Speaker identification based on adaptive discriminative vector quantisation

G. Zhou and W.B. Mikhael

Abstract: A novel adaptive discriminative vector quantisation technique for speaker identification (ADVQSI) is introduced. In the training mode of ADVQSI, for each speaker, the speech feature vector space is divided into a number of subspaces. The feature space segmentation is based on the difference between the probability distribution of the speech feature vectors from each speaker and that from all speakers in the speaker identification (SI) group. Then, an optimal discriminative weight, which represents the subspace's role in SI, is calculated for each subspace of each speaker by employing adaptive techniques. The largest template differences between speakers in the SI group are achieved by using optimal discriminative weights. In the testing mode of ADVQSI, discriminative weighted average vector quantisation (VQ) distortions are used for SI decisions. The performance of ADVQSI is analysed and tested experimentally. The experimental results confirm the performance improvement employing the proposed technique in comparison with existing VQ techniques for SI and recently reported discriminative VQ techniques for SI (DVQSI).

1 Introduction

Speaker recognition refers to the capability of recognising a person based on his or her voice [1–7]. Speaker recognition can be divided into two categories, namely, speaker identification and speaker verification. Speaker identification is achieved by distinguishing a speaker from a group of speakers, whereas in speaker verification, by setting a threshold, a decision is made about whether the speaker is who he/she claims to be. Speaker recognition can be text-dependent or text-independent. The former requires the speaker to issue an utterance on some predefined text, whereas the latter does not rely on a specific text being spoken [8–10].

The most popular SI methods include vector quantisation for speaker identification (VQSI) [8, 11], dynamic time-warping (DTW) [12], hidden Markov models (HMMs) [13, 14], and neural networks (NNs) [2, 15]. The Gaussian mixture models (GMMs) Based approach [16] is a special case of the HMMs-based approach. DTW is used exclusively for text-dependent applications, whereas vector quantisation (VQ), HMMs and NNs deal with both text-dependent and text-independent speaker identification (SI).

In the training mode of the DTW approach, the speaker templates, which are the sequences of feature vectors obtained from the text-dependent speech waveforms, are created. In the testing mode, matching scores are produced by using DTW to align and measure the similarities between the test waveform and the speaker templates [8, 12].

In the VQSI approach, in the training mode, a codebook for each speaker is obtained as a reference template for the speaker. In the testing mode, SI is usually performed by finding the codebook and its corresponding speaker that gives the smallest average VQ distortion to represent the unknown speaker's waveform [11, 17]. The average VQ distortion here shows the similarity between the unknown speaker's speech and the reference template. The smaller the average VQ distortion, the better the match between the testing speech and the reference template. The lack of time warping in the VQ approach greatly simplifies the system. However, some speaker-dependent temporal information, which is present in the waveforms, is neglected in VQSI [8].

In the HMMs approach, the sequences of feature vectors, which are extracted from the speech waveforms, are assumed to be a Markov process and can be modelled with an HMM. During the training mode, HMMs' parameters are estimated from the speech waveforms. In the testing mode, the likelihood of the test feature sequence is computed based on the speaker's HMMs [14]. It is reported that the performance of the continuous HMMs is about the same as that of the VQ method and is much higher than that of the discrete HMMs [10, 18].

In the neural networks-based method, each speaker has a personalised neural network that is trained to be activated only by that speaker's utterances. The testing waveforms are tested by the speakers' personalised neural networks to make SI decisions. It has been found that if the architecture of the neural network is suitable and the number of the training utterances is enough, the performance of the neural network approach is comparable to that of the VQSI approach [2, 15].

When the same type of the speech feature is used, all the speakers in the SI group share the same speech feature vector space. For two different speaker groups (the group may contain only one speaker), the probability distributions of their speech feature vector sets in a certain subspace are different. In this work, this difference is called the inter-speaker variation between these two speaker groups in the subspace. If the subspace equals the whole speech feature vector space, this difference is called the interspeaker

IEE Proceedings online no. 20050074

doi:10.1049/ip-vis:20050074

Paper first received 11th March and in revised form 30th October 2005

The authors are with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida 32816, USA

E-mail: guangyu.zhou@gmail.com

variation between these two speaker groups. The larger the interspeaker variation, the larger the speech template difference between these two speaker groups.

The proposed adaptive discriminative VQSI (ADVQSI) technique exploits the interspeaker variation between each speaker and all speakers in the SI group in order to enlarge the speakers' template differences. For each speaker, its speech feature vector space is divided into subspaces. Different discriminative weights are given to different subspaces. Subspaces with larger discriminative weights play a more important role in the SI decision. However, in the existing VQSI technique, all regions of the speech feature vector space are given equal weights [8, 11].

The ADVQSI technique has two modes, namely, the training mode and the testing mode. In the training mode, a VQ codebook is constructed for each speaker in the SI group, and a general VQ codebook is constructed for the entire group of speakers. Then, for each speaker, the feature vector space is segmented into a number of subspaces on the basis of interspeaker variation between this speaker and all speakers in the SI group. Next, a discriminative weight is determined for each subspace of each speaker by employing adaptive techniques. The adaptively trained discriminative weights are used to represent the optimal roles of subspaces for SI. The VQ codebook for each speaker together with the feature space segmentation and discriminative weights for each speaker represent the template of that speaker. In the testing mode, for each input waveform, discriminative weighted average VQ distortions are calculated as matching scores between speakers' templates and the testing waveform. The testing waveform is identified to the speaker that leads to the highest matching score.

Recently reported DVQSI approaches also consider the interspeaker variation [19–21]. Although both DVQSI and ADVQSI employ interspeaker variation, their techniques for the speech feature vector space segmentation, the discriminative weights determination, and the SI decision in the testing mode are different. The DVQSI approach is based on each speaker pair in the SI group and discriminative weights are obtained by trial and error, whereas the ADVQSI technique is based on each speaker in the SI group and discriminative weights are calculated by adaptive techniques. The computational burden of the proposed ADVQSI is proportional to the number of speakers in the SI group, whereas the computational burden of the previously reported DVQSI increases with the square of the number of speakers in the SI group [19–21].

2 Adaptive discriminative vector quantisation for speaker identification

In the training mode of ADVQSI, the training speech waveforms for each speaker in the SI group are available. First, each speaker's training speech feature vector set is created from this speaker's training waveforms using feature-extraction techniques. After feature extraction, a VQ codebook for each speaker and a VQ codebook for all speakers are constructed. Then, for each speaker, its feature vector space is segmented into a number of subspaces based on the interspeaker variation between this speaker and all speakers in the SI group. Finally, a discriminative weight for each subspace of each speaker is calculated by employing adaptive techniques. In the ADVQSI testing mode, speech waveforms of the unknown speakers are presented to identify speakers. A testing feature vector set is created for each testing waveform in this mode. Instead of equally weighted average VQ distortions, discriminative weighted average VQ distortions are used as similarity scores between the speakers' templates and the testing waveform for SI decisions.

2.1 The training mode

In the training mode, training speech waveforms for each speaker in the SI group are available. Through feature extraction, the training speech feature vector set $\boldsymbol{T}(\boldsymbol{k})$ is extracted from the training waveforms of each speaker $k \in \Lambda$ [13], where $\Lambda = \{\text{speaker } 1, \text{ speaker } 2, \ldots, \text{speaker } h\}$ is the closed set of speakers in the SI group. The training speech feature vector set $\boldsymbol{T}(\boldsymbol{k})$ for each speaker k shares the same speech feature vector space but has a different probability distribution.

A VQ codebook $\boldsymbol{C}(\boldsymbol{k})$ for speaker k is constructed by employing $\boldsymbol{T}(\boldsymbol{k})$ of speaker k for the codebook training [22, 23]. Meanwhile, a general codebook $\boldsymbol{C}^{\boldsymbol{g}}$ is constructed for all the speakers in the SI group by using $\boldsymbol{T}^{\boldsymbol{g}}$ as the training set for the codebook construction [22, 23], where $\boldsymbol{T}^{\boldsymbol{g}} = \{\boldsymbol{T}(\boldsymbol{1}), \boldsymbol{T}(\boldsymbol{2}), \ldots, \boldsymbol{T}(\boldsymbol{h})\}$ is the set of all training speech feature vectors for all speakers.

After the codebooks have been constructed, for each speaker, the speech feature vector space is segmented into a number of subspaces on the basis of the interspeaker variation between this speaker and all speakers in the SI group. The speech feature vector space is first segmented into two subspaces. Then, the process is repeated to segment each subspace into two parts until the desired number of the subspaces is obtained. The desired number of subspaces for the space segmentation is denoted by m. The space segmentation procedure for $m = 4$ is given in Fig. 1 as an example. The process, which segments the space or the subspace into two parts, can be divided into two stages. In the first stage, the space-segmentation problem is converted into a pattern-classification problem by defining two pattern-classification training categories. Then, in the second stage, a decision surface is created by linear discriminant function techniques to divide the feature space or subspace into two subspaces [24].

The pattern classification problem in this work is to find a suitable linear discriminant function, with which to classify two linearly non-separable categories $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ based on the mean square error (MSE) criterion.

A linear discriminant function that is a linear combination of the components of $\boldsymbol{x}$ ($\boldsymbol{x} \in R^d$, $\boldsymbol{x}$ is from categories ω_1 or ω_2) can be written as

$$g(\boldsymbol{x}) = \boldsymbol{a}'\boldsymbol{y} \tag{1}$$

where prime means transpose, $\boldsymbol{y} = [1, \boldsymbol{x}']'$ and $\boldsymbol{a} = [w_0, \boldsymbol{w}]'$ is the weight vector to be calculated.

The equation $g(\boldsymbol{x}) = 0$ defines a decision surface that divides the d-dimension vector space into two subspaces. Thus, the two-category linear classifier implements the following decision rule: $\boldsymbol{x}$ is from category $\boldsymbol{\omega}_1$ if $g(\boldsymbol{x}) > 0$ and from category $\boldsymbol{\omega}_2$ if $g(\boldsymbol{x}) < 0$. If $g(\boldsymbol{x}) = 0$, $\boldsymbol{x}$ can ordinarily be assigned to either class [24].

Then, the pattern-classification problem is converted into finding a weight vector $\boldsymbol{a}$ that minimises the MSE criterion function

$$J_s(\boldsymbol{a}) = \|\boldsymbol{Ya} - \boldsymbol{b}\| = \sum_i (\boldsymbol{a}'\boldsymbol{y_i} - b_i)^2$$

where $\boldsymbol{b} = [b_1, b_2, \ldots, b_n]'$ is a column vector [24].

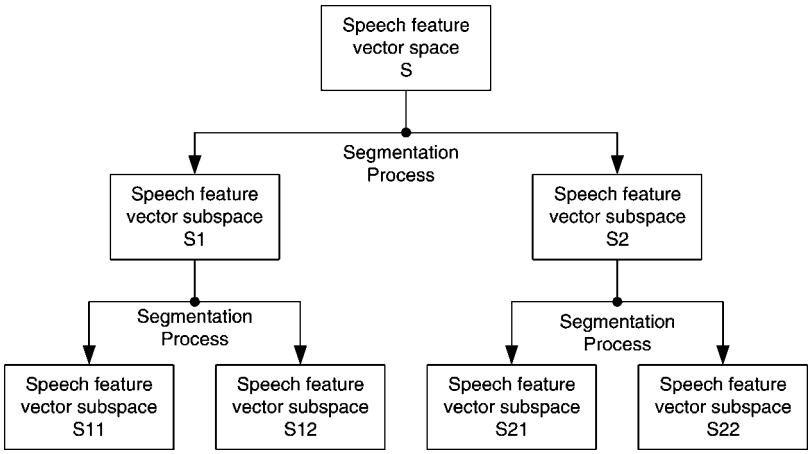

Fig. 1 *Speech feature vector space segmentation procedure for the number of subspaces $m = 4$*

If the matrix $\boldsymbol{Y}'\boldsymbol{Y}$ is non-singular [24], the solution is given by

$$\boldsymbol{a} = (\boldsymbol{Y}'\boldsymbol{Y})^{-1}\boldsymbol{Y}'\boldsymbol{b} \tag{2}$$

Typically, $b_i = 1$ is selected for the vectors from one category and $b_i = -1$ is assigned for the vectors from the other category. It has been shown that, in this case, the MSE solution approximates the Bayes discriminant function as the number of training vectors tends to infinity [24].

In the first stage of the space-segmentation process, for each speaker k and each training feature vector $\boldsymbol{v} \in \boldsymbol{T}(\boldsymbol{k})$ of speaker k, the distortion $d(\boldsymbol{v}, k)$ of $\boldsymbol{v}$ quantised by codebook $\boldsymbol{C}(\boldsymbol{k})$ of speaker k and the distortion $d(\boldsymbol{v}, g)$ of $\boldsymbol{v}$ quantised by general codebook $\boldsymbol{C}^{\boldsymbol{g}}$ are calculated. Let $d(\boldsymbol{v}) = d(\boldsymbol{v}, k)/d(\boldsymbol{v}, g)$. Typically, when $d(\boldsymbol{v})$ is lower, $\boldsymbol{v}$ is located in the region of the feature space with a higher interspeaker variation between speaker k and all speakers, and vice versa. Then, the training feature vector set $\boldsymbol{T}(\boldsymbol{k})$ of speaker k is divided into two subsets, namely $\boldsymbol{T_1}$ and $\boldsymbol{T_2}$. $\boldsymbol{T_1}$ contains the feature vector with larger $d(\boldsymbol{v})$ while $\boldsymbol{T_2}$ contains the remaining feature vectors. The numbers of feature vectors in $\boldsymbol{T_1}$ and $\boldsymbol{T_2}$ are the same. As $\boldsymbol{v}$ with larger $d(\boldsymbol{v})$ is typically located in the region of the feature space with a lower interspeaker variation, most feature vectors in $\boldsymbol{T_1}$ are located in the regions of the feature space with lower interspeaker variations. In contrast, feature vectors in $\boldsymbol{T_2}$ are mainly located in the regions of the feature space with higher interspeaker variations. The space-segmentation problem is converted into a pattern-classification problem by letting $\boldsymbol{T_1}$ and $\boldsymbol{T_2}$ be the two training categories of the linear pattern-classification problem.

In the second stage of the space segmentation, a linear discriminant function $g(\boldsymbol{x})$ in (1) is constructed with its weight vector $\boldsymbol{a}$ given by (2), where $b_i = 1$ for vectors from $\boldsymbol{T_1}$ and $b_i = -1$ for vectors from $\boldsymbol{T_2}$. The corresponding decision surface $g(\boldsymbol{x}) = 0$ divides the speech feature vector space S into two subspaces S_1 and S_2. The subspace for $\boldsymbol{T_1}$ has a lower interspeaker variation between speaker $k1$ and all speakers than the subspace for $\boldsymbol{T_2}$, as feature vectors in $\boldsymbol{T_1}$ are typically located in regions of the feature space with lower interspeaker variations than feature vectors in $\boldsymbol{T_2}$. The feature space segmentation of ADVQSI is based on the interspeaker variation between each speaker and all speakers. Similar procedures are repeated to divide S_1 and S_2, and their subspaces, until the desired number of subspaces for ADVQSI is met. The feature space segmentation for each speaker is decided by the linear discriminant functions for this speaker.

In ADVQSI, each speaker's template is represented by this particular speaker's codebook, discriminative weights for subspaces, and feature space segmentation. In order to obtain optimal discriminative weights for all speakers by adaptive techniques, an initial positive discriminative weight is assigned to each subspace of each speaker. Then the differences for templates of various speakers are measured on the basis of the initial discriminative weights.

The average VQ distortion $d_{kj}(k1, k2)$ of $\boldsymbol{T_j}(\boldsymbol{k1}, \boldsymbol{k2})$ quantised by $\boldsymbol{C}(\boldsymbol{k2})$ is calculated for each speaker $k1$ and each subspace j of speaker $k2$, where $\boldsymbol{T_j}(\boldsymbol{k1}, \boldsymbol{k2})$ is the set of all speech feature vectors of $\boldsymbol{T}(\boldsymbol{k1})$ located in subspace j of speaker $k2$; speaker $k1 \in \boldsymbol{\Lambda}$ and speaker $k2 \in \boldsymbol{\Lambda}$; and $j = 1, 2, \ldots, m$ is the subspace index. Similarly, the average VQ distortion of $\boldsymbol{T_j}(\boldsymbol{k1}, \boldsymbol{k2})$ quantised by $\boldsymbol{C}^{\boldsymbol{g}}$ is obtained and denoted by $d_{gj}(k1, k2)$. Let $d_j(k1, k2) = d_{gj}(k1, k2) - d_{kj}(k1, k2)$.

The weighted average distortion $d_{\text{dis}}(k1, k2)$ is defined as

$$d_{\text{dis}}(k1, k2) = \frac{\boldsymbol{W}(\boldsymbol{k2})'\,\boldsymbol{N}(\boldsymbol{k1}, \boldsymbol{k2})\,\boldsymbol{D}(\boldsymbol{k1}, \boldsymbol{k2})}{\boldsymbol{W}(\boldsymbol{k2})'\,\boldsymbol{n}(\boldsymbol{k1}, \boldsymbol{k2})} \tag{3}$$

where

$$\begin{aligned}
\boldsymbol{D}(\boldsymbol{k1}, \boldsymbol{k2}) &= [d_1(k1, k2), d_2(k1, k2), \ldots, d_m(k1, k2)]' \\
\boldsymbol{W}(\boldsymbol{k2}) &= [w_1(k2), w_2(k2), \ldots, w_m(k2)]' \\
\boldsymbol{N}(\boldsymbol{k1}, \boldsymbol{k2}) &= diag[n_1(k1, k2), n_2(k1, k2), \ldots, n_m(k1, k2)] \\
\boldsymbol{n}(\boldsymbol{k1}, \boldsymbol{k2}) &= [n_1(k1, k2), n_2(k1, k2), \ldots, n_m(k1, k2)]'
\end{aligned}$$

$w_j(k2)$ is the discriminative weight for each subspace j of each speaker $k2$ and $n_j(k1, k2)$ is the number of the feature vectors of $\boldsymbol{T_j}(\boldsymbol{k1}, \boldsymbol{k2})$.

$d_{\text{dis}}(k1, k2)$ is the measure of the similarity score between the training set of speaker $k1$ and the template of speaker $k2$ under current discriminative weights. $d_{\text{dis}}(k1, k1)$ is always larger than $d_{\text{dis}}(k1, k2)$ $(k \neq k2)$ for any positive discriminative weights, as the training set always best matches the speaker's template that is created from this training set.

Let $h_{\text{dis}}(k1, k2) = d_{\text{dis}}(k1, k1) - d_{\text{dis}}(k1, k2)$. $h_{\text{dis}}(k1, k2)$ is the measure of the template difference between the speaker $k1$ and $k2$ under current discriminative weights. $h_{\text{dis}}(k1, k2)$ equals zero when $k1 = k2$, and $h_{\text{dis}}(k1, k2)$ is larger than zero for $k1 \neq k2$. The larger the $h_{\text{dis}}(k1, k2)$, the larger the template difference between speaker $k1$ and $k2$.

The cost function to obtain optimal discriminative weights is given by

$$J = \sum_{k1=1}^{N} \sum_{k2=1, k1 \neq k2}^{N} f(h_{\mathrm{dis}}(k1, k2)) \tag{4}$$

where

$$f(x) = \mathrm{e}^{-\alpha x + \beta}$$

$\alpha > 0$ and β are scalars.

To increase the SI accuracy, $h_{\mathrm{dis}}(k1, k2)$ and the corresponding template difference between speaker $k1$ and $k2$ are required to be as large as possible, thus the cost function (4) needs to be minimised. It is desired to find discriminative weights that minimise the cost function J, so that the template differences between different speakers are maximised. The selection of $f(x)$ in (4) will be explained in detail later.

The gradient vector $\nabla \boldsymbol{J}(\boldsymbol{W}(\boldsymbol{k2}))$ is given by

$$\begin{aligned}\nabla \boldsymbol{J}(\boldsymbol{W}(\boldsymbol{k2})) &= \frac{\partial \boldsymbol{J}}{\partial[\boldsymbol{W}(\boldsymbol{k2})]} \\ &= \sum_{k1=1}^{N} \frac{\mathrm{d}[f(h_{\mathrm{dis}}(k1, k2))]}{\mathrm{d}[h_{\mathrm{dis}}(k1, k2)]} \frac{\partial[\boldsymbol{h}_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2})]}{\partial[\boldsymbol{W}(\boldsymbol{k2})]} \\ &\quad + \sum_{k1=1}^{N} \frac{\mathrm{d}[f(h_{\mathrm{dis}}(k2, k1))]}{\mathrm{d}[h_{\mathrm{dis}}(k2, k1)]} \frac{\partial[\boldsymbol{h}_{\mathrm{dis}}(\boldsymbol{k2}, \boldsymbol{k1})]}{\partial[\boldsymbol{w}(\boldsymbol{k2})]}\end{aligned} \tag{5}$$

where

$$\begin{aligned}\frac{\mathrm{d}[f(x)]}{\mathrm{d}[x]} &= -\alpha \mathrm{e}^{-\alpha x + \beta} \\ \frac{\partial[\boldsymbol{h}_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2})]}{\partial[\boldsymbol{W}(\boldsymbol{k2})]} &= -\frac{\mathrm{d}[\boldsymbol{d}_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2})]}{\mathrm{d}[\boldsymbol{W}(\boldsymbol{k2})]} \\ \frac{\partial[\boldsymbol{h}_{\mathrm{dis}}(\boldsymbol{k2}, \boldsymbol{k1})]}{\partial[\boldsymbol{W}(\boldsymbol{k2})]} &= \frac{\mathrm{d}[\boldsymbol{d}_{\mathrm{dis}}(\boldsymbol{k2}, \boldsymbol{k2})]}{\mathrm{d}[\boldsymbol{W}(\boldsymbol{k2})]} \\ \frac{\mathrm{d}[\boldsymbol{d}_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2})]}{\mathrm{d}[\boldsymbol{W}(\boldsymbol{k2})]} &= \frac{\boldsymbol{N}(\boldsymbol{k1}, \boldsymbol{k2})\, \boldsymbol{D}(\boldsymbol{k1}, \boldsymbol{k2})}{\boldsymbol{W}(\boldsymbol{k2})' \boldsymbol{n}(\boldsymbol{k1}, \boldsymbol{k2})} \\ &\quad - \frac{\boldsymbol{n}(\boldsymbol{k1}, \boldsymbol{k2})(\boldsymbol{W}(\boldsymbol{k2})' \boldsymbol{N}(\boldsymbol{k1}, \boldsymbol{k2})\, \boldsymbol{D}(\boldsymbol{k1}, \boldsymbol{k2}))}{(\boldsymbol{W}(\boldsymbol{k2})' \boldsymbol{n}(\boldsymbol{k1}, \boldsymbol{k2}))^2} \\ \frac{\mathrm{d}[\boldsymbol{d}_{\mathrm{dis}}(\boldsymbol{k2}, \boldsymbol{k2})]}{\mathrm{d}[\boldsymbol{W}(\boldsymbol{k2})]} &= \frac{\boldsymbol{N}(\boldsymbol{k2}, \boldsymbol{k2})\, \boldsymbol{D}(\boldsymbol{k2}, \boldsymbol{k2})}{\boldsymbol{W}(\boldsymbol{k2})' \boldsymbol{n}(\boldsymbol{k2}, \boldsymbol{k2})} \\ &\quad - \frac{\boldsymbol{n}(\boldsymbol{k2}, \boldsymbol{k2})(\boldsymbol{W}(\boldsymbol{k2})' \boldsymbol{N}(\boldsymbol{k2}, \boldsymbol{k2}) \boldsymbol{D}(\boldsymbol{k2}, \boldsymbol{k2}))}{(\boldsymbol{W}(\boldsymbol{k2})' \boldsymbol{n}(\boldsymbol{k2}, \boldsymbol{k2}))^2}\end{aligned} \tag{6}$$

The updating function for discriminative weights is expressed as

$$\boldsymbol{W} = \boldsymbol{W} - \Gamma \times \nabla \boldsymbol{J}(\boldsymbol{W}) \tag{7}$$

where

$$\begin{aligned}\boldsymbol{W} &= [\boldsymbol{W}(\boldsymbol{1}), \boldsymbol{W}(\boldsymbol{2}), \ldots, \boldsymbol{W}(\boldsymbol{h})] \\ \nabla \boldsymbol{J}(\boldsymbol{W}) &= [\nabla \boldsymbol{J}(\boldsymbol{W}(\boldsymbol{1})), \nabla \boldsymbol{J}(\boldsymbol{W}(\boldsymbol{2})), \ldots, \nabla \boldsymbol{J}(\boldsymbol{W}(\mathbf{h}))]\end{aligned}$$

and scalar Γ is the convergence factor.

$h_{\mathrm{dis}}(k1, k2)$ represents the template difference between the speaker $k1$ and $k2$ under current discriminative weights. When two speakers have large $h_{\mathrm{dis}}(k1, k2)$ and a corresponding larger template difference between them, the testing waveforms from these speakers are less likely to be misidentified to each other. Further increasing large $h_{\mathrm{dis}}(k1, k2)$ has little advantage for the SI accuracy improvement. However, increasing smaller $h_{\mathrm{dis}}(k1, k2)$ is more likely to increase the SI accuracy. In order to increase the SI accuracy, in the discriminative weights updating, it is desirable to give priority to increasing the smaller $h_{\mathrm{dis}}(k1, k2)$ than larger ones.

In (5), $-\partial[\boldsymbol{h}_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2})]/\partial[\boldsymbol{W}(\boldsymbol{k2})]$ is the direction in which to increase only $h_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2})$. The term $\mathbf{d}[\boldsymbol{f}(\boldsymbol{h}_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2}))]/\boldsymbol{d}[\boldsymbol{h}_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2})]$ that appears in (5) is the multiplier factor of $-\partial[\boldsymbol{h}_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2})]/\partial[\boldsymbol{W}(\boldsymbol{k2})]$. It is smaller for larger $h_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2})$ and larger for smaller $h_{\mathrm{dis}}(\boldsymbol{k1}, \boldsymbol{k2})$. Compared with the cost function, which is the direct summation of $h_{\mathrm{dis}}(k1, k2)$, the effect of smaller $h_{\mathrm{dis}}(k1, k2)$ for the discriminative weight updating in (7) has been enlarged by introducing $f(x) = \mathrm{e}^{-\alpha x + \beta}$ in (4). Thus, smaller $h_{\mathrm{dis}}(k1, k2)$ has higher priority for increasing than larger $h_{\mathrm{dis}}(k1, k2)$ in the discriminative weight updating.

Similarly, $h_{\mathrm{dis}}(k2, k1)$ also represents the template difference between the speaker $k2$ and $k1$ under current discriminative weights. Again, smaller $h_{\mathrm{dis}}(k2, k1)$ has higher priority for increasing than larger $h_{\mathrm{dis}}(k2, k1)$ in the discriminative weight updating.

The diagram of the training mode of ADVQSI is shown in Fig. 2. Codebook $\mathbf{C}(\boldsymbol{k})$, discriminative weight $\boldsymbol{W}(\boldsymbol{k})$ and space segmentation for speaker k represent the template of speaker k. All the templates of speakers in the SI group together with general codebook $\boldsymbol{C}^{\boldsymbol{g}}$ are used in the testing mode of ADVQSI.

2.2 The testing mode

In the testing mode, testing waveforms from unknown speakers in the SI group are presented for speaker identification. For each testing waveform R, a testing speech feature vector set $\boldsymbol{T}(\boldsymbol{R})$ is created from waveform R. In this mode, for each testing waveform, discriminative weighted average VQ distortions are calculated. Then, the SI decision is made on the basis of these weighted average VQ distortions.

For each testing waveform R, the discriminative weighted average VQ distortion $d_{\mathrm{dis}}(R, k)$ for speaker k is given by

$$d_{\mathrm{dis}}(R, K) = \frac{\boldsymbol{W}(\boldsymbol{k})' \boldsymbol{N}(\boldsymbol{R}, \boldsymbol{k})\, \boldsymbol{D}(\boldsymbol{R}, \boldsymbol{k})}{\boldsymbol{W}(\boldsymbol{k})' \boldsymbol{n}(\boldsymbol{R}, \boldsymbol{k})} \tag{8}$$

where

$$\begin{aligned}\boldsymbol{D}(\boldsymbol{R}, \boldsymbol{k}) &= [d_1(R, k), d_2(R, k), \ldots, d_m(R, k)]' \\ d_j(R, k) &= d_{gi}(R, k) - d_{kj}(R, k) \\ \boldsymbol{N}(\boldsymbol{R}, \boldsymbol{k}) &= diag[n_1(R, k), n_2(R, k), \ldots, n_m(R, k)] \\ \boldsymbol{n}(\boldsymbol{R}, \boldsymbol{k}) &= [n_1(R, k), n_2(R, k), \ldots, n_m(R, k)]'\end{aligned}$$

$n_j(R, k)$ is the number of the feature vectors in $\boldsymbol{T}_{\boldsymbol{j}}^{\boldsymbol{R}}(\boldsymbol{k})$, $d_{kj}(R, k)$ is the average VQ distortion of $\boldsymbol{T}_{\boldsymbol{j}}(\boldsymbol{R}, \boldsymbol{k})$ quantised by $\boldsymbol{C}(\boldsymbol{k})$, $d_{gj}(R, k)$ is the average VQ distortion of $\boldsymbol{T}_{\boldsymbol{j}}(\boldsymbol{R}, \boldsymbol{k})$ quantised by $\boldsymbol{C}^{\boldsymbol{g}}$ and $\boldsymbol{T}_{\boldsymbol{j}}(\boldsymbol{R}, \boldsymbol{k})$ is the set for all speech feature vectors of $\boldsymbol{T}(\boldsymbol{R})$ located in subspace j of speaker k.

$d_{\mathrm{dis}}(R, k)$ is the similarity matching score between the testing waveform R and the template of speaker k. The larger the $d_{\mathrm{dis}}(R, k)$, the better the match. The definition of $d_{\mathrm{dis}}(R, k)$ is similar to the definition of $d_{\mathrm{dis}}(k1, k2)$, except the former uses the testing speech feature vector set and the latter considers the training speech feature vector set. The definitions of $d_{\mathrm{dis}}(k1, k2)$ in the training mode and $d_{\mathrm{dis}}(R, k)$ in the testing mode are consistent.

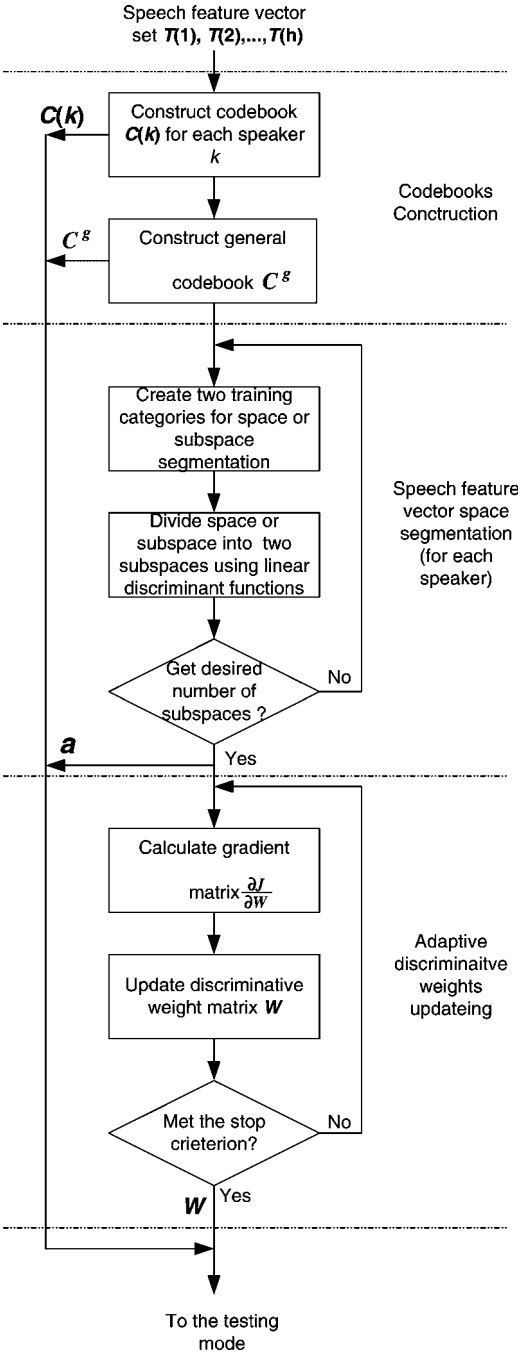

Fig. 2 *Diagram of the training mode of ADVQSI*

The SI decision rule is expressed as follows: the unknown waveform R comes from speaker i, if $d_{\text{dis}}(R, i) = \max_{k=1, 2, \ldots, h} d_{\text{dis}}(R, k)$. The testing waveform is classified to the speaker whose template most closely matches the testing waveform.

3 Experimental Results

In this section, experiments are given to evaluate the effectiveness of the proposed ADVQSI approach. Speech records are obtained from CSLU (Center for Spoken Language Understanding, Oregon Health & Science University) Speaker Recognition V1.1 corpus. For each speaker, the speech records collected on different collection dates are packaged into different recording sessions. There are mismatches between the speech utterances taken from different speakers. Also, there are mismatches because of different recording sessions of the same speaker. All the speech files in the corpus were sampled at 8 kHz and 8-bits per sample.

Table 1: SI accuracy rates employing VQSI, DVQSI and ADVQSI

Technique	VQSI	DVQSI	ADVQSI
SI accuracy, %	62.9	71.4	71.4

Thirty-five speakers are used in the text-independent SI experiments. Four spontaneous speeches for each speaker are used in the training mode. Two other spontaneous speeches, taken about one year after the training speech waveform for each speaker, are used in the testing mode. Each speech waveform lasts about 4 seconds.

Silenced and unvoiced segments are discarded based on an energy threshold. The analysis Hamming window size is 32 ms, 256 samples, with 24 ms overlapping [13]. The feature vector used in the experiment is composed of 15 Mel frequency cepstral coefficients (MFCCs) [25].

The codebook sizes of VQSI, DVQSI and ADVQSI are 64. In this work, the speech feature vector space is divided into four subspaces, that is, $m = 4$. All the codebooks are constructed by the generalised Lloyd algorithm [22, 23]. The initial values of codebooks are obtained by using the splitting algorithm [22, 23]. The parameters for the adaptive discriminative weight updating are $\alpha = 0.3$, $\beta = 9$ and $\Gamma = 0.05$. The initial values for all discriminative weights are 100.

The performance of SI using VQ, HMM, DTW and NN techniques are compared in the literatures [2, 10, 15, 18]. The experimental results in Matsui and Furui [10] and Yu *et al.* [18] show that VQ performs better than an equivalent continuous HMM if a small amount of the training data is provided, but is outperformed by continuous HMM when the amount of the training data is large. VQ works much better than discrete HMM [10]. In text-dependent experiments, DTW outperforms VQ and continuous HMM for small amounts of training data, but with more data, these three methods are indistinguishable [18]. The performances of NN are comparable with VQ [2, 15]. For small model size, NN does better than VQ. However, as the model size is increased, NN falls behind [15].

Table 1 shows the SI accuracy results employing VQSI, DVQSI and ADVQSI. It is observed that ADVQSI and

Table 2: Number of operations in VQSI, DVQSI and ADVQSI

	VQSI	DVQSI	ADVQSI
Training mode			
VQ codebook construction	h	$h(h-1)k$	$h+1$
Speech feature vector space segmentation	N/A	$h(h-1)k/2$	h
Subspace average VQ distortion calculation	N/A	mh^2k	mh^2
Adaptive discriminative weights calculation	N/A	N/A	1
Testing mode (for each input waveform)			
Average VQ distortion calculation	h	$2(h-1)$	h

k is the number of trials and errors for DVQSI, h is the number of speakers in the SI group and m is the number of subspaces.

Table 3: Average d(v) for the first speaker in the speech feature vector space segmentation

For all the training feature vectors	For feature vectors in subspace 1	For feature vectors in subspace 2	For feature vectors in subspace 3	For feature vectors in subspace 4
0.6763	0.4110	0.6420	0.7028	0.8862

DVQSI result in higher SI accuracy than VQSI. The discussions and simulation results of the parameters selection for DVQSI are given in Zhou and Mikhael [19, 20] and Zhou *et al.* [21]. Compared with VQSI, DVQSI and ADVQSI exploit interspeaker variations between different speakers (or speaker groups). The ADVQSI approach employs adaptive techniques to find optimal discriminative weights, whereas the DVQSI approach obtains discriminative weights by trial and error [19–21].

Numbers of all time-consuming operations in VQSI, DVQSI and ADVQSI are given in Table 2. In the training mode of DVQSI and ADVQSI, VQ codebook construction takes most of the computational time. Typically, the number of iterations for the adaptive discriminative weight calculation in ADVQSI is less than 300. Much less time is needed for the adaptive discriminative weight calculation than any other training mode operations listed in Table 2. In the training mode, the computational burden of ADVQSI is higher than VQSI, but much less than that of DVQSI.

To calculate the average VQ distortion in the testing mode of VQSI, each speech feature vector for testing is compared with all the codewords in the codebook to obtain the best match. However, for DVQSI and ADVQSI, besides this process, each speech feature vector needs to be classified into a certain subspace to obtain the corresponding discriminative weight. Then, a small amount of calculations that is proportional to $\log_2 m$ is added in DVQSI and ADVQSI. As m is always much smaller than the number of codewords, the computational burden of each average VQ distortion calculation in DVQSI and ADVQSI is almost as much as that of VQSI. Consequently, in the testing mode, the computational burden of ADVQSI is slightly higher than that of VQSI, and almost half of DVQSI.

For simplification, in ADVQSI, the subspaces are ranked from the highest interspeaker variation to the lowest interspeaker variation for all speakers. Table 3 shows the average values of d(v) for the speech feature vector space segmentation of the first speaker. The average values of d(v) for different subspaces are not equal in Table 3. This means that different subspaces have various interspeaker variations between speaker 1 and all speakers in the SI group, that is, the lower the average value of d(v), the higher is the interspeaker variation in the subspace. The feature vector space of ADVQSI is segmented on the basis of the interspeaker variation between each speaker and all speakers in the SI group.

The mean value of the discriminative weights for all the speakers in each subspace against the number of adaptive iterations is presented in Fig. 3. From Fig. 3, it is seen that the subspaces with the higher interspeaker variation increase their discriminative weights as the adaptive algorithm converges. In contrast, the adaptive algorithm reduces discriminative weights of subspaces, which have lower interspeaker variations. As a result, the subspaces with higher interspeaker variations play more important roles in the SI decision than the ones with lower interspeaker variations by assigning different discriminative weights to different subspaces. Although the mean values of the discriminative weights in different subspaces are different at the end of the discriminative weight updating, all of them are positive. This means that all the subspaces play positive roles in SI.

The value of the cost function J in (4) against the number of adaptive iterations is given in Fig. 4. The value of the cost function decreases as the adaptive algorithm converges. The average value of $h_{\text{dis}}(k1, k2)$ for all possible speaker pairs against the adaptive iteration number is given in Fig. 5. This value increases when the number of adaptive iterations

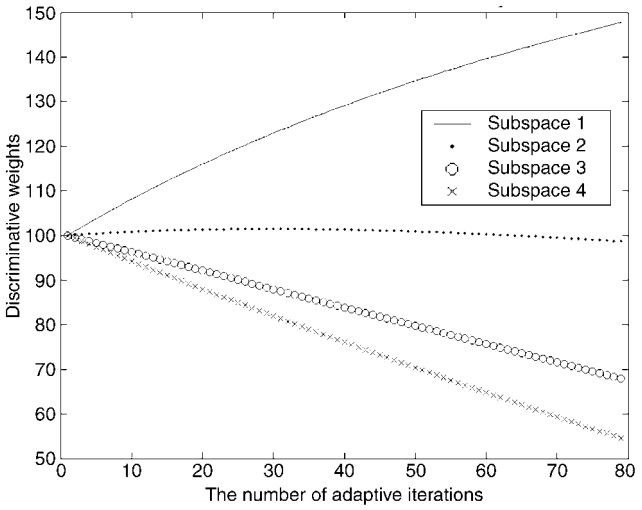

Fig. 3 *Average discriminative weights for different subspaces against the number of adaptive iterations*

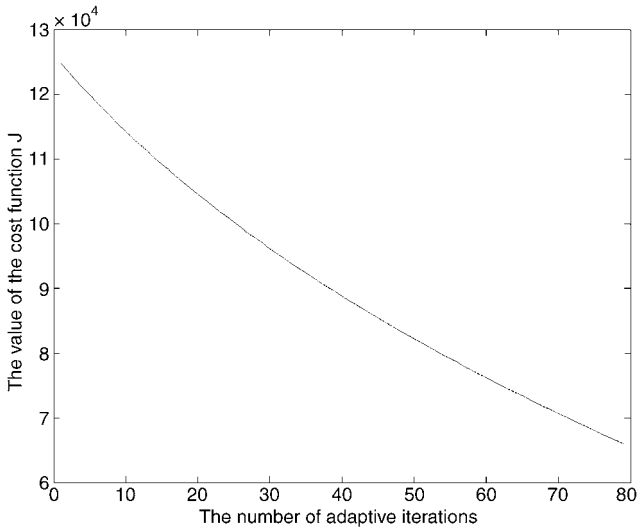

Fig. 4 *Value of the cost function J in (4) against the number of adaptive iterations*

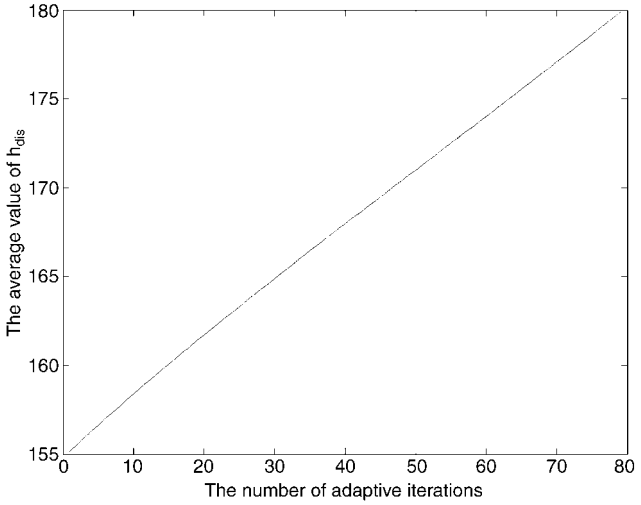

Fig. 5 *Average value of $h_{dis}(k1, k2)$ for all speaker pairs against the number of adaptive iterations*

increases. The results confirm that the adaptive algorithm converges successfully.

4 Conclusions

In this work, a new SI approach based on adaptive discriminative VQ is developed and presented. The ADVQSI technique takes advantage of the interspeaker variation between each individual speaker and all speakers in the SI group. In the training mode of this technique, for each speaker, the speech feature vector space is divided into a number of subspaces on the basis of interspeaker variation between this speaker and all speakers. Then, an optimal discriminative weight is adaptively trained for each speaker and each subspace in order to maximise the template differences between different speakers for SI. In the test mode of ADVQSI, discriminative weighted average VQ distortions are used as similarity measurements between speakers' templates and each testing waveform. The testing waveform is classified to the speaker whose template leads to the highest similarity score.

The effectiveness of the ADVQSI approach is demonstrated experimentally. It is shown that the proposed technique yields better SI accuracy than the VQSI approaches.

Compared with the recently reported DVQSI approach, AVDQSI determines discriminative weights by using adaptive techniques instead of trial and error. Because ADVQSI considers each speaker instead of each speaker pair, the computational requirement of ADVQSI is considerably reduced relative to DVQSI, in which discriminative weights are assigned for each speaker pair [19–21].

Although the ADVQSI technique is applied to SI, this technique can be gainfully extended to other pattern identification applications, such as handwritten character identification and face identification.

5 Acknowledgment

This work was supported by the Conexant Corporation.

6 References

1 Campell, W.M., Assaleh, K.T., and Broun, C.C.: 'Speaker recognition with polynomial classifiers', *IEEE Trans. Speech Audio Process.*, 2002, **10**, (4), pp. 205–211
2 Farrell, K.R., Mammone, R.J., and Assaleh, K.T.: 'Speaker recognition using neural networks and conventional classifiers', *IEEE Trans. Speech Audio Process.*, 1994, **2**, (1), pp. 194–205
3 Furui, S.: 'Recent advance in speaker recognition', *Pattern Recognit. Lett.*, 1997, **18**, pp. 859–872
4 Gish, H., and Schmidt, M.: 'Text-independent speaker identification', *IEEE Signal Process. Mag.*, 1994, **11**, pp. 18–32
5 Gold, B., and Morgan, N.: 'Speech and audio signal processing: processing and perception of speech and music' (John Wiley & Sons, NY, 2000)
6 Higgins, A., Bhaler, L., and Porter, J.: 'Voice identification using nearest neighbour distance measure'. ICASSP-93, 1993, pp. 375–378
7 Lapidot, I., Guterman, H., and Cohen, A.: 'Unsupervised speaker recognition based on competition between self-organizing maps', *IEEE Trans. Neural Netw.*, 2002, **13**, (4), pp. 877–887
8 Campbell, J.P.: 'Speaker recognition: a tutorial', *Proc. IEEE*, 1997, **85**, pp. 1437–1462
9 Mammone, R., Zhang, X., and Ramachandran, R.: 'Robust speaker recognition—a feature-based approach', *IEEE Signal Process. Mag.*, 1996, **13**, pp. 58–71
10 Matsui, T., and Furui, S.: 'Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMM's', *IEEE Trans. Speech Audio Process.*, 1994, **2**, (3), pp. 456–459
11 Soong, F.K., Rosenberg, A.E., Rabiner, L.R., and Juang, B.-H.: 'A vector quantisation approach to speaker recognition'. ICASSP-85, 1985, pp. 387–390
12 Sakoe, H., and Chiba, S.: 'Dynamic programming algorithm optimization for spoken word recognition', *IEEE Trans. Acoust. Speech Signal Process.*, 1978, **26**, pp. 43–49
13 Rabiner, L., and Juang, B.: 'Fundamentals of speech recognition' (Prentice-Hall, London, 1993)
14 Tishby, N.Z.: 'On the application of mixture AR hidden Markov models to text independent speaker recognition', *IEEE Trans. Acoust. Speech Signal Process.*, 1991, pp. 563–570
15 Oglesby, J., and Mason, J.S.: 'Optimisation of neural models for speaker identification'. ICASSP-90, 1990, pp. 261–264
16 Reynolds, D.A.: 'Speaker identification and verification using Gaussian Mixture speaker models', *Speech Commun.*, 1995, **17**, pp. 91–108
17 Mikhael, W.B., and Premakanthan, P.: 'Speaker identification employing redundant vector quantisers', *Electron. Lett.*, 2002, **38**, pp. 1396–1398
18 Yu, K., Mason, J., and Oglesby, J.: 'Speaker recognition using hidden Markov models, dynamic time warping and vector quantisation', *IEE Proc., Vis. Image Signal Process.*, 1995, **142**, (5), pp. 313–318
19 Zhou, G., and Mikheal, W.B.: 'Speaker identification based on discriminative vector quantisation'. 46th IEEE Int. Midwest Symp. on Circuits and Systems, Cairo, Egypt, December 2003
20 Zhou, G., and Mikhael, W.B.: 'Analysis of discriminative vector quantization approach for speaker identification'. 8th World Multi-Conf. on Systemic, Cybernetics and Information, Orlando, FL, USA, 2004, vol. IV, pp. 479–483
21 Zhou, G., Mikhael, W.B., and Myers, B.: 'A novel discriminative vector quantisation approach for speaker identification', *J. Circuits, Syst. Comput.*, 2005, **14**, (3), pp. 581–596
22 Gresho, A., and Gray, R.M.: 'Vector quantisation and signal compression' (Kluwer Academic Publisher, Boston, 1991)
23 Linde, Y., Buzo, A., and Gray, R.M.: 'An algorithm for vector quantizer design', *IEEE Trans. Commun.*, 1980, **28**, pp. 702–710
24 Duda, R.O., Hart, P.E., and Stork, D.G.: 'Pattern classification' (John Wiley & Sons, NY, 2001)
25 Vergin, R., O'Shaughnessy, D., and Farhat, A.: 'Generalised mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition', *IEEE Trans. Speech Audio Process.*, 1999, **7**, pp. 525–532