Unsupervised speaker recognition based on competition between self-organizing maps — Source link ⬈

Itshak Lapidot, Hugo Guterman, Arnon D. Cohen

Institutions: Ben-Gurion University of the Negev

Related papers:

- Speaker Verification Using Adapted Gaussian Mixture Models

- The self-organizing map

- Speaker recognition: a tutorial

- Fundamentals of speech recognition

- Robust text-independent speaker identification using Gaussian mixture speaker models

Unsupervised Speaker Recognition Based on Competition Between Self-Organizing Maps

Itshak Lapidot (Voitovetsky), Hugo Guterman, and Arnon Cohen

Abstract—We present a method for clustering the speakers from unlabeled and unsegmented conversation (with known number of speakers), when no *a priori* knowledge about the identity of the participants is given. Each speaker was modeled by a self-organizing map (SOM). The SOMs were randomly initiated. An iterative algorithm allows the data move from one model to another and adjust the SOMs. The restriction that the data can move only in small groups but not by moving each and every feature vector separately force the SOMs to adjust to speakers (instead of phonemes or other vocal events). This method was applied to high-quality conversations with two to five participants and to two-speaker telephone-quality conversations. The results for two (both high- and telephone-quality) and three speakers were over 80% correct segmentation. The problem becomes even harder when the number of participants is also unknown. Based on the iterative clustering algorithm a validity criterion was also developed to estimate the number of speakers. In 16 out of 17 conversations of high-quality conversations between two and three participants, the estimation of the number of the participants was correct. In telephone-quality the results were poorer.

Index Terms—Competitive learning, segmentation, self-organizing maps (SOMs), speaker recognition, temporal data clustering, vector quantization (VQ).

I. INTRODUCTION

THIS paper describes a system for unsupervised speaker recognition (otherwise known as "speaker segmentation"), based on the piecewise-dependent-data (PDD) clustering method. Most speaker recognition problems have been solved by using supervised methods. A survey of issues and methods regarding supervised speaker recognition can be found in [1]–[3]. A training data for each speaker is given *a priori* and a model of each speaker is produced. Supervised methods have been applied for speaker identification and verification, for example, for entering computers or security sites by vocal passwords. A less common problem is unsupervised speaker recognition (speaker segmentation), in this case, no training set is given and the data is unlabeled. Since no labeled training data is available, the unsupervised training is performed by initially clustering the data into different clusters where each cluster, is assumed to represents a different speaker. Unlike most clustering approaches where each vector is associated with a specific cluster, here a sequence of vectors has to be associated

with the same cluster. We name this case PDD clustering. Unsupervised speaker recognition has many applications; the most common of which is probably that of audio browsing [4], [5].

Most PDD clustering problems, unlike the unsupervised speaker recognition problem, try to separate the signal into several stationary models (such as AR models). However, a stationary model cannot describe a speaker and therefore another approach must be taken. In the approach that was used in this research, every speaker cluster was described by a Code-Book (CB). An algorithm was developed to train all the CBs, such that every CB represented a different speaker. Then an iterative competitive algorithm between all the CBs was applied. Each CB was created using a Kohonen self-organizing map (SOM) [6], [7]. The convergence of the algorithm, in terms of distance minimization, was proved and a validity criterion was developed to determine the number of speakers in a given conversation.

The complexity of a speaker recognition problem depends on the population size and the duration of the speech segment. Furthermore, supervised speaker recognition problems depend on whether they are text-dependent or text-independent, on whether the set is closed or open, and on whether the problem is to identify or to verify the speaker. In the unsupervised case, knowledge or lack of knowledge about the boundaries of each speech segment influences the problem's complexity. In addition, speaker recognition problems depend on the signal bandwidth (the telephone line bandwidth), environmental noise, whether or not a real time problem is involved, and the equipment in use, such as the speed and resolution of the sampler, the microphone type, etc.

PDD clustering has many applications in time-signals including speaker recognition [4], [5], [8]–[18], machine monitoring [19], switching chaos [20]–[22], prediction of systems output [21], clustering of EEG signals [22], and music clustering [23]. Similar methods have also been applied in other areas, such as protein modeling [24].

PDD clustering must be used when there is a successive dependence between a group of data vectors. An additional problem of PDD clustering is to determine the point of change between the segments (each segment belongs to a different cluster). Clustering algorithms are applied in the time domain and in the feature space as well. Sometimes the transitions between the models are not sharp (e.g., one model appears before the end of the previous one) which is known as drifting dynamics [22]. In this case, it is necessary to find the transients and to give membership weights to each cluster at every time point.

Many approaches have been applied for PDD clustering, e.g., the Dendrogram [8]; the vector quantization (VQ) algorithms [9], [10], the expectation–maximization (EM) algorithm [4], [5] and [11], HMM [12]–[15], [19], [24], [25], hidden Markov network (HMnet) + VQ [16], neural networks (NNs) [17], [18], [20]–[23], maximum *a posteriori* (MAP) probability estimation [25], and fuzzy logic [26], [27].

Additionally to the PDD clustering method a validity criterion to estimate the number of clusters (speakers), R, is suggested. For this criterion we define a conditional distance between two CBs. The distance depends not only on the CBs but on the input vector as well [18].

The system we present gets as input an unsegmented and unlabeled conversation, with unknown number of speakers. The output is the estimation of the number of the participants, R, segmentation of the conversation and labeling according to the R clusters that were estimated.

Preliminary results were already presented in [17], [18]. Specifically, in [17] the principles of the basic segmentation algorithm were presented and the system was tested with a reduced data set. In [18] a method or criteria for the determination of the number of speakers was described and tested. In this paper the complete segmentation system was presented and its performance truthfully evaluated. The proposed system was tested with on data from two Hebrew databases. The first database was high quality, recorded at an acoustic room. This database includes conversations between two to five speakers. The second database includes telephone-quality two-speaker conversation. Different aspects of the system are discussed and a comparison with other segmentation systems is presented.

II. Systems Description

In general, given a conversation the goals are to estimate the number of participating speakers, R, and to cluster the conversation into R clusters. Fig. 1 shows the block diagram of the PDD clustering algorithm (where R is given). First the procedure for distortion-based-models will be presented in Section II-A. In Section II-B a proof of the systems' convergence will be presented (a detailed proof is given in [28]). The validity criterion is presented in Section II-C. A description of Kohonen SOM that is used as distortion-based-models is given in Section II-D. The description of the entire system is summarized in Section II-E. Section II-F describes the adjustment of the system to unsupervised speaker recognition task.
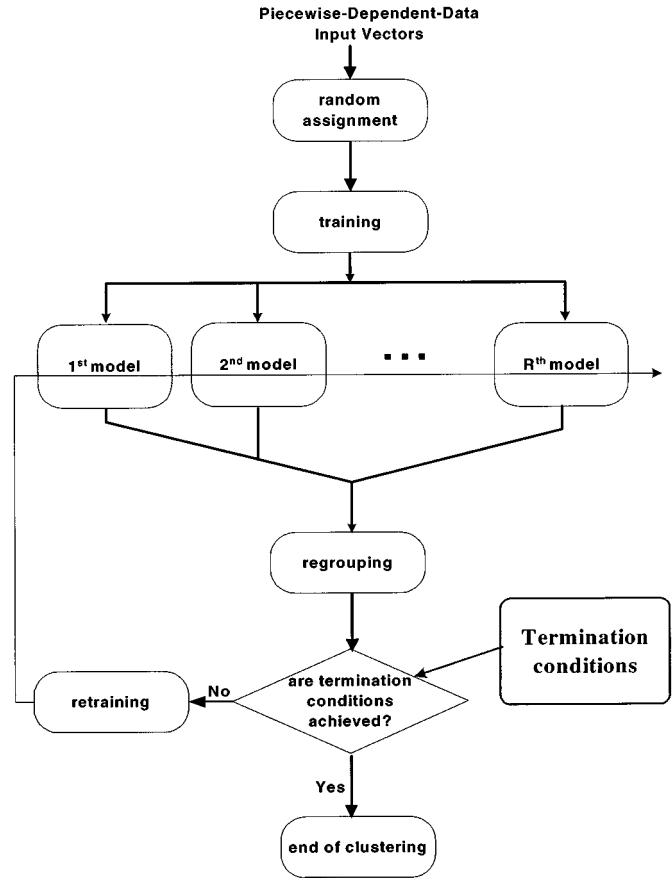

Fig. 1. Piecewise-dependent-data system (R given).

A. Distortion Measure-Based Model Clustering Algorithm

The goal of the algorithm is to cluster the input data into R clusters. The assumption is that the switching points between the segments are all known (the switching points refer to the boundary between two adjacent segments), so the clustering algorithm can be defined as follows. The PDD consists of N vectors, $\mathbf{V} = \{v_n\}_{n=1,\ldots,N}$. These vectors are partitioned into M segments, $\mathbf{V} = \{\mathbf{v_m}\}_{m=1,\ldots,M}$ (1) according to the switching points. The segments have to be clustered into R clusters, i.e., two vectors that belong to the same segment must be clustered to the same cluster. Each cluster is defined by a distortion-measure-based model. A CB is created, for each model, using VQ algorithm. Every CB_r, is of size L_r, and it presents the rth cluster shown in (2) at the bottom of the page, where $\{c_r^l\}_{r=1,\ldots,R;\ l=1,\ldots,L_r}$ is the union of all the codewords (CW) that belong to CB_r.

$$\begin{cases} \mathbf{V} = \left\{ \underbrace{v_1^1,\ldots,v_{n_1}^1}_{\mathbf{V_1}}, \ldots, \underbrace{v_1^m,\ldots,v_{n_m}^m}_{\mathbf{V_m}}, \ldots, \underbrace{v_1^M,\ldots,v_{n_M}^M}_{\mathbf{V_M}} \right\} = \{\mathbf{v_m}\}_{m=1,\ldots,M} \\ \sum_{m=1}^{M} n_m = N \end{cases} \tag{1}$$

$$CB_r = \{c_r^l\}_{r=1,\ldots,R;\ l=1,\ldots,L_r} \tag{2}$$

The initiation of the process is performed by randomly assigning equal number of segments to all CB_rs ($\mathbf{V^{r,0}}$-segments that partitioned to CB_r at the beginning). Each model is trained using the data assigned to it during the partitioning. After the training the regrouping process is applied. The distortion between each segment and each model is calculated and the new segment attribution is given according to the minimal distortion. The regrouping process produces a new partition and the models is retrained again. After i iterations the partition will be

$$
\begin{cases}
\mathbf{V^{r,i}} = \left\{\mathbf{v_m^{r,i}}\right\}_{m=1,...,M_{r,i};\; r=1,...,R} \\
\mathbf{v_m^{r,i}} = \left\{v_{m,n}^{r,i}\right\}_{n=1,...,n_m;\; m=1,...,M_{r,i};\; r=1,...,R} \\
\sum\limits_{r=1}^{R} M_{r,i} = M
\end{cases}
\tag{3}
$$

and the code-books are

$$
CB_r^i = \left\{c_r^{l,i}\right\}_{r=1,...,R;\; l=1,...,L_r}.
\tag{4}
$$

The VQ training can use any VQ algorithm, such as the LBG [29], SOM [7], fuzzy C-means [30], etc.

After the retraining the data regroup again by checking which CB_r^i best fit every $\mathbf{v_m} \in \mathbf{V}$ according to a given distance measure and a new partition is produced, $\mathbf{V^{r,i+1}}$

$$
\arg\min_{r=1,...,R} \left\{D_m^i(r)\right\} \Rightarrow \mathbf{v_m} \in CB_j^i
\tag{5}
$$

where $D_m^i(r)$ is the distance between mth segment and rth CB at the ith retraining.

The system has to be retrained according to the new partition, and the termination condition is met when

$$
\mathbf{V^{r,i}} = \mathbf{V^{r,i+1}}.
\tag{6}
$$

Hence, an iteration of the process is defined as follows:

1) Retrain the models with the new partition achieved by the previous iteration.
2) Regroup the data using (5).
3) Test for termination: if the termination criterion is met, exit; if not return to 1).

At the end of this iterative procedure, the system provides R models, for the R clusters.

Different termination conditions can be applied. In the present work the following termination criterion was applied:

$$
\frac{\text{Number of segments that change their assignment}}{\text{Total number of segments}} \times 100 \leq 3\%.
$$

B. Algorithms' Convergence Proof

The algorithm that was presented in Section II-A described the iterative training procedure. It is necessary to know if this process converges. In this subsections a general proofs for the convergence of the process will be given when each model is a VQ (more detailed proof can be found at [28]).

Given M segments, $\mathbf{V} = \{\mathbf{V_m}\}_{m=1,...,M}$, it is required to separate them into R clusters. Every CB_r, with L_r size, presents the rth cluster.

To proof the convergence of the proposed algorithm the applied VQ algorithm (e.g., LBG [29] or SOM [7]) must converge at least to a local minimum.

After the ith iteration the partition of the data between the models will be according to (3), where $\mathbf{V^{r,i-1}}$ is the data set associated with the rth model at the $(i-1)$th iteration, and the CB_r at the ith iteration is $CB_r^i = \left\{c_r^{l,i}\right\}_{r=1,...,R;\; l=1,...,L_r}$ (note that CB_r^i is the code-book that was produced using the previous partition $\mathbf{V^{r,i-1}}$).

The initial partition $\mathbf{V^{r,0}}$ can be chosen randomly or in some other way.

Let the distance between $v_{m,n}^{q,i-1}$ and CB_r^i be $d_{m,n}^i(r,q)$. The distance between $\mathbf{v_m^{q,i-1}}$ and CB_r^i is

$$
D_m^i(r,q) = \sum_{n=1}^{n_m} d_{m,n}^i(r,q)
\tag{7}
$$

and the minimal distance between the segment that belongs to CB_q^{i-1} from all the CBs is

$$
D_m^i(j) = \min_{r=1,...,R} \left\{D_m^i(r,q)\right\}
$$
$$
j = \arg\min_{r=1,...,R} \left\{D_m^i(r,q)\right\} \Rightarrow \mathbf{v_m^{q,i-1}} = \mathbf{v_m^{j,i}}.
\tag{8}
$$

If after the ith iteration, the overall distance is calculated with the old partition be D^i, and after the regrouping (new partition) $\tilde{D}^i$. It is easy to show that the next inequality holds

$$
\tilde{D}^i \leq D^i \leq \tilde{D}^{i-1}.
\tag{9}
$$

In other words, there exists a better partition of $\mathbf{V}$ that gives a lower distance $\tilde{D}^i$. If the new partition is chosen, then the previous VQ is not optimal because it was trained according to the other partition. It can be seen that from the ith to $(i+1)$th iteration the overall distance do not increase. The iterative process will stop when

$$
\begin{cases}
\tilde{D}^{i+1} = D^{i+1} \\
\mathbf{V^{r,i+1}} = \mathbf{V^{r,i}}.
\end{cases}
\tag{10}
$$

In this case there is no change in the partition between the two consecutive iterations.

C. Validity Criterion

In a good clustering the intracluster distance should be small while intercluster distance should be large. It is therefore logical to define the validity of a given partition to be proportional to the ratio between clusters' intradistances and intercluster distance.

Lets R be the unknown number of clusters, $R_1 \leq R \leq R_2$, where R_1 and R_2 are some given bounds of R. The estimated number of clusters will be the one that minimizes a certain validity criterion.

Let

M_r—the number of segments that belong to the rth cluster.

n_m—the number of vectors in the mth segment.

v_n—an input vector of CB_r; $v_n \in CB_r$.

$d_n(r)$—the distance between v_n and CB_r.

$\text{Dcb}_n(r,p)$—the distance between CB_r and CB_p given v_n.

If $c_n(r)$ is the closest codeword of the CB_r to v_n, and $c_n(r,p)$ is the closest codeword of CB_p to $c_n(r)$, $\mathrm{Dcb}_n(r,p)$ is the Euclidean distance between $c_n(r)$ and $c_n(r,p)$.

$$\mathrm{Dcb}_n(r,p) = \left[(c_n(r) - c_n(r,p))^T (c_n(r) - c_n(r,p))\right]^{1/2}. \tag{11}$$

From Fig. 2, it can be seen that $\mathrm{Dcb}_n(r,p)$ is not the distance of the closest codeword of CB_r and CB_p. In other words $\mathrm{Dcb}_n(r,p)$ can be named as conditional distance between CB_r and CB_p given v_n.

The contribution of the rth cluster to the validity coefficient, Q_r^R, will be define as

$$Q_r^R = \frac{1}{M_r} \sum_{m \in r} \frac{1}{n_m} \sum_{n=1}^{n_m} \frac{d_n(r)}{\sum_{p=1,\dots,R \; p \neq r} M_p \mathrm{Dcb}_n(r,p)}. \tag{12}$$

The validity coefficient of the R clusters partition will be a sum of all the contributions, Q_r^R

$$Q^R = \sum_{r=1}^{R} Q_r^R. \tag{13}$$

To estimate the number of clusters it is needed to find $\{Q^r\}_{r=R_1,\dots R_2}$. The estimated number $\hat{R}$ of clusters will be

$$\hat{R} = \operatorname*{arg\,min}_{r=R_1,\dots,R_2} \{Q^r\}. \tag{14}$$

One of the most popular ways to reduce the number of clusters is multilevel dendrogram cutting [8], [11]. In this method the algorithm starts with a large number of clusters. At each stage the algorithm finds the two closest clusters and merges them. The process continues until the final number of clusters is achieved. The assumption of this method is that if two clusters have been merged they cannot be separated again.

In this work, clusters are not merged but rather reduced. The reduction of cluster cause to the data of the reduced cluster to be divided among the other clusters according to minimal distance between each segment and all the remaining models, i.e., not all the segments of the reduced cluster are added to the same model. Two ways were checked for cluster reduction. The first is the "knock one cluster out" method. Each time a cluster was knocked out, regrouping process of its segments was applied, and total segmentation distortion was calculated. The cluster that was removed is the one whose removal caused the smallest distortion. The second method was to choose the one with the minimum speech duration and regroup its segments. There were no differences in the validity or clustering results, but the second method proved to be much faster. As the second method was much faster all the results are related to the use of the second approach of model reduction.

The validity coefficients were calculated as follows. First, the data was clustered into R_2 clusters and (13) was employed to calculate Q^{R_2}. Then, one cluster was reduced applying the cluster reduction algorithm above described. The $R_2 - 1$ clusters were retrained, and the validity coefficient was calculated again. This process of cluster reduction, retraining, and validity coefficient calculation continue until the number of clusters reaches
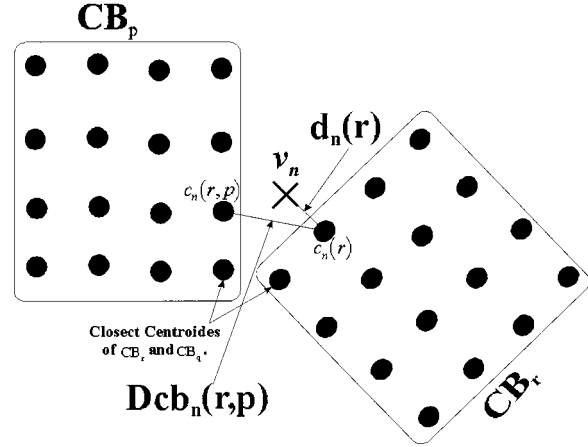

Fig. 2. The distance between two codebooks as a function of the input vector.

R_1 (the minimum number of clusters allowed). Equation (14) was used to estimate the $\hat{R}$ number of clusters.

D. Kohonen SOM

In this work, the algorithm that was applied for producing the VQ of each model, was SOM [6], [7]. The SOMs' structure can be seen in Fig. 3. The training algorithm of the SOM says that if the winner neuron to a given input at iteration t is $\mathbf{m_c^t}$ than it is necessary to adapt not only the winner neuron but also all the neurons in its neighborhood, $N_c(t)$. The area of the lateral interaction is called the neuron's neighborhood (N) and the winner neuron index is c.

The learning algorithm of the Kohonen SOM as applied here is presented below.

Let $\mathbf{V} = \{\mathbf{v}_n\}_{n=1}^N$, $\mathbf{v}_n = \begin{bmatrix} \mathbf{v}_n^1 & \mathbf{v}_n^2 & \cdots & \mathbf{v}_n^p \end{bmatrix}^T$ be the training data set, and let $\{\mathbf{m_k^t}\}_{k=1,\dots,K}$, $\mathbf{m_k^t} = \begin{bmatrix} m_k^{1,t} & m_k^{2,t} & \cdots & m_k^{p,t} \end{bmatrix}^T$ be the SOM's neurons. The unsupervised training algorithm is:

1) Initiate the neurons' weights with "small" random values.
2) Randomly choose a vector, $\mathbf{v_n}$, from the training data set.
3) Find $c = \arg\min_k \left\{ \begin{bmatrix} \mathbf{v}_n - \mathbf{m_k^t} \end{bmatrix}^T \begin{bmatrix} \mathbf{v}_n - \mathbf{m_k^t} \end{bmatrix} \right\}$.
4) Update the SOM by updating the neurons $\{\mathbf{m_k}\}_{k=1,\dots,K}$ at the iteration $t+1$:

$$\begin{cases} \mathbf{m_k^{t+1}} = \mathbf{m_k^t} + h_{ck}(t) \begin{bmatrix} \mathbf{v}_n - \mathbf{m_k^t} \end{bmatrix} & \text{if } k \in N_c(t) \\ \mathbf{m_k^{t+1}} = \mathbf{m_k^t} & \text{if } k \notin N_c(t). \end{cases} \tag{15}$$

$h_{ck}(t)$ is an updating function, at iteration t.
5) If the number of iterations is equal to ten times the number of training input vectors, exit; if not return to step 2).

$h_{ck}(t)$ and $N_c(t)$ are monotonically nonincreasing functions. Usually the training process consists of two phases. The first is the "fast training" phase, which involves about 10% of the entire training process. In this phase, $h_{ck}(t)$ and $N_c(t)$ start from a high value and decrease very quickly. In the second phase (the tuning phase), $h_{ck}(t)$ and $N_c(t)$ are small and decrease slowly to zero. The number of iterations employed was ten times the number of training input vectors $(10 \times N)$.
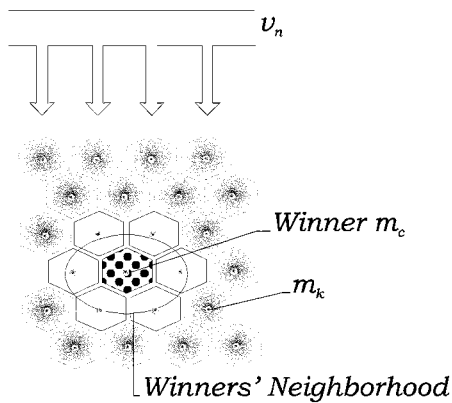
Fig. 3. Kohonen SOM.

E. The Complete System

The entire process can be described as follows:

1) Determine the minimal number of cluster, R_1, and maximal number R_2.
2) Set $R = R_2$. Randomly and equally divide the segments between the models as initial training data.
3) Cluster the data for R clusters, using Kohonen SOM as clusters model.
4) Save the segmentation, labeling, and calculate the validity coefficient.
5) If $R = R_1$—find $\hat{R}$ (minimal validity coefficient) and use the segmentation and labeling. Else—set $R = R - 1$ and return to step 3).

F. Adjustment to Unsupervised Speaker Recognition

For the PDD clustering that was described in Section II-A, it was necessary to know the start and end points of each segment. In reality this information is not available. For this reason we cut the data into segments of fixed length (this length was defined empirically and described in the first experiment). This fact cause some of the segments to split between two or more clusters (speakers or speaker and nonspeech event). Additionally to increase the resolution of the segmentation the segments were overlapping one each other (75% overlapping).

In each conversation the data includes, in addition to speech data, nonspeech events as silence, chair movement, coughing, etc. For this purpose additional cluster was created for nonspeech events.

The general block diagram of the clustering system of speaker conversation is shown in Fig. 4. The preprocessing of the sampled data includes pre-emphasize HPF. The filtered data was framed into 15 ms frames with 10 ms overlapping (5 ms frame rate). Each frame was multiplied by a Hamming window and the feature extraction of the speech data was performed. The features were twelfth-order LPC based cepstrum and twelfth-order del-cepstrum coefficients, and the mean absolute values of accumulated 50 ms frames were calculated for speech/nonspeech evaluation. Preliminary segmentation of speech and nonspeech data was performed by thresholding the absolute value feature. The threshold level was set at 3% of the maximum for high-quality speech and 1% for telephone data. These levels were determined experimentally. Although setting
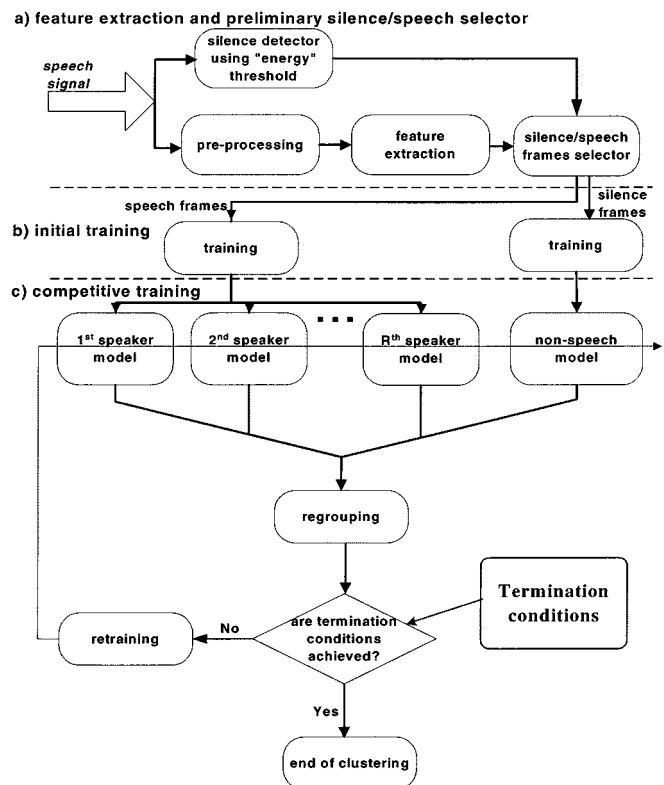

Fig. 4. General description of the unsupervised speaker classification system.

a higher level for high-quality speech might seem illogical, it can be justified by the fact that in high-quality data, the variance of the speech amplitude is much lower than that of telephone speech.

The initial conditions for the system were determined as follows: all segments classified by the crude speech/nonspeech classifier as nonspeech were used to train the nonspeech network. Segments crudely classified as speech segments were randomly and equally divided and used to train the R speaker models.

III. EXPERIMENTS AND RESULTS

The speech database employed in this research is composed of Hebrew conversations. The data was recorded in two ways: the first method was to record the conversations in an acoustic room (Table I). Nine males and one female took part in these conversations. The sampling parameters were 16 kHz sampling rate, 12 bits/sample, 50 Hz–7.8 kHz antialiasing filter. The second data set consisted of telephone quality conversations. Twenty-four male speakers participated in 12 two-speaker conversations. The data was sampled with 8 kHz sampling rate, 12 Bits/Sample, 0–3.8 kHz antialiasing filter. The SNR was approximately 35 dB and 25 dB for high- and telephone-quality conversations, respectively.

Clustering conversations between four and five participants, mostly, produced very poor results. The low performance of the clustering might be related to the nonuniform distribution of the data between the participants. The distribution of the speech data between the speakers is described in Table II. Other factors that might have affected the clustering results are the amount of nonspeech data and its structure (e.g., silence,

TABLE I
HIGH–QUALITY CONVERSATIONS

Number of speakers	Number of Conversations
2	12
3	5
4	6
5	2

laughter, chair movements, and simultaneous speech that was included in the nonspeech data), and the length of the speech segments (Table II).

In this section, we present the results of the experiments. Some of the results of these experiments have already been published [17], [18]. Five experiments were performed in order to test the ability of the SOM-based system to cluster two to five speakers, and to evaluate the validity criterion. The experiments are summarized in Table III. Each experiment had at least five repetitions.

Experiment no. 1: The goal of this experiment was to test the ability of the SOM to cluster two speakers, and to determine the optimal SOMs' size and speech segment duration for clustering. For this experiment, two-speaker high-quality conversations were employed. The silence segments were removed and the clustering was produced by using only speech data. The length of the conversations varied between 79 to 198 s and the speech duration between 72 to 180 s (the rest of the data was nonspeech). The worst ratio between the speakers' speech duration was 33/59 for a conversation of 92 s duration. The shortest speech duration of a speaker for one conversation was 32 s.

The system performances were tested with segments that were 0.25 s, 0.5 s, 0.75 s and 1 s in length. The sizes of the SOMs that were used were 3×5, 5×6, 5×8 and 6×10.

In this work a sliding window with constant length (constant segment length) was employed. Therefore, we could not assure that all the frames in a given segment belonged to the same speaker. A segment that contains data either from more than one speaker or speech and nonspeech data was defined as a splitting segment.

Six out of the 12 high-quality conversations were used for this experiment. The results for all the conversations were quite similar and were not affected by the SOM size or segment length. Clustering errors are summarized in Table IV.

The results in the table show the following: 1) the error rate at the splitting segment is higher than for the non-splitting segments; 2) the error rate of the short segments (0.25 s) is much higher than in long segments when small SOM is applied as speakers model (see results of a 3×5 SOM). If the SOM is large (see results for a 6×10 SOM) there were no difference in the error rate; and 3) as the segments became shorter, the appearance of the splitting segment became more rare, and therefore the influence of the splitting segments decreased.

One problem with segments of 0.25 s was the time of convergence. It took 80–300 iterations while only 25–50 iterations were needed for 0.5-s segments.

Until this point the shortest conversation was 72 s in length. In order to explore the influence of conversation length on the clustering performances, first 60 s, 50 s, and 40 s length were used in that order. The results for 0.25-s segments always had over a 30% error rate. For 0.5-s segments error rate was less than 10% for 60 s and 50 s conversations, and less than 15% for 40 s length. Note that the errors include split segments.

Because of the time of convergence and the large error for short conversations using 0.25-s segments, the chosen segment length for the next experiment was 0.5 s.

Experiment no. 2: The goal of this experiment was to examine the performance of the clustering algorithm using high-quality conversations, using the SOM size and segment length determined in the first experiment (R was assumed to be known). All the high-quality conversations included nonspeech segments were used for training and error evaluation. An additional SOM was used to represent the nonspeech model. A description of the clustering system can be found in Section II-F. The number of speakers is assumed to be known.

The preliminary automatic segmentation of the speech/nonspeech algorithm was employed. An empirical threshold was found but it turned out to be not very accurate. A comparison between 60 s of preliminary speech/nonspeech segmentation, final segmentation, and manual segmentation is shown in Fig. 5. It can be seen that the improvement is very impressive. Because the segments were half a second in duration it is possible that some may contain speech and nonspeech. The resolution of the SOM segmentation was half a second, and since manual segmentation can change at each point some of the errors observed might be attributed to finite resolution.

a) *Two speakers:* For the high-quality data conversations (between two males), the error rate was always less than 6.0%. An example of the confusion matrix is given in Table V. Three of the conversations were between a male and a female, for which the error rate was approximately 4.3%.

b) *Three speakers:* Conversations between three speakers always converged and the results were not worse than 15%.

c) *Four speakers:* All the clustering results of four speakers (conversations 2–6 in Table II) do not converged to meaningful clusters, except for the first conversation. The clusters share the data of several speakers or one speaker occupies more than one model. Table II shows that the most uniformly divided data is found in the first and the fifth conversations. The amount of data per speaker was at least 60 s. Two other factors that affected the clustering performance were the amount of nonspeech data (including simultaneous speech data) and the average segment length (see Table II). The average segment length was at least a second longer in the

TABLE II
DISTRIBUTIONS OF FOUR- AND FIVE-SPEAKER CONVERSATIONS (SPK.—SPEAKER, SIM.—SIMULTANEOUS)

Conversation Number	1	2	3	4	5	6	7	8
Number of Speakers	4	4	4	4	4	4	5	5
Spk. No. 1 Percents [%]	29.5	31.5	46.3	21.8	23.6	37.9	23.6	26.8
Spk. No. 2 Percents [%]	27.4	23.0	16.1	31.5	20.5	6.2	18.1	14.8
Spk. No. 3 Percents [%]	19.3	13.3	13.2	18.2	17.2	16.2	14.5	11.3
Spk. No. 4 Percents [%]	15.2	14.9	8.7	7.9	18.5	17.0	16.5	7.0
Spk. No. 5 Percents [%]	–	–	–	–	–	–	9.4	11.3
Non-Speech Percent [%]	8.7	17.4	15.7	20.7	20.3	22.7	17.8	28.8
Total Time in Sec	433.4	771.7	780.3	164.0	268.2	350.8	404.7	348.4
The Mean of the Speech Segments [Sec.]	3.4	2.1	2.3	2.1	2.4	2.2	2.1	1.9
Remarks	–	–	–	Sim. Speech	Sim. Speech	Sim. Speech	Sim. Speech	Sim. Speech

TABLE III
SOM-BASED EXPERIMENTS

# Ex.	No. of Speakers	Speech Quality	Tests		SOM Size	Segment Length [sec]	Including Non Speech Data
			Clustering	Validity			
1	2	High	√	—	Variable	Variable	No
2	2-5	High	√	—	6×10	0.5	Yes
3	2-4	High	√	√	6×10	0.5	Yes
4	2	Telephone	√	—	6×10	0.5	Yes
5	2	Telephone	√	√	6×10	0.5	Yes

first conversation than in the other conversations. As the duration of the speech segments became shorter, more split segments participated in training the models, which caused degradation in the models' fit to the correct parameters (i.e., split segments cannot accurately train any model). In summary, wrong clustering can occur for several reasons: nonuniform data distribution, a small amount of data per speaker, the presence of nonspeech data (particularly simultaneous speech), and short segments.

The fourth speaker of the fifth conversation was a female. She and the first speaker each got one model. The other two models were mixed and belonged to the other two participants. Clustering for three clusters yielded one male model, one female model, and one mixed model of the remaining speakers with 10% interference by the first male and the female. The data of the mixed cluster was clustered again for two clusters and the results had about 80% accuracy.

d) *Five speakers:* As in the four-speaker case, the conversations had many nonspeech segments including simultaneous speech and a comparatively short average segment length. The eighth conversation was highly nonuniformly distributed (see Table II). The results were not good in that usually more than one speaker belonged to one model, and because of the large amount of nonspeech data, that covered different places in the feature-space (e.g., silence and simultaneous speech), more than one model described the nonspeech data.

As can be seen, it is very difficult to record a spontaneous conversation that includes more than three participants such that the data will be "close" to a uniform distribution. The speaker who talks most of the time will usually be separated from the others, and at the same time will occupy more than one model.

In this experiment the error rate was found to be less than 6% and 15% for two and three speakers, respectively. Only one conversation out of six con-

TABLE IV
EXAMPLE OF SEGMENTATION RESULTS (EXPERIMENT 1) AS A FUNCTION OF THE SEGMENT LENGTH. THE MEAN AND THE STANDARD DEVIATION (STD IN PARENTHESIS) VALUES ARE PROVIDED

Error (in %) With Splitting Segments – Mean (STD)				Error (in %)Without Splitting Segments – Mean (STD)				SOM Size
Segments Duration								
0.25 sec	0.5 sec	0.75 sec	1 sec	0.25 sec	0.5 sec	0.75 sec	1 sec	
9.1 (2.3)	4.6 (1.1)	7.6 (5.3)	6.2 (1.0)	7.6 (2.5)	0.6 (0.5)	4.3 (5.2)	0.2 (0.3)	3×5
6.3 (4.1)	5.4 (1.5)	9.0 (4.3)	6.3 (0.7)	4.7 (4.2)	2.3 (1.3)	5.0 (5.4)	0.7 (0.6)	5×6
4.6 (1.4)	4.8 (1.8)	6.5 (2.1)	6.8 (1.3)	3.0 (1.4)	1.7 (1.9)	2.3 (1.4)	0.7 (0.9)	5×8
4.8 (1.8)	5.9 (2.8)	7.4 (2.3)	7.2 (1.2)	2.9 (1.5)	2.8 (2.6)	4.2 (2.2)	1.7 (1.7)	6×10
6.2 (3.1)	5.2 (2.0)	7.6 (3.9)	6.6 (1.1)	4.5 (3.3)	1.9 (1.9)	4.0 (4.1)	0.8 (1.1)	Mean

verged to a meaningful clustering for four speakers while the system never converged to meaningful clustering for five speakers.

Experiment no. 3: The goal of this experiment was to test the validity criterion using high-quality conversations. The conditions were similar to the previous test but the number of speakers was unknown (suppose to be between two to six speakers). The criterion for estimating the number of speakers and the cluster reduction algorithm were described in Section II-C.

When the clustering was checked for the correct number of speakers, the results were the same as in experiment 2. As the clustering results had no effect on the estimation of the number of speakers, the results of the clustering are not presented here but can be seen in the results of the second experiment. Only the validity results will be presented for this experiment.

Fig. 6 show the results of the validity functions for two and three speakers. All the validity functions for a two-participant conversation present a minimum value for the right number of speakers, i.e., two speakers [Fig. 6(a)]. For three participants, four out of five validity functions received the minimal value for three-speakers, and in one case [Fig. 6(b)] the minimum value was reached for four speakers. The value for three speakers was very close to the four-speaker value (0.6051 and 0.5937 for three and four speakers, respectively).

It is important to mention that at the end of each training process of the system, the validity value was calculated and one model was removed. Each model was a SOM and consisted of 60 neurons (60 CWs), i.e., the new clustering, for $r - 1$ speakers, was done with a smaller number of CWs so the overall distortion increased. Despite the increase in the distortion, the validity value can still decrease. Fig. 7(a) shows the distortions as a function of the number of iterations in one of the conversations between three participants. The vertical dashed lines indicate the places where a cluster was removed, and the number in each zone is the number of models that were trained. It is clear that the distortion increased after each model reduction. Fig. 7(b) shows that the validity minimum was achieved at three clusters although the distortions of four, five, and six clusters were lower than in the three-model case, i.e., the inter-
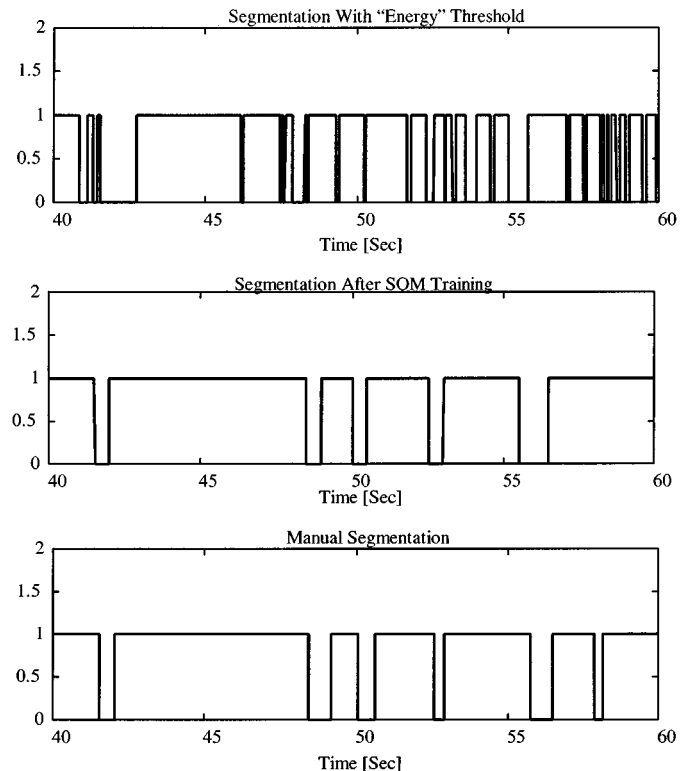

Fig. 5. Two-speaker high-quality conversation segmentation (speech—1/nonspeech—0. (a) "Energy" threshold segmentation. (b) After SOM training segmentation. (c) Manual segmentation.

cluster distance in the three-model case was much larger than in the four, five and six model cases.

Experiment no. 4: The goal of this experiment was to cluster telephone-quality conversations (R was assumed to be known). For this experiment, the clustering algorithm was applied to telephone-quality conversations between two participants. The duration of the conversations was two minutes.

The clustering results showed that eight (out of a total of 12) conversations converged approximately to a 6% weighted error. An example of the clustering results of a telephone-quality conversation is presented in Table V. Four conversations (out of the 12) did not converge. These four conversations were examined by a human listener and were found to be of low quality. In fact one of the
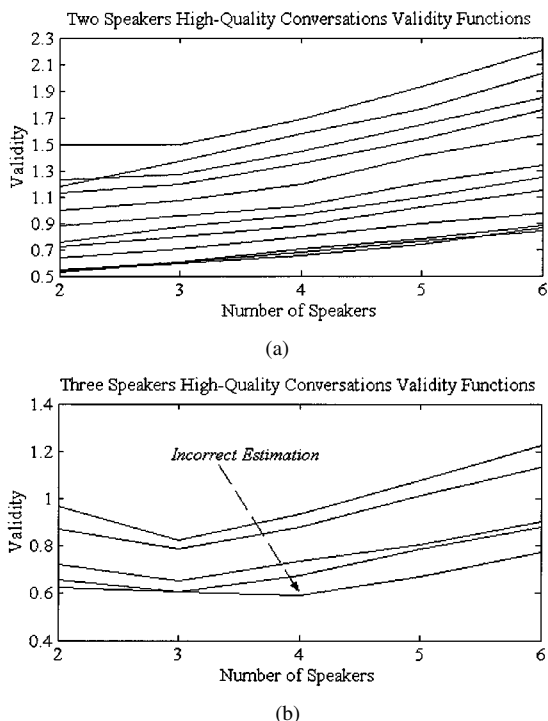
Fig. 6. The validity functions. (a) Twelve high-quality conversations between two speakers. (b) Five high-quality conversations between three speakers.
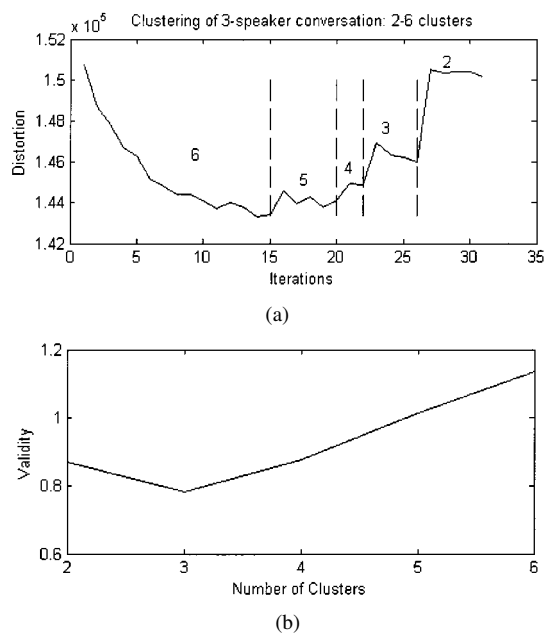

Fig. 7. (a) Distortion as a function of the iteration number and cluster number in the conversation between three participants. (b) Validity as a function of the number of clusters.
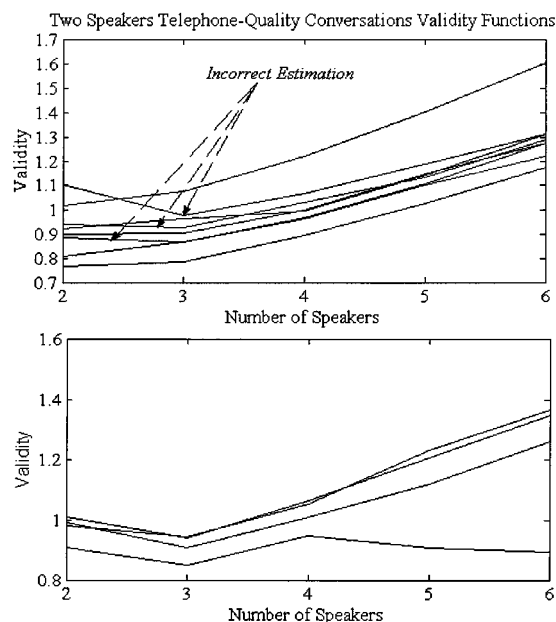

Fig. 8. Validity graphs. (a) Eight telephone-quality conversations that converged well. (b) Four telephone-quality conversations that did not converge well.

TABLE V
TWO-MALE CONVERSATION CONFUSION MATRIX OF HIGH-QUALITY
AND TELEPHONE-QUALITY CONVERSATIONS

	High-quality conversation error			Telephone-quality conversation error		
	A	B	NS	A	B	NS
A	93.5	0	1.6	93.6	2.4	1.8
B	4.3	94.9	3.4	0.9	94.0	4.6
NS	2.2	5.1	95.0	5.5	3.5	93.6
Total error [%]	5.6			6.2		

The additional model was either a second nonspeech model, a simultaneous speech model, second model for one of the speakers or a model that contained data of both speakers.

Experiment no. 5: The goal of this experiment was to test the validity criterion on telephone-quality data. The clustering algorithm was applied to telephone quality conversations together with the validity criterion. Similar to high quality clustering the validity process did not affect the clustering quality.

The results of this experiment are presented in Fig. 8. Fig. 8(a) shows that the validity function of the eight conversations converged well when the number of speakers was known. In five cases the number of speakers were correctly estimated as two; in the other three cases the minimum value of the validity function was situated at the three-speaker place (marked with arrows). In two of the wrong cases the validity coefficients for two and three clusters were very close. For the other four conversations [Fig. 8(b)] the validity function minimum value was for three speakers. This experiment confirmed the results obtained in the previous experiment.

conversations was judged by the listener as a conversation between three, rather than two speakers.

An algorithm for three speakers (plus nonspeech) for these conversations was applied. The results yielded one nonspeech model, one model mostly (at least 70% of the data that clustered to the model) that belonged to one speaker, and one model that belonged to another speaker.

IV. CONCLUSION

In this work a time-series clustering approach, based on an iterative process with competition between SOM models was investigated.

Unsupervised speaker recognition task seems to be very difficult task. It is probably due to the fact that, to begin with, the amount of information about the speaker in speech signal is relatively low, as compared to the information on the message. Without *a priori* labeled information it is difficult to model the speakers. The fact that in the general segmentation problem the number of speakers is unknown makes the problem extremely difficult. To achieve good clustering results we first had to determine the optimal size of the models to represent a speaker and the shortest segment length to derive sufficient statistic of the speaker. Shorter segments enable better segmentation resolution. The experiments showed that SOM of size 6×10 is sufficient for speaker modeling. For short conversations (about 60 s for two-speaker conversations) segments of half-second were needed.

After the segments length and models size were defined, the algorithm was applied for high-quality conversations between two to five participants and for two-speakers telephone-quality conversations. For two- and three-speakers high-quality data conversations and for two-speakers telephone-quality data, the results were usually good (more than 80% success). For four and five speakers, only one conversation of four speakers converged correctly. This was the only conversation were the data was approximately homogeneously divided among the participants, the average segment length was more than 3 s, there was almost no simultaneous data, and there was small amount of nonspeech data. This shows that the algorithm is sensitive to the amount of data of every cluster, especially when the data is overlapping and, as well as to the amount of noise (nonspeech and simultaneous speech data). Therefore, it might be necessary to develop an effective speech/nonspeech and simultaneous speech detector.

A validity criterion was suggested. The validity estimation never affected the quality of the clustering in two- and three-speaker high-quality conversation the estimation of the number of clusters was correct except for one three-speaker conversation. In telephone-quality eight out of 12 conversations were correctly clustered. In this conversations five out of eight conversations the number of clusters was correctly estimated. In three conversations the number of speakers was estimated as three instead of two speakers.

In order to compare the performance of the proposed approach to existing algorithms, a systematic review of the available literature was made. Four articles describing several algorithms appeared to be relevant for the comparison [12]–[15]. The algorithms cover different variations of HMM. Due to the different databases and the lack of information about the performed evaluations, only a very crude comparison could be made in the present study. It was found that in all these works all the conversations were at least 90 s. The results were similar to the reported here but in all the cases the algorithms were very sensitive to the initial conditions. The research of Cohen and Lapidus [14] and [15] was the only one done on the same telephone-quality database and the results were similar.

To conclude, it can be said that half-second segments can be sufficient duration for unsupervised speaker recognition, and SOM of size 6×10 can accurately model each speaker. If the data is not highly nonuniformly distributed between the speakers a correct clustering can be performed, and the number of speakers can be well estimated, in high- and telephone-quality conversations.

REFERENCES

[1] S. Furui, "An overview of speaker recognition technology," in *Proc. ESCA Workshop Automatic Speaker Recognition, Identification, Verification*, Apr. 1994, pp. 1–9.

[2] J. P. Campbell Jr., "Speaker recognition: A tutorial," *Proc. IEEE*, vol. 85, no. 9, pp. 1437–1462, Sept. 1997.

[3] S. Furui, "Recent advances in speaker recognition," *Pattern Recognition Lett.*, vol. 18, no. 9, pp. 859–872, Sept. 1997.

[4] H. Gish, M.-H. Siu, and R. Rohlicek, "Segregation of speaker for speech recognition and speaker identification," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, 1991, pp. 873–876.

[5] M.-H. Siu, G. Yu, and H. Gish, "An unsupervised, sequential learning algorithm for the segmentation of speech waveform with multiple speakers," in *Proc. Int. Conf. Acoust. Speech, Signal Processing*, vol. 2, 1992, pp. 189–192.

[6] T. Kohonen, *Self-Organization and Associative Memory*. Berlin, Germany: Springer-Verlag, 1989.

[7] ——, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sept. 1990.

[8] M. H. Kuhn, "Speaker recognition accounting for different voice conditions by unsupervised classification (cluster analysis)," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, pp. 54–57, Jan. 1980.

[9] A. Cohen and V. Lapidus, "Unsupervised text independent speaker classification," in *Proc. 18th Convention Elect. Electron. Eng. Israel*, 1995, pp. 3.2.2 1–5.

[10] M. Sugiyama, J. Murakami, and H. Watanabe, "Speech segmentation and clustering based on speaker features," in *Proc. Int. Conf. Acoust. Speech, Signal Processing*, vol. 2, 1993, pp. 395–398.

[11] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish, "Clustering speakers by their voices," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, 1998, pp. 557–560.

[12] J. O. Olsen, "Separation of speakers in audio data," in *Proc. 4th Europ. Conf. Speech Commun. Technol.*, vol. 1, 1995, pp. 355–358.

[13] L. Wilcox, F. Chen, D. Kimber, and V. Balasubramanian, "Segmentation of speech using speaker identification," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, 1994, pp. 161–164.

[14] A. Cohen and V. Lapidus, "Unsupervised speaker segmentation in telephone conversations," in *Proc. 19th Convention Elect. Electron. Eng. Israel*, 1996, pp. 102–105.

[15] ——, "Unsupervised, text independent, speaker classification," in *Proc. Int. Conf. Signal Processing Applicat. Technol.*, 1996, pp. 1745–1749.

[16] T. Kosaka, S. Matsunaga, and S. Sagayama, "Speaker-independent speech recognition based on tree-structured speaker clustering," *Comput. Speech Language*, vol. 10, no. 1, pp. 55–74, Jan. 1996.

[17] I. Voitovetsky, H. Guterman, and A. Cohen, "Unsupervised speaker classification using self-organizing maps (SOM)," in *Proc. 1997 IEEE Workshop Neural Networks Signal Processing VII*, Amalia Island, FL, Sept. 1997, pp. 578–587.

[18] ——, "Validity criterion for unsupervised speaker recognition," in *Proc. 1st Workshop Text, Speech, Dialogue*, Brno, Czech Republic, Sept. 1998, pp. 321–326.

[19] L. M. D. Owsley, L. E. Atlas, and G. D. Bernard, "Self-organizing feature maps and hidden Markov models for machine-toll monitoring," *IEEE Trans. Signal Processing*, vol. 45, pp. 2787–2798, Nov. 1997.

[20] A. Kehagias and V. Petridis, "Time-series segmentation using predictive modular neural networks," *Neural Networks*, vol. 9, no. 8, pp. 1691–1709, Nov. 1997.

[21] K. Pawelzik, J. Kohlmorgen, and K.-R. Muller, "Annealed competition of expert for segmentation and classification of switching dynamics," *Neural Comput.*, vol. 8, no. 2, pp. 340–356, Feb. 1996.

[22] J. Kohlmorgen, K.-R. Muller, and K. Pawelzik, "Segmentation and identification of drifting dynamical systems," in *Proc. IEEE Workshop Neural Networks Signal Processing VII*, Amalia Island, FL, 1997, pp. 326–335.

[23] O. A. S. Carpinteiro, "A hierarchical self-organizing map model for sequence recognition," *Pattern Anal. Applicat.*, vol. 3, no. 3, pp. 287–289, 2000.

[24] A. Krogh, M. Brown, I. S. Mian, K. Sjolander, and D. Haussler, "Hidden Markov models in computational biology applications to protein modeling," *J. Mol. Biol.*, vol. 235, no. 5, pp. 1501–1531, Feb. 1994.

[25] P. Smyth, "Clustering sequences with hidden Markov models," in *Proc. Advance Neural Inform.*, vol. 9, 1997, pp. 648–654.

[26] T. K. Moon, "Temporal pattern recognition using fuzzy clustering," in *Proc. 3rd IEEE Int. Conf. Fuzzy Syst.*, vol. 1, 1994, pp. 432–435.

[27] S. Abe and M.-S. Lan, "Fuzzy rules extraction directly from numerical data for function approximation," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 119–129, Jan. 1995.

[28] I. Lapidot (Voitovetsky) and H. Guterman, "VQ-based clustering algorithm of piecewise-dependent-data," in *Advances in Self-Organizing Maps*, N. Allison, H. Yin, L. Allison, and J. Slack, Eds: Springer, 2001, pp. 95–101.

[29] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[30] W. Pedrycz, "Fuzzy sets in pattern recognition: Methodology and methods," *Pattern Recognition*, vol. 23, no. 1/2, pp. 121–146, 1990.

Itshak Lapidot (Voitovetsky) was born in 1971 in Russia. He received the B.Sc. , M.Sc., and Ph.D. degrees in electrical and computer engineering from the Ben-Gurion University, Beer-Sheva, Israel, in 1991, 1994, and 2001, respectively.

He is currently a Lecturer at the Software Engineering Department, Negev Academic College of Engineering, Beer-Sheva, Israel. His primary research interests are speaker recognition, pattern recognition, time series clustering, and neural networks.

Hugo Guterman received the Bachelor's degree in electronic engineering from the National Technological University, Buenos Aires, Argentina in 1978 and the M.Sc. and Ph.D. degrees in computer and electrical engineering from the Ben-Gurion University, Beer-Sheva, Israel, in 1982 and 1988, respectively.

From 1988 to 1990, he was a Postdoctoral Fellow at the Massachusetts Institute of Technology, Cambridge. Since 1990, he has been at the Department of Computer and Electrical Engineering at the Ben-Gurion University. His research interests include control, image and signal processing, neural networks, and fuzzy logic.

Arnon Cohen was born in Haifa, Israel, in 1938. He received the B.Sc. and M.Sc. degrees from the Technion—Israel Institute of Technology, Haifa, in 1964 and 1966, respectively, and the Ph.D. degree from Carnegie-Mellon University, Pittsburgh, PA, in 1970.

Since 1972, he has been with the Department of Electrical and Computer Engineering and the Biomedical Engineering Program, Ben Gurion University, Beer-Sheva, Israel, where he is a Professor of Electrical and biomedical Engineering. His research interests are in signal processing, mainly with biomedical and speech applications. He is the author of the book *Biomedical Signal Processing* (Boca Raton, FL: CRC, 1986).