

Speaker Modeling in Conversational Speech with Application to Speaker-Count

Ananth N. Iyer[†], Uchechukwu O. Ofoegbu[†], Robert E. Yantorno[†], Stanley Wenndt[‡]

[†]Speech Processing Laboratory, Temple University
Philadelphia, PA 19122 USA, http://www.temple.edu/speech_lab
{aniyer, uchel, byantorn}@temple.edu

[‡]Airforce Research Laboratory/IFEC
525 Brooks Rd, Rome, NY 13441-4505, USA
Stanley.Wenndt@rl.af.mil

Abstract

Determining the number of speakers participating in a conversation is an important area of research as it has applications in telephone monitoring systems, meeting transcriptions and speaker segmentation algorithms. The speaker count problem poses a greater challenge when no information about the speakers is available. To circumvent this problem, the use of a generic model set is proposed. The general approach taken is to determine the subset of the generic models that best represents any speech data. The speaker count is derived from the number of models in the subset, and the speakers participating in the conversation are represented by the models. Both the models and the number of models in the subset are obtained using an evolutionary iterative procedure. This method involves performing a sequence of four operations: 1.) speaker identification, 2.) pruning the model set, 3.) constrained adaptation and 4.) updating the model set. These operations are performed repeatedly on the model set until the convergence criterion is met. The proposed technique has produced promising results on preliminary speaker count experiments.

1. Introduction

The problem of learning speaker models in a conversation is addressed in this paper. The problem can be subdivided into three parts: i) determining the number of speakers participating in the conversation, ii) learning the speaker models and iii) indexing the temporal activity of each speaker. The main application presented in this paper is on the first part, and the other two parts is expected to be produced as by-products. The need for a method to determine the speaker count in a conversation arises mainly in telephone monitoring systems. The developed method is motivated toward monitoring telephone lines of a prison, where placing three-way telephone calls are forbidden.

In this research, the speaker building process is treated as an evolutionary iterative procedure which involves performing a sequence of operations until some convergence criterion is met. An initial *generic model set*, which can represent any speaker, is mapped into a speaker specific model set representing the speakers in the conversation.

One of the first concerns in the use of generic speaker models choice of a speaker model. The Gaussian Mixture Model (GMM) has been a favourite choice for modeling as it delivers excellent

results in speaker identification, when large amounts of data is available for testing [1]. The concept of generic speaker models using GMMs has been successfully applied in speaker indexing on news broadcast data [2]. However, the GMM seldom performs well when the data is limited. A method to choose the optimal model (GMM or VQ), using the Bayesian Information Criterion [3], however the technique is extremely complex. To obtain a balance between performance and complexity, the VQ model is chosen as the speaker representation. The VQ model, due to its non-parametric nature, outperforms the GMM when the amount of data available is less.

Studies have shown that techniques based on generic models are dependent on their initialization. Various methods based on the Universal background model [4], Monte Carlo Markov Chain, and Speaker Model Quantization methods [5] have been proposed. An initial model set is desired to span the entire model space and the models chosen to be equidistant. A new method, which ensures, the models in the initialization to be "maximally separated" is proposed in this paper and is presented in the next section. The sequence of operations performed on the model set is also presented in the next section. Experimental setup and results are discussed in section 3. Finally conclusions are drawn in section 4.

2. Methods

In this section, the process of building the speaker models is elaborated. A speaker model is represented by a matrix \mathbf{C} with each row containing a code-vector \mathbf{c}_i ; $i = 1, 2, \dots, P$, obtained using the splitting method to performing vector quantization. A set of speaker models is created with speech data from a standard database and is represented as $\Omega = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_Q\}$, with Q being the total number of speakers in the database. A sequence of operations is performed on the models, and these operations are presented in the general process diagram show in figure 1. The first step is to initialize the procedure with a generic model set \mathbf{S} and is discussed next.

2.1. Initialization

It was inferred, based on experimentation, that initialization of the generic model set plays an important role for the successful termination of the speaker building procedure. A desired feature in

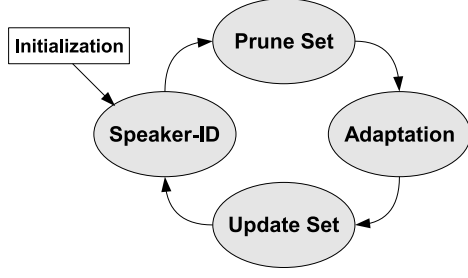


Figure 1: The general diagram showing the sequence of operations performed iteratively on the generic model set.

the initial model set is that they are *maximally separated*, i.e., they span a large area in the model space and have small overlap regions. To obtain such a set, a distance between models and an algorithm to sub-sample the set Ω are required. The need for a distance is addressed by introducing two distances. The first distance is defined as:

$$d_p(\mathbf{C}_p, \mathbf{C}_q) = \|\mathbf{C}_p - \mathbf{P}\mathbf{C}_q\|_F, \quad (1)$$

where \mathbf{P} is a permutation which permutes the rows of \mathbf{C}_p such that it aligns with \mathbf{C}_q . The permutation matrix has only one non-zero element in each row. The distance permutes the code-vectors of the speaker models and hence is aptly termed as code-vector permutation distance (CVPD). To determine the matrix \mathbf{P} , say models \mathbf{C}_x and \mathbf{C}_y represent that same speaker with the order of their code-vectors permuted. Such a situation can occur when the models are generated at two different instantiations of the vector quantization procedure. Therefore, by definition,

$$\mathbf{C}_x = \mathbf{P}\mathbf{C}_y \quad \text{and} \quad \mathbf{C}_x\mathbf{C}_x^T = \mathbf{P}(\mathbf{C}_y\mathbf{C}_y^T)\mathbf{P}^T. \quad (2)$$

Consider the eigen value decomposition of similar matrices, $\mathbf{C}_x\mathbf{C}_x^T = \mathbf{Q}_x\mathbf{\Lambda}\mathbf{Q}_x^{-1}$ and $\mathbf{C}_y\mathbf{C}_y^T = \mathbf{Q}_y\mathbf{\Lambda}\mathbf{Q}_y^{-1}$. Using 2 and the fact that the eigen values of similar matrices are identical,

$$\mathbf{Q}_y\mathbf{\Lambda}\mathbf{Q}_y^{-1} = (\mathbf{P}\mathbf{Q}_y)\mathbf{\Lambda}(\mathbf{P}\mathbf{Q}_y)^{-1}, \quad (3)$$

and hence $\mathbf{P} = \mathbf{Q}_y\mathbf{Q}_x^{-1}$. The distance between two models is written as:

$$d_p(\mathbf{C}_p, \mathbf{C}_q) = \|\mathbf{C}_p - \mathbf{Q}_q\mathbf{Q}_p^{-1}\mathbf{C}_q\|_F. \quad (4)$$

The second distance the minimum of the distances between each code-vector of one model to the nearest code-vectors in the other model. This can be represented as:

$$d_m(\mathbf{C}_p, \mathbf{C}_q) = \min_{i,j;i \neq j} \|\mathbf{c}_{pi} - \mathbf{c}_{qj}\|_2, \quad (5)$$

where \mathbf{c}_{pi} represents the i^{th} code-vector of the p^{th} model. Note that both the distances are symmetric and abide the basic rules of a distance. The distance measure is used to construct a metric space with the models as points in the space, and will be referred to as the model space. To determine a maximally separated subset of models from a larger set of models, one can think of applying any clustering method to group similar models and represent them using a centroid model [5]. This approach is not applicable in the speaker model space because the addition operation between models is not well defined.

The problem of obtaining the maximally separated subset \mathbf{S} can be setup as an optimization problem, maximizing:

$$J = \sum_{\mathbf{C}_i, \mathbf{C}_j \in \mathbf{S}} d(\mathbf{C}_i, \mathbf{C}_j), \quad (6)$$

where the total summation is performed on all the models in the subset $\mathbf{S} \subset \Omega$. The solution to this optimization problem can be obtained by a brute-force combinatorial search for the optimal subset, and in general, such a searching procedure is computationally expensive. For example, consider a set Ω with $Q = 50$ and the number of models in the subset $P = 10$ is required. It can be determined that cost function J has to be evaluated ${}^Q C_P \simeq 10^{10}$ times. To reduce the computational magnitude, a suboptimal strategy is proposed and is described below.

Algorithm 1 Determine the maximally separated subset

Require: Number of models in subset K ; Initialize set $\mathbf{S}^{(2)} = \{\mathbf{C}_p, \mathbf{C}_q\}$, such that \mathbf{C}_p and \mathbf{C}_q are the two farthest models; $k = 2$;

- 1: **while** $k < K$ **do**
- 2: Choose model \mathbf{C}_i which has the largest distance to set $\mathbf{S}^{(2)}$. The distance between a point and a set is defined as the minimum of all the distances from the model to all the models in the set. This step can be mathematically written as

$$\mathbf{C}_i = \operatorname{argmax}_{\mathbf{C}_i \in \Omega} \min_{\mathbf{C}_j \in \mathbf{S}^{(k)}} d(\mathbf{C}_i, \mathbf{C}_j) \quad (7)$$

- 3: $\mathbf{S}^{(k+1)} = \mathbf{S}^{(k)} \cup \mathbf{C}_i$
 - 4: $k = k + 1$.
 - 5: **end while**
 - 6: **return** $\mathbf{S} = \mathbf{S}^{(k+1)}$
-

The validity of the algorithm can be easily determined by evaluating it on a random set of points sampled from a uniform distribution and the result is shown in figure 2. The gray circles represents the points in the set Ω and the black asterisks represents the points in the subset \mathbf{S} .

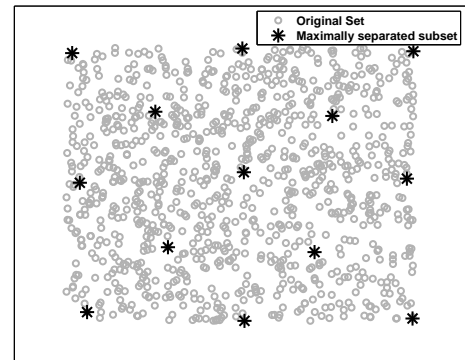


Figure 2: Illustration of the algorithm to determine the maximally separated subset (asterisks) on a random set sampled from uniform distribution (circles).

2.2. Discriminative Speaker Identification

The initial generic model set S determined in the previous step is used as the database to perform speaker identification on the test data being analyzed. Speaker identity is obtained at an utterance level and the utterance markings are determined based on statistics obtained from the conversational speech database SWITCHBOARD. The database consists of conversations recorded over a telephone line. The length of utterances, i.e., the length of speaker homogeneous segments are determined and a histogram of the results is shown in figure 3. A total of 2435 conversations were used to generate the statistics with each conversation lasting approximately 15 minutes.

An inference can be made based on the histogram that most of the utterances have a duration of at least 1 second and hence an utterance is defined as a group of consecutive voiced segments which amounts to 1 second of speech data.

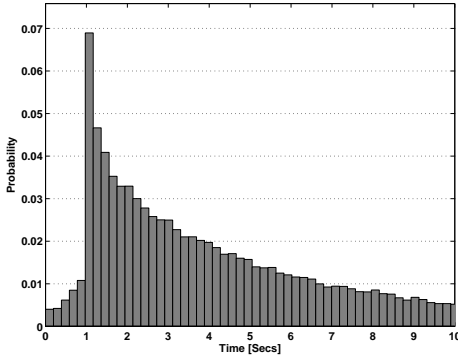


Figure 3: Histogram of utterance durations in conversations of the SWITCHBOARD speech database.

The goal of discriminative is to give preference to code-vectors that have less ambiguity in representing the speaker. One approach would be to identify the code-vectors that are very close to code-vectors of other speaker models and not include them in the speaker identification process. However, a feasible approach would be to weigh the distances between code-vectors and test data proportionally to their discriminative capability. The use of a proximity measure is chosen against the distance metric as the weighing scheme to be discussed can be intuitively understood. It is defined as:

$$\wp_{ijk} = \exp \{-\gamma \|\mathbf{x}_i - \mathbf{c}_{jk}\|_2\}, \quad (8)$$

and represents the proximity measure between the i^{th} frame and the k^{th} code-vector of the j^{th} speaker model. The constant γ was chosen equal to 2 based on experimentation. Note that the range of the distance d is $[0, \infty)$ and that of the proximity measure \wp_{ijk} is $(0, 1]$.

The speaker identification is performed as follows: Let \mathbf{x}_i , $i = 1, 2, \dots, M$ represent feature vectors corresponding the M frames in the utterance and $\wp(\mathbf{x}_i, \mathbf{C}_j) = \max_k \wp_{ijk}$ is the proximity measure between speech frame \mathbf{x}_i and the speaker model \mathbf{C}_j . The code-vector closest to the speech frame is said to have the index k' .

A weighted similarity measure is computed between the utter-

ance and the speaker model as:

$$\wp(X, \mathbf{C}_j) = \frac{1}{M} \sum_{i=1}^M w_{jk'} \wp(\mathbf{x}_i, \mathbf{C}_j), \quad (9)$$

where w_{ik} is a discriminative weighting function associated with each code-vector and is computed as follows:

$$w_{ik} = \frac{1}{\sum_{j \neq k} \wp(\mathbf{c}_{ik}, \mathbf{C}_j)}. \quad (10)$$

The equation in (10) determines the summed proximity measure between each code-vector and closest code-vectors in the other models and the reciprocal serves as the desired weighting function. A speaker model which yields the largest weighted similarity measure as in (9) is chosen as the identity for the utterance. The identity of all the utterances in the conversation is similarly determined.

2.3. Pruning the Model Set

The next operation in the sequence is the removal of extraneous models from the generic model set. This operation is key in determining the number of models needed to represent the presented data. The pruning of the model set is carried out on a model quality measure Q computed based on two factors. The first factor is how well the model can represent the data (P_f), and the second factor is the amount of data it represents (P_d). The use of in the quality measure is obvious; however P_d plays an important role in removing models that are chosen spuriously. The quality measure is thus defined as:

$$Q = \lambda P_f + (1 - \lambda) P_d \quad (11)$$

where λ is a tuning parameter and determined based on the overall performance of the method. The quantity P_f is the average proximity measure between the model and the all the speech frames associated with the model, and P_d is determined as the ratio of the data length represented by the model to the overall length of the data being analyzed. The usefulness of the data length parameter is explored in a separate study [7]. The quality measure is computed for all the models in the generic model set and the models that do not meet a set level of quality are removed from the set. The quality measure threshold is currently left as a free parameter and will be chosen based on experimental evaluation. Note that the models that were not selected by the test data under analysis will be automatically removed. The advantages of the pruning operation are two-fold: 1) it is the key operation which results in the speaker count and 2) it reduces the computational complexity of the next iteration to a much lower level as compared with the current iteration.

2.4. Constrained Adaptation

The models that survived the pruning operation are subjected to the adaptation operation, where the parameters of the model are adjusted to better represent the associated data. This is an important operation for obtaining speaker specific models for the conversation. The adaptation is performed in a constrained manner as the models tend to over fit the data. The constraint is applied on the rate of change of the code-vectors in each iteration. Mathematically this can be represented as:

$$\|\mathbf{C}_j - \tilde{\mathbf{C}}_j\|_F < \xi \quad (12)$$

where \tilde{C}_j is the adapted speaker model and $\xi > 0$ is a constant chosen based on the operating conditions. The constrained adaptation can be graphically visualized and is presented in figure 4 below.

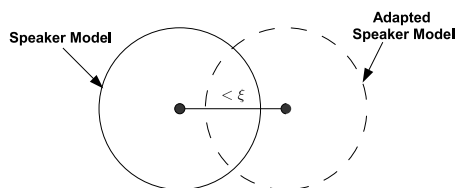


Figure 4: Graphical representation of the constrained adaptation operation.

2.5. Update Model Set

The final operation is to update the generic model set with the pruned and updated set. The sequence of operations is repeated, with the updated model set, until no change in any of the models is observed. During the update step, a foreign model is chosen from the standard database set, and is inserted into the generic model set. This inclusion is expected to help if an error occurs in the pruning stage

3. Experiments and Results

3.1. Model Specifications

The initial generic model set consisted of 20 speaker models. The speaker model is a 32 code-vector vector-quantization model. The model order chosen is lower compared to the model order chosen in standard speaker identification algorithms, and this helps to maintain generality of the speaker models. The choice of a low model order is further supported due to the fact that these models can be easily adapted to represent a specific speaker in the conversation. The models were generated using data from the HTIMIT database [8] to form the standard database set. A total of 384 speakers available in the HTIMIT database were used. The short-time LPCC (Linear Predictive Cepstral Coefficients) were computed every 10msec on 20msec speech windows. The order of LPCC analysis was chosen at 14. The speech data was pre-emphasized and was passed through an unvoiced speech removal pre-processor before computing the short-time features.

3.2. Results

4. Conclusions

This is the conclusions

5. Acknowledgements

This effort was sponsored by the Air Force Research Laboratory, Air Force Material Command, and USAF, under agreement number FA8750-04-1-0146.

6. References

[1] Reynolds, D., A., and Rose, R., C., "Robust Text-Independent Speaker Identification using Gaussian Mixture

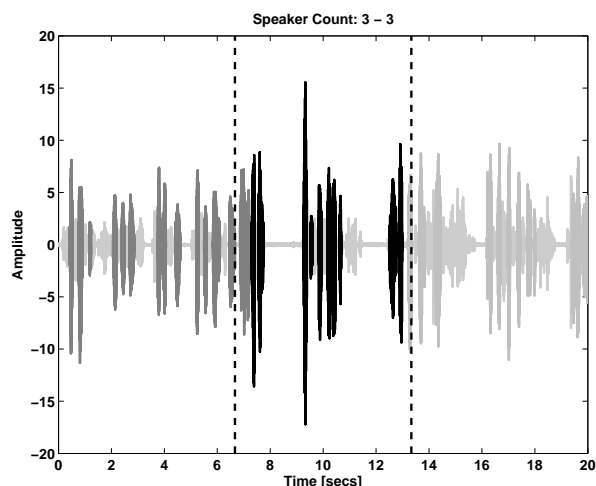


Figure 5: Illustrations of speaker modelling on conversational data: the data associated with each model is represented by a different shade of gray in a three-speaker conversation. The vertical dashed line shows the boundaries of speaker homogeneous utterances.

Models", IEEE Trans. on Speech and Audio Processing, vol. 3 (1), pp.72-83, January, 1995.

- [2] Kown, S., and Narayanan, S., "Unsupervised speaker indexing using generic models", IEEE Trans. on Speech and Audio Processing, vol. 13 (5), pp.1004-1013, September, 2004.
- [3] Nishida, M., and Kawahara T., "Unsupervised speaker indexing using speaker model selection based on bayesian information criterion. In Proc. of ICASSP, vol. 1, pp.172-175, Hong Kong, April, 2003.
- [4] Wu T., Lu, L., Chen, K., and Zhang, H., "UBM-based real-time speaker segmentation for broadcasting news", In Proc. ICASSP, vol. 2, 2003, pp. 193196.
- [5] Kwon, S. and Narayanan, S., "Speaker model quantization for unsupervised speaker indexing". In Proc. of ICSLP, Jeju, Korea, October, 2004.
- [6] Godfrey, J., J., Holliman, E., C., and McDaniel, J., "SWITCHBOARD: Telephone speech corpus for research and development," In Proc. ICASSP, pp. 517-520, 1992.
- [7] Ofoegbu, U., O., Iyer, A., N., Yantorno, R., E., Wemndt, S., J., "A Novel Approach to Automatic Three-Way Call Detection", ICSLP, 2006 (submitted).
- [8] Reynolds, D., "HTIMIT and LLHDB : Speech corpora for the study of handset transducer effects", In Proc. ICASSP, pp. 1535 -1538, 1997.