Int J Speech Technol (2012) 15:365–379
DOI 10.1007/s10772-012-9154-4

Speaker recognition using pyramid match kernel based support vector machines

A.D. Dileep · C. Chandra Sekhar

Received: 20 March 2012 / Accepted: 26 May 2012 / Published online: 12 June 2012

Abstract Gaussian mixture model (GMM) based approaches have been commonly used for speaker recognition tasks. Methods for estimation of parameters of GMMs include the expectation-maximization method which is a non-discriminative learning based method. Discriminative classifier based approaches to speaker recognition include support vector machine (SVM) based classifiers using dynamic kernels such as generalized linear discriminant sequence kernel, probabilistic sequence kernel, GMM supervector kernel, GMM-UBM mean interval kernel (GUMI) and intermediate matching kernel. Recently, the pyramid match kernel (PMK) using grids in the feature space as histogram bins and vocabulary-guided PMK (VGPMK) using clusters in the feature space as histogram bins have been proposed for recognition of objects in an image represented as a set of local feature vectors. In PMK, a set of feature vectors is mapped onto a multi-resolution histogram pyramid. The kernel is computed between a pair of examples by comparing the pyramids using a weighted histogram intersection function at each level of pyramid. We propose to use the PMK-based SVM classifier for speaker identification and verification from the speech signal of an utterance represented as a set of local feature vectors. The main issue in building the PMK-based SVM classifier is construction of a pyramid of histograms. We first propose to form hard clusters, using *k*-means clustering method, with increasing number of clusters at different levels of pyramid to design the codebook-based PMK (CBPMK). Then we propose the GMM-based PMK (GMMPMK) that uses soft clustering. We compare the performance of the GMM-based approaches, and the PMK and other dynamic kernel SVM-based approaches to speaker identification and verification. The 2002 and 2003 NIST speaker recognition corpora are used in evaluation of different approaches to speaker identification and verification. Results of our studies show that the dynamic kernel SVM-based approaches give a significantly better performance than the state-of-the-art GMM-based approaches. For speaker recognition task, the GMMPMK-based SVM gives a performance that is better than that of SVMs using many other dynamic kernels and comparable to that of SVMs using state-of-the-art dynamic kernel, GUMI kernel. The storage requirements of the GMMPMK-based SVMs are less than that of SVMs using any other dynamic kernel.

Keywords Speaker identification · Speaker verification · Kernel methods · Support vector machines · Dynamic kernel · Pyramid match kernel

1 Introduction

Speaker recognition tasks include speaker identification and speaker verification (Reynolds 1995; Kinnunen and Li 2010). Speaker recognition tasks involve processing continuous valued feature vectors extracted from the speech signal of an utterance. For a text independent speaker recognition task, the sequence information in the utterance is not considered to be important. Therefore an utterance is represented by a set of feature vectors. The size of the set is dependent on the duration of the utterance. The generative models such as Gaussian mixture models (GMMs) (Reynolds 1995;

A.D. Dileep (✉) · C.C. Sekhar
Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai 600036, Tamilnadu, India
e-mail: dileepad@cse.iitm.ac.in

C.C. Sekhar
e-mail: chandra@cse.iitm.ac.in

Reynolds et al. 2000) are commonly used for classification of varying length patterns represented as sets of feature vectors. The maximum likelihood (ML) based method is commonly used for estimation of parameters of the GMM for each class. The ML based method yields robust estimates of the model parameters only when the sufficient training data is available. When the amount of training data available in each speaker is limited, robust estimates of model parameters can be obtained through maximum a posteriori (MAP) adaptation of the class-independent GMM (CIGMM) (also called as the universal background model (UBM)) to the training data of each speaker (Reynolds et al. 2000). Recently, the discriminative training based large margin method has been proposed for estimation of parameters of GMM (Sha and Saul 2006). In this method, the parameters of the GMMs of all the classes are estimated simultaneously by solving an optimization problem to maximize the distance between the boundaries of the classes. When the number of classes is large, the optimization problem solving in the large margin method for GMMs is computationally highly intensive.

The discriminative model based approaches such as the support vector machine (SVM) based approaches (Burges 1998) construct the decision boundaries between the classes without having to capture the distributions of the data of the classes. The choice of the kernel function used is important for the performance of SVM-based approaches, and several kernel functions have been proposed for static patterns. The kernel functions designed for static patterns are called static kernels. Recently, the SVM-based approaches have been proposed for varying length pattern classification tasks in which a varying length pattern is represented by a set of feature vectors. These approaches are suitable for speaker recognition tasks in which the speech signal of an utterance from a speaker is represented by a set of feature vectors. The main issue in building an SVM-based classifier for varying length patterns represented by sets of feature vectors is the design of a suitable kernel function that gives a measure of similarity between a pair of varying length patterns. Kernel functions designed for varying length patterns are referred to as dynamic kernels (Wan and Renals 2002). Different approaches to designing dynamic kernels are as follows: (1) Explicit mapping based approaches (Campbell et al. 2006a), (2) Probabilistic distance metric based approaches (Campbell et al. 2006b), and (3) Matching based approaches (Boughorbel et al. 2005). Recently, several types of dynamic kernels have been proposed and the SVM-based classifiers using these dynamic kernels have been developed for speaker recognition tasks (Campbell et al. 2006a, 2006b; Lee et al. 2007; You et al. 2009). The generalized linear discriminant sequence kernel (Campbell et al. 2006a) and probabilistic sequence kernel (Lee et al. 2007) are the dynamic kernels designed using explicit mapping based approaches. The GMM supervector kernel (Campbell et al. 2006b) and GMM-UBM mean interval kernel (You et al. 2009) are the dynamic kernels designed using the probabilistic distance metric based approaches. In this work, we focus on the matching based approaches to designing dynamic kernels for SVM-based speaker recognition systems.

The summation kernel (Boughorbel et al. 2004) for a pair of examples represented as sets of feature vectors is computed as a combination of base kernels computed on all possible pairs of local feature vectors selected from the two examples. The base kernel is a static kernel. An intermediate matching kernel (IMK) (Dileep and Sekhar 2011) for a pair of examples is constructed as a combination of base kernels computed on pairs of the local feature vectors selected by matching the local feature vectors of the examples with a fixed number of virtual feature vectors. The main issue in the construction of IMK is the choice of the set of virtual feature vectors used for matching. In Boughorbel et al. (2005), the set of the centers of clusters formed from the training data of all classes is considered as the set of virtual feature vectors. In Dileep and Sekhar (2011), the components of the class-independent GMM (CIGMM) is used as the set of virtual feature vectors. Another dynamic kernel using the matching based approach is the pyramid matching kernel (PMK) (Grauman and Darrell 2007). In PMK, an example represented as a set of feature vectors is mapped onto a multi-resolution histogram pyramid. The histogram at a level is computed by binning the feature vectors of an example into discrete regions. An histogram intersection function is used to compute the number of matches between a pair of examples at each level. The PMK between a pair of examples is computed as a weighted sum of the number of new matches at the different levels of pyramid. The main issue in the computation of PMK is the construction of histogram pyramids. In Grauman and Darrell (2007), histograms with grids as bins are considered. The computation of the histogram becomes difficult when the dimension of the feature vectors in the set increases. For the set of feature vectors with large dimension, the vocabulary-guided pyramid match kernel (VGPMK) is introduced in Grauman (2006). In VGPMK, the feature vectors of training examples are clustered using k-means clustering method, where the feature vectors in each cluster at a level are further clustered into b clusters at the next level. In VGPMK, each cluster at a level is considered as a bin in the histogram at that level. Irrespective of the cluster size, each cluster at a level is further divided into b clusters at next level. In this work, we propose to cluster the feature vectors of all the training examples into b^j clusters at each level j. This way of clustering helps to restructure the clusters at each level. The clusters at each level are represented by a codebook and hence the kernel is called as codebook-based PMK (CBPMK). The construction of VGPMK and CBPMK involves hard clustering. In this work, we also propose a novel approach to build a PMK that makes use of GMMs. The class-independent GMMs built with increasingly larger number of components are used to construct the histograms at the different levels. We study the performance of SVM-based classifiers using the proposed GMM-based pyramid match kernel (GMMPMK) for the speaker identification and verification task. We also compare the speaker identification and verification performance with the GMM-based classifiers, and SVM-based classifiers with other matching based dynamic kernels and the state-of-the-art dynamic kernels such as GMM supervector kernel and GMM-UBM mean interval kernel.

The organization of the rest of the paper is as follows: Sect. 2 presents the dynamic kernels and the SVM-based approaches to speaker recognition. The pyramid matching kernel based SVM is described in Sect. 3. Our studies on speaker identification and verification using the GMM-based approaches and the SVM-based approaches are presented in Sect. 4. The summary and conclusions are presented in Sect. 5.

2 SVM-based approaches to speaker recognition

In the speaker recognition task, the speech signal of an utterance processed using a short-time analysis technique is represented as a set of feature vectors. The size of the set of feature vectors depends on the duration of the utterance. An SVM using a kernel such as Gaussian kernel and polynomial kernel can not handle the varying length patterns. The kernels designed for varying length patterns are referred to as dynamic kernels (Wan and Renals 2002). Different approaches for designing dynamic kernels are as follows: (1) Explicit mapping based approaches (Campbell et al. 2006a; Lee et al. 2007), where a set of feature vectors is mapped onto a fixed dimensional representation and a kernel function is defined in the space of that representation, (2) Probabilistic distance metric based approaches (Campbell et al. 2006b; You et al. 2009), where a suitable distance measure for two sets of feature vectors is kernelized, and (3) Matching based approaches (Boughorbel et al. 2005; Dileep and Sekhar 2011), where a kernel function is defined by matching the feature vectors in the pair of examples. Some of the dynamic kernels commonly used for speaker identification and verification tasks are the generalized linear discriminant sequence kernel (Campbell et al. 2006a), probabilistic sequence kernel (Lee et al. 2007), GMM supervector kernel (Campbell et al. 2006b), GMM-UBM mean interval kernel (You et al. 2009) and intermediate matching kernel (Dileep and Sekhar 2011).

2.1 Generalized linear discriminant sequence kernel

Generalized linear discriminant sequence (GLDS) kernel (Campbell et al. 2006a) uses an explicit expansion into a kernel feature space defined by the polynomials of degree p. The GLDS kernel is derived from the generalized linear discriminant method used in Campbell et al. (2002). The GLDS kernel is derived using the polynomial expansions of feature vectors in the examples represented as sets of feature vectors. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, where $\mathbf{x}_t \in \mathbb{R}^d$ is a set of T feature vectors. A feature vector $\mathbf{x}_t$ is represented in a higher dimensional feature space $\boldsymbol{\Psi}$ as a polynomial expansion $\boldsymbol{\Psi}(\mathbf{x}_t) = [\psi_1(\mathbf{x}_t), \psi_2(\mathbf{x}_t), \ldots, \psi_r(\mathbf{x}_t)]^t$, where r is the number of monomials of elements of $\mathbf{x}_t$. The expansion $\boldsymbol{\Psi}(\mathbf{x}_t)$ includes all monomials of elements of $\mathbf{x}_t$ up to and including degree p. The set of feature vectors $\mathbf{X}$ is represented as a fixed dimensional vector $\boldsymbol{\Phi}(\mathbf{X})$ obtained as follows:

$$\boldsymbol{\Phi}_{\text{GLDS}}(\mathbf{X}) = \frac{1}{M} \sum_{t=1}^{T} \boldsymbol{\Psi}(\mathbf{x}_t) \tag{1}$$

The GLDS kernel between two examples $\mathbf{X}_m = \{\mathbf{x}_{m1}, \mathbf{x}_{m2}, \ldots, \mathbf{x}_{mT_m}\}$ and $\mathbf{X}_n = \{\mathbf{x}_{n1}, \mathbf{x}_{n2}, \ldots, \mathbf{x}_{nT_n}\}$ is given as

$$K_{\text{GLDS}}(\mathbf{X}_m, \mathbf{X}_n) = \left(\frac{1}{T_m} \sum_{m=1}^{T_m} \boldsymbol{\Psi}(\mathbf{x}_m)\right)^t \mathbf{S}^{-1} \left(\frac{1}{T_n} \sum_{n=1}^{T_n} \boldsymbol{\Psi}(\mathbf{x}_n)\right) \tag{2}$$

Let L be the total number of feature vectors from all the examples in the training dataset which includes the data belonging to two classes. The correlation matrix $\mathbf{S}$ is defined as follows:

$$\mathbf{S} = \frac{1}{L} \mathbf{R}^t \mathbf{R} \tag{3}$$

where $\mathbf{R}$ is the matrix whose rows are the polynomial expansions of the feature vectors in the training set.

2.2 Probabilistic sequence kernel

Probabilistic sequence kernel (PSK) (Lee et al. 2007) maps a set of feature vectors onto a probabilistic feature vector obtained using generative models. It is known that the Gaussian components of a GMM trained for a speaker correspond to the underlying broad phonetic classes for that speaker (Reynolds et al. 2000) and the different phonetic classes have unequal discrimination power between the speakers (Auckenthaler et al. 1999). This is the motivation for using speaker-dependent weighing of the Gaussian probabilities in the design of PSK to enhance separation of that speaker from the others. The PSK uses the universal background model (UBM) with Q mixtures (Reynolds et al. 2000) and the class-specific GMM obtained by adapting UBM. The likelihood of a feature vector $\mathbf{x}$ being generated by the $2Q$-mixture GMM that includes the UBM and class-specific GMM is given as

$$p(\mathbf{x}) = \sum_{q=1}^{2Q} p(\mathbf{x}|q)P(q) \tag{4}$$

where $P(q)$ denotes the mixture weight and $p(\mathbf{x}|q) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$. The normalized Gaussian basis function for the qth component is defined as

$$\psi_q(\mathbf{x}) = \frac{p(\mathbf{x}|q)P(q)}{\sum_{q'=1}^{2Q} p(\mathbf{x}|q')P(q')} \tag{5}$$

A feature vector $\mathbf{x}$ is represented in a higher dimensional feature space as a vector of normalized Gaussian basis functions, $\boldsymbol{\Psi}(\mathbf{x}) = [\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \ldots, \psi_{2Q}(\mathbf{x})]^t$. Since the element $\psi_q(\mathbf{x})$ indicates the probabilistic alignment of $\mathbf{x}$ to the qth component, $\boldsymbol{\Psi}(\mathbf{x})$ is called as the probabilistic alignment vector. A set of feature vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ is represented as a fixed dimensional vector $\boldsymbol{\Phi}(\mathbf{X})$ in the higher dimensional space, as given by

$$\boldsymbol{\Phi}_{\text{PSK}}(\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\Psi}(\mathbf{x}_t) \tag{6}$$

Then, the PSK between two examples $\mathbf{X}_m = \{\mathbf{x}_{m1}, \mathbf{x}_{m2}, \ldots, \mathbf{x}_{mT_m}\}$ and $\mathbf{X}_n = \{\mathbf{x}_{n1}, \mathbf{x}_{n2}, \ldots, \mathbf{x}_{nT_n}\}$ is given as

$$K_{\text{PSK}}(\mathbf{X}_m, \mathbf{X}_n) = \left(\frac{1}{T_m}\sum_{m=1}^{T_m} \boldsymbol{\Psi}(\mathbf{x}_m)\right)^t \mathbf{S}^{-1} \left(\frac{1}{T_n}\sum_{n=1}^{T_n} \boldsymbol{\Psi}(\mathbf{x}_n)\right) \tag{7}$$

where $\mathbf{S}$ is the correlation matrix as in (3), except that it is obtained using the probabilistic alignment vectors.

2.3 GMM supervector kernel

The GMM supervector (GMMSV) kernel (Campbell et al. 2006b) performs a mapping of a set of feature vectors onto a higher dimensional vector corresponding to a GMM supervector. An UBM is built using the training examples of all the classes. An example-specific GMM is built for each example by adapting only the means of the UBM using the data of that example. Let $\boldsymbol{\mu}_q^{(X)}$ be the mean vector of qth component in the example-specific GMM for an example $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$. Let $p(\mathbf{x}) = \sum_{q=1}^{Q} \pi_q \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ be the likelihood for a feature vector $\mathbf{x}$ using the UBM with Q components. Let $p_X(\mathbf{x}) = \sum_{q=1}^{Q} \pi_q \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_q^{(X)}, \boldsymbol{\Sigma}_q)$ be the likelihood for $\mathbf{x}$ using the example-specific GMM. The Kullback-Leibler (KL) divergence (Kullback and Leibler 1951) is used to represent the dissimilarity between two GMMs. A GMM vector $\boldsymbol{\Psi}_q(\mathbf{X})$ for an example $\mathbf{X}$ is obtained as follows:

$$\boldsymbol{\Psi}_q(\mathbf{X}) = \left[\sqrt{\pi_q}\,\boldsymbol{\mu}_q^{(X)} \boldsymbol{\Sigma}_q^{-\frac{1}{2}}\right]^t \tag{8}$$

The GMM supervector for the example $\mathbf{X}$ is given by

$$\boldsymbol{\Phi}_{\text{GMMSV}}(\mathbf{X}) = \left[\boldsymbol{\Psi}_1(\mathbf{X})^t, \boldsymbol{\Psi}_2(\mathbf{X})^t, \ldots, \boldsymbol{\Psi}_Q(\mathbf{X})^t\right]^t \tag{9}$$

The GMMSV kernel between a pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ is given by

$$K_{\text{GMMSV}}(\mathbf{X}_m, \mathbf{X}_n) = \boldsymbol{\Phi}_{\text{GMMSV}}(\mathbf{X}_m)^t \boldsymbol{\Phi}_{\text{GMMSV}}(\mathbf{X}_n) \tag{10}$$

2.4 GMM-UBM mean interval kernel

The construction of GMM-UBM mean interval (GUMI) kernel (You et al. 2009) also involves building the UBM. An example-specific GMM is built for each example by adapting the means and covariance matrices of the UBM using the data of that example. Let $\boldsymbol{\mu}_q^{(X)}$ and $\boldsymbol{\Sigma}_q^{(X)}$ be the mean vector and the covariance matrix of qth component in the example specific GMM for an example $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$. Let $p(\mathbf{x}) = \sum_{q=1}^{Q} \pi_q \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ be the likelihood for a feature vector $\mathbf{x}$ using the UBM with Q components. Let $p_X(\mathbf{x}) = \sum_{q=1}^{Q} \pi_q \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_q^{(X)}, \boldsymbol{\Sigma}_q^{(X)})$ be the likelihood for a feature vector $\mathbf{x}$ using the example-specific GMM. The Bhattacharyya mean distance (Kailath 1967) is used to represent the dissimilarity between two GMMs. A GUMI vector $\boldsymbol{\Psi}_q(\mathbf{X})$ for an example $\mathbf{X}$ is obtained as follows:

$$\boldsymbol{\Psi}_q(\mathbf{X}) = \left(\frac{\boldsymbol{\Sigma}_q^{(X)} + \boldsymbol{\Sigma}_q}{2}\right)^{-\frac{1}{2}} \left(\boldsymbol{\mu}_q^{(X)} - \boldsymbol{\mu}_q\right) \tag{11}$$

The GUMI supervector is obtained by concatenating the GUMI vectors of different components as

$$\boldsymbol{\Phi}_{\text{GUMI}}(\mathbf{X}) = \left[\boldsymbol{\Psi}_1(\mathbf{X})^t, \boldsymbol{\Psi}_2(\mathbf{X})^t, \ldots, \boldsymbol{\Psi}_Q(\mathbf{X})^t\right]^t \tag{12}$$

The GUMI kernel between a pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ is given by

$$K_{\text{GUMI}}(\mathbf{X}_m, \mathbf{X}_n) = \boldsymbol{\Phi}_{\text{GUMI}}(\mathbf{X}_m)^t \boldsymbol{\Phi}_{\text{GUMI}}(\mathbf{X}_n) \tag{13}$$

2.5 Matching-based dynamic kernels

In this section, we present the matching based approaches to designing dynamic kernels. Let $\mathbf{X}_m = \{\mathbf{x}_{m1}, \mathbf{x}_{m2}, \ldots, \mathbf{x}_{mT_m}\}$ and $\mathbf{X}_n = \{\mathbf{x}_{n1}, \mathbf{x}_{n2}, \ldots, \mathbf{x}_{nT_n}\}$ be the sets of local feature vectors for two examples. The summation kernel (Boughorbel et al. 2004) is computed by matching every local feature vector in $\mathbf{X}_m$ with every local feature vector in $\mathbf{X}_n$ as follows:

$$K_{\text{S}}(\mathbf{X}_m, \mathbf{X}_n) = \sum_{t=1}^{T_m}\sum_{t'=1}^{T_n} k(\mathbf{x}_{mt}, \mathbf{x}_{nt'}) \tag{14}$$

where $k(.,.)$ is a base kernel. The matching kernel (Wallraven et al. 2003) is constructed by considering the closest local feature vector of an example for each local feature vector in the other example as follows:

$$K_{\mathrm{MK}}(\mathbf{X}_m, \mathbf{X}_n) = \sum_{t=1}^{T_m} \max_{t'} k(\mathbf{x}_{mt}, \mathbf{x}_{nt'}) + \sum_{t'=1}^{T_n} \max_{t} k(\mathbf{x}_{mt}, \mathbf{x}_{nt'}) \quad (15)$$

The summation kernel is a Mercer kernel. However, the matching kernel is not proven to be a Mercer kernel (Boughorbel et al. 2004, 2005). Construction of the summation kernel or the matching kernel is computationally intensive. The number of base kernel computations is $T_m * T_n$ for the summation kernel and $2 * T_m * T_n$ for the matching kernel. Hence the computation complexity of both summation kernel and matching kernel is $O(T^2)$, where T is the maximum of set cardinalities T_m and T_n. The summation kernel and the matching kernel give a measure of global similarity between a pair of examples.

An intermediate matching kernel (IMK) (Boughorbel et al. 2005) is constructed by matching the sets of local feature vectors using a set of virtual feature vectors. Let $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_Q\}$ be the set of Q virtual feature vectors. For the qth virtual feature vector $\mathbf{v}_q$, the local feature vectors $\mathbf{x}^*_{mq}$ and $\mathbf{x}^*_{nq}$ in $\mathbf{X}_m$ and $\mathbf{X}_n$ that are closest to $\mathbf{v}_q$ are determined as follows:

$$\mathbf{x}^*_{mq} = \arg\min_{\mathbf{x}\in\mathbf{X}_m} \mathcal{D}(\mathbf{x}, \mathbf{v}_q) \quad \text{and} \quad \mathbf{x}^*_{nq} = \arg\min_{\mathbf{x}\in\mathbf{X}_n} \mathcal{D}(\mathbf{x}, \mathbf{v}_q) \quad (16)$$

where $\mathcal{D}(.,.)$ is a function that measures the distance of a local feature vector to a virtual feature vector. A pair of feature vectors from $\mathbf{X}_m$ and $\mathbf{X}_n$ is selected for each of the virtual feature vectors in $\mathbf{V}$. The selection of the closest local feature vectors for each virtual feature vector involves computation of $T_m + T_n$ distance functions. A base kernel is computed for each of the Q pairs of selected local feature vectors. The IMK is computed as the sum of all the Q base kernel values as follows:

$$K_{\mathrm{IMK}}(\mathbf{X}_m, \mathbf{X}_n) = \sum_{q=1}^{Q} k\left(\mathbf{x}^*_{mq}, \mathbf{x}^*_{nq}\right) \quad (17)$$

The computation of IMK involves a total of $Q * (T_m + T_n)$ computations of distance function $\mathcal{D}$, $Q * (T_m + T_n)$ comparison operations to select the local feature vectors using each of the virtual feature vectors and Q computations of the base kernel. Hence the computation complexity of intermediate matching kernel is $O(QT)$, where T is the maximum of set cardinalities T_m and T_n. When Q is significantly smaller than T_m and T_n, the construction of IMK is computationally less intensive than constructing the summation kernel in (14). In Boughorbel et al. (2005), the set of the centers of clusters formed from the training data of all classes is considered as the set of virtual feature vectors. The local feature vectors $\mathbf{x}^*_{mq}$ and $\mathbf{x}^*_{nq}$ in $\mathbf{X}_m$ and $\mathbf{X}_n$ that are closest to the qth center $\mathbf{v}_q$ are determined as follows:

$$\mathbf{x}^*_{mq} = \arg\min_{\mathbf{x}\in\mathbf{X}_m} \|\mathbf{x} - \mathbf{v}_q\| \quad \text{and} \quad \mathbf{x}^*_{nq} = \arg\min_{\mathbf{x}\in\mathbf{X}_n} \|\mathbf{x} - \mathbf{v}_q\| \quad (18)$$

The Gaussian kernel $k(\mathbf{x}^*_{mq}, \mathbf{x}^*_{nq}) = \exp(-\delta\|\mathbf{x}^*_{mq} - \mathbf{x}^*_{nq}\|^2)$ is used as the base kernel. Here δ is a kernel parameter that is empirically chosen.

In Dileep and Sekhar (2011), the components of a class-independent Gaussian mixture model (CIGMM) are used as the set of virtual feature vectors. This representation for the set of virtual feature vectors makes use of information in the means, covariance matrices and mixture coefficients of components of the CIGMM. The responsibility of the component q for a local feature vector $\mathbf{x}$, $\gamma_q(\mathbf{x})$, is given by

$$\gamma_q(\mathbf{x}) = \frac{w_q \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)}{\sum_{j=1}^{Q} w_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (19)$$

where w_q is the mixture coefficient of the component q. The local feature vectors $\mathbf{x}^*_{mq}$ and $\mathbf{x}^*_{nq}$ in $\mathbf{X}_m$ and $\mathbf{X}_n$ that are closest to the component q are given by

$$\mathbf{x}^*_{mq} = \arg\max_{\mathbf{x}\in\mathbf{X}_m} \gamma_q(\mathbf{x}) \quad \text{and} \quad \mathbf{x}^*_{nq} = \arg\max_{\mathbf{x}\in\mathbf{X}_n} \gamma_q(\mathbf{x}) \quad (20)$$

A pair of local feature vectors from $\mathbf{X}_m$ and $\mathbf{X}_n$ are selected for each of the components. The GMM-based IMK, $K_{\mathrm{GMM\text{-}IMK}}$ is computed as the sum of the values of the base kernel computed for each of the Q pairs of selected local feature vectors as in (17).

In this work we propose to use the pyramid match kernel (PMK) for the speaker recognition task. In the next section, we describe the PMK based SVM system.

3 Pyramid match kernel based SVM

In pyramid match kernel (PMK) (Grauman and Darrell 2007) a set of feature vectors is mapped onto histograms of increasingly larger number of bins to form a pyramid of histograms. A set of feature vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, $\mathbf{x}_t \in \mathbb{R}^d$, is represented as a set of histogram vectors $\mathbf{h}_0(\mathbf{X})^t, \mathbf{h}_1(\mathbf{X})^t, \ldots, \mathbf{h}_J(\mathbf{X})^t$ forming a pyramid like structure with $J + 1$ number of levels. Let $\mathbf{h}_j(\mathbf{X})$ be the histogram vector and Q_j be the number of bins at jth level. For a univariate feature x, a bin corresponds to an interval

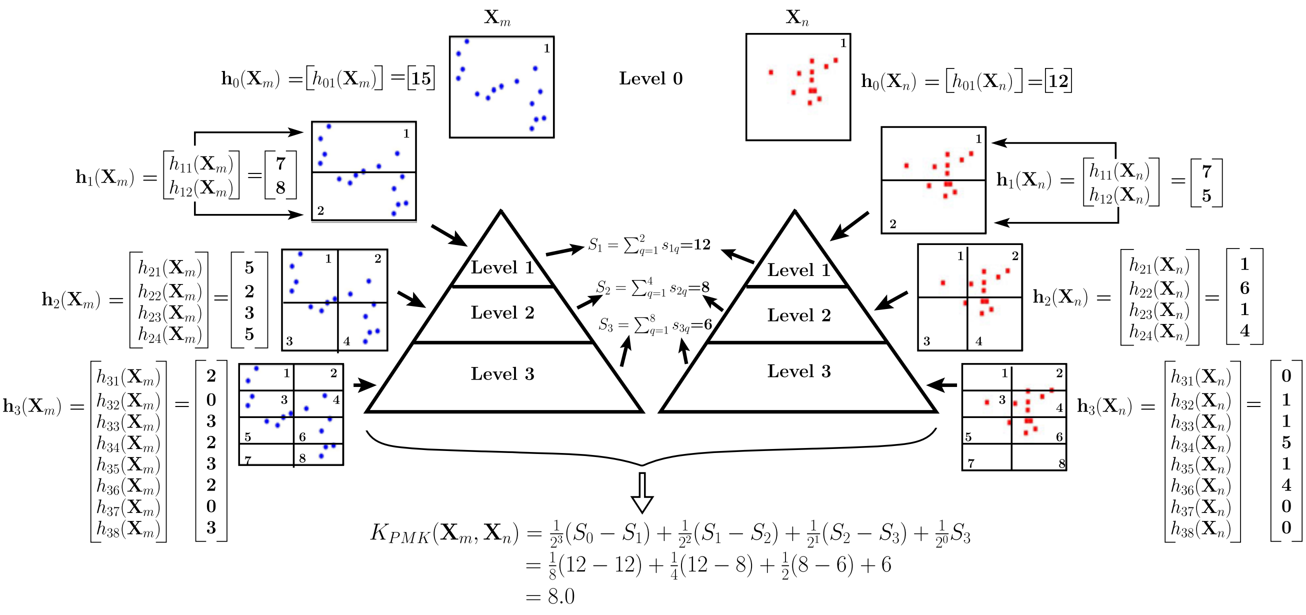

X_m
X_n
Level 0
h_0(X_m) = [h_01(X_m)] = [15]
h_0(X_n) = [h_01(X_n)] = [12]
h_1(X_m) = [h_11(X_m); h_12(X_m)] = [7; 8]
h_1(X_n) = [h_11(X_n); h_12(X_n)] = [7; 5]
h_2(X_m) = [h_21(X_m); h_22(X_m); h_23(X_m); h_24(X_m)] = [5; 2; 3; 5]
h_2(X_n) = [h_21(X_n); h_22(X_n); h_23(X_n); h_24(X_n)] = [1; 6; 1; 4]
h_3(X_m) = [h_31(X_m); … ; h_38(X_m)] = [2; 0; 3; 2; 3; 2; 0; 3]
h_3(X_n) = [h_31(X_n); … ; h_38(X_n)] = [0; 1; 1; 5; 1; 4; 0; 0]
1
2
3
4
5
6
7
8
Level 1
Level 2
Level 3
S_1 = Σ_{q=1}^{2} s_1q = 12
S_2 = Σ_{q=1}^{4} s_2q = 8
S_3 = Σ_{q=1}^{8} s_3q = 6
K_PMK(X_m, X_n) = 1/2^3 (S_0 − S_1) + 1/2^2 (S_1 − S_2) + 1/2^1 (S_2 − S_3) + 1/2^0 S_3
= 1/8 (12 − 12) + 1/4 (12 − 8) + 1/2 (8 − 6) + 6
= 8.0

Fig. 1 Illustration of construction of a pyramid of histograms with grids as bins of the same size and computation of pyramid match kernel between a pair of sets of 2-dimensional feature vectors $\mathbf{X}_m$ and $\mathbf{X}_n$. For the qth bin at the jth level, the number of matches between the pair of examples, s_{jq} is computed using the histogram intersection function as $s_{jq} = \min\,(h_{jq}(\mathbf{X}_m), h_{jq}(\mathbf{X}_n))$

of values that x can take. For a multivariate feature vector $\mathbf{x}$, a bin corresponds to a grid or cluster in the space of $\mathbf{x}$. The levels in the pyramid are indexed from root level ($j = 0$) to the leaves level ($j = J$) and $Q_j < Q_{j+1}$. Let $\mathbf{X}_m = \{\mathbf{x}_{m1}, \mathbf{x}_{m2}, \ldots, \mathbf{x}_{mT_m}\}$ and $\mathbf{X}_n = \{\mathbf{x}_{n1}, \mathbf{x}_{n2}, \ldots, \mathbf{x}_{nT_n}\}$ be the sets of feature vectors for two examples. The histogram vectors $\mathbf{h}_j(\mathbf{X}_m)$ and $\mathbf{h}_j(\mathbf{X}_n)$ at a particular level j for the pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ are compared using the histogram intersection function (Swain and Ballard 1991). Let $h_{jq}(\mathbf{X}_m)$ and $h_{jq}(\mathbf{X}_n)$ be the number of feature vectors from $\mathbf{X}_m$ and $\mathbf{X}_n$ respectively in the qth bin at the level j. The number of matches in the qth bin at the level j is given by histogram intersection function (Swain and Ballard 1991), defined as follows:

$$s_{jq} = \min\big(h_{jq}(\mathbf{X}_m), h_{jq}(\mathbf{X}_n)\big) \tag{21}$$

Total number of matches at level j is obtained as

$$S_j = \sum_{q=1}^{Q_j} s_{jq} \tag{22}$$

The matching is a hierarchical process from the bottom of the pyramid to the top of the pyramid. Feature vectors that are not matched at a lower level of the pyramid have an opportunity to be matched at a higher level of pyramid. In other words, the matches found at a level j include the matches at the level $j+1$. The number of new matches at the level j is calculated by computing the difference between the number of matches at levels j and $j+1$ as $S_j - S_{j+1}$. The number of new matches at each level of the pyramid is weighted according to the number of bins at that level. The pyramid match kernel, which gives the matching score for a pair of examples, is computed as a weighted sum of the number of new matches at different levels of the pyramid. The PMK for $\mathbf{X}_m$ and $\mathbf{X}_n$ is defined as

$$K_{\text{PMK}}(\mathbf{X}_m, \mathbf{X}_n) = \sum_{j=0}^{J-1} w_j(S_j - S_{j+1}) + w_J S_J \tag{23}$$

where w_j is the weight at the level j.

The main issue in the design of PMK is the construction of histograms. In certain datasets, the range of values for each of the features is uniformly the same. Let the range be 0 to D. In such cases, each example can be represented by a histogram with bins of the same size. For example, pixels in a gray scale image may have a fixed range of 0 to 255. For a univariate feature x, a bin corresponds to an interval of values that x can take. For a multivariate feature vector $\mathbf{x}$, a bin corresponds to a grid in the space of $\mathbf{x}$. This can also be seen as dividing the d-dimensional feature space into grids of side d as shown in the Fig. 1. When the value for each of the features is in the range 0 to D, the value of J is computed as $\lceil \log_2 D \rceil + 1$ and 2^j bins are considered at the jth level (Grauman and Darrell 2007). The histogram vector for $\mathbf{X}$ at jth level, $\mathbf{h}_j(\mathbf{X})$ is formed using the 2^j d-dimensional bins and it has the dimension $Q_j = 2^j$. The histogram intersection function is used for matching the histogram vectors at each level for the pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ as in (21) and (22). The weight for the new matches found at each level is chosen as $w_j = \frac{1}{2^{J-j}}$, which is inversely proportional to

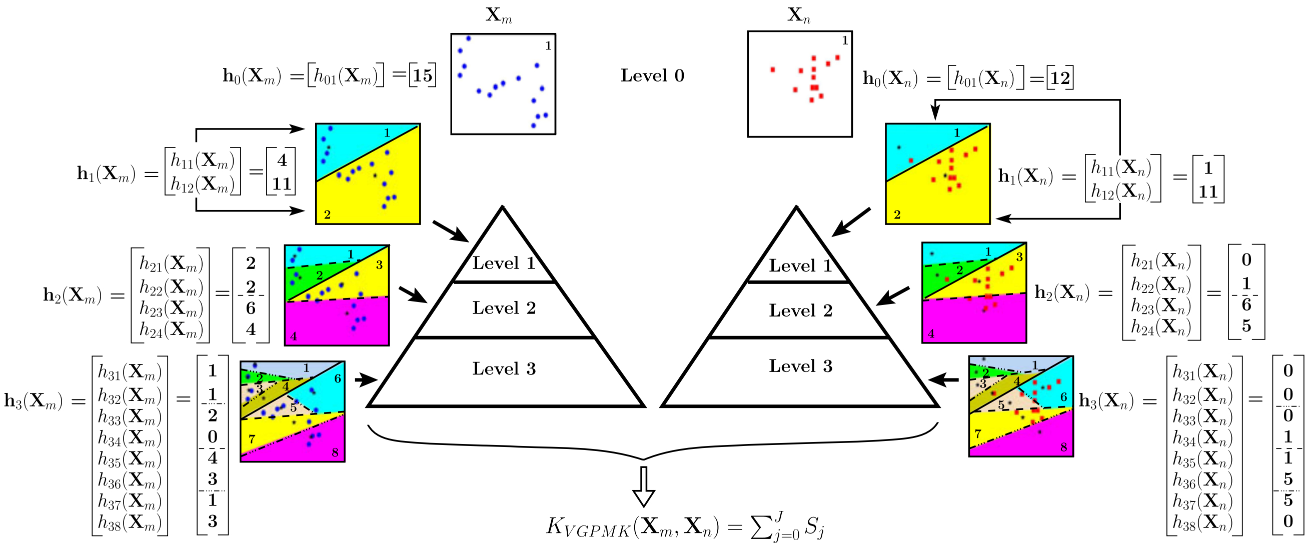

$\mathbf{X}_m$
$\mathbf{X}_n$
Level 0
$\mathbf{h}_0(\mathbf{X}_m) = [h_{01}(\mathbf{X}_m)] = [15]$
$\mathbf{h}_0(\mathbf{X}_n) = [h_{01}(\mathbf{X}_n)] = [12]$
$\mathbf{h}_1(\mathbf{X}_m) = \begin{bmatrix} h_{11}(\mathbf{X}_m) \\ h_{12}(\mathbf{X}_m) \end{bmatrix} = \begin{bmatrix} 4 \\ 11 \end{bmatrix}$
$\mathbf{h}_1(\mathbf{X}_n) = \begin{bmatrix} h_{11}(\mathbf{X}_n) \\ h_{12}(\mathbf{X}_n) \end{bmatrix} = \begin{bmatrix} 1 \\ 11 \end{bmatrix}$
$\mathbf{h}_2(\mathbf{X}_m) = \begin{bmatrix} h_{21}(\mathbf{X}_m) \\ h_{22}(\mathbf{X}_m) \\ h_{23}(\mathbf{X}_m) \\ h_{24}(\mathbf{X}_m) \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 6 \\ 4 \end{bmatrix}$
$\mathbf{h}_2(\mathbf{X}_n) = \begin{bmatrix} h_{21}(\mathbf{X}_n) \\ h_{22}(\mathbf{X}_n) \\ h_{23}(\mathbf{X}_n) \\ h_{24}(\mathbf{X}_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 6 \\ 5 \end{bmatrix}$
$\mathbf{h}_3(\mathbf{X}_m) = \begin{bmatrix} h_{31}(\mathbf{X}_m) \\ h_{32}(\mathbf{X}_m) \\ h_{33}(\mathbf{X}_m) \\ h_{34}(\mathbf{X}_m) \\ h_{35}(\mathbf{X}_m) \\ h_{36}(\mathbf{X}_m) \\ h_{37}(\mathbf{X}_m) \\ h_{38}(\mathbf{X}_m) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 0 \\ 4 \\ 3 \\ 1 \\ 3 \end{bmatrix}$
$\mathbf{h}_3(\mathbf{X}_n) = \begin{bmatrix} h_{31}(\mathbf{X}_n) \\ h_{32}(\mathbf{X}_n) \\ h_{33}(\mathbf{X}_n) \\ h_{34}(\mathbf{X}_n) \\ h_{35}(\mathbf{X}_n) \\ h_{36}(\mathbf{X}_n) \\ h_{37}(\mathbf{X}_n) \\ h_{38}(\mathbf{X}_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 5 \\ 5 \\ 0 \end{bmatrix}$
Level 1
Level 2
Level 3
Level 1
Level 2
Level 3
1
2
3
4
5
6
7
8
$K_{VGPMK}(\mathbf{X}_m, \mathbf{X}_n) = \sum_{j=0}^{J} S_j$

Fig. 2 Illustration of construction of the vocabulary-guided pyramid for a pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ represented as sets of 2-dimensional feature vectors. The value of J and b are 3 and 2 respectively. The centers of clusters are denoted by the symbol $*$. Each cluster at a level is further divided into two clusters at the next level

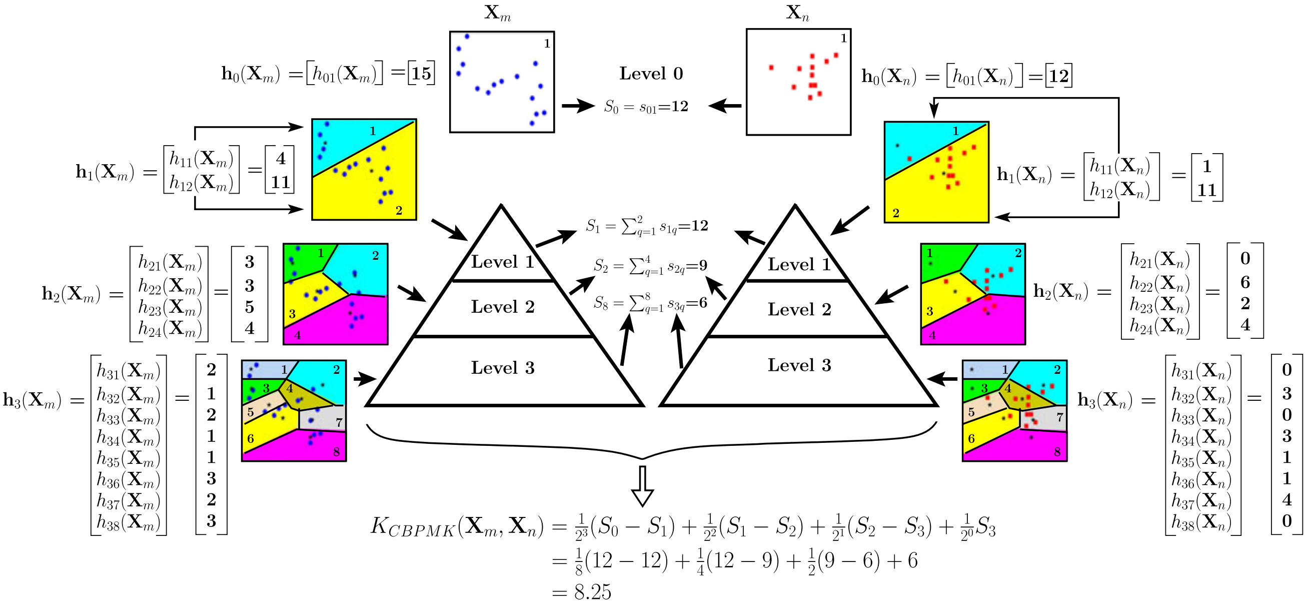
the size of the histogram bin at that level. The PMK for $\mathbf{X}_m$ and $\mathbf{X}_n$ using the grids as bins is defined as

$$K_{\text{PMK}}(\mathbf{X}_m, \mathbf{X}_n) = \sum_{j=0}^{J-1} \frac{1}{2^{J-j}}(S_j - S_{j+1}) + S_J \tag{24}$$

The process of constructing the pyramid match kernel between a pair of sets of 2-dimensional feature vectors is shown in Fig. 1. For the illustration, the value of J is chosen as 3. The bins correspond to the uniformly shaped and sized grids. The feature vectors are placed into bins using values of features. The pyramid match kernel is computed using a total of $J * (T_m + T_n)$ computations of number of matches and J computations of number of new matches. Hence the computation complexity of PMK using grids as bins is $O(JT)$, where T is the maximum of set cardinalities T_m and T_n. It is computationally less intensive compared to IMK as $J \ll Q$.

However, it is shown in Grauman (2006) that the expected approximation error bound increases with the feature vector dimension and that the computation of histogram with d-dimensional bins becomes difficult. In Grauman (2006), a vocabulary-guided pyramid match is introduced for sets of feature vectors with a large dimension. In Grauman (2006), the given feature space is partitioned into non-uniformly shaped regions using the k-means clustering method with the Euclidean distance. At the top level ($j = 0$), the feature vectors from all the training examples are considered to fall in a single cluster. In the next level ($j = 1$), the feature vectors are clustered into b groups and the cluster centers form a vocabulary at that level. The cluster membership for a feature vector is determined using the Euclidean distances of the feature vector to the b cluster centers. A feature vector is assigned to a cluster for which the Euclidean distance is minimum. In the next level, feature vectors in each of these b clusters are further clustered into b clusters. This leads to b^j number of clusters (bins) at level j. The values of J and the branching factor b are chosen empirically. For an example represented as a set of feature vectors $\mathbf{X}$, the histogram vector at the level j, $\mathbf{h}_j(\mathbf{X})$, is obtained by counting the number of feature vectors of $\mathbf{X}$ assigned to each of the b^j clusters. Thus, $\mathbf{h}_j(\mathbf{X})$ is considered as a histogram vector with $Q_j = b^j$ number of non-uniform bins at the jth level. Let $h_{jq}(\mathbf{X})$ be the qth entry in the jth level histogram vector $\mathbf{h}_j(\mathbf{X})$ indicating the number of feature vectors assigned to the qth cluster at the jth level. The pyramid can be represented as a tree with $J + 1$ levels and b^j nodes at level j. Every non-leaf node in the tree has b children. Let $c_i(h_{jq}(\mathbf{X}))$ denote the number of feature vectors in the cluster corresponding to the ith child node of the node corresponding to the qth cluster at level j. For a pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ the multiple resolution vocabulary-guided pyramid is constructed for matching these examples. The process of constructing the vocabulary-guided pyramid for a pair of examples represented as sets of 2-dimensional feature vectors is shown in Fig. 2. The number of new matches at level j of the pyramid is computed as the weighted sum of the new matches within the nodes at the level and is given as

$$S_j = \sum_{q=1}^{Q_j} w_{jq} \Bigg[\min\big(h_{jq}(\mathbf{X}_m), h_{jq}(\mathbf{X}_n)\big) - \sum_{i=1}^{b} \min\Big(c_i\big(h_{jq}(\mathbf{X}_m)\big), c_i\big(h_{jq}(\mathbf{X}_n)\big)\Big) \Bigg] \tag{25}$$



$\mathbf{X}_m$
$\mathbf{X}_n$
$\mathbf{h}_0(\mathbf{X}_m) = [h_{01}(\mathbf{X}_m)] = [\mathbf{15}]$
$\mathbf{h}_0(\mathbf{X}_n) = [h_{01}(\mathbf{X}_n)] = [\mathbf{12}]$
Level 0
$S_0 = s_{01}$=12
$\mathbf{h}_1(\mathbf{X}_m) = \begin{bmatrix} h_{11}(\mathbf{X}_m) \\ h_{12}(\mathbf{X}_m) \end{bmatrix} = \begin{bmatrix} \mathbf{4} \\ \mathbf{11} \end{bmatrix}$
$\mathbf{h}_1(\mathbf{X}_n) = \begin{bmatrix} h_{11}(\mathbf{X}_n) \\ h_{12}(\mathbf{X}_n) \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{11} \end{bmatrix}$
$\mathbf{h}_2(\mathbf{X}_m) = \begin{bmatrix} h_{21}(\mathbf{X}_m) \\ h_{22}(\mathbf{X}_m) \\ h_{23}(\mathbf{X}_m) \\ h_{24}(\mathbf{X}_m) \end{bmatrix} = \begin{bmatrix} \mathbf{3} \\ \mathbf{3} \\ \mathbf{5} \\ \mathbf{4} \end{bmatrix}$
$\mathbf{h}_2(\mathbf{X}_n) = \begin{bmatrix} h_{21}(\mathbf{X}_n) \\ h_{22}(\mathbf{X}_n) \\ h_{23}(\mathbf{X}_n) \\ h_{24}(\mathbf{X}_n) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{6} \\ \mathbf{2} \\ \mathbf{4} \end{bmatrix}$
$\mathbf{h}_3(\mathbf{X}_m) = \begin{bmatrix} h_{31}(\mathbf{X}_m) \\ h_{32}(\mathbf{X}_m) \\ h_{33}(\mathbf{X}_m) \\ h_{34}(\mathbf{X}_m) \\ h_{35}(\mathbf{X}_m) \\ h_{36}(\mathbf{X}_m) \\ h_{37}(\mathbf{X}_m) \\ h_{38}(\mathbf{X}_m) \end{bmatrix} = \begin{bmatrix} \mathbf{2} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{3} \\ \mathbf{2} \\ \mathbf{3} \end{bmatrix}$
$\mathbf{h}_3(\mathbf{X}_n) = \begin{bmatrix} h_{31}(\mathbf{X}_n) \\ h_{32}(\mathbf{X}_n) \\ h_{33}(\mathbf{X}_n) \\ h_{34}(\mathbf{X}_n) \\ h_{35}(\mathbf{X}_n) \\ h_{36}(\mathbf{X}_n) \\ h_{37}(\mathbf{X}_n) \\ h_{38}(\mathbf{X}_n) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{3} \\ \mathbf{0} \\ \mathbf{3} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{4} \\ \mathbf{0} \end{bmatrix}$
1
2
3
4
5
6
7
8
Level 1
Level 2
Level 3
$S_1 = \sum_{q=1}^{2} s_{1q}$=12
$S_2 = \sum_{q=1}^{4} s_{2q}$=9
$S_8 = \sum_{q=1}^{8} s_{3q}$=6
$K_{CBPMK}(\mathbf{X}_m, \mathbf{X}_n) = \frac{1}{2^3}(S_0 - S_1) + \frac{1}{2^2}(S_1 - S_2) + \frac{1}{2^1}(S_2 - S_3) + \frac{1}{2^0}S_3$
$= \frac{1}{8}(12 - 12) + \frac{1}{4}(12 - 9) + \frac{1}{2}(9 - 6) + 6$
$= 8.25$

Fig. 3 Illustration of construction of the pyramid of codebooks and computation of codebook-based pyramid match kernel for a pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$. The centers of the clusters are denoted using the symbol $*$. The values of j and b are 3 and 2 respectively. The jth level contains 2^j number of clusters formed using the k-means clustering technique on the feature vectors of the training examples of all the classes. For the qth bin at the jth level, the number of matches between the pair of examples, s_{jq} is computed using the histogram intersection function as $s_{jq} = \min\,(h_{jq}(\mathbf{X}_m), h_{jq}(\mathbf{X}_n))$

The diameter of the smallest enclosing hypersphere for the qth cluster at the jth level is considered as the weight w_{jq}. At the leaf level ($j = J$) the number of new matches is computed as

$$S_J = \sum_{q=1}^{Q_J} w_{Jq} \min\left(h_{Jq}(\mathbf{X}_m), h_{Jq}(\mathbf{X}_n)\right) \qquad (26)$$

The vocabulary-guided pyramid match kernel (VGPMK) between a pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ is computed as a weighted sum of these new matches and is given as

$$K_{\text{VGPMK}}(\mathbf{X}_m, \mathbf{X}_n) = \sum_{j=0}^{J} S_j \qquad (27)$$

As the values of T_m and T_n vary for different pairs of examples, the kernel values are normalized. The normalized vocabulary-guided PMK for $\mathbf{X}_m$ and $\mathbf{X}_n$ is given as

$$\widehat{K}_{\text{VGPMK}}(\mathbf{X}_m, \mathbf{X}_n) = \frac{K_{\text{VGPMK}}(\mathbf{X}_m, \mathbf{X}_n)}{\sqrt{K_{\text{VGPMK}}(\mathbf{X}_m, \mathbf{X}_m)}\sqrt{K_{\text{VGPMK}}(\mathbf{X}_n, \mathbf{X}_n)}} \qquad (28)$$

In VGPMK, irrespective of the size of the cluster, each cluster at a level is divided into b clusters in the next level resulting in b^j clusters at each level j. This may lead to clustering the feature vectors in a small cluster also into b very small clusters. This phenomenon can be avoided by clustering the feature vectors of all the training examples into b^j clusters at level j using the k-means clustering. The centers of clusters at each level are used to build a codebook. The codebooks at different levels are used for matching a pair of examples. In the next subsection we propose codebook-based PMK.

3.1 Codebook-based pyramid match kernel

The k-means clustering technique is used at each level to cluster the feature vectors of all the training examples. At the top level ($j = 0$), the feature vectors of all the training examples are considered to be in a single cluster. At the next level ($j = 1$), all the feature vectors are clustered into b groups. Similarly, for any level j, all the feature vectors are grouped into b^j number of clusters and the cluster centers are used to build a codebook at that level. An example $\mathbf{X}$ is represented as a histogram vector $\mathbf{h}_j(\mathbf{X})$ at jth level of pyramid and has the dimension $Q_j = b^j$. The values of J and b are chosen empirically. For a pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ the multiple resolution codebook-based pyramid is constructed for matching these examples. The process of constructing the codebook-based pyramid match kernel for a pair of examples represented as sets of 2-dimensional feature vectors is shown in Fig. 3. The histogram intersection function is used for matching the histogram vectors at each level for the pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ as in (21) and (22). The number of new matches at the level j is calculated by computing the difference between the number of matches at levels j and $j+1$ as $S_j - S_{j+1}$. The codebook-based pyramid match kernel (CBPMK) for the pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ is computed as a weighted sum of the number of new matches at different levels of the pyramid. The CBPMK for $\mathbf{X}_m$ and $\mathbf{X}_n$, $K_{\text{CBPMK}}(\mathbf{X}_m, \mathbf{X}_n)$ is computed as in (24). The weight for the number of new matches found at each level is considered as $w_j = \frac{1}{2^{J-j}}$. The normalized codebook-based PMK for $\mathbf{X}_m$ and $\mathbf{X}_n$ is given as

$$\widehat{K}_{\text{CBPMK}}(\mathbf{X}_m, \mathbf{X}_n) = \frac{K_{\text{CBPMK}}(\mathbf{X}_m, \mathbf{X}_n)}{\sqrt{K_{\text{CBPMK}}(\mathbf{X}_m, \mathbf{X}_m)}\sqrt{K_{\text{CBPMK}}(\mathbf{X}_n, \mathbf{X}_n)}} \tag{29}$$

The construction of VGPMK or CBPMK is computationally intensive compared to the PMK that uses grids as bins as described in Grauman and Darrell (2007). The PMK with uniform bins in Grauman and Darrell (2007) does not require computing measure of closeness to place the feature vectors into bins (clusters). The computation of VGPMK or CBPMK involves a total of $\sum_{j=0}^{J} Q_j * (T_m + T_n)$ computations for measuring the closeness to place the feature vectors into bins (clusters) at each level, $\sum_{j=0}^{J} Q_j * (T_m + T_n)$ computations for sorting the measures of closeness of each feature vector to centers of clusters at each level, $\sum_{j=0}^{J} Q_j * (T_m + T_n)$ computations for determining the number of matches and J computations for determining the number of new matches. Hence the computation complexity of VGPMK or CBPMK is $O(JQT)$, where Q is the maximum of all the Q_js and T is the maximum of the set cardinalities T_m and T_n. The VGPMK and CBPMK are slightly more computationally intensive compared to IMK.

The key issue in the design of PMK is the choice of the technique for constructing the pyramid of histograms. The k-means clustering method makes use of information about the centers of clusters and the distances of a feature vector to the centers of clusters to assign that feature vector to one of the clusters. A better pyramid of histograms can be obtained by considering the clustering method that considers additional information like the widths of the clusters and the sizes of the clusters along with the centers of the clusters. Moreover, the construction of VGPMK or CBPMK involves hard clustering. A better PMK is constructed by using soft clustering. In the next subsection, we propose the Gaussian mixture model (GMM) based pyramid match kernel. The Gaussian mixture model uses the information about the widths and the sizes of the clusters along with the centers of the clusters for soft clustering of feature vectors.

3.2 GMM-based pyramid match kernel

In this approach, we propose to use a class-independent GMM built using the training data of all the classes for forming the clusters at each level of pyramid. The GMMs make use of information about mean vectors of components, covariance matrices of components and mixture coefficients. The additional information in the form of covariance matrices and mixture coefficients is expected to give better clusters at each level of the pyramid as compared to the clusters obtained using the k-means clustering technique. A class-independent GMM (CIGMM) is a large GMM of Q components built using the feature vectors in the training examples of all the classes. The feature vectors from the pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ that are closest to the component q are assigned to that cluster. The responsibility term, i.e., the probability of a feature vector being generated by a component, is considered as a measure of closeness of a feature vector to the component q. The responsibility of the component q of a CIGMM for a feature vector $\mathbf{x}_t$, γ_{tq}, is computed using (19). For a set of feature vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, the effective number of feature vectors h_q assigned to a component q is given by

$$h_q = \sum_{t=1}^{T} \gamma_{tq} \tag{30}$$

Let $J + 1$ be the number of levels in the multi-resolution histogram pyramid constructed using the GMMs. At the top level ($j = 0$), the feature vectors of all the training example are considered to be assigned to a single Gaussian component. At any level j, a CIGMM of b^j components is built using the training data of all the classes. For a pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ the multiple resolution GMM-based pyramid is constructed for matching the examples. The process of constructing the GMM-based pyramid match kernel for a pair of examples represented as sets of 2-dimensional feature vectors is shown in Fig. 4. Let $\mathbf{h}_j(\mathbf{X}_m)$ and $\mathbf{h}_j(\mathbf{X}_n)$ be the histogram vectors at the jth level formed using b^j components of GMM at that level. The qth entry in the histogram vectors, $h_{jq}(\mathbf{X}_m)$ and $h_{jq}(\mathbf{X}_n)$ correspond to the effective number of feature vectors from $\mathbf{X}_m$ and $\mathbf{X}_n$ respectively assigned to the qth component at jth level. The effective number of matches in the qth component of the GMM at the level j between a pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ is given as in (21) and (22). The effective number of new matches at the level j is calculated by computing the difference between the effective number of matches at levels j and $j + 1$ as $S_j - S_{j+1}$. The GMM-based pyramid match kernel (GMMPMK) for the pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$ is computed as a weighted sum of the effective number of new matches at different levels of the pyramid. The GMMPMK for $\mathbf{X}_m$ and $\mathbf{X}_n$, $K_{\text{GMMPMK}}(\mathbf{X}_m, \mathbf{X}_n)$ is computed as in (24). The weight at the level j is considered as $w_j = \frac{1}{2^{J-j}}$. The normalized GMM-based PMK for $\mathbf{X}_m$ and $\mathbf{X}_n$ is given as

$$\widehat{K}_{\text{GMMPMK}}(\mathbf{X}_m, \mathbf{X}_n) = \frac{K_{\text{GMMPMK}}(\mathbf{X}_m, \mathbf{X}_n)}{\sqrt{K_{\text{GMMPMK}}(\mathbf{X}_m, \mathbf{X}_m)}\sqrt{K_{\text{GMMPMK}}(\mathbf{X}_n, \mathbf{X}_n)}} \tag{31}$$

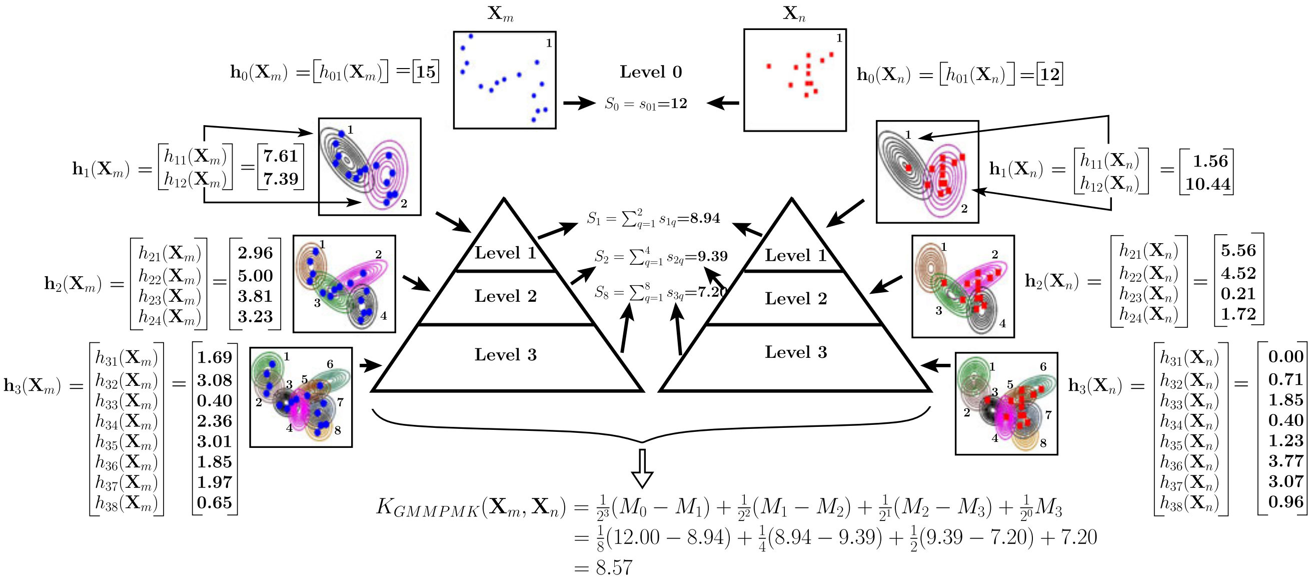

$\mathbf{X}_m$
$\mathbf{X}_n$
$\mathbf{h}_0(\mathbf{X}_m) = [h_{01}(\mathbf{X}_m)] = [15]$
Level 0
$S_0 = s_{01} = 12$
$\mathbf{h}_0(\mathbf{X}_n) = [h_{01}(\mathbf{X}_n)] = [12]$
$\mathbf{h}_1(\mathbf{X}_m) = \begin{bmatrix} h_{11}(\mathbf{X}_m) \\ h_{12}(\mathbf{X}_m) \end{bmatrix} = \begin{bmatrix} 7.61 \\ 7.39 \end{bmatrix}$
$\mathbf{h}_1(\mathbf{X}_n) = \begin{bmatrix} h_{11}(\mathbf{X}_n) \\ h_{12}(\mathbf{X}_n) \end{bmatrix} = \begin{bmatrix} 1.56 \\ 10.44 \end{bmatrix}$
$\mathbf{h}_2(\mathbf{X}_m) = \begin{bmatrix} h_{21}(\mathbf{X}_m) \\ h_{22}(\mathbf{X}_m) \\ h_{23}(\mathbf{X}_m) \\ h_{24}(\mathbf{X}_m) \end{bmatrix} = \begin{bmatrix} 2.96 \\ 5.00 \\ 3.81 \\ 3.23 \end{bmatrix}$
$\mathbf{h}_2(\mathbf{X}_n) = \begin{bmatrix} h_{21}(\mathbf{X}_n) \\ h_{22}(\mathbf{X}_n) \\ h_{23}(\mathbf{X}_n) \\ h_{24}(\mathbf{X}_n) \end{bmatrix} = \begin{bmatrix} 5.56 \\ 4.52 \\ 0.21 \\ 1.72 \end{bmatrix}$
$\mathbf{h}_3(\mathbf{X}_m) = \begin{bmatrix} h_{31}(\mathbf{X}_m) \\ h_{32}(\mathbf{X}_m) \\ h_{33}(\mathbf{X}_m) \\ h_{34}(\mathbf{X}_m) \\ h_{35}(\mathbf{X}_m) \\ h_{36}(\mathbf{X}_m) \\ h_{37}(\mathbf{X}_m) \\ h_{38}(\mathbf{X}_m) \end{bmatrix} = \begin{bmatrix} 1.69 \\ 3.08 \\ 0.40 \\ 2.36 \\ 3.01 \\ 1.85 \\ 1.97 \\ 0.65 \end{bmatrix}$
$\mathbf{h}_3(\mathbf{X}_n) = \begin{bmatrix} h_{31}(\mathbf{X}_n) \\ h_{32}(\mathbf{X}_n) \\ h_{33}(\mathbf{X}_n) \\ h_{34}(\mathbf{X}_n) \\ h_{35}(\mathbf{X}_n) \\ h_{36}(\mathbf{X}_n) \\ h_{37}(\mathbf{X}_n) \\ h_{38}(\mathbf{X}_n) \end{bmatrix} = \begin{bmatrix} 0.00 \\ 0.71 \\ 1.85 \\ 0.40 \\ 1.23 \\ 3.77 \\ 3.07 \\ 0.96 \end{bmatrix}$
Level 1
Level 2
Level 3
Level 1
Level 2
Level 3
$S_1 = \sum_{q=1}^{2} s_{1q} = 8.94$
$S_2 = \sum_{q=1}^{4} s_{2q} = 9.39$
$S_8 = \sum_{q=1}^{8} s_{3q} = 7.20$
1
2
3
4
5
6
7
8
$K_{GMMPMK}(\mathbf{X}_m, \mathbf{X}_n) = \frac{1}{2^3}(M_0 - M_1) + \frac{1}{2^2}(M_1 - M_2) + \frac{1}{2^1}(M_2 - M_3) + \frac{1}{2^0}M_3$
$= \frac{1}{8}(12.00 - 8.94) + \frac{1}{4}(8.94 - 9.39) + \frac{1}{2}(9.39 - 7.20) + 7.20$
$= 8.57$

Fig. 4 Illustration of construction of the pyramid of histogram where each bin in histogram corresponds to a GMM component and computation of GMM-based pyramid match kernel for a pair of examples $\mathbf{X}_m$ and $\mathbf{X}_n$. The values of j and b are 3 and 2 respectively. The jth level contains 2^j number of components formed using the CIGMM on the feature vectors of the training examples of all the classes. For the qth bin at the jth level, the effective number of matches between the pair of examples, s_{jq} is computed using the histogram intersection function as $s_{jq} = \min\,(h_{jq}(\mathbf{X}_m), h_{jq}(\mathbf{X}_n))$

The construction of a pyramid of GMMs is a one time process. The computation of GMMPMK involves a total of $\sum_{j=0}^{J} Q_j * (T_m + T_n)$ computations for measuring the closeness of the feature vectors to the components at different levels, $\sum_{j=0}^{J} Q_j * (T_m + T_n)$ computations for determining the effective number of feature vectors assigned to the components, $\sum_{j=0}^{J} Q_j * (T_m + T_n)$ computations for determining the effective number of matches and J computations for determining the effective number of new matches. Hence the computation complexity of GMMPMK is also $O(JQT)$, where Q is the maximum of all the Q_js and T is the maximum of the set cardinalities T_m and T_n.

In the next section we present our studies on speaker identification using the PMK based SVMs. We compare its performance with that of the MLGMM-based system, the adapted GMM-based system, and the SVM based systems with dynamic kernels such as GLDS kernel, PSK, GMMSV kernel, GUMI kernel, summation kernel and IMK.

4 Studies on speaker identification and verification

In this section, we present our studies on speaker identification and verification. We first describe the implementation details of studies on speaker identification and verification. Then we present the speaker recognition accuracy obtained using GMM based systems and SVM based systems.

4.1 Dataset and features used

We performed experiments on the 2002 and 2003 NIST speaker recognition (SRE) corpora (NIST 2002, 2003). We considered the 122 male speakers that are common to the 2002 and 2003 NIST SRE corpora. Training data for a speaker includes a total of about 3 minutes of speech from the single conversations in the training set of 2002 and 2003 NIST SRE corpora. The test data from the 2003 NIST SRE corpus is used for testing the speaker recognition systems. Each test utterance is around 30 seconds long. The silence portions in the speech utterances are removed. Each utterance in the training and test sets is divided into segments of around 5 seconds. Each speech segment is considered as an example. This leads to a total of 3617 training examples with each speaker class having about 30 examples. The test set includes a total of 3044 examples. A frame size of 20 ms and a shift of 10 ms are used for feature extraction from the speech signal of an example. Every frame is represented using a 39-dimensional feature vector consisting of 12 Mel frequency cepstral coefficients (MFCC), log energy, and their delta and acceleration coefficients. Each of the training and test examples is represented by a set of about 500 local feature vectors. The performance of speaker verification task presented in this section is the equal error rate (EER) obtained for 3044 test examples. The speaker identification accuracy presented in this section is the classification accuracy obtained for 3044 test examples. The classification accuracy gives the percentage of test examples that are correctly

Table 1 Comparison of classification accuracy (in %), estimated at 95 % confidence interval, obtained by the MLGMM based system and the adapted GMM based system for speaker identification task

Model	Number of components (Q)	Classification accuracy (in %)
MLGMM	32	75.81 ± 1.52
	64	76.50 ± 1.51
	128	71.26 ± 1.61
Adapted GMM	1024	83.08 ± 1.33

classified by the classifier. In the context of speaker identification, the classification accuracy indicates the speaker identification rate. In order to ascertain the statistical importance of the result, the classification accuracy is presented along with the 95 % confidence interval. A simple asymptotic method (Wald method) (Newcombe 1998) is employed to estimate the 95 % confidence interval of the classification accuracy. The confidence interval (CI) of classification accuracy is computed as

$$CI = z\sqrt{\frac{\alpha(1-\alpha)}{L_{test}}} \tag{32}$$

where α is the accuracy in decimals, and L_{test} is the number of test examples. Here z is the $(1-\alpha/2)$ point of the standard normal distribution associated with a two-tailed probability α. For 95 % confidence interval, z takes the value of 1.96.

In this study, we compare the accuracy of speaker recognition systems built using GMMs, adapted GMMs, the state-of-the-art dynamic kernel based SVMs, and the IMK and the PMK based SVMs.

4.2 Studies on speaker identification using GMM and SVM based classifiers

For the GMM based systems, we consider diagonal covariance matrices. We perform a line search to select the number of components in GMMs and estimate the parameters of the model using the maximum likelihood (ML) method. The best performance of the MLGMM based system is obtained for 64 components in the GMM for each speaker. The adapted GMM based system uses an UBM with 1024 components. The adapted GMMs are built by adapting the means, variances, and mixture coefficients with a relevance factor of 16 (Reynolds et al. 2000). The classification accuracies on test data for the MLGMM based system and the adapted GMM based system are given in Table 1. It is seen that the adapted GMM based system gives a better performance than the GMM based system.

Next we study the performance of the SVM based approaches to speaker recognition. We consider the GLDS kernel based SVM, PSK based SVM, GMMSV kernel based SVM, GUMI kernel based SVM, IMK based SVM and PMK based SVM for building the speaker recognition systems. The LIBSVM (Chang and Lin 2011) tool is used for building the SVM classifiers. The one-against-the-rest approach is used to build the 122-class speaker recognition systems. The value of trade-off parameter, C in SVM is chosen empirically as 10. The GLDS kernel for a pair of examples is constructed using a degree of 2 and 3 for the polynomial kernel. The classification accuracy, estimated at 95 % confidence interval, is 76.77 ± 1.5 % for degree 2 and 78.62 ± 1.46 % for degree 3. This performance is better than that of the MLGMM-based system using 64 components. For the PSK based SVM, we considered UBMs with 512, and 1024 components. We adapt the UBM with the data of each speaker to get the speaker-dependent GMM. Table 2 presents the classification accuracy for the PSK based SVM. It is seen that the PSK constructed using the 1024 components gives the best performance.

Table 2 Classification accuracy (in %), estimated at 95 % confidence interval, and the average number of support vectors for the PSK based SVM classifiers for speaker identification

Number of components (Q)	Classification accuracy (in %)	Average number of support vectors
512	84.32 ± 1.29	176
1024	86.18 ± 1.23	182

Table 3 Classification accuracy (in %), estimated at 95 % confidence interval, and average number of support vectors for the GMM supervector (GMMSV) kernel based SVM and GMM-UBM mean interval (GUMI) kernel based SVM classifiers for speaker identification

Number of components (Q)	Classification accuracy (in %)		Average number of support vectors	
	GMMSV	GUMI	GMMSV	GUMI
512	87.93 ± 1.16	90.31 ± 1.05	396	387
1024	86.23 ± 1.22	89.91 ± 1.07	405	396

For the GMM supervector (GMMSV) kernel based SVM and the GMM-UBM mean interval (GUMI) kernel based SVM, we considered UBMs with 512 and 1024 components. Table 3 presents the classification accuracy, estimated at 95 % confidence interval, for GMMSV kernel and GUMI kernel based SVMs. It is seen that the GMMSV kernel and GUMI kernel constructed using the 512 components give the best performance. This performance is significantly better than that of the PSK based SVM (Table 2). The performance of GUMI kernel based SVM is significantly better compared to that of the GMMSV kernel based SVM and the adapted GMM based system (Table 1).

The IMK based SVMs are built using a value of 128 or 256 or 512 for Q corresponding to the size of the set of virtual feature vectors. The classification accuracy, estimated at 95 % confidence intervals, for the IMK constructed using the components of CIGMM and the IMK constructed using the set of centers of clusters is given in Table 4, for different values of Q. It is seen that the IMK constructed using the CIGMM with 512 components as the set of virtual feature vectors gives the best performance of 88.12 ± 1.15 % (Table 1). It is seen that the average number of support vectors for the SVMs using IMK is in the range of 225 to 300.

Next we study the performance of the pyramid match kernel (PMK) based SVM approaches to speaker identification. As the dimension of the feature vector is large ($d = 39$), the computation of histogram with 39-dimensional grids as bins is difficult. Hence we did not consider the PMK that uses grids as bins. The SVMs using the vocabulary-guided PMK (VGPMK), codebook-based PMK (CBPMK) and GMM-based PMK (GMMPMK) are built using different values for J corresponding to the number of levels in the multi-resolution histogram pyramid and the branching factor b. In VGPMK, each cluster at a level is further clustered into b clusters in the next level resulting in a total of b^j clusters at any level j. However, the proposed CBPMK and the GMMPMK, consider b^j number of clusters constructed from the feature vectors of all the training examples at each level j. In GMMPMK, the CIGMM at each level is built using centers of clusters used in CBPMK at that level as the starting point. The classification accuracies, estimated at 95 % confidence interval, for the VGPMK, the CBPMK and the GMMPMK based SVMs for speaker identification are given in Table 5, for different values of J and b. It is seen that the GMMPMK constructed using the CIGMMs for $J = 6$ and $b = 4$ gives the best performance of 90.26 %. It is also seen that the GMMPMK based SVM performs significantly better than the CBPMK and the VGPMK based SVMs. It is also seen that the performance of the CBPMK based SVM is marginally better compared to that of the VGPMK based SVM. It is observed that the accuracies of the PMK based SVMs constructed using the pyramids with $J = 12$ and $b = 2$, and $J = 6$ and $b = 4$ are close. However, the PMK constructed using pyramid with $J = 6$ and $b = 4$ is computationally less intensive than the PMK constructed using pyramid with $J = 12$ and $b = 2$ as the number of levels is less. The performance of the VGPMK, the CBPMK and the GMMPMK based SVMs is better than that of the adapted GMM based system (Table 1). It is also seen that the performance of the GMMPMK is close to the performance of the state-of-the-art kernel, GUMI kernel, given in Table 3.

Table 6 compares the speaker identification accuracies obtained using the GMM-based classifiers, SVM-based classifiers using generalized linear discriminant sequence (GLDS) kernel, probabilistic sequence kernel (PSK), GMM supervector (GMMSV) kernel, GMM-UBM mean inter-

Table 4 Classification accuracy (in %), estimated at 95 % confidence interval, and average number of support vectors for the IMK based SVM classifiers for speaker identification

Set of virtual feature vectors	Number of virtual feature vectors (Q)	Classification accuracy (in %)	Average number of support vectors
Components of UBM	128	84.30 ± 1.29	279
	256	86.21 ± 1.22	226
	512	88.12 ± 1.15	254
Center of clusters	128	76.52 ± 1.51	263
	256	78.54 ± 1.46	290
	512	79.38 ± 1.44	287

Table 5 Classification accuracy (in %), estimated at 95 % confidence interval, of the SVM-based classifiers with vocabulary-guided PMK (VGPMK), codebook-based PMK (CBPMK) and GMM-based PMK (GMMPMK) for speaker identification for the for different values of J and b. Here, L_c indicate the number of clusters at leaf level

J	b	L_c	Classification accuracy (in %)			Average number of support vectors		
			VGPMK	CBPMK	GMMPMK	VGPMK	CBPMK	GMMPMK
8	2	256	75.66 ± 1.52	77.23 ± 1.49	81.79 ± 1.37	160	159	165
9	2	512	78.42 ± 1.46	80.58 ± 1.41	86.05 ± 1.23	180	180	183
10	2	1024	80.72 ± 1.40	81.57 ± 1.38	88.17 ± 1.15	207	205	206
11	2	2048	82.23 ± 1.36	82.79 ± 1.34	89.79 ± 1.08	240	241	234
12	2	4096	81.83 ± 1.37	82.85 ± 1.34	**90.21 ± 1.06**	286	289	280
5	3	243	76.02 ± 1.52	77.73 ± 1.48	82.29 ± 1.36	158	156	160
6	3	729	80.29 ± 1.41	81.57 ± 1.38	88.05 ± 1.15	192	192	193
7	3	2187	81.73 ± 1.37	81.96 ± 1.37	88.54 ± 1.13	244	244	240
4	4	256	76.02 ± 1.52	77.23 ± 1.49	81.86 ± 1.37	161	159	158
5	4	1024	80.75 ± 1.40	81.63 ± 1.38	88.84 ± 1.12	208	205	206
6	4	4096	82.26 ± 1.36	82.85 ± 1.34	**90.26 ± 1.05**	290	289	278
4	5	625	80.26 ± 1.41	80.81 ± 1.40	86.26 ± 1.22	188	185	189
5	5	3125	82.26 ± 1.36	82.29 ± 1.36	89.45 ± 1.09	268	268	246

Table 6 Comparison of classification accuracy (in %), estimated at 95 % confidence interval, of the GMM-based classifiers and dynamic kernel based SVM classifiers for speaker identification. Here, Q indicates the number of GMM components considered. The pair (J, b) indicates values of J and b considered in constructing the pyramid

Classification model	Q or (J, b)	Classification accuracy (in %)	Average number of support vectors
GMM	64	76.50 ± 1.51	–
Adapted GMM	1024	83.08 ± 1.33	–
SVM using			
GLDS kernel (polynomial degree 3)	–	78.62 ± 1.46	187
PSK	1024	86.18 ± 1.23	182
GMMSV kernel	512	87.93 ± 1.16	396
GUMI kernel	512	$\mathbf{90.31 \pm 1.05}$	387
Summation kernel	–	78.93 ± 1.45	243
CBIMK	512	79.38 ± 1.44	287
CIGMMIMK	512	88.12 ± 1.15	254
VGPMK	(6, 4)	82.26 ± 1.36	290
CBPMK	(6, 4)	82.85 ± 1.34	289
GMMPMK	(6, 4)	$\mathbf{90.26 \pm 1.05}$	278

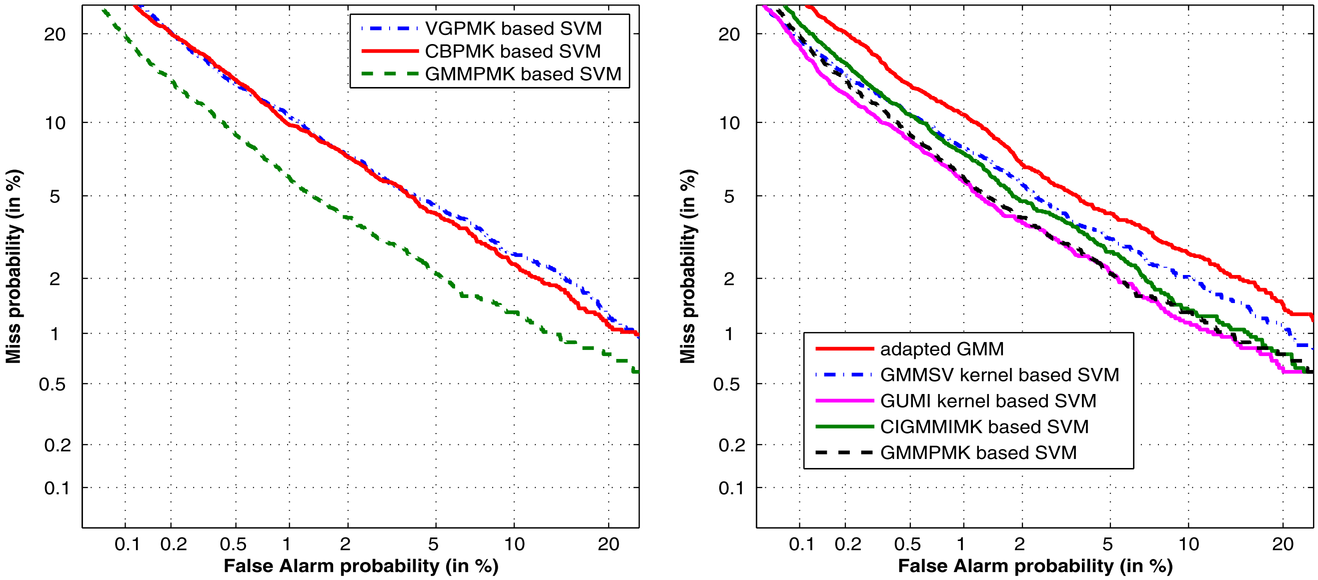
val (GUMI) kernel, summation kernel (SK), codebook-based IMK (CBIMK), CIGMM-based IMK (CIGMMIMK), vocabulary-guided PMK (VGPMK), codebook-based PMK (CBPMK), and GMM-based PMK (GMMPMK). It is seen that the SVM classifiers using the dynamic kernels such as GMMSV kernel, GUMI kernel, CIGMM-IMK, VGPMK, CBPMK and GMMPMKs give a better performance than the adapted GMM-based classifier. Though the GUMI kernel gives a marginally better performance than the GMMPMK, the average number of support vectors for SVMs using the GUMI kernel is significantly higher than the average number of support vectors for the GMMPMK based SVM. The GUMI kernel (You et al. 2009) is computed by adapting CIGMM to every example in the training set. Hence, the representation of an example in the GUMI kernel based method is of a large dimension that is proportional to the number of components, Q. Therefore, the storage requirements and the computational complexity during the recognition phase of the classifier are significantly lower for the GMMPMK-based SVM compared to the GUMI kernel based SVM. The performance of the GMMPMK-based SVM is better compared to the performance of the CIGMMIMK-based SVM. The limitation of the IMK is that it is necessary to use a set of virtual feature vectors and the performance of the IMK-based SVM is dependent on the choice of the set of virtual feature vectors. However, the computation complexity of the GMMPMK is higher when compared to the CIGMMIMK.

4.3 Studies on speaker verification using GMM and SVM based systems

In this section we present the performance of speaker verification using GMM-based systems and SVM-based systems with GLDS kernel, PSK, GMMSV kernel, GUMI kernel, SK, CBIMK, CIGMMIMK, VGPMK, CBPMK and GMMPMK. Table 7 presents the equal error rate (EER) for the GMM-based speaker verification systems and SVM-based speaker verification systems. It is seen from the Table 7 that the performance of SVM-based speaker verification systems using GLDS kernel, PSK, GMMSV kernel, GUMI kernel, SK, CBIMK, CIGMMIMK, VGPMK, CBPMK and GMMPMK in terms of EER is significantly better than that of adapted GMM-based speaker verification system. It is also seen that the EER for the SVM-based speaker verification system using the proposed GMM-based PMK is close to or better than that of the SVM-based

Table 7 Comparison of equal error rate (EER) of the GMM-based speaker verification systems and SVM-based speaker verification systems using dynamic kernels. Here, Q indicates the number of GMM components considered. The pair (J, b) indicates values of J and b considered in constructing the pyramid

Speaker verification system	Q or (J, b)	EER (in %)
GMM	64	6.01
Adapted GMM	1024	4.36
SVM using		
GLDS kernel (polynomial degree 3)	–	5.93
PSK	1024	4.02
GMMSV kernel	512	3.74
GUMI kernel	512	**3.04**
Summation kernel	–	5.47
CBIMK	512	4.99
CIGMMIMK	512	3.53
VGPMK	(6, 4)	4.66
CBPMK	(6, 4)	4.44
GMMPMK	(6, 4)	**3.05**



(a) (b)

Fig. 5 (**a**) DET curves for the SVM-based speaker verification systems using VGPMK, CBPMK and GMMPMK. (**b**) DET curves for the adapted GMM-based speaker verification systems and SVM-based speaker verification systems using GMMSV kernel, GUMI kernel, CIGMMIMK and GMMPMK

speaker verification systems with the state-of-the-art dynamic kernels. Figure 5(a) shows the detection error tradeoff (DET) curves (Martin et al. 1997) for the SVM-based speaker verification systems using VGPMK, CBPMK and GMMPMK. It is seen that SVM-based speaker verification system using GMMPMK performs significantly better than the SVM-based speaker verification systems using VGPMK and CBPMK. It is also seen that the performance of the SVM-based speaker verification system using CBPMK is better than that of the system using VGPMK. Figure 5(b) shows the DET curves for the adapted GMM based speaker verification system and the best performing SVM-based speaker verification systems. The DET curves also confirms that the performance of SVM-based speaker verification system using the proposed GMM-based PMK is close to or better than that of the SVM-based speaker verification system using state-of-the-art dynamic kernels.

5 Summary and conclusions

In this paper, we presented the GMM based approaches and SVM based approaches to speaker recognition. The discriminative training based large margin method for estimation of GMM parameters is expected to give a better performance than the maximum likelihood method. However, the optimization problem solving in large margin GMMs is computationally highly intensive. Development of the discriminative training based SVM classifiers used for speaker recognition requires the design of a suitable dynamic kernel for sequential patterns represented by sets of feature vectors. The dynamic kernels such as the generalized linear discriminant sequence (GLDS) kernel and probabilistic sequence kernel are computationally intensive. The construction of dynamic kernels such as GMM supervector kernel and GMM-UBM mean interval kernel involves building a probabilistic model for each example. The summation kernel is a simple kernel. However, it is computationally intensive when the number of feature vectors in each example is large. The intermediate matching kernel (IMK) is computationally less intensive than the GLDS kernel and summation kernel. The construction of IMK does not involve building a probabilistic model for each example. The main issue in the construction of IMK is the choice of the set of virtual feature vectors used for intermediate matching. The components of the class-independent GMM (CIGMM) are considered as the set of virtual feature vectors. The construction of a PMK involves representation of a set of feature vectors as multiresolution histogram pyramid. The PMK is computed by comparing a pair of histogram pyramids using the weighted histogram intersection function. We proposed to use CIGMMs at each level of pyramids. The histogram vector at each level is obtained using the effective number of feature vectors assigned to each component. Our studies on the 122-class speaker recognition task show that the SVM using the GMM-based PMK gives a better performance than the adapted GMMs and SVMs using IMK. It also shows that the performance of the SVM using the GMM-based PMK is close to that of the GUMI kernel based SVM which is the state-of-the-art SVM system for speaker recognition. However, the CIGMM-based PMK is computationally more intensive compared to that of the CIGMM-based IMK.

References

Auckenthaler, R., Parris, E. S., & Carey, M. J. (1999). Improving a GMM speaker verification system by phonetic weighting. In *Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP 1999)*, Phoenix, Arizona, USA, March 1999 (Vol. 1, pp. 313–316).

Boughorbel, S., Tarel, J. -P., & Fleuret, F. (2004). Non-Mercer kernels for SVM object recognition. In *Proceedings of British machine vision conference (BMVC 2004)* (pp. 137–146).

Boughorbel, S., Tarel, J. P., & Boujemaa, N. (2005). The intermediate matching kernel for image local features. In *Proceedings of the international joint conference on neural networks*, Montreal, Canada, July 2005 (pp. 889–894).

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167.

Campbell, W., Assaleh, K., & Broun, C. (2002). Speaker recognition with polynomial classifiers. *IEEE Transactions on Speech and Audio Processing*, *10*(4), 205–212.

Campbell, W. M., Campbell, J. P., Reynolds, D. A., Singer, E., & Torres-Carrasquillo, P. A. (2006a). Support vector machines for speaker and language recognition. *Computer Speech & Language*, *20*(2–3), 210–229.

Campbell, W. M., Sturim, D. E., & Reynolds, D. A. (2006b). Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, *13*(5), 308–311.

Chang, C. -C., & Lin, C. -J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Dileep, A. D., & Sekhar, C. C. (2011). Speaker recognition using intermediate matching kernel based support vector machines. In A. Neustein & H. Patil (Eds.), *Speaker forensics: new developments in voice technology to combat and detect threats to homeland security*. Berlin: Springer.

Grauman, K. L. (2006). *Matching sets of features for efficient retrieval and recognition*. PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 2006.

Grauman, K., & Darrell, T. (2007). The pyramid match kernel: efficient learning with sets of features. *Journal of Machine Learning Research*, *8*, 725–760.

Kailath, T. (1967). The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, *15*(1), 52–60.

Kinnunen, T., & Li, H. (2010). An overview of text-independent speaker recognition: from features to supervectors. *Speech Communication*, *52*, 12–40.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, *22*(1), 79–86.

Lee, K.-A., You, C. H., Li, H., & Kinnunen, T. (2007). A GMM-based probabilistic sequence kernel for speaker verification. In *Proceedings of INTERSPEECH*, Antwerp, Belgium, August 2007 (pp. 294–297).

Martin, A., Doddington, G., Kamm, T., Ordowski, M., & Przybocki, M. (1997). The DET curve in assessment of detection task performance. In *Proceedings of EUROSPEECH* (pp. 1895–1898).

Newcombe, R. G. (1998). Two-sided confidence intervals for the single proportion: comparison of seven methods. *Statistics in Medicine*, *17*(8), 857–872.

Reynolds, D. A. (1995). Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, *17*, 91–108.

Reynolds, D. A., Quatieri, T. F., & Dunn, R. B. (2000). Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, *10*, 19–41.

Sha, F., & Saul, L. (2006). Large margin Gaussian mixture modeling for phonetic classification and recognition. In *Proceedings of IEEE international conference on acoustics, speech and signal processing (ICASSP 2006)*, Toulouse, France, May 2006 (pp. 265–268).

Swain, M. J., & Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, *7*, 11–32.

The NIST year 2002 speaker recognition evaluation plan. http://www.itl.nist.gov/iad/mig/tests/spk/2002/ (2002).

The NIST year 2003 speaker recognition evaluation plan. http://www.itl.nist.gov/iad/mig/tests/sre/2003/ (2003).

Wallraven, C., Caputo, B., & Graf, A. (2003). Recognition with local features: the kernel recipe. In *Proceedings of the ninth IEEE international conference on computer vision (ICCV 2003)* (pp. 257–264).

Wan, V., & Renals, S. (2002). Evaluation of kernel methods for speaker verification and identification. In *Proceedings of IEEE international conference on acoustics, speech and signal processing*, Orlando, Florida, US, May 2002 (pp. 669–672).

You, C. H., Lee, K. A., & Li, H. (2009). An SVM kernel with GMM-supervector based on the Bhattacharyya distance for speaker recognition. *IEEE Signal Processing Letters*, *16*(1), 49–52.