TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 5, October 2020, pp. 2488~2497
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i5.14655

Kernal based speaker specific feature extraction and its applications in iTaukei cross language speaker recognition

Satyanand Singh[1], Pragya Singh[2]
[1]School of Electrical and Electronics Engineering, Fiji National Universisty, Fiji
[2]School of Public Health and Primary Care, Fiji National Universisty, Fiji

Article Info

Article history:

Received Nov 21, 2019
Revised May 2, 2020
Accepted May 11, 2020

Keywords:

Automatic speaker recognition system
Kernel independent component analysis
Kernel linear discriminant analysis
Kernel principal component analysis
Principal component analysis

ABSTRACT

Extraction and classification algorithms based on kernel nonlinear features are popular in the new direction of research in machine learning. This research paper considers their practical application in the iTaukei automatic speaker recognition system (ASR) for cross-language speech recognition. Second, nonlinear speaker-specific extraction methods such as kernel principal component analysis (KPCA), kernel independent component analysis (KICA), and kernel linear discriminant analysis (KLDA) are summarized. The conversion effects on subsequent classifications were tested in conjunction with Gaussian mixture modeling (GMM) learning algorithms; in most cases, computations were found to have a beneficial effect on classification performance. Additionally, the best results were achieved by the Kernel linear discriminant analysis (KLDA) algorithm. The performance of the ASR system is evaluated for clear speech to a wide range of speech quality using ATR Japanese C language corpus and self-recorded iTaukei corpus. The ASR efficiency of KLDA, KICA, and KLDA technique for 6 sec of ATR Japanese C language corpus 99.7%, 99.6%, and 99.1% and equal error rate (EER) are 1.95%, 2.31%, and 3.41% respectively. The EER improvement of the KLDA technique-based ASR system compared with KICA and KPCA is 4.25% and 8.51% respectively.

Corresponding Author:

Satyanand Singh,
School of Electrical and Electronics Engineering,
Fiji National Universisty, Fiji.
Email: satyanand.singh@fnu.ac.fj

1. INTRODUCTION

ASR is implemented using very conventional statistical modeling techniques such as GMM or ANN modeling. But in the past few years, machine learning theory has evolved into a variety of new algorithms for learning and classification. The so-called kernel-based method, in particular, has recently become a promising new path to science. Kernel-based classification and regression techniques like the well-known SVM found a fairly slow expression. That may be because to address theoretical and practical problems it needs to be applied to large tasks such as speech recognition. Recently, however, more and more authors have been concerned about the use of support vector machines in speech recognition [1].

Besides using kernel-based classifiers, an alternate way is to use kernel-based technologies only to convert the feature space and leave the classification job to more conventional methods. The purpose of this paper is to study the applicability of some of these methods to classify phonemes, using kernel-based pre-learning speaker-specific feature extraction methods to improve ASR classification rates. This paper mainly discusses KPCA [2], KICA [3], KLDA.

Usually, a traditional ASR process consists of two phases: a training phase, and a test phase. In the training phase, the device extracts speaker-specific characteristics from the speech signal to be used to create a speaker model [1], where the aim of the test phase is to determine the speaking samples that fit the individual of the training sample. he original speech signal is transformed into a vector representation of the function [2] in all audio signal processing. Linear prediction cepstral coefficients (LPCC) and perceptual linearity predicted cepstrum coefficient (PLPCC), mal frequency cepstral coefficient (MFCC) [3] approach is most commonly used in the ASR system to obtain speaker-specific features. For modeling, discriminant classifiers in support vector machine (SVM) [4] representation have achieved impressive results in many ASR systems. SVM will definitely effectively train non-linear boundaries for decision-making by classifying interesting speakers/imposters as they are distinct.

Although these feature extraction techniques are effective, non-linear mapping of speech features to new suitable spaces may generate new features that can better identify speech categories. Kernel-based technology has been applied to a variety of learning machines, including support vector machines (SVM), Kernel discriminant analysis (KDA), kernel principal component analysis (KPCA) [5]. The latter two methods are widely used in image recognition. Their performance in speaker recognition, however, has not been carefully investigated.

The purpose of this paper is to examine the applicability of some of these methods to classify phonemes, using kernel-based feature extraction methods applied before learning to boost classification levels. Essentially, this paper deals with the strategies of KPCA, KICA [6, 7], KLDA [5], and Kernel springy discriminant analysis (KSDA) [8]. In this work, KPCA, KICA, and KLDA is used for speaker specific feature extraction with an ASR system. With KPCA, speaker-specific features can be expressed in a high dimension space which can possibly generate more distinguishable speaker features.

2. FUNDAMENTAL OF PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) is a very common method of dimensionality reduction and feature extraction. PCA attempts to find linear subspaces that are smaller in size than the original feature space, with new features having the largest variance [9]. Consider the data set $\{x_i\}$ where $i = 1, 2, 3, \ldots, N$, each x_i is a D-dimensional vector. Now we project the data into the M-dimensional subspace, here, $M < D$. The projection is represented as $y = Ax$, where $A = [u_1^T, u_2^T, \ldots, u_M^T], and\ u_k^T u_k = 1$ for $k = 1,2,3, \ldots, M$. We want to maximize the variance of $\{y_i\}$, which is the trace of the covariance matrix of $\{y_i\}$.

$$A^* = \arg \max_A tr(S_y) \quad (1)$$

where,

$$S_y = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})(y_i - \bar{y})^T \quad (2)$$

and

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} X_i \quad (3)$$

Covarience matrix of $\{X_i\}$ is the S_X, since $tr(S_y) = tr(AS_XA^T)$, by using the Lagrangian multiplier and taking the derivative, we get,

$$S_X u_k = \lambda_k u_k \quad (4)$$

which indicates u_k is the eigenvector of S_X and now X_i can be represented as follows;

$$X_i = \sum_{k=1}^{D}(X_i^T u_k)\, u_k \quad (5)$$

X_i can be approximated by $\tilde{X}_i$ and expressed as follows:

$$\tilde{X} = \sum_{k=1}^{D}(X_i^T u_k)\, u_k \quad (6)$$

where $u_k$is the eigenvector of $S_X$corresponding to the kth largest eigenvalue. Standard PCA results for the two-speaker's audio data shown in Figure 1 (a).

2.1. Kernel PCA methodology for dimensionality reduction in ASR system

Standard PCA only allows linear size reduction. However, standard PCA is not very useful when the data has a more complex structure that cannot be represented well in linear subspaces. Fortunately, the kernel PCA allows us to extend the standard PCA to nonlinear dimensionality reduction [10]. Assume that a set of observations is given $X_i \in \mathbb{R}^n, i = 1,2,3,\dots,m$. Consider the inner dot product space F associated with the input space by a map $\phi: \mathbb{R}^n \to F$ may be non-linear. The feature space F has an arbitrary size and in some cases has an infinite dimension. Here, uppercase letters used for elements of F, and lowercase letters are used for elements of $\mathbb{R}^n$. Suppose we are working on centered data $\sum_{i=1}^m \phi(X_i) = 0$. In F, the covariance matrix has the form as follows:

$$C = \frac{1}{m}\sum_{j=1}^m \phi(X_j)\phi(X_j)^T \tag{7}$$

Eigenvalues $\lambda \geq 0$ and nonzero eigenvectors $V \in F\backslash(0)$ satisfying $CV = \lambda V$. It is well known that all solutions V with $\lambda \neq 0$ are in the range of $\{\phi(X_i)\}_{i=1}^m$. This has two consequences. First, consider a set of equations $\langle \phi(X_k), CV\rangle = \lambda\langle \phi(X_k), V\rangle$, for all $k = 1,2,3,\dots,m$ and second there exist coefficients $\alpha_i, i = 1,2,3,\dots,m$ in such a way that $V = \sum_{i=1}^m \alpha_i \phi(X_i)$. Combining $\langle \phi(X_k), CV\rangle = \lambda\langle \phi(X_k), V\rangle$ and $V = \sum_{i=1}^m \alpha_i \phi(X_i)$ we get the dual representation of the eigenvalue problem as $\frac{1}{m}\sum_{i=1}^m \alpha_i\langle \phi(X_k), \sum_{j=1}^m \phi(X_j)\langle \phi(X_j), \phi(X_i)\rangle\rangle = \lambda\sum_{i=1}^m \alpha_i\langle \phi(X_k), \phi(X_i)\rangle$ for all $k = 1,2,3,\dots,m$. We are defining a mXm matrix by $K_{ij} = \phi(X_i), \phi(X_j)$, this makes $K^2\alpha = m\lambda K\alpha$. Where α denoted as a column vectors with $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_m$ entries.

Let $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_m$ be the eigenvalue of K, $\alpha^1, \alpha^2, \dots, \alpha^m$ be the set of corresponding eigenvectors, and λ_r be the last non-zero eigenvalue. Normalizing $\alpha^1, \alpha^2, \dots, \alpha^r$ by needing the corresponding vectors in F be normalized $\langle V^k, V^k\rangle = 1$, for all $k = 1,2,\dots,r$. Considering $V = \sum_{i=1}^m \alpha_i \phi(X_i)$ and $K\alpha = m\lambda\alpha$, the normalization condition of $\alpha^1, \alpha^2, \dots, \alpha^r$ can be rewritten as follows;

$$1 = \sum_{i,j}^m \alpha_i^k \alpha_j^k \langle \phi(x_i), \phi(x_j)\rangle = \sum_{i,j}^m \alpha_i^k \alpha_j^k K_{i,j} = \langle \alpha^k, K\alpha^k\rangle = \lambda_k\langle \alpha^k, \alpha^k\rangle \tag{8}$$

for the purpose of principal component extraction, we need to compute the projections onto the eigenvectors V^k in F, for $k = 1,2,\dots,r$. Let y be the test point, with an image $\phi(y)$ in F.

$$\langle V^k, \phi(y)\rangle = \sum_{i=1}^m \alpha_i^k \langle \phi(x_i), \phi(y)\rangle \tag{9}$$

$\langle V^k, \phi(y)\rangle$ nonlinear principal component corresponding to ϕ.

2.2. Computation of covariance matrix and dot product matrix by positioning on feature space

For the sake of simplicity, we assume that the observations are at the center. This is easy to implement in the input space because it is not possible to explicitly calculate the average of the observations mapped with F, but it is more difficult to use F. Assume that any ϕ and any series of observations $X_1, X_2, \dots, X_m$ are given then let us define $\bar{\phi} = \frac{1}{m}\sum_{i=1}^m \phi(X_i)$ and then the point $\tilde{\phi}(X_i) = \phi(X_i) - \bar{\phi}$ will be centered. Therefore, the above assumption holds, we defin the covariance matrix and the dot product matrix $\tilde{K}_{ij} = \langle \tilde{\phi}(X_i), \tilde{\phi}(X_j)\rangle$ in F. We known eigenvalue problems as $m\tilde{\lambda}\tilde{\alpha} = \tilde{K}\tilde{\alpha}$ with $\tilde{\alpha}$ is the expansion coefficient of the eigenvector relative to the center point $\tilde{\phi}(X_i)$. Since there is no central data, $\tilde{K}$ cannot be explicitly calculated, but it can be represented by a corresponding K without a center therefore $\tilde{K}_{ij} = \langle \tilde{\phi}(X_i) - \phi, \phi(X_j) - \phi\rangle$ $= K_{i,j} - \frac{1}{m}\sum_{t=1}^m k_{it} - \frac{1}{m}\sum_{s=1}^m k_{sj} + \frac{1}{m^2}\sum_{s,t=1}^m k_{st}$. We can get more compact expression by using the vector $1_m = (1, \dots, 1)^T$. The compact expression is $\tilde{K} = K - \frac{1}{m}K1_m 1_m^T - \frac{1}{m}1_m 1_m^T K + \frac{1}{m^2}(1_m^T K 1_m)1_m^T K 1_m$. We can calculate $\tilde{K}$ from K and solve the eigenvalue problem. Consider test point Y projection of the center point of the center ϕ-image of Y to the feature vector of the covariance matrix is computed to find its coordinates [11].

$$\langle \tilde{\phi}(Y), \tilde{V}^k\rangle = \langle \phi(Y) - \vec{\phi}, \tilde{V}^k\rangle = \sum_{i=1}^m \tilde{\alpha}_i^k \langle \phi(Y) - \vec{\phi}, \phi(X_i) - \vec{\phi}\rangle$$

$$= \sum_{i=1}^m \tilde{\alpha}_i^k \left\{K(Y, X_i) - \frac{1}{m}\sum_{s=1}^m K(X_s, X_i) - \frac{1}{m}\sum_{s=1}^m K(Y, X_s) + \frac{1}{m^2}\sum_{s,t=1}^m K(X_s, X_t)\right\} \tag{10}$$

Introducing the vector Z.

$$Z = \left(K(Y, X_i)\right)_{mX1} \tag{11}$$

$$\left(\langle\tilde{\phi}(Y), \tilde{V}^k\rangle\right)_{1Xr} = Z^T\tilde{V} - \frac{1}{m}1_m^T K\tilde{V} - \frac{1}{m}(Z^T 1_m)1_m^T\tilde{V} + \frac{1}{m^2}(1_m^T K 1_m)1_m^T\tilde{V}$$

$$= Z^T\left(I_m - \frac{1}{m}1_m 1_m^T\right)\tilde{V} - \frac{1}{m}1_m^T K\left(I_m - \frac{1}{m}1_m 1_m^T\right)\tilde{V}$$

$$= \left(Z^T - \frac{1}{m}1_m^T K\right)\left(I_m - \frac{1}{m}1_m 1_m^T\right)\tilde{V} \tag{12}$$

Note that KPCA implicitly uses only input variables because the algorithm uses kernel function evaluation to represent the reduction in feature space dimensions. Therefore, KPCA is useful for nonlinear feature extraction by reducing the size; it does not explain the characteristics of the input variable selection.

3. KERNEL- BASED SPEAKER SPECIFIC FEATURE EXTRACTION AND ITS APPLICATION IN ASR

Classification algorithms must represent the objects to be classified as points in a multidimensional feature space. However, one can apply other vector space transformations to the initial features before running the learning algorithm. There are two reasons for doing this. First, they can improve the performance of classification and second, they can reduce the data's dimensionality. The selection of initial features and their transformation are sometimes dealt with in the literature under the title "feature extraction". To avoid misunderstanding, this section describes only the latter and describes the first feature set. Hopefully it will be more effective and classification will be faster. The approach to the extraction of features may be either linear or nonlinear, but there is a technique that breaks down the barrier between the two forms in some way. The key idea behind the kernel technique was originally presented in [12] and applied again in connection with the general purpose SVM [13-15] followed by other kernel-based methods.

3.1. Supplying input variable information into kernel PCA

Additional information to the KPCA representation for interpretability. We have developed a process to project a given input variable into a subspace spanned by feature vectors $\tilde{V} = \sum_{i=1}^{m}\tilde{\alpha}\tilde{\phi}(X_1)$. We can think of our observation as a random vector $X = (X_1, X_2, \ldots\ldots, X_n\ )$ implementation then to represent the prominence of the input variable X_k in the KPCA. Considering a set of points of mathematical forms $y = a + se_k \in \mathbb{R}^n$ where $e_k = (0, \ldots\ldots, 1, \ldots\ldots, 0)$ of kth component is either 0 or 1. Next, the projection points $\phi(y)$ of these images onto the subspace spanned by the feature vector $\tilde{V} = \sum_{i=1}^{m}\tilde{\alpha}\tilde{\phi}(X_1)$ can be calculated. Considering in (12) the row vector gives the induction curve in the Eigen space expressed in matrix form:

$$\sigma(s)_{1Xr} = \left(Z_s^T - \frac{1}{m}1_m^T K\right)\left(I_m - \frac{1}{m}1_m 1_m^T\right)\tilde{V} \tag{13}$$

Furthermore, by projecting the tangent vector to s = 0, we can express the maximum change direction of $\sigma(s)$ associated with the variable X_k. Matrix form of the expression represented as follows:

$$\left.\frac{d\sigma}{ds}\right|_{s=0} = \left.\frac{dZ_s^T}{ds}\right|_{s=0}\left(I_m - \frac{1}{m}1_m 1_m^T\right)\tilde{V} \tag{14}$$

where

$$\left.\frac{dZ_s^T}{ds}\right|_{s=0} = \left(\left.\frac{dZ_s^1}{ds}\right|_{s=0}, \ldots\ldots\ldots, \left.\frac{dZ_s^m}{ds}\right|_{s=0}\right)^T$$

and

$$\left.\frac{dZ_s^i}{ds}\right|_{s=0} = \left.\frac{dK(Y,X_i)}{ds}\right|_{s=0} = \left(\sum_{t=1}^{m}\frac{\partial K(Y,X_i)}{\partial Y_t}\frac{dY_t}{ds}\right)\Big|_{s=0} = \sum_{t=1}^{m}\left.\frac{\partial K(Y,X_i)}{\partial Y_t}\right|_{Y=a}\delta_t^k = \left.\frac{\partial K(Y,X_i)}{\partial Y_k}\right|_{Y=a}$$

where delta of Kronecker is represented as δ_t^k and radial basis kernel as $k(Y, X_i) = exp(-c\|Y - X_i\|^2) = exp(-c\sum_{t=1}^{n}(Y_i - X_{it})^2)$. After considering $y = a + se_k \in \mathbb{R}^n$:

$$\left.\frac{dZ_s^i}{ds}\right|_{s=0} = \left.\frac{\partial K(Y,X_i)}{\partial Y_k}\right|_{y=a} = -2cK(a, X_i)(a_k - X_{ik}) = -2cK\left(X_\beta, X_i\right)\left(X_{\beta k} - X_{ik}\right)$$

where the training point $a = X_\beta$. Thus, by applying (13), it is possible to locally represent any given input variable plot in KPCA. Furthermore, by using (14), it is possible to represent the tangent vector associated with any given input variable at each sample point [16]. Therefore, a vector field can be drawn on KPCA indicating the growth direction of a given variable.

There are some existing techniques to compute z for specific kernels [17]. For a Gaussian kernel $(X,Y)) = exp(-\|X-Y\|^2/2\sigma^2)$, z must satisfy the following condition;

$$Z = \frac{\sum_{i=1}^{m} \gamma_i(\|Z-X_i\|^2/2\sigma^2)X_i}{\sum_{i=1}^{m} \gamma_i(-\|Z-X_i\|^2)/2\sigma^2} \quad (15)$$

Kernel PCA results for the two- speaker's audio data with is shown in Figure 1 (b).

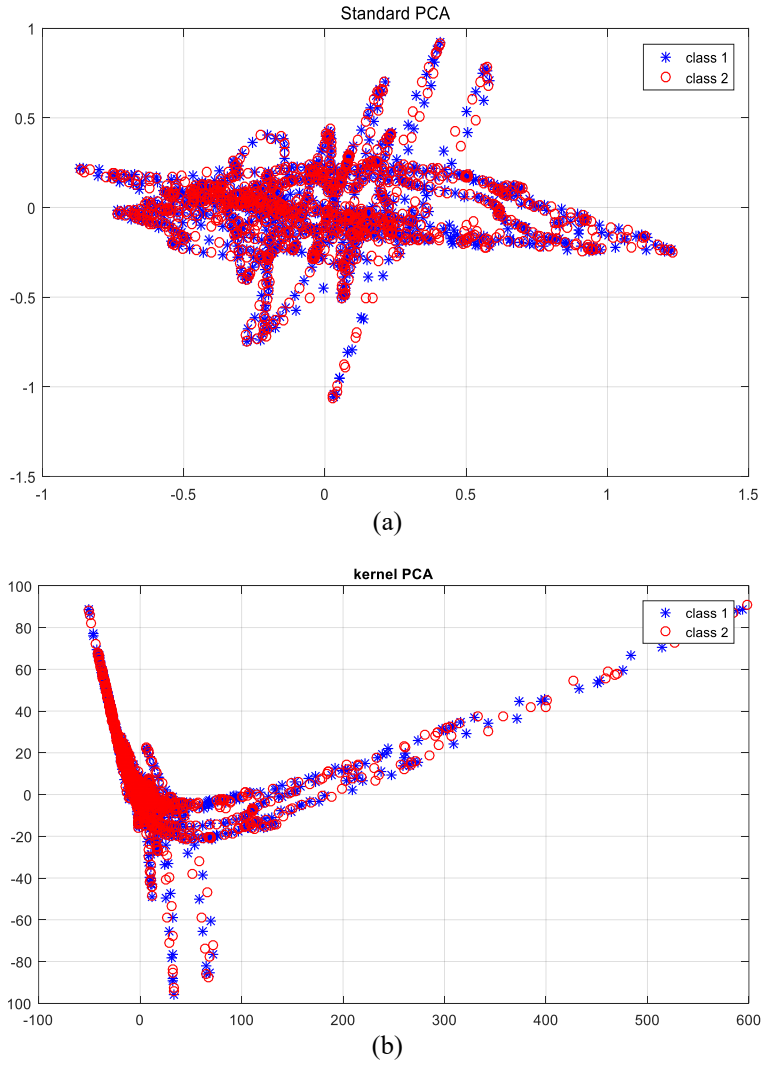


Figure 1. Standard PCA and Kernel PCA results for the two-speaker's audio data;
(a) standard PCA and (b) kernel PCA

3.2. Application of kernel independent component analysis (KICA)

Independent component analysis is a general statistical approach originally born from the study of separation from blind sources. Another application of ICA is the unsupervised extraction of features. This is intended to transform input data linearly into uncorrelated elements, using at least a distribution of the Gaussian sample set [18]. The explanation for this is that classification of data in certain directions would be simpler. This is in accordance with the most popular speech modeling technique, i.e. fitting Gaussian mixtures on each

class. This obviously means that Gaussian mixtures can approximate the distributions of the groups KICA extends this by assuming, on the contrary, that when all classes are fused, the distribution is not Gaussian; thus, using non-Gaussianism as a heuristic for the uncontrolled extraction of features would prefer those directions which separate classes.

Several objective functions for optimal selection of independent directions were described using approximately equivalent approaches. The KICA algorithm's goal itself is to find such objective functions as optimally as possible [19]. For KICA output most iterative methods are available. Others need to be preprocessed, i.e. focused and whitened while others do not. Overall, experience shows that all of these algorithms can converge faster with oriented and whitewashed data, even those that don't really need it [20].

Let's first investigate how the centering and whitening pre-processing steps can be done in the kernel function space. To this end, allow the kernel function $\varkappa$ in $\mathcal{F}$ to implicitly define the inner product with the associated transformation ϕ. Step one Centering $\mathcal{F}$ - Shifting the data $\phi(X_1), \phi(X_2), \ldots\ldots, \phi(X_k)$ along with its mean $E\{\phi(X)\}$ to get the data as follows:

$$\begin{cases} \phi'(X_1) = \phi(X_1) - E\{\phi(X)\} \\ \phi'(X_2) = \phi(X_2) - E\{\phi(X)\} \\ \qquad\qquad . \\ \qquad\qquad . \\ \phi'(X_k) = \phi(X_k) - E\{\phi(X)\} \end{cases} \tag{16}$$

Step two Whitening in $\mathcal{F}$. Transfoprming the centered samples $\phi'(X_1), \phi'(X_2), \ldots\ldots, \phi'(X_k)$ via an orthogonal transformation Q into its vectors $\hat{\phi}(X_1) = Q\phi'(X_1), Q\phi'(X_2), \ldots, Q\phi'(X_k) = Q\phi(X_k')$. $\hat{C} =$ is the covariance matrix. Because standard PCA converts the covariance matrix into a diagonal form just like its kernel based equivalent, where the diagonal elements are the unique values of the data covariance matrix $E\{\hat{\phi}(X)\hat{\phi}(X)^T\}$, all that remains is to transform the diagonal element into 1. Based on this finding, a slight modification of the formulas provided in the KPCA section will obtain the necessary whitening transformation [21]. Here $(\alpha_1\lambda_1), (\alpha_2\lambda_2), \ldots\ldots, (\alpha_k\lambda_k)$ and $\lambda_1 \geq \lambda_2 \geq \lambda_k$ are the eighpairs of $E\{\hat{\phi}(X)\hat{\phi}(X)^T\}$ then the transformation matrix Q will take a form $\left[\lambda_1^{-\frac{1}{2}}\alpha_1, \lambda_2^{-\frac{1}{2}}\alpha_2, \ldots\ldots \lambda_m^{-\frac{1}{2}}\alpha_m\right]^T$. Kernel Independent component analysis results for the two-speaker's audio data is shown in Figure 2 (a).

3.3. Application of kernel linear discriminant analysis (KLDA)

LDA is a conventional, supervised method of extracting speaker-specific characteristics [19] that has proven to be one of the most effective pre-processing classification techniques. It has also long been used in speech recognition [22]. The main goal of LDA is to find a new orthogonal data set to provide the optimal class separation.

In KLDA we are essentially following the discussion of its linear counterpart, except in this case this is intended to happen implicitly in the kernel feature space F. Let's say again that a kernel function with a feature map and a kernel field space has been chosen. In order to define the transformation matrix A of KLDA, we define the objective function first as $\Gamma : \mathcal{F} \to \mathcal{R}$, because of the supervised nature of this method, it depends not only on the test data X but also on the indicator $\mathcal{L}$. Let's describe ubiquitous Γ(V).

$$\Gamma(\mathrm{V}) = \frac{V^T \mathcal{B} V}{V^T \mathcal{W} V}, \quad V \in: \mathcal{F}\backslash\{0\} \tag{17}$$

where $\mathcal{B}$ is the scatter matrix of the interclass, while $\mathcal{W}$ is the scatter matrix of the interclass. Here, the scatter matrix $\mathcal{B}$ between classes shows the scatter of the mean vectors μ_j around the overall mean vector μ.

$$\mathcal{B} = \sum_{j=1}^{r} \frac{k_j}{k}\left(\mu_j - \mu\right)\left(\mu_j - \mu\right)^T \; ; \; \mu = \frac{1}{k}\sum_{i=k}^{k} \phi(x_i) \; ; \quad \mu_j = \frac{1}{k_j}\sum_{\mathcal{L}(i)} \phi(x_i) \tag{18}$$

with the class label J, the in-class scatter matrix $\mathcal{W}$ represents the weighted average scatter of the sample vector covariance matrices C_j.

$$\mathcal{W} = \sum_{j=1}^{r} \frac{k_j}{k} C_j \; ; \;\; C_j = \frac{1}{k_j}\sum_{\mathcal{L}(i)=j}\left(\phi(x_i) - \mu_j\right)\left(\phi\left(x_i - \mu_j\right)\right)^T \tag{19}$$

Kernel linear discriminant analysis results for the two-speaker's audio data is shown in Figure 2 (b).

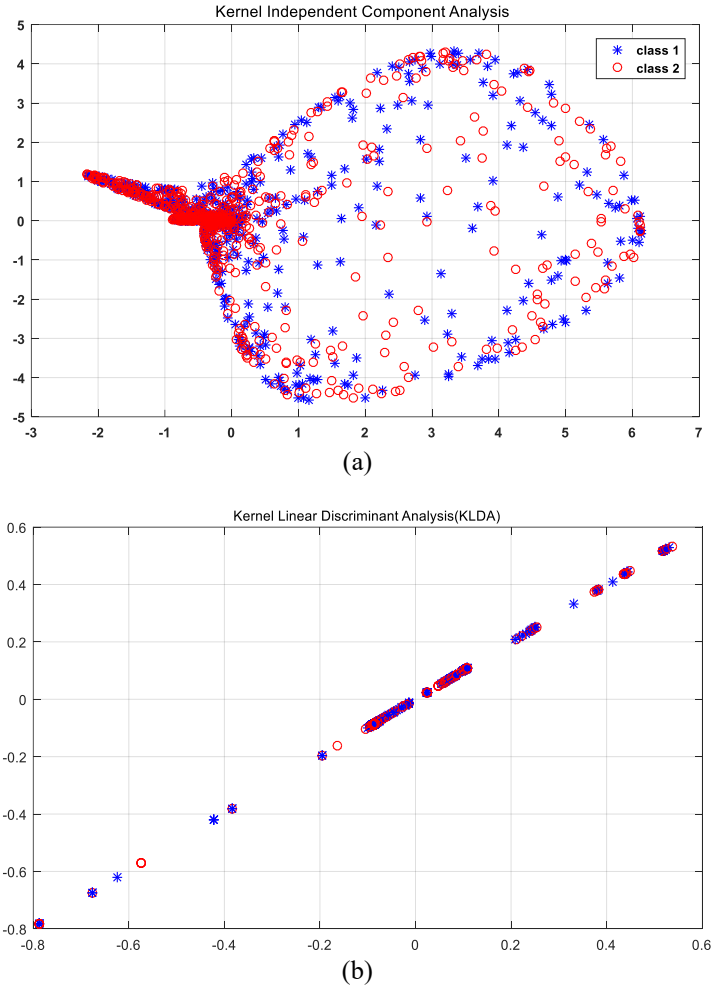


Figure 2. Kernel independent component analysis and kernel linear discriminant analysis results for the two-speaker's audio data; (a) kernel independent component analysis and (b) kernel lnear discriminant analysis

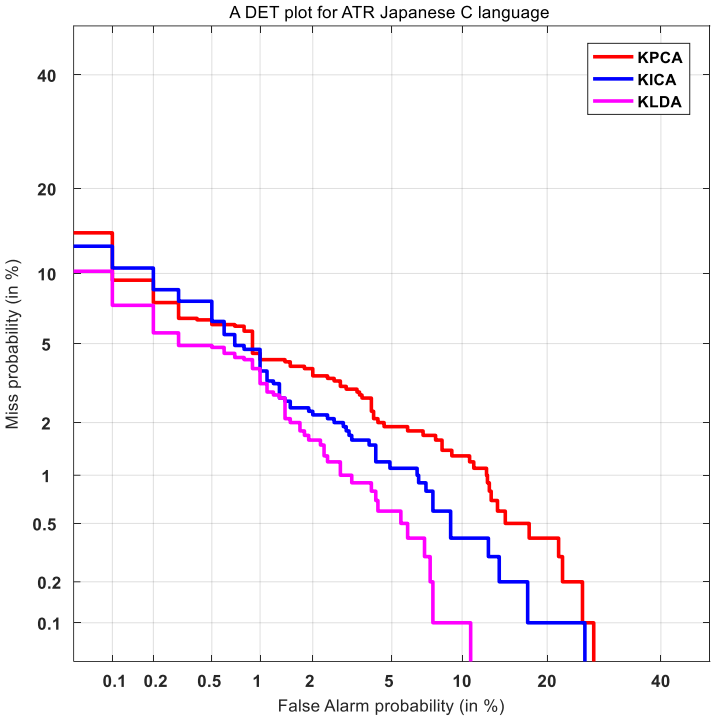
4. EXPERIMENTAL SETUP

To evaluate the efficiency of kernel-based speaker-specific feature extraction techniques, an isolated word recognition experiment was performed. The experiment includes 520 Japanese words from the ATR Japanese C language set Voice database, 80 speakers (40 men and 40 Female). Audio samples of 10 iTaukei speakers were collected at random and under unfavourable conditions. The average duration of the training samples was six seconds per speaker for all 10 speakers and out of twenty utterances of each speaker just one was used for training purpose [23-27]. For matching purposes the remaining 19 voice samples were used from the corpus. We have recorded utterances for this investigation were at one sitting for each speaker. The text for the utterances was randomly selected by speaker. The main voice recordings consist of both male and female speakers of twenty utterance of each using sampling rate of 16 kHz with 16 bits/sample.

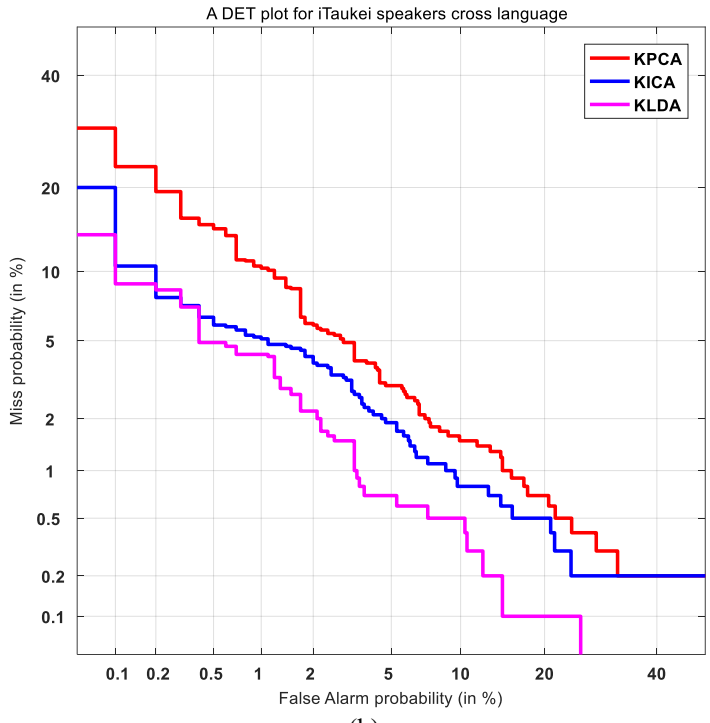
Throughout the experiment, 10400 utterances were used as training data and the remaining 31,200 utterances were used as test data. The sampling rate of the audio signal is 10 kHz. 12 Mel-Cepstral coefficients extracted using 25.6 ms Hamming windows with 10 ms shifts [28-32]. The features of KPCA were extracted from 13 Mel-cepstral coefficients including zero coefficients corresponding to 39 vector coefficients and their increment and acceleration coefficients. Around 1,000,000 frames were used as training data in this experiment, and it is computationally impossible to calculate matrix K with this amount of data. N frames are randomly picked from the training data to reduce the number of frames. The number N= 1024 was chosen to make the system computationally feasible.

Table 1 represents efficiency and EER of the ASR system for KLDA, KICA and KLDA respectively for ATR Japanese C language. Table 2 represents efficiency and EER of the ASR system for KLDA, KICA and KLDA respectively for10 iTaukei speakers cross language. Figure 3 show the equal error rate (EER) of

KLDA KICA, and KPCA based modeling technique. The ASR efficiency of of KLDA KICA, and KPCA based modeling technique are 99.9%, 99.6%, and 98.1% and EER are 4.7%, 4.9% and 5.1% respectively for 6 sec of audio signal. The EER improvement of KLDA technique based ASR system compared with KICA and KPCA is 4.25% and 8.51% respectively.



(a)



(b)

Figure 3. EER of KLDA, KICA and KLDA technique for 6 sec of voice data; (a) ATR Japanese C language and (b) iTaukei speaker's cross language

Table 1. Efficiency and EER of the ASR system for KLDA, KICA and KLDA respectively for ATR Japanese C language

	KLDA		KICA		KPCA	
	Efficiency in %	EER in %	Efficiency in %	EER in %	Efficiency in %	EER in %
6 sec	99.7	1.95	99.6	2.31	99.1	3.41
4 sec	99.5	2.29	99.1	3.20	98.2	4.11
2 sec	98.8	3.23	98.3	4.32	97.6	5.3

Table 2. Efficiency and EER of the ASR system for KLDA, KICA and KLDA respectively for10 iTaukei speakers cross language

	KLDA		KICA		KPCA	
	Efficiency in %	EER in %	Efficiency in %	EER in %	Efficiency in %	EER in %
6 sec	94.9	2.04	94.6	3.4	94.1	4.1
4 sec	94.3	2.34	94.1	3.7	93.5	4.8
2 sec	93.5	3.2.0	93.1	4.1	92.6	5.3

5. CONCLUSION

An experimental evaluation of the performance of the ASR system has been done on 6 sec of voice data of ATR Japanese C language. For the 10400, voice samples of the ATR Japanese C language speaker recognition accuracy 99.7%, 99.6%, and 99.1% and equal error rate (EER) is 1.95%, 2.31%, and 3.41% respectively for KLDA, KICA, and KPCA. The EER improvement of the KLDA technique-based ASR system compared with KICA and KPCA is 4.25% and 8.51% respectively. We find that non-linear transformations usually lead to better classification than non-linear transformations, and are therefore a promising new research direction. We also found that the supervised transformations are usually stronger than those not supervised. We think it would be worth searching for other supervised approaches which could be built similarly to the KLDA or KICA-based ASR application methodology. Such transformations significantly improved the phonological knowledge ASR training framework by providing a comprehensive and accurate classification of speaking contextual features unique to real-time speakers.

REFERENCES

[1] S. Singh, "Support Vector Machine Based Approaches For Real Time Automatic Speaker Recognition System," *International Journal of Applied Engineering Research*, pp. 8561-8567, 2018.

[2] B. Schölkopf, A. J. Smola, and K. R. Muller et al., "Kernel Principal Component Analysis," *Advances in Kernel Methods: Support Vector Learning*, pp. 327–352, 1999.

[3] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research,* vol. 3, no. 2002, pp. 1-48, 2002.

[4] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput*., vol. 12, no. 10, pp. 2385–2404, 2000.

[5] A. Kocsor and K. Kovács et al., "Kernel springy discriminant anlysis and its application to a phonological awareness teaching system," *International Conference on Text, Speech and Dialogue*, vol. 2448, pp. 325-328, 2002.

[6] F. R. Bach and M. I. Jordan, "Kernel independent component analysis*," J. Machine Learning Res.*, vol. 3, pp. 1–48, 2002.

[7] A. Kocsor and J. Csirik, "Fast independent component analysis in kernel feature spaces," *SOFSEM 2001: Theory and Practice of Informatics, 28th Conference on Current Trends in Theory and Practice of Informatics Piestany*, vol. 2234, pp. 271–281, 2001.

[8] A. Kocsor and K. Kovács et al., "Kernel springy discriminant anlysis and its application to a phonological awareness teaching system," *International Conference on Text, Speech and Dialogue*, vol. 2448, pp. 325–328, 2002.

[9] D. Lay, "Linear Algebra and its applications," 4th ed., Pearson, 2012

[10] B. Scholkopf, A. J. Smola, K.R. Muller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation,* vol. 10, pp. 1299-1319, 1998.

[11] Weinberger, Kilian Q., Packer, Benjamin D., and Saul, Lawrence K. "Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization," *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pp. 381-388, 2005.

[12] Gerazov, B.; Ivanovski, Z. "Kernel Power Flow Orientation Coefficients for Noise-Robust Speech Recognition," *IEEE/ACM Trans. Audio Speech Lang. Process*, vol. 23, pp.407-419, 2015.

[13] J. Geiger, B. Schuller, G. Rigoll, "Large-scale audio feature extraction and SVM for acoustic scene classification," *Proceedings of the 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA),* New Paltz, NY, USA, 20–23 October, pp. 1-4, 2013.

[14] A. Rabaoui, M. Davy, S. Rossignaol, N. Ellouze, "Using One-Class SVMs and Wavelets for Audio Surveillance," *IEEE Trans. Inf. Forensics Secur*, vol. 3, 763-775, 2018.
[15] H. Jiang, J. Bai, S. Zhang, B. Xu, "SVM-based audio scene classification," *Proceedings of the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering*, Wuhan, China, pp. 131-136, 2005.
[16] S. Mika, B. Scholkopf, A. J. Smola, K. R. Muller, M. Scholz, and G. Ratsch, "Kernel PCA and de–noising in feature spaces," in M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 536–542. MIT Press, 1999.
[17] K. I. Kim, K. Jung, H. J. Kim, "Face Recognition Using Kernel Principal Component Analysis," *IEEE Signal Processing Letters*, vol. 9, no. 2, pp. 40-42, February 2002).
[18] L. B. Almeida, "MISEP - linear and nonlinear ICA based on mutual information," *Journal of Machine Learning Research,* vol. 4, no. 2002, pp.1297-1318, 2003.
[19] K. Zhang, L. Chan, "Nonlinear independent component analysis with minimum nonlinear distortion," *ICML 2007, Corvallis*, OR, US, pp. 1127-1134, 2007.
[20] M. S. Bartlett, J. R. Movellan, T. J. Sejnowski, "Face Recognition by Independent Component Analysis," *IEEE Transactions on Neural Networks,* vol. 13, no. 6, pp. 1450-1464, 2002.
[21] F. R. Bach, M. I. Jordan, "Kernel Independent Component Analysis," *J. Machine Learning Res,* vol. 3, no. 2002, pp.1-48, 2002.
[22] O. Siohan, "On the robustness of linear discriminant analysis as a preprocessing step for noisy speech recognition," *Proc. ICASSP*, Detroit, MI, pp. 125-128, 1995.
[23] S. Singh, EG Rajan," Application of different filters in Mel frequency cepstral coefficients feature extraction and fuzzy vector quantization approach in speaker recognition," *International Journal of Engineering Research & Technology*, vol. 2, no. 6, pp. 419-425, 2013.
[24] G. S. Morrison, and F. Kelly, "A statistical procedure to adjust for time-interval mismatch in forensic voice comparison" *Speech Communication*, vol. 112, pp. 15- 21, 2019.
[25] S. Singh and Ajeet Singh "Accuracy Comparison using Different Modeling Techniques under Limited Speech Data of Speaker Recognition Systems," *Global Journal of Science Frontier Research: F Mathematics and Decision Sciences*, vol. 16, no. 2, pp. 1-17, 2016.
[26] P. Boersma and D. Weenink, "Praat: doing phonetics by computer", version 6.0.37, 2020. Available: http://www.praat.org/, 2020
[27] S. Singh. "The Role of Speech Technology in Biometrics, Forensics and Man-Machine Interface" *International Journal of Electrical and Computer Engineering,* vol. 9, no. 1, pp. 281-288, 2019.
[28] Chen-Chou Lo, Szu-Wei Fu, Wen-Chin Huang, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min Wang, "MOSNet: Deep Learning based Objective Assessment for Voice Conversion," *Interspeech ISCA*, pp. 1542–1545, 2019.
[29] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *ICASSP 2018*, 2018.
[30] Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and Alexandr Kozlov, "STC Antispoofing Systems for the ASVspoof2019 Challenge," *Proc. Interspeech 2019*, pp. 1033-1037, 2019.
[31] S. Singh and Mansour H. Assaf, "A Perfect Balance of Sparsity and Acoustic hole in Speech Signal and Its Application in Speaker Recognition System," *Middle-East Journal of Scientific Research*, vol. 24, no.11, pp. 3527-3541, 2016.
[32] A. Alexander, O. Forth, A. A. Atreya, F. Kelly, "VOCALISE: A Forensic Automatic Speaker Recognition System Supporting Spectral, Phonetic, and User-Provided Features," Research and Development Oxford Wave Research Ltd, United Kingdom, 2016.