AN ADAPTIVE SPEAKER VERIFICATION SYSTEM WITH SPEAKER DEPENDENT A PRIORI DECISION THRESHOLDS

Nikki Mirghafori and Larry Heck

Nuance Communications, 1380 Willow Rd., Menlo Park, CA 94025
{nikki,heck}@nuance.com

ABSTRACT

This paper presents a practical approach to deploying *a priori* speaker dependent thresholds (SDT) for adaptive speaker verification applications. Our motivations for exploring SDTs are two fold: one is to eliminate the externally pre-set overall system thresholds and replace them with automatically-set internal thresholds calculated at runtime; the second is to counter the verification score shifts resulting from online adaptation. The second motivation is based on the observation that after adaptation, verification scores for both true speakers and impostors increase, which in turn increases the false accept (FA) rates. The rise of FA rates, in an adaptive system, can be costly because of the possibility of model corruption. In this work, an approach similar to ZNORM [3] is used to calculate a threshold for each speaker, which is automatically updated every time the claimant model is adapted. The paper explores various computational efficiency strategies to make the deployment of this approach practical for a fielded system. Results of experiments on one Japanese and one English digits database are presented.

1. INTRODUCTION

Setting thresholds appropriately for a speaker verification application is a challenging task. If there is a mismatch between the development test in the lab and the real world test material, the effective operating point of the fielded application could be different than expected. Furthermore, the customer's desired operating point may not be the same as the pre-set threshold. For example, a financial application may need to operate in the "high security" region (lower FA rate, higher FR rate) whereas a voice portal may choose to operate in a "high convenience" zone (higher FA rate, lower FR rate). Obviously, a one-size pre-set threshold would not fit all applications.

One of our motivations in this work is to allow the user to set the operating point for the application according to the desired security level. In addition to the ability to specify the desired operating point, it is important for an application to perform *consistently* for all users. That is, it is not sufficient to have an overall low error rate for the system if there are users for whom the system works very poorly. It is more desirable to have a consistent behavior and avoid the risk of irate customers. Speaker dependent thresholds (SDT) are an attractive option, where the threshold for each speaker is calculated and saved in each speaker model. In this way, the system may accomplish consistent error rates for both *goats* and *sheep*.

A second motivation for exploring SDTs is to improve the functionality of online speaker adaptation. Adaptation techniques have long been known to improve accuracy both in speech and speaker recognition. The gains are particularly significant for speaker recognition, where a claimant model must be created from little enrollment data. As a side-effect of online adaptation, undesirable score shifts in both speech and speaker recognition have been observed [2, 9]. For speech recognition, confidence scores of both in-grammar and out-of-grammar utterances increase and an approach has been developed to automatically map post-adaptation scores to pre-adaptation scores [9]. This side-effect can be particularly problematic in speaker verification, because as the impostor scores increase, the probability of adapting and corrupting the claimant models on impostor data also increases. Countering this drift in scores has previously not been addressed in the speaker verification community. This is one of the original contributions of this paper.

In Section 2 the algorithm for SDT calculation is discussed. Practical issues are addressed in Section 3. In Section 4 we present the experimental results of setting automatic *a priori* thresholds and countering score drifts post adaptation. Conclusions and future work are discussed in Section 5.

2. APPROACH

There have been various approaches to setting SDT in speaker verification [10, 4]. SDTs may be set to either optimize the overall equal error rate (EER) and/or set the operating point of the system for a certain FA rate[1]. For fielded applications, the security level of the system, or FA rate, is of utmost importance. Our goal is to calculate internal thresholds automatically so that the system operates at the specified FA rate.

Our approach to SDT calculation is a score normalization approach based on ZNORM [3]. The basic idea of this approach is to normalize the verification score according to the mean and standard deviation of the impostor distribution, namely:

$$S_{M,norm} = \frac{S_M - \mu_{M,imp}}{\sigma_{M,imp}} \tag{1}$$

where $S_{M,norm}$ is the normalized score of a test utterance on claimant model M, S_M is the unnormalized score, and $\mu_{M,imp}$ and $\sigma_{M,imp}$ are the mean and standard deviation of the impostor distribution on claimant model M. $\mu_{M,imp}$ and $\sigma_{M,imp}$ are calculated by running a pre-selected set of impostor utterances (called Impostor Batch, or IB) on a claimant model to generate score distributions.

Assuming that score distribution of the IB utterances are similar to those of the actual impostors in the test situation, the normalized scores of the actual impostor distribution should be similar to a unit

[1] Given the dearth of true speaker data, it is often challenging to set the threshold according to the FR rate.

Normal distribution. Z-scores can be calculated from the inverse normal cumulative distribution. For the unit Normal distribution, a z-score of 1.64, for example, corresponds to a point where 95% of the data lie below and 5% of the data lie above. To set the FA rate to 5%, a z-score of 1.64 would be subtracted from $S_{M,norm}$ and the result would be compared to a threshold of zero. In other words:

$$S_{M,norm,x\%FA} = S_{M,norm} - Z_{@x\%FA} \begin{array}{c} accept \\ > \\ <= \\ reject \end{array} 0 \qquad (2)$$

The desired FA rate can be specified at runtime, which through a lookup table, is mapped to a z-score, $Z_{@x\%FA}$. Other SDT parameters which are claimant dependent, $\mu_{M,imp}$ and $\sigma_{M,imp}$, are calculated using the IB and stored in each claimant model. This calculation is done after enrollment and automatically updated every time the claimant model is updated after online adaptation.

3. PRACTICAL ISSUES

Various practical questions arise: what is the minimum number of utterances needed in the IB to get a reliable estimate of the mean and the standard deviation? How should the impostors be selected? Should they be channel or gender dependent? Would the implementation burden make this approach computationally viable in conjunction with online adaptation, given that the thresholds have to be updated after every occurrence of online adaptation? These questions are addressed in the following sections.

3.1. The Shape of the Impostor Distribution

In Section 2, where we explained the algorithm, we surmised that the normalized scores distribution would be similar to a unit Normal distribution. The tails of the distribution, however, do not seem to be identical to a Normal distribution. Since it is often desirable to set FA rates low, we are most interested in the z-scores in the tail of distribution where using the Normal distribution may be a poor approximation.

We experimented with generating the impostor distribution empirically on a held-out dataset and calculating the z-scores for the resulting distribution. The empirically derived z-scores were then compared to Normal z-scores for SDT calculation and used in Equation 2. The claimant and impostor utterances were chosen from a Nuance internal digits database (3K true speaker and 10K impostor trials) and the z-scores were calculated using the impostor distributions of the NIST [6] 96 and 98 (conversational speech) test databases.

Table 1 shows the difference between the FA rates set using z-scores from either the Normal or the empirically derived distributions. There is a consistent bias in the Normal approximation (roughly 10%). The FA rates from the empirical distributions are closer to the desired FA rate, even though the textual data from the test database (digits) and the empirical impostor distribution (conversational speech) are clearly different. We chose to use the empirical distribution, although using the Normal distribution and removing the consistent bias could be another alternative that deserves further exploration.

FA goal	**Normal**	**Empirical**
1%	1.18%	0.94%
2%	2.26%	1.82%
3%	3.41%	3.00%
4%	4.44%	4.22%
5%	5.50%	5.21%

Table 1. The table shows desired and effective FA rates for a digits database calculated with z-scores from either a Normal distribution or calculated empirically from the NIST database. The empirical z-scores produce FA rates closer to the goal.

3.2. Selection of the Impostor Batch

In previous work [8] roughly 200 utterances per handset type per gender was used to determine the mean and standard deviation of the impostor distribution. A large number of utterances are often needed to get a reliable estimate of the standard deviation. We surmised that by selecting impostor utterances such that the speaker space is well covered, the number of utterances may be reduced such that computational requirements are limited. A measure of pair-wise distance [7] was used according to:

$$dist(g_i, g_j) = log\frac{p(X_i|g_i)}{p(X_i|g_j)} + log\frac{p(X_j|g_j)}{p(X_j|g_i)} \qquad (3)$$

where $dist(g_i, g_j)$ is the distance between speaker g_i and g_j. The seed utterance was chosen to be maximally distant from all utterance. Additional utterances were selected sequentially to maximize the average pairwise distance from the selected utterances. In addition to this method, we also used random selection.

The claimant and impostor utterances were chosen from NIST 1998 and the normalization data was selected from NIST 1996. There were 21K female and 25K male verification trials for NIST 1998 test set. DET curves are shown in Figure 1.

Choosing normalization utterances randomly was better than choosing the utterances based on the above distance measure (in terms of EER improvements). This may be because the distance measure had a bias for choosing outlying utterances. As few as 30 utterances were sufficient to observe improvements in verification performance (EER) and saturation occurred around 180 utterances. Similar EER improvements were observed whether both mean and standard deviation were used, or if only the mean was used and the standard deviation was set to be constant (i.e., set to the average of the pooled standard deviations). When using few normalization utterances (e.g., 30), using model means with pooled standard deviation proved to be more effective, most likely because model standard deviation estimates were noisy.

We also observed that IB should be gender matched, but need not be channel-matched to the test data. In the field application, gender detection can be performed on the enrollment utterances to decide which IB gender to use for calculating the SDT parameters. For text-dependent verification, the IB should match the test data as closely as possible. For example, if the test text is digit strings, the IB could be speakers uttering zero through nine.

3.3. Reduction in Computation

In a GMM based system, with cohort and claimant GMMs [7], the implementation could be made efficient by storing the top n

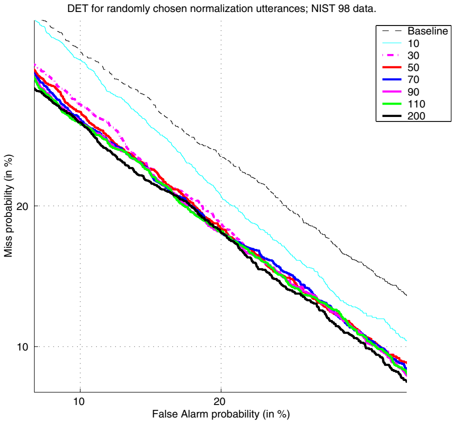


Fig. 1. DET curves for randomly chosen IB utterances. As few as 30 utterances were sufficient to observe improvements in the EER performance.

Gaussians for the cohort models in memory. This results in great savings, as only the top n Gaussian for the claimant model need to be calculated for each utterance in the IB.

Furthermore, by decimating the feature stream, i.e., using every n^{th} feature vector, we can get a further saving in computation. Previous work suggested that a decimation level as high as 10:1 would cause little performance degradation [5]. Our experiments confirmed this. We chose a decimation rate of 12:1. Combined with the above mentioned method, the computational savings amounted to 99%.

4. EXPERIMENTAL RESULTS

4.1. Setting Security Level

4.1.1. Experimental Setup

We tested the setting of security levels on one American English (EDigit) and one Japanese digits (JDigit) databases. EDigit has 230 unique speakers and the testset comprises 230 voiceprints, 324 true speaker (TS) and 53K impostor (IM) trials. JDigit has 162 unique callers and the test set comprises 6K voiceprints, 12K TS and 9K IM trials. Results as well as further information about experimental setup can be found in Table 2. SDTs were calculated during enrollment on the training set and were *not* modified during testing. The differences between the EERs of the baseline system and the system with SDT do not appear to be significant.

Our English IB contained 60 utterances, 30 for each gender. The utterances were randomly selected from another English digits database. Every utterance included digits zero through nine, each spoken once. The Japanese IB contained 90 utterances and the utterances were randomly selected from another Japanese digits databases. Every utterance included a random selection of digits zero through nine, with some repetitions. We chose more Japanese IB utterances to assure that every digit had sufficient coverage. Note that though the textual data is similar in the IBs and the tested datasets, it is not identical.

DB	**EER**	**EER w/**	**# Trials x1000**	**# Digits**		
Name	**Base**	**SDT**	**TS/IMP**	**Enrol**	**TS**	**IM**
EDigit	1.25%	0.92%	0.3/53	6x2+ 6x2+ 10	6	10
JDigit	6.81%	7.05%	12/9	8x3	8	8

Table 2. The table shows EERs and the makeup of the test databases. The column TS/IMP lists the number of true speaker (TS), impostor (IM) trials. The Enrol, TS, and IM columns list the number of repetitions of digits for each condition.

4.1.2. Experimental Results

Table 3 shows the results for setting the desired security level. The results are better for the English Digit database (EDigit). The impostor utterances in EDigit were composed of digits zero through nine, which perfectly matched the textual content of the IB[2]. The overshooting of the goal FA for the other three databases suggests that either the mean and/or the standard deviation of the IB distribution is smaller than that of the actual impostor distribution. We have observed that verification scores for textually mismatched test data are often smaller than those of test data which textually matches the enrollment utterances. An approach similar to [10] or phone based verification [1] may alleviate this problem.

FA goal	**EDigit**	**JDigit**
0.15%	0.16%	0.17%
0.85%	0.88%	0.95%
2.25%	2.45%	3.35%
4.00%	4.23%	5.97%
6.00%	6.01%	9.08%

Table 3. Table shows the desired and effective FA rates for the two datasets with using both model mean and standard deviations.

4.2. Countering Post-Adaptation Score Shift

4.2.1. Experimental Setup

We used a database of Japanese digit strings for the *unsupervised* adaptation experiment. We trained 5K speaker models (on three repetitions of an 8-digit utterance) and tested on 67K mixed-gender impostor trials and 12K true-speaker trials, each composed of one repetition of an 8-digit utterance. The adaptation set contained eight utterances for each speaker model, of which seven utterances were from the true-speaker and one utterance was by an impostor. The rate of impostor attempts in the adaptation set was 12.5%, which compared to impostor attempt rates in the real world, was rather aggressive. The number of impostor attempts were uniformly distributed across all trials.

In the adaptation experiments, the speaker models were first trained on the enrollment data. A held-out verification test set was run to establish the baseline performance. Next, each model was

[2] The IB utterances are were *not* selected from any of the *x*Digit databases.

adapted on one adaptation utterance (only if the verification score was higher than the adaptation threshold). This was followed by a round of verification test on the held-out test set. The last two steps (adapt & test) were repeated eight times.

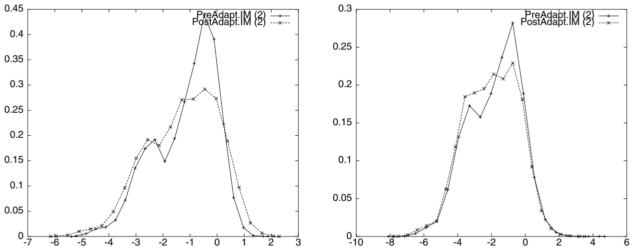
4.2.2. Experimental Results

Table 4 shows the EER, FA, and FR rates on the held out data set before and after adaptation. The increase in the FA rates is well controlled for the system which has SDT, an increase of about 12% in FA rate, as opposed to a factor of almost 4 for the baseline system. This can also be observed in Figures 2, where in Figure (a), the shift of the impostor distribution's right "skirt" to the right hand side is visible. This shift has been arrested in Figure (b). However, there is a price to be paid in terms of the EER. The final EER of the adapted system without SDT is 2.30%, 25% percent less than the final EER of 3.07% for the system with SDT. For the last iteration of adaptation, for example, the impostor corruption rate dropped from 10.14% to 5.96%. Considering the cost of model corruption in a deployed system with rising FAs, this price seems justifiable.

The problem of textual mismatch also affected the efficacy of controlling the FA growth. After successive iterations of adaptation, we observed that the mean of the IB became negative at a faster rate than the actual impostor distribution mean. Smoothing the post-adaptation IB mean with the original IB mean (post-enrollment), where $\mu_{new} = \alpha\mu_{enrol} + (1 - \alpha)\mu_{post-adapt}$, alleviated this problem. An α of 0.9 was used for the results reported in Table 4.

	No SDT			**With SDT**		
Iteration	**EER**	**FA**	**FR**	**EER**	**FA**	**FR**
Baseline	7.13	**1.10**	25.43	7.75	**1.10**	33.70
Adapted	2.30	**4.23**	1.39	3.07	**1.26**	6.96

Table 4. The table shows the increase in FA rates in the baseline system (No SDT) and new system (With SDT) thresholds.



(a) Without SDT

(b) With SDT

Fig. 2. The shift in the impostor distributions with and without SDTs. Note that due to SDT normalization, the range of the scores has changed.

5. CONCLUSION AND FUTURE WORK

This paper presents a practical approach to deploying *a priori* speaker dependent thresholds (SDT) for adaptive speaker verification applications. Our motivation was to devise a system with automatically calculated *a priori* thresholds, where the operating point of the application could be specified at run-time according to the desired FA rate. Our second motivation was to use SDTs to counter the drift in scores which result from online speaker adaptation. We presented efficiency changes which made this approach practical for real-time applications. Experimental results showed that, when the IB data textually matches the actual impostor test data, this approach performs well in setting the internal thresholds. Regardless of the textual matching, however, this method can be effective in controlling the increase in FA rates post adaptation.

As mentioned, one limitation of this approach is that the data chosen in the IB should ideally be matched to the test data in the field. Using an approach similar to [10] or using phone based verification [1] could alleviate this problem. In the case of a phone based verifier, regardless of the textual content of the IB, the phonemes which match either the phonemes in the test material or the ones used to train the claimant model could be sub-selected from the IB and evaluated. By removing this restriction, this approach could be used for text-independent verification, with one universal IB for all applications and, perhaps, languages.

6. REFERENCES

[1] R. Auckenthaler, E. S. Parris, and M. J. Carey. Improving a gmm speaker verification system by phonetic weighting. In *ICASSP*, Phoenix, AZ, 1999.

[2] L.P. Heck and N. Mirghafori. Unsupervised on-line adaptation in speaker verification: Confidence-based updates and improved parameter estimation. In *Proc. Adaptation in Speech Recognition*, Sophia-Antipolis, France, 2001.

[3] Kung-Pu Li and Jack E. Porter. Normalizations and selection of speech segments for speaker recognition scoring. In *ICASSP*, pages 595–597, 1988.

[4] J. Lindberg, J.W. Koolwaaij, H.-P. Hutter, D. Genoud, M. Blomberg, J.-B. Pierrot, and F. Bimbot. Normalizations and selection of speech segments for speaker recognition scoring. In *Proceedings of RLA2C*, pages 89–92, 1998.

[5] Jack McLaughlin, Douglas A. Reynolds, and Terry Gleason. A study of computation speed-ups of the GMM-UBM speaker recognition system. In *EUROSPEECH*, 1999.

[6] NIST. Speaker recognition workshop. In *NIST Workshop Notebook*, Linthicum Heights, Maryland, 1996-2000.

[7] D.A. Reynolds. Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, 17:91–108, 1995.

[8] D.A. Reynolds. Comparison of background normalization methods for text-independent speaker verification. *EUROSPEECH*, 1997.

[9] A. Sankar and A. Kannan. Automatic Confidence Score Mapping For Adapted Speech Recognition Systems. In *ICASSP*, 2002.

[10] A.C. Surendran and C.-H.Lee. A priori threshold selection in fixed vocabulary speaker verification systems. In *ICSLP*, Beijing, China, October 2000.