Embedded Kernel Eigenvoice Speaker Adaptation and its Implication to Reference Speaker Weighting

Brian Mak, Roger Hsiao, Simon Ho, and James T. Kwok

Abstract— **Recently, we proposed an improvement to the conventional eigenvoice (EV) speaker adaptation using kernel methods. In our novel *kernel eigenvoice (KEV) speaker adaptation* [1], speaker supervectors are mapped to a kernel-induced high dimensional feature space, where eigenvoices are computed using kernel principal component analysis. A new speaker model is then constructed as a linear combination of the leading eigenvoices in the kernel-induced feature space. KEV adaptation was shown to outperform EV, MAP, and MLLR adaptation in a TIDIGITS task with less than 10s of adaptation speech [2]. Nonetheless, due to many kernel evaluations, both adaptation and subsequent recognition in KEV adaptation are considerably slower than conventional EV adaptation. In this paper, we solve the efficiency problem and eliminate all kernel evaluations involving adaptation or testing observations by finding an approximate *pre-image* of the implicit adapted model found by KEV adaptation in the feature space; we call our new method *embedded kernel eigenvoice (eKEV) adaptation*. eKEV adaptation is faster than KEV adaptation, and subsequent recognition runs as fast as normal HMM decoding.**

eKEV adaptation makes use of multi-dimensional scaling technique so that the resulting adapted model lies in the span of a subset of carefully chosen training speakers. It is related to the *reference speaker weighting* (RSW) adaptation method that is based on speaker clustering. Our experimental results on Wall Street Journal show that eKEV adaptation continues to outperform EV, MAP, MLLR, and the original RSW method. However, by adopting the way we choose the subset of reference speakers for eKEV adaptation, we may also improve RSW adaptation so that it performs as well as our eKEV adaptation.

Keywords— **Eigenvoice speaker adaptation, kernel eigenvoice speaker adaptation, kernel PCA, composite kernels, pre-image problem, reference speaker weighting**

I. Introduction

A well-trained speaker-dependent (SD) model generally achieves better performance than a speaker-independent (SI) model on recognizing speech from the specific speaker. However, it is usually hard to acquire a large amount of data from a user to train a good SD model; even if one manages to do so, the speaker-specific data will not have a phonetic coverage as broad as the SI model. A more practical approach to attain the SD performance without sacrificing the phonetic coverage is to adapt the SI model with a relatively small amount of SD speech using speaker adaptation methods. Adaptation methods like the speaker-clustering-based methods [3], [4], the Bayesian-based *maximum a posteriori* (MAP) adaptation [5], and the transformation-based *maximum likelihood linear regression* (MLLR) adaptation [6] have been popular for many years. Nevertheless, when the amount of available adaptation speech is really small — for example, only a few seconds — the eigenvoice-based (or eigenspace-based) adaptation method recently has drawn a lot of attention. The (original) eigenvoice (EV) adaptation method [7] was motivated by the eigenface approach in face recognition [8]. The idea is to derive from a diverse set of speaker-specific parametric vectors a small set of basis vectors called *eigenvoices* that are believed to represent principal voice characteristics (e.g. gender, age, accent, etc.), and any training or new speaker is then a point in the eigenspace. In practice, a few to a few tens of eigenvoices are found adequate for fast speaker adaptation. Since the number of estimation parameters is greatly reduced, fast speaker adaptation using EV adaptation is possible with a few seconds of speech. The simple algorithm was later extended to work for large-vocabulary continuous speech recognition [9], [10], eigenspace-based MLLR [11], [12], and to approximate the model prior in MAP adaptation [13], [14], [15]. In addition, the eigenspace may be learned automatically by MLES [16], or during model training as in CAT [17].

Meanwhile, in the machine learning research community, recently there has been a lot of interest in the study of kernel methods [18], [19], [20]. The basic idea is to map data in the input space to a high dimensional feature space via some nonlinear map, and then apply a linear method there. The computational procedure depends only on the inner products in the feature space, which can be obtained efficiently with a suitable kernel function. Thus, the use of kernels provides elegant nonlinear generalizations of many existing linear algorithms. A well-known example in supervised learning is the support vector machines (SVMs). In unsupervised learning, the kernel idea has also led to methods such as kernel principal component analysis (PCA) [21], kernel-based clustering algorithms [22], and kernel independent component analysis (ICA) [23].

In [1], we proposed a kernel version of EV adaptation called *kernel eigenvoice (KEV) speaker adaptation* that exploits possible nonlinearity in the input speaker supervector space using kernel methods in order to improve its adaptation performance. Speaker supervectors are mapped to a kernel-induced high dimensional feature space[1] via some nonlinear map φ, and PCA is then applied there. During the actual computation, the exact nonlinear map does not

The authors are with the Department of Computer Science, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. E-mail: {mak,hsiao,csho,jamesk}@cs.ust.hk

[1]In kernel methods terminology, the original space where raw data reside is called the *input space* and the space to which raw data are mapped is called the *feature space*. In order not to confuse this with the acoustic feature space in speech, the latter will always be called the "acoustic feature space", while the feature space in kernel methods will be simply called the "feature space" but may be sometimes called the "*kernel-induced feature space*" when additional clarity is necessary.

need to be known, and the eigenvoices in KEV adaptation are obtained in the kernel-induced feature space using *kernel PCA*. In principle, since KEV adaptation is a nonlinear generalization of EV adaptation, the former should be more powerful than the latter, and KEV adaptation is expected to give better performance. In fact, KEV adaptation will be reduced to the traditional EV adaptation method if a linear kernel is employed. In a TIDIGITS adaptation task, it was shown that KEV adaptation outperformed the SI model by about 30% using only 2.1, 4.1, or 9.6 seconds of adaptation speech, and was better than MAP and MLLR adaptation [1].

However, there is a price to pay for using kernel PCA in KEV adaptation: adaptation and subsequent recognition can be substantially slower than EV adaptation due to many online kernel evaluations during the computation of observation likelihoods. The problem is due to the fact that the eigenvoices found by KEV adaptation reside in the kernel-induced feature space, and since a speaker acoustic model is represented as a linear combination of these kernel eigenvoices, after adaptation, a new speaker adapted (SA) model exists only *implicitly* in the feature space. As there is no *explicit* model for the new speaker in the input speaker supervector space, any computation involving it has to be done online on the *implicit* SA model in the feature space via expensive kernel evaluations. Finding an exact or a good approximate explicit model of an object in the input space from its image in the feature space is known as the *pre-image* problem in kernel methods. There are a few solutions: a fixed-point iterative method in [24], an analytical solution using distance constraints in [25], and by learning the inverse map in [26]. In this paper, we integrate the finding of an implicit SA model in the feature space using kernel PCA and the computation of its approximate pre-image to arrive at an *explicit* SA model in the input speaker supervector space. The novelty of our method is that there are no kernel evaluations during adaptation involving adaptation speech from the new speaker, and there are no kernel evaluations at all during recognition. Consequently, adaptation is faster and subsequent recognition is as fast as conventional EV adaptation. Our new method will be called *embedded kernel eigenvoice (eKEV) speaker adaptation*.

The pre-imaging procedure makes use of multidimensional scaling technique, and the adapted speaker model is confined to the span of a set of carefully chosen reference speakers in the input space. In this perspective, our eKEV adaptation method is similar to *reference speaker weighting* (RSW) adaptation [3], [4]. RSW adaptation is one kind of speaker-clustering-based adaptation methods in which the adapted speaker model is assumed to be a linear combination of a set of reference speakers. In [3], the set of combination weights are equal, whereas in [4], the weights are found by maximizing the likelihood of the adaptation data of the new speaker. eKEV adaptation is different from the RSW method in [4] in the way the reference speakers are defined, and eKEV adaptation further requires the solution to be constrained to the part of reference speakers' span that is related to the eigenspace found by KEV adaptation in the kernel-induced feature space. We will compare the two adaptation methods empirically to check if such prior information is useful.

This paper is organized as follows. We first review the conventional eigenvoice speaker adaptation method in Section II, and kernel eigenvoice speaker adaptation in Section III. The new method, embedded kernel eigenvoice speaker adaptation, is detailed in Section IV. In Section V, eKEV adaptation is evaluated and compared with other common adaptation methods using TIDIGITS (a small-vocabulary task) and WSJ0 (a large-vocabulary task) corpora. Conclusions are finally drawn in Section VI.

II. Eigenvoice Speaker Adaptation (EV)

In standard eigenvoice speaker adaptation [7], a set of speaker-dependent (SD) acoustic models are estimated from speech data collected from many training speakers with diverse speaking or voicing characteristics. All SD models are hidden Markov models (HMMs) of the same topology and the state probability density functions (pdf) are Gaussian mixture models. For simplicity, we will assume that each HMM state consists of a single Gaussian; the extension to mixture of Gaussians is straightforward. Then a speaker model is represented by what is called a *speaker supervector* that is composed by concatenating all the mean vectors of all his/her HMM state Gaussians. That is, for the ith speaker, if there are R Gaussians in his/her HMMs, each having a mean vector $\mathbf{x}_{ir}, r = 1, \ldots, R$, then his/her speaker supervector is denoted by $\mathbf{x}_i = [\mathbf{x}'_{i1}, \ldots, \mathbf{x}'_{iR}]'$. If the dimension of each mean vector is n_1, then each speaker supervector has a dimension of $n_2 = Rn_1$. Suppose that there are N training speaker models represented by their supervectors, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. In EV adaptation, linear principal component analysis (PCA) is performed on the N speaker supervectors and the resulting eigenvectors are called *eigenvoices*. Any speaker, either a training speaker or a new speaker, can now be represented as a linear combination of these eigenvoices. In order to reduce the number of estimation parameters for fast adaptation and to avoid unwanted variances, only the leading $M < N$ eigenvoices $\{\mathbf{v}_1, \ldots, \mathbf{v}_M\}$ having the largest eigenvalues are kept to represent a new speaker supervector $\mathbf{s}^{(ev)}$. That is, the centered supervector of the new speaker $\tilde{\mathbf{s}}^{(ev)}$ (where $\tilde{}$ is added to any quantity in this paper to denote its centered version) is

$$\tilde{\mathbf{s}}^{(ev)} = \sum_{m=1}^{M} w_m \mathbf{v}_m \ , \tag{1}$$

where $\tilde{\mathbf{s}}^{(ev)} = \mathbf{s}^{(ev)} - \bar{\mathbf{x}}$ and $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$ is the mean of all training speaker supervectors, and $\mathbf{w} = [w_1, \ldots, w_M]'$ is the eigenvoice weight vector. Usually, only a few eigenvoices (e.g., $M < 50$) are employed so that a small amount of adaptation speech (e.g., a few seconds) is sufficient for adaptation. Given the adaptation data $\boldsymbol{O} =$

$\{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T\}$, the eigenvoice weights are usually estimated by maximizing the likelihood of $\boldsymbol{O}$. Mathematically, one finds the optimal $\hat{\mathbf{w}}$ by *maximizing* the following $Q_b(\mathbf{w})$ function:

$$Q_b(\mathbf{w}) = \sum_{r=1}^{R}\sum_{t=1}^{T} \gamma_t(r) \log\left(b_r(\mathbf{o}_t, \mathbf{w})\right) \ , \tag{2}$$

where $\gamma_t(r)$ is the posterior probability of the observation sequence being at state r at time t, and b_r is the Gaussian pdf of the rth state of the speaker adapted model. By expanding the Gaussian pdf and ignoring all terms that are independent of $\mathbf{w}$, one may find the optimal $\hat{\mathbf{w}}$ that maximizes the following reduced $Q(\mathbf{w})$ function instead:

$$\begin{aligned} \hat{\mathbf{w}} &= \underset{\mathbf{w}}{\text{argmax}}\ Q(\mathbf{w}) \\ &= \underset{\mathbf{w}}{\text{argmax}} \left\{ -\sum_{r=1}^{R}\sum_{t=1}^{T} \gamma_t(r) \|\mathbf{o}_t - \mathbf{s}_r(\mathbf{w})\|^2_{\mathbf{C}_r} \right\} \end{aligned} \tag{3}$$

where $\mathbf{s}_r$ is the mean vector of the rth Gaussian of the adapted speaker supervector; $\|\mathbf{o}_t - \mathbf{s}_r(\mathbf{w})\|^2_{\mathbf{C}_r} \equiv (\mathbf{o}_t - \mathbf{s}_r(\mathbf{w}))'\mathbf{C}_r^{-1}(\mathbf{o}_t - \mathbf{s}_r(\mathbf{w}))$ and $\mathbf{C}_r$ is the covariance matrix of the rth Gaussian. By differentiating Eqn. (3) with respect to $\mathbf{w}$, the optimal $\hat{\mathbf{w}}$ can be found by solving a system of M linear equations (with M unknown weights, $w_m, m = 1, \ldots, M$). In theory, one may iterate the above steps in the expectation-maximization (EM) fashion until the optimal value of $\mathbf{w}$ converges. Details can be found in [7].

III. Kernel Eigenvoice Speaker Adaptation (KEV)

In [1], [2], [27], we generalized the computation of eigenvoices by performing kernel principal component analysis (PCA) instead of linear PCA. Linear PCA, on the other hand, can be considered as a special case of kernel PCA with the use of linear kernel. In this section, we will review the theory of KEV adaptation and its use of composite kernel. The description will also set the notations for the ensuing discussion of our new embedded KEV adaptation.

A. Kernel Principal Component Analysis

Let $k(\cdot,\cdot)$ be the kernel with an associated mapping φ that maps a pattern $\mathbf{x} \in \mathbb{R}^{n_2}$ (a speaker supervector in the eigenvoice approach) in the input space $\mathcal{X}$ to $\varphi(\mathbf{x}) \in \mathbb{R}^{n_3}$ (which may be infinite though) in the kernel-induced high dimensional feature space $\mathcal{F}$. Given a set of N patterns $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ contained in $\mathcal{X}$, their φ-mapped feature vectors are $\{\varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_N)\}$ contained in $\mathcal{F}$. The N mapped patterns are first centered in the feature space by finding the mean of the feature vectors $\bar{\varphi} = \frac{1}{N}\sum_{i=1}^{N}\varphi(\mathbf{x}_i)$. Let the "centered" mapping be $\tilde{\varphi}$ so that $\tilde{\varphi}(\mathbf{x}) = \varphi(\mathbf{x}) - \bar{\varphi}$. In addition, let $\mathbf{K} = [K_{ij}]$ be the kernel matrix with

$$K_{ij} \equiv k(\mathbf{x}_i, \mathbf{x}_j) \equiv \varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_j) \ , \tag{4}$$

and $\tilde{\mathbf{K}}$ be the centered version of $\mathbf{K}$ with $\tilde{K}_{ij} = \tilde{\varphi}(\mathbf{x}_i)'\tilde{\varphi}(\mathbf{x}_j)$.

To perform kernel PCA, instead of directly working on the covariance matrix in the feature space, one may carry out eigendecomposition on the centered kernel matrix $\tilde{\mathbf{K}}$ as

$$\tilde{\mathbf{K}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}' \ , \tag{5}$$

where $\mathbf{U} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$ with $\boldsymbol{\alpha}_i = [\alpha_{i1}, \ldots, \alpha_{iN}]'$, and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$. The mth orthonormal eigenvector of the covariance matrix in the feature space is then given by [21]

$$\mathbf{v}_m = \sum_{i=1}^{N} \frac{\alpha_{mi}}{\sqrt{\lambda_m}} \tilde{\varphi}(\mathbf{x}_i) \ . \tag{6}$$

Notice that all eigenvectors with non-zero eigenvalues are in the span of the φ-mapped data in the feature space.

B. Composite Kernel

As seen from Eqn. (3), an estimation of the eigenvoice weights requires the Mahalanobis distances between any adaptation data $\mathbf{o}_t$ and Gaussian means of the new speaker in the acoustic observation space $\mathcal{O}$. In the standard eigenvoice method, this is done by breaking down the speaker-adapted supervector $\mathbf{s}^{(ev)}$ to obtain its R constituent Gaussian means $\mathbf{s}_1^{(ev)}, \ldots, \mathbf{s}_R^{(ev)}$ (recall that $\mathbf{s}^{(ev)} = [\mathbf{s}_1^{(ev)'}, \ldots, \mathbf{s}_R^{(ev)'}]'$). However, in general, the use of kernel PCA does not allow us to access each constituent Gaussian directly because the state information is lost during the φ-mapping of supervectors from the input supervector space $\mathcal{X}$ to the high dimensional kernel-induced feature space $\mathcal{F}$. Our solution in KEV adaptation [1] is to preserve the necessary state information by using a possibly different mapping for each of the R constituent Gaussian means, and then apply a composite kernel function. For example, the following direct-sum composite kernel had been tried with good results:

$$\begin{aligned} k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_j) &= \begin{bmatrix} \varphi_1(\mathbf{x}_{i1}) \\ \vdots \\ \varphi_R(\mathbf{x}_{iR}) \end{bmatrix}' \begin{bmatrix} \varphi_1(\mathbf{x}_{j1}) \\ \vdots \\ \varphi_R(\mathbf{x}_{jR}) \end{bmatrix} \\ &= \sum_{r=1}^{R} \varphi_r(\mathbf{x}_{ir})'\varphi_r(\mathbf{x}_{jr}) \\ &= \sum_{r=1}^{R} k_r(\mathbf{x}_{ir}, \mathbf{x}_{jr}) \ , \end{aligned} \tag{7}$$

where $k_r(\mathbf{x}_{ir}, \mathbf{x}_{jr}), r = 1, \ldots, R$, is the kernel for the rth constituent Gaussian mean.

C. New Speaker in the Feature Space

Let the centered supervector of a new speaker found by KEV adaptation in the feature space $\mathcal{F}$ be $\tilde{\varphi}^{(kev)}(\mathbf{s})$. Conceptually, it corresponds to a speaker $\mathbf{s}$ in the input su-

pervector space, even though $\mathbf{s}$ may not exist[2]. However, the KEV adaptation method does *not* require the existence of the pre-image $\mathbf{s}$ in the input supervector space. Analogous to the formulation of a new speaker in the standard eigenvoice approach (Eqn. (1)), $\tilde{\varphi}^{(kev)}(\mathbf{s})$ is assumed to be a linear combination of the M leading eigenvoices found by kernel PCA in $\mathcal{F}$. That is, using Eqn. (1) and Eqn. (6), we have

$$\tilde{\varphi}^{(kev)}(\mathbf{s}) = \sum_{m=1}^{M} w_m \mathbf{v}_m = \sum_{m=1}^{M}\sum_{i=1}^{N} \frac{w_m \alpha_{mi}}{\sqrt{\lambda_m}} \tilde{\varphi}(\mathbf{x}_i) \ . \tag{8}$$

And the rth constituent of $\tilde{\varphi}^{(kev)}(\mathbf{s})$ is then given by

$$\tilde{\varphi}_r^{(kev)}(\mathbf{s}_r) = \sum_{m=1}^{M}\sum_{i=1}^{N} \frac{w_m \alpha_{mi}}{\sqrt{\lambda_m}} \tilde{\varphi}_r(\mathbf{x}_{ir}) \ . \tag{9}$$

Hence, the similarity between the rth constituent of the adapted model and adaptation samples in the *feature space* can be obtained as

$$\begin{aligned} & k_r^{(kev)}(\mathbf{s}_r, \mathbf{o}_t) \\ & \quad \equiv \quad \varphi_r^{(kev)}(\mathbf{s}_r)' \varphi_r(\mathbf{o}_t) \\ & \quad = \quad \left[\left(\sum_{m=1}^{M}\sum_{i=1}^{N} \frac{w_m \alpha_{mi}}{\sqrt{\lambda_m}} \tilde{\varphi}_r(\mathbf{x}_{ir}) \right) + \bar{\varphi}_r \right]' \varphi_r(\mathbf{o}_t) \\ & \quad = \quad \left[\left(\sum_{m=1}^{M}\sum_{i=1}^{N} \frac{w_m \alpha_{mi}}{\sqrt{\lambda_m}} (\varphi_r(\mathbf{x}_{ir}) - \bar{\varphi}_r) \right) + \bar{\varphi}_r \right]' \varphi_r(\mathbf{o}_t) \\ & \quad = \quad A_r(t) + \sum_{m=1}^{M} \frac{w_m}{\sqrt{\lambda_m}} B_r(m,t) \ , \end{aligned} \tag{10}$$

where $\bar{\varphi}_r = \frac{1}{N}\sum_{i=1}^{N} \varphi_r(\mathbf{x}_{ir})$ is the rth part of $\bar{\varphi}$,

$$A_r(t) = \bar{\varphi}_r' \varphi_r(\mathbf{o}_t) = \frac{1}{N}\sum_{i=1}^{N} k_r(\mathbf{x}_{ir}, \mathbf{o}_t) \ , \tag{11}$$

and

$$B_r(m,t) = \sum_{i=1}^{N} \alpha_{mi} \left(k_r(\mathbf{x}_{ir}, \mathbf{o}_t) - A_r(t) \right) \ . \tag{12}$$

D. ML Estimation of Kernel Eigenvoice Weights

To estimate the kernel eigenvoice weights $\mathbf{w}$, one will express the Q function, hence, the Mahalanobis distance $\|\mathbf{o}_t - \mathbf{s}_r\|^2_{\mathbf{C}_r}$ in terms of the kernel function. This can usually be done with many common kernels (Appendix I). Good results had been obtained using the following isotropic Gaussian kernel,

$$k_r^{(kev)}(\mathbf{s}_r, \mathbf{o}_t) = \exp(-\beta_r \|\mathbf{o}_t - \mathbf{s}_r\|^2_{\mathbf{C}_r}) \ , \quad r = 1, \ldots, R. \tag{13}$$

[2]The notation for a new speaker in the feature space requires some explanation. If $\mathbf{s}$ exists, then its centered image is $\tilde{\varphi}^{(kev)}(\mathbf{s})$. However, since the pre-image of a speaker found in the feature space may not exist [20], the notation $\tilde{\varphi}^{(kev)}(\mathbf{s})$ is not exactly correct. However, the notation is adopted for its intuitiveness and the readers are advised to infer the existence of $\mathbf{s}$ based on the context.

Then the Mahalanobis distance between the rth constituent of the adapted speaker model and the adaptation data in the *input speaker supervector space* can be found via the rth constituent kernel as follows:

$$\|\mathbf{o}_t - \mathbf{s}_r\|^2_{\mathbf{C}_r} = -\frac{1}{\beta_r} \log \left(k_r^{(kev)}(\mathbf{s}_r, \mathbf{o}_t) \right) \ .$$

Hence, the KEV weights $\mathbf{w}$ may be estimated by modifying the $Q(\mathbf{w})$ function of Eqn. (3) as

$$Q(\mathbf{w}) = \sum_{r=1}^{R}\sum_{t=1}^{T} \frac{\gamma_t(r)}{\beta_r} \log \left(k_r^{(kev)}(\mathbf{s}_r, \mathbf{o}_t) \right) \ . \tag{14}$$

Its derivative with respect to each KEV weight w_m is given by

$$\frac{\partial Q}{\partial w_m} = \frac{1}{\sqrt{\lambda_m}} \sum_{r=1}^{R}\sum_{t=1}^{T} \frac{\gamma_t(r) B_r(m,t)}{\beta_r k_r^{(kev)}(\mathbf{s}_r, \mathbf{o}_t)} \ , \quad m = 1, \ldots, M \ . \tag{15}$$

Due to the nonlinear nature of kernel PCA, and thus Eqn. (15), there is no closed form solution for the optimal $\hat{\mathbf{w}}$. The optimal kernel eigenvoice weights are solved using generalized expectation-maximization (GEM) algorithm [28] in which numerical methods like gradient ascent method is used to improve the value of $\mathbf{w}$ during each maximization step.

IV. Embedded Kernel Eigenvoice Speaker Adaptation (eKEV)

In our new embedded kernel eigenvoice (eKEV) speaker adaptation method [29], [30], all online kernel evaluations are eliminated by finding an approximate *pre-image* of the adapted model found by KEV adaptation which resides in the kernel-induced feature space $\mathcal{F}$. Conceptually, if ν is the adapted model found by KEV adaptation in $\mathcal{F}$, we would like to map it back to its pre-image $\varphi^{-1}(\nu)$ in the input space $\mathcal{X}$. However, the exact pre-image, in general, does not exist, and one can only settle for an approximate solution. The problem is known as the "pre-image problem" in the kernel method community.

Here we would like to apply an analytical solution we previously proposed in [25] to find the pre-image of the KEV adapted model. The method uses the distances between the expected (approximate) pre-image and a set of "*reference points*" (which in our case will be called "*reference speakers*") as constraints and solves for the optimal pre-image in the least-square sense[3]. In general, these reference speakers are independent of the speaker-adapted (SA) model to be found, but, as will be discussed in Experiment 2 of Section V-A.3, better performance is obtained if they are sufficiently close to the expected SA model. Although the definition as well as the size of the set of reference speakers can be important to the performance of eKEV adaptation in practice, they are immaterial to the theory of the adaptation method; we will leave their discussion to Section V.

[3]It is analogous to finding the location of an object using a set of global positioning system satellites.

For consistency with the description of KEV adaptation in Section III, the composite kernels again will be used for the following discussion. However, we would like to emphasize that the use of composite kernels is not necessary, and one may perform eKEV adaptation with common "non-composite" kernels. Nevertheless, since Gaussian kernel is commonly used in the kernel community which can be also viewed as a tensor product composite kernel, our discussion using composite kernels is applicable to the common Gaussian kernel as well.

A. eKEV Algorithm Formulation

The eKEV adaptation method is illustrated pictorially in Fig. 1. In the figure, all the five training speakers $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_5\}$ are used to derive the eigenvoices in the feature space $\mathcal{F}$ by kernel PCA. The new speaker-adapted model[4] $\varphi^{(kev)}(\mathbf{s}_x)$ in the feature space is restricted to the feature subspace spanned by the selected kernel eigenvoices. For many commonly used kernels, there is a simple relationship between the input-space distance and the feature-space distance. Thus, from the distances between $\varphi^{(kev)}(\mathbf{s}_x)$ and the feature-space reference speakers $\{\varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2), \varphi(\mathbf{x}_3)\}$, one can also obtain the corresponding distances between $\mathbf{s}_x^{(ekev)}$, the (approximate) pre-image of $\varphi^{(kev)}(\mathbf{s}_x)$, and the input-space reference speakers $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$. By confining $\mathbf{s}_x^{(ekev)}$ to lie in the subspace spanned by these three reference speakers, it is shown in [25] that $\mathbf{s}_x^{(ekev)}$ can be analytically obtained by satisfying all three distance constraints between $\mathbf{s}_x^{(ekev)}$ and $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ in the least-squares sense. Mathematically, this mainly relies on computing the singular value decomposition (SVD) of the matrix $[\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$, which obtains a basis in the subspace spanned by these three reference speakers.

In the algorithm, two sets of distances are actually computed in the *input speaker supervector* space $\mathcal{X}$: the Euclidean distances $\mathbf{d}_0 = [\|\mathbf{z}_1\|^2, \|\mathbf{z}_2\|^2, \ldots, \|\mathbf{z}_n\|^2]' \in \mathbb{R}^n$ between the n reference speakers and their centroid, and the Euclidean distances $\mathbf{d} = [d_1, d_2, \ldots, d_n]' \in \mathbb{R}^n$ between the n reference speakers and the pre-image $\mathbf{s}_x^{(ekev)}$. Both set of distances are labelled in Fig. 1 and will be explained in details in STEP 2 and STEP 4 below.

Details of the method are described step-by-step as follows.

STEP 1: Variance Normalization

Because the pre-image finding algorithm uses Euclidean distance constraints, whereas the Gaussian kernel we employ in KEV or eKEV adaptation involves Mahalanobis distance (between speaker supervectors or acoustic observations), we will first normalize each of the R constituents of any speaker supervector $\mathbf{x}$ by its own covariance. The normalized model of $\mathbf{x}$ is represented by $\mathbf{y} = \mathbf{C}^{-\frac{1}{2}}\mathbf{x}$ where

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}_R \end{bmatrix} .$$

Hereafter, the pre-image of the new speaker-adapted model will be represented by $\mathbf{s}_x^{(ekev)}$ in the original input supervector space, and $\mathbf{s}_y^{(ekev)}$ in the normalized input space.

STEP 2: Finding the Distance between Reference Speakers and Their Centroid in the Input Space

Without loss of generality, let $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ be the n *reference speakers*, and they are collected into a $n_2 \times n$ matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$. (Recall that n_2 is the dimension of each speaker supervector.) They are first centered at their centroid $\bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i$ by using the $n \times n$ centering matrix $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'$ so that the centered $\mathbf{Y}$ is given by $\tilde{\mathbf{Y}} = \mathbf{Y}\mathbf{H}$. Assuming that these n reference speakers span a q-dimensional space (i.e. the rank of $\mathbf{Y}$ is q), we can obtain the SVD of $\tilde{\mathbf{Y}}$ as

$$\tilde{\mathbf{Y}} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}' = \mathbf{U}\mathbf{Z} , \tag{16}$$

where $\mathbf{U} = [\mathbf{e}_1, \ldots, \mathbf{e}_q]$ is an $n_2 \times q$ matrix with orthonormal columns $\mathbf{e}_i$; $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_q)$ is a $q \times q$ diagonal matrix containing the eigenvalues; $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$ is a $q \times n$ matrix with columns $\mathbf{z}_i$ being the projections of $\mathbf{y}_i$ onto the $\mathbf{e}_j$'s.

The squared Euclidean distance of each $\mathbf{y}_i, i = 1, \ldots, n$, from the centroid $\bar{\mathbf{y}}$ can now be easily computed as $\|\mathbf{z}_i\|^2$. They are collected into an n-dimensional vector,

$$\mathbf{d}_0 = [\|\mathbf{z}_1\|^2, \|\mathbf{z}_2\|^2, \ldots, \|\mathbf{z}_n\|^2]' \in \mathbb{R}^n . \tag{17}$$

STEP 3: Similarity between the New Speaker and the Reference Speakers in the Feature Space

Analogous to Eqn. (10), the similarity between the rth constituent of the SA model $\mathbf{s}_{yr}^{(ekev)}$ and that of the jth reference speaker $\mathbf{y}_{jr}$ in the kernel-induced feature space can be found by replacing $\mathbf{o}_t$ of the equation by $\mathbf{y}_{jr}$ as follows:

$$k_r^{(kev)}(\mathbf{s}_{yr}^{(ekev)}, \mathbf{y}_{jr}) = A_r(j) + \sum_{m=1}^{M} \frac{w_m}{\sqrt{\lambda_m}} B_r(m, j) , \tag{18}$$

where

$$A_r(j) = \frac{1}{N}\sum_{i=1}^{N} k_r(\mathbf{y}_{ir}, \mathbf{y}_{jr}) , \tag{19}$$

and

$$B_r(m, j) = \sum_{i=1}^{N} \alpha_{mi} \left(k_r(\mathbf{y}_{ir}, \mathbf{y}_{jr}) - A_r(j)\right) . \tag{20}$$

[4]The notation of the various models related to the new speaker-adapted (SA) model may need further explanation. $\mathbf{s}_x^{(ekev)}$ is used to represent the final SA model in the input space. Its *exact* image in the feature space should be $\varphi(\mathbf{s}_x^{(ekev)})$. On the other hand, conceptually eKEV adaptation first employs KEV adaptation to compute an *implicit* SA model $\varphi^{(kev)}(\mathbf{s}_x)$ in the feature space and $\mathbf{s}_x^{(ekev)}$ is found as an *approximate* pre-image of $\varphi^{(kev)}(\mathbf{s}_x)$. Notice that, in general, $\varphi(\mathbf{s}_x^{(ekev)})$ and $\varphi^{(kev)}(\mathbf{s}_x)$ are different, and they are assumed to be close to each other in this paper.

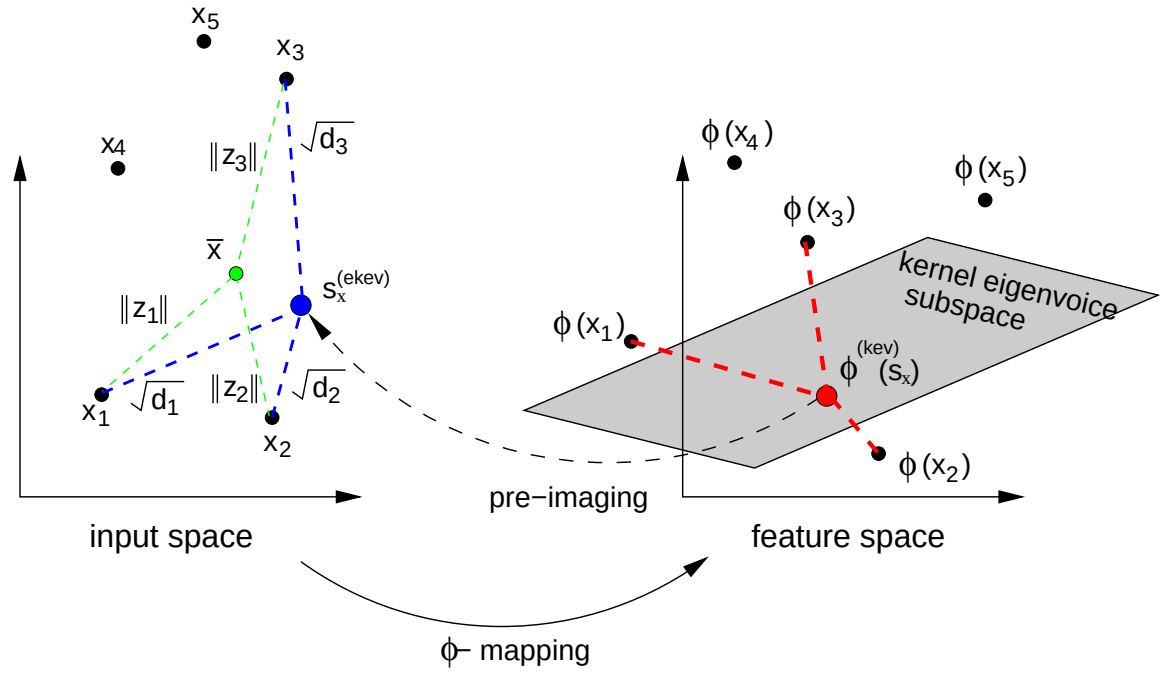

Fig. 1. The eKEV adaptation method.

STEP 4: Finding the Distance Constraints between the New Speaker and the Reference Speakers in the Input Space

It is further assumed that the required pre-image $\mathbf{s}_y^{(ekev)}$ lies in the span of the n reference speakers, and its squared Euclidean distances from them are collected into the following n-dimensional vector:

$$\mathbf{d} = [d_1, d_2, \ldots, d_n]' \in \mathbb{R}^n \ . \tag{21}$$

The squared Euclidean distance d_j between $\mathbf{s}_y^{(ekev)}$ and the jth reference speaker can be computed from the distances between each of their corresponding R constituents since

$$d_j \equiv \|\mathbf{s}_y^{(ekev)} - \mathbf{y}_j\|^2 = \sum_{r=1}^{R} \|\mathbf{s}_{yr}^{(ekev)} - \mathbf{y}_{jr}\|^2 \equiv \sum_{r=1}^{R} d_{jr} \ .$$

If the direct-sum composite kernel of Eqn. (7) is used, and each constituent kernel is similar to the Gaussian kernel of Eqn. (13), then we have

$$k_r^{(kev)}(\mathbf{s}_{yr}^{(ekev)}, \mathbf{y}_{jr}) = e^{-\beta_r \|\mathbf{s}_{yr}^{(ekev)} - \mathbf{y}_{jr}\|^2} = e^{-\beta_r d_{jr}} \ .$$

Therefore, the distance between $\mathbf{s}_y^{(ekev)}$ and the jth reference speaker $\mathbf{y}_j$ in the *input space* can be deduced from their similarity in the *feature space* using the corresponding kernel value as follows:

$$d_j \equiv \sum_{r=1}^{R} d_{jr} = -\sum_{r=1}^{R} \frac{1}{\beta_r} \log k_r^{(kev)}(\mathbf{s}_{yr}^{(ekev)}, \mathbf{y}_{jr}) \ . \tag{22}$$

Notice that each distance component d_{jr} can be computed from the kernel evaluation of $k_r^{(kev)}(\mathbf{s}_{yr}^{(ekev)}, \mathbf{y}_{jr})$ as given by Eqns. (18, 19, 20). The kernel evaluation does not involve any adaptation or testing observations, though it depends on the adaptation observations indirectly through the eigenvoice weights $\mathbf{w}$. Instead, it only requires the evaluation of constituent kernel values $k_r(\mathbf{y}_{ir}, \mathbf{y}_{jr}), r = 1, \ldots, R$, between any two training speakers which can be pre-computed offline.

STEP 5: Finding the Pre-image

From [25], an approximate (normalized) pre-image that optimally satisfies the distance constraints in $\mathbf{d}$ of Eqn. (21) in the least-squares sense is given by the following equation:

$$\mathbf{s}_y^{(ekev)} = -\frac{1}{2}\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{V}'(\mathbf{d} - \mathbf{d}_0) + \bar{\mathbf{y}} \ , \tag{23}$$

where $\mathbf{U}$, $\mathbf{\Lambda}$, and $\mathbf{V}$ are the results of SVD of $\tilde{\mathbf{Y}}$ given by Eqn. (16). To show the dependence of $\mathbf{s}_y^{(ekev)}$ on the eigenvoice weights, let's re-write $\mathbf{s}_y^{(ekev)}$ as

$$\mathbf{s}_y^{(ekev)}(\mathbf{w}) = \mathbf{P}\mathbf{d}(\mathbf{w}) + \mathbf{q} \tag{24}$$

where

$$\mathbf{P} = -\frac{1}{2}\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{V}' \ , \tag{25}$$

and

$$\mathbf{q} = -\mathbf{P}\mathbf{d}_0 + \bar{\mathbf{y}} \ . \tag{26}$$

Notice that only $\mathbf{d} \in \mathbb{R}^n$ depends on $\mathbf{w} \in \mathbb{R}^M$ as shown in Eqns. (18, 22), and both $\mathbf{P} \in \mathbb{R}^{n_2 \times n}$ and $\mathbf{q} \in \mathbb{R}^{n_2}$ are independent of $\mathbf{w}$.

Finally, the speaker's unnormalized adapted model $\mathbf{s}_x^{(ekev)}(\mathbf{w})$ can be obtained from Eqn. (24) as

$$\mathbf{s}_x^{(ekev)}(\mathbf{w}) = \mathbf{C}^{\frac{1}{2}}\mathbf{s}_y^{(ekev)} = \mathbf{C}^{\frac{1}{2}}(\mathbf{P}\mathbf{d}(\mathbf{w}) + \mathbf{q}) \ . \tag{27}$$

STEP 6: Gradient Computation

From Eqn.(27), the rth constituent of a new speaker's model $\mathbf{s}_{xr}^{(ekev)}$, which is also the mean vector of the rth Gaussian of his/her HMMs, is given by

$$\mathbf{s}_{xr}^{(ekev)}(\mathbf{w}) = \mathbf{C}_r^{\frac{1}{2}}(\mathbf{P}_r\mathbf{d}(\mathbf{w}) + \mathbf{q}_r) \ , \tag{28}$$

where $\mathbf{P}_r \in \mathbb{R}^{n_1 \times n}$ consists of the $((r-1)n_1+1)$th to (rn_1)th rows of $\mathbf{P}$ that are used in the computation of $\mathbf{s}_{yr}^{(ekev)}(\mathbf{w})$, and $\mathbf{q}_r = -\mathbf{P}_r\mathbf{d}_0 + \bar{\mathbf{y}}_r$. Substituting Eqn. (28) into the $Q(\mathbf{w})$ function of Eqn. (3), and differentiating the result w.r.t. the mth weight w_m, we obtain the following weight gradient:

$$\frac{\partial Q}{\partial w_m} = \sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_t(r)(\mathbf{o}_t - \mathbf{s}_{xr}^{(ekev)}(\mathbf{w}))'\mathbf{C}_r^{-1}\frac{\partial \mathbf{s}_{xr}^{(ekev)}(\mathbf{w})}{\partial w_m} \ . \tag{29}$$

From Eqn. (27), we may obtain the derivative of $\mathbf{s}_{xr}^{(ekev)}(\mathbf{w})$ as

$$\frac{\partial \mathbf{s}_{xr}^{(ekev)}(\mathbf{w})}{\partial w_m} = \mathbf{C}_r^{\frac{1}{2}}\mathbf{P}_r\frac{\partial \mathbf{d}(\mathbf{w})}{\partial w_m} \ . \tag{30}$$

Combining Eqn. (22) and Eqn. (18), and differentiating the result w.r.t. $w_m, m = 1, \ldots, M$, the jth element of $\dfrac{\partial \mathbf{d}(\mathbf{w})}{\partial w_m}$ is found to be:

$$\frac{\partial d_j}{\partial w_m} = -\frac{1}{\sqrt{\lambda_m}}\sum_{r=1}^{R}\frac{B_r(m,j)}{\beta_r k_r^{(kev)}(\mathbf{s}_{yr}^{(ekev)}(\mathbf{w}), \mathbf{y}_{jr})} \ , \quad j = 1, \ldots, n \ . \tag{31}$$

Finally, substituting the results of Eqns. (30, 31) onto Eqn. (29), the derivative of $Q(\mathbf{w})$ w.r.t. each eigenvoice weight w_m can be readily obtained.

STEP 7: Estimation of Eigenvoice Weights

The gradient of Eqn. (29) is nonlinear in $\mathbf{w}$ and there is no closed form solution for the optimal $\hat{\mathbf{w}}$. Again, as in KEV adaptation, we apply GEM algorithm to find the optimal weights. GEM is similar to the conventional EM algorithm except for the maximization step: EM looks for a $\mathbf{w}$ that maximizes the expected likelihood found in the E-step but GEM only requires a $\mathbf{w}$ that improves the likelihood. Many numerical methods may be used to update $\mathbf{w}$ based on the derivatives of Q. In this paper, gradient-based algorithms are used to compute $\mathbf{w}(l)$ from $\mathbf{w}(l-1)$ based only on the first-order derivative: for the small vocabulary TIDIGITS evaluation, the simple gradient ascent algorithm is employed; for the large vocabulary WSJ0 evaluation, the more advanced BFGS method is used for faster convergence.

B. Robust eKEV Adaptation

Since the amount of data in fast speaker adaptation is so small, the adaptation performance may vary widely as overfitting may readily occur. To get a more robust performance, the pre-image of the speaker-adapted model found by eKEV adaptation $\mathbf{s}_x^{(ekev)}$ is interpolated with the speaker-independent (SI) supervector $\mathbf{x}^{(si)}$ to obtain the final robust SA model $\mathbf{s}_x^{(rekev)}$. That is,

$$\mathbf{s}_x^{(rekev)} = w_0\mathbf{x}^{(si)} + (1-w_0)\mathbf{s}_x^{(ekev)} \ , \ 0 \le w_0 \le 1 \ . \tag{32}$$

The required derivatives for gradient ascent are then updated as follows:

$$\frac{\partial Q}{\partial w_m} = \sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_t(r)(\mathbf{o}_t - \mathbf{s}_{xr}^{(rekev)}(\mathbf{w}))'\mathbf{C}_r^{-1}\frac{\partial \mathbf{s}_{xr}^{(rekev)}(\mathbf{w})}{\partial w_m} \tag{33}$$

for $m = 0, 1, \ldots, M$, where

$$\frac{\partial \mathbf{s}_{xr}^{(rekev)}}{\partial w_0} = \mathbf{x}_r^{(si)} - \mathbf{s}_{xr}^{(ekev)} \ , \tag{34}$$

and

$$\frac{\partial \mathbf{s}_{xr}^{(rekev)}}{\partial w_m} = (1-w_0)\frac{\partial \mathbf{s}_{xr}^{(ekev)}}{\partial w_m} \ . \tag{35}$$

The derivative $\dfrac{\partial \mathbf{s}_{xr}^{(ekev)}}{\partial w_m}$ in the last equation is again given by Eqns. (30, 31).

Similar robust adaptation method had been proposed in our previous work on KEV adaptation [1].

C. Remarks on Speed

The use of kernel methods, in general, may significantly increase the total computation. Both KEV and eKEV adaptation have to compute the kernel matrix (Eqn. (4)) in order to perform kernel PCA to derive the kernel eigenvoices (Eqn. (8)). This requires kernel evaluations $k_r(\mathbf{x}_{ir}, \mathbf{x}_{jr})$ between any two training speaker supervectors, which, fortunately, can be pre-computed offline. In addition, KEV adaptation has to compute kernel evaluations $k_r(\mathbf{x}_{jr}, \mathbf{o}_t)$ between any training speaker supervector $\mathbf{x}_j$ and adaptation speech frames $\mathbf{o}_t$ during adaptation, and $k_r^{(kev)}(\mathbf{s}_r, \mathbf{o}_t)$ between the adapted model $\varphi^{(kev)}(\mathbf{s})$ and testing speech frames $\mathbf{o}_t$ during recognition (Eqns. (10, 11, 12)). Obviously, these kernel values must be computed online during adaptation and recognition. On the other hand, no observations are involved in any kernel evaluations in eKEV adaptation: adaptation only requires kernel evaluations between any reference speaker supervectors and the training speaker supervectors (Eqns. (18, 19, 20)), which are only a subset of the kernel evaluations that have been already computed for kernel PCA. Thus, eKEV adaptation is expected to be faster than KEV adaptation in both adaptation and recognition. In fact, since an explicit speaker-adapted model $\mathbf{s}_x^{(ekev)}$ is produced by eKEV adaptation, subsequent recognition should be as fast as normal HMM decoding.

V. Experimental Evaluation

The proposed embedded kernel eigenvoice (eKEV) adaptation method was evaluated on a small-vocabulary continuous speech recognition task using the TIDIGITS speech

corpus [31], and on a large-vocabulary continuous speech recognition (LVCSR) task using the Wall Street Journal (WSJ0) speech corpus. We first used the simpler task of TIDIGITS to familiarize ourselves with the behavior of the new eKEV adaptation method. This includes the investigation of different methods to find the set of reference speakers, the effect of its size on the adaptation performance, and the speed of eKEV adaptation. Then its adaptation performance was compared with other common adaptation methods on both corpora. Specifically, the following models or adaptation methods were compared:

SI*:* the baseline speaker-independent model.

(robust) eKEV*:* the speaker-adapted (SA) model found by our new robust eKEV adaptation method as described by Eqn. (32) of Section IV-B.

(robust) KEV*:* the SA model $\varphi^{(rkev)}(\mathbf{s})$ found by our previously robust KEV adaptation method as described in [1]. It is the result of interpolation between the SA model $\varphi^{(kev)}(\mathbf{s})$ found by KEV adaptation and the φ-mapped SI supervector $\varphi(\mathbf{x}^{(si)})$ in the feature space given by the following formula:

$$\begin{aligned}\varphi^{(rkev)}(\mathbf{s}) &= w_0\varphi(\mathbf{x}^{(si)}) + (1-w_0)\varphi^{(kev)}(\mathbf{s}),\\ &\quad 0.0 \le w_0 \le 1.0\ . \end{aligned} \tag{36}$$

This is analogous to the robust eKEV adaptation.

(robust) EV*:* the SA model $\mathbf{s}^{(rev)}$ computed as the interpolation between the SI supervector $\mathbf{x}^{(si)}$ and the supervector $\mathbf{s}^{(ev)}$ found by EV adaptation. That is,

$$\mathbf{s}^{(rev)} = w_0\mathbf{x}^{(si)} + (1-w_0)\mathbf{s}^{(ev)}\ , \quad 0.0 \le w_0 \le 1.0\ , \tag{37}$$

where w_0 is estimated jointly with the other eigenvoice weights by maximizing the adaptation data. In this paper, EV was actually implemented as a special case of KEV adaptation using a linear kernel [5].

MAP*:* the SA model found by MAP adaptation [5].

MLLR*:* the SA model found by MLLR adaptation [6].

A. Evaluation on Small-vocabulary Continuous Speech Recognition

In this part, we would use simple digit models to investigate the behavior of eKEV adaptation on the smaller TIDIGITS corpus. The simple task allows us to run many experiments for the investigation.

A.1 TIDIGITS Corpus

The TIDIGITS corpus contains clean connected-digit utterances sampled at 20 kHz. It is divided into a standard training set and a test set. There are 163 speakers (of both genders) in each set, each pronouncing 77 utterances of one to seven digits (out of the eleven digits: "0", "1", ..., "9", and "oh"). There is no overlap between the training speakers and test speakers. The speaker characteristics are quite diverse with speakers coming from 22 dialect regions of USA, and their ages ranging from 6 to 70 years old.

A.2 Acoustic Models

All training data were processed to extract 12 mel-frequency cepstral coefficients and the normalized frame energy from each speech frame of 25 ms at every 10 ms. Each of the eleven digit models was a strictly left-to-right HMM comprising 16 states with one diagonal-covariance Gaussian per state. In addition, there were a 3-state "sil" model to capture silence and a 1-state "sp" model to capture short pauses between digits. All HMMs were trained by the EM algorithm. Thus, the dimension of the observation space n_1 is 13 and that of the speaker supervector space n_2 is 11 models $\times$ 16 states/model $\times$ 13/state = 2288.

Firstly, a set of speaker-independent (SI) digit models were trained. Then a set of speaker-dependent (SD) digit models were trained for each individual training speaker by borrowing the covariances and transition matrices from the corresponding SI models, and only the Gaussian means were estimated. Furthermore, the "sil" and "sp" models were simply copied to each SD model. In our pilot experiments, it was found that SD models trained in this way performed better than SD models that did not share any model parameters with the SI models.

On the test data, the word accuracies of the baseline SI model is 96.25%[6] In addition, we also checked the quality of the SD models using a 7-fold cross-validation: for each training speaker, his data was divided into 7 roughly equal subsets, and 6 subsets were used for training his acoustic model which was then tested on the remaining subset. The average word accuracy over all 163 training speakers is found to be 98.76%. It shows that our way of training SD models produces sufficiently good acoustic models for subsequent eigenvoice determination.

A.3 Experiments

In all experiments, only the training set was used to train the SI HMMs and SD HMMs from which the SI and SD speaker supervectors were derived. Adaptation was performed on the test speakers. Five, ten, and twenty digits were used for adaptation, which correspond to an average of 2.1s, 4.1s, and 9.6s of adaptation speech (or 3.0s, 5.5s, and 13.0s of speech if the leading and ending silences are counted as well). To improve the statistical reliability of

[5]Using the composite linear kernel: $k_r(\mathbf{x},\mathbf{y}) = \mathbf{x}'C_r^{-1}\mathbf{y}$, and Eqn. (40) in the Appendix, the Mahalanobis distance in the $Q(\mathbf{w})$ function can be expressed as: $\|\mathbf{o}_t - \mathbf{s}_r\|^2_{\mathbf{C}_r} = \mathbf{o}_t'C_r^{-1}\mathbf{o}_t + k_r(\mathbf{s}_r(\mathbf{w}),\mathbf{s}_r(\mathbf{w})) - 2k_r(\mathbf{s}_r(\mathbf{w}),\mathbf{o}_t)$. The term $k_r(\mathbf{s}_r(\mathbf{w}),\mathbf{o}_t)$ can be computed by Eqn. (10), while the term $k_r(\mathbf{s}_r(\mathbf{w}),\mathbf{s}_r(\mathbf{w})) = \varphi_r^{(kev)}(\mathbf{s}_r)'\varphi_r^{(kev)}(\mathbf{s}_r)$ can be computed from Eqn. (9). As a result, the $Q(\mathbf{w})$ function is quadratic and its derivative is linear, and the optimal weights can be found by solving a system of linear equation as expected.

[6]The word accuracy of our SI model is not as good as the best reported result on TIDIGITS which is about 99.7%. The main reason is that we used only 13-dimensional static cepstra and energy features, and each state was modeled by a single Gaussian. Furthermore, one of the methods we were comparing with, namely, KEV adaptation requires online computation of many kernel function values and is computationally very expensive. Since the task is mainly employed to investigate the behavior of the new eKEV adaptation method, we think the use of the simple model is justified.

the results, all results are the averages of a 5-fold cross-validation over all 163 test speakers. Moreover, all adaptation experiments were performed in the supervised mode[7], and only one GEM iteration was run as in some preliminary experiments it was found that more GEM iterations did not further improve the adaptation performance.

Parameter initialization and settings

In the following TIDIGITS experiments, the simple iterative gradient ascent algorithm was used to compute the (locally) optimal eigenvoice weights in each maximization step of the GEM algorithm. Proper initialization of various system parameters can be important for its success.

- Kernel eigenvoice weights initialization: Since we are adapting the SI model to the new speaker, it is reasonable to start searching from the kernel eigenvoice weights of the speaker supervector of the SI model $\mathbf{x}^{(si)}$. For eKEV adaptation, these kernel eigenvoice weights were found by projecting the normalized SI supervector $\mathbf{y}^{(si)} = \mathbf{C}^{-\frac{1}{2}}\mathbf{x}^{(si)}$ onto each kernel eigenvoice $\mathbf{v}_m, m = 1, \ldots, M$, in the kernel-induced feature space as follows:

$$\begin{aligned} w_m(0) &= \mathbf{v}_m'\tilde{\varphi}(\mathbf{y}^{(si)}) \\ &= \sum_{i=1}^{N} \frac{\alpha_{mi}}{\sqrt{\lambda_m}} \tilde{\varphi}(\mathbf{y}_i)'\tilde{\varphi}(\mathbf{y}^{(si)}) \\ &= \sum_{i=1}^{N} \frac{\alpha_{mi}}{\sqrt{\lambda_m}} (\varphi(\mathbf{y}_i) - \bar{\varphi})'(\varphi(\mathbf{y}^{(si)}) - \bar{\varphi}) \\ &= \sum_{i=1}^{N} \frac{\alpha_{mi}}{\sqrt{\lambda_m}} \Bigg[k(\mathbf{y}_i, \mathbf{y}^{(si)}) + \frac{1}{N^2} \sum_{p=1}^{N}\sum_{q=1}^{N} k(\mathbf{y}_p, \mathbf{y}_q) \\ &\quad - \frac{1}{N} \sum_{p=1}^{N} \Big(k(\mathbf{y}_i, \mathbf{y}_p) + k(\mathbf{y}^{(si)}, \mathbf{y}_p) \Big) \Bigg] . \end{aligned}$$

- The width of all direct-sum composite Gaussian kernels were set identical to the value of 0.0005. That is, $\beta_r = \beta = 0.0005$ for $r = 1, \ldots, R$. The value was empirically found to give good performance for KEV adaptation on a subset of training speakers [1].
- The initial learning rate was set empirically to 0.0001.
- The number of kernel eigenvoices was fixed to 7 as it empirically gave the best performance in some preliminary experiments.
- The gradient ascent algorithm stopped when either the relative improvement on the likelihood of the adaptation data was less than 0.00015, or 1000 iterations was reached.

Experiment 1: Different methods to find the reference speakers

[7]According to our previous work on KEV adaptation [1], supervised KEV adaptation and unsupervised KEV adaptation on this TIDIGITS task had very similar performance. We expect eKEV adaptation to have the same behavior too.

TABLE I
EFFECT OF DIFFERENT TYPES OF REFERENCE SPEAKERS ON EKEV ADAPTATION ON TIDIGITS. (THE NUMBER OF REFERENCE SPEAKERS IS 10.)

Amount of Adaptation Data	ML Neighbors	SI Neighbors	
		Euclidean	Mahalanobis
2.1s	**97.41**	96.33	96.52
4.1s	**97.53**	96.43	96.60
9.6s	**97.58**	96.50	96.68

The computation of the pre-image relies on its distances to a set of reference speakers. In the reference paper of the pre-image finding method [25], the neighbors of a de-noised image in the kernel-induced feature space are used as the reference set. However, in our problem, the whereabouts of the speaker-adapted (SA) model is not known beforehand, neither in the feature space nor in the input supervector space, and so are the locations of its neighbors. In this paper, we investigated two ways to determine the initial set of reference speakers of the SA model to be found:

- SI model's neighbors: If no additional information is available, it is reasonable to start with the neighbors of the SI model since the adaptation method begins its search from the SI model. The neighbors can be computed using either Euclidean distance or Mahalanobis distance. One advantage of using SI neighbors is that they can be computed *offline*.
- Maximum likelihood (ML) neighbors: Conceptually, since we are using the maximum likelihood criterion for determining the SA model, it should be close to those training speakers that also have high likelihood of the adaptation data.

The effect of different types of neighbors on the adaptation performance of the eKEV method is shown in Table I. The number of neighbors was fixed to 10 for the investigation. From the results, it indeed seems that the final SA model is closer to its ML neighbors than the SI neighbors. Since there can be many local maxima in the solution of the gradient method, we hypothesize that a good initialization of its neighborhood to the ML neighbors may have avoided the poorer local maxima.

In the last experiment, the neighbors were initialized and pre-determined before the start of eKEV adaptation and remained unchanged during the course. In general, these neighbors may be updated after each GEM iteration to the real neighbors of the SA model as determined by their Mahalanobis distances. We had run additional experiments with such neighbor updates in the case of ML neighbors. It was found that most of the neighbors remained the same, and the final model had very similar performance as that of the SA model obtained without neighbor updates.

Experiment 2: Effect of the number of ML reference speakers

Another issue about the reference speakers is how many

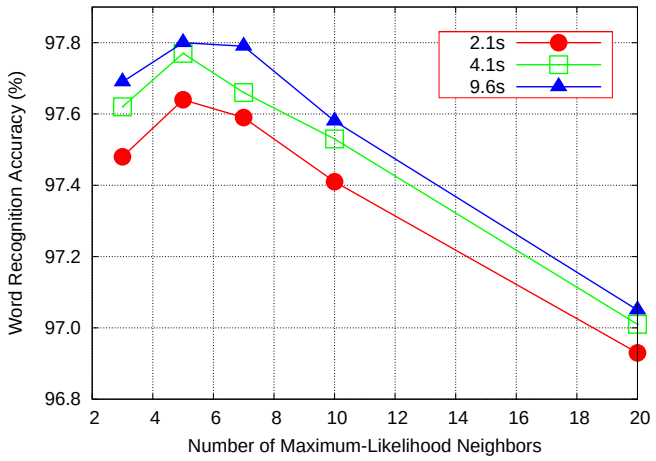

Fig. 2. Effect of the number of maximum-likelihood reference speakers on eKEV adaptation on TIDIGITS.

of them are adequate. On the one hand, adaptation is faster with fewer reference speakers as fewer distance constraints have to be computed. On the other hand, the current method of using distances from reference speakers of a neighborhood to find the pre-image tries to exploit localized information to constrain the solution space. If there are too few reference speakers[8], the distance constraints may be too weak to lead to a good pre-image solution. However, if too many reference speakers are included, those that are far away will dominate the distance constraints (as the pre-image is obtained from a least-squares approximation), and the idea of using localized information for the determination of the pre-image is not utilized.

Fig. 2 shows the performance of various adapted models found by eKEV adaptation using different numbers of ML neighbors. It is concluded that for this particular problem, five ML neighbors give the best performance. In practice, the optimal number of reference speakers may be determined by cross-validation.

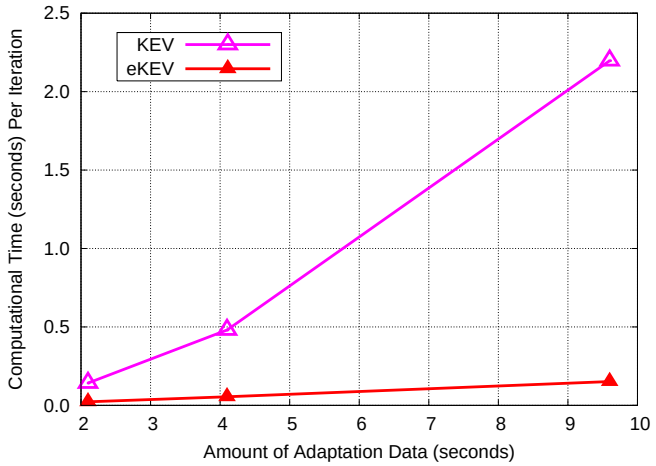

Fig. 3. Computational time taken by each gradient ascent iteration during eKEV adaptation on TIDIGITS.

[8]Since the pre-image is always constrained to lie on the span of neighbors, the theoretical minimum number of neighbors is 2.

Experiment 3: Speed Comparison

The main objective of eKEV adaptation is to improve the speed of adaptation and recognition of KEV adaptation as discussed in Section IV-C. Figure 3 shows that the adaptation speed of eKEV adaptation is indeed an order of magnitude faster than that of KEV adaptation. (The exact speedup factors by eKEV adaptation over KEV adaptation are 6.24, 8.75, and 14.5 for 2.1s, 4.1s, and 9.6s of adaptation speech respectively.) We also checked the recognition speed of their adapted models. It was found that, on average, KEV adapted models took 227s to recognize one second of test speech, while eKEV adapted models — regular HMMs — only took 1.67s; that is, a speed up of 136 times. (All experiments were run on a Pentium III 1GHz machine with 512MB RAM.)

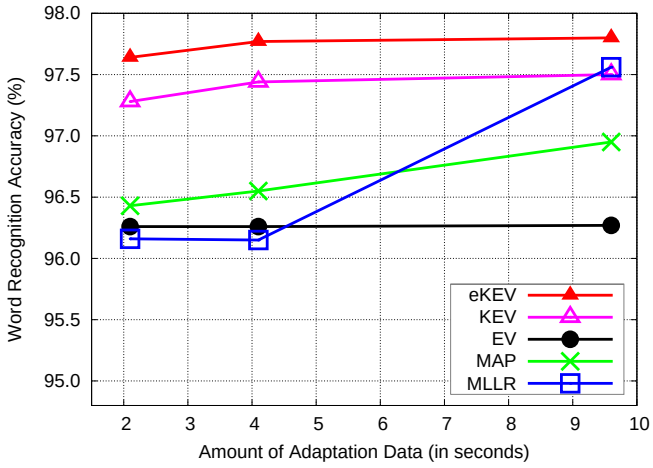

Fig. 4. Performance comparison among MLLR, MAP, EV, KEV, and eKEV adaptation methods on TIDIGITS. (Recall that the accuracy of the corresponding baseline SI model is 96.25%. Since the performance of the SI model and EV adaptation are almost the same, they cannot be differentiated in the plots. Thus, we do not plot the SI performance in the figure.)

Experiment 4: Comparison with other adaptation methods

In this experiment, eKEV adaptation was compared with the standard EV adaptation and our previous KEV adaptation, as well as the conventional MAP and MLLR adaptation. For each adaptation method, we tried to find the best setup for the method so as to obtain its best results for comparison purpose. That means, for eKEV adaptation, five ML neighbors and seven kernel eigenvoices were employed; for EV and KEV adaptation, the best results were obtained with the optimal number of eigenvoices which were one and eight respectively; for MAP adaptation, the best results were achieved with the best scaling factors in the range of 1–30; for MLLR adaptation, only global MLLR was tried, and the better results from using either diagonal or full transformation matrices were used for comparison. Notice

that for MLLR adaptation, no efforts were made to interpolate the raw MLLR results with the SI model.

The results are plotted in Fig. 4. We have the following observations:

• eKEV adaptation outperforms all other methods in all three cases with different amount of adaptation data. It reduces the word error rate (WER) of the SI model by 37.0%, 40.5%, and 41.3% respectively with 2.1s, 4.1s, and 9.6s of adaptation speech.

• Among the three conventional adaptation methods, MAP adaptation gives the best performance when there are only 2.1s or 4.1s of adaptation speech. When there are about 10s of data, MLLR adaptation performs the best.

• It is surprising and disappointing that the standard EV adaptation only has comparable performance as the SI model's in this task[9].

• All the three EV-based methods saturate quickly: their adaptation performance only improves very slightly after 5s of adaptation speech.

• Both versions of kernelized EV adaptation, namely KEV or eKEV adaptation, outperform standard EV adaptation. The results suggest that nonlinear kernel PCA using composite kernels can be more effective in finding the eigenvoices.

• Although the robust versions of EV, KEV, and eKEV adaptation are tried, it is found that the weighting w_0 of the SI model always went to zero during robust eKEV adaptation; this does not happen in robust EV or KEV adaptation. One possible explanation is that the reference speakers in eKEV adaptation provide much stronger prior information for adaptation than the SI model; this is consistent with the motivation of RSW adaptation. (For the difference in performance between robust EV/KEV adaptation and their non-robust counterparts, please refer [1].).

• eKEV adaptation is consistently better than KEV adaptation by an average of (absolute) 0.33%. The two methods differ in how they evaluate the $Q(\mathbf{w})$ function that maximizes the likelihood of the adaptation speech. KEV adaptation maps the acoustic observations to the feature space to compute their likelihoods on an *implicit* adapted speaker model in the feature space, while eKEV adaptation maps the adapted model from the feature space back to the input space before computing acoustic observation likelihoods. Theoretically speaking, it is hard to tell which of the two adaptation methods should be better in terms of recognition performance. However, there may be three reasons for eKEV's better performance:

– Since there is no analytical solution for both KEV and eKEV adaptation, numerical methods are used to search for the optimal kernel eigenvoice weights, and there can be many local optima. The use of reference speakers seem to provide a guidance for a better local maximum solution than KEV adaptation.

– The use of Gaussian kernels requires that the kernel value in Eqn. (10) of KEV adaptation and that in Eqn. (18) of eKEV adaptation must be positive. Hence, the optimization of the eigenvoice weight vector $\mathbf{w}$ is subject to the constraint that these kernel values are strictly greater than zero. In our current KEV and eKEV implementation, we simply check that the constraint is not violated otherwise adaptation stops before meeting the convergence requirement. In our experience, the constraint was violated much more frequently in KEV adaptation than in eKEV adaptation[10]. We believe that the use of reference speakers in eKEV adaptation help confine the search space to stay in a feasible region. As a result, eKEV adaptation seems to converge to a better solution.

– In practice, since eKEV adaptation runs much faster than KEV adaptation (Experiment 3 above), we may run more gradient ascent iterations in eKEV adaptation than in KEV adaptation. For instance, we may set the maximum number of iterations to about 1000 in eKEV adaptation, but only about 100 iterations in KEV adaptation. Thus, KEV adaptation is more likely to stop without reaching the convergence requirement.

B. Evaluation on Large-vocabulary Continuous Speech Recognition (LVCSR)

In this section, we would like to check if eKEV adaptation is also effective on a relatively large-vocabulary recognition task using triphone HMMs with Gaussian-mixture states. The use of a large number of context-dependent models and multiple-Gaussian mixtures poses new challenges and some changes in the eKEV adaptation implementation are deemed necessary.

B.1 WSJ0 Corpus

The Wall Street Journal corpus WSJ0 [32] with 5K vocabulary was chosen. The standard SI-84 training set was used for training the speaker-independent (SI) model. It consists of 83 speakers and 7138 utterances for a total of about 14 hours of training speech (after discarding the problematic data from one speaker as in the Aurora4 corpus [33]). The standard nov'92 5K non-verbalized test set was used for evaluation. It consists of 8 speakers, each with about 40 utterances.

B.2 Acoustic Modeling

The traditional 39-dimensional MFCC vectors were extracted at every 10ms over a window of 25ms from the training and testing data. The speaker-independent (SI) model consists of 15,449 cross-word triphones based on 39 base phonemes. Each triphone was modeled as a continuous density HMM which is strictly left-to-right and has three states with a Gaussian mixture density of 16 components per state. State tying was performed to give 3131 tied states in the final SI model. In addition, the same type of "sil" and "sp" models were trained as in the last TIDIGITS experiments.

[9]The apparently poor performance of EV adaptation has been discussed thoroughly in [1].

[10]Actually, in the new implementation of eKEV adaptation used in the WSJ evaluation in Section V-B, by using BFGS plus line search, it is found that the constraint was never violated. However, for the TIDIGITS evaluation, we keep the old implementation which was closer to the implementation of KEV adaptation in [1] so that the two methods can be fairly compared.

Because of the large number of triphone models and Gaussians, there are not sufficient data to train a speaker-dependent (SD) model for each of the 83 training speakers. Instead, following the common practice of EV adaptation for LVCSR [10], we created the SD models by MLLR adaptation using a regression tree of 32 classes. Notice that the dimension of the training speaker supervectors in this WSJ0 evaluation is much higher than that in the TIDIGITS evaluation: $n_2 = 3131$ tied states $\times 16$ Gaussians/state $\times 39$/Gaussian $= 1,958,736$. One way to save models storage is to store only the MLLR transforms for each SD model, and the actual means are computed on-the-fly when needed.

B.3 Experiment: Comparison with other adaptation methods

eKEV adaptation was compared with EV, MAP, and MLLR adaptation on the WSJ0 corpus. KEV adaptation was not tried as the online kernel value computations now would involve speaker supervectors of over a million dimensions, and would run very slowly. Again efforts were made to find the best setup for each method as in the TIDIGITS evaluation. For the conventional EV adaptation, 10 eigenvoices were found giving good results; for MAP adaptation, the best results with a scaling factor in the range of 3–12 were reported.

For each of the 8 testing speakers, 1–3 utterances of his speech were randomly selected so that the amount of adaptation speech is about 4s or 8s (or, 5s and 10s respectively if one includes the silence portions), and his adapted model was tested on his remaining speech in the test set. This was repeated three times and the three adaptation results are averaged before they are reported. Finally, a bigram language model of perplexity 147 was employed in this recognition task.

To speed up the convergence of the gradient-based search in each M-step of the GEM procedure, the simple gradient-ascent algorithm was replaced by the quasi-Newton BFGS algorithm [34] plus line search. BFGS is similar to the traditional Newton's method and makes use of the Hessian to retrieve the Newton's direction. However, it approximates the Hessian with an estimate that can be derived solely from the gradient. As a result, it is more efficient and it can enforce the Hessian estimate to be strictly positive-definite. It was found that only about 10–20 BFGS iterations are now required.

Parameter initialization and settings

We used a simple adaptation task on the Resource Management [35] to help set the system parameters, and then they were applied to the WSJ0 task without modification. These parameter settings are listed below for readers' reference:

- $\beta_r = \beta = 0.005$ for $r = 1, \ldots, R$.
- The learning rate was initialized to 0.1, but it was subsequently changed during a heuristic line search procedure.
- The number of kernel eigenvoices was fixed to 7.
- The number of ML reference speakers was fixed to 5.
- The gradient ascent algorithm stopped when either the relative improvement on the likelihood of the adaptation data was less than 0.00015, or 30 iterations was reached.

TABLE II
PERFORMANCE OF MLLR, MAP, EV, AND EKEV ADAPTATION ON WSJ0.

Model/Adaptation Method	4s	8s
SI	92.26	92.26
MAP	92.48	92.47
MLLR	92.32	92.98
EV	92.46	92.51
eKEV	92.86	92.92

Results and Discussions

Table II summarizes the performance of the various adaptation methods. Below are some additional or different observations we have beyond those we have already made in the TIDIGITS evaluation:

- All the three conventional adaptation methods — EV, MAP, and MLLR — now give slight improvement over the SI model when 4s of adaptation data are available. With 8s of adapting speech, MLLR adaptation again outperforms the other two methods.
- While EV adaptation has no improvement in the TIDIGITS experiments, it now outperforms the SI model and is comparable with MAP adaptation.
- eKEV adaptation again outperforms all the other methods under comparison in the 4s case, and is comparable with MLLR adaptation in the 8s case. It reduces the WER of the SI model by 7.75% and 8.52% respectively with 4s and 8s of adaptation speech.

TABLE III
COMPARISON BETWEEN EKEV AND RSW USING DIFFERENT TYPES OF REFERENCE SPEAKERS.

Adaptation Method	Reference Speakers	4s	8s
SI	—	92.26	92.26
eKEV	ML	92.86	92.92
RSW	cluster	92.33	92.41
RSW	ML	92.89	92.83

C. Implication to Reference Speaker Weighting (RSW)

As we mentioned in the Introduction section that eKEV adaptation and RSW are similar in that both methods restrict a speaker-adapted model to lie in the span of a set of reference speakers. The two methods are also different in some details:

• The definition of the reference speakers are different. From the experiments in Section V-A, eKEV adaptation suggests to use maximum-likelihood (ML) reference speakers, but RSW uses speaker clusters defined by their speaking rates [4].
• eKEV adaptation further requires the adapted model to lie on the part of the reference speakers' span that is related to the eigenspace found by KEV adaptation in the kernel-induced feature space. The conjecture is that the constraint may provide some useful prior information in the spirit of the eigenvoice approach to improve the adaptation performance.

Two additional experiments were run on the WSJ0 task to investigate the adaptation performance of eKEV and RSW with regards to the above two differences. The experimental procedure is the same as in the last Section V-B. For eKEV adaptation, five ML reference speakers were employed. For RSW, the procedure described in [4] were implemented. However, we define the speaker-adapted model simply as a linear combination of M reference speakers $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$:

$$\mathbf{s}^{(rsw)} = \sum_{m=1}^{M} w_m \mathbf{x}_m \ . \tag{38}$$

In addition, no restriction is placed on the values of $w_m, m = 1, \ldots, M$.

RSW was tested with two different definitions of reference speakers:
• Clustered speaker groups as defined in [4]. Thus, six speaker clusters were hierarchically defined: first based on the gender and then their speaking rates; each cluster consists of roughly 14 training speakers.
• The exact ML speakers as used by eKEV adaptation.

The results are shown in Table III. It can be seen that the definition of reference speakers is essential to the performance of RSW and eKEV adaptation. The clustered speaker groups based on speaking rate give only small improvement. However, the use of ML reference speakers may boost the performance of RSW so that it is as good as that of eKEV adaptation.

VI. Conclusions

In this paper, we attempt to solve the efficiency problem of our previously proposed kernel eigenvoice (KEV) speaker adaptation method by embedding the kernel PCA procedure in the computation of the speaker-adapted (SA) model. Although both KEV and eKEV adaptation methods try to improve the standard EV adaptation by exploiting the nonlinearity in the speaker supervector space via kernel PCA, eKEV adaptation using embedded kernel PCA has the additional advantage of eliminating all kernel evaluations between the training speaker supervectors and the adaptation or testing observations. This is achieved by finding an approximate pre-image of the implicit SA model in the kernel-induced feature space so that, at the end, there is an explicit SA model in the input supervector space from which regular acoustic HMMs can be constructed. As a result, both eKEV adaptation and subsequent recognition using its SA model run much faster than those of KEV adaptation with no performance degradation. In terms of adaptation performance, eKEV adaptation also outperform EV, MAP, and MLLR adaptation when less than 10s of adaptation speech are available. For instance, with only 4s of adaptation data, eKEV adaptation reduces the WER of the SI model by 40.5% in our simple TIDIGITS task, and 7.75% in the more complex WSJ0 task.

The successful use of a set of carefully chosen reference speakers in our novel eKEV adaptation prompts us to re-visit the reference speaker weighting (RSW) technique. It turns out that our use of maximum-likelihood (ML) reference speakers can greatly boost the adaptation performance of RSW. At the end, by adopting the ML reference speakers, both eKEV and RSW adaptation have similar performance. It shows that local speaker information is of great importance to speaker adaptation. On the other hand, our experiments using the WSJ0 task does not support our conjecture about the possible advantage of the additional prior information provided by the kernel eigenspace; further investigations will be needed.

Acknowledgments

This research is partially supported by the Research Grants Council of the Hong Kong SAR under the grant numbers HKUST6195/02E, HKUST6201/02E, and CA02/03.EG04.

Appendix

I. Relation between Distance and Kernel Functions

Without loss of generality, the Euclidean distance $d(\mathbf{x}, \mathbf{y})$ between 2 vectors: $\mathbf{x}$ and $\mathbf{y}$ in the input space, can be expressed in terms of many common kernel functions. Let's rewrite the Euclidean distance in terms of inner products as follows:

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2 \quad = \quad \mathbf{x}'\mathbf{x} + \mathbf{y}'\mathbf{y} - 2\mathbf{x}'\mathbf{y} \ . \tag{39}$$

Case I: Linear Kernel. Let $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}'\mathbf{y}$, then

$$d(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{x}) + k(\mathbf{y}, \mathbf{y}) - 2k(\mathbf{x}, \mathbf{y}) \ . \tag{40}$$

Case II: Polynomial Kernel. Let $k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}'\mathbf{y})^n$, then

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= (k(\mathbf{x}, \mathbf{x})^{\frac{1}{n}} - 1) + (k(\mathbf{y}, \mathbf{y})^{\frac{1}{n}} - 1) - 2(k(\mathbf{x}, \mathbf{y})^{\frac{1}{n}} \\ &= k(\mathbf{x}, \mathbf{x})^{\frac{1}{n}} + k(\mathbf{y}, \mathbf{y})^{\frac{1}{n}} - 2k(\mathbf{x}, \mathbf{y})^{\frac{1}{n}} \ . \end{aligned} \tag{41}$$

References

[1] B. Mak, J. T. Kwok, and S. Ho, "Kernel eigenvoice speaker adaptation," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 984–992, September 2005.
[2] B. Mak, J. T. Kwok, and S. Ho, "A study of various composite kernels for kernel eigenvoice speaker adaptation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004, vol. I, pp. 325–328.

[3] T. Kosaka, S. Matsunaga, and S. Sagayama, "Speaker-independent speech recognition based on tree-structured speaker clustering," *Journal of Computer Speech and Language*, vol. 10, pp. 55–74, 1996.

[4] Tim J. Hazen, "A comparison of novel techniques for rapid speaker adaptation," *Speech Communications*, vol. 31, pp. 15–33, May 2000.

[5] J. L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, April 1994.

[6] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Journal of Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[7] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 695–707, Nov 2000.

[8] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[9] 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*, May 2001, vol. 1, pp. 373–376.

[10] H. Botterweck, "Very fast adaptation for large vocabulary continuous speech recognition using eigenvoices," in *Proceedings of the International Conference on Spoken Language Processing*, 2000, vol. 4, pp. 354–357.

[11] 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*, 2000, vol. 3, pp. 742–745.

[12] N. Wang, S. 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, pp. 345–348.

[13] D. K. Kim and N. S. Kim, "Bayesian speaker adaptation based on probabilistic principal component analysis," in *Proceedings of the International Conference on Spoken Language Processing*, 2000, pp. 734–737.

[14] E. Jon, D. K. Kim, and N. S. Kim, "EMAP-based speaker adaptation with robust correlation estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001, pp. 321–324.

[15] 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, vol. 1, pp. 353–356.

[16] P. Nguyen and C. Wellekens, "Maximum likelihood eigenspace and MLLR for speech recognition in noisy environments," in *Proceedings of the European Conference on Speech Communication and Technology*, 1999, pp. 2519–2522.

[17] M. F. J. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 417–428, July 2000.

[18] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[19] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

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

[21] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[22] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *Journal of Machine Learning Research*, vol. 2, pp. 125–137, 2001.

[23] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[24] S. Mika, B. Schölkopf, A. Smola, K.R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," in *Advances in Neural Information Processing Systems 11*, M.S. Kearns, S.A. Solla, and D.A. Cohn, Eds., San Mateo, CA, 1998, Morgan Kaufmann.

[25] J.T. Kwok and I.W. Tsang, "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, vol. 15, no. 6, pp. 1517–1525, Nov. 2004.

[26] G.H. Bakir, J. Weston, and B. Schölkopf, "Learning to find pre-images," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds., Cambridge, MA, 2004, MIT Press.

[27] J. T. Kwok, B. Mak, and S. Ho, "Eigenvoice speaker adaptation via composite kernel PCA," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. MIT Press, Cambridge, MA, 2004.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[29] B. Mak, S. Ho, and J. T. Kwok, "Speedup of kernel eigenvoice speaker adaptation by embedded kernel PCA," in *Proceedings of the International Conference on Spoken Language Processing*, Jeju Island, South Korea, October 14–18 2004, vol. IV, pp. 2913–2916.

[30] B. Mak and S. Ho, "Various reference speakers determination methods for embedded kernel eigenvoice speaker adaptation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, USA, March 18–23 2005, vol. 1, pp. 981–984.

[31] R. G. Leonard, "A database for speaker-independent digit recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1984, vol. 3, pp. 4211–4214.

[32] D. B. Paul and J. M. Baker, "The design of the Wall Street Journal-based CSR corpus," in *Proceedings of the DARPA Speech and Natural Language Workshop*, Feb. 1992.

[33] N. Parihar and J. Picone, "DSR front end LVCSR evaluation," *AU/384/02, Aurora Working Group*, Dec. 2002, (http://www.isip.msstate.edu/projects/aurora).

[34] J. Frédéric Bonnans, J. Charles Gilbert, Claude Lemaréchal, and Claudia A. Sagastizábal, *Numerical Optimization: Theoretical and Practical Aspects*, Springer-Verlag, Berlin Heidelberg, 2003.

[35] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett, "The DARPA 1000-word Resource Management database for continuous speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1988, vol. 1, pp. 651–654.