accepted to appear in Computer Speech and Language

Evolutionary Minimization of the Rand Index for Speaker Clustering

Wei-Ho Tsai[1] *and Hsin-Min Wang*[2]

[1]Department of Electronic Engineering & Graduate Institute of Computer and Communication Engineering, National Taipei University of Technology, Taipei 10608, Taiwan
E-mail: whtsai@ntut.edu.tw

[2]Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan
E-mail: whm@iis.sinica.edu.tw

ABSTRACT

We propose an effective method for clustering unknown speech utterances based on their associated speakers. The method jointly optimizes the generated clusters and the required number of clusters by estimating and minimizing the Rand index. The metric reflects the clustering errors that arise when utterances from the same speaker are placed in different clusters; or when utterances from different speakers are placed in the same cluster. One useful characteristic of the Rand index is that its value only reaches the minimum when the number of clusters is equal to the size of the true speaker population. We approximate the Rand index by a function of the similarity measures between utterances and then use a genetic algorithm to determine the cluster in which each utterance should be located, such that the function is minimized. Our experiment results show that this novel speaker-clustering method outperforms conventional methods that use the Bayesian information criterion to determine the required number of clusters.

Keywords: Genetic algorithm, Rand index, Speaker clustering

1. Introduction

Motivated by the need for effective methods to index and archive the burgeoning amount of spoken data being generated universally, recent research on automatic classification of speech samples based on speakers' voice characteristics has been extended from the traditional supervised problem of speaker identification/verification (Campbell, 1997) to an unsupervised paradigm (Makhoul et al., 2000). Basically, the paradigm involves two tasks: segmenting an audio recording into speech utterances that contain only one speaker's voice (Siegler et al., 1997; Johnson, 1999; Zhou and Hansen, 2000), and grouping utterances from the same speaker into one cluster (Gish et al., 1991; Jin et al., 1997; Solomonoff et al., 1998; Chen and Gopalakrishnan, 1998; Reynolds et al., 1998). The tasks can be addressed jointly by a process called *speaker diarization* (Tranter and Reynolds, 2006; Ben et al., 2004; Tranter, 2005; Zhu et al., 2005; Sinha et al., 2005). It is hoped that, by locating utterances from the same speaker, the human effort required to index speech data can be greatly reduced, i.e., from having to listen to every audio recording to only checking a few utterances in each cluster. In this paper, we concentrate on the latter problem, referred to as *speaker clustering*. Assume that we have a collection of N speech utterances, each of which is from one of P unknown speakers, where $N \geq P$, and P is also unknown. The aim of speaker clustering is to partition the N utterances into M clusters such that $M = P$ and each cluster only contains utterances from one speaker.

Since no prior information regarding the speakers involved and the speaker population size is available in most practical applications, a common strategy used to solve the speaker-

clustering problem involves three steps: characterizing the voice similarities between utterances, generating clusters based on those similarities, and determining the optimal number of clusters. The most popular speaker-clustering method employs hierarchical agglomerative clustering (HAC) (Gish et al., 1991; Jin et al., 1997; Solomonoff et al., 1998; Chen and Gopalakrishnan, 1998; Reynolds et al., 1998; Johnson and Woodland, 1998; Faltlhauser and Ruske, 2001; Ajmera et al., 2002; Moh et al., 2003; Liu, 2005). This approach generates a cluster tree by sequentially merging the utterances deemed similar to each other. Then, the tree is cut using the Bayesian information criterion (BIC) (Schwarz, 1978; Chen and Gopalakrishnan, 1998; Zhou and Hansen, 2000) to retain the appropriate number of clusters. Although various modifications to the method have been proposed (Ben et al., 2004; Tranter, 2005; Zhu et al., 2005; Sinha et al., 2005), most of them focus on improving the combination of speaker clustering and speaker segmentation in a speaker diarization system. There is a dearth of research on improving the performance of speaker clustering per se.

As noted in our previous works (Tsai and Wang, 2005, 2006), one major drawback of most speaker-clustering systems is the problem of error propagation in HAC. Specifically, although HAC merges the most similar utterances sequentially, it is possible that, in some merging operations, utterances by different speakers may be mis-grouped into the same cluster. In such cases, the utterances cannot be separated in subsequent merging operations; hence, the mis-grouping errors will proliferate as more clusters are merged. To resolve this problem, we proposed clustering methods that maximize the within-cluster homogeneity of speakers' voice characteristics by jointly considering all the clusters to be generated, instead of by the cluster-by-cluster technique used in HAC. However, like most speaker-clustering systems, our approach followed the principle of BIC-based methods by determining the optimal number of clusters after completion of the cluster generation process. Since the back-end determination of the optimal number of clusters trusts the front-end cluster generation process completely, the inevitable errors generated by the front-end can propagate to the back-end, which may lead to inaccurate estimation of the speaker population size.

To overcome the above-mentioned limitations, in this paper, we propose a new clustering method that simultaneously optimizes the generated clusters and the required number of clusters by estimating and then minimizing the Rand index (Rand, 1971; Hubert and Arabie, 1985; Solomonoff, 1998). The metric indicates clustering errors that place utterances from the same speaker in different clusters, or place utterances from different speakers in the same cluster. A useful characteristic of the Rand index is that its value only reaches the minimum when the number of clusters is equal to the true size of the speaker population. We approximate the Rand index by a function of the similarity measures between utterances, and use a genetic algorithm (Goldberg, 1989) to determine the cluster in which each utterance should be located, such that the function is minimized. The resulting clusters are thus optimized in a global fashion, rather than in the pair-by-pair manner used in HAC-based methods. Furthermore, the number of clusters derived by minimizing the approximated Rand index naturally reflects the speaker population size.

The remainder of the paper is organized as follows. In Section 2, we explain our motivation for studying the problem of speaker clustering and describe the performance assessment method used in this study. Section 3 introduces the proposed method for estimating and minimizing the Rand index, whereby the resulting partition of utterances approaches an optimal state in terms of within-cluster homogeneity and the number of clusters. Section 4 details our experiment results. Then, in Section 5, we present our conclusions and indicate the direction of our future work.

2. Problem Formulation

We begin by defining the notations used in this paper.

$\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$: N speech utterances to be clustered, each of which is represented by a frame-based spectral feature stream;

$s_1, s_2, \ldots, s_P$: P unknown speakers involved in the N utterances, where P is also unknown;

$c_1, c_2, \ldots, c_M$: M clusters to be generated;

o_n : the index of the speaker producing utterance $\mathbf{X}_n$;

h_n : the index of the cluster to which utterance $\mathbf{X}_n$ is assigned;

n_{m*} : the number of utterances in c_m;

n_{*p} : the number of utterances spoken by s_p;

n_{mp} : the number of utterances in c_m spoken by s_p.

Speaker clustering can be viewed as a problem of determining a set of indices $\mathbf{H} = \{h_1, h_2, \ldots, h_N\}$ that satisfy $h_i = h_j$ for any $\mathbf{X}_i$ and $\mathbf{X}_j$ from the same speaker, and $h_i \neq h_j$ for any $\mathbf{X}_i$ and $\mathbf{X}_j$ from different speakers.

Depending on the application, there are a number of ways to evaluate the performance of speaker clustering. In this study, we use two metrics: cluster purity (Solomonoff, 1998) and the Rand index (Rand, 1971; Hubert and Arabie, 1985; Solomonoff, 1998). Cluster purity indicates the degree of correct clustering. It is represented by the probability that if we pick any utterance from a cluster twice at random, with replacement, both of the selected utterances will be from the same speaker. Specifically, the average purity of M clusters is computed by

$$\overline{\rho} = \frac{1}{N}\sum_{m=1}^{M} n_{m*}\,\rho_m, \tag{1}$$

where ρ_m is the purity of cluster c_m:

$$\rho_m = \sum_{p=1}^{P}\left(\frac{n_{mp}}{n_{m*}}\right)^2. \tag{2}$$

Obviously, a perfect clustering should produce an average purity of one. However, this does not work both ways. The value of the average purity generally increases as the number of clusters increases, since the metric does not consider errors that place utterances from the same speaker in different clusters. Hence, the cluster purity is only suitable for comparing the performance of different clustering methods if the number of clusters is specified.

In contrast, the Rand index indicates the extent of incorrect clustering. It is defined as the number of utterance pairs from the same speaker that are in different clusters, or utterance pairs from different speakers that are in the same cluster, i.e.,

$$R(M) = \sum_{m=1}^{M} n_{m*}^2 + \sum_{p=1}^{P} n_{*p}^2 - 2\sum_{m=1}^{M}\sum_{p=1}^{P} n_{mp}^2. \tag{3}$$

Alternatively, the Rand index can be represented as a mis-clustering rate:

$$R(M) \text{ in percentage} = \frac{\sum_{m=1}^{M} n_{m*}^2 + \sum_{p=1}^{P} n_{*p}^2 - 2\sum_{m=1}^{M}\sum_{p=1}^{P} n_{mp}^2}{\sum_{m=1}^{M} n_{m*}^2 + \sum_{p=1}^{P} n_{*p}^2} \times 100\%. \tag{4}$$

Obviously, the smaller the value of $R(M)$, the better the clustering performance will be. Unlike the cluster purity metric, which favors a large M value, the Rand index usually decreases with an increase in the value of M initially, and reaches the minimum at $M = P$. When $M > P$, the Rand index starts to increase as the value of M increases.

To illustrate why the minimal value of $R(M)$ only occurs when $M = P$, let us consider the following cases.

(i) The clustering is perfect, which satisfies

$$\begin{pmatrix} n_{11} & n_{21} & \dots & n_{M1} \\ n_{12} & n_{22} & \dots & n_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ n_{1P} & n_{2P} & \dots & n_{MP} \end{pmatrix} = \begin{pmatrix} n_1 & 0 & \dots & 0 \\ 0 & n_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & n_P \end{pmatrix}_{P\times P}, \tag{5}$$

where $n_k = n_{*k} = n_{k*}$, $1 \le k \le P$. Then, the resulting Rand index is

$$R^*(P) = \sum_{m=1}^{P} n_{m*}^2 + \sum_{p=1}^{P} n_{*p}^2 - 2\sum_{m=1}^{P}\sum_{p=1}^{P} n_{mp}^2 = \sum_{m=1}^{P} n_m^2 + \sum_{p=1}^{P} n_p^2 - 2\sum_{k=1}^{P} n_k^2 = 0. \tag{6}$$

(ii) Let $M = P + 1$, and modify Eq. (5) by splitting cluster c_k into two clusters, c_k and c_{P+1}, i.e.,

$$\begin{pmatrix} n_{11} & n_{21} & \dots & n_{M1} \\ n_{12} & n_{22} & \dots & n_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ n_{1P} & n_{2P} & \dots & n_{MP} \end{pmatrix} = \begin{pmatrix} n_1 & 0 & \dots & 0 & \dots & 0 & 0 \\ 0 & n_2 & \dots & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & n_{kk} & \dots & 0 & n_{(P+1)k} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \dots & n_P & 0 \end{pmatrix}_{P\times(P+1)}, \tag{7}$$

where $n_{kk} + n_{(P+1)k} = n_k$. Then, the resulting Rand index is

$$\begin{aligned} R(P+1) &= \sum_{m=1}^{P+1} n_{m*}^2 + \sum_{p=1}^{P} n_{*p}^2 - 2\sum_{m=1}^{P+1}\sum_{p=1}^{P} n_{mp}^2 \\ &= \left(\sum_{m=1}^{P} n_m^2 - n_k^2 + n_{kk}^2 + n_{(P+1)k}^2\right) + \sum_{p=1}^{P} n_p^2 - 2\left(\sum_{m=1}^{P} n_m^2 - n_k^2 + n_{kk}^2 + n_{(P+1)k}^2\right) \\ &= n_k^2 - n_{kk}^2 - n_{(P+1)k}^2 = n_k^2 - n_{kk}^2 - (n_k - n_{kk})^2 = 2n_{kk}(n_k - n_{kk}) > 0. \end{aligned} \tag{8}$$

(iii) Let $M = P - 1$, and modify Eq. (5) by merging cluster c_P into cluster c_k, i.e.,

$$\begin{pmatrix} n_{11} & n_{21} & \dots & n_{M1} \\ n_{12} & n_{22} & \dots & n_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ n_{1P} & n_{2P} & \dots & n_{MP} \end{pmatrix} = \begin{pmatrix} n_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & n_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & n_k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & n_{P-1} \\ 0 & 0 & \dots & n_P & \dots & 0 \end{pmatrix}_{P\times(P-1)}. \tag{9}$$

Then, the resulting Rand index is

$$\begin{aligned} R(P-1) &= \sum_{m=1}^{P-1} n_{m*}^2 + \sum_{p=1}^{P} n_{*p}^2 - 2\sum_{m=1}^{P-1}\sum_{p=1}^{P} n_{mp}^2 \\ &= \left(\sum_{m=1}^{P} n_m^2 - n_P^2 - n_k^2 + (n_k + n_P)^2\right) + \sum_{p=1}^{P} n_p^2 - 2\sum_{m=1}^{P} n_m^2 \\ &= 2n_k n_P > 0. \end{aligned} \tag{10}$$

From these cases, we observe that, in general, $R(M) > R(P)$ if $M \neq P$. Therefore, the Rand index can be used to ascertain if each generated cluster is homogeneous in terms of the speaker, and also as a criterion to determine the size of the true speaker population. This property motivates us to develop a clustering method that jointly optimizes the generated clusters and the required number of clusters by estimating and then minimizing the Rand index.

3. Minimum Rand Index Clustering (MRIC)

Our basic strategy is to determine a set of indices $\mathbf{H}(M) = \{h_1(M), h_2(M), \ldots, h_N(M)\}$ for the N utterances to be clustered, such that the resulting Rand index is minimized, where $h_i(M)$, $1 \le i \le N$, is an integer between 1 and M, and the value of M is to be determined. This can be achieved by first representing the Rand index as a function of the indices, and then minimizing it with respect to the indices. Since in Eq. (3)

$$\begin{aligned}\sum_{m=1}^{M} n_{m*}^2 &= \sum_{m=1}^{M}\left[\sum_{i=1}^{N}\delta\big(h_i(M),m\big)\right]^2 \\ &= \sum_{m=1}^{M}\left[\sum_{i=1}^{N}\delta\big(h_i(M),m\big)\right]\left[\sum_{j=1}^{N}\delta\big(h_j(M),m\big)\right] \\ &= \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\delta\big(h_i(M),m\big)\delta\big(h_j(M),m\big) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}\delta\big(h_i(M),h_j(M)\big),\end{aligned} \tag{11}$$

$$\begin{aligned}\sum_{m=1}^{M}\sum_{p=1}^{P} n_{mp}^2 &= \sum_{m=1}^{M}\sum_{p=1}^{P}\left[\sum_{i=1}^{N}\delta\big(h_i(M),m\big)\delta(o_i,p)\right]^2 \\ &= \sum_{m=1}^{M}\sum_{p=1}^{P}\left[\sum_{i=1}^{N}\delta\big(h_i(M),m\big)\delta(o_i,p)\right]\left[\sum_{j=1}^{N}\delta\big(h_j(M),m\big)\delta(o_j,p)\right] \\ &= \sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N}\delta\big(h_i(M),m\big)\delta(o_i,p)\delta\big(h_j(M),m\big)\delta(o_j,p) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}\delta\big(h_i(M),h_j(M)\big)\delta(o_i,o_j),\end{aligned} \tag{12}$$

and $\sum_{p=1}^{P} n_{*p}^2 = \Omega$ is a constant that does not affect the clustering, the optimal set of cluster indices can be determined by

$$\mathbf{H}^* = \underset{\mathbf{H}(M),1\le M\le N}{\arg\min}\ \hat{R}(\mathbf{H}(M)), \tag{13}$$

and

$$\hat{R}(\mathbf{H}(M)) = \sum_{i=1}^{N}\sum_{j=1}^{N}\delta\big(h_i(M),h_j(M)\big) + \Omega - 2\sum_{i=1}^{N}\sum_{j=1}^{N}\delta\big(h_i(M),h_j(M)\big)\delta(o_i,o_j), \tag{14}$$

where $\delta(\cdot)$ in Eqs. (11)–(14) is a Kronecker Delta function.

However, as the computation of $\delta(o_i, o_j)$ in Eq. (14) requires that the true speaker of each utterance be known in advance, it is impossible to find $\mathbf{H}^*$ directly from Eqs. (13) and (14). To solve this problem, we estimate $\delta(o_i, o_j)$ based on the similarity between $\mathbf{X}_i$ and $\mathbf{X}_j$. Specifically,

$$\delta(o_i,o_j) \leftarrow \begin{cases} 1 & , \text{ if } i = j \\ S(\mathbf{X}_i,\mathbf{X}_j)/S_{\max} & , \text{ if } i \ne j, \text{ and } S_{\max} > 0, \\ S_{\max}/S(\mathbf{X}_i,\mathbf{X}_j) & , \text{ if } i \ne j, \text{ and } S_{\max} < 0 \end{cases} \tag{15}$$

where $S(\mathbf{X}_i, \mathbf{X}_j)$ denotes a certain similarity measure between $\mathbf{X}_i$ and $\mathbf{X}_j$ that could be either positive or negative, but it cannot be zero; and $S_{\max}$ is the largest of the similarity measures $S(\mathbf{X}_i, \mathbf{X}_j)$, $\forall\ i \ne j$. Hence, the approximation of $\delta(o_i, o_j)$ in Eq. (15) is a positive value less than 1.

In our implementation, $S(\mathbf{X}_i, \mathbf{X}_j)$ is computed by the following generalized likelihood ratio (GLR) (Gish et al., 1991; Solomonoff, 1998):

$$S(\mathbf{X}_i, \mathbf{X}_j) = \log\Pr(\mathbf{X}_{ij} \mid \lambda_{ij}) - \log\Pr(\mathbf{X}_i \mid \lambda_i) - \log\Pr(\mathbf{X}_j \mid \lambda_j), \quad (16)$$

where $\mathbf{X}_{ij}$ is the concatenation of $\mathbf{X}_i$ and $\mathbf{X}_j$; and λ_i, λ_j, and λ_{ij} are parametric models trained using $\mathbf{X}_i$, $\mathbf{X}_j$, and $\mathbf{X}_{ij}$, respectively. Using this approximation, we can solve Eq. (13) by assigning an arbitrary positive constant to Ω in order to ensure that $\hat{R}(\mathbf{H}(M)) \geq 0$.

Given that neither a gradient-based optimization method nor an exhaustive search is applicable in this scenario, to find $\mathbf{H}^*$, we use the genetic algorithm (GA) (Goldberg, 1989) because of its global scope and parallel searching power. Basically, the GA explores a given search space in parallel by iteratively modifying a population of chromosomes. Each chromosome, encoded as a string of alphabets or real numbers called genes, represents a potential solution to a given problem. In our task, a chromosome is exactly a legitimate $\mathbf{H}(M)$, and a gene corresponds to a cluster index associated with an utterance. However, since the index of one cluster can be interchanged with that of another cluster, multiple chromosomes may yield an identical clustering result. For example, the chromosomes {1 1 1 2 2 3 3}, {1 1 1 3 3 2 2}, {2 2 2 1 1 3 3}, and {1 1 1 5 5 4 4} represent the same clustering result derived by grouping seven utterances into three clusters. Such a non-unique representation of the solution would significantly increase the GA search space, and may lead to an inferior clustering result. To avoid this problem, we limit the inventory of chromosomes to conform to a baseform representation defined as follows.

Let $I(c_m)$ be the lowest index of the utterance in cluster c_m. Then, a chromosome is a baseform

$$\text{iff } \forall\, c_m, c_l \neq \{\phi\}, \text{ if } m < l, \text{ then } I(c_m) < I(c_l), \quad (17)$$

where $\{\phi\}$ indicates that a cluster does not contain any utterance. Among the above chromosomes, {1 1 1 2 2 3 3} is a baseform, since the lowest index of the utterance in clusters c_1, c_2, and c_3 is 1, 4, and 6, respectively, which satisfies Eq. (17). In contrast, chromosomes {1 1 1 3 3 2 2} and {2 2 2 1 1 3 3} are not baseforms, since the lowest index of the utterance in clusters c_1, c_2, and c_3 does not satisfy Eq. (17). Meanwhile, chromosome {1 1 1 5 5 4 4} implies that clusters c_2 and c_3 do not contain any utterances; hence it is not a baseform either. However, it is conceivable that all the non-baseform chromosomes can be converted into a unique baseform representation by re-arranging the cluster indices.

Fig. 1 shows the flow diagram of GA optimization, which starts by randomly generating chromosomes $\mathbf{H}^{(1)}(M^{(1)})$, $\mathbf{H}^{(2)}(M^{(2)})$,…, $\mathbf{H}^{(Z)}(M^{(Z)})$ according to a pre-defined population size, Z, and the number of generated clusters, $M^{(z)}$, $1 \leq z \leq Z$. For example, to cluster seven utterances, we can generate chromosomes like {1 1 1 2 2 3 3}, {1 2 2 3 3 4 2}, {1 2 2 2 2 2 1}, in which the number of generated clusters is 3, 4, and 2, respectively. Then, the fitness of all chromosomes is evaluated via the inverse of the estimated Rand index, i.e., $F\left(\mathbf{H}^{(z)}(M^{(z)})\right) = 1/\hat{R}\left(\mathbf{H}^{(z)}(M^{(z)})\right)$, $1 \leq z \leq Z$. Based on this evaluation, a particular group of chromosomes is selected from the population to generate offspring by subsequent recombination. To prevent premature convergence of the population, the selection operation is performed with the linear ranking scheme (Baker, 1985), which sorts chromosomes in decreasing order of fitness, and then assigns the expected number of offspring according to their relative ranking. Chromosomes with large fitness values produce several copies, while those with very small fitness values may be eliminated; hence, the total chromosome population size does not change.

Next, crossover between the selected chromosomes is performed by exchanging the substrings of two chromosomes at two randomly selected crossover points. For example, the

crossover for chromosomes {1 1 1 2 2 3 3} and {1 2 3 1 2 3 2} generates {1 1 3 1 2 3 3} and {1 2 1 2 2 3 2} respectively, if the selected crossover points are 2 and 6 (indicated by the underlined parts). However, as shown in this example, the resulting chromosomes, such as {1 1 3 1 2 3 3}, may not conform to Eq. (17). Therefore, the procedure for interchanging the clusters' indices has to be repeated to ensure that all the offspring are baseforms. In this example, the chromosome {1 1 3 1 2 3 3} is converted to {1 1 2 1 3 2 2} by swapping index "2" with index "3". In addition, a crossover probability is assigned to control the ratio of the number of offspring produced in each generation to the size of the chromosome population.

After the crossover operation, a mutation operator is used to introduce random variations into the genetic structure of the chromosomes. To do this, we generate a random number and then replace one gene of an existing chromosome with a mutation probability. The resulting chromosomes that do not conform to the baseform representations are converted into their baseform counterparts. Then, the fitness evaluation, selection, crossover, and mutation steps are repeated continuously, in the hope that the overall fitness of the population will increase from generation to generation. When a pre-set maximum number of generations, say Q, is reached, the best chromosome in the final population is taken as the solution, $\mathbf{H}^*$.

4. Experiments

4.1 Speech data

The speech data used for performance evaluation consisted of six excerpts of broadcasts from the evaluation set of the *2002 Rich Transcription* (*RT-02*) *Broadcast News and Conversational Telephone Speech Corpus* (Linguistic Data Consortium, http://www.ldc.upenn.edu/), in which the speech was digitized with a 16 kHz sampling rate and 16-bit quantization resolution. Using the annotation files attached to the corpus, we segmented each excerpt into a collection of isolated speech utterances, each containing only one speaker's voice. Table 1 summarizes the utterance duration and speaker population size for each excerpt. In the speaker-clustering experiments we used each excerpt separately.

To tune the system parameters that cannot be optimized automatically, we used another dataset extracted from the *2001 NIST Speaker Recognition Evaluation Corpus* (Linguistic Data Consortium, http://www.ldc.upenn.edu/). The dataset contained 197 speaker-homogeneous utterances spoken by 15 randomly-selected male speakers. The utterances were recorded via cellular phones and digitized with an 8 kHz sampling rate and 16-bit quantization resolution. We denoted this dataset as SRE-01.

Prior to the experiments, every utterance was converted from its digital waveform representation into a sequence of feature vectors, each consisting of 12 Mel-scale frequency cepstral coefficients (MFCCs) and 12 delta MFCCs, computed using a 20-ms Hamming window (frame) with a 10-ms frame shift. Then, the similarities between the utterances were computed using Eq. (16), in which each parametric model is a uni-Gaussian model with a full covariance matrix.

4.2 Baseline systems

For the performance comparison, we implemented two baseline systems, denoted as Baseline-I and Baseline-II. The first system used the HAC framework in conjunction with a BIC-based method to determine the optimal number of clusters (Chen and Gopalakrishnan, 1998). In the agglomeration procedure, the similarities between clusters were computed using the *complete linkage* of the GLR-based inter-utterance similarities. The penalty weight of the BIC method was set to one, based on the optimization using SRE-01. The second system, Baseline-II, stems from (Tsai and Wang, 2005). It first specifies a certain number of clusters, corresponding to the size of one of the possible speaker populations, and then maximizes the

within-cluster homogeneity represented by the cluster purity[*]. The system then examines various legitimate numbers of clusters and uses BIC to determine the size of the most likely speaker population. The parameter values of Baseline-II were tuned using SRE-01 as well.

4.3 Experiment results

In GA optimization of MRIC, the parameter values used for the maximum number of generations, the chromosome population size, the crossover probability, and the mutation probability were empirically[†] determined to be 2000, 5000, 0.5, and 0.1, respectively. Tables 2, 3, and 4 show the speaker-clustering results obtained by Baseline-I, Baseline-II, and the proposed MRIC methods, respectively. The results in the "#Clusters = True #Speakers" column of each table were derived by taking the number of generated clusters as the true number of speakers in each excerpt. This serves as the upper bound of the speaker-clustering performance that can be achieved by determining the size of the speaker population automatically. From Tables 2 and 4, we observe that the MRIC method consistently yields larger purity values and smaller Rand index values than Baseline-I. This clearly demonstrates the superiority of the global optimization technique applied in MRIC over the pairwise optimization technique used in HAC. However, the results in Tables 2 and 3 show that it is hard to tell whether Baseline-II or MRIC performs better if the true speaker population is known *a priori*.

The "#Clusters = Estimated #Speakers" column in each table shows the speaker-clustering performance when the true speaker population size is unknown and must be estimated. The field "|ES – TS|" indicates the difference between the size of the estimated population and that of the true speaker population. Table 4 shows that the number of speakers estimated by MRIC for each excerpt is very close to the true speaker population size. In addition, for the estimated speaker population size, MRIC consistently yields smaller Rand index values than Baseline-I and Baseline-II. Overall, the values of |ES – TS| derived by MRIC are smaller than those derived by Baseline-I and Baseline-II, which indicates that MRIC improves the estimation of the speaker population size.

Comparing Tables 2 and 3, we observe that, although Baseline-II performs better than Baseline-I when the number of clusters is taken as the true speaker population size, several Rand index values derived by Baseline-II are larger than those obtained by Baseline-I when the optimal number of clusters is determined automatically. This is attributed to the inferior estimation of the speaker population size by Baseline-II. In contrast, MRIC jointly optimizes the generated clusters and the required number of clusters and thereby resolves the shortcomings of Baseline-I and Baseline-II.

4.4 Discussion on computational complexity

In addition to evaluating the effectiveness of MRIC and the two baseline systems, it is worth comparing their computational complexities. Since all the three systems begin with the measurement of inter-utterance similarities, we can ignore this common part in the analysis of computational complexity. In Baseline-I, the major operations include: the measurement of between-cluster similarities, search of the most similar clusters, and computation of BIC value before each merging process is performed. Since the first two operations involve logical comparisons only, their computational complexities are negligibly lower than that of the computation of BIC value, which involves the computation of covariance matrix and its determinant for every cluster. It is observed that, regardless of the number of clusters

[*] Cluster purity is also approximated by a function of the GLR-based similarities between utterances.

[†] This follows our previous work (Tsai and Wang, 2005), which performs optimization using SRE-01.

generated, the complexity of computing all the covariance matrices is on the order of the total number of feature vectors, T, multiplied by the dimensionality of the feature vectors, D. In addition, the complexity of computing a determinant of D-dimensional covariance matrix is on the order of $D\log^2 D$. If $D << T$, the complexity of computing the determinants is significantly lower than that of computing covariance matrices. Thus, the computational complexity of Baseline-I scanning from 1 cluster to N clusters can be characterized by $O(NTD)$. In MRIC, the major operations include: fitness evaluation, selection, crossover, and mutation in GA. Since the last three operations involve only logical comparisons and bit modifications, their computational complexities are significantly lower than that of the fitness evaluation, which involves a series of multiplications and additions. If the number of chromosomes is Z, and the maximum number of generations is Q, the computational complexity of MRIC can be roughly characterized by $O(NZQ)$, where N arises from the dimensionality of chromosome (i.e., the number of utterances to be clustered). Although it is hard to quantify $O(NTD)$ in Baseline-I and $O(NZQ)$ in MRIC as comparable values, we can see that MRIC has an advantage for clustering long utterances, since its computational complexity is independent of the utterance duration. Our experiments conducted on RT-02 show that the running time of MRIC is roughly double that of Baseline-I. As to Baseline-II, its operations are similar to MRIC, except that the determination of the optimal number of clusters needs to scan from 1 cluster to N clusters. Thus, the computational complexity of Baseline-II can be roughly characterized by $O(N^2ZQ)$, which is notably higher than that of MIRC.

5. Conclusions

We have investigated techniques for clustering speech data, whereby utterances from the same speaker can be grouped into a single cluster. This requirement is formulated as a problem of estimating and minimizing the clustering errors characterized by the Rand index. We represent the Rand index as a function of the inter-utterance similarities and apply a genetic algorithm to determine the index of the cluster in which each utterance should be located. The experiment results demonstrate that, in terms of speaker-clustering performance, the proposed approach significantly outperforms conventional methods that use the Bayesian information criterion to estimate the speaker population size.

It is worth noting that, in many applications, assigning a long utterance to the wrong cluster can be more detrimental than assigning a short utterance to the wrong cluster. To reflect this point and improve the clustering performance, we can also compute the Rand index at the frame level, instead of the utterance level used in this study. Specifically, the frame-based Rand index can be defined by how many times two randomly-selected frames from the same speaker are placed in different clusters, or two randomly-selected frames from different speakers are placed in the same cluster. Minimization of the frame-based Rand index is particularly useful in the speaker diarization task, since the diarization error is computed on a frame basis. In addition, it is worth emphasizing that the proposed MRIC can be applied to various clustering problems by simply substituting the appropriate similarity measurement into Eq. (15).

Finally, to speed up the proposed approach, we could take advantage of GA's intrinsic parallelism to perform distributed computing. There are a number of parallelization methods to enhance the computational speed of GA (Baluja, 1993; Zomaya et al., 1999), each reflects the fact that the nature of GA's population structure and recombination mechanisms are highly advantageous to distributed computing. Nevertheless, for some low-latency applications that require running on a single computing unit, there is still a need to develop on-line speaker clustering techniques as studied in Liu et al. (2003).

Acknowledgements

This work was supported in part by the National Science Council, Taiwan under Grants NSC94-2213-E-001-009, NSC95-2221-E-001-034, and NSC95-2218-E-027-020.

References

Ajmera, J., Bourlard, Lapidot, H. I., and McCowan, I., 2002. Unknown-multiple speaker clustering using HMM. In: *Proc. of the ICSLP 2002*.

Baker, J. E., 1985. Adaptive selection methods for genetic algorithm. In: *Proc. of the International Conference on Genetic Algorithms and Their Applications, 1985*.

Baluja, S., 1993. Structure and performance of fine-grain parallelism in genetic search. In: *Proc. of the 5th International Conference on Genetic Algorithm.*

Ben, M., Betser, M., Bimbot, F., and Gravier, G., 2004. Speaker diarization using bottom-up clustering based on a parameter-derived distance between adapted GMMs. In: *Proc. of the ICSLP 2004*.

Campbell, J. P., 1997. Speaker recognition: a tutorial. *Proc. IEEE, 85(9)*:1437-1462.

Chen, S. S. and Gopalakrishnan, P. S., 1998. Clustering via the Bayesian information criterion with applications in speech recognition. In: *Proc. of the ICASSP 1998*.

Faltlhauser, R. and Ruske, G., 2001. Robust speaker clustering in eigenspace. In: *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding 2001*.

Gish, H., Siu, M. H., and Rohlicek, R., 1991. Segregation of speakers for speech recognition and speaker identification. In: *Proc. of the ICASSP 1991*.

Goldberg, D. E., 1989. Genetic Algorithm in Search, Optimization and Machine Learning. New York: Addison-Wesley.

Hubert, L., and Arabie, P., 1985. Comparing partitions. *Journal of Classification* 2:193-218.

Jin, H., Kubala, F., and Schwartz, R., 1997. Automatic speaker clustering. In: *Proc. of the DARPA Speech Recognition Workshop 1997.*

Johnson , S. E. and Woodland, P. C., 1998. Speaker clustering using direct maximization of the MLLR-adapted likelihood. In: *Proc. of the ICSLP 1998*.

Johnson, S. E., 1999. Who spoke when?−Automatic segmentation and clustering for determining speaker turns. In: *Proc. of the Eurospeech 1999*.

Liu, Z., 2005. An efficient algorithm for clustering short spoken utterances. In: *Proc. of the ICASSP 2005.*

Liu, D. and Kubala, F., 2003. Online speaker clustering. In: *Proc. of the ICASSP 2003*.

Makhoul, J., Kubala, F., Leek, T., Liu, D., Nguyen, L., Schwartz, R., and Srivastava, A., 2000. Speech and language technologies for audio indexing and retrieval. *Proc. IEEE 88(8), 1338-1353.*

Moh, Y., Nguyen, P., and Junqua, J. C., 2003. Towards domain independent speaker clustering. In: *Proc. of the ICASSP 2003.*

Rand, W. M., 1971. Objective criteria for the evaluation of clustering methods. *J. Amer. Stat. Assoc. 66*:846-850.

Reynolds, D. A., Singer, E., Carson, B. A., O'Leary, G. C., McLaughlin, J. J., and Zissman, M. A., 1998. Blind clustering of speech utterances based on speaker and language characteristics. In: *Proc. of the ICSLP 1998.*

Schwarz, G., 1978. Estimating the dimension of a model. *The Annals of Statistics 6*:461-464.

Siegler, M. A., Jain, U., Raj, B. and Stern, R. M., 1997. Automatic segmentation, classification and clustering of broadcast news audio. In: *Proc. of the DARPA Speech Recognition Workshop 1997.*

Sinha, R., Tranter, S. E., Gales, M. J. F., and Woodland, P. C., 2005. The Cambridge University March 2005 speaker diarisation system. In: *Proc. of the Interspeech 2005*.

Solomonoff, A., Mielke, A., Schmidt, M., and Gish, H., 1998. Clustering speakers by their voices. In: *Proc. of the ICASSP 1998.*

Tranter, S. E., 2005. Two-way cluster voting to improve speaker diarisation performance. In: *Proc. of the ICASSP 2005.*

Tranter, S. E. and Reynolds, D. A., 2006. An overview of automatic speaker diarization systems. *IEEE Trans. Audio Speech Language Proc.*, 14(5): 1557 - 1565.

Tsai, W. H. and Wang, H. M., 2005. Speaker clustering of unknown utterances based on maximum purity estimation. In: *Proc. of the Interspeech 2005.*

Tsai, W. H. and Wang, H. M., 2006. Speech utterance clustering based on the maximization of within-cluster homogeneity of speaker voice characteristics. *J. Acoust. Soc. Amer. 120(3)*, 1631-1645.

Zhou, B. and Hansen, J. H. L., 2000. Unsupervised audio stream segmentation and clustering via the Bayesian information criterion. In: *Proc. of the ICSLP 2000*.

Zhu, X., Barras, C., Meignier, S., and Gauvain, J. L., 2005. Combining speaker identification and BIC for speaker diarization. In: *Proc. of the Interspeech 2005*.

Zomaya, A. Y., Ward, C., and Macey, B., 1999. Genetic scheduling for parallel processor systems: comparative studies and performance issues. *IEEE Trans. Parallel Distributed Systems*, 10(8):795 - 812.

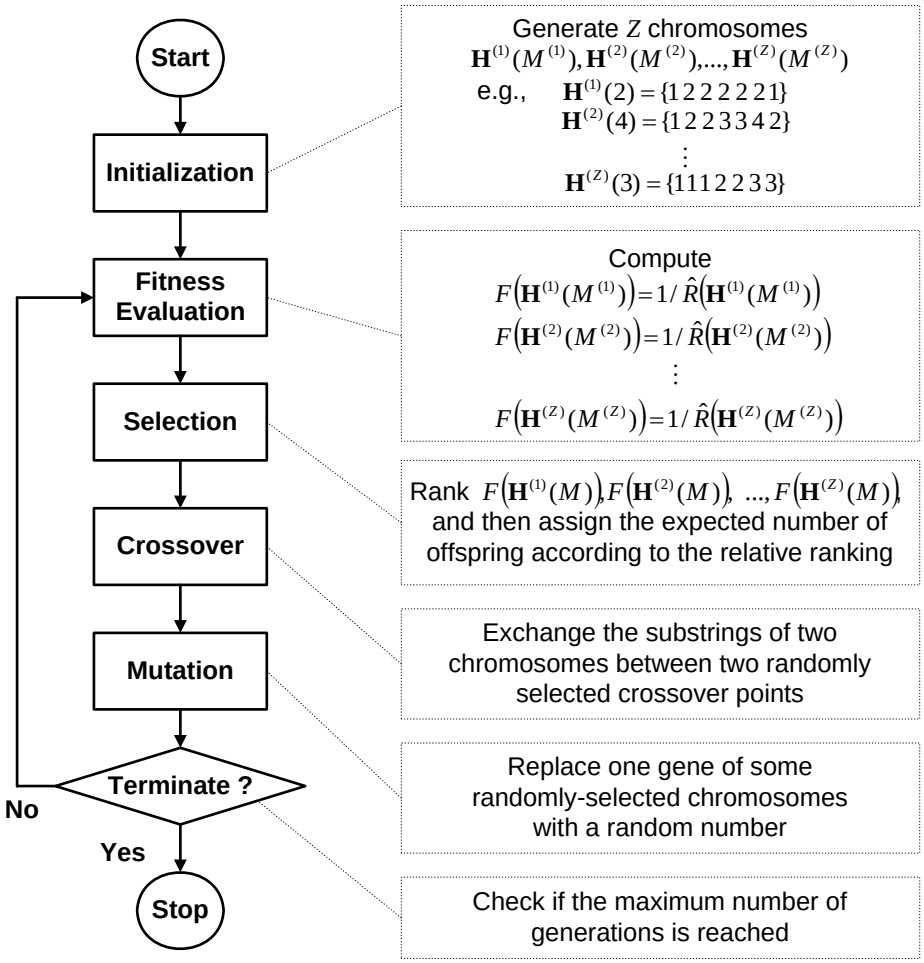

Fig. 1. Flow diagram of GA optimization

Table 1. RT-02 Speech data profile

Excerpt	No. of Utterances	Maximum/Minimum/Average Utterance Duration (in sec)	No. of Speakers
bn02en_1	44	67.8 / 0.5 / 13.0	16
bn02en_2	29	60.0 / 0.6 / 20.2	9
bn02en_3	13	84.8 / 5.0 / 37.8	6
bn02en_4	43	70.3 / 0.2 / 13.3	16
bn02en_5	26	65.0 / 5.0 / 23.1	10
bn02en_6	45	51.0 / 1.8 / 12.7	14

Table 2. Speaker-clustering results obtained by Baseline-I

Excerpt	True # Speakers (TS)	# Clusters = True # Speakers		# Clusters = Estimated # Speakers		
		Purity	Rand Index (in %)	Estimated # Speakers (ES)	|ES – TS|	Rand Index (in %)
bn02en_1	16	0.89	17.6	8	8	40.1
bn02en_2	9	0.94	6.7	13	4	21.1
bn02en_3	6	1.00	0.0	6	0	0.0
bn02en_4	16	0.90	21.3	18	2	29.7
bn02en_5	10	0.72	27.9	11	1	31.9
bn02en_6	14	0.86	12.8	15	1	16.6

Table 3. Speaker-clustering results obtained by Baseline-II

Excerpt	True # Speakers (TS)	# Clusters = True # Speakers		# Clusters = Estimated # Speakers		
		Purity	Rand Index (in %)	Estimated # Speakers (ES)	|ES – TS|	Rand Index (in %)
bn02en_1	16	0.95	9.4	22	6	23.6
bn02en_2	9	0.94	6.6	14	5	24.2
bn02en_3	6	1.00	0.0	8	2	11.7
bn02en_4	16	0.89	22.6	14	2	28.6
bn02en_5	10	0.80	18.9	13	3	33.9
bn02en_6	14	0.84	17.8	14	0	17.8

Table 4. Speaker-clustering results obtained by MRIC

Excerpt	True # Speakers (TS)	# Clusters = True # Speakers		# Clusters = Estimated # Speakers		
		Purity	Rand Index (in %)	Estimated # Speakers (ES)	|ES – TS|	Rand Index (in %)
bn02en_1	16	0.93	9.6	17	1	12.5
bn02en_2	9	0.95	6.1	11	2	11.6
bn02en_3	6	1.00	0.0	6	0	0.0
bn02en_4	16	0.91	17.1	15	1	19.5
bn02en_5	10	0.76	22.4	11	1	26.1
bn02en_6	14	0.88	10.4	15	1	14.3