2001: A Speaker Odyssey, The Speaker Recognition Workshop, pp. 169-173, June 18-22, 2001, Crete, Greece

Resolution Limitation in Speakers Clustering and Segmentation Problems

Itshak Lapidot (Voitovetsky)[1] and Hugo Guterman[2]

[1]Department of Software Engineering
Negev Academic College of Engineering
P.O.B. 45 Beer-Sheva, 84100, Israel
itsik@nace.ac.il

[2]Department of Electrical and Computer Engineering
Ben-Gurion University of the Negev
P.O.B. 653, Beer-Sheva, 84105, Israel
hugo@ee.bgu.ac.il

Abstract

In unlabeled and unsegmented conversation, i.e. no *a-priori* knowledge about speakers' identity and segments boundaries is provided, it is very important to cluster the conversation (make a segmentation and labeling) with the best possible resolution. For low-resolution cases, i.e. the duration of the segment is long; the segments might contain data from several speakers. On the other hand, when short segments are used (high resolution) not enough statistics is provided to allow correct decision about the identity of the speakers. In this work the performance of a system, which employs different segment lengths, is presented. We assumed that the number of speakers, *R*, is known, and high-quality conversations were used. Each speaker was modeled by a Self-Organizing-Map (SOM). 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). We found that the optimal segment duration was half-second. The system has a clustering performance of about 90% for tow-speaker conversation and over 80% for three-speaker conversations.

1. Introduction

When the speech data is segmented and labeled according to the speakers' identity, the problem is supervised. In this case each speaker model can be trained according to its own data. If the data is only segmented but not labeled the problem can be either supervised or unsupervised. In many applications even the segments' boundaries are not given. In this case, in addition to the labeling, segmentation of the data has to be performed and the problem is unsupervised. In our work we divided the data into short segments and try to cluster them. The assumptions were that each segment belongs only to one speaker and that each segment is long enough to determine the identity of the speakers. Therefore it is crucial to find the shortest segment that contains enough statistics to determine the identity of the speakers.

Different approaches have been applied for speakers' segmentation and labeling. In many cases the segmentation is well defined [1] and [2], in other cases segments of one second or higher have been employed [3]. One-second segment may be too long for a short utterance. We found that half-second segments length was sufficient for good clustering each model was produced by a SOM [4] of size 6×10. The proposed system, cluster the data into *R* speakers (*R* assumed to be known) by performing competition between SOMs. At the end of the competition each SOM represent different speaker and the segmentation and labeling are performed.

In section 2 the unsupervised system is described. A brief introduction of the SOM's algorithm is provided in section 3. The experiments and results are present in sections 4 and 5, while conclusion can be found in section 6.

2. The system

The general system is shown in Fig. 1. Detailed description of the algorithm can be found in previous works [5] and [6]. A preliminary segmentation of speech and non-speech was produced. The preliminary detection of the speech/non-speech segments was attained by calculating the sum of absolute values over frames of 50msec duration. Then by applying a pre-defined threshold a speech/non-speech rude classification is produced.

Initially, all the segments that were under the threshold were used to train the non-speech SOM. Speech segments were randomly and equally divided between the R models. Each models, including the non-speech model, was a Kohonen 6×10 SOM [4]. At the end of each iteration, after the training process, the data was regrouped between the models. If two consecutive partitions of the data differed, the retraining process of the SOMs was applied using the new partition. The clustering process stops when a termination condition achieved. The training was stopped when in two consecutive iterations the partitions were identical. A less strict termination criterion such as the difference in partitions smaller than pre-defined percent of segments can be used.

The algorithm clusters the data such that the total Euclidean error, during regrouping, is minimized. Proof of the algorithm convergence can be found in [7]. Hence the i-th iteration of the process is defined as follows:

1. Retrain the models with the new partition achieved by the previous iteration.

2001: A Speaker Odyssey, The Speaker Recognition Workshop, pp. 169-173, June 18-22, 2001, Crete, Greece

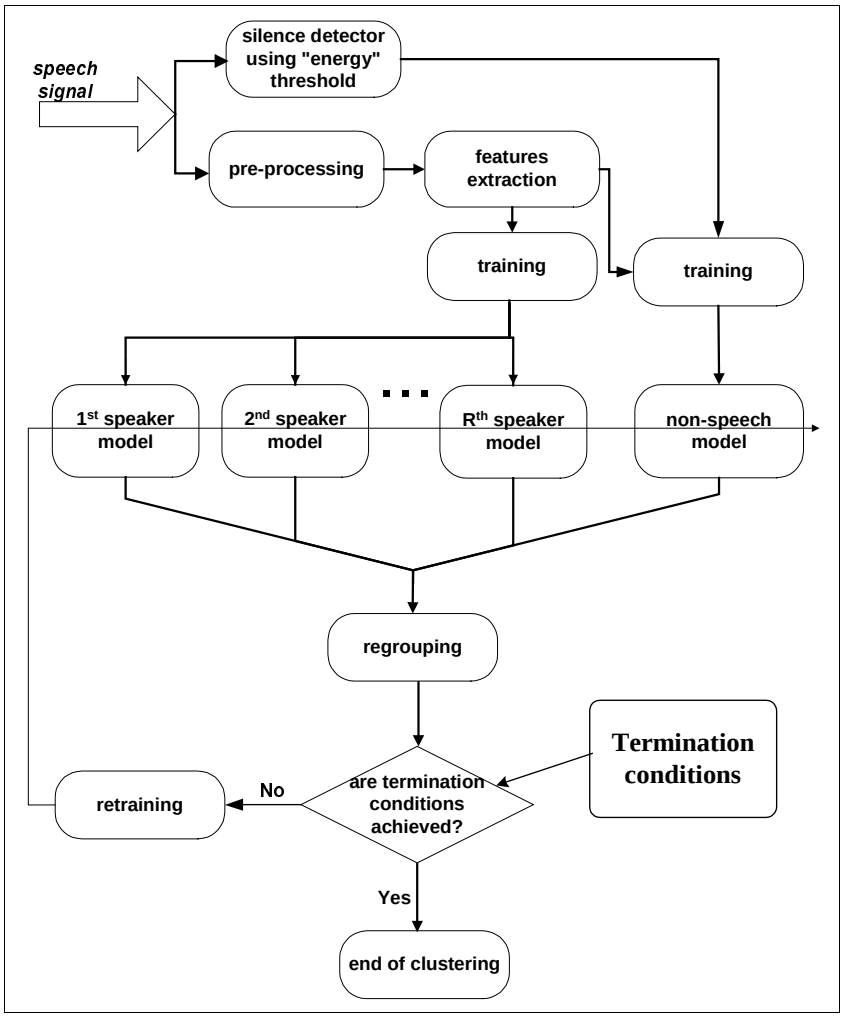

Figure 1: General description of the unsupervised speaker classification system.

2. Regroup the data.
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+1* models, for the *R* speakers and one for the non-speech data. The data is segmented and labeled as required.

The termination criterion used here was based on the regrouping step. Termination was attained when two consecutive iterations showed no change in the partition or when the number of changes was less than a given percentage.

3. Kohonen SOM

In this work, each speaker and non-speech models were SOMs [4]. The SOMs' structure can be seen in Fig. 2. In the training algorithm of the SOM, it is necessary to adapt not only the winner neuron to a given input at iteration t is $\mathbf{m}_\mathbf{c}^t$, 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 is presented below.

Let $\mathbf{V} = \{\mathbf{v}_n\}_{n=1}^N$, $\mathbf{v}_n = \left[\mathbf{v}_n^1 \quad \mathbf{v}_n^2 \quad \cdots \quad \mathbf{v}_n^p\right]^T$ be the training data set, and let $\{\mathbf{m}_\mathbf{k}^t\}_{k=1,\ldots,K}$, $\mathbf{m}_\mathbf{k}^t = \left[m_k^{1,t} \quad m_k^{2,t} \quad \cdots \quad m_k^{p,t}\right]^T$ be the SOM's neurons. The unsupervised training algorithm is:

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

$$\begin{cases} \mathbf{m}_\mathbf{k}^{t+1} = \mathbf{m}_\mathbf{k}^t + h_{ck}(t)\left[\mathbf{v}_n - \mathbf{m}_\mathbf{k}^t\right] & \text{if } k \in N_c(t) \\ \mathbf{m}_\mathbf{k}^{t+1} = \mathbf{m}_\mathbf{k}^t & \text{if } k \notin N_c(t) \end{cases} \quad (1)$$

$h_{ck}(t)$ is an updating function, at iteration t.

5. If the termination criterion is met, exit; if not return to step 2.

$h_{ck}(t)$ and $N_c(t)$ are monotonically non-increasing functions. There are two standard ways to define $h_{ck}(t)$:

$$h_{ck}(t) = \begin{cases} h_0(t)\exp\left(-\dfrac{\|r_k - r_c\|^2}{\sigma(t)^2}\right) & \text{(a)} \\ \text{or} & \\ \alpha(t) & \text{(b)} \end{cases} \quad (2)$$

where $\alpha(t)$, $h_0(t)$ and $\sigma(t)$ are decreasing functions of time. r_k and r_c are the locations of $\mathbf{m_k}$ and $\mathbf{m_c}$ in the network respectively (we use a rectangle structure of SOM and the location refers to the row and the column of the neuron in the two-dimensional array). 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 this stage we apply (2a). In the second phase (the tuning phase), $h_{ck}(t)$ and $N_c(t)$ are small and decrease slowly to zero and the adaptation uses (2b). The number of iterations employed was ten times the number of training input vectors ($10\times N$).

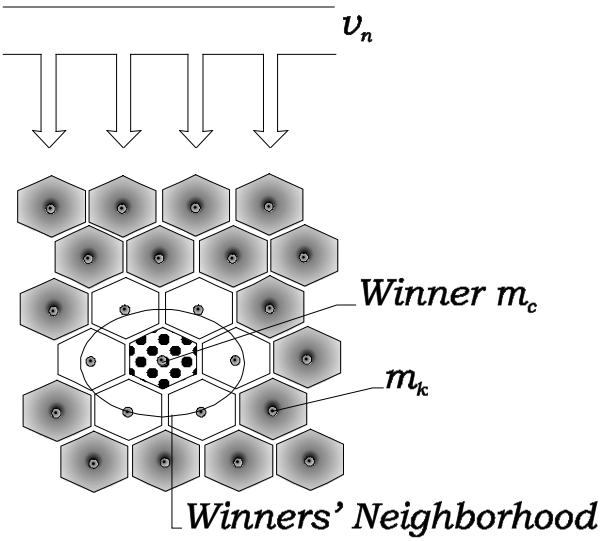

Figure 2: Kohonen SOM.

4. The database

The speech database employed in this research is composed of Hebrew conversations. The data was recorded in an acoustic room. The database consist of twelve two-speaker conversations and five three-speaker conversations. Nine males and one female took part in these conversations. The sampling parameters were 16KHz sampling rate, 12Bits/Sample, 50Hz-7.8Khz anti-aliasing filter. The SNR was approximately 35dB.

5. Experiments and results

For all the experiments, 12 LPC based cepstrum and 12 delta-cepstrum features were used. The features were extracted from 15msec frame with 5msec frame rate (10msec overlapping between consecutive frames). In the first experiment the non-speech segments were taken out before the training. In other experiments preliminary speech/non-speech segmentation was performed (see section 2).

5.1. First experiment

The goal of this experiment was to test the ability of the SOM to cluster two speakers, and to determine the optimal 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 198sec and the speech duration between 72 to 180sec (the rest of the data was non-speech). The worst ratio between the speakers' speech duration was 33/59 for a conversation of 92sec duration. The shortest speech duration of a speaker for one conversation was 32sec.

The system performances were tested with segments that were 0.25sec, 0.5sec, 0.75sec and 1sec in length.

Six out of the twelve high-quality conversations were used for this experiment. The results for all the conversations were quite similar and were not affected by the length of the segment. Clustering errors are summarized in Tables 1 and 2. Due to the finite resolution some of the segments splits between the speakers. Table 1 summarized the results were all the segments that splits (a segment containing data from two speakers) were taken out for clustering results evaluation. In table 2 the results evaluated including all the segments.

It can be observed that: 1) the error rate at the splitting segment is higher than for the non-splitting segments, 2) there were no difference in the error rate as a function of the segments' duration, 3) as the segments became shorter, the appearance of the splitting segment became more rare, and therefore the influence of the splitting segments decreased.

Table 1: Example of segmentation results as a function of the segment length (without splitting points).

Error (in %)Without Splitting Segments – Mean (STD)			
0.25sec	0.5sec	0.75sec	1sec
2.9 (1.5)	2.8 (2.6)	4.2 (2.2)	1.7 (1.7)

Table 2: Example of segmentation results as a function of the segment length (including splitting points).

Error (in %) With Splitting Segments – Mean (STD)			
0.25sec	0.5sec	0.75sec	1sec
4.8 (1.8)	5.9 (2.8)	7.4 (2.3)	7.2 (1.2)

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

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

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

5.2. Second experiment

The goal of this experiment was to perform a clustering of high-quality conversations, using the segment length determined in the first experiment (*R* was assumed to be known). All the high-quality conversations (non-speech segments were included) were used for training and error evaluation. An additional SOM was used to represent the non-speech model.

The preliminary automatic segmentation of the speech/non-speech algorithm was employed. An empirical threshold was found but it turned out to be not very accurate. A comparison between 60sec of preliminary speech/non-speech segmentation, final segmentation, and manual segmentation is shown in Fig. 3. 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 non-speech. 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.

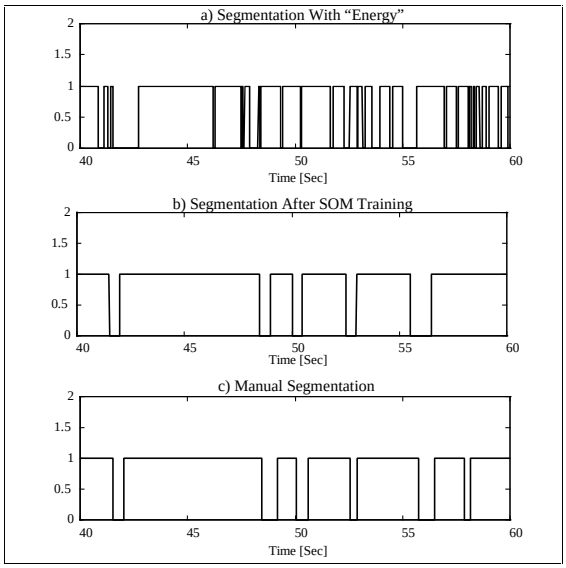

Figure 3: Two-speaker high-quality conversation segmentation (speech – 1 / non-speech – 0). a) "Energy" threshold segmentation; b) After SOM training segmentation, c) Manual segmentation.

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 3. Three of the conversations were between a male and a female, for which the error rate was approximately 4.3%.

Table 3: Two-male conversation confusion matrix of high-quality conversations (NS – non-speech).

	High-quality conversation error		
	A	B	NS
A	93.5	0	1.6
B	4.3	94.9	3.4
NS	2.2	5.1	95.0
Total error [%]	5.6		

The clustering algorithm was applied for three-speaker conversations. Conversations between three speakers always converged and the results were not worse than 15%. Fig. 4 shows the convergence of the Euclidean error as a function of the iteration number. It can be seen that the clustering process can be terminated after 10 iteration.

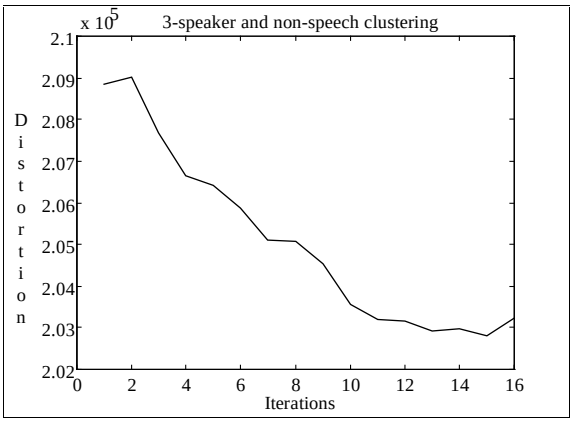

Figure 4: Clustering distortion as a function of iteration number for a three-speaker conversation.

6. Conclusions

From the experiments it can be conclude that the error probability at the splitting segment is significantly higher than for the non-switching segments. Consequently, it is useful to reduce the probability of appearance of such segments. As the segments become shorter the appearance of splitting segments becomes rare, and therefore the influence of the switching segments decreased. Half-second segments can be used for clustering of relatively short conversations (50 second in duration). Despite the fact that the preliminary speech/non-speech detector performance is rather poor, the final speech/non-speech segmentation achieved is close to manual segmentation. Finally, it can be seen that the iterative process is finite and the overall distortion decreases as a function of number of iterations.

7. References

[1] M. H. Kuhn, "Speaker recognition accounting for different voice conditions by unsupervised classification (cluster analysis)," *IEEE Trans. Syst., Man, Cybern.*, SMC-10(1): 54-57, 1980.

[2] M. Sugiyama, J. Murakami, and H. Watanabe, "Speech segmentation and clustering based on speaker features," *Proc. International Conference on Acoustic Speech and Signal Processing*, 2: 395-398, 1993.

[3] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish, "Clustering speakers by their voices," *Proc. International Conference on Acoustic Speech and Signal Processing*, 2: 575-560, 1998.

[4] T. K. Kohonen, "The self-organizing map," *Proc. IEEE*, 78(9): 1464-1480, September, 1990

[5] I. Voitovetsky, H. Guterman, and A. Cohen, "Unsupervised speaker classification using self-organizing maps (SOM)," *Proc. Neural Networks for Signal Processing VII IEEE Workshop*, 578-587, 1997.

[6] I. Voitovetsky, H. Guterman, and A. Cohen, "Validity criterion for unsupervised speaker recognition," *Proc. of the First Workshop on Text Speech and Dialogue*, 321-326, 1998.

[7] I. Lapidot (Voitovetsky) and H. Guterman, "VQ-Based Clustering Algorithm of Piecewise-Dependent-Data," *Workshop on Self-Organizing Maps*, 2001.