Speaker verification normalization sequence kernel based on Gaussian mixture model super-vector and Bhattacharyya distance

YuJuan Xing, Ping Tan and Xin Wang

Abstract
Due to the accessibility and economy of human speech, speaker verification has become the research hotspot in the field of biometric authentication. A novel normalization sequence kernel based on Bhattacharyya distance clustering and within class covariance normalization was proposed in this paper. In this kernel, the high computation complexity and channel interference susceptibility that commonly exist in speaker verification could be restrained. In our method, we calculated the Bhattacharyya distance between pairs of Gaussian mixture models first. And then, a clustering algorithm was designed according to Gaussian mixture model's Bhattacharyya distance to obtain clustering center models. Maximum a posteriori was applied on these clustering center models to generate super-vectors immediately following. The sequence kernel was generated based on Bhattacharyya distance transformation and super-vectors. Finally, within class covariance normalization was utilized to restrain the channel distortion in kernel space. We adopted support vector machine as classifier to decide the target speaker. The experiment results on TIMIT corpus and NIST 2008 SRE showed that our proposed kernel has superior recognition accuracy and better robustness.

Keywords
Speaker verification, Gaussian mixture model super-vector, Bhattacharyya distance, within class covariance normalization, sequence kernel, support vector machine

Introduction
Speaker recognition has played an important role in biometric authentication field, which includes speaker identification and speaker verification. It has broad application prospect in the fields of military, E-bank, information security, and so on. Speaker verification is usually formulated as a hypothesis test that verifies an identity claim by estimating the similarity of the claimant’s speech and the enrolled utterance(s). The effective feature extraction and high-efficiency recognition model design have important effect on speaker verification system performance.

The speaker personality characteristics are largely reflected in the speaker's pronunciation channel variation, that is the channel frequency characteristics. Features that can characterize speaker personality are short-term energy, short-term average amplitude, short-term zero-crossing rate, short-time pitch period, pitch frequency, linear prediction coefficient, line-spectrum pair features, short-term spectrum, formant frequency and bandwidth, cepstrum features, Mel frequency cepstral coefficient (MFCC), and so on. MFCC can reflect the spectrum amplitude of speech and describe sound channel more accurately. And it is also easy to calculate. Therefore, it is more
widely and effectively applied in the field of speaker recognition. In this paper, we adopt MFCC as the speech parameters.

In text-independent speaker verification, support vector machine (SVM) has been proven to be effective classifier and most popularly used for many years. It has many desirable properties inherently, including the ability to classify patterns with least expected risk principle, to classify sparse data without over-training problem, and to make non-linear decisions via kernel function. Sloin and Burshtein presented a discriminative training algorithm based on SVM to improve the classification of hidden Markov models. Good experimental results had been obtained. Chu and Wang used SVM for cancer classification with microarray data. Khandoker et al. applied SVM for automated recognition of obstructive sleep apnea syndrome types from their nocturnal electrocardiograph recordings.

The key issue of SVM is kernel function. It is used primarily as an alternative to the complicated inner product operation. In this way, the dimension disaster could be avoided because of the complexity reduction. Nevertheless, SVM has its own inherent limitations in dealing with fixed-length vectors. Spectral features cannot be used directly for SVM, since the spectral features are extracted from utterances of various lengths. In order to overcome this defect of SVM, many new kernels based on fixed vectors were proposed. Longworth and Gales called these SVM kernels as sequence kernel, which could convert variable length feature vectors into fixed length. Campbell proposed Gaussian mixture model (GMM) super-vector based on the in-depth study of GMM parameters and maximum a posteriori (MAP) algorithm. He computed Kullback–Leibler (KL) divergence between pairs of GMM super-vectors to generate KL divergence sequence kernel. You CH, Lee KA and Li H. proposed a novel kernel based on GMM super-vector and Bhattacharyya distance. However, the computing complexity of this kernel increased sharply with the increase of speech data. This problem also would enhance the modeling complexity of SVM. And then, it would affect the recognition speed of system. In addition, speaker voice would be affected by noise and channel distortion inevitably that make the dynamic range of dimension value in SVM become large. Simultaneously, cross-channel degradation is one of the important challenges facing speaker recognition. So, the channel variations between training voice and testing voice have affected and inhibited the performance of the recognition system largely in the practical application of speaker recognition. Kanagasundaram et al. investigated advanced channel compensation techniques such as linear discriminant analysis (LDA), within class covariance normalization (WCCN), and weighted LDA (WLDA) for speaker verification. The experiments based on NIST 2008 and NIST 2010 speaker recognition evaluation (SRE) corpora demonstrated the effectiveness of each channel compensation method. And the experimental results also showed that WCCN performs better than LDA and WLDA as channel variations mainly depended on the within-speaker variation than between-speaker variation.

For the sake of solving the problems of high computing complexity and channel interference susceptibility of GMM super-vectors, we proposed a novel SVM kernel based on Bhattacharyya distance and WCCN smoothing technique inspired by above related research. In our algorithm, we clustered GMMs of all registered speaker according to their Bhattacharyya distance first. By doing so, we expected that the computing complexity of GMM super-vectors would be reduced. And then, the sequence kernel was generated based on the super-vectors of the clustering center models. In order to relieve the influence of noise and channel distortion, WCCN was adopted to restrain the cross-channel interference for this sequence kernel. The main goal of our method is to improve robustness and performance of speaker verification system. The novelty of our method mainly includes as follows. First, we use Bhattacharyya distance instead of the Euclidean distance in K-means clustering. This gives full play to the effective similarity measure between GMMs of Bhattacharyya distance. Second, the Bhattacharyya sequence kernel function is generated by computing Bhattacharyya distance between clustering center model super-vectors and testing speech model super-vectors. Third, WCCN is used to suppress channel interference in the sequence kernel function space.

The remainder of this paper is organized as follows. In the next section, we give a detailed description of our new sequence kernel based on Bhattacharyya distance clustering and WCCN. Experimental evaluation and results discussion based on TIMIT corpus and NIST 2008 SRE are presented in the “Experiments and discussion” section. Finally, conclusions are drawn in the “Conclusions” section.

**SVM kernel based on Bhattacharyya distance clustering and WCCN**

With the increase of system registered speakers, the speech data of speaker verification system will enhance amazingly. It is a serious problem in speaker verification, since that it will lead to huge amount of input samples for training and testing in subsequent process. And it will also slower the classifier training speed. In addition, the
input speech data are vulnerable to variable-length speech sequence and noise interference. Because of the above factors, the good performance could not be achieved in SVM speaker verification. The key problems in SVM speaker verification are how to get SVM to classify on whole sequence and how to remove channel interference. In order to solve these problems, we propose a novel sequence kernel based on in-depth study of Bhattacharyya distance and WCCN. The system framework is shown in Figure 1.

Figure 2 shows the generation diagram of our proposed sequence kernel based on Bhattacharyya distance cluster and WCCN. From Figures 1 and 2, we can easily see that the training process of our proposed speaker verification can be divided into four phases. The first phase was GMM clustering, in which Bhattacharyya distance of GMMs was adopted as clustering measure. The generation of GMM super-vectors was immediately followed by this phase, called the second phase. In order to generate the GMM mean super-vector, a GMM must first be trained from GMM-UBM using MAP adaptation. And the third phase was the generation of our new sequence kernel based on Bhattacharyya distance transformation and WCCN. Finally, we trained SVM using new kernel. In the testing process, we extract the super-vector of testing speech first after SVM was trained successfully, which has the same preprocessing as training speech based on GMM-UBM and MAP. Finally, this testing super-vector is regarded as input of SVM directly to verify the identity of the speaker.

**Speaker GMM clustering based on Bhattacharyya distance**

As previously mentioned, the increase of the enrollment amount will lead to the size of the input data rapidly increase. This will be time-consuming in GMM and then will affect the system performance. In order to reduce the size of input data, we considered to cluster speaker GMM models. Speaker clustering based on distance between models is the most widely used means. Bhattacharyya distance is a common distance measure of Gaussian distributions. Gomathy et al. analyzed the clustering performance of Euclidean distance, Mahalanobis distance, Manhattan distance, and Bhattacharyya distance in speech processing gender clustering and classification. The better performance was achieved in Bhattacharyya distance. Therefore, Bhattacharyya distance has the advantages of simple form and stability in model similarity measurement. Prasanth and Chandra Mouli proposed a robust blind watermarking method using Bhattacharyya distance and exponential function to preserve the copyright protection and identify the ownership of digital data. Inspired by K-means clustering algorithm, we used the Bhattacharyya distance instead of the Euclidean distance to measure the similarity of GMM models. So, we clustered speaker GMMs according to the Bhattacharyya distance between models. The purpose of this was to reduce the number of input models.

![Figure 1. The framework of speaker verification based on WCCN clustering sequence kernel.](image1)

**Figure 1.** The framework of speaker verification based on WCCN clustering sequence kernel.

**Figure 2.** The generation diagram of WCCN clustering kernel.

**Figure 2.** The generation diagram of WCCN clustering kernel.
K denoted as clustering number, and \( p^{(c)} = \{ \omega_{i}^{(c)}, m_{i}^{(c)}, \Sigma_{i}^{(c)} \mid i = 1, \ldots, M \} \) denoted as clustering center model, where \( \omega_{i}^{(c)} = \frac{1}{s_{c}} \sum_{j=1}^{s_{c}} \omega_{ij}^{(c)} \), \( \Sigma_{i}^{(c)} = \frac{1}{s_{c}} \sum_{j=1}^{s_{c}} \Sigma_{ij}^{(c)} \), and \( m_{i}^{(c)} = \frac{1}{s_{c}} \sum_{j=1}^{s_{c}} m_{ij}^{(c)} \) were denoted as the weight, covariance matrix, and means, respectively, and \( s_{c} \) represented the number of GMMs in current category \( c \), where \( c = 1, \ldots, K \).

In our clustering algorithm, we utilized Bhattacharyya distance to replace Euclidean distance in conventional K-means clustering algorithm. The Bhattacharyya distance between speaker GMM \( p^{(s)} \) and clustering center model \( p^{(c)} \) was computed as follows

\[
D_{\text{Bhatt}}(p^{(s)}||p^{(c)}) = -\ln \left( \int_{R^{n}} \sqrt{\sum_{i=1}^{M} p_{i}^{(s)}(x)} \sqrt{\sum_{j=1}^{M} p_{j}^{(c)}(x)} \, dx \right) \\
= \frac{1}{8} (m_{i}^{(c)} - m_{j}^{(c)}) \left[ \frac{\Sigma_{i}^{(c)} + \Sigma_{j}^{(j)}}{2} \right]^{-1} (m_{i}^{(c)} - m_{j}^{(j)})^{T} \\
+ \frac{1}{2} \ln \frac{\Sigma_{i}^{(c)} + \Sigma_{j}^{(j)}}{2} - \frac{1}{2} \ln \left( \Sigma_{i}^{(c)} || \Sigma_{j}^{(j)} \right) - \frac{1}{2} \ln \left( \omega_{i}^{(c)} \omega_{j}^{(j)} \right)
\]

(1)

Since we only adapted the means of GMM according to the generation principle of GMM super-vector, all speakers had same weights and covariance matrix. The upper bound of equation (1) can be achieved as equation (2)

\[
D_{\text{Bhatt}}(p^{(s)}||p^{(c)}) \leq \sum_{i=1}^{M} D_{\text{Bhatt}}(p^{(s)}||p^{(c)}) \\
= \frac{1}{8} \sum_{i=1}^{M} \left\{ (m_{i}^{(c)} - m_{j}^{(c)}) \left[ \frac{\Sigma_{i}^{(c)} + \Sigma_{j}^{(j)}}{2} \right]^{-1} (m_{i}^{(c)} - m_{j}^{(j)})^{T} \right\} \\
+ \frac{1}{2} \sum_{i=1}^{M} \ln \frac{\Sigma_{i}^{(c)} + \Sigma_{j}^{(j)}}{2} + \frac{1}{2} \sum_{i=1}^{M} \ln \left( \omega_{i}^{(c)} \omega_{j}^{(j)} \right)^{-1} \\
\approx \sum_{i=1}^{M} \left\{ (m_{i}^{(c)} - m_{j}^{(c)}) \left[ \frac{\Sigma_{i}^{(c)} + \Sigma_{j}^{(j)}}{2} \right]^{-1} (m_{i}^{(c)} - m_{j}^{(j)})^{T} \right\}
\]

(2)

In order to present our proposed clustering algorithm clearly, we described its procedure as follows.

**Step 1:** Set the number of categories as \( K \), assign \( K \) speaker GMMs to initialize the clustering center model randomly. We defined \( p^{(c)} = \{ \omega_{i}^{(c)}, m_{i}^{(c)}, \Sigma_{i}^{(c)} \mid i = 1, \ldots, M \} \) as clustering center model. \( s_{c} = 0 \) \( (c = 1, \ldots, K) \) was used to record the number of clustered models in current category \( c \).

**Step 2:** Computed Bhattacharyya distance between speaker model \( p^{(s)}(s = 1, \ldots, S) \) and cluster center model \( p^{(c)} \) by equation (2).

**Step 3:** Selected speaker model that had the smallest Bhattacharyya distance and merged it into current category, \( s_{c} = s_{c} + 1 \) and recomputed the new cluster center model.

**Step 4:** Repeated steps 2 and 3 until the cluster center models changed no longer.

**Super-vector extraction of GMM clustering center model**

After speaker GMMs have been clustered into \( K \) clustering center GMM models, we extract super-vectors of these models. The key problem of GMM super-vector is to train the speaker’s GMM by adapting from system UBM. GMM-UBM could solve the problem of data deficiency in the training of speaker’s GMM via EM algorithm. This is mainly because GMM-UBM is trained by the whole speech from all system registered speakers. By doing so, it could represent the speaker independent distribution of speech features. Hence, GMM-UBM can provide a priori knowledge for the training of each utterance’s GMM.\(^{21}\) We assumed the GMM-UBM of our speaker
veriﬁcation system as follows

\[ p(x)^{(u)} = \sum_{i=1}^{M} \omega_i N(x; \mu_i, \Sigma_i) \]  

(3)

where \( N(x; \mu, \Sigma) \) indicates the Gaussian density function of vector \( x \). \( \omega_i \), \( \mu_i \), and \( \Sigma_i \) are the mixture weights, mean, and covariance of \( i \)th Gaussian density component, respectively. GMM-UBM could also be denoted as \( p^{(u)} = \{ \omega_i^{(u)}, \mu_i^{(u)}, \Sigma_i^{(u)} \mid i = 1, \ldots, M \} \). And the single clustering center model could be denoted as \( p^{(c)} = \{ \omega_i^{(c)}, \mu_i^{(c)}, \Sigma_i^{(c)} \mid i = 1, \ldots, M \} \), \( c = 1 \ldots K \) in the same way. Commonly, GMM-UBM is trained via the EM algorithm\(^22\) on feature vectors from all registered speakers. After GMM-UBM was obtained, GMM super-vector set \( S = \{ s_1, s_2, \ldots, s_k \} \) is obtained by adapting mean vectors of speaker’s GMMs, where \( s_i = [s_i^{(1)}, s_i^{(2)}, \ldots, s_i^{(K)}]^T, i = 1, 2, \ldots, K \). The generation process of super-vector is shown in Figure 3.

From Figure 3, we can see that GMM super-vectors have ﬁxed length. Therefore, we can use the GMM super-vectors as the input vectors for SVM learning. However, the dimensions of GMM super-vectors are high. It will slow down the SVM training speed. In order to reduce the computing complexity, covariance matrix usually is adopted in diagonalization form.

The generation of sequence kernel

The defect of classiﬁcation running on the whole speech sequence has become a big obstacle to develop good classiﬁcation performance on SVM in speaker veriﬁcation system. For the sake of overcoming this problem, we proposed a novel sequence kernel on the basis of above speaker clustered models and super-vectors.

Sequence kernel based on Bhattacharyya distance transformation. We denoted the mean distance \( s_i^{(c)} \) between speaker’s GMM clustered model \( p^{(c)} \) and GMM-UBM \( p^{(u)} \) as follows

\[ s_i^{(c)} = \left( \frac{\Sigma_i^{(c)} + \Sigma_i^{(u)}}{2} \right)^{-\frac{1}{2}} (m_i^{(c)} - m_i^{(u)}), \quad i = 1, \ldots, M \]  

(4)

where \( m_i^{(c)} \) and \( m_i^{(u)} \) indicated the means of \( i \)th Gaussian density component of \( p^{(c)} \) and \( p^{(u)} \), respectively. In this paper, we deﬁned \( s_i^{(c)} \) as super-vector of \( p^{(c)} \), so the super-vectors of speaker clustering GMM were \( S = [s_1^{(c)}, s_2^{(c)}, \ldots, s_K^{(c)}]^T \). Therefore, the Bhattacharyya distance between \( p^{(c)} \) and \( p^{(u)} \) can be deﬁned as equation (5) according to equation (2)

\[ D_{\text{Bhatt}}(p^{(c)}|p^{(u)}) = \sum_{i=1}^{M} \left\{ (m_i^{(c)} - m_i^{(u)}) \left( \frac{\Sigma_i^{(c)} + \Sigma_i^{(u)}}{2} \right)^{-\frac{1}{2}} (m_i^{(c)} - m_i^{(u)})^T \right\} \]

\[ = \sum_{i=1}^{M} \left\{ \left( \frac{\Sigma_i^{(c)} + \Sigma_i^{(u)}}{2} \right)^{-\frac{1}{2}} (m_i^{(c)} - m_i^{(u)}) \times \left( \frac{\Sigma_i^{(c)} + \Sigma_i^{(u)}}{2} \right)^{-\frac{1}{2}} (m_i^{(c)} - m_i^{(u)})^T \right\} \]

\[ = \sum_{i=1}^{M} s_i^{(c)} (s_i^{(c)})^T = S^{(c)} (S^{(c)})^T \]  

(5)

**Figure 3.** The generation diagram of GMM super-vector.

GMM: Gaussian mixture model; MAP: maximum a posteriori; UBM: universal background model.
Suppose that the GMM of testing speech is defined as $p^{(test)} = \{ \omega_k^{(test)}, m_k^{(test)}, \Sigma_k^{(test)} \}$. According to equation (5), the mean distance between testing speech super-vector and clustering center model super-vector can be defined as follows

$$D_{Bhatt}(p^{(test-c)}|p^{(u)}) = (S^{(test)} - S^{(c)})(S^{(test)} - S^{(c)})^T$$

Therefore, we define our Bhattacharyya sequence kernel as follows

$$K_{Bhatt}(X_{test}, X_c) = \frac{1}{2} \left[ D_{Bhatt}(p^{(test)}|p^{(a)}) + D_{Bhatt}(p^{(a)}|p^{(u)}) - D_{Bhatt}(p^{(test-c)}|p^{(u)}) \right] = (S^{(test)})(S^{(c)})^T$$

Obviously, equation (7) satisfies the Mercer condition. It could be regarded as the inner product of GMM-UBM mean distance super-vectors. This is a novel clustering sequence kernel. It could make SVM classify on whole speech sequence.

**Channel compensation using WCCN for new kernel.** In speaker verification system, speech is vulnerable to noise and channel distortion. This will lead to the decline of system performance. In order to solve this problem, we utilized WCCN\textsuperscript{23} to restrain noise and channel distortion in kernel function space. The covariance matrix of single speaker only reflects the influence of noise. It does not reflect the channel characteristic yet. Therefore, we use more than one speaker’s average covariance matrix.

We denote $K$ as the number of clustering, the average covariance matrix of registered speakers was

$$\tilde{\Sigma} = \sum_{i=1}^{K} \rho_i \Sigma_i,$$

where $\rho_i (i = 1, \ldots, K)$ indicates the proportion of the number of samples to total number of samples in $i$th category, $\Sigma_i = E(x_i - \bar{x}_i)(x_i - \bar{x}_i)^T$, $\forall i$ is the sample covariance matrix of $i$th category. We adopt smoothing model as follows

$$\tilde{\Sigma} = (1 - \alpha) \cdot \Sigma + \alpha I$$

$$= \left( 1 - \frac{\alpha}{\alpha} \right) \begin{bmatrix} \Sigma & \Sigma \end{bmatrix}$$

where $\tilde{\Sigma}$ is within class covariance matrix for $\tilde{\Sigma}$ after application of smoothing technique, and $I$ is square matrix of $N \times N$. The diagonal elements of $I$ are 1. $N$ is the dimension of speaker feature vector. The parameter $\alpha \in [0, 1]$ is an adjustable smoothing factor. The speaker verification experimental results of Hatch and Stolcke\textsuperscript{24} showed that the equal error rate (EER) and minimum decision cost function (minDCF) achieved minimum when $\alpha = 0.3$. Thus, we set $\alpha = 0.3$ in this paper. We introduce $\tilde{\Sigma}$ into Bhattacharyya distance clustering kernel

$$K_{Bhatt}(X_{test}, X_c) = (S^{(test)})(S^{(c)})^T$$

where $U\Lambda U^T$ feature is decomposed matrix of $\tilde{\Sigma}$. WCCN could remove noise and channel information effectively. By doing so, the performance of SVM could be improved greatly.

**SVM based on new kernel for speaker verification**

SVM, invented by Vapnik,\textsuperscript{25} is a powerful tool for data classification. According to the principle of structural risk minimization, SVM could find the optimal decision boundary in two classes of input samples. It is one of the most robust binary classifier in speaker verification. The basic idea of SVM is illustrated in Figure 4.
In our method, the new feature vectors of target speaker and imposters are used to train SVM, so the class decision function for each speaker can be obtained as follows:

$$f(x) = \sum_{i=1}^{l} y_i a_i K(x_i, x) + b$$  \hspace{1cm} (10)

where $x_i \in \mathbb{R}^n, i = 1, 2, \ldots, l$ are the training new Fisher feature vectors. Each $x_i$ belongs to one of two classes identified by the label $y_i \in \{-1, 1\}$. The coefficients $a_i$ and $b$ are the solution of a quadratic programming problem. $a_i$ is non-zero for support vectors (SV) and is zero otherwise. $K(,)$ is the kernel function. In this paper, we selected our proposed new kernel. Sequential minimal optimization is selected to train SVM.

We define the Lagrange function of SVM as

$$L(w, b, a) = \frac{1}{2} ||w||^2 - \sum_{i=1}^{n} a_i \{y_i [(w \cdot x_i) + b] - 1 \}$$  \hspace{1cm} (11)

We will refer to the $a_i$ as Lagrange multipliers. It could be obtained by solving the following dual problem

$$\max_a \sum_{i=1}^{l} a_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j a_i a_j K(x_i, x_j)$$  \hspace{1cm} (12)

s.t. $\sum_{i} y_i a_i = 0, \ 0 \leq a_i \leq C$

where $C$ is the penalty coefficient. Once $a_i$ is obtained, $w$ and $b$ can be determined by using the Karush–Kuhn–Tucker condition

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l} y_i a_i x_i = 0$$  \hspace{1cm} (13)

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{l} y_i a_i = 0$$  \hspace{1cm} (14)

The separation equation can be determined by using bound SV $x_i \in SV$ as follows

$$\sum_{x_i \in SV} y_i a_i K(x_i, x) + b = 0$$  \hspace{1cm} (15)

Figure 4. The principle presentation drawing of SVM.

- class 1
- class 2

$w + b = 0$

support vectors

marg
Where

\[ b = -\frac{1}{2} \left[ \max_{y_i \in S^P} \left( \sum_{x_i \in S^P} y_i x_i \right) \right] + \max_{y_i \in S^P} \left[ \sum_{x_i \in S^P} y_i x_i x_i \right] \right] \quad (16) \]

**Experiments and discussion**

**Speech database and preprocessing**

Speaker recognition experiments were carried out based on TIMI database and NIST 2008 SRE dataset. We employed TIMI database to test new kernel performance. And NIST 2008 SRE was utilized to test the robustness of new kernel.

In our experiments, two speech databases had the same speech preprocessing. The first-order digital filter was \( H(z) = 1 - 0.95z^{-1} \). In the stage of framing and adding Hamming window, the frame size was 30 ms and frame shift was 15 ms. The utterance of every speaker owned 1999 frames. We extracted 13 dimensional MFCCs and their first and second derivatives and combine them into a 39 dimensional vector as input features of all our experiments. In our experiment, the number of training vectors was 5 \( \times \) 1999 = 9995. The number of GMM-UBM mixtures was 1024. The original MFCC coefficients extracted from the feature vectors have wide variations. These variations not only come from speaking at different times but also come from different channels. We adopted cepstral mean variance normalization to compensate for these changes. The calculation was shown as follows

\[ x'_i = \frac{x_i - \mu}{\sigma} \quad (17) \]

where \( x_i \) is the original coefficient and \( x'_i \) indicated normalized speech coefficient of the \( i \)th frame. \( \mu \) is mean vector and \( \sigma \) is standard deviation

\[ \mu = \frac{1}{N} \sum_{i=1}^{N} x_i \quad (18) \]

\[ \sigma = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N - 1}} \quad (19) \]

where \( N \) is the frame number of a single speech utterance. The processed MFCC parameters constitute the input data of our system. In the training phase, the input data moved through three stages first: GMM clustering, GMM super-vector generation, and new kernel generation. And then SVM training was run on this new kernel. We adopted EER, minDCF, and recognition time (RT) as metrics for evaluation.

**Experiments based on TIMI corpus**

TIMIT speech database contains broadband recordings of 630 speakers including 438 male and 192 females. And all speakers had eight major dialects of American English labeled from D1 to D8. Each speaker was asked for reading 10 different phonetically rich sentences. The total sentences are 6300. The sample information distribution is listed in Table 1. Two of these sessions were dialect uttering by every speaker, and other eight sessions were different for each speaker. The speech signal was recorded through a high-quality microphone in quiet environment, with a 0–8 kHz bandwidth. The signal was sampled at 16 kHz, on 16 bits, on a linear amplitude scale. In our experiments, the training set contains five utterances of each speaker, randomly chosen from the 10 sessions, and the testing set contains the rest of five utterances of each speakers.

**The clustering performance testing of new kernel.** In this experiment, we placed emphasis on the clustering performance testing of our proposed kernel. In order to compare the clustering performance of our proposed kernel, we set the
values of $K$ as 630, 460, 300, 200, 150, 100, and 50 in the clustering algorithm, respectively. The experimental results are shown in Table 2.

$K$ is the number of clusters. If there are too many clusters, there is no clustering effect. The number of registered speakers is not effectively reduced, and the amount of data is large, which affects the computational complexity of subsequent processing. The system RT will be longer. In Table 2, when $K = 630$, the RT is 7.44 s.

With the slow reduction of value of $K$, a large number of similar models are gathered together to generate a new model. It effectively reduces the amount of data. However, when the value of $K$ becomes smaller, the system EER has become larger. This indicates that the number of training data is too small to improve the recognition performance. However, the RT of the system is very short due to the small amount of data, such as when $K = 50$, the RT is only 1.33 s. From Table 2, the condition of $K = 300$ obtained the best performance in our proposed kernel. So, we selected $K = 300$ in subsequent experiment and reduction ratio of training GMM size reached 56.52%. The RT comparison in different clustering number is shown in Figure 5.

The performance comparison between our kernel and other different kernels. In our experiments, we selected linear kernel, polynomial kernels ($n = 3$), radial basis function (RBF) kernels ($\sigma = 1.6$), KL divergence sequence kernel, and

| Table 1. The distribution of training and testing sample information. |
|---------------------------------|
| Dialect region | Training speakers | Testing speakers | Total |
|-----------------|-------------------|------------------|-------|
| D1              | 38                | 11               | 49    |
| D2              | 76                | 26               | 102   |
| D3              | 76                | 26               | 102   |
| D4              | 68                | 32               | 100   |
| D5              | 68                | 30               | 98    |
| D6              | 35                | 11               | 46    |
| D7              | 77                | 23               | 100   |
| D8              | 22                | 11               | 33    |
| Total           | 460               | 170              | 630   |

| Table 2. The clustering performance comparison of our proposed new kernel. |
|---------------------------------|
| Cluster number | EER (%) | minDCF x 100 | RT (s) |
|----------------|---------|--------------|--------|
| $K = 630$      | 7.08    | 6.04         | 7.44   |
| $K = 460$      | 6.33    | 5.25         | 3.46   |
| $K = 300$ (The best cluster number) | 4.32 | 1.78 | 2.03 |
| $K = 200$      | 4.94    | 3.04         | 1.84   |
| $K = 150$      | 5.39    | 4.03         | 1.79   |
| $K = 100$      | 6.92    | 5.76         | 1.52   |
| $K = 50$       | 10.31   | 8.72         | 1.33   |

EER: equal error rate; minDCF: minimum decision cost function; RT: recognition time.

Figure 5. The RT comparison in different clustering number.
Bhattacharyya linear kernel as the baseline kernel compared to our proposed kernel. The experimental results are showed in Table 3.

| Kernel                             | EER (%) | minDCF × 100 | RT (s) |
|------------------------------------|---------|--------------|--------|
| Linear kernel                      | 12.41   | 6.23         | 3.45   |
| Polynomial kernels \((n = 3)\)     | 9.73    | 5.09         | 3.07   |
| RBF \((\gamma = 1.6)\)            | 6.92    | 4.67         | 2.89   |
| KL divergence sequence kernel      | 5.61    | 4.15         | 2.54   |
| Bhattacharyya linear kernel        | 5.11    | 3.72         | 2.31   |
| Our proposed kernel (the best performance) | 4.32    | 1.78         | 2.03   |

EER: equal error rate; KL: Kullback–Leibler; minDCF: minimum decision cost function; RBF: radial basis function; RT: recognition time.

Table 3. EER and minDCF comparison of different kernels.

Obviously, from Table 3 we could see as follows:
The performance of Bhattacharyya linear kernel improved significantly compared to linear kernel, of which EER decreased by 7.3% and minDCF decreased by 0.0251. Compared to polynomial kernels \((n = 3)\), EER of Bhattacharyya linear kernel decreased by 4.62% and its minDCF decreased by 0.0137. And compared to RBF, the EER of Bhattacharyya linear kernel is down nearly 2%, the minDCF fell by 0.0095. Compared to KL divergence sequence kernel, EER decreased by 0.5% and minDCF decreased by 0.0043. Therefore, Bhattacharyya sequence kernel could make SVM classify on whole speech sequence and improved the performance of SVM drastically. It is an efficient and feasible SVM kernel.

Our proposed kernel was superior to Bhattacharyya linear kernel in EER, minDCF, and RT. The EER of our proposed kernel decreased by 0.79%, minDCF fell by 0.0194, and RT shortened 0.28 s. The experimental results showed that our proposed kernel not only had the advantages of the Bhattacharyya linear kernel, but also had a shorter recognition time. The superiority of our proposed kernel is attributed to the speaker models clustering first. By doing so, it could reduce the computational complexity of GMM-UBM model and shorten the RT. Second, the use of WCCN could restrain the influence of noise and channel distortion to kernel, and improve the system performance effectively. The EER comparison of different kernels is shown in Figure 6.

| Kernel                             | EER (%) |
|------------------------------------|---------|
| Linear kernel                      | 12.41   |
| Polynomial kernel                  | 9.73    |
| RBF                                | 6.92    |
| KL divergence sequence kernel      | 5.61    |
| Bhattacharyya linear kernel        | 5.11    |
| Our proposed kernel (the best performance) | 4.32    |

Figure 6. The EER comparison of different kernels.

Experiments based on NIST 2008

The robustness test of new kernel. In order to test the robustness of new kernel, we carried out our testing experiment on NIST 2008 SRE database. We denoted our kernel without WCCN channel compensation as clustering kernel. NIST 2008 evaluation was performed using the telephone–telephone, interview–interview, telephone–microphone, and interview–telephone enrolment-verification conditions. The voice parameter extraction was the same as TIMI corpus. Performance measurement also was EER and minDCF. The experimental results are shown in Table 4.

Form Table 4, it was seen that the performance of clustering kernel based on WCCN was completely superior to kernel without channel compensation in telephone–telephone, interview–interview, telephone–microphone, and
interview–telephone enrolment-verification conditions. In telephone–telephone condition, EER decreased by 0.54% and minDCF fell by 0.0072. In other conditions, EER and minDCF of our proposed kernel had obvious reduction. Therefore, the proposed new kernel had better robustness.

The performance comparison between our speaker verification system and other systems.

In this experiment, we mainly focus on comparing the performance of our proposed speaker verification system with the state-of-the-art system such as GMM-SVM and SVM. The number of GMM-UBM mixtures was 1024. We selected EER and minDCF as performance measurement. The experimental results are showed in Table 5.

From Table 5, we could easily see that our proposed system has superior performance compared with GMM-SVM and SVM. In the telephone–telephone condition, the EER of our system decreased by 2%, minDCF fell by 0.0134 compared to GMM-SVM. Compared with SVM, the EER of our system decreased by 5.1% and minDCF fell by 0.0436. In the same way, our system also showed superior performance in other enrolment-verification conditions. Our system has shown its effectiveness.

Conclusions

On the in-depth study of k-means clustering algorithm and GMM super-vector, a novel sequence kernel based on Bhattacharyya distance clustering and WCCN was proposed in this paper. With the aid of WCCN smoothing technique, the noise and channel distortion were eliminated in kernel space. This improved the system recognition accuracy effectively. At the same time, our proposed clustering algorithm based on the Bhattacharyya distance could effectively reduce the computational complexity of speaker models and speed up the system RT. Our proposed kernel was proven to be an effective and feasible sequence kernel on the corpus of TIMIT and NIST 2008 SRE database in SVM speaker verification. However, there are large numbers of matrix operations in the process of our kernel generation and WCCN smooth normalization. These will enhance the computing complexity of kernel. Therefore, how to simplify the solving process of our kernel is our focus in the follow-up work.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This paper is supported by National Natural Science Foundation of China (NSFC) (61841203), China.
References

1. Nematollahi MA and Al-Haddad SAR. Distant speaker recognition: an overview. *Int J Hum Robot* 2016; 13: 45.
2. Ning J, Qiu M and Dai W. SROC: a speaker recognition with data decision level fusion method in cloud environment. *J Signal Process Syst* 2016; 86: 123–133.
3. Chakroun R, Zouari LB and Frihka M. A hybrid system based on GMM-SVM for speaker identification. In: *International conference on intelligent systems design and applications*, 2016, pp.654–658, Marrakech, Morocco: IEEE.
4. Schnall A and Heckmann M. Feature-space SVM adaptation for speaker adapted word prominence detection. *Comput Speech Lang* 2018; 53: 198–216.
5. Sloin A and Burshtein D. Support vector machine training for improved hidden Markov modeling. *IEEE Trans Signal Process* 2008; 56: 172–188.
6. Chu F and Wang LP. Applications of support vector machines to cancer classification with microarray data. *Int J Neural Syst* 2005; 15: 475–484.
7. Khandoker AH, Palaniswami M and Karmakar CK. Support vector machines for automated recognition of obstructive sleep apnea syndrome from ECG recordings. *IEEE Trans Inform Technol Biomed* 2009; 13: 37–48.
8. Faragallah OS. Robust noise MKMFCC–SVM automatic speaker identification. *Int J Speech Technol* 2018; 21: 185–192.
9. Hu YJ and Ling ZH. Extracting spectral features using deep autoencoders with binary distributed hidden units for statistical parametric speech synthesis. *IEEE/ACM Trans Audio Speech Lang Process* 2018; 26: 713–724.
10. Zha C, Zhang X and Zhao L. Speaker-independent speech emotion recognition based multiple kernel learning of collaborative representation. *IEICE Trans Fundam Electron Commun Comput Sci* 2016; 99: 756–759.
11. Longworth C and Gales MJF. Combining derivative and parametric kernels for speaker verification. *IEEE Trans Audio Speech Lang Process* 2009; 17: 748–757.
12. Das RK, Manam AB and Prasanna SRM. Exploring kernel discriminant analysis for speaker verification with limited test data. *Pattern Recognit Lett* 2017; 98: 26–31.
13. Kanagasundaram A, Dean D and Sridharan S. I-vector based speaker recognition using advanced channel compensation techniques. *Comput Speech Lang* 2014; 28: 121–140.
14. Gomathy M, Gandhi SI and Meena K. Gender clustering and classification algorithms in speech processing: a comprehensive performance analysis. *Int J Comput Appl* 2012; 51: 9–17.
15. Prasanth VS and Chandra Mouli PVSSR. Adaptive, robust and blind digital watermarking using Bhattacharyya distance and bit manipulation. *Multimed Tools Appl* 2018; 77: 5609–5635.
16. Qin S, Liu C and Huang, S. Identification rice varieties based on k-means clustering algorithm and BP neural network. In: *International conference on advanced materials science and environment engineering*, 2017, pp.252–257. Sanya, China: DEStech.