A HIERARCHICAL APPROACH TO LARGE-SCALE SPEAKER RECOGNITION

Homayoon S. M. Beigi, Stéphane H. Maes, Upendra V. Chaudhari and Jeffrey S. Sorensen

Human Language Technologies Group
IBM Research, T.J. Watson Center
P.O. Box 218, Yorktown Heights, NY 10598
EMail: beigi@watson.ibm.com

ABSTRACT

This paper presents a hierarchical approach to the Large-Scale Speaker Recognition problem. In here the authors present a binary tree data-base approach for arranging the trained speaker models based on a distance measure designed for comparing two sets of distributions. The combination of this hierarchical structure and the distance measure [1] provide the means for conducting a large-scale verification task. In addition, two techniques are presented for creating a model of the complement-space to the cohort which is used for rejection purposes. Results are presented for the drastic improvements achieved mainly in reducing the false-acceptance of the speaker verification system without any significant false-rejection degradation.

1. INTRODUCTION

Let us consider a possible model for speech as being a collection of distributions (e.g., Gaussian distributions). To be able to rank speakers within a database based on their similarity of speech characteristics, one needs a distance measure which would be appropriate for comparing sets of distributions. Once this distance measure is established, a ranking process may be applied to order the speakers in a database, in a hierarchical fashion for future reference. Last year, the authors presented a method for computing a meaningful distance between two collections of statistical distributions which is very useful for ranking models consisting of collections of distributions. [1] Once such hierarchical structure is established for the speakers in the database, the job of cohort computation becomes much easier. This paper presents a classification technique for the speakers in a database based on a binary tree structure and provides the means for a quick computation of the cohort for any speaker in the tree. Then, false-rejection and false-acceptance results are given on a database of 184 speakers.

The way the speaker verification is implemented, the claimed speaker ID is used to find the cohort of the speaker from the binary tree by considering the speakers whose models are children of the same parent some number of generation up the tree.

The above speaker verification will have limited rejection capabilities. Two techniques for reducing the false-acceptance of this verification system are presented in the form of models created from a space, complementary to the cohort space. The two techniques have pros and cons associated with them and are named by the authors as Graduated Complementary Model (GCM) and Cumulative Complementary Model (CCM). The details of the derivation of these two complementary models as well as implementation issues are presented along with improvement results for the verification task.

The next section will briefly describe the procedure for building the speaker models [2]. Then, the details of building the binary speaker-tree using the distance measure of [1] are presented after which a brief description of speaker recognition using these models and the speaker-tree is given. Then, two very effective methods are established for creating a rejection mechanism used for speaker verification as well as open-set speaker identification. These methods are shown to reduce the false-acceptance rate of the speaker recognition by presenting results on a speaker verification task conducted over a 184-member database of speakers. Finally, some concluding remarks are given for the improvement of the hierarchical structure to increase performance and accuracy. Please note that the speaker recognition techniques presented here are text and language-independent.

2. MODEL BUILDING

As we mentioned in [2], a speaker model is created as a collection of parameters (Means and Covariances) for a set of Multi-Dimensional Gaussian distributions. These distributions model the features produced by the signal-processing front-end of the engine.

Speaker model $\mathcal{M}_i$ is computed for the i^{th} speaker based on a sequence of M frames of speech, with the d-dimensional feature vector, $\{\vec{f}_m\}_{m=1,\ldots,M}$. These models are stored in terms of their statistical parameters, such as,
$\{\vec{\mu}_{i,j}, \mathbf{\Sigma}_{i,j}, \vec{C}_{i,j}\}_{j=1,\ldots,n_i}$, consisting of the Mean vector, the Covariance matrix, and the Counts, for the

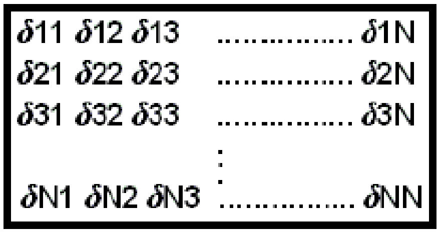
case when a Gaussian distribution is selected. Each speaker, i, may end up with a model consisting of n_i distributions. The distance measure of [1] enables us to devise a speaker recognition system with capabilities for Speaker Identification, Verification and eventually Clustering by creating a hierarchical structure.

3. HIERARCHICAL CLASSIFICATION – BINARY TREE

A binary tree is constructed using the distance measure of [1]. See figure 2. Each speaker model is computed as described above and in detail in [2]. Once the models are created, they are ranked using a a bottom up technique in which each individual model (a collection of multi-dimensional Gaussian distributions) is associated with a distinct speaker and constitutes a leaf of the tree. To perform the primary building operation of the tree, these models are compared with each-other using the distance of [1].

3.1. Pairing

Figure 1 shows a set of sorted distances δ_{km} which associated with speakers i and j. Please note that k and m are the indices of the sorted list and generally differ from i and j. The sorting is done in a way that $\delta_{1m} = 0$. Then, going down the table and left to right, the pair ij with the smallest distance δ_{km} are paired based on the next available non-paired speakers with the smallest distance between their models. Due to the nature of the distance measure, these distance computations between models are orders of magnitude faster than a traditional Maximum Likelihood approach.



$$\begin{array}{llll} \delta 11 & \delta 12 & \delta 13 & \ldots\ldots\ldots\ldots\ldots\, \delta 1N \\ \delta 21 & \delta 22 & \delta 23 & \ldots\ldots\ldots\ldots\ldots\, \delta 2N \\ \delta 31 & \delta 32 & \delta 33 & \ldots\ldots\ldots\ldots\ldots\, \delta 3N \\ & & & \vdots \\ \delta N1 & \delta N2 & \delta N3 & \ldots\ldots\ldots\ldots\ldots\, \delta NN \end{array}$$

Figure 1: Pairing Speaker Models

3.2. Merging

Once all speaker pairs are determined, each pair of models is merged using the following technique for producing a new model with the characteristics of both contributing models. Figure 4 shows a small example with two models being merged, each having a different number of Gaussian distributions associated with them. The superscript in the notation denotes the model number and the subscript denotes the distribution number. Please note that the pairing of the distributions follow the techniques given in [1]. The merged Gaussian distribution with the left and right subscripts i and j denotes the distribution created from the i^{th} distribution of model 1 and the j^{th} distribution of model 2. The counts for the merged distributions are simply the sum of counts of the two building distributions. The new model will have the same number of distributions as the maximum of the two models used in its conception. S_x and S_{x^2} denote the first and second order sums of the feature data. These parameters are used as an alternative set of parameters defining the Gaussian distributions of interest.

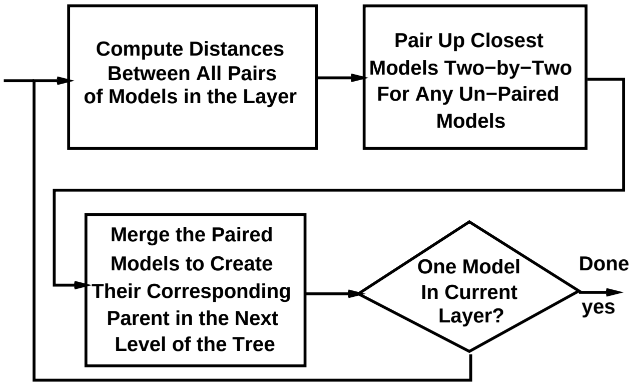

Compute Distances Between All Pairs of Models in the Layer
Pair Up Closest Models Two−by−Two For Any Un−Paired Models
Merge the Paired Models to Create Their Corresponding Parent in the Next Level of the Tree
One Model In Current Layer?
Done
yes

Figure 2: Tree Building Procedure

Each merged pair of models creates a new parent model for the two, in the next level of the binary tree. If the number of models in a level are not divisible by two, then the remaining model in that level may be merged with the members of the next generation (level) in the tree. As each level of the tree is created, the new models in that generation are treated as new speaker models containing their two children and the process is continued layer by layer until one root model is reached at the top of the tree. In this structure, finding the cohort of a speaker is as simple as matching the label of the claimed ID with one of the leaf members; going up the tree by as many layers as desired (based on the required size of the cohort); finally, going back down from the resulting parent to all the leaves leading to that parent. The models in these leaves will be the closest speakers to the claimed speaker.

4. SPEAKER RECOGNITION

Corpus	False Rejection	False Acceptance
No CM	4/60 = 6.66%	11/60 = 18.33%
CCM	5/60 = 8.33%	5/60 = 8.33%
GCM	5/60 = 8.33%	0/60 = 0%

Figure 3: Speaker Verification Results

4.1. Speaker Verification

The training data is stored using a hierarchical structure so that accessing the models would be optimized at the time of recognition. The Speaker Verification is implemented by extracting a set of speakers (with their models) from the training database considering only those speakers with close proximity, as given by the distance measure of [1], to the speaker with the claimed ID. The claimant's sample speech is then used to generate a test model which is compared to the models in the Cohort set. The models are sorted by the distance and the training model with the smallest distance from the test model is used to obtain the verification result. If the background model or any speaker other than the claimant comes up at the top of the sorted list, the claim is rejected. Otherwise, it is accepted.

Alternatively, a thresholding method may be used to compare the likelihood of the input speech given the claimant model versus the average likelihood given the rest of the cohort members.

$\mathcal{M}_i = \{\vec{\mu}_{i,j}, \mathbf{\Sigma}_{i,j}, p_{i,j}\}_{j=1,\ldots,32} = \{\Theta_{i,j}\}_{j=1,\ldots,32}$, denotes the set of speaker models, consisting of the mean vector, diagonal covariance matrix, and mixture weight for each of the 32 components of the i^{th} 12-dimensional Gaussian Mixture Model (GMM) used to model the training data.

The test data is denoted as $O = \{\vec{f}_n\}_{n=1,\ldots,N}$, and we assume that it is i.i.d. Let $\mathbf{\Sigma}_{i,j}(k)$ denote the variance of the k^{th} dimension. Given the observed testing data and an identity claim i, verification proceeds by first computing

$$\log P(O|\mathcal{M}_i) = \sum_{n=1}^{N} \log \left[\sum_{j=1}^{n_i} p_{i,j} p(\vec{f}_n|\Theta_{i,j}) \right] \qquad (1)$$

where, $p(\vec{f}_n|\cdot)$ is a Normal pdf. We compare this to

$$\sum_{j \,\in\, \text{cohort - i}} w_j \log P(O|\mathcal{M}_j),$$

where we chose w_j to be uniform. This is an approximation of

$\log P(O|\{\text{cohort of } \mathcal{M}_i\} - \mathcal{M}_i)$.

The verification score used in obtaining the ROC curves presented later is given by the difference of these two values. The procedure is thus text-independent.

4.2. Open-Set Speaker Identification

In case of Speaker Identification, the claimant's test model may be compared, using the same distance measure, to all the models in the database including that of the background model. This may be expedited using a top-down sweep of the tree to arrive at the correct leaf with only $log_2 N$ comparisons, each time going in the direction of the child with the smallest distance. Please note This may constitute a rejection if the background model is the closest model to the test model.

4.3. Speaker Classification

Speaker Classification as a direct product of the tree building is very useful in many different occasions including the narrow-down of the search space for doing Speaker Recognition. The systems presented in [3, 4, 5] use speaker classification for performing speaker segmentation as well as improving speech recognition accuracies through adaptation.

5. INITIAL RESULTS

Please also note that if the claimant is an imposter and just happens to be closest to the claimed identity in the cohort which is picked, with the probability of $1/(CohortSize)$ a false-acceptance is reached. The first row of results presented in the table of figure 3 present the false-rejection and false-acceptance results conducted on 60 speakers out of a population of 184 speakers in the database. This data is collected using nine different microphones including *Tie-Clip*, *Hand-Held* and *Far-Field* microphones. The training data lasts an average of 40 seconds. The test was performed using an average of 6 seconds of independent data. 60 of the 184 speakers were randomly used for the testing.

The next section presents two novel techniques for solving the false-acceptance problem of verification.

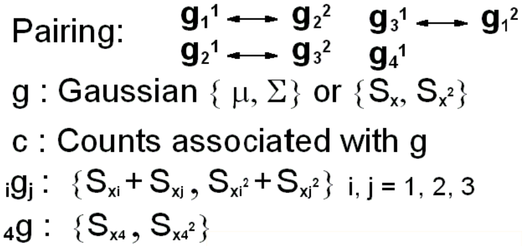

Pairing: $g_1^1 \longleftrightarrow g_2^2$ $g_3^1 \longleftrightarrow g_1^2$
$g_2^1 \longleftrightarrow g_3^2$ g_4^1
g : Gaussian $\{\mu, \Sigma\}$ or $\{S_x, S_{x^2}\}$
c : Counts associated with g
${}_i g_j$: $\{S_{xi} + S_{xj}, S_{xi^2} + S_{xj^2}\}$ i, j = 1, 2, 3
${}_4 g$: $\{S_{x4}, S_{x4^2}\}$

Figure 4: Merging Models

6. COMPLEMENTARY MODEL

Two Complementary Model Techniques are proposed to solve the false-acceptance problem. The first technique will create a single model, used as a representation of all the models in the tree and outside the tree (given some background data). This model is called the Cumulative Complementary Model (CCM) by the authors. CCM is basically a merged model based on the complement of the cohort. Figure 5 shows a speaker-tree with a graphic representation of the models used to create the CCM for an example cohort. Note that this is a very quick computation since the tree structure is used to minimize the computation. The following sections list the model production and pros and cons of the two techniques:

6.1. Cumulative Complementary Model (CCM)

- The complementary model for each node is computed by merging the siblings with the complementary model of the parent as we travel down the tree.

• There no confidence information available by the rejection mechanism. Also, the similar and dis-similar data are merged giving a non-robust merged model. Too many merges are done and since the merging is suboptimal, this will degrade accuracy.
• Decoding is faster in GCM since the modified cohort consisting of the original cohort and the CCM is smaller. Training is slower due to many merges.

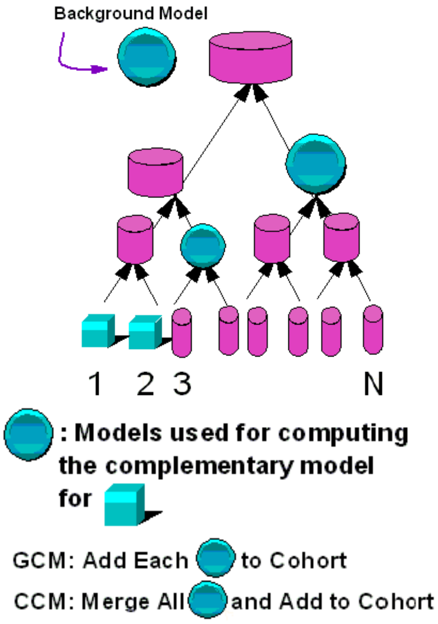

Background Model
1
2
3
N
: Models used for computing the complementary model for
GCM: Add Each to Cohort
CCM: Merge All and Add to Cohort

Figure 5: Speaker Verification Results

6.2. Graduated Complementary Model (GCM)

• The complementary model for each node is the model merged from all its siblings. See figure 5.
• When building the modified cohort, the complementary model of the node and of its parents are added to the cohort list and if the verification finds one of these complementary models to be the closest to the test speaker, it is rejected.
• There is an inherent confidence level associated with this method. The higher the level (closer to root), the more confident the rejection decision.
• No merges are necessary, hence the training is faster than CCM, but the testing is slower.

7. FINAL RESULTS AND CONCLUSION

The background model denoted in figure 5 may be computed by obtaining a lot of data not present in the tree and pooling the data together to create a single model. This will allow further rejection capability for imposters who were not enrolled in the database.

7.1. Coplementary Model Results

The table of figure 3 shows a drastic reduction in the false-acceptance of the verification system when using the two proposed complementary models. As we had expected, the GCM produces much better results. In fact it reduces the false-acceptance of the system to 0 by not much of a degradation in the false-rejection.

In order to perform a quick speaker identification of $log_2 N$ distance computations versus N, the tree should be optimized for better top-down performance. This allows an Identify and Verify scheme for better performance of the verification as well, when compared to using the claimed ID as the cohort identifier. The authors are currently working on this optimization problem.

7.2. Likelihood-Based Verification Results

Using the Likelihood-Based scheme, we have obtained the following preliminary results which take into account mis-match conditions. All training data for a given speaker was collected from only one of 8 microphones. The testing data for the speaker was collected on the training microphone (the matched case) as well as on one of the other 8 microphones (the mismatched case). The imposter trials can be from any of the 8 microphones.

In the experiments 28, (male and female) speakers were used, however for any given piece of training or testing data, the gender was unknown. In addition, we tried to get an even distribution of microphones for training and testing. We limited the amount of training and testing data to approximately 10 seconds. There were a total of 125 speakers in the tree. There were 199 matched verification tests, 214 mismatched tests, and 382 imposter tests. The imposters were taken from a population that excluded any of the enrolled speakers. The equal error rate was 13.8%.

8. REFERENCES

[1] Homayoon S. M. Beigi and Stéphane H. Maes, "A Distance Measure Between Collections of Distributions and its Application to Speaker Recognition", *ICASSP'98*, Seattle, Washington, May 23-27, 1998.

[2] Homayoon S. M. Beigi, Stéphane H. Maes, Jeffrey S. Sorensen, and Upendra V. Chaudhari, "IBM Frame-by-Frame Speaker Recognition Technology", *Speaker Recognition and its Commercial and Forensic Applications*, Avignon, Fr., Apr. 20-23, 1998.

[3] L. Heck and A. Sankar, "Acoustic clustering and adaptation for improved speaker recognition", *Proceedings of Speech Recognition Workshop, ARPA*, "Chantilly, VA, Feb. 1997.

[4] H. Jin and F. Kubala and R. Scwartz, "Automatic speaker clustering", *Proceedings of Speech Recognition Workshop, ARPA*, "Chantilly, VA, Feb. 1997.

[5] Homayoon S. M. Beigi and Stéphane H. Maes, "Speaker, Channel and Environment Change Detection", *World Automation Congress, ISSCI98*, Anchorage, Alaska, May 18-22, 1998.