PARAMETERIZATION OF THE SCORE THRESHOLD FOR A TEXT-DEPENDENT ADAPTIVE SPEAKER VERIFICATION SYSTEM

*Nikki Mirghafori**

International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704
nikki@icsi.berkeley.edu

Matthieu Hébert

Nuance Communications
1380 Willow Road
Menlo Park, CA 94025
hebert@nuance.com

ABSTRACT

In this work we present a computationally efficient strategy for setting *a priori* thresholds in an adaptive speaker verification system. Our motivations are two-fold: one is to eliminate the externally pre-set overall system thresholds and replace them with automatically-set internal thresholds conditioned by a target FA rate and calculated at runtime, and the other is to counter the verification score shifts resulting from online adaptation. Our approach entails calculating the trajectory of the score threshold as a function of 1) length of the password, 2) target FA, and 3) the number of training frames in the speaker model. The solution is successful at both achieving target FA rates and keeping the FA rate constant during online adaptation. Furthermore, it is algorithmically simple and requires negligible computational resources. The threshold function is calibrated on a Japanese database and experimental results are presented on 12 databases in four different languages.

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. A second motivation for this exploration is to improve the functionality of online speaker adaptation [2, 3]. 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 [3, 8]. 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.

In previous work [7] we presented an algorithm to address the issues above. Even though highly optimized, it proved too computationally expensive for an online real-world system. Furthermore, when calibration data lexically differed from the test material, the performance suffered. This work presents a new algorithm with negligible computational cost which is also robust to the phonetic content of the password.

In Section 2 we discuss our approach. In Section 3, we discuss the lessons learned from studying the data. Section 4 includes the details of the calibration. In Section 5 we present the experimental results on databases in four languages. Conclusions and future work are discussed in Section 6.

2. THE APPROACH

There are various approaches to setting speaker dependent thresholds in speaker verification [9, 5, 7]. Thresholds 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 old approach was a score normalization approach based on ZNORM [4]. The basic idea was to normalize the verification score using the mean and standard deviation of the score distribution of the impostor attempts. A set of waveforms were preselected and stored for the calculation of impostor distribution score statistics. To set the threshold, the z-score corresponding to the desired FA rate was subtracted from the normalized score. This approach suffered from two problems: 1) although the normalization step was very quick, the computational cost associated with calculating the scores of the impostor score distribution was large. This cost was compounded by the system being adaptive and the models potentially changing, and hence having to recalculate the impostor population scores after every adapted true speaker attempt; 2) the second problem was that if the lexical content of the impostor data differed from the actual password data, the population statistics were affected, and the predictability of FA suffered. Other normalization approaches such as Hnorm [1] and Tnorm [6] are similarly problematic, as computationally they are too expensive when combined with online-adaptation. To get away from these two problems, we tried an approach which would not be dependent on the fine details of the speaker model which are most

*The first author performed the work while at Nuance Communications.

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

affected by the lexical content of the passwords.

The new approach is based on the simple observation that as the speaker model gets adapted on more data, the scores of both true speakers and impostors increase. To keep the FA rate constant, therefore, the threshold must be increased. If 1) the score threshold could be expressed as a parametric function of the amount of adaptation data (asserted as the number of claimant training frames in the speaker model) and target FA rate, and 2) such a function could be ported from one database to another, we could have a simple way of estimating thresholds. This is the main idea of this algorithm, which we call Frame Count Dependent Thresholding, or FCDT.

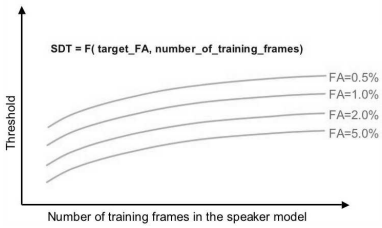

Fig. 1. Our hypothesis is that the threshold could be expressed as a function of two parameters: number of training frames in the speaker model (*NF*) and the Target FA rate (*TFA*).

Figure 1 is a schematic depiction of the hypothesized parameterized model. The first step is to verify this hypothesized relationship between Threshold, *NF*, and *TFA* in a couple of databases. Next, the function should be parameterized on one database and applied to another to see if FA target rates can be achieved both before and after online adaptation. In the sections below, we discuss the verification of this hypothesis (Section 3), calibration of the parametric function (Section 4), and experiments on multiple databases (Section 5).

3. STUDYING THE DATA

3.1. Is Threshold a Function of FA and Frame Count?

For the purposes of calibration, we chose an in-house Japanese digits database, which had enough speaker data for gradual adaptation. A GMM system (described in [10]) was used. 6,477 speaker models were trained on three repetitions of eight-digit passwords. The average duration of the password was 2.5 seconds. The speaker models were adapted, in a supervised fashion, on one speaker utterance at a time and testing was done on the models after every adaptation step using a fixed held out evaluation set. Figure 2 shows how the the score threshold has to be increased in order to keep the FA rate constant as the speaker models are adapted. The trends look promising and the function can easily be parameterized with a second order polynomial of two parameters.

We then attempted to confirm this observation on another database. A similar experiment on a Canadian French Text database was set up. The results showed that the increase in threshold, as adaptation occurs, is not as monotonic and well defined as in Figure 2. This can be best observed by concentrating on the FA=1% curve on Figure 3. were similar, except for the curious appearance of a bi-modal distribution. As we can see in figure 3, there is a plateau at about 300 and again around 1200 frames. We attempted to explain the abberations by bugs in our experiments. The evidence, however, prevailed when another database (this time

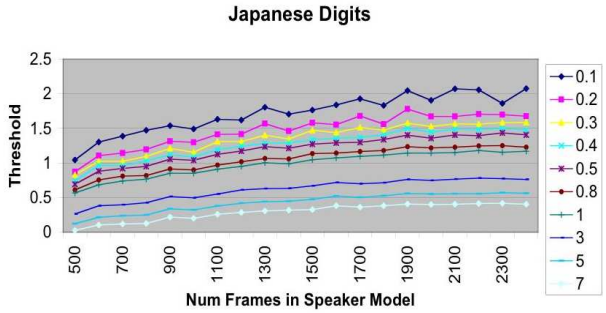

Fig. 2. The graph shows the growth of threshold after online adaptation for various FA rates. As hypothesized, threshold appears to be well predicted by the two hypothesized parameters, namely, number of training frames in the speaker model, *NF* (x-axis) and the Target FA rate (*TFA*) (*%TFA* in legend).

UK English text) showed the same trend. Are these two parameters not sufficient to predict the threshold? If so, what else is the threshold dependent on?

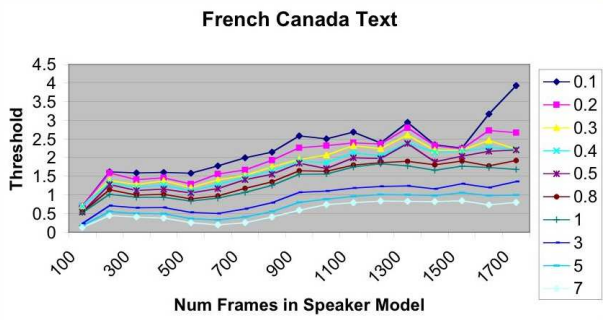

Fig. 3. The Canadian French Text database shows a bi-modal distribution, suggesting that threshold is dependent on (at least) one more parameter.

3.2. The Missing Parameter: Password Length

The answer is simple: it is well known that the raw verification performance (and descriminative power of verification) degrades when the password gets shorter; the implication on our algorithm is that to maintain a fixed FA rate, we have to increase the threshold. As expected, we discovered that the Japanese digits dataset was fairly uniform in terms of password length for all models (8-digit strings), whereas the other two tested text databases both have one group of short and one group of long passwords. This difference in password length appears in Figure 3. At 200 frames on the x-axis, we see the threshold increase for all the short-password models. At around 400 frames, the increase for the short set starts to plateau. At around 900 frames, we see the rising slope for the long-password models. By this point, the short password models have run out of adaptation data, or if still being adapted, the threshold increase has saturated.

4. CALIBRATION

We used the Japanese text and digit database (which have multiple password lengths) for calibrating the parameters of the threshold function. 10,314 speaker models were trained on three repetitions

of either a digit or a text password. Supervised adaptation was performed on the speaker models using at least seven more password tokens. Impostor testing was done after each adaptation step to generate the calibration data. We divided the password length region between 0 and 700 frames into 8 *pass-length bins*, shown as separate lines in Figure 4, such that each bin contained between 120K and 196K impostor trials. The mean password lengths are 1.1 and 2.5 seconds for short and long passwords respectively.

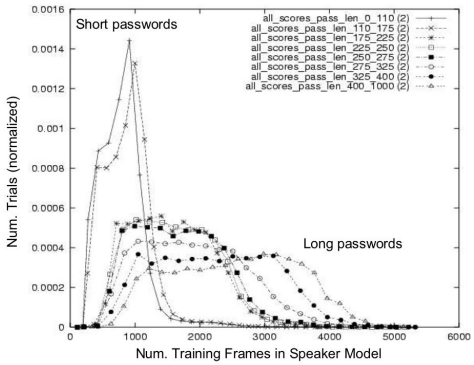

Fig. 4. The figure shows the normalized histogram of number of impostor attempts for each pass-length bin (in frames) in the database.

Note that each pass-length bin contained impostor trial data points tested on a given speaker model after none, one, ..., and *n* adaptation steps. Next, the impostor trials in each pass-length bin were divided into 10 sub-bins, based on the number of training frames in the speaker model (*NF*). For each sub-bin, we calculated the average number of training frames (*ANF*) and the average password length (*APL*). Figure 5 shows the two dimensional space spanned by these two parameters for all the sub-bins. The sub-bins on the right edge seemed questionable as they did not have much wider spans and were excluded. To maximize modeling efficiency, we preferred better coverage of the 2-D space, so we replaced *ANF* with *ANF/APL*. For each sub-bin we calculated the following key parameters: Threshold to achieve target FA (*T*), Target FA (*TFA*), Average Password Length (*APL*), Average Number of Training Frames in the Speaker Model divided by the Average Password Length (*ANF/APL*). We then modeled the threshold by fitting a second order polynomial of three parameters (total of 10 free coefficients) as *T = G(TFA, ANF, ANF/APL)*.

Figure 6 shows the fit for *TFA* of 1% in the three dimensional space. Visual inspection of the fits for all *TFA* data (not shown) indicates a reasonably good fit of the model to the data.

5. EXPERIMENTS

As mentioned in Section 4, the calibration of the parameters was done using a Japanese digits and text database. We then applied the parameterized equation to 12 databases in four languages (American English, UK English, Canadian French, and American Spanish).

As mentioned, one goal of this work was to set thresholds for a desired FA rate without requiring tuning. Considering the level of challenge of this task, our goal was to be within a reasonable target range. The *TFA* parameter was set at 1/4th of the range, specifically, lowbound + 0.25 * (upbound-lowbound). Table 1 shows the FA rates achieved for the goal range of [0.2-1.5%] (aim:

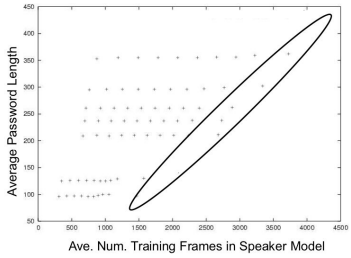

Fig. 5. The figure shows the two dimensional space spanned by *ANF* and *APL* of the sub-bins. The questionable regions on the far right edge of the distributions were excluded. To get better distribution, we chose to use parameter *ANF/APL* instead of *ANF* alone.

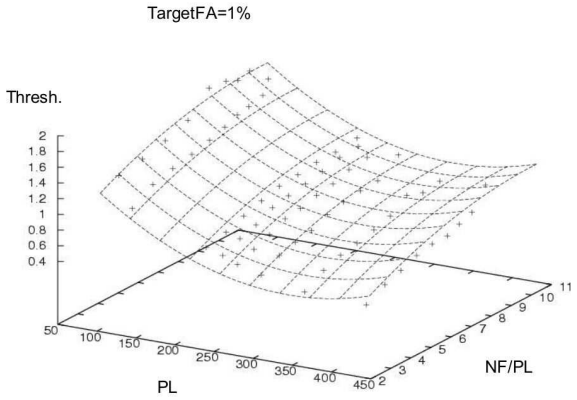

Fig. 6. The plot shows the fit of the second order polynomial to the data set with target-FA = 1%.

0.525%) for the twelve datasets we tested. Two of the databases overshot (American English Text 1 and 2) and two barely undershot (Canadian French Digits, and just barely, American Spanish Digits) the goal. The other eight tested databases were within the target range. These databases covered a wide variety of applications, lexical content, and channel conditions.

Table 1 also compares the EER performance of the baseline system with FCDT. The overall EER for the system can change since the decision threshold (or subtraced bias) for each speaker model is set independently and is different. The goal of this work was to achieve better prediction for FA rates and not to improve EER. For some databases we observe an EER improvement (max: +25%), and for some, a degradation (max: -15%). Overall, the EER effect can be considered a wash, which considering the goals of this work, is satisfactory.

It is desirable to meet the FA targets in all security levels. Table 2 shows the FA rate for various security level ranges for one of the English digit databases (for space concerns, the other eleven datasets are not shown). In all except for the least secure region, the target FA range is met. Even for that region, the achieved FA is reasonable.

Finally, one main motivation for this work was to keep the growth of FA rates after adaptation in check. Figure 7 shows the

DB	TS/IM trials	FA FCDT	EER FCDT	EER Base	Pct. Imprv
EA Dig1	0.8K/0.8K	0.53%	5.79%	7.76%	25%
EA Dig2	38K/38K	0.68%	4.74%	5.20%	9%
EA Dig3	1.2K/25K	0.42%	2.86%	2.54%	-13%
EA Dig4	3K/10K	0.40%	4.03%	3.75%	-7%
EA Txt1	38K/38K	2.16%	7.25%	7.80%	7%
EA Txt2	2K/15K	1.97%	3.15%	2.82%	-12%
EUK Dig	5K/5K	1.13%	1.92%	2.22%	14%
EUK Txt	13K/13K	1.20%	5.61%	5.97%	6%
FR Dig	4K/4K	0.11%	3.52%	3.05%	-15%
FR Txt	27K/27K	0.41%	7.36%	7.22%	-2%
SP Dig	50K/50K	0.17%	3.38%	3.62%	7%
SP Txt	49K/49K	1.08%	4.36%	5.10%	15%

Table 1. Table shows FA and EERs for 12 digits and text databases for FCDT, and EERs for the baseline system. The target FA range was [0.2-1.5%]. TS/IM is the number of true speaker and impostor trials. In the database (DB) column, 'EA' stands for American English, 'EUK' is UK English, 'FR' is Canadian French, and 'SP' is American Spanish.

Security Level	Target Range	Target FA	Actual FA
VERY-HIGH	[0.1-0.2%]	0.125%	0.10%
HIGH	[0.2-1.5%]	0.525%	0.40%
MEDIUM-HIGH	[1.5-3.0%]	1.875%	1.57%
MEDIUM	[3.0-5.0%]	3.5%	3.10%
MEDIUM-LOW	[5.0-7.0%]	5.5%	4.83%

Table 2. Table shows actual FA rates for various target ranges for English Digits 4, with 3K/10K of claimant/impostor trials.

change in FA rate after multiple iterations of **unsupervised** online adaptation for the FCDT and the baseline system. Test data is English digits with 10K/10K of claimant/impostor trials. The adaptation data is uniformly distributed, such that all speaker models have the same probability of being adapted, and the impostor adaptation attempts are roughly 10% of the total. We see that the FA rates of the baseline system more than double, whereas the FCDT algorithm updates the overall threshold successfully to keep the FA rates relatively constant.

For due diligence, we also compared the FCDT algorithm with our previous approach presented in [7]. The current algorithm performes similarly in maintaining a constant FA rate after adaptation, and outperforms the previous one in achieving the target FA rate and not degrading the EER. And finally, the current algorithm is computationally far simpler and more efficient than its previous incarnation.

6. CONCLUSION

In this work we presented a computationally efficient strategy for setting *a priori* thresholds in an adaptive speaker verification system. We had two main motivations: 1) to eliminate the externally pre-set overall system thresholds and replace them with automatically-set internal thresholds calculated at runtime; and 2) to counter the verification score shifts resulting from online adaptation.

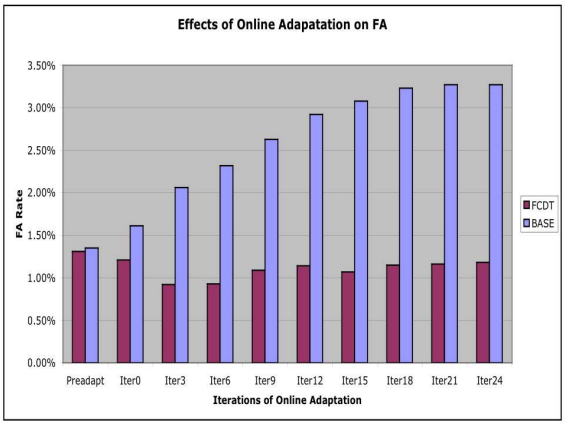

Fig. 7. Plot shows FA rate growth after adaptaion. We see that the FCDT algorithm controls the growth of FA by updating the thresholds effectively. Test data is English digits, with 10K/10K claimant/impostor trials.

We learned that score threshold can be modeled as a function of three parameters: 1) Goal FA, 2) password length, and 3) number of training frames in the speaker model. We estimated the parameters for a second order polynomial (10 free coefficients) on a Japanese database and then successfully applied it to 12 test sets in four languages (American English, UK English, Canadian French, and American Spanish).

Although there was significant difference between the calibration and test material, the estimated threshold function appeared to be portable. Performance in the desired FA rate region was achieved for eight out of the 12 test cases. Considering how challenging it is to set the operating point without application-specific tuning data, the achieved results are considered very satisfactory.

We also demonstrated how this algorithm succeeds in maintaining a constant FA rate after multiple rounds of online adaptation. Finally, the algorithm neither degrades nor improves the EER consistently, as expected.

7. REFERENCES

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

[2] C. Fredouille, J. Mariéthoz, C. Jaboulet, J. Hennebert, J.-F. Bonastre, C. Mokbel, and F. Bimbot. Behavior of a bayesian adaptation method for incremental enrollment in speaker verification. In *ICASSP*, Istanbul, Turkey, 2000.

[3] 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.

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

[5] 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.

[6] Carey M., Auckenthaler R., and Lloyd-Thomas H. Score normalization for text-independent speaker verification systems. In *Digital Signal Processing*, volume 10, pages 42–54, 2000.

[7] N. Mirghafori and L.P. Heck. An adaptive speaker verification system with speaker dependent a priori decision thresholds. In *ICSLP*, Denver, Colorado, 2002.

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

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

[10] R. Teunen, B. Shahshahani, and L.P. Heck. A model-based transformational approach to robust speaker recognition. In *ICSLP*, Bejing, China, 2000.