD models for eigenvoice, two approaches are investigated:

- **Training approach A (Illustrated in Figure 3.4**)

  The SI model and the SD models are trained independently using the flat-start procedure. The means of the SD models are then used for eigen-decomposition. In addition to eigenvectors, variances, transition probability matrices, silence (SIL) and short pause (SP) HMMs from SI model are used for eigenvoice adaptation. This is the simplest approach. One drawback of this approach is that there may be a mismatch between the SD models and the borrowed quantities.

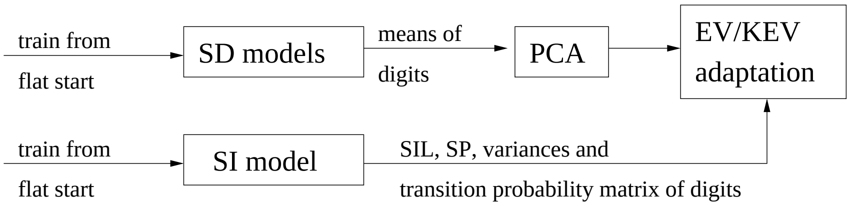

Figure 3.4: Illustration of the training approach A (SI and SD models are trained independently)

- **Training approach B (Illustrated in Figure 3.5)**

  The SI model is trained first. It is copied as the initialization for SD models instead of a flat-start initialization. In HMM parameters re-estimation in SD models, only the means of digit HMMs are updated. The SIL, SP, variances and transition probability matrices are identical to the corresponding one in SI. These specially trained SD models are used for eigenvoice adaptation.

  The advantage of this method is that it ensures SIL, SP and digit HMMs match. Since only one set of SIL and SP as well as the variances and transition probability can be used in the adapted model, the ones from the SI model are generally good for all speakers. If SD models share them in the expected maximization (EM) re-estimation of the means, it ensures their consistence.

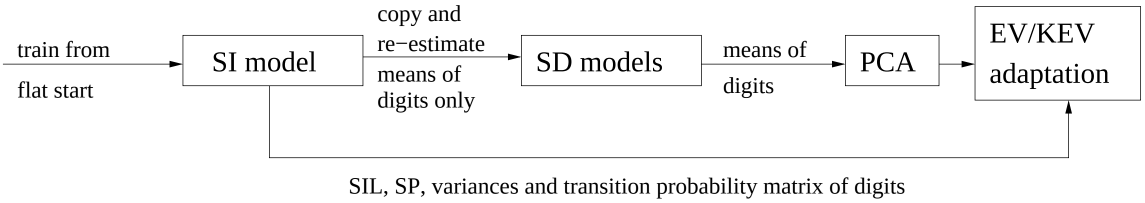

Figure 3.5: Illustration of the training approach B (SI and SD share SIL, SP, variances and transition probability matrices of digits)

3.5 Conventional eigenvoice adaptation experiment

The first experiment compares the two proposed training approaches described in 3.4. Eigenvoice adaptation using covariance matrix for eigen-decomposition is conducted. Only 10-second of adaptation-set is used. The results are shown in Figure 3.6. It shows that approach B is better than approach A and is used in the rest of the thesis.

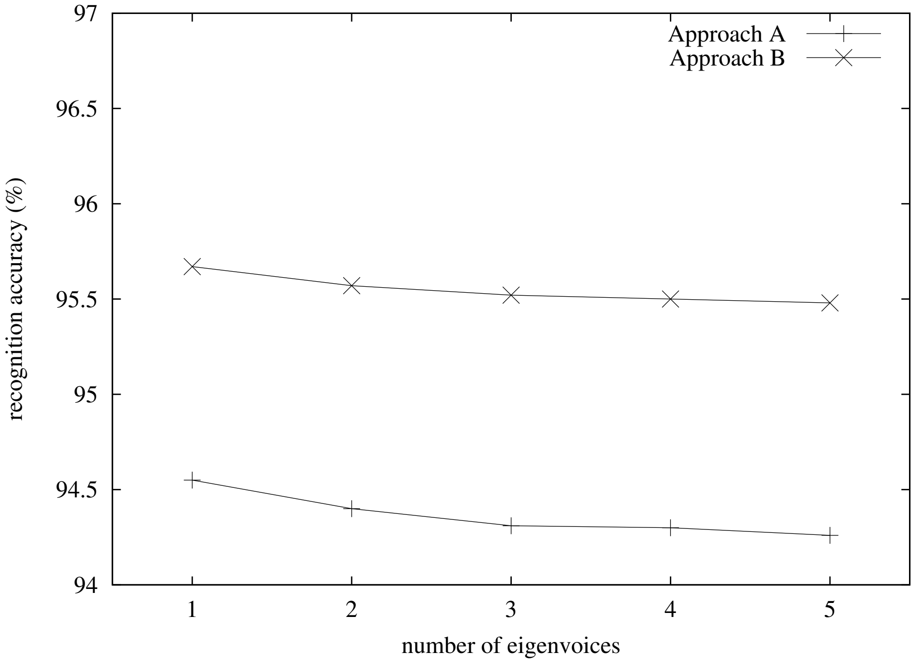

Figure 3.6: Comparison of the two suggested training approaches for eigenvoice adaptation

The second experiment compares the covariance and correlation approaches in conventional eigenvoice adaptation. Various amounts of adaptation data (2-second and 10-second adaptation- sets) and numbers of eigenvectors (1-5) are tried. The baseline is the accuracy of the SI model, which is 96.25%. The results are plotted in Figure 3.7.

Firstly, by comparing the correlation approach and the covariance approach, the correlation one is better than the covariance one for using one or two eigenvoices. It could be explained that in the correlation approach, components are normalized before PCA. It then avoids some components with large dominating

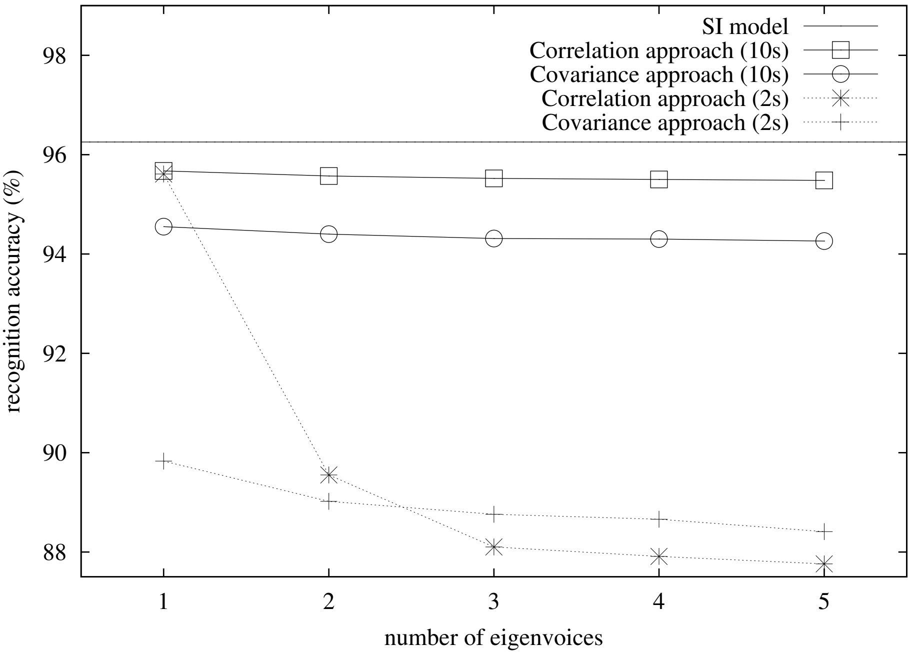

Figure 3.7: Comparison of the covariance approach and the correlation approach in conventional eigenvoice adaptation

values. Secondly, we find that the conventional eigenvoice is worse than the baseline SI model. It reflects that the linearity assumption in EV may not be good enough for all tasks. This is also an important motivation for proposing KEV.

CHAPTER 4

KERNEL EIGENVOICE

4.1 Revisit the definition of parameter spaces

Before introducing the KEV, two more parameter spaces are introduced in addition to the spaces discussed in Section 3.2, which are speaker feature space and kernel eigenspace. These ideas are illustrated in Figure 4.1.

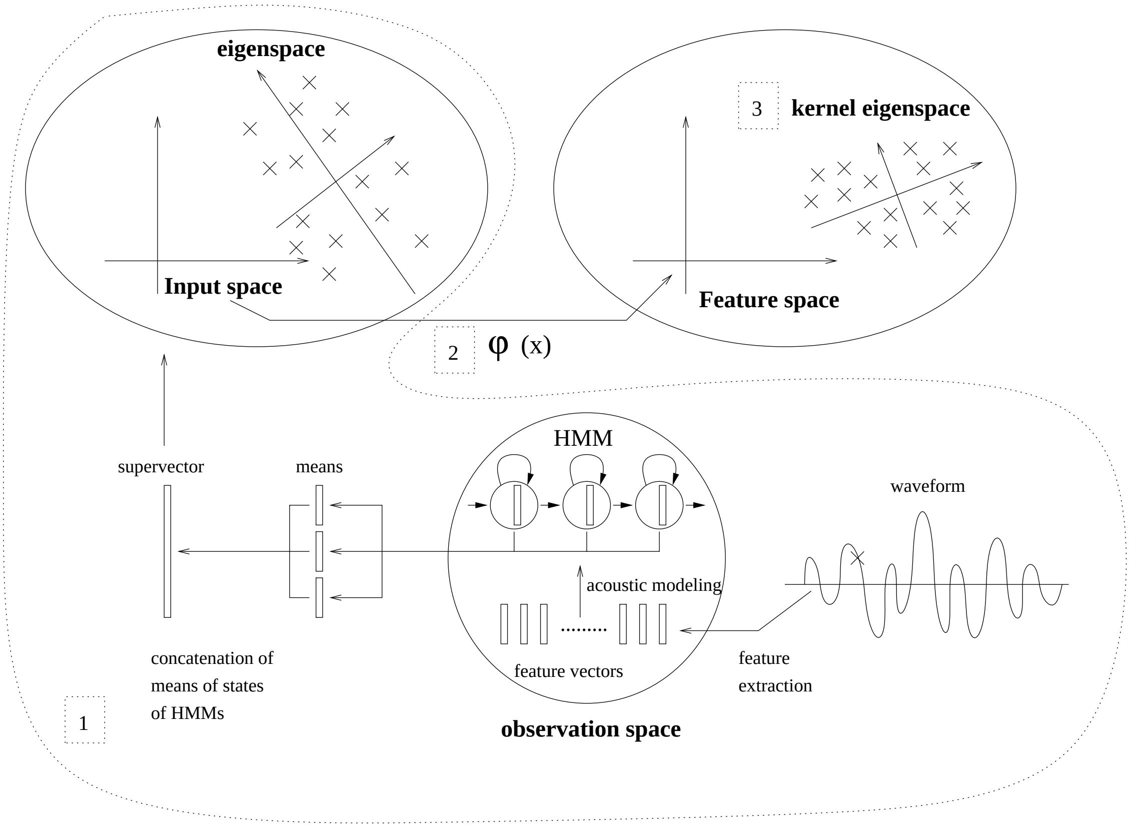

Figure 4.1: Illustration of parameter spaces used in KEV (1) Things inside dotted region is the same as EV. (2) $\varphi(x)$ is a mapping to a high dimensional feature space. (3) KPCA is used to find the kernel eigenspace.

The basic idea behind kernel methods is that if a function $\varphi(x)$ exists, the speaker input space can be mapped to a high dimensional feature space in a non-linear manner in step two of Figure 4.1. However, $\varphi(x)$ does not necessarily exist and it is, in fact, undesirable to work with $\varphi(x)$ explicitly because both expressing and computing the high dimensional vectors is very expensive. Therefore, if a kernel function is defined as the dot product of vectors in the feature space, then any linear algorithm that works on dot products is equivalent to a non-linear algorithm in the input space.

Similar to EV, there is a space called kernel eigenspace in step three of Figure 4.1. This is the space after eigen-decomposition in the feature space, which is found by PCA in the feature space. It is described by a set of orthogonal vectors in the feature space with eigenvalues in sorted order, which represent the variances in the corresponding eigenvectors. So, the first few eigenvectors with the largest eigenvalues are chosen to describe the kernel eigenspace. This guarantees to minimize the re-construction error in the feature space.

Remember that the observation space has dimension D_0 which is the smallest one. The input speaker space (of dimension D_1) is then the concatenation of Gaussian means and $D_0 \ll D_1$. The feature space (of dimension D_2) is a high dimensional space mapped from the input speaker space and usually $D_1 \ll D_2$. The eigenspace (of dimension D_3) and the kernel eigenspace (of dimension D_4) is the "most useful" subspace in the input speaker space and the feature space respectively.

In summary,

$$\begin{cases} D_0 \ll D_1 \ll D_2 \\ D_3 \ll D_1 \\ D_4 \ll D_2 \end{cases}$$

4.2 Overview of KEV

One of the crucial limitations of conventional eigenvoice adaptation is that unseen speakers are assumed to be a linear combination of eigenvoices. However, a linear constraint may not be good enough. Therefore, incorporating non-linearity is desired. In [3], KPCA was proposed. This is used to extract components in a non-linear manner in the feature space. The KEV makes use of the KPCA for components extraction and kernel trick is used in the adaptation algorithm which is discussed in the rest of this section. The overall idea includes four main steps as follows:

1. **Define kernel function**

   A kernel function in the input space defines the dot product of two data in the feature space. Different kernel functions represent different forms of non-linearity. A kernel matrix gives the similarity measure between each pair of training vectors. The element in the i^{th} row and the j^{th} column is the dot product between the i^{th} sample and the j^{th} sample in the dataset. In this thesis, Gaussian kernel and polynomial kernel were studied.

2. **KPCA**

   Principal components are derived from the kernel matrix (which is defined in step 1) by KPCA. The details of the algorithm will be discussed in Section 4.3.

3. **Express speaker vector and distance**

   The feature vector of a new speaker is expressed as a linear combination of the eigenvectors in the feature space while the distance in the input space is expressed in terms of dot products in the feature space using the kernel trick.

4. **ML estimation of eigenvoice weights**

   Similar to the EV, the expected log-likelihood is maximized on a set of speaker-specific adaptation data. Due to the non-linearity, there is no analytical solution and gradient-based numerical methods are used. The Generalized Expectation Maximization (GEM) is used instead of EM.

4.3 KPCA

The idea of KPCA is to perform PCA algorithm in terms of dot products in the feature space so that kernel tricks can be used. The detailed derivation and discussion of the KPCA can be found in [3]. Here is a summary of the major steps.

- Let function φ be the mapping from the input space to the feature space, $\tilde{\varphi}$ be its centered version and $\bar{\boldsymbol{\varphi}}$ be the mean of the training vectors in the feature space. In PCA (or KPCA), the centered covariance is needed and it is given by the following (the proof is given in A.1 of appendix A):

$$\tilde{\mathbf{C}} = \mathbf{HCH} \tag{4.1}$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{11}'$ and $\mathbf{1} = [11...1]'$.

- The centered covariance matrix $\tilde{C}$ is defined as:

$$\begin{aligned} \tilde{\mathbf{C}} &= \frac{1}{N}\sum_{n=1}^{N} \tilde{\varphi}(\mathbf{x}_n)\tilde{\varphi}(\mathbf{x}_n)' \\ &= \frac{1}{N}\tilde{\mathbf{\Phi}}_x\tilde{\mathbf{\Phi}}_x' \end{aligned} \tag{4.2}$$

for $n = 1 \cdots N$, x_n is the n^{th} training speaker supervector and $\tilde{\mathbf{\Phi}}_x = \Big(\tilde{\varphi}(\mathbf{x}_1), \cdots, \tilde{\varphi}(\mathbf{x}_N)\Big)$.

- In [3], it is shown that all eigenvectors $\mathbf{u}_m$ lies in the span of training vectors $\tilde{\varphi}(\mathbf{x}_1), \cdots, \tilde{\varphi}(\mathbf{x}_N)$. Then,

$$\begin{aligned} \mathbf{u}_m &= \sum_{n=1}^{N} \alpha_{mn}\tilde{\varphi}(\mathbf{x}_n) \\ &= \tilde{\mathbf{\Phi}}_x\boldsymbol{\alpha}_m \\ \mathbf{u} &= \tilde{\mathbf{\Phi}}_x\boldsymbol{\alpha} \end{aligned} \tag{4.3}$$

where α_{mn} is the n^{th} element of vector $\boldsymbol{\alpha}_m$ and $\boldsymbol{\alpha} = \Big(\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_N\Big)$.

- Eigenvalue problem in the high dimensional space is presented as:

$$\begin{aligned} \tilde{\mathbf{C}}\mathbf{v} &= \boldsymbol{\lambda}\mathbf{v} \\ \frac{1}{N}\tilde{\mathbf{\Phi}}_x\tilde{\mathbf{\Phi}}_x'\tilde{\mathbf{\Phi}}_x\boldsymbol{\alpha} &= \boldsymbol{\lambda}\tilde{\mathbf{\Phi}}_x\boldsymbol{\alpha} \end{aligned} \tag{4.4}$$

where $\boldsymbol{\lambda}$ are the eigenvalues corresponding to the eigenvectors $\mathbf{v}$.

By multiplying $\tilde{\mathbf{\Phi}}'_x$ to both sides of Equation 4.4, it becomes

$$\tilde{\mathbf{K}}\tilde{\mathbf{K}}\boldsymbol{\alpha} = N\boldsymbol{\lambda}\tilde{\mathbf{K}}\boldsymbol{\alpha} \tag{4.5}$$

where $\tilde{\mathbf{K}} = \tilde{\mathbf{\Phi}}'_x\tilde{\mathbf{\Phi}}_x$ and it is shown that a problem in $\tilde{\mathbf{K}}\boldsymbol{\alpha} = N\boldsymbol{\lambda}\boldsymbol{\alpha}$ yields all solution of Equation 4.5 (proved in [11], Lemma 21.1.3).

- Eigenvector $\mathbf{v}_m$ is normalized to be a unit vector. Then, it becomes:

$$\mathbf{v}_m = \sum_{n=1}^{N} \frac{\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}(\mathbf{x}_n) \tag{4.6}$$

(The proof of the normalizing factor is given in A.2 of appendix A).

Then, performing eigen-decomposition on the kernel matrix $\tilde{\mathbf{K}}$ gives $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$, which describe the eigenvectors in the feature space.

4.4 Composite Kernels

In EV, speaker supervector $\mathbf{s}$ is splitted into constituents $\mathbf{s}_r$ for calculating the distance between a given Gaussian and an observation frame $||\mathbf{s}_r - \mathbf{o}_t||^2$ required by the computation of expected log likelihood. In KEV, since speaker supervector is defined in the feature space only and there is no exact pre-image back to the input space, so we need to transform the observation $\mathbf{o}_t$ to the feature space too. Then, we can compute their dot product in the feature space and the Manhalonbis distance can be expressed in terms of the dot product. But, in this process, it needs the dot product between certain segment of a supervector and another vector (observation). If the whole supervector is put to a single Gaussian kernel in KPCA. Then, we can only obtain the dot product between the whole speaker supervector and another supervector. This raises a challenge in KEV. Composite kernel is the proposed solution. Each Gaussian constituent is mapped to its high dimensional space by a base kernel and the composite kernel is defined in Equation 4.7.

$$k(\mathbf{x}_i, \mathbf{x}_j) \quad = \quad k\left(\begin{bmatrix} \mathbf{x}_{i1} \\ \vdots \\ \mathbf{x}_{iR} \end{bmatrix}, \begin{bmatrix} \mathbf{x}_{i1} \\ \vdots \\ \mathbf{x}_{iR} \end{bmatrix} \right)$$

$$
\begin{aligned}
&= \begin{bmatrix} \varphi_1(\mathbf{x}_{i1}) \\ \vdots \\ \varphi_R(\mathbf{x}_{iR}) \end{bmatrix}' \begin{bmatrix} \varphi_1(\mathbf{x}_{j1}) \\ \vdots \\ \varphi_R(\mathbf{x}_{jR}) \end{bmatrix} \\
&= \sum_{r=1}^{R} k_r(\mathbf{x}_{ir}, \mathbf{x}_{jr}) \qquad (4.7)
\end{aligned}
$$

Any kernels, such as Gaussian kernel, polynomial kernel can be chosen as a base kernel. They are discussed in Section 4.6. The composite kernel is the summation of the base kernel and it is used for KPCA. In addition, since each constituent maps to the high dimensional feature space by its base kernel, the speaker supervector in the feature space can be splitted into constituents for both adaptation and recognition. A similar idea to composite kernel is discussed in [39].

4.5 KEV adaptation

The adaptation algorithm of KEV is the same as the conventional one except that the Manhalonbis distance measure is replaced by one expressed in terms of dot products in the feature space. In short, there are three major steps. Firstly, the auxiliary function is expressed as the dot products. Secondly, the parameters (weight) is initialized. Thirdly, generalized expectation maximization (GEM) is adopted for optimization. The details are as follows:

1. **Expression of the auxiliary function**

   Similar to the conventional eigenvoice, the auxiliary function is defined as the expected likelihood and it is further expressed in terms of weights of eigenvectors and dot products in the feature space. (The detailed derivation for Gaussian kernel and polynomial kernel are presented in Section 4.6.)

$$
Q_b(\mathbf{w}) = -\frac{1}{2}\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\left(d_1\log(2\pi) + \log|\mathbf{C}_r| + ||\mathbf{s}_r - \mathbf{o}_t||^2_{\mathbf{C}_r}\right) \qquad (4.8)
$$

   where s_r is the r-constituent of the speaker $\mathbf{s}$ (the speaker to be adapted), which is defined to be a linear combination of the eigenvectors in the feature space.

$$\tilde{\varsigma} \;=\; \sum_{m=1}^{M} w_m \mathbf{v}_m$$

Since the new speaker $\mathbf{s}$ is not found in the input space but only its image ς in the feature space is estimated as a linear combination of the kernel eigenvoices and $\tilde{\varsigma} is its centered version.$

By Equation 4.6,

$$\begin{aligned}
\tilde{\varsigma} &= \sum_{m=1}^{M}\sum_{n=1}^{N} \frac{w_m \alpha_{mn}}{\sqrt{\lambda_m}} \tilde{\varphi}(\mathbf{x}_n) \\
&= \begin{bmatrix} \sum_{m=1}^{M}\sum_{n=1}^{N} \frac{w_m \alpha_{mn}}{\sqrt{\lambda_m}} \tilde{\varphi}_1(\mathbf{x}_{n1}) \\ \vdots \\ \sum_{m=1}^{M}\sum_{n=1}^{N} \frac{w_m \alpha_{mn}}{\sqrt{\lambda_m}} \tilde{\varphi}_R(\mathbf{x}_{nR}) \end{bmatrix} \qquad (4.9)
\end{aligned}$$

2. **GEM**

Due to the non-linearity in KEV, no close form solution to the weights. Then, GEM is used instead of EM. In the M step of GEM, gradient ascent is used for improving the likelihood. The weights are updated by:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \eta(n)\mathbf{Q}'|_{\mathbf{w}=\mathbf{w}(n-1)} \qquad (4.10)$$

where
$\mathbf{Q}' = \left[\frac{\partial Q_b}{\partial w_1} \frac{\partial Q_b}{\partial w_2} \cdots \frac{\partial Q_b}{\partial w_m}\right]$ and
$\frac{\partial Q_b}{\partial w_j} = -\frac{1}{2}\sum_{r=1}^{R}\sum_{t=1}^{T} \gamma_{tr} \frac{\partial}{\partial w_j} ||\mathbf{s}_r - \mathbf{o}_t||^2_{\mathbf{C}_r}$ and
$\eta(n)$ is the learning rate at $n^{th} iteration$

3. **Initialization**

The weights are required to be initialized before the first iteration in the GEM. The SI model is a good choice for initialization due to its robustness. Therefore, it is suggested that the SI model is projected to each utilized eigenvector as initialization. The initial values of weights are derived as follows:

because

$$\mathbf{v}_i'\mathbf{v}_j = \begin{cases} 1 & \text{if } i = j; \\ 0 & \text{if } i \neq j. \end{cases}$$

$$\begin{aligned} \mathbf{v}_m'\tilde{\varphi}(\mathbf{x}^{(SI)}) &= \mathbf{v}_m' \sum_{m=1}^{M} w_m^{(SI)} \mathbf{v}_m \\ &= w_m^{(SI)}(\mathbf{v}_m'\mathbf{v}_m) \\ &= w_m^{(SI)} \end{aligned}$$

So,

$$\begin{aligned} w_m^{(SI)} &= \mathbf{v}_m'\tilde{\varphi}(\mathbf{x}^{(SI)}) \\ &= \sum_{n=1}^{N} \frac{\alpha_{mn}}{\sqrt{\lambda_m}} \tilde{\varphi}(\mathbf{x}_n)'\tilde{\varphi}(\mathbf{x}^{(SI)}) \\ &= \sum_{n=1}^{N} \frac{\alpha_{mn}}{\sqrt{\lambda_m}} \Big[\varphi(\mathbf{x}_n) - \bar{\boldsymbol{\varphi}}\Big]' \Big[\varphi(\mathbf{x}^{(SI)}) - \bar{\boldsymbol{\varphi}}\Big] \\ &= \sum_{n=1}^{N} \frac{\alpha_{mn}}{\sqrt{\lambda_m}} \Big[k(\mathbf{x}_n, \mathbf{x}^{(SI)}) + \frac{1}{N^2} \sum_{p=1}^{N} \sum_{q=1}^{N} k(\mathbf{x}_p, \mathbf{x}_q) - \\ & \quad \frac{1}{N} \sum_{p=1}^{N} \Big(k(\mathbf{x}_n, \mathbf{x}_p) + k(\mathbf{x}^{(SI)}, \mathbf{x}_p)\Big)\Big] \end{aligned} \tag{4.11}$$

However, it is noticed that the initialization in this projection method is not identical to the SI model because there is projection loss. This observation causes further investigation in Chapter 5.

4.6 Kernels

There are various types of kernel [20] such as Gaussian radial basis function (RBF) kernel (or in this thesis, simply call Gaussian kernel), polynomial kernel, exponential kernel and fisher kernel.

Two types of kernel are evaluated in this thesis, which are Gaussian kernel and polynomial kernel.

4.6.1 Gaussian kernel

1. **Definition**

   Gaussian kernel is defined as follows:

   $$k_r(\mathbf{x}_i, \mathbf{x}_j) = e^{-\beta_r(\mathbf{x}_i - \mathbf{x}_j)'\mathbf{C}_r{}^{-1}(\mathbf{x}_i - \mathbf{x}_j)} \tag{4.12}$$

   where β_r is a tunable parameter.

2. **Dot product in an unseen speaker and its derivative**

   The dot product between an observation and the corresponding constituent of an unseen speaker supervector ($\varsigma_r'\varphi_r(\mathbf{o}_t)$) is expressed in terms of the dot product between the observation and the corresponding constituent of the training speakers ($k_r(\mathbf{x}_{nr}, \mathbf{o}_t)$) and their weights.

   By B.2,

   $$\varsigma_r'\varphi_r(\mathbf{o}_t) = A(r,t) + \sum_{m=1}^{M} w_m B(m,r,t)$$

   By B.3, its derivative is:

   $$\frac{\partial \varsigma_r'\varphi_r(\mathbf{o}_t)}{\partial w_j} = B(j,r,t)$$

   where

   $$\begin{aligned} A(r,t) &= \frac{1}{N}\sum_{n=1}^{N} k_r(\mathbf{x}_{nr}, \mathbf{o}_t) \\ B(m,r,t) &= \sum_{n=1}^{N} \frac{\alpha_{mn}}{\sqrt{\lambda_m}}\Big(k_r(\mathbf{x}_{nr}, \mathbf{o}_t) - A(r,t)\Big) \end{aligned}$$

3. **Expression of distance measure**

   Then, the distance measure $||\mathbf{x}_{nr} - \mathbf{o}_t||^2_{\mathbf{C}_r}$ is expressed in terms of $k_r(\mathbf{x}_{nr}, \mathbf{o}_t)$ as follows:

   $$\begin{aligned} ||\mathbf{s}_r - \mathbf{o}_t||^2_{\mathbf{C}_r} &= -\frac{1}{\beta_r}\log \varsigma_r'\varphi_r(\mathbf{o}_t) \\ &= -\frac{1}{\beta_r}\log\Big(A(r,t) + \sum_{m=1}^{M} w_m B(m,r,t)\Big) \end{aligned} \tag{4.13}$$

4. **Expression of the auxiliary function**

   The auxiliary function is expressed in terms of $k_r(\mathbf{x}_{nr}, \mathbf{o}_t)$ as follows:

$$\begin{aligned} Q_b(\mathbf{w}) &= -\frac{1}{2}\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\Big(d_1\log(2\pi) + \log|\mathbf{C}_r| + ||\mathbf{s}_r - \mathbf{o}_t||^2_{\mathbf{C}_r}\Big) \\ &= -\frac{1}{2}\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\Big(d_1\log(2\pi) + \log|\mathbf{C}_r| - \\ &\quad \frac{1}{\beta_r}\log\Big(A(r,t) + \sum_{m=1}^{M} w_m B(m,r,t)\Big)\Big) \end{aligned} \tag{4.14}$$

5. **The first derivative is:**

$$\begin{aligned} \frac{\partial Q_b(\mathbf{w})}{\partial w_j} &= \frac{1}{2}\sum_{r=1}^{R}\sum_{t=1}^{T}\frac{\gamma_{tr}}{\beta_r}\frac{\frac{\partial}{\partial w_j}\varsigma_r'\varphi_r(\mathbf{o}_t)}{\varsigma_r'\varphi_r(\mathbf{o}_t)} \\ &= \frac{1}{2}\sum_{r=1}^{R}\sum_{t=1}^{T}\frac{\gamma_{tr}}{\beta_r}\frac{B(j,r,t)}{\varsigma_r'\varphi_r(\mathbf{o}_t)} \end{aligned} \tag{4.15}$$

4.6.2 Polynomial kernel

1. **Definition**

   Polynomial kernel is defined as follows:

$$k_r(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i'\mathbf{C}_r^{-1}\mathbf{x}_j + 1)^d \tag{4.16}$$

   where d is the polynomial degree.

2. **Dot product in an unseen speaker and its derivative**

   In addition to $\varsigma_r'\varphi_r(\mathbf{o}_t)$ and $\frac{\partial\varsigma_r'\varphi_r(\mathbf{o}_t)}{\partial w_j}$ mentioned in subsection 4.6.1, item 2, $\varsigma_r'\varsigma_r$ and $\frac{\partial\varsigma_r'\varsigma_r}{\partial w_j}$ are expressed as follows:

   By B.4,

$$\varsigma_r'\varsigma_r = \sum_{m=1}^{M}\sum_{m'=1}^{M} w_m w_{m'} D(m,m',r) + \sum_{m=1}^{M} w_m E(m,r) + F(r)$$

   By B.5,

$$\frac{\partial\varsigma_r'\varsigma_r}{\partial w_j} = E(j,r) + \sum_{m=1}^{M} 2w_m D(m,j,r)$$

where

$$
\begin{aligned}
D(m,m',r) &= \sum_{n=1}^{N}\sum_{n'=1}^{N} \frac{\alpha_{mn}\alpha_{m'n'}}{\sqrt{\lambda_m \lambda_{m'}}} \\
&\quad \Big[k_r(\mathbf{x}_{nr},\mathbf{x}_{n'r}) - \frac{1}{N}\sum_{i=1}^{N}\Big[k_r(\mathbf{x}_{nr},\mathbf{x}_{ir}) + k_r(\mathbf{x}_{n'r},\mathbf{x}_{ir})\Big] + F(r) \\
E(m,r) &= 2\sum_{n=1}^{N}\frac{\alpha_{mn}}{\sqrt{\lambda_m}}\Big[\frac{1}{N}\sum_{i=1}^{N}k_r(\mathbf{x}_{nr},\mathbf{x}_{ir}) - F(r)\Big] \\
F(r) &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}k_r(\mathbf{x}_{ir},\mathbf{x}_{jr})
\end{aligned}
$$

3. **Expression of distance measure**

Then, the distance measure is expressed in terms of $\varsigma_r'\varsigma_r$ and $\varsigma_r'\varphi_r(\mathbf{o}_t)$ as:

$$||\mathbf{s}_r - \mathbf{o}_t||^2_{\mathbf{C}_r} = ||\mathbf{s}_r||^2_{\mathbf{C}_r} + ||\mathbf{o}_t||^2_{\mathbf{C}_r} - \mathbf{s}_r'\mathbf{C}_r^{-1}\mathbf{o}_t - \mathbf{o}_t'\mathbf{C}_r^{-1}\mathbf{s}_r \tag{4.17}$$

where

$||\mathbf{s}_r||^2_{\mathbf{C}_r} = \mathbf{s}_r'\mathbf{C}_r^{-1}\mathbf{s}_r = \Big[\varsigma_r'\varsigma_r\Big]^{\frac{1}{d}} - 1$

$\mathbf{s}_r'\mathbf{C}_r^{-1}\mathbf{o}_t = \mathbf{o}_t'\mathbf{C}_r^{-1}\mathbf{s}_r = \Big[\varsigma_r'\varphi_r(\mathbf{o}_t)\Big]^{\frac{1}{d}} - 1$

(Since it is assumed that C_r is a symmetric matrix, $\mathbf{s}_r'\mathbf{C}_r^{-1}\mathbf{o}_t = \mathbf{o}_t'\mathbf{C}_r^{-1}\mathbf{s}_r$.)

So,

$$||\mathbf{s}_r - \mathbf{o}_t||^2_{\mathbf{C}_r} = \Big[\varsigma_r'\varsigma_r\Big]^{\frac{1}{d}} + ||\mathbf{o}_t||^2_{\mathbf{C}_r} - 2\Big[\varsigma_r'\varphi_r(\mathbf{o}_t)\Big]^{\frac{1}{d}} + 1 \tag{4.18}$$

4. **Expression of the auxiliary function**

The auxiliary function is expressed in terms of $\varsigma_r'\varsigma_r$ and $\varsigma_r'\varphi_r(\mathbf{o}_t)$ as follows:

$$
\begin{aligned}
Q_b(\mathbf{w}) &= -\frac{1}{2}\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\Big(d_1\log(2\pi) + \log|\mathbf{C}_r| + ||\mathbf{s}_r - \mathbf{o}_t||^2_{\mathbf{C}_r}\Big) \\
&= -\frac{1}{2}\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\Big(d_1\log(2\pi) + \log|\mathbf{C}_r| + \\
&\quad \Big[\varsigma_r'\varsigma_r\Big]^{\frac{1}{d}} + ||\mathbf{o}_t||^2_{\mathbf{C}_r} - 2\Big[\varsigma_r'\varphi_r(\mathbf{o}_t)\Big]^{\frac{1}{d}} + 1\Big)
\end{aligned} \tag{4.19}
$$

5. **The first derivative is:**

$$
\begin{aligned}
\frac{\partial Q_b(\mathbf{w})}{\partial w_j} &= -\frac{1}{2d}\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\Bigg[\varsigma_r'\varsigma_r^{(\frac{1}{d}-1)}\frac{\partial}{\partial w_j}\varsigma_r'\varsigma_r - \\
&\quad 2\varsigma_r'\varphi_r(\mathbf{o}_t)^{(\frac{1}{d}-1)}\frac{\partial}{\partial w_j}\varsigma_r'\varphi_r(\mathbf{o}_t)\Bigg] \\
&= -\frac{1}{2d}\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\Bigg[\varsigma_r'\varsigma_r^{(\frac{1}{d}-1)}\Big(E(j,r) + \sum_{m=1}^{M} 2w_m D(m,j,r)\Big) - \\
&\quad 2\varsigma_r'\varphi_r(\mathbf{o}_t)^{(\frac{1}{d}-1)}B(j,r,t)\Bigg] \qquad (4.20)
\end{aligned}
$$

4.6.3 Contour plots

In order to have a feeling of the output of the KPCA, some contour plots on different kernels are presented in order to compare with the linear PCA. It contains data points in the original input space and contour lines, which mean principal component values are constant along the line and they are orthogonal to the eigenvectors. A toy example with 3 Gaussian clusters in two dimensions is presented, which is provided by [24]. The contour plots corresponding to the first few eigenvectors of the linear kernel (equivalent to linear PCA), the Gaussian kernel and the polynomial kernel (of degree three and four) are presented in Figures 4.2, 4.3, 4.4 and 4.5 respectively. There are two non-zero eigenvectors for the linear kernel and three non-zero eigenvectors for the polynomial kernel.

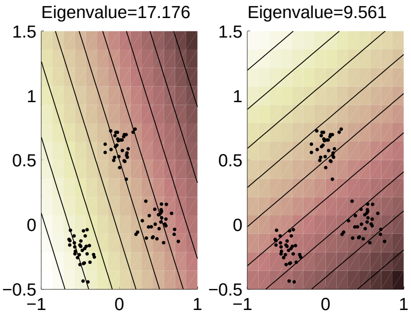

Figure 4.2: Contour plot of the linear PCA (The x-axis and y-axis are the dimension 1 and 2 respectively)

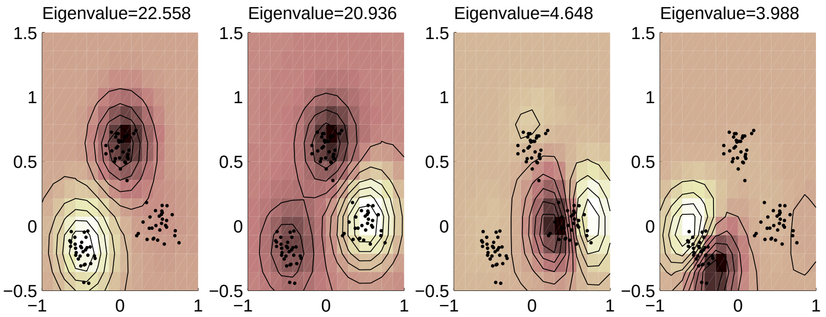

Figure 4.3: Contour plot of the KPCA - Gaussian kernel ($K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{0.1}}$)

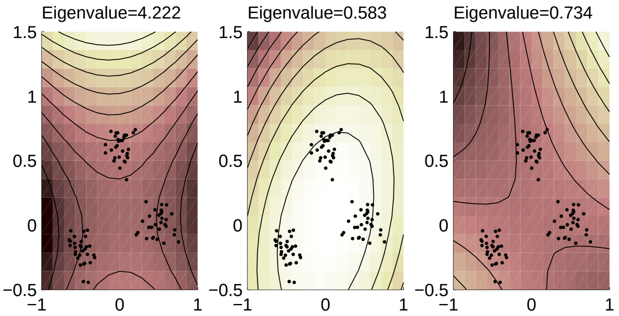

Figure 4.4: Contour plot of the KPCA - polynomial kernel in power 2 ($K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i' x_j)^2$)

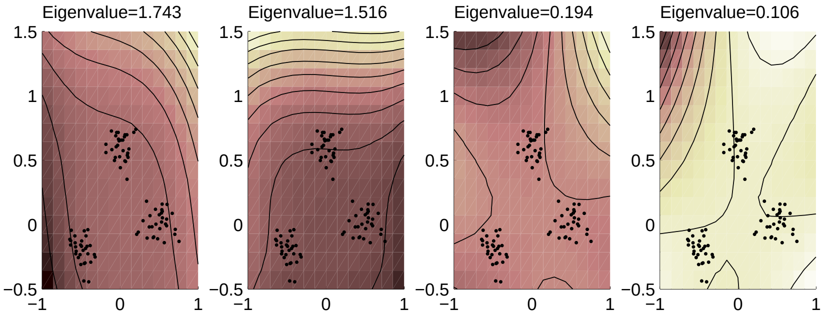

Figure 4.5: Contour plot of the KPCA - polynomial kernel in power 3 ($K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i' x_j)^3$)

4.7 Time complexity

In both the adaptation algorithm and the recognition algorithm, the calculation of the dot product dominates the computation. Therefore, they are chosen as the measure of the time complexity of the algorithm. For the kernel computation in both the Gaussian kernel and the polynomial kernel, it is essential to compute $k_r(\mathbf{x}_{nr}, \mathbf{o}_t)$ in adaptation (for $n = 1 \cdots N$, $r = 1 \cdots R$ and $t = 1 \cdots T$) so that its time complexity is $O(NRT)$ (where N is the number of training speakers, R is the total number of states in the HMMs and T is the number of frames in the utterances. In recognition, without pruning, its overall time complexity is $O(NRT)$ again.

4.8 KEV adaptation experiment

In Gaussian kernel, β is a tunable parameter (all β_r equal to a single β in the experiment). For tuning, 10 speakers are sampled from the training-set. Around 4 seconds of adaptation data are selected for each speaker. KEV is performed on several beta values. The recognition accuracy against beta value is then plotted. The tuning results are in figure 4.6 and $\beta = 0.0005$ is the best and it is chosen for the rest of the experiments.

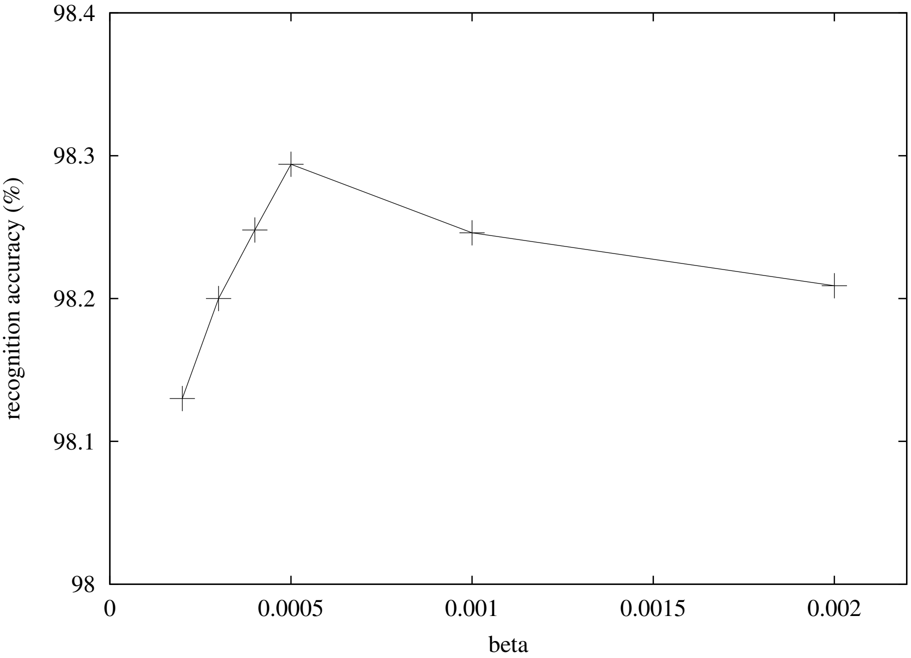

Figure 4.6: Tuning result of beta value for Gaussian kernel

The Gaussian kernel and the polynomial kernel with power three were adopted. Only 10-second adaptation-set were done for comparison. This is shown in Figure 4.7. The results show that the Gaussian kernel was better than the polynomial kernel when the number of eigenvoices is more than one. A possible reason is that the model we used is HMM where each state is in the form of Gaussian. A Gaussian kernel is then a reasonable choice for KEV.

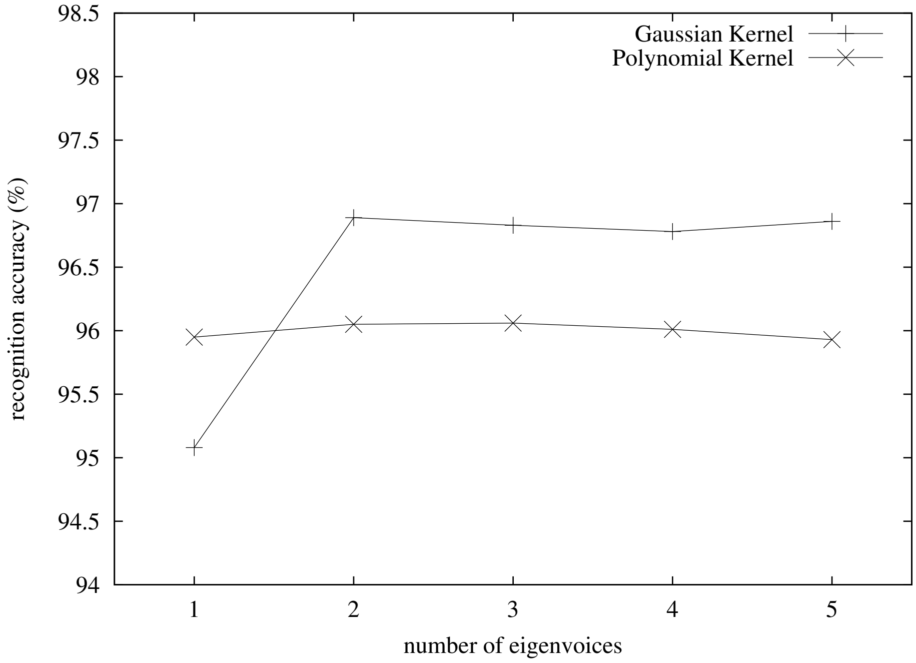

Figure 4.7: Comparison of the Gaussian kernel and polynomial kernel

Through the previous experiment, it is believed that Gaussian kernel is a reasonably good choice for KEV. Although it is possible that other kernels could give even better results, the focus of this thesis is on the framework of KEV. Therefore, the Gaussian kernel is taken for further investigation for the rest of this paper. In Figure 4.8, it shows that the KEV on 2-second, 4-second and 10-second adaptation-set with various number of eigenvoices from one to ten.

The results show that the base KEV outperforms the SI model when the number of eigenvoices exceed two. The relative error rate reduction (ERR) of the base KEV from SI model is 16%, 21% and 21% for 2-second, 4-second and 10-second adaptation-sets respectively while the ERR of the base KEV from base EV is 28%, 32% and 32% for 2-second, 4-second and 10-second adaptation-sets respectively. There are two observations. Firstly, the KEV with only one eigenvoice is the worst. This is due to the fact that there are too few parameters for estimation and there is a projection loss from the SI model in the initialization. Due to this weakness, some solutions are proposed and discussed in Chapter 5. Secondly, it is noticed that the recognition accuracy saturates quickly as the number of eigenvoices increases.

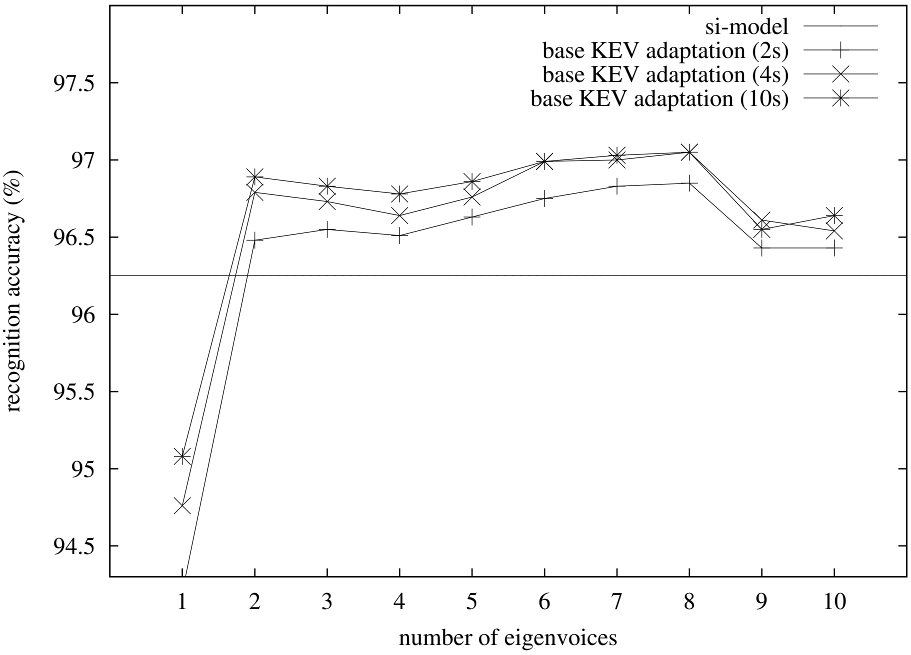

Figure 4.8: Summary of the recognition results of the base KEV adaptation

CHAPTER 5

ROBUST KEV

5.1 Robust KEV 1 - addition approach

The major weakness of KEV is that it does not guarantee that the eigenspace spanned contains a speaker model not worse than the SI one (in terms of likelihood) for the given adaptation data, especially when the number of eigenvectors decreases (smaller the eigenspace spanned). It is possible that some unseen speakers in test-sets who cannot be well represented by a combination of the eigenvoices. The speaker-adapted model can then perform worse than the SI-model.

The first proposal is to include the SI supervector in the optimization. It means that, in addition to the kernel eigenspace obtained by KPCA, the SI supervector is treated as a compulsory component. It could ensure the result in optimization is not worse than the SI one (in terms of the likelihood).

A speaker supervector is defined as:

$$\tilde{\varsigma} = \tilde{\varphi}(\mathbf{x}^{(SI)}) + \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{w_m \alpha_{mn}}{\sqrt{\lambda_m}} \tilde{\varphi}(\mathbf{x}_n) \tag{5.1}$$

By equation C.2,

$$||\mathbf{s}_r - \mathbf{o}_t||^2 \quad = \quad -\frac{1}{\beta_r} \log\Big(k_r(\mathbf{x}_r^{(SI)}, \mathbf{o}_t) + \sum_{m=1}^{M} w_m B(m,r,t) \Big)$$

By equation C.3, $\frac{\partial \varsigma_r' \varphi_r(\mathbf{o}_t)}{\partial w_j}$ is the same as the one in base KEV. Then, the first derivative of $Q_b(\mathbf{w})$ also remains unchanged too. The initialization of both KEV has to be modified as follows:

$$w_i = \begin{cases} 0 & \text{for } i = 1 \cdots M \\ 1 & \text{for } i = 0 \end{cases}$$

5.2 Robust KEV 2 - interpolation approach

In proposal 1, adding the SI model is, in fact, a specific case of interpolation. Therefore, an extension to an interpolation was proposed and investigated. There are four modifications, which are the definition of the speaker supervector, distance expression, the first derivative of the auxiliary function for the gradient ascent and weights initialization.

- **Definition of the speaker supervector**

  A speaker supervector is defined as an interpolation between the SI model and the KEV-adapted model:

$$\tilde{\varsigma} = w_0 \tilde{\varphi}(\mathbf{x}^{(SI)}) + (1 - w_0) \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{w_m \alpha_{mn}}{\sqrt{\lambda_m}} \tilde{\varphi}(\mathbf{x}_n) \tag{5.2}$$

- **Distance expression**

  The distance expression becomes:

$$||\mathbf{s}_r - \mathbf{o}_t||^2_{\mathbf{C}_r} \quad = \quad -\frac{1}{\beta_r} \log\Big(\varsigma'_r \varphi_r(\mathbf{o}_t)\Big)$$

  where $\varsigma'_r \varphi_r(\mathbf{o}_t) = w_0 k_r(\mathbf{x}_r^{(SI)}, \mathbf{o}_t) + (1 - w_0)\Big[A(r,t) + \sum_{m=1}^{M} w_m B(m,r,t)\Big]$ by D.3.

- **The first derivative of the auxiliary function**

  By 4.15, for $i = 0 \cdots M$,

$$\frac{\partial Q_b(\mathbf{w})}{\partial w_i} \quad = \quad \frac{1}{2} \sum_{r=1}^{R} \sum_{t=1}^{T} \frac{\gamma_{tr}}{\beta_r} \frac{\frac{\partial}{\partial w_i} \varsigma'_r \varphi_r(\mathbf{o}_t)}{\varsigma'_r \varphi_r(\mathbf{o}_t)}$$

  By D.4 and D.5,

$$\begin{aligned} \frac{\partial \varsigma'_r \varphi_r(\mathbf{o}_t)}{\partial w_0} \quad &= \quad k_r(\mathbf{x}_r^{(SI)}, \mathbf{o}_t) - A(r,t) - \sum_{m=1}^{M} w_m B(m,r,t) \\ \frac{\partial \varsigma'_r \varphi_r(\mathbf{o}_t)}{\partial w_j} \quad &= \quad (1 - w_0) B(j,r,t) \end{aligned}$$

where

$$
\begin{aligned}
A(r,t) &= \frac{1}{N}\sum_{n=1}^{N} k_r(\mathbf{x}_{nr},\mathbf{o}_t) \\
B(m,r,t) &= \sum_{n=1}^{N} \frac{\alpha_{mn}}{\sqrt{\lambda_m}}\Big(k_r(\mathbf{x}_{nr},\mathbf{o}_t) - A(r,t)\Big)
\end{aligned}
$$

By substituting D.4 into 4.15 and D.5 into 4.15,

$$
\frac{\partial Q_b(\mathbf{w})}{\partial w_0} = \frac{1}{2}\sum_{r=1}^{R}\sum_{t=1}^{T}\frac{\gamma_{tr}}{\beta_r}\frac{k_r(\mathbf{x}_r^{(SI)},\mathbf{o}_t) - A(r,t) - \sum_{m=1}^{M} w_m B(m,r,t)}{\varsigma'_r\varphi_r(\mathbf{o}_t)} \quad (5.3)
$$

$$
\frac{\partial Q_b(\mathbf{w})}{\partial w_j} = \frac{1}{2}\sum_{r=1}^{R}\sum_{t=1}^{T}\frac{\gamma_{tr}}{\beta_r}\frac{(1-w_0)B(j,r,t)}{\varsigma'_r\varphi_r(\mathbf{o}_t)} \quad (5.4)
$$

- **Weights initialization**

  The initialization of weights w_j (for $j = 1 \cdots M$) is the same as the one in base KEV (projection method) and the weight w_o is initialized to be 0.5.

5.3 Robust EV

To have a fair comparison on the EV and KEV, a robust EV is proposed. Similar to the KEV, we have the "addition" approach and the "interpolation" approach.

- **robust EV 1 - addition approach**

  The speaker definition is modified as in equation 5.5.

$$
s = \mathbf{x}^{(SI)} + \sum_{m=1}^{M} w_m\tilde{\mathbf{e}}_m \quad (5.5)
$$

  After differentiate Q_b with respect to w_j and set it zero, the solution becomes:

$$
\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\tilde{\mathbf{e}}'_{jr}\mathbf{C}_r^{-1}(\mathbf{o}_t - \mathbf{x}^{(SI)}) = \sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\Big(\sum_{m=1}^{M} w_m\tilde{\mathbf{e}}'_{mr}\mathbf{C}_r^{-1}\tilde{\mathbf{e}}_{jr}\Big) \quad (5.6)
$$

- **robust EV 2 - interpolation approach**

  The speaker definition is:

$$
s = w_0\mathbf{x}^{(SI)} + (1 - w0)\Big(\bar{\mathbf{e}}_r + \sum_{m=1}^{M} w_m\mathbf{e}_{mr}\Big) \quad (5.7)
$$

After differentiate Q_b with respect to w_j and set it zero, the solution becomes:

$$\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\tilde{\mathbf{e}}'_{jr}\mathbf{C}_r^{-1}\Big(\mathbf{o}_t-(1-w_0)\bar{\mathbf{e}}_r-w_0\mathbf{x}^{(SI)}\Big)=\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{tr}\Big(\sum_{m=1}^{M}w_m\tilde{\mathbf{e}}'_{mr}\mathbf{C}_r^{-1}\tilde{\mathbf{e}}_{jr}\Big) \tag{5.8}$$

The weights w_j (for $j = 1\cdots M$) are solved analytically while w_0 is simply found exhaustively.

5.4 Experimental results

Since it does not guarantee that the base KEV is better than the SI one in terms of likelihood, two solutions are proposed in the last two sections. Experiments are conducted in 2-second, 4-second and 10-second adaptation-sets. Comparisons of recognition results of the base KEV and the two suggested methods are summarized in Figures 5.4, 5.5 and 5.6 respectively.

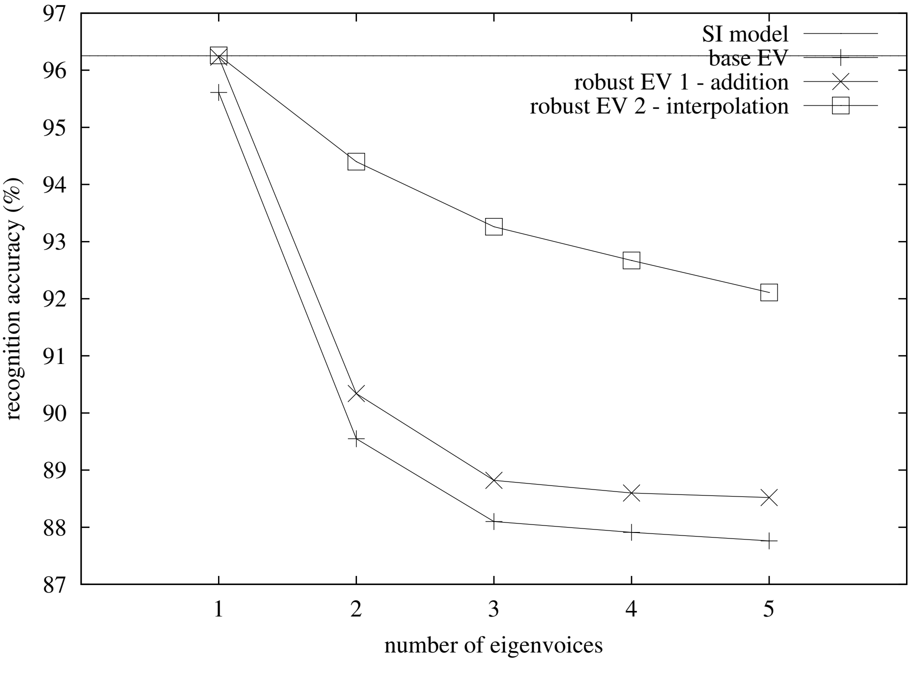

Figure 5.1: Comparison of the recognition results of the base EV and robust EV adaptations in 2-second adaptation-set

These figures show that both proposed modifications outperform the base KEV while the robust KEV 2, which interpolates the SI supervector with the KEV one, is the best. The robust KEV 2 is slightly better than the robust KEV

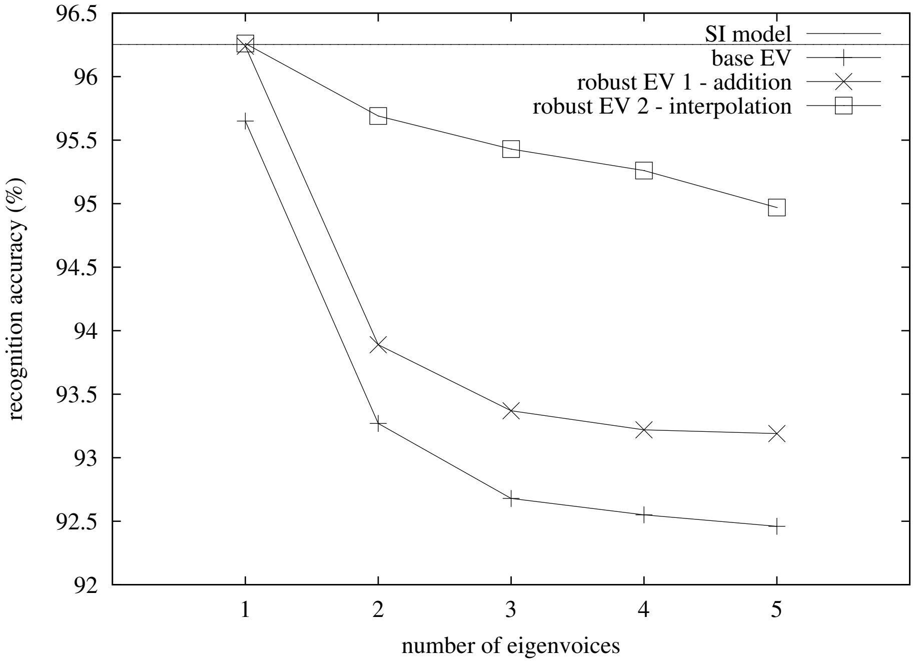

Figure 5.2: Comparison of the recognition results of the base EV and robust EV adaptations in 4-second adaptation-set

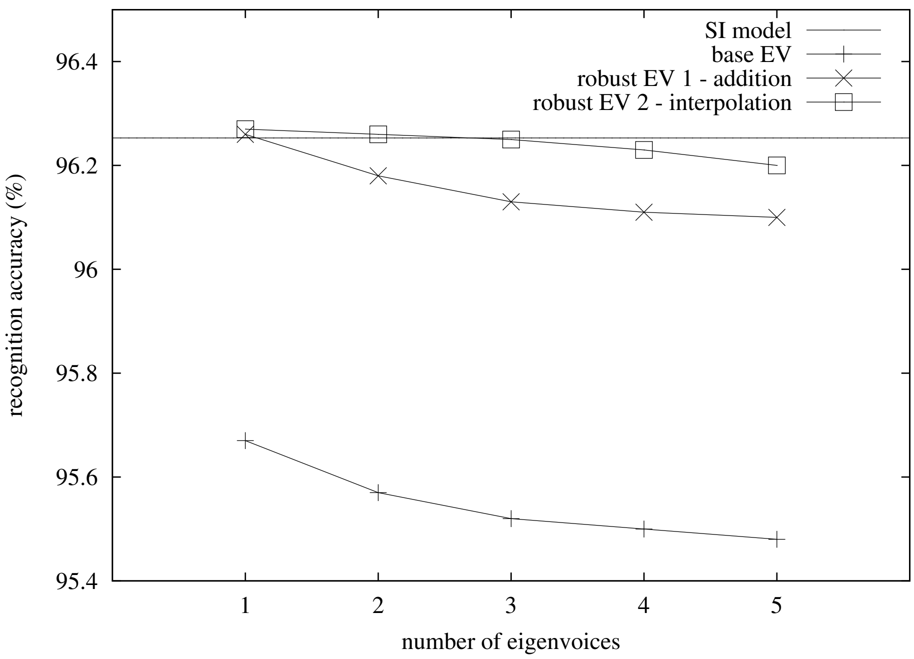

Figure 5.3: Comparison of the recognition results of the base EV and robust EV adaptations in 10-second adaptation-set

1 in small amounts of adaptation data (2-second adaptation-set). As the amount of adaptation data increases, the robust KEV 2 has a greater advantage over the robust KEV 1.

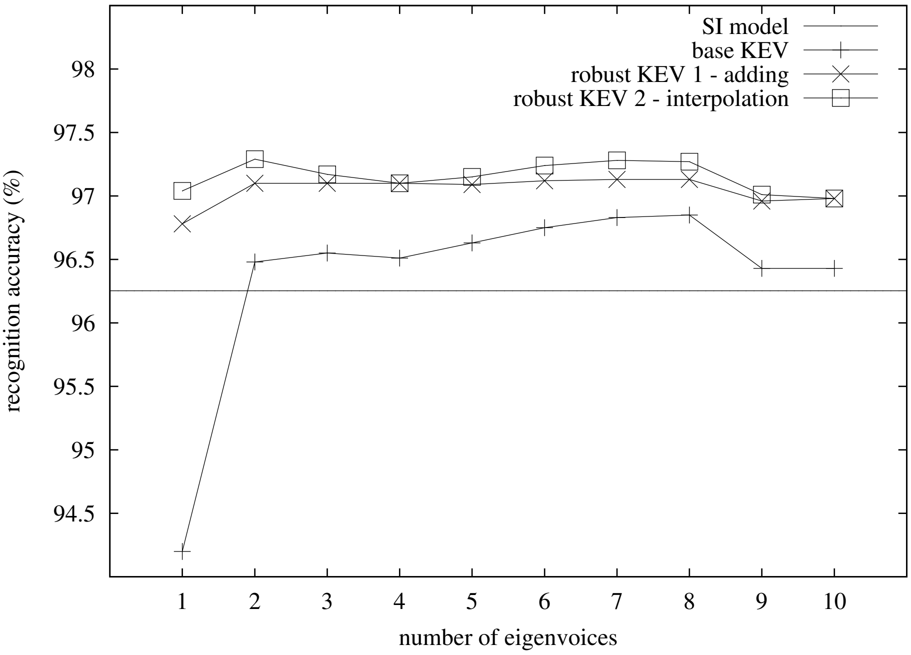

Figure 5.4: Comparison of the recognition results of the base KEV and robust KEV adaptations in 2-second adaptation-set

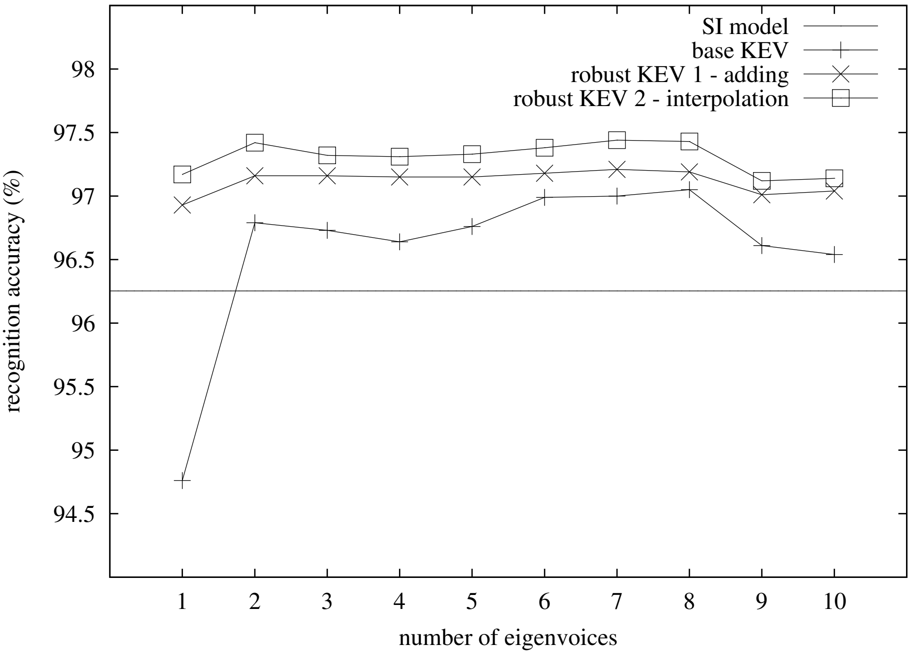

Figure 5.5: Comparison of the recognition results of the base KEV and robust KEV adaptations in 4-second adaptation-set

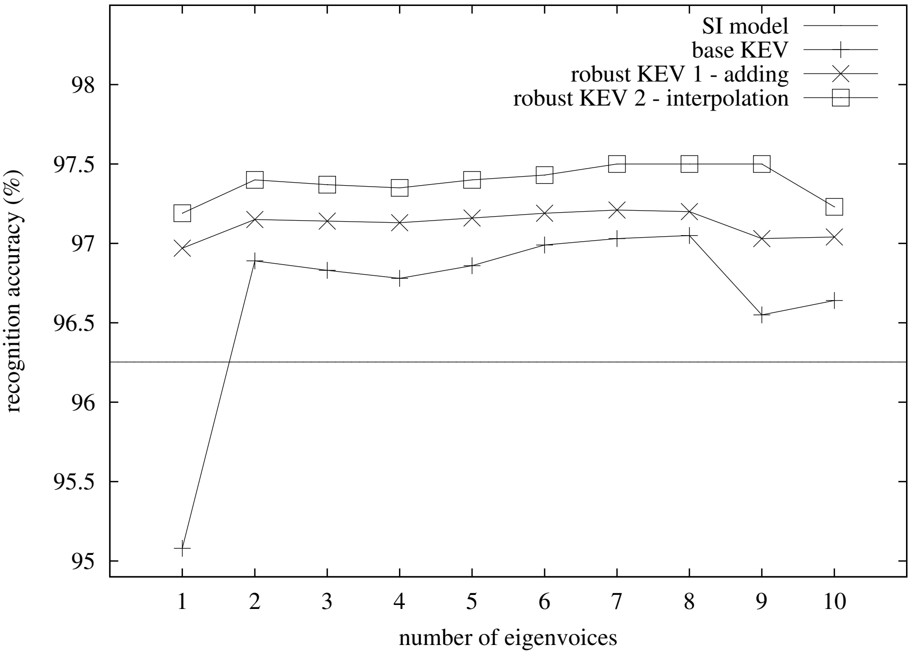

Figure 5.6: Comparison of the recognition results of the base KEV and robust KEV adaptations in 10-second adaptation-set

CHAPTER 6

FURTHER INVESTIGATION

6.1 KEV versus conventional adaptation methods

In this chapter, KEV is compared with conventional adaptation methods including MAP and MLLR. For MLLR adaptation, a global transformation matrix is estimated since it is conducted with small amounts of adaptation data. MLLR with a full transformation matrix or a diagonal transformation matrix are tried. In MAP adaptation, there is a back-off scaling factor. Various values are tried and the best results are presented. For KEV, the best results (among experiments on different numbers of eigenvectors) are summarized. The comparison is presented in Figure 6.1. All the above experiments are done under supervised conditions.

amount of adaptation data	**MLLR**		**MAP**	**base EV**	**robust EV 2**	**base KEV**	**robust KEV 2**
	full	diag.					
2s	96.16%	96.16%	95.50%	95.61%	96.26%	96.85%	97.28%
4s	96.06%	96.15%	95.63%	95.65%	96.26%	97.05%	97.44%
10s	97.56%	96.24%	96.47%	95.67%	96.27%	97.05%	97.50%

Table 6.1: Comparison of recognition accuracies of various adaptation methods

Overall, MAP gives the worst performance due to the fact that MAP requires a lot of adaptation data. The MLLR gives good results in a 10-second adaptation-set. However, all three variations of KEV outperform both MAP and MLLR in 2-second and 4-second adaptation-sets. This shows that KEV is good in rapid speaker adaptation. Out of the three variations, robust KEV 2, which incorporates the SI model by interpolation, is found to be the best.

6.2 Significant tests

Section 6.1 shows that KEV gives promising results in small amounts of adaptation data. However, when comparing several adaptation methods, it is important

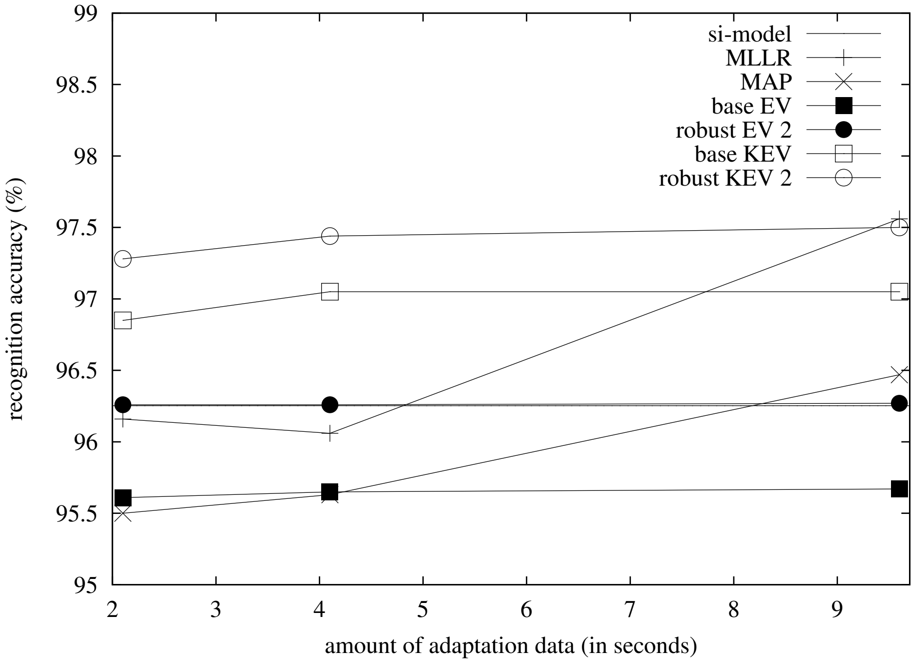

Figure 6.1: Comparison of the various adaptation methods

to understand how significant the performance gains are between one system and another. Therefore, significance tests are done. A software for significance test by the National Institute of Standards and Technology (NIST) is used. A two-tail 5% significant level is used. The results are shown in Table F.1, F.2 and F.3 in appendix F.

6.3 Analysis of the eigenvectors

In both EV and KEV, it is expected that the extracted components represent certain kinds of inter-speaker variations such as gender, age, accents and so on. Therefore, in this section, we try to analyze the relationship between the weights of the components and the inter-speaker variations in a qualitative manner.

First of all, there are 326 speakers in the corpus, 163 speakers for each of the training-sets and test-sets. PCA and KPCA are performed on the 163 training speakers for EV (base-EV with 2 eigenvectors) and KEV (base-KEV with 8 eigenvectors) respectively and two sets of experiments are conducted (the chosen configurations give the best recognition accuracy in base-EV and base-KEV respectively). The first set of experiments is to project the SD supervectors of the training speakers on the eigenvectors. The relationship between the projected weight of the component and the inter-speaker variations are analyzed. Scatter graphs (each speaker represents a point) on the first and second weights are plotted. Speakers are grouped as "girl", "boy", "man" or "woman". The scatter plot of the first two eigenvectors shows whether EV and KEV can extract inter-speaker variations successfully.

The above experiments are an ideal analysis on the components obtained by PCA and KPCA. Therefore, the second set of experiments compare the weights estimated by EV and KEV adaptation against the inter-speaker variations. Although it is expected that the class could become more confused in the test-set, it is meaningful to see if the distribution in the scatter plot of the training-set is similar to that of the test-set. The first set of experiments are treated as an analysis on the extraction of component while the second one is treated as an analysis on whether the adaptation can make use of extracted components.

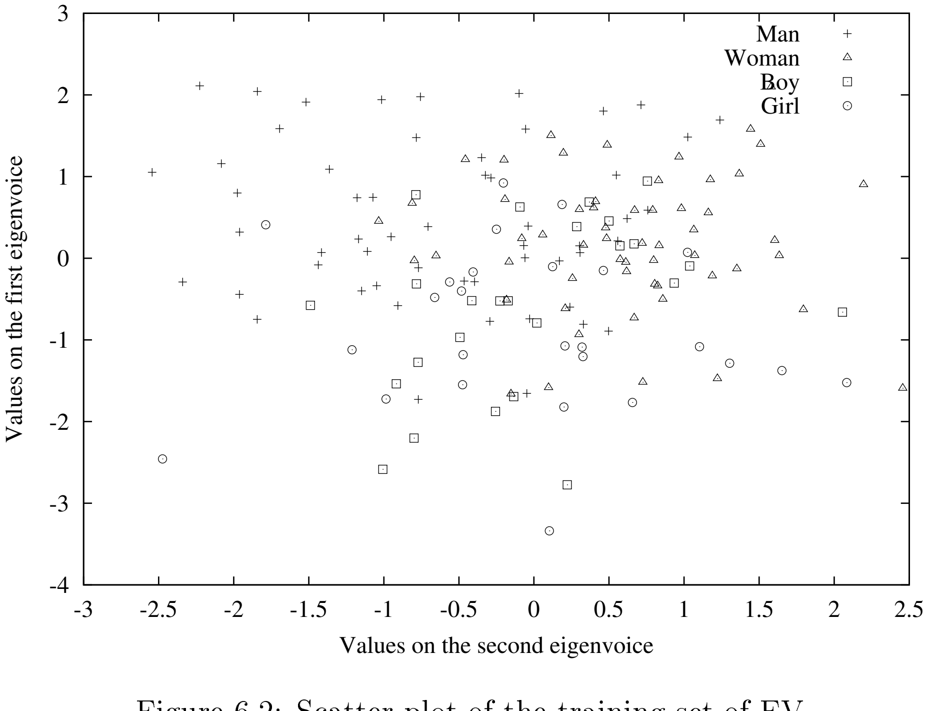

Figure 6.2: Scatter plot of the training-set of EV

The scatter plot of the training-set is in Figure 6.2. It is found that "boy" and "girl" speakers are too diverse and it makes the scatter plot un-readable. It means that the first two eigenvector does not help in distinguishing boy and girl from man and woman. Therefore, only "man" and "woman" speakers are plotted in 6.2 (for a training-set again).

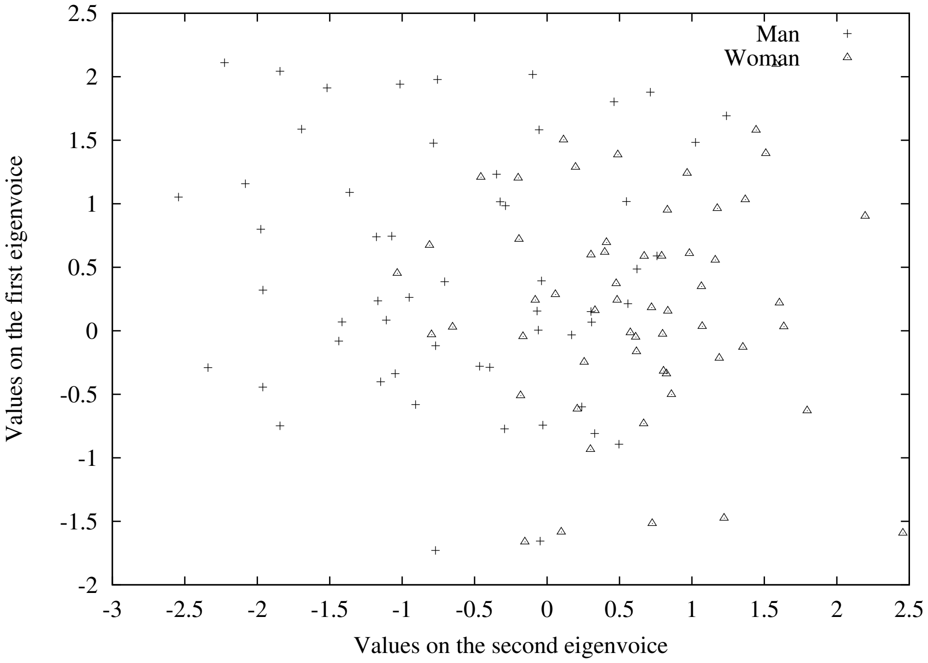

Figure 6.3: Scatter plot of the training-set of EV

It is found that, in EV, the second eigenvector separates the "man" and "woman" speakers ("man" is on the left while "woman" is on the right).

In order to see if the adaptation is making use of this information, the scatter plot of a test-set is plotted in Figure 6.4 (only "man" and "woman" speakers are plotted). The trend holds but it is more confused in the middle.

Similar to EV, the scatter plot of the training-set and the test-set of KEV are plotted in Figures 6.5 and 6.6.

It is found that the distribution of "man" and "woman" are separated clearly in the first eigenvectors of KEV ("woman" is on the top-right corner and "man" is in the bottom) while the children (both "boy" and "girl" subsets) are in the top-left corner. "Woman" and "kids" are slightly confused. "Boy" and "girl" are confused a lot. That observation matches the idea that children's voices are more similar to women's voices rather than men and children's gender is difficult to distinguish by their voices.

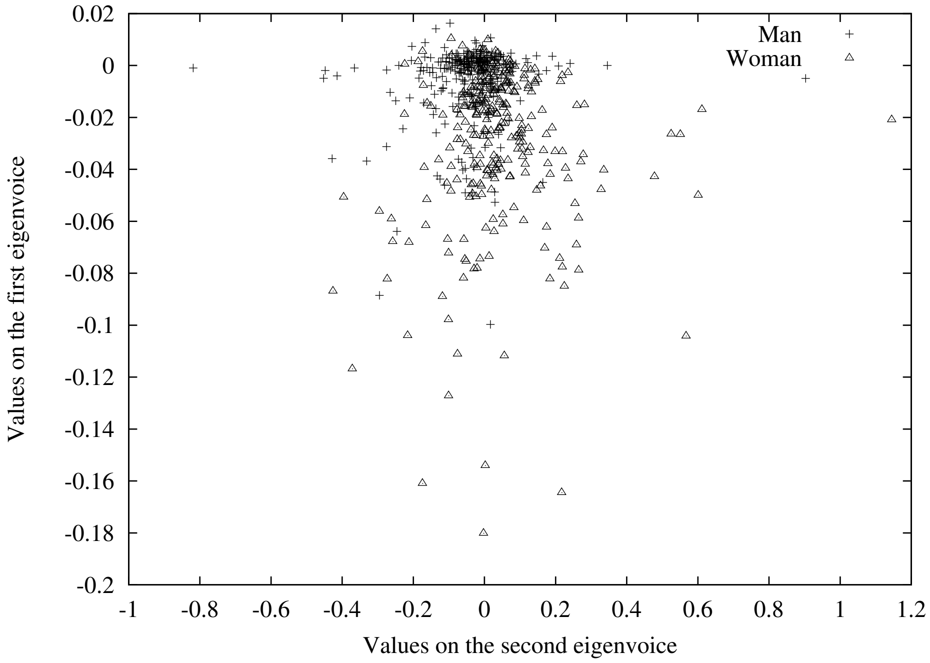

Figure 6.4: Scatter plot of the test-set of EV

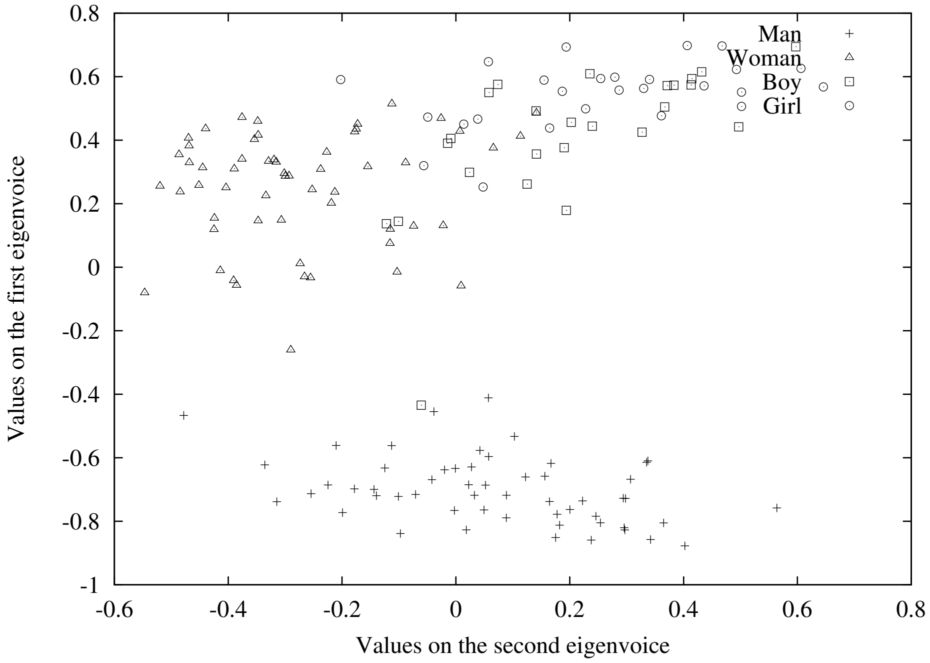

Figure 6.5: Scatter plot of the training-set of KEV

In the study of the relationship between the four groups and the eigenvectors, it implicitly investigates the gender and age effect in a blurred way. In addition, the effect of accents is also studied but no clear correlation can be found.

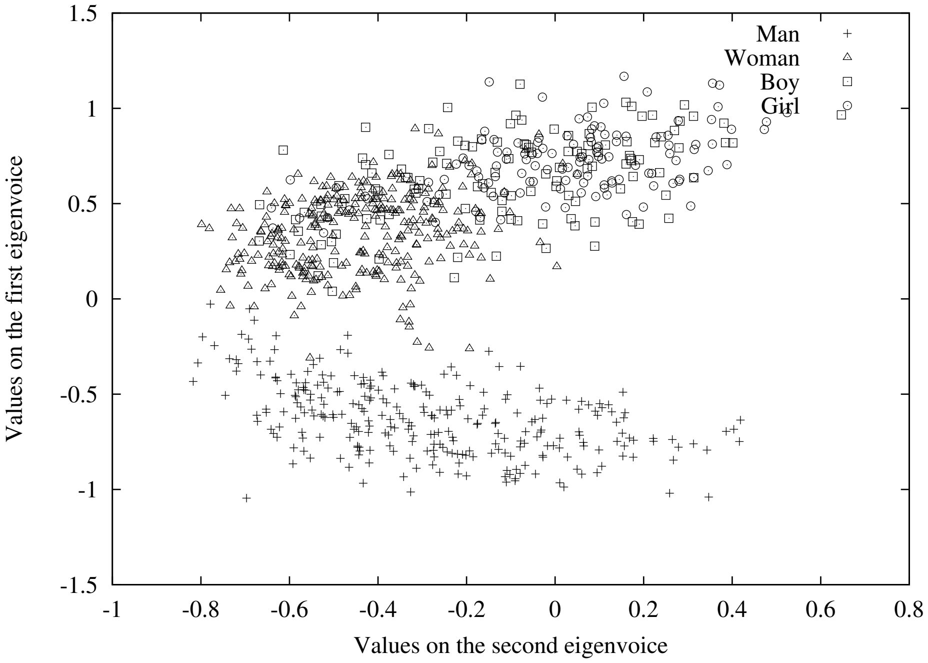

Figure 6.6: Scatter plot of the test-set of KEV

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Contribution

We have made several contribution in this thesis. Firstly, KEV is proposed which is a non-linear generalization of EV. By using different base kernel, it can handle different kinds of data distribution. This enhances the capability of eigenvoice family. Secondly, the derivation of the formula for the Gaussian kernel and polynomial kernel is conducted. The major works include expressing the Manhalonbis distance in the input space in terms of dot products in the feature space. In addition, if we put the whole supervectors in a single non-linear kernel in the KPCA, dot products between certain segment of a supervector and another vector are unable to be calculated which is needed by the computation of Manhalonbis distance. Therefore, the composite kernel is proposed to solve this specific problem in KEV speaker adaptation. Thirdly, due to the observation that KEV speaker adapted model does not guarantee to be better than the SI model, robust KEV is investigated, which combines the SI model and the KEV adapted model. It is showed that robust KEV improves its robustness in small amount of adaptation data. Fourthly, eigenvoice analysis in scatter plot is used to study the relationship between the extracted eigenvoice and the underlying inter-speaker variations. It shows that the first two eigenvectors in KEV captures gender and age in this digit recognition task.

7.2 Conclusion

In this thesis, EV has been revised. It is found that the correlation approach is better than the covariance approach because it avoids some features with large dominating values. Both EV and SI model are taken for comparing with the KEV. However, EV does not show improvement. A possible reason is that linear

PCA may not be effective enough for this digit recognition problem. This is the reason for proposing KEV. The importance of KEV is to generalize EV from a linear manner to a non-linear one so as to enhance its capability on different problems. In establishing the KEV architecture, the major difficulty is to map the feature space eigenvoices to observation space. Composite kernel is the proposed solution which is able to split the eigenvoice into constituents in the features space in order to compute the likelihood which is used in both the adaptation algorithm and recognition algorithm.

Following the investigation of kernel eigenface in the development of face recognition, KEV has been proposed as a non-linear extension of EV. The polynomial kernel and the Gaussian kernel have been studied. KEV using the Gaussian kernel showed promising results in a digit recognition task. By an observation that both base-EV and base-KEV do not guarantee it is better than SI model in terms of likelihood, two enhancements (addition approach and interpolation approach) have been proposed on KEV. Both of them incorporate the SI model to improve the robustness of the adaptation. In 2-second and 4-second adaptation set, KEV is not only better than EV model and SI model, but also outperforms the conventional adaptation approaches including MAP and MLLR adaptation. However, as more adaptation data are available (for example, 10-second adaptation set), MLLR becomes better.

In order to show the EV and KEV are extracting and utilizing some underlying inter-speaker variations, eigenvalues are analyzed. According to the analysis, it is found that the second eigenvectors in EV discriminate ”man” from ”woman” while the first two eigenvectors in KEV is highly correlated with the gender and age. It can be used to discriminate ”man”, ”woman” and ”children”.

7.3 Future work

There are three main extensions of the current work.

1. **Extension to the Gaussian mixtures or context-dependent mod-**

eling

In this thesis, all the experiments are based on single Gaussian HMM. It would be good to extend on Gaussian mixtures or context-dependent modeling. However, the method concatenation of means is kept, the dimension of supervector becomes huge. There is also a problem of sequence in Gaussian mixtures. A direct extension is to use MLLR-based eigenspace speaker adaptation by applying the eigen-decomposition on the MLLR space instead of the supervector of the means of HMMs. There are similar investigations in [8, 48, 31].

2. **KEV based on phone-classes**

   Currently, all units (digits or phonemes) are concatenated into a single supervector, which implies a huge constraint to the estimated weights. However, each unit could have different behavior, but if we consider each phoneme independently, it could result in insufficient data or require large amounts of adaptation data. This would violate one of the most important motivations of EV or KEV. So, KEV based on phoneme-classes is a desirable choice. For example, people from country A and country B may pronounce vowels in different ways. Aside from that, though their pronunciation of consonant are similar. We can then group the vowels and consonants into two clusters. Weights for each cluster are estimated independently. It then can release the constraints on the weights. In order to have an automatic architecture to form clusters, regression class tree can be considered, which uses a Euclidean distance measure for a centroid splitting algorithm.

3. **Speed-up issues**

   Although KEV gives an encouraging performance gain, it is costly in the computation. When performing recognition, it is at least N times slower than the conventional methods.

   Various speed-up methods are possible, which can be sub-divided into three areas. The first area is to reduce the number of kernels to be computed and it can be realized by sparse KPCA. or finding an approximated pre-image in the input space. The second area is for saving computation in the adaptation algorithm. A common approach is that, instead of computing all decoding paths in the adaptation, only the Viterbi path is used. This is due to the fact that the Viterbi path accounts for the major component in the auxiliary function. The time complexity of the kernel computation can be reduced from $O(NRT)$ to $O(NT)$. The third area is the recognition concern. One idea is to find an approximated pre-image in the input space. Another idea is to use a two-pass decoding. The SI model is used as the first-pass decoding, giving the N-best lattice. The KEV-adapted model is used as the second-pass decoding on the N-best lattice generated in the first-pass. In two-pass decoding, the first-pass decoding using a less expensive model, prunes most of the unlikely candidates. The search space can be reduced significantly by the n-best lattice. The second-pass decoding using an expensive model could be more efficient.

4. **Design and selection of kernel functions**

   In this thesis, Gaussian kernel and polynomial kernel are only compared experimentally. Deep analysis on the suitability and selection of kernel are absent, which is a very interesting area of study.

REFERENCES

[1] V. Abrash, A. Sankar, H. Franco, and M. Cohen. Acoustic adaptation using nonlinear transformation of HMM parameters. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 729–732, 1996.

[2] B. Schölkopf and A. Smola. *Learning with Kernels.* MIT Press, 2002.

[3] B. Schölkopf and A. Smola and K. R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319, 1998.

[4] B. Schölkopf and C. Burges and A. Smola. *Advances in Kernel Methods: Support Vector Learning.* MIT Press, 1999.

[5] F. R. Bach and M. I. Jordan. Kernel Independent Component Analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces, v.s. Fisherfaces: Recognition using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Learning*, 19(7):711–720, 1997.

[7] H. Botterweck. Anisotropic MAP defined by eigenvoices for large vocabulary continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[8] K. T. Chen, W. W. Liau, H. M. Wang, and L. S. Lee. Fast speaker adaptation using eigenspace-based maximum likelihood linear regression. In *Proceedings of the International Conference on Spoken Language Processing*, volume 4, pages 354–357, 2000.

[9] K. T. Chen and H. M. Wang. Eigenspace-based maximum a posteriori linear regression for rapid speaker adaptation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[10] L. Couvreur, S. Dupont, C. Ris, J. M. Boite, and C. Couvreur. Fast adaptation for robust speech recognition in Reverberant environments. In *Proceedings of International Workshop on Adaptation Methods for Speech Recognition*, pages 85–88, 2001.

[11] D. A., Harville. *Matrix Algebra From a Statistican's Perspective.* Springer, New York, Berlin, 1997.

[12] S. Douglas. Hypothesis-driven adaptation(HYDRA): a flexible eigenvoice architecture. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[13] E. Eide and H. Gish. A Parametric Approach to Vocal Tract Length Normalization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 346–348, 1996.

[14] S. Mika et al. Fisher Discriminant Analysis with Kernels. In *Proceedings of IEEE Neural Netwroks for Signal Processing Society Workshop*, pages 41–48, 1999.

[15] R. Falthauser and G. Ruske. Robust Speaker Clustering in Eigenspace. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2002.

[16] S. Fine, G. Saon, and R.A. Gopinath. Digit Recognition in noisy environments via a sequential GMM/SVM system. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002.

[17] M. J. F. Gales. Cluster Adaptive Training of Hidden Markov Models. *IEEE Transactions on Speech and Audio Processing*, 8(4):417–428, 2000.

[18] M. J. F. Gales. Multiple-cluster adaptive training schemes. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[19] N. Gandhi and S. Lakshmanan. An Eigenface Approach for Estimating Driver Pose. In *Proceedings of the 3rd Annual IEEE Conference on Intelligent Transportation Systems*, 2000.

[20] M. G. Genton. Classes of Kernels for Machine Learning: A Statistics Perspective. In *Journal of Machine Learning Research*, volume 2, pages 299–312, 2001.

[21] Y. Gong. Speech recognition in noisy environments: A survey. *Speech Communications*, 16:261–291, 1995.

[22] C. Huang, T. Chen, S. Li, E. Chang, and J. L. Zhou. Analysis of speaker variability. In *Proceedings of the European Conference on Speech Communication and Technology*, 2001.

[23] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, second edition, 2002.

[24] K. R. Müller and S. Mika and G. Rätsch and K. Tsuda and B. Schölkopf. An Introduction to Kernel-Based Learning Algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–202, March 2001.

[25] D. K. Kim and N. S. Kim. Rapid speaker adaptation using probabilistic principle component analysis. In *Signal Processing Letters, IEEE*, pages 180–183, June 2001.

[26] K. I. Kim, K. C. Jung, and H. J. Kim. Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters*, 9(2), 2002.

[27] R. Kuhn, J. C. Junqua, P. Nguyen, and N. Niedzielski. Rapid Speaker Adaptation in Eigenvoice Space. In *IEEE Transactions on Speech and Audio Processing*, volume 8, pages 695–707, November 2000.

[28] R. Kuhn, P. Nguyen, J. C. Junqua, R. Boman, N. Niedzielski, S. Fincke, K. Field, and M. Contolini. Fast speaker adaptation using a priori knowledge. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 749–752, 1999.

[29] R. Kuhn, P. Nguyen, J. C. Junqua, and L. Goldwasser. Eigenfaces and Eigenvoices: Dimensionality reduction for specialized pattern recognition. In *Multimedia Signal Processing, 1998 IEEE Second Workshop*, pages 71–76, December 1998.

[30] R. Kuhn, P. Nguyen, J. C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini. Eigenvoices for Speaker Adaptation. In *Proceedings of the International Conference on Spoken Language Processing*, volume 5, pages 1771–1774, 1998.

[31] R. Kuhn, F. Perronnin, P. Nguyen, J. C. Junqua, and L. Rigazio. Very fast adaptation with a compact context-dependent eigenvoice model. In

Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pages 373–376, 2001.

[32] R. Kuhn, F. Perronnon, and J.C. Junqua. Time is money: Why very rapid adaptation matters. *The International Speech Communication Association (ISCA) workshop*, 2001.

[33] C. H. Lee and J. L. Gauvain. Speaker adaptation based on MAP estimation of HMM parameters. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 558–561, 1993.

[34] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. In *Journal of Computer Speech and Language*, volume 9, pages 171–186, April 1995.

[35] R. G. Leonard. A Database for Speaker-Independent Digit Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1984.

[36] M. Kuss. *Nonlinear Multivariate Analysis.* PhD thesis, Technische Universität Berlin, 2002.

[37] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. In *Proceedings International Conference on Computer Vision*, pages 1131–1136, September 1999.

[38] P. Nguyen, L. Rigazio, C. Wellekens, and J. C. Junqua. Construction of model-space constraints. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 69–72, December 2001.

[39] P. Pavlidis, J. Weston, J Cai, and W. S. Noble. Learning Gene Functional Classifications from Multiple Data Types. *Journal of Computational Biology*, 9(2):401–411, November 2002.

[40] E. Pusateri and T. J. Hazen. Rapid Speaker Adaptation Using Speaker Clustering. In *Proceedings of the International Conference on Spoken Language Processing*, pages 61–64, September 2002.

[41] S. Mika and G. Ratsch and J. Weston and B. Schölkopf and K. R. Müller. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[42] R. E. Schapire. A Brief Introduction to Boosting. In *Proceedings of the International Joint Omega Conference on Artificial Intelligence*, pages 1401–1406, 1999.

[43] N. N. Schraudolph. A Fast, Compact Approximation of the Exponential Function. Technical Report IDSIA-07-98, Dalle Molle Institute for Artificial Intelligence, October 1998.

[44] O. Thyes, R. Kuhn, P. Nguyen, and J. C. Junqua. Speaker identification and verification using eigenvoices. In *Proceedings of the International Conference on Spoken Language Processing*, 2000.

[45] M. E. Tipping and C. M. Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B*, 61(3):611–622, 1999.

[46] M. A. Turk and A. P. Pentland. Face Recognition using Eigenfaces. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–590, Hawai, June 1992.

[47] V. Wan and S. Renals. Evaluation of kernel methods for speaker verification and identification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002.

[48] J. C. Wang, S. M. Lee, F. Seide, and L. S. Lee. Rapid speaker adaptation using a priori knowledge by eigenspace analysis of MLLR parameters. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[49] R. Westwood. *Speaker Adaptation Using Eigenvoices.* PhD thesis, Cambridge University, 1999.

[50] P. C. Woodland. Speaker adaptation: Techniques and Challenges. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 85–90, 2000.

[51] M. H. Yang. Face Recognition Using Kernel Methods. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[52] M. H. Yang. Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 215–220, 2002.

[53] M. H. Yang, N. Ahuja, and D. Kriegman. Face Recognition Using Kernel Eigenfaces. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 37–40, 2000.

[54] A. Yilmaz and M. Gokmen. Eigenhill vs. Eigenface and Eigenedge. In *Proceedings of International Conference Pattern Recognition*, pages 827–830, 2000.

[55] B. Yu, P. Chen, and L. F. Jin. Recognizing faces with expressions: within-class space and between-class space. In *International Conference on Pattern Recognition*, volume 1, August 2002.

APPENDIX A

PROOFS FOR KPCA

A.1 Proof of centering of covariance matrix

$$\begin{aligned}
\tilde{K}_{ij} &= \tilde{\varphi}(\mathbf{x}_i)'\tilde{\varphi}(\mathbf{x}_j) \\
&= (\varphi(\mathbf{x}_i) - \bar{\boldsymbol{\varphi}})'(\varphi(\mathbf{x}_j) - \bar{\boldsymbol{\varphi}}) \\
&= \varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_j) - \frac{1}{N}\sum_{q=1}^{N}\varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_q) - \\
&\quad \frac{1}{N}\sum_{p=1}^{N}\varphi(\mathbf{x}_p)'\varphi(\mathbf{x}_j) + \frac{1}{N^2}\sum_{p=1}^{N}\sum_{q=1}^{N}\varphi(\mathbf{x}_p)'\varphi(\mathbf{x}_q) \\
&= K_{ij} - \frac{1}{N}\sum_{q=1}^{N}K_{iq} - \frac{1}{N}\sum_{p=1}^{N}K_{pj} + \frac{1}{N^2}\sum_{p=1}^{N}\sum_{q=1}^{N}K_{pq}
\end{aligned} \tag{A.1}$$

Then,

$$\tilde{\mathbf{K}} = \mathbf{HKH} \tag{A.2}$$

where
$\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{11}'$ and
$\mathbf{1} = [11...1]'$

A.2 Proof of the normalizing factor used in KPCA

For each eigenvector $\mathbf{v}_m$, it is normalized by c_m as follows:

$$\mathbf{v}_m = \sum_{n=1}^{N}\frac{\alpha_{mn}}{c_m}\tilde{\varphi}(\mathbf{x}_i) \tag{A.3}$$

By definition of orthonormality, for any eigenvector $\mathbf{v}_m$ in the feature space,

$$\mathbf{v}_m'\mathbf{v}_m = 1 \tag{A.4}$$

By substituting A.3 into A.4, it becomes:

$$\begin{aligned} c^2 &= \sum_{n=1}^{N}\sum_{n'=1}^{N}\alpha_{mn}\alpha_{mn'}\tilde{\varphi}(\mathbf{x}_n)'\tilde{\varphi}(\mathbf{x}_{n'}) \\ &= \sum_{n=1}^{N}\sum_{n'=1}^{N}\alpha_{mn}\alpha_{mn'}K_{nn'} \\ &= \boldsymbol{\alpha}_m'\mathbf{K}\boldsymbol{\alpha}_m \end{aligned}$$

Since $\mathbf{K}\boldsymbol{\alpha}_m = \lambda_m\boldsymbol{\alpha}_m$,

$$\begin{aligned} c^2 &= \boldsymbol{\alpha}_m'\lambda_m\boldsymbol{\alpha}_m \\ &= \lambda_m(\boldsymbol{\alpha}_m'\boldsymbol{\alpha}_m) \\ &= \lambda_m \\ c &= \sqrt{\lambda_m} \end{aligned}$$

Therefore,

$$\mathbf{v}_m = \sum_{n=1}^{N}\frac{\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}(\mathbf{x}_i) \tag{A.5}$$

APPENDIX B

DERIVATION FOR ORIGINAL KERNEL EIGENVOICE

B.1 Derivation of $\varsigma_r'\varphi_r(\mathbf{o}_t)$

$$\begin{aligned}
&\varsigma_r'\varphi_r(\mathbf{o}_t)\\
= &\Big[(\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{x}_{nr})) + \bar{\boldsymbol{\varphi}}_r\Big]'\varphi_r(\mathbf{o}_t)\\
= &\Big[\Big(\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}(\varphi_r(\mathbf{x}_{nr}) - \bar{\boldsymbol{\varphi}}_r)\Big) + \bar{\boldsymbol{\varphi}}_r\Big]'\varphi_r(\mathbf{o}_t)\\
= &\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}k_r(\mathbf{x}_{nr},\mathbf{o}_t) + \Big(1 - \sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\Big)\bar{\boldsymbol{\varphi}}_r'\varphi_r(\mathbf{o}_t)\\
= &\bar{\boldsymbol{\varphi}}_r'\varphi_r(\mathbf{o}_t) + \sum_{m=1}^{M}w_m\sum_{n=1}^{N}\frac{\alpha_{mn}}{\sqrt{\lambda_m}}\Big[k_r(\mathbf{x}_{nr},\mathbf{o}_t) - \bar{\boldsymbol{\varphi}}_r'\varphi_r(\mathbf{o}_t)\Big] \qquad \text{(B.1)}\\
= &A(r,t) + \sum_{m=1}^{M}w_m B(m,r,t) \qquad \text{(B.2)}
\end{aligned}$$

where

$$\begin{aligned}
A(r,t) &= \bar{\boldsymbol{\varphi}}_r'\varphi_r(\mathbf{o}_t) = \frac{1}{N}\sum_{n=1}^{N}k_r(\mathbf{x}_{nr},\mathbf{o}_t)\\
B(m,r,t) &= \sum_{n=1}^{N}\frac{\alpha_{mn}}{\sqrt{\lambda_m}}\Big(k_r(\mathbf{x}_{nr},\mathbf{o}_t) - A(r,t)\Big)
\end{aligned}$$

Differentiate $\varsigma_r'\varphi_r(\mathbf{o}_t)$ with respect to w_j,

$$\frac{\partial\varsigma_r'\varphi_r(\mathbf{o}_t)}{\partial w_j} = B(j,r,t) \qquad \text{(B.3)}$$

B.2 Derivation of $\varsigma_r'\varsigma_r$

$$
\begin{aligned}
&\varsigma_r'\varsigma_r \\
&= \Big[\Big(\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{x}_{nr})\Big)+\bar{\boldsymbol{\varphi}}_r\Big]'\Big[\Big(\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{x}_{nr})\Big)+\bar{\boldsymbol{\varphi}}_r\Big] \\
&= \Big(\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{x}_{nr})\Big)'\Big(\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{x}_{nr})\Big)+ \\
&\quad 2\Big(\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{x}_{nr})\Big)'\bar{\boldsymbol{\varphi}}_r+\bar{\boldsymbol{\varphi}}_r'\bar{\boldsymbol{\varphi}}_r \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{m'=1}^{M}\sum_{n'=1}^{N}(\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{x}_{nr}))'(\frac{w_{m'}\alpha_{m'n'}}{\sqrt{\lambda_{m'}}}\tilde{\varphi}_r(\mathbf{x}_{nr}))+ \\
&\quad 2\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{x}_{nr})'\bar{\boldsymbol{\varphi}}_r+\bar{\boldsymbol{\varphi}}_r'\bar{\boldsymbol{\varphi}}_r \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{m'=1}^{M}\sum_{n'=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\frac{w_{m'}\alpha_{m'n'}}{\sqrt{\lambda_{m'}}}\tilde{k}_r(\mathbf{x}_{nr},\mathbf{x}_{n'r})+ \\
&\quad 2\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{x}_{nr})'\bar{\boldsymbol{\varphi}}_r+\bar{\boldsymbol{\varphi}}_r'\bar{\boldsymbol{\varphi}}_r
\end{aligned}
$$

where

$$
\begin{aligned}
\tilde{k}_r(\mathbf{x}_{nr},\mathbf{x}_{n'r}) &= \tilde{\varphi}_r(\mathbf{x}_{nr})'\tilde{\varphi}_r(\mathbf{x}_{n'r}) \\
&= \Big[\varphi_r(\mathbf{x}_{nr}-\bar{\boldsymbol{\varphi}}_r)\Big]'\Big[\varphi_r(\mathbf{x}_{n'r}-\bar{\boldsymbol{\varphi}}_r)\Big] \\
&= k_r(\mathbf{x}_{nr},\mathbf{x}_{n'r})-\frac{1}{N}\sum_{i=1}^{N}\Big[k_r(\mathbf{x}_{nr},\mathbf{x}_{ir})+k_r(\mathbf{x}_{n'r},\mathbf{x}_{ir})\Big]+ \\
&\quad \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}k_r(\mathbf{x}_{ir},\mathbf{x}_{jr})
\end{aligned}
$$

$$
\begin{aligned}
\tilde{\varphi}_r(\mathbf{x}_{nr})'\bar{\boldsymbol{\varphi}}_r &= \tilde{\varphi}_r(\mathbf{x}_{nr})'\Big(\frac{1}{N}\sum_{i=1}^{N}\varphi_r(\mathbf{x}_{ir})\Big) \\
&= \frac{1}{N}\sum_{i=1}^{N}\Big(\varphi_r(\mathbf{x}_{nr})-\bar{\boldsymbol{\varphi}}_r\Big)'\varphi_r(\mathbf{x}_{ir}) \\
&= \frac{1}{N}\sum_{i=1}^{N}\varphi_r(\mathbf{x}_{nr})'\varphi_r(\mathbf{x}_{ir})-\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\varphi_r(\mathbf{x}_{ir})'\varphi_r(\mathbf{x}_{jr})
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{N}\sum_{i=1}^{N} k_r(\mathbf{x}_{nr}, \mathbf{x}_{ir}) - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k_r(\mathbf{x}_{ir}, \mathbf{x}_{jr}) \\
\bar{\boldsymbol{\varphi}}_r' \bar{\boldsymbol{\varphi}}_r &= \Big[\frac{1}{N}\sum_{i=1}^{N} \varphi_r(\mathbf{x}_{ir})\Big]' \Big[\frac{1}{N}\sum_{j=1}^{N} \varphi_r(\mathbf{x}_{jr})\Big] \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k(\mathbf{x}_{ir}, \mathbf{x}_{jr})
\end{aligned}
$$

So, The final equation is:

$$
\varsigma_r' \varsigma_r = \sum_{m=1}^{M}\sum_{m'=1}^{M} w_m w_{m'} D(m, m', r) + \sum_{m=1}^{M} w_m E(m, r) + F(r) \tag{B.4}
$$

where

$$
\begin{aligned}
D(m, m', r) &= \sum_{n=1}^{N}\sum_{n'=1}^{N} \frac{\alpha_{mn}\alpha_{m'n'}}{\sqrt{\lambda_m \lambda_{m'}}} \Big[k_r(\mathbf{x}_{nr}, \mathbf{x}_{n'r}) - \\
&\quad \frac{1}{N}\sum_{i=1}^{N}\Big[k_r(\mathbf{x}_{nr}, \mathbf{x}_{ir}) + k_r(\mathbf{x}_{n'r}, \mathbf{x}_{ir})\Big] + F(r)\Big] \\
E(m, r) &= 2\sum_{n=1}^{N} \frac{\alpha_{mn}}{\sqrt{\lambda_m}} \Big[\frac{1}{N}\sum_{i=1}^{N} k_r(\mathbf{x}_{nr}, \mathbf{x}_{ir}) - F(r))\Big] \\
F(r) &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k_r(\mathbf{x}_{ir}, \mathbf{x}_{jr})
\end{aligned}
$$

Differentiate $\varsigma_r' \varsigma_r$ with respect to w_j,

$$
\frac{\partial \varsigma_r' \varsigma_r}{\partial w_j} = E(j, r) + \sum_{m=1}^{M} 2 w_m D(m, j, r) \tag{B.5}
$$

APPENDIX C

DERIVATION FOR ROBUST KEV 1 - ADDITION

C.1 Derivation of $\varsigma_r'\varphi_r(\mathbf{o}_t)$

For robust KEV, the only difference is the definition of the new speaker in the feature space which is as follows:

$$\begin{aligned}
\tilde{\varsigma} &= \tilde{\varphi}(\mathbf{x}^{(SI)}) + \sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}(\mathbf{x}_n) \\
&= \Big(\varphi(\mathbf{x}^{(SI)}) - \bar{\boldsymbol{\varphi}}\Big) + \sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\Big(\varphi(\mathbf{x}_n) - \bar{\boldsymbol{\varphi}}\Big) \qquad (\text{C.1})
\end{aligned}$$

$$\begin{aligned}
&\varsigma_r'\varphi_r(\mathbf{o}_t) \\
&= \Big[\tilde{\varsigma}_r + \bar{\boldsymbol{\varphi}}_r\Big]'\varphi_r(\mathbf{o}_t) \\
&= \Big[\Big(\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}(\varphi_r(\mathbf{x}_{nr}) - \bar{\boldsymbol{\varphi}}_r)\Big) + \varphi_r(\mathbf{x}^{(SI)})\Big]'\varphi_r(\mathbf{o}_t) \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}k_r(\mathbf{x}_{nr},\mathbf{o}_t) - \sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\bar{\boldsymbol{\varphi}}_r'\varphi_r(\mathbf{o}_t) + \varphi_r(\mathbf{x}^{(SI)})'\varphi_r(\mathbf{o}_t) \\
&= \varphi_r(\mathbf{x}^{(SI)})'\varphi_r(\mathbf{o}_t) + \sum_{m=1}^{M}w_m\sum_{n=1}^{N}\frac{\alpha_{mn}}{\sqrt{\lambda_m}}\Big[k_r(\mathbf{x}_{nr},\mathbf{o}_t) - \bar{\boldsymbol{\varphi}}_r'\varphi_r(\mathbf{o}_t)\Big] \\
&= k_r(\mathbf{x}_r^{SI},\mathbf{o}_t) + \sum_{m=1}^{M}w_m B(m,r,t) \qquad (\text{C.2})
\end{aligned}$$

where

$$\begin{aligned}
A(r,t) &= \bar{\boldsymbol{\varphi}}_r'\varphi_r(\mathbf{o}_t) = \frac{1}{N}\sum_{i=1}^{N}k_r(\mathbf{x}_{ir},\mathbf{o}_t) \\
B(m,r,t) &= \sum_{n=1}^{N}\frac{\alpha_{mn}}{\sqrt{\lambda_m}}\Big(k_r(\mathbf{x}_{nr},\mathbf{o}_t) - A(r,t)\Big)
\end{aligned}$$

Differentiate $\varsigma_r'\varphi_r(\mathbf{o}_t)$ with respect to w_j,

$$\frac{\partial\varsigma_r'\varphi_r(\mathbf{o}_t)}{\partial w_j} = B(j,r,t) \tag{C.3}$$

APPENDIX D

DERIVATION FOR ROBUST KEV 2 - INTERPOLATION

D.1 Derivation of $\varsigma_r'\varphi_r(\mathbf{o}_t)$

For robust KEV, the only difference is the definition of the new speaker in the feature space which is as follows:

$$\begin{aligned}
\tilde{\varphi}(s) &= w_0\tilde{\varphi}(\mathbf{x}^{(SI)}) + (1-w_0)\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\tilde{\varphi}(\mathbf{x}_n) \\
&= w_0\Big(\varphi(\mathbf{x}^{(SI)}) - \bar{\boldsymbol{\varphi}}\Big) + \\
&\quad (1-w_0)\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\Big(\varphi(\mathbf{x}_n) - \bar{\boldsymbol{\varphi}}\Big) \qquad \text{(D.1)}
\end{aligned}$$

$$\begin{aligned}
\varphi(s) &= \tilde{\varphi}(s) + \bar{\boldsymbol{\varphi}}(s) \\
&= w_0\varphi(\mathbf{x}^{(SI)}) + (1-w_0)\bar{\boldsymbol{\varphi}} + \\
&\quad (1-w_0)\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\Big(\varphi(\mathbf{x}_n) - \bar{\boldsymbol{\varphi}}\Big) \qquad \text{(D.2)}
\end{aligned}$$

$$\begin{aligned}
&\varsigma_r'\varphi_r(\mathbf{o}_t) \\
&= \Big[w_0\varphi_r(\mathbf{x}_r^{(SI)}) + (1-w_0)\bar{\boldsymbol{\varphi}}_r + \\
&\quad (1-w_0)\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{w_m\alpha_{mn}}{\sqrt{\lambda_m}}\Big(\varphi_r(\mathbf{x}_{nr}) - \bar{\boldsymbol{\varphi}}_r\Big)\Big]'\varphi_r(\mathbf{o}_t) \\
&= w_0 k_r(\mathbf{x}_r^{(SI)}, \mathbf{o}_t) + (1-w_0)\Big[A(r,t) + \sum_{m=1}^{M} w_m B(m,r,t)\Big] \qquad \text{(D.3)}
\end{aligned}$$

where

$$A(r,t) = \bar{\boldsymbol{\varphi}}_r'\varphi_r(\mathbf{o}_t) = \frac{1}{N}\sum_{n=1}^{N} k_r(\mathbf{x}_{nr}, \mathbf{o}_t)$$

$$B(m,r,t) \;=\; \sum_{n=1}^{N} \frac{\alpha_{mn}}{\sqrt{\lambda_m}} \Big(k_r(\mathbf{x}_{nr}, \mathbf{o}_t) - A(r,t) \Big)$$

- Differentiate $\varsigma_r' \varphi_r(\mathbf{o}_t)$ with respect to w_0

$$\frac{\partial \varsigma_r' \varphi_r(\mathbf{o}_t)}{\partial w_0} = k_r(\mathbf{x}_r^{(SI)}, \mathbf{o}_t) - A(r,t) - \sum_{m=1}^{M} w_m B(m,r,t) \tag{D.4}$$

- Differentiate $\varsigma_r' \varphi_r(\mathbf{o}_t)$ with respect to w_j, for $j = 1 \cdots M$,

$$\frac{\partial \varsigma_r' \varphi_r(\mathbf{o}_t)}{\partial w_j} = (1 - w_0) B(j,r,t) \tag{D.5}$$

APPENDIX E

PRACTICAL SPEED-UP METHODS IN KEV

Some practical methods are considered in this thesis for speed-up. The comparison of the CPU time and the accuracy of the lookup table, approximation method I and II to the exponential function are summarized in Table E.1.

1. **Pruning**

   In decoding, pruning is a common approach for speed-up. Reducing the search space by pruning can directly decrease the number of distance measure computation. This can improve the speed.

2. **Lookup table**

   In the distance calculation, exponential function is the most costly part for the Gaussian kernel. It is found that the input of the exponential function is usually within a narrow range. Therefore, pre-computing exponential values in that range could speed-up.

3. **Schraudolph fast approximation method to exponential function**

   According to the [43], a fast and compact method is proposed for approximating the exponential function.

4. **Series-based approximation method to exponential function**

   In the calculation of the $k_r(s_r, o_t)$, computing exponential is most costly component. Therefore, we first express exponential function as a series. Then, the input value is bound to a given range and the first four terms of the series is used as the approximation. It is defined in E.2. The comparison of the CPU time and the absolute error for using various number of terms are stated in Table E.1.

   Exponential function can be expressed as a series which is:

   $$exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ... + \frac{x^n}{n!} \tag{E.1}$$

let $y = \frac{x}{c}$ where $c > x$ so that $-1 < y < 1$

$$\begin{aligned} exp(x) &= \Big[exp(y)\Big]^c \\ &\approx \Big[1 + y + \frac{y^2}{2} + \frac{y^3}{6}\Big]^c \\ &= \Big[1 + x\Big(1 + x(\frac{1}{2} + \frac{x}{6})\Big)\Big]^c \end{aligned} \tag{E.2}$$

if it is approximated by the first four terms and $c = 2^d$, power function means d multiplications ($d = 6$ in the experiment).

Therefore, the exponential function is replaced by 8 multiplications and 3 additions.

	CPU time	**Relative error**
Build-in exponential function	1.3	NIL
Lookup table	0.64	0.050%
Schraudolph fast approx. method	0.55	1.517%
Series-based approx. method	0.77	0.280%

Table E.1: Comparison of the CPU time and the accuracy of the 3 proposed approximation methods to exponential function with the build-in one

Finally, in order to have a balance in minimizing the relative error and CPU time, the lookup table approach is used in calculating exponential function.

APPENDIX F

SIGNIFICANCE TESTS

In the significant tests, MAP, MLLR, base-EV, robust-EV, base-KEV and robust-KEV are compared. Their abbreviations are summarized as follows:

b-EV	:	base-EV
r-EV	:	robust-EV
b-KEV	:	base-KEV
r-KEV	:	robust-KEV (the interpolation one)
MLLR.d	:	MLLR with diagonal transformation matrix
MLLR.f	:	MLLR with full transformation matrix
SI-m	:	SI modeling

MP	:	Matched Pair Sentence Segment (Word Error) Test
SI	:	Signed Paired Comparison (Speaker Word Accuracy Rate (%)) Test
WI	:	Wilcoxon Signed Rank (Speaker Word Accuracy Rate (%)) Test
MN	:	McNemar (Sentence Error) Test

	MAP	**MLLR.f**	**MLLR.d**	**SI-m**	**r-EV**	**b-KEV**	**r-KEV**
b-EV	MP: MAP SI: MAP WI: MAP MN: MAP	MP: MLLR.f SI: MLLR.f WI: MLLR.f MN: MLLR.f	MP: MLLR.d SI: MLLR.d WI: MLLR.d MN: MLLR.d	MP: SI-m SI: SI-m WI: SI-m MN: SI-m	MP: r-EV SI: r-EV WI: r-EV MN: r-EV	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
MAP		MP: MLLR.f SI: MLLR.f WI: MLLR.f MN: MLLR.f	MP: MLLR.d SI: MLLR.d WI: MLLR.d MN: MLLR.d	MP: SI-m SI: SI-m WI: SI-m MN: SI-m	MP: r-EV SI: r-EV WI: r-EV MN: r-EV	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
MLLR.f			MP: same SI: same WI: same MN: same	MP: SI-m SI: SI-m WI: SI-m MN: SI-m	MP: r-EV SI: r-EV WI: r-EV MN: r-EV	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
MLLR.d				MP: SI-m SI: SI-m WI: SI-m MN: SI-m	MP: r-EV SI: r-EV WI: r-EV MN: r-EV	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
SI-m					MP: same SI: same WI: same MN: same	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
r-EV						MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
b-KEV							MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV

Table F.1: Significance Tests on *2-second* adaptation data

Table F.2: Significance Tests on *4-second* adaptation data

	MLLR.f	**MAP**	**MLLR.d**	**SI-m**	**r-EV**	**b-KEV**	**r-KEV**
b-EV	MP: MLLR.f SI: MLLR.f WI: MLLR.f MN: MLLR.f	MP: MAP SI: MAP WI: MAP MN: MAP	MP: MLLR.d SI: MLLR.d WI: MLLR.d MN: MLLR.d	MP: SI-m SI: SI-m WI: SI-m MN: SI-m	MP: r-EV SI: r-EV WI: r-EV MN: r-EV	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
MLLR.f		MP: same SI: same WI: same MN: MAP	MP: MLLR.d SI: same WI: same MN: MLLR.d	MP: SI-m SI: SI-m WI: SI-m MN: SI-m	MP: r-EV SI: r-EV WI: r-EV MN: r-EV	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
MAP			MP: MLLR.d SI: MLLR.d WI: MLLR.d MN: same	MP: SI-m SI: SI-m WI: SI-m MN: SI-m	MP: r-EV SI: r-EV WI: r-EV MN: r-EV	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
MLLR.d				MP: SI-m SI: SI-m WI: SI-m MN: SI-m	MP: r-EV SI: r-EV WI: r-EV MN: r-EV	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
SI-m					MP: same SI: same WI: same MN: same	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
r-EV						MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV
b-KEV							MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV

Table F.3: Significance Tests on *10-second* adaptation data

	MLLR.d	**SI-m**	**r-EV**	**b-KEV**	**MAP**	**r-KEV**	**MLLR.f**
b-EV	MP: MLLR.d SI: MLLR.d WI: MLLR.d MN: MLLR.d	MP: SI-m SI: SI-m WI: SI-m MN: SI-m	MP: r-EV SI: r-EV WI: r-EV MN: r-EV	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: MAP SI: MAP WI: MAP MN: MAP	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV	MP: MLLR.f SI: MLLR.f WI: MLLR.f MN: MLLR.f
MLLR.d		MP: same SI: same WI: same MN: same	MP: same SI: same WI: same MN: same	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: MAP SI: MAP WI: MAP MN: MAP	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV	MP: MLLR.f SI: MLLR.f WI: MLLR.f MN: MLLR.f
SI-m			MP: same SI: same WI: r-EV MN: same	MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: MAP SI: MAP WI: MAP MN: MAP	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV	MP: MLLR.f SI: MLLR.f WI: MLLR.f MN: MLLR.f
r-EV				MP: b-KEV SI: b-KEV WI: b-KEV MN: b-KEV	MP: MAP SI: MAP WI: MAP MN: MAP	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV	MP: MLLR.f SI: MLLR.f WI: MLLR.f MN: MLLR.f
b-KEV					MP: MAP SI: MAP WI: MAP MN: MAP	MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV	MP: MLLR.f SI: MLLR.f WI: MLLR.f MN: MLLR.f
MAP						MP: r-KEV SI: r-KEV WI: r-KEV MN: r-KEV	MP: MLLR.f SI: MLLR.f WI: MLLR.f MN: MLLR.f
r-KEV							MP: MLLR.f SI: MLLR.f WI: MLLR.f MN: MLLR.f