BOOTSTRAPPING FOR SPEAKER RECOGNITION

*Walter D. Andrews, Joseph P. Campbell, and Douglas A. Reynolds**

U.S. Department of Defense, *M.I.T. Lincoln Laboratory

ABSTRACT

The technique known as *bootstrapping* or *resampling* has been used effectively in the field of statistics to obtain good estimates of statistics from only a small set of observations. In this paper we explore the use of this powerful technique to aid in improving the performance of a GMM-UBM text-independent speaker recognition system. We apply the bootstrap to the training process in the generation of speaker models for the GMM-UBM system. We also aggregate the outputs of the bootstrap's multiple speaker models in our *bagging* system. Speaker recognition results of our bootstrap and bagging systems are presented on NIST corpora.

1. INTRODUCTION

The technique known as *bootstrapping* or *resampling* has been used effectively in the field of statistics to obtain good estimates of statistics from only a small set of observations [1]. In this paper we propose to explore the use of this powerful technique to aid in obtaining better parameter estimates for the speaker models from a limited amount of training data. The basic idea behind bootstrap estimation is that the combination of multiple parameter estimates from resampling (with replacement) of the limited training data can produce better (lower variance) parameter estimates than a single estimate using all the data in a single iteration. A natural extension of the bootstrap is *bagging*. The aggregation of bootstrap results is known as bagging. Here we present a family of text-independent speaker recognition systems based on bootstrap and bagging concepts.

The current state of the art in speaker recognition is obtained via Gaussian Mixture Models (GMM) adapted from a Universal Background Model (UBM) [2]. The GMM-UBM framework has been shown to work exceptionally well in the area of robust text-independent speaker recognition given enough training data. The normal mode for speaker model training in the GMM-UBM is typically performed in a single pass over the entire training set, where most of the techniques for computing variances of parameter estimates assume a large sample size.

There are a number of potential applications for the bootstrap in the GMM-UBM speaker recognition system. The approach taken in this paper is to perform the bootstrap on the training data, where the parameters of interest are the weights, means, and variances. The input training set is sampled with replacement N times, where N represents the number of training samples. This sampling also represents one bootstrap replication. The sampling process is repeated B times, where B is the number of bootstrap replications, hence the term resampling. For each bootstrap replication, there are S speaker models trained. The UBM is held constant for the entire process. A number of approaches were experimented with to determine the best method for merging the bootstrap replications into a single set of speaker models.

Choosing the “best one” speaker model from each bootstrap replication (a cheating experiment), the bootstrap average, and bagging were tested. The results presented are drawn from National Institute for Standards and Technology (NIST) Speaker Recognition Evaluation (SRE) data sets from 1999 and 2000 [4]. Performance of the bootstrap and bagging GMM-UBM speaker recognition systems is reported as Detection Trade-off Curves (DET), which are produced by sweeping speaker independent thresholds over all test scores and plotting the miss and false alarm rates at each point [3].

2. GMM-UBM

The core text-independent speaker recognition system in this paper is based on the GMM-UBM of Reynolds, et al. [2]. The system operates on mel-cepstral based feature vectors consisting of 19 cepstral coefficients and 19 delta cepstral coefficients. These cepstra are derived from the bandlimited mel-filterbank spectra. The 38 dimensional feature vectors are computed every 10 ms using a 20 ms window. RASTA processing is used for channel compensation.

The GMM-UBM speaker recognition system is a likelihood ratio detector consisting of speaker models and a gender-independent universal background model. The UBM contains 1,024 male mixtures and 1,024 female mixtures, thus giving a 2,048 mixture UBM. The speaker or claimant models are derived from the UBM via Bayesian adaptation [2]. For a test utterance, the verification score for a given speaker is the difference between the log-likelihoods of the claimant and the UBM. In this paper no handset or score normalization was used since we are primarily concerned with improvements relative to a baseline.

The UBM for the 1999 and 2000 NIST evaluations were generated using test utterances from the 1997 and 1996 NIST SRE data sets, respectively. The speaker training data for the 1999 NIST evaluation consists of two 1-minute speech segments and the 2000 NIST evaluation consists of one 2-minute speech segment [4].

3. BOOTSTRAP

3.0. Generalized Bootstrap

The bootstrap is a computer-based method for assigning a measure of accuracy to the statistic of interest [1]. The generalized bootstrap is accomplished by resampling with replacement the measured data $\boldsymbol{x}$, where the length of $\boldsymbol{x}$ is N.

Each of the samples in the measured data set ***x*** have equal weighting 1/*N*. This resampling process produces the bootstrap samples ***x****, which are also of length *N*. Some of the samples in ***x*** will not be represented by ***x**** and other samples will appear more than once. This general bootstrap process is shown in Figure 1.

The typical number of bootstraps *B* has been reported to be between 50 – 200 [1]. After the resampling process, the statistic of choice is computed for each bootstrap, thus producing the bootstrap statistic ***v***. This statistic is used in a number of applications, e.g., the bootstrap standard error for calculating confidence intervals [1], threshold selection for signal detection algorithms, and clustering.

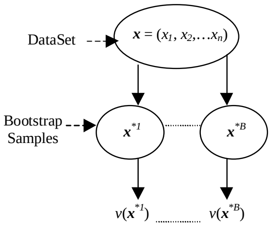

Figure 1. Generalized Bootstrap

3.1. Bagging

Bagging is an acronym for "bootstrap and aggregating" [5]. Here we use bagging to generate multiple versions of a classifier and aggregating them to obtain a single classification result. Multiple versions of the classifier are obtained via the bootstrap described in Figure 1, thus the bootstrap portion of bagging. Breiman showed that bagging can provide an improvement given an unstable classifier (small perturbations in the training data cause large changes in the constructed classifier) [5]. This result allows the bootstrap classifiers to be combined into a single, more powerful classifier. Bagging has been successfully demonstrated on a basic vector quantization (VQ) classifier for speaker recognition operating on the closed-set TIMIT corpus [6].

4. EXPERIMENTS

4.0. Baseline System

The baseline system in this paper is a GMM-UBM, which uses all of the training data provided by the 1999 and 2000 NIST SREs unless stated otherwise [7,8]. For the "cheating" experiment described in section 4.1, a subset of speaker models were trained and tested from the 1999 NIST SRE data set. The speaker models in the cheating experiment were trained using mean-only adaptation.

For the bootstrap experiment, all the speaker models in the 2000 NIST SRE data set were trained and tested. Again the speaker models were trained using mean-only adaptation.

The bagging experiment used a subset of speakers from the 2000 NIST SRE data set. All the training data was used to train the subset of speaker models. The speaker models were trained by adapting the weights, means, and variances.

4.1. "Cheating" System

The first experiment was the "cheating" approach. Given *B* bootstrap models, choose the best model for each speaker with respect to equal error rate (EER). In this experiment, *B* = 50. The block diagram of the "cheating" system is shown in Figure 2. The "cheating" experiment was performed on the 1999 NIST SRE data set on a subset of speakers (40 female and 35 male).

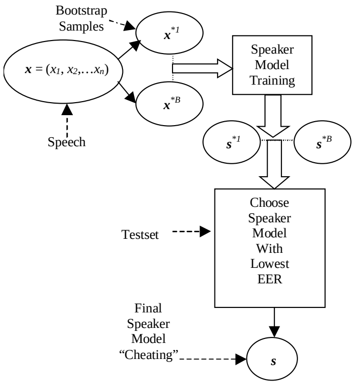

Figure 2. "Cheating" System

The speaker training data for the 1999 NIST evaluation is supplied in two 1-minute segments. For the purpose of the bootstrap, the speaker training data is split into 12 contiguous 10-seond segments totaling approximately 120 seconds. The resampling is then performed on the segmented training data.

The baseline (reference) system uses the two 1-minute cuts for mean-only adaptation for generating the speaker models. A second system, which validates the speaker training data segmentation process, is referred to as the segmented baseline. The segmented baseline system uses all of the segmented data for speaker model training. It in effect uses the same data for speaker model training as the baseline system, except the data is presented to the training algorithm in a segmented fashion. The third system is developed by "cheating." The *B* bootstrap models for each speaker are scored individually on the test set. The bootstrap speaker model with the lowest EER, $\tilde{s}$, is selected to represent a particular speaker in the cheating system via

$$\tilde{s} = \arg\min\left(EER\left[s^*_{m,b}\right]\right),$$

where *m* is the speaker, 1….*M*, and *b* is the bootstrap, 1…*B*. This selection process is performed for all possible speakers. The DET curves for these systems are provided in Figure 3.

The solid line represents the baseline system and the dotted line represents the segmented baseline system. Note the similarity of the baseline system compared to the segmented baseline system. This suggests that the segmentation of the speaker training data into smaller segments is not detrimental to the speaker model training process. The dashed line represents the cheating system.

It is obvious from Figure 3 that a better set of speaker models may exist, but the essential question is “How do we select these better speaker models without cheating?” One approach, discussed next, is using the statistical technique referred to as bootstrap or resampling.

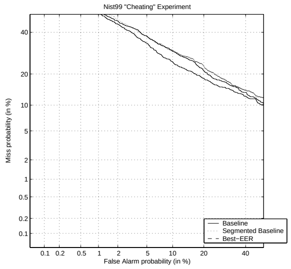

Figure 3. DET Curve for "Cheating" Experiment

4.2. Bootstrap Results

This experiment concentrated on techniques for combining the bootstrap speaker models. This experiment trained speaker models on the entire speaker set and tested the speaker models as specified in the 2000 NIST SRE plan [4]. The speaker training data was supplied as one 2-minute segment. As in the cheating experiment, the baseline speaker models are trained using the single 2-minute segments while the segmented baseline speaker models are trained using segmented data as described in section 4.1. The bootstrap system for this experiment is shown in Figure 4.

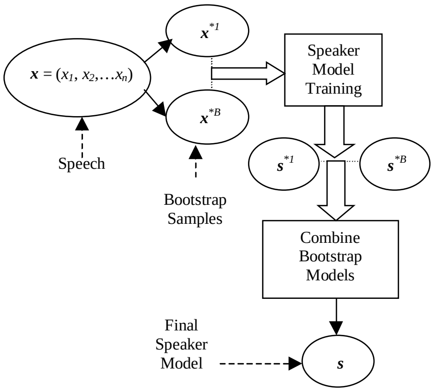

Figure 4. Bootstrap System

For this experiment, 10 and 21 bootstrap models were generated per speaker with the 10 included with the 21. These models were combined using a simple averaging function. This was reasonable given that the speaker models were generated using mean-only adaptation. The average model is generated by computing the sample mean over each of the gaussians in the bootstrap speaker models via

$$\bar{s}_g = \frac{1}{B}\sum_{b=1}^{B} s^*_{g,b} \ ,$$

where g represents the number of gaussians in the speaker model; which for this experiment $g = 1\ldots 2{,}048$, the same size as the UBM. This averaging is computed for both the 10 and 21 bootstrap models.

Figure 5 shows the baseline, segmented baseline, and a number of individual bootstrap systems (not averaged). The individual bootstrap systems each performed worse than the baseline and segmented baseline, as expected. The interesting aspect of Figure 5 is the close clustering of the individual bootstrap systems’ DET curves.

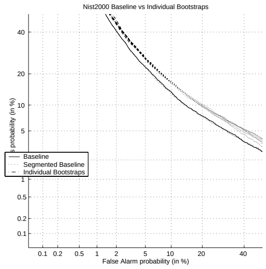

Figure 5. Baseline versus Individual Bootstraps

Figure 6 shows the result of the averaging process for 10 and 21 bootstrap speaker models. As expected, the averaged bootstrap systems performed better than the individual bootstrap systems. Although the bootstrap systems did not perform any better than the baselines. The median of the bootstrap models was also tried and it yielded nearly identical performance to the averaged bootstrap. The disappointing result in all these cases is that the bootstrap systems converged to the baseline, thus showing no improvement over the baseline system.

4.3. Bagging Results

The natural extension of the bootstrap is to apply the bagging procedure. This experiment used the bootstrap system (except the bootstrap models were not combined into a single speaker model set) where the scores of the bootstrap systems were combined to form a single score. A subset of speakers (35 female and 27 male) from the NIST 2000 SRE data set were trained and tested. The general block diagram is shown in Figure 7. The scores from the bootstrap system can be represented in a score matrix where the first index identifies the bootstrap and the second index identifies the speaker.

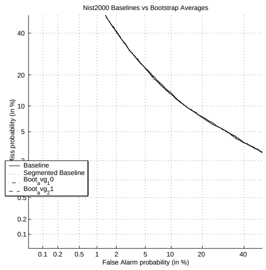

Figure 6. Baselines versus Bootstrap Averages

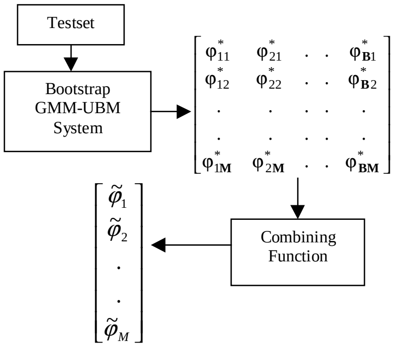
Testset
Bootstrap GMM-UBM System
φ^*_{11} φ^*_{21} . . $\varphi^*_{\mathbf{B}1}$
φ^*_{12} φ^*_{22} . . $\varphi^*_{\mathbf{B}2}$
$\varphi^*_{1\mathbf{M}}$ $\varphi^*_{2\mathbf{M}}$ . . $\varphi^*_{\mathbf{BM}}$
Combining Function
$\tilde{\varphi}_1$
$\tilde{\varphi}_2$
$\tilde{\varphi}_M$

Figure 7. Bagging System

In this paper the combing function used is a simple averaging of the *B* bootstrap scores φ^*_{bm} for each speaker *m*. This is defined as

$$\tilde{\varphi}_m = \overline{\varphi}_m = \frac{1}{B}\sum_{b=1}^{B} \varphi^*_{m,b} \ ,$$

where *m* represents the particular speaker; for this experiment *m* = 1…*M*. The performance for a given set of *B* bootstrap models for full adaptation is shown in Figure 8. The number of bootstrap scores averaged is 2, 4, 6, 8, and 10. As the number of bootstrap models averaged is increased from 2 to 10, the performance of the bagging system improves. Although the system does converge rather quickly as the number of bootstrap models averaged is increased.

Figure 9 shows the comparison between the baseline system and the bagging system. The results are shown for bootstrap scores averaged for 2 and 8 systems. The system with two bootstrap averages performs worst while the one with 8 bootstrap averages performs best with a slight improvement over the baseline. The median of the bootstrap scores for each speaker was also tried and it yielded nearly identical performance to the average bootstrap scores.

5. CONCLUSION

This paper demonstrated via the "cheating" experiment that a better set of models generated via the bootstrap does exist. The question of finding these models is yet to be solved. It also showed that simply combining the bootstrap models with a simple averaging function provided no improvement over the baseline system, independent of the number of bootstraps averaged. As the number of bootstraps used in the averaging increased, the speaker recognition system converged to the baseline system. The bagging system showed more potential and even provided some slight improvement over the baseline. This system also converged quickly as the number of bootstraps increased. There are other potential applications for the bootstrap in a GMM-UBM speaker recognition system that we will be exploring, such as, threshold selection an important aspect of the system.

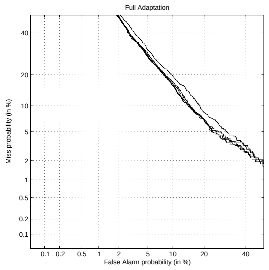

Figure 8. Bagging Results

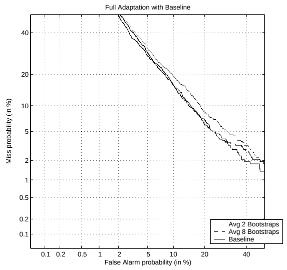

Figure 9. Baseline versus Bagging

6. REFERENCES

1. Efron, B., and Tibshirani, R., *An Introduction to the Bootstrap*, Chapman and Hall, New York, 1993.

2. Reynolds, D., Quatieri, T., and Dunn, R., "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing* 10 (2000), pp. 19-41.

3. Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Pryzbocki, M., "The DET Curve in Assessment of Detection Task Performance," *Proc. Eurospeech 1997*, Vol. 4, pp. 1895-1898, 1997.

4. NIST Speaker Recognition Evaluation Plans, website: www.nist.gov/iaui/894.01/.

5. Breieman, L., "Bagging Predictors," *Machine Learning*, Vol. 24, pp. 123-140, 1996.

6. Kyung, Y. and Lee, H., "Bootstrap and Aggregating VQ Classifier for Speaker Recognition," *Electronics Letters*, Vol. 35, No. 12, June 1999.

7. Martin, A. and Przybocki, M.,"The NIST 1999 Speaker Recognition-An Overview", *Digital Signal Processing* 10 (2000), pp. 1-18.

8. Andrews, W. and Campbell, J., "Walt's and Joe's Excellent Adventure," *2000 NIST Speaker Recognition Evaluation*, June 25 – 26, Linthicum, MD, 2000.