Pattern Recognition Letters 29 (2008) 1427–1432

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Text-independent speaker recognition using graph matching

Ville Hautamäki *, Tomi Kinnunen, Pasi Fränti

Speech and Image Processing Unit, Department of Computer Science and Statistics, University of Joensuu, P.O. Box 111, FI-80101, Joensuu, Finland

ARTICLE INFO

Article history:
Received 31 October 2006
Received in revised form 19 February 2008
Available online 7 March 2008

Communicated by O. Siohan

Keywords:
Affine transformation invariance
Graph matching
Structural matching
kNN graph
Clustering
Speaker recognition

ABSTRACT

Technical mismatches between the training and matching conditions adversely affect the performance of a speaker recognition system. In this paper, we present a matching scheme which is invariant to feature rotation, translation and uniform scaling. The proposed approach uses a neighborhood graph to represent the global shape of the feature distribution. The reference and test graphs are aligned by graph matching and the match score is computed using conventional template matching. Experiments on the NIST-1999 SRE corpus indicate that the method is comparable to conventional Gaussian mixture model (GMM) and vector quantization (VQ)-based approaches.

1. Introduction

One of the biggest challenges in automatic speaker recognition is obtaining *invariance* across varying operating conditions, while retaining maximum speaker variability. Different handset type, transmission line/coding, and background noise are typical factors, which lead to signal mismatch across training and recognition. For a speaker recognition system to be useful in practice it needs to be optimized against the mismatch problem.

Various approaches have been proposed for tackling the invariance problem, including robust feature extraction (Mammone et al., 1996), feature normalization (Pelecanos and Sridharan, 2001), model transformation (Kenny et al., 2007; Teunen et al., 2002; Vogt and Sridharan, 2008), and match score normalization (Auckenthaler et al., 2000; Reynolds et al., 2000).

State-of-the-art text-independent speaker recognizers use mean subtraction at the utterance level, often referred to as *cepstral mean subtraction* (CMS) in the context of cepstral features. The assumption in mean subtraction is that all the feature vectors have been translated by an unknown channel-dependent vector. By subtracting the mean from both the training and testing vectors, the matching is less affected by this bias. For clean data (no channel mismatch), CMS degrades accuracy.

A general affine channel/environment model (Mak and Tsang, 2004; Mammone et al., 1996) includes rotation and scaling of the feature vectors in addition to the additive bias. The three transformations – rotation, scaling, and translation – can be collectively expressed as an affine feature distortion model: $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$. The matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$ are channel-dependent transformation parameters, whereas $\boldsymbol{x}$ and $\boldsymbol{y}$ are the "clean" and the "noisy" (observed) vectors, respectively.

In image- and video-based biometrics, invariance against rotation, translation and scaling is often desirable. For instance, a face recognizer would produce the same match score, independent of face tilting (rotation), location with respect to the background (translation) and distance from the camera (scale). A natural idea to achieve invariance is to construct a graph from certain feature points from the images and then to use *graph matching* (Bunke and Shearer, 1998) methodology. In the matching phase, only the graph structures – and not the original feature points – are compared. For example, Burge and Burger (2000) use Voronoi diagram graphs to model ear shape. The graphs of the reference ear and the unknown ear were matched using error-correcting graph matching.

It is an open question whether similar ideas could be adopted to speaker recognition. In our view, formulation of a transformation invariant matching scheme for speech features poses several challenges. First, images are two-dimensional, and the semantic meaning of the constructed graph can be visually verified. However, commonly used speech spectrum features are high-dimensional (10–40 dimensions), and it is difficult to give an intuitive meaning to the graph calculated from the extracted features. Second, in text-independent speaker recognition, the feature distributions of

* Corresponding author. Fax: +358 132517955.
E-mail address: villeh@cs.joensuu.fi (V. Hautamäki).

0167-8655/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2008.02.021

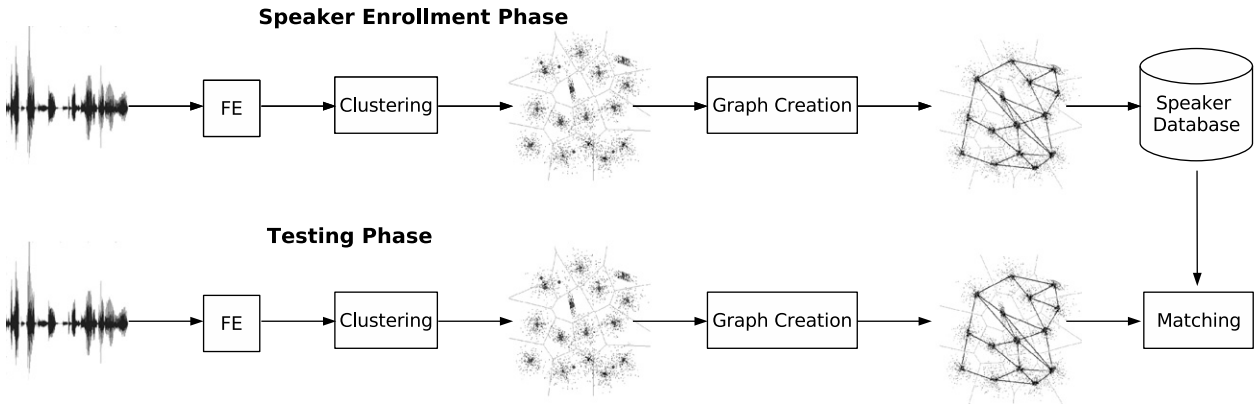

Fig. 1. Speaker recognition system diagram.

the training vectors and the test utterance are likely to vary because of text mismatch in addition to the technical mismatches mentioned above. It is also unclear whether the matching should use the whole distributions, or should a good match be indicated if the *sub*-graphs from the reference vectors and the test utterance match well.

The motivation of this paper is to experiment with a few simple ideas. To our knowledge, no graph-based matching has previously been proposed for text-independent speaker recognition. The overview of the proposed scheme is illustrated in Fig. 1. We first cluster both the training and the testing vectors into a small number of clusters, represented by a set of centroid vectors. Neighborhood graphs are then constructed for both sets. Finally, structural similarity of the reference and the test graphs is evaluated by calculating the degree of isomorphism between the graphs.

We also propose a matching framework which is a hybrid between graph-based *structural matching* and vector-based *template matching*. Graph matching is used as a pairing tool between the reference and the test centroids. The paired vectors from each set are then used for finding the parameters of the affine transformation model. Finally, the match score is computed as the distortion between the compensated centroids.

Feature and speaker model transformations, including the affine transformation, have been studied by different authors (Kenny et al., 2007; Mak and Tsang, 2004; Mammone et al., 1996; Siohan and Lee, 1997; Vogt and Sridharan, 2008). These methods usually require either parallel training data recorded simultaneously through various handsets, or a large number of training utterances collected from multiple recording sessions from a number of speakers. These datasets are then used for estimating the transformation parameters. The method that we propose, in turn, aligns the test vectors to the claimed speaker's model during verification. Therefore, the proposed method does not require any external data or training of a channel/session variability model.

The rest of the paper is organized as follows. In Section 2, we give details of the structural graph matching framework. Section 3 describes the hybrid structural and template matching algorithm. Experimental setup and the results are described in Section 4. Finally, conclusions are drawn in Section 5.

2. Graph matching based on maximum common subgraph

In order to match two high-dimensional feature sets, we need to extract some stable "feature points" from each set. This is somewhat analogous to finding minutia points from fingerprint images, or eye locations from face images. We extract the same number of feature points (N) from each set. Our assumption here is that when the two feature sets originate from the same speaker, the *relative positions* of the feature points, *i.e.* the *shape of the distribution*, remains similar despite the channel effects and noise.

The steps of the proposed graph-based alignment are summarized as follows:

(1) *Feature extraction*: Extract N feature points from both the test and the training sets by clustering.
(2) *Graph creation*: Construct neighborhood graphs from the training and testing feature points.
(3) *Pairing of the feature points*: Find the maximum common subgraph (MCS) between the two graphs.
(4) *Computing the match score*: Based on the size of the maximum common subgraph.

To implement Step 1, we use k-means (MacQueen, 1967). As a result of clustering, we have a set of code vectors located in "dense" regions of the feature distribution. It is assumed that the dense regions are related to broad phonetic classes of the given speaker and that they form stable points that can be used as the reference points in alignment.

2.1. Graph creation

The output of clustering is a set of centroid vectors. We model the relationship between the centroids using a *k-nearest neighbour graph* (kNNG), in which each vertex represents a feature vector, and the edges are pointers to the neighbouring vectors (Fränti et al., 2006). Each vertex has exactly k edges to its k-nearest neighbors in the sense of Euclidean distance. The final graph is obtained by removing the distance information (weights), and converting all the edges to undirected ones as illustrated in Fig. 2. As an example, the edge set of the graph in the right side of Fig. 2 is $E = \{(a,b),(b,c),(c,d),(d,e),(g,f),(h,i),(i,j),(j,k),(l,m),(m,n)\}$. In (Fränti et al., 2006), neighbourhood size was studied in the context of agglomerative clustering. It was found that, practical values of k vary between 5 and 12. Here we set $k = 8$.

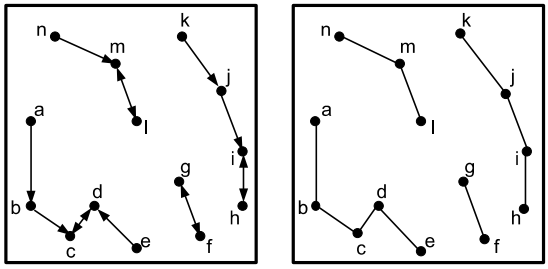

Fig. 2. Directed kNNG with $k = 1$ (left), undirected version (right).

2.2. Defining the match score

A matching function is defined between two undirected and unweighted graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where $V_{1,2}$ and $E_{1,2}$ are the discrete sets of vertices and edges, respectively. The most natural measure of similarity between two graphs is based on graph isomorphism (Diestel, 2000). Two graphs are *isomorphic* if their vertices and edges can be relabeled using a common set of labels. More precisely, the graphs G_1 and G_2 are isomorphic, if there exists a bijection from the vertices and edges of G_1 to the vertices and edges of G_2. Whenever two graphs are isomorphic, their structure (topology) is said to *match perfectly*.

Unfortunately, graphs constructed from real-world signals never match perfectly due to measurement inaccuracies and signal distortions. For that reason, we need to define a non-binary match score which measures the degree of isomorphism (Bunke and Messmer, 1997). We adopt the following distance measure (Bunke and Shearer, 1998):

$$D_{\mathrm{MCS}}(G_1, G_2) = 1 - \frac{|\mathrm{MCS}(G_1, G_2)|}{\max(|G_1|, |G_2|)}, \tag{1}$$

where $\mathrm{MCS}(G_1, G_2)$ is the *maximum common subgraph* (MCS) between G_1 and G_2, and $|\cdot|$ denotes the number of vertices in the given graph. A maximum common subgraph is defined as the largest graph, which is isomorphic to subgraphs found in G_1 and G_2. The limits of the distance function D_{MCS} are 0 and 1, where 0 is obtained when the graphs G_1 and G_2 are isomorphic. The distance reaches a maximum when the size of the MCS is 0, and this happens only when one of the graphs is empty. If both graphs contain vertices, at least the individual vertices are isomorphic to each other and D_{MCS} is less than one. In our graph matching system, $|G_1| = |G_2| = N$, so the maximum size of the MCS is N. By noting that the smallest size of MCS is 1, we obtain that (1) can have at most $N - 1$ unique values.

There are both advantages and disadvantages in using the distance (1). The advantage is that the measure (1) is theoretically a very good candidate for a similarity score: it satisfies all the metric space axioms (Bunke and Shearer, 1998), including the triangular inequality. The drawback is that finding the maximum common subgraph is an NP-complete problem (Kann, 1992). For this reason, we have to restrict the experiments to a small number of vectors.

2.3. Computing the match score

We use the *Durand–Pasari* algorithm (Durand et al., 1999) to find the maximum common subgraph. It finds the MCS in two steps. First, a so-called *association graph* (Levi, 1972) is constructed from the graphs G_1 and G_2. The association graph $G_{\mathrm{assc}} = (V_{\mathrm{assc}}, E_{\mathrm{assc}})$ is a description of the isomorphic relationships between the pairs of vertices in both graphs. We take a pair of vertices from G_1 and G_2, and if the pairs are isomorphic, the association graph has an edge denoting that relation. From the definition it follows that the association graph can be easily computed in $O(N^4)$ time. On the other hand, an association graph has N^2 vertices, so the worst case memory consumption will occur when the association graph is a complete graph, resulting in $|E_{\mathrm{assc}}| = N^4$. In practice, the memory consumption depends on the *edge density*. Edge density refers to the number of edges normalized by the number of vertices.

In the second step, the *maximum clique* (Diestel, 2000) is found from the association graph. A clique is a subgraph, which has edges between all its vertices. This step, however, takes $O(N^2 \cdot N^2!)$ time (Conte et al., 2007) and is the bottleneck of the algorithm. We therefore consider a faster heuristic known as *Reactive Local Search for Maximum Clique* (RLSMC) (Battiti and Mascia, 2007). It is an iterative local search heuristic, in which a new maximum clique solution is found in each iteration, taking into account the solutions from the previous iterations. One iteration of this algorithm works in $O(\max\{N^2, |E_{\mathrm{assc}}|\})$ time. Drawback of the heuristic is that there is no guarantee on the result compared to the exact algorithm. The steps of the matching algorithm are summarized in Algorithm 1.

Algorithm 1. Computation of D_{MCS}
Cluster the input data sets X_1 and X_2
Construct kNNG's G_1 and G_2 from the clustering result of X_1 and X_2
Construct association graph from G_1 and G_2
Find maximum clique from the association graph
Calculate score using $D_{\mathrm{MCS}}(G_1, G_2)$

2.4. Graph invariance to affine transformation

To show that the (1) is invariant to rotation, uniform scaling and translation, we need to show that kNN graph is invariant under these properties. If kNN graph is invariant, the corresponding maximum common subgraph is invariant also and the claim is true. Next we show that kNN graph is indeed invariant. Let $\boldsymbol{A}$ be a *rotation matrix* which includes uniform scaling by factor α, that is, $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} = \alpha\boldsymbol{I}$. Furthermore, let $f(\boldsymbol{x})$ be the affine map $\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$. The squared Euclidean distance between two arbitrary vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ which both have been subjected to transformation f is,

$$\begin{aligned}
\|f(\boldsymbol{x}) - f(\boldsymbol{y})\|^2 &= \|(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}) - (\boldsymbol{A}\boldsymbol{y} + \boldsymbol{b})\|^2 = (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{y}) \\
&= (\boldsymbol{A}\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{y}) - (\boldsymbol{A}\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{y}) \\
&= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y} \\
&= \alpha(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}) = \alpha\|\boldsymbol{x} - \boldsymbol{y}\|^2,
\end{aligned} \tag{2}$$

which is the distance in the original space multiplied by a constant. Therefore, the kNN neighborhood assignments in the transformed space do not change.

3. Hybrid graph- and template-based matching

Conventional speaker recognition methods based on vector quantization (VQ) or Gaussian mixture modeling (GMM) directly match the unknown person's feature vectors against the reference model(s). This *template*-based approach gives a real-valued match score but is not robust against affine transformations of features. The graph matching based on the MCS distance (1), on the other hand, implements *structural* matching as the match score is computed from the graph structures without any reference to the original vectors. The match score, however, is discrete and cannot be expected to be a very discriminative dissimilarity value: often no unique best-matching speaker was observed in our preliminary speaker identification experiments. We therefore formulate a hybrid approach that combines the good properties of the graph- and template-based approaches by using graph matching as an alignment tool prior to template matching.

The method has two main phases. In the first phase, the reference and test vector sets are paired by graph matching. The corresponding vectors from the isomorphic vertices from the kNN graphs are paired together. Denoting the N cluster centroids from the test and reference sets by $X = \{\boldsymbol{x}_i\},\ i = 1, \ldots, N$ and $Y = \{\boldsymbol{y}_i\}, \ldots, N$, respectively, graph matching produces the index pairs $(i_k, j_k),\ k = 1, 2, \ldots, K$ where $i_k, j_k \in \{1, 2, \ldots, N\}$ and $K = |\mathrm{MCS}(G_1, G_2)|$.

To this end, we have defined a pairing between the two sets of vectors that allows optimization under the affine distortion model. Assuming that $\widetilde{X} = \{\boldsymbol{x}_{i_k}\},\ k = 1, 2, \ldots, K$ contains the "clean" and $\widetilde{Y} = \{\boldsymbol{y}_{j_k}\},\ k = 1, 2, \ldots, K$ contains the "noisy" vectors, we want to

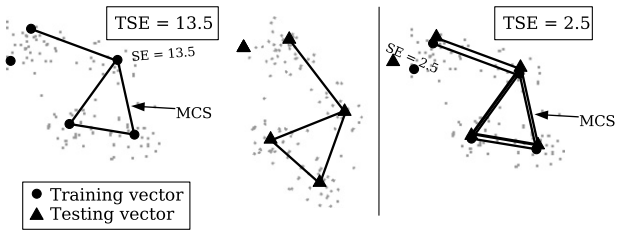

Fig. 3. Matching before (left) and after the alignment by MCS (right).

find the affine transformation that minimizes the squared error between the two paired vector sets,

$$(\boldsymbol{A}^*, \boldsymbol{b}^*) = \arg\min_{(\boldsymbol{A},\boldsymbol{b})} \sum_{k=1}^{K} \|\boldsymbol{y}_{j_k} - (\boldsymbol{A}\boldsymbol{x}_{i_k} + \boldsymbol{b})\|^2. \tag{3}$$

To solve the optimization problem (3), we represent the vectors in extended form by adding one more dimension with scalar 1, $\hat{\boldsymbol{x}} = [\boldsymbol{x}^{\mathrm{T}}, 1]^{\mathrm{T}}$. The extended vectors from each set are re-arranged into two matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ according to the pairing, that is, $\boldsymbol{X} = (\hat{\boldsymbol{x}}_{i_1}, \hat{\boldsymbol{x}}_{i_2}, \ldots, \hat{\boldsymbol{x}}_{i_K})$ and $\boldsymbol{Y} = (\hat{\boldsymbol{y}}_{j_1}, \hat{\boldsymbol{y}}_{j_2}, \ldots, \hat{\boldsymbol{y}}_{j_K})$. The parameters of the affine transformation, on the other hand, can be represented as

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{A} & & & \boldsymbol{b} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

and the optimization problem (3) can be written as a system of linear equations:

$$\boldsymbol{Y} = \widehat{\boldsymbol{W}}\boldsymbol{X}, \tag{5}$$

which has the standard least squares solution via normal equations:

$$\widehat{\boldsymbol{W}} = \boldsymbol{Y}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}. \tag{6}$$

The least squares approximation of the transform parameters can then be obtained directly from the matrix $\widehat{\boldsymbol{W}}$.

In the second phase, all the remaining (non-matched) code vectors from the sets $\overline{X} = X \setminus \widetilde{X}$ and $\overline{Y} = Y \setminus \widetilde{Y}$ are matched. Test vectors are first transformed using the affine transformation and then matched by the quantization distortion defined as:

$$D_{\mathrm{TSE}}(\overline{X}, \overline{Y}) = \sum_{\boldsymbol{x}\in\overline{X}} \min_{\boldsymbol{y}\in\overline{Y}} \|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{7}$$

The reason for excluding the MCS-paired vectors is that these vectors already match perfectly in the sense of the graph distance. A nonzero distortion value is obtained when the reference and test distribution shapes (in the sense of the graph structure) differ from each other. In this sense, the score can said to be a hybrid structural and template score. The hybrid matching algorithm is summarized in Algorithm 2, and illustrated in Fig. 3.

Algorithm 2. Hybrid graph- and template-based matching
Find the pairing (i_k, j_k), $k = 1, 2, \ldots, K$ using MCS
Find the solution $(\boldsymbol{A}^*, \boldsymbol{b}^*)$ to (3) using least squares
$\overline{X} \leftarrow X \setminus \{\boldsymbol{x}_{i_k}\}, k = 1, \ldots, K$
$\overline{Y} \leftarrow Y \setminus \{\boldsymbol{y}_{j_k}\}, k = 1, \ldots, K$
$\overline{X} \leftarrow$ replace each vector $\boldsymbol{x}$ in $\overline{X}$ by $\boldsymbol{A}^*\boldsymbol{x} + \boldsymbol{b}^*$
Compute the score as $D_{\mathrm{TSE}}(\overline{Y}, \overline{X}) + D_{\mathrm{TSE}}(\overline{X}, \overline{Y})$

4. Experiments

4.1. Effectiveness of the heuristic graph matching algorithm

We have two graph matching algorithms, the exact Durand–Pasari algorithm (exact MCS) and the heuristic RLSMC algorithm (heuristic MCS). With the exact algorithm, the size of the graph is restricted to 20 vertices in practise due to its exponential time complexity. This is clearly too small a number for modeling spectral features. Therefore, for any large scale experiments, the heuristic variant must be used.

We first study the quality of the heuristic algorithm by generating a random graph G with N vertices using the so-called *Erdős-Renyi* (Diestel, 2000) algorithm. We then randomly permutate the vertex labels in order to obtain a graph G' which is isomorphic to G. The result of the heuristic algorithm is compared to the known optimum (N), and the relative differences are summarized in Fig. 4 as a function of the iterations in the algorithm. For smaller graphs, the heuristic solution is close (within 15%) to the exact solution which is acceptable for our method. For the further experiments, we fix the number of iterations to 5000.

4.2. Demonstration of rotation invariance

Next, we demonstrate the rotation invariance property of the graph-based methods. We extract mel-frequency cepstral coefficients (MFCCs) from two speakers (see Section 4.3). We then rotate these feature sets with pre-specified rotation matrices and study whether the graph-based approaches are capable of undoing the rotation. In the ideal case, the match score (distortion) of 0 would be obtained regardless of rotation.

We refer to the two speakers as A and B, and denote the original codebooks by A0 and B0, respectively. Both of these sets are then rotated by 45° three times (in a sequence) by using different rotation axes. Rotations are performed subsequently after each other, yielding rotated feature sets denoted as A1, A2 and A3 (B1, B2 and B3, respectively). The first two dimensions of the feature sets are displayed in Figs. 5 and 6 along with $N = 20$ codebook centroids.

We match the dataset A0 against all the eight datasets and compute the match scores using the template-based method (TSE), the structural approach (graph matching), and the hybrid method (graph matching followed by TSE matching). For the graph matching, we use either the Durand–Pasari algorithm (exact) or the RLSMC algorithm (heuristic). The match scores (distances) are given in Table 1.

The TSE distance increases when the datasets become rotated more, as expected. The graph-based approach, on the other hand, is much less affected by rotation. The exact MCS variant gives a perfect match for the correct speaker in all cases. The heuristic MCS variant is slightly less accurate, but the order of the speaker scores is still correct: larger distortions are obtained for speaker B as should be. It is also noted that the hybrid approach gives a

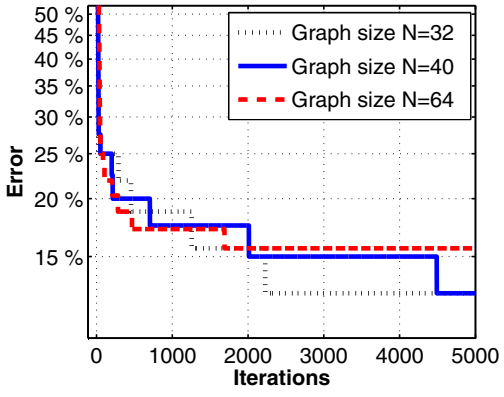

Fig. 4. The effect of the number of iterations on the accuracy of the local search heuristic (RLSMC algorithm) relative to the exact algorithm (Durand–Pasari).

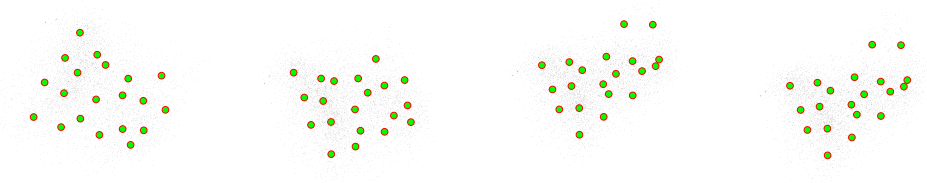

Fig. 5. The first two dimensions of the original A0 (left), along with the rotated sets A1, A2 and A3.

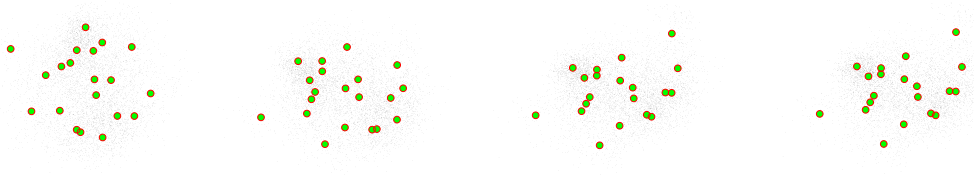

Fig. 6. The first two dimensions of the original feature set B0 (left), along with the rotated sets B1, B2 and B3.

Table 1
Match scores from the rotation experiment

Algorithm	Match A0 against							
	A0	A1	A2	A3	B0	B1	B2	B3
Template-based (TSE)	0.00	1.00	1.22	1.62	0.61	1.44	2.03	2.47
Structural (graph)								
Exact MCS	0.00	0.00	0.00	0.00	0.35	0.35	0.35	0.35
Heuristic MCS	0.05	0.05	0.05	0.05	0.35	0.35	0.35	0.35
Hybrid (graph + TSE)								
Exact MCS	0.00	0.00	0.00	0.00	1.35	2.25	2.85	2.43
Heuristic MCS	0.00	0.03	0.05	0.03	0.37	0.87	0.92	1.53

wider range of match scores compared with the purely structural matching as expected.

4.3. Speaker recognition experiments

We use a subset of the NIST-1999 SRE speaker recognition evaluation corpus for the speaker recognition experiments. The data is conversational speech acquired over a landline telephone network, and the sampling rate is 8 kHz. We use the 12 lowest mel-frequency cepstral coefficients (MFCCs) as the acoustic features, excluding the 0th coefficient. The window size is 30 milliseconds, and the mel-frequency filterbank consists of 27 triangular-shaped filters.

We have selected 30 male target speakers for our experiments from the training section of the corpus. In the original corpus, each speaker has two training files from two different recording sessions denoted as "a" and "b". We use the "a" files as the training files and the "b" files as the test data. Identification accuracy is measured by performing closed-set identification on these 30 speakers. For the verification experiments, we match all the (a, b) cross-pairs from these speakers, and an additional 50 genuine trials from additional speakers from the same corpus. This amounts to a total number of 80 genuine and 870 impostor trials. Verification accuracy is measured using the detection error trade-off (DET) plots and equal error rate (EER).

We include two standard approaches as reference systems: adapted GMM (Reynolds et al., 2000) and adapted VQ (Hautamäki et al., 2008). Both of them use a previously trained universal background model (UBM) to train the target-specific GMM, or a codebook using maximum a posteriori (MAP) criterion. We use 70 speakers from the other training segments of the NIST-1999 corpus to train the UBMs. Adaptation relevance factors of $r = 16$ and $r = 12$ are used for the GMM- and VQ-based models, respectively. Only the mean vectors are adapted in the GMM system (Reynolds et al., 2000). In the matching phase, the target score is normalized using the background model score.

The EERs are given in Table 2 and the DET plot for model size $N = 64$ is shown in Fig. 7. Overall, the graph-based approach is comparable to GMM and VQ. The graph-based approach is slightly better at small false alarm rates. The accuracy of GMM increases as a function of model size whereas the accuracy of the graph-based

Table 2
Speaker verification accuracies (EER %) on a subset of the NIST-1999 corpus

Model size	GMM baseline	VQ baseline	Graph + TSE (heuristic MCS)
32	15.3	12.5	20.0
64	17.1	12.5	13.8
128	17.2	12.5	15.0
256	15.0	11.3	N/A
512	13.8	12.5	N/A
1024	12.5	11.3	N/A

Model size refers to the number of Gaussians (GMM), code vectors (VQ) or graph vertices (graph + TSE).

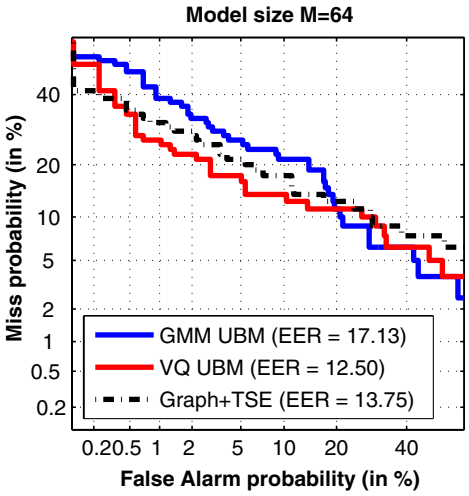

Fig. 7. Verification accuracies of the methods on the NIST-1999 corpus.

Table 3
Speaker identification error rates on a subset of the NIST-1999 corpus

Model	Individual systems		Fusion		
	GMM-UBM (%)	Graph + TSE (%)	Score-level (%)	Rank-level (%)	Voting (%)
32	30	43	27	30	37
64	23	30	20	23	27
128	23	30	23	29	27

approach degrades at $N = 128$. A likely explanation is that the heuristic MCS algorithm has not converged and the number of iterations should be set higher since the size of the association graph is much larger compared with $N = 64$. For larger graphs, the space required by the association graph and the running times were impractically high.

Finally, we choose the GMM- and graph-based systems for a fusion experiment. In *score fusion*, we first normalize the scores of each classifier to have zero mean and unit variance, followed by equal weights summation. In *rank fusion*, the scores of each classifier are converted to rank values and summed. In *voting*, the ranks are hardened to binary decisions. In the case of a tie, we randomly choose between the best-matching speakers. The fusion results are given in Table 3. Score fusion improves accuracy for the model sizes $N = 32$ and $N = 64$, which suggests that the two approaches might contain mutually complementary information.

5. Conclusions

In this study, we have introduced graph-based matching approach to text-independent speaker recognition. The approach was motivated by the fact that a neighborhood graph encodes *structural* information about the feature space. Under the affine distortion model – including rotation, translation, and uniform scaling – ideally the neighborhood graph should not change.

The performance of the proposed method was comparable to the GMM- and VQ-based approaches. A fusion experiment demonstrated that GMM- and graph-based methods might contain mutually complementary information. The approach has potential to complement or replace currently used statistical and template-based methods.

The method, however, has several practical problems to be solved before it can be utilized in real-life speaker recognition systems. First, exact graph matching is computationally expensive, and heuristic algorithm needs to be used which weakens the performance. Second, the size of the association graph grows fast for large models, which implies increased running time. The largest graph that we could test in reasonable time was 128. A possible future solution could be based on a heuristic algorithm, which solves the graph matching problem directly, without reducing it first to the maximum clique search from the association graph. To further speed up scoring in the identification task, some form of decision tree in which the "feature points" represent tree nodes, could be used.

In the current approach, the "feature points" from the reference and test sets were found by clustering and implicitly assumed to correspond to phonetic classes. In general, the joint effects of channel and text (phonetic) mismatch are not well understood. Recently, excellent results have been obtained by using phone-class constrained GMMs which reduces text mismatch by phone recognition (Castaldo et al., 2007). The graph-based method could be used by restricting matching onto the same phone classes between training and test utterances.

Different graph structures are also possible. In this study, we considered unweighted kNN graph where a node is either connected or not to another node. A possible future direction would be using real-valued weights (such as Euclidean distances between points) and re-defining the matching framework for such graphs. Current likelihood-based (or frame-based) approaches also assume independence of the frames, largely ignoring utterance-level structural information. Graph matching could be potentially used as an alternative matching tool for the existing GMM-based systems. These are points for future research.

References

Auckenthaler, R., Carey, M., Lloyd-Thomas, H., 2000. Score normalization for text-independent speaker verification systems. Digital Signal Process. 10, 42–54.

Battiti, R., Mascia, F., 2007. Reactive local search for maximum clique: A new implementation. Technical Report DIT-07-018, Informatica e Telecomunicazioni, University of Trento.

Bunke, H., Messmer, B., 1997. Recent advances in graph matching. J. Pattern Recognition Artif. Intell. 11 (1), 169–203.

Bunke, H., Shearer, K., 1998. A graph distance metric based on the maximal common subgraph. Pattern Recognition Lett. 19 (3–4), 255–259.

Burge, M., Burger, W., 2000. Ear biometrics in computer vision. In: Internat. Conf. on Pattern Recognition (ICPR), vol. 2, Barcelona, Spain, September 2000, pp. 822–826.

Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., Vair, C., 2007. Compensation of nuisance factors for speaker and language recognition. IEEE Trans. Audio Speech Language Process. 15 (7), 1969–1978.

Conte, D., Foggia, P., Vento, M., 2007. Challenging complexity of maximum common subgraph detection algorithms: A performance analysis of three algorithms on a wide database of graphs. J. Graph Algorithms Appl. 11 (1), 99–143.

Diestel, R., 2000. Graph Theory, second ed. Springer-Verlag, New York.

Durand, P., Pasari, R., Tsai, C.C., 1999. An efficient algorithm for similarity analysis of molecules. Internet J. Chem. 2 (17).

Fränti, P., Virmajoki, O., Hautamäki, V., 2006. Fast agglomerative clustering using a *k*-nearest neighbor graph. IEEE Trans. Pattern Anal. Machine Intell. 28 (11), 1875–1881.

Hautamäki, V., Kinnunen, T., Kärkkäinen, I., Saastamoinen, J., Tuononen, M., Fränti, P., 2008. Maximum a posteriori adaptation of the centroid model for speaker verification. IEEE Signal Process. Lett. 15, 162–165.

Kann, V., 1992. On the approximability of the maximum common subgraph problem. In: Proc. 9th Annual Symp. on Theoretical Aspects Computer Science (STACS), Cachan, France, February, pp. 377–388.

Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P., 2007. Speaker and session variability in GMM-based speaker verification. IEEE Trans. Audio Speech Language Process. 15 (4), 1448–1460.

Levi, G., 1972. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. Calcolo 9, 341–352.

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symp. on Mathematical Statistics and Probabilit, vol. I, University of California, pp. 281–297.

Mak, M.-W., Tsang, C.-L., 2004. Stochastic feature transformation with divergence-based out-of-handset rejection for robust speaker verification. EURASIP J. Appl. Signal Process. 4, 452–465.

Mammone, R.J., Zhang, X., Ramachandran, R.P., 1996. Robust speaker recognition: A feature-based approach. IEEE Signal Process. Mag. 13 (5), 58–71.

Pelecanos, J., Sridharan, S., 2001. Feature warping for robust speaker verification. In: Proc. Speaker Odyssey, Crete, Greece, pp. 213–218.

Reynolds, D., Quatieri, T., Dunn, R., 2000. Speaker verification using adapted gaussian mixture models. Digital Signal Process. 10 (1), 19–41.

Siohan, O., Lee, C.-H., 1997. Iterative noise and channel estimation under the stochastic matching algorithm framework. IEEE Signal Process. Lett. 4 (11), 304–306.

Teunen, R., Shahshahani, B., Heck, L., 2002. A model-based transformational approach to robust speaker recognition. In: Proc. ICSLP 2000, vol. 2, Beijing, China, pp. 495–498.

Vogt, R., Sridharan, S., 2008. Explicit modeling of session variability for speaker verification. Comput. Speech Language 22 (1), 17–38.