AFRL-HE-WP-TP-2006-0025

AIR FORCE RESEARCH LABORATORY

Supervised and Unsupervised Speaker Adaptation in the NIST 2005 Speaker Recognition Evaluation

Eric G. Hansen
Raymond E. Slyh
Timothy R. Anderson

Human Effectiveness Directorate
Warfighter Interface Division
Wright-Patterson AFB OH 45433-7022

February 2006

Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
Collaborative Interfaces Branch
Wright-Patterson AFB OH 45433

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
February 2006	Conference Proceedings	

4. TITLE AND SUBTITLE
Supervised and Unsupervised Speaker Adaptation in the NIST 2005 Speaker Recognition Evaluation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Hansen, Eric G., Slyh, Raymond E., Anderson, Timothy R.

5d. PROJECT NUMBER
7184

5e. TASK NUMBER
08

5f. WORK UNIT NUMBER
71

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
***Air Force Materiel Command
Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
Collaborative Interfaces Branch
Wright-Patterson AFB OH 45433-7022

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/HECP

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-HE-WP-TP-2006-0025

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited. Cleared by AFRL/PA and AFMC/PAX as AFMC-06-069 and AFRL/WS-06-0431 on 17 February 2006.

13. SUPPLEMENTARY NOTES
This will be published in the Proceedings of Odyssey 2006: The Speaker and Language Recognition Workshop.

14. ABSTRACT
Starting in 2004, the annual NIST Speaker Recognition Evaluation (SRE) has added an optional unsupervised speaker adaptation track where test files are processed sequentially and one may update the target model. In this paper, various model adaptation techniques are implemented using a supervised (ideal) adaptation scheme. Once the best performing model adaptation is found, unsupervised adaptation experiments are run using a threshold to determine when to update the target model. Three NIST training conditions, 10sec4w, 1conv4w, and 8conv4w, all with the 1conv4w test condition are used for experiments with the NIST 2005 SRE. MinDCP values for the three training conditions are reduced from 0.0708 to 0.0277 for 10sec4w, from 0.0385 to 0.0199 for 1conv4w, and from 0.0264 to 0.0176 for 8conv4w using the supervised adaptation compared to the baseline. For the unsupervised adaptation compared to the baseline. For the unsupervised adaptation, minDCF values were reduced to 0.0590, 0.0302, and 0.0210 for the respective conditions.

15. SUBJECT TERMS
Speaker recognition, speaker model

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
Unclassified					Raymond Slyh
a. REPORT	b. ABSTRACT	c. THIS PAGE	SAR	10	19b. TELEPHONE NUMBER *(include area code)*
UNC	UNC	UNC			

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

Supervised and Unsupervised Speaker Adaptation in the NIST 2005 Speaker Recognition Evaluation

Eric G. Hansen, Raymond E. Slyh, Timothy R. Anderson

Air Force Research Laboratory, Human Effectiveness Directorate, Wright-Patterson AFB OH, USA

Abstract

Starting in 2004, the annual NIST Speaker Recognition Evaluation (SRE) has added an optional unsupervised speaker adaptation track where test files are processed sequentially and one may update the target model. In this paper, various model adaptation techniques are implemented using a supervised (ideal) adaptation scheme. Once the best performing model adaptation method is found, unsupervised adaptation experiments are run using a threshold to determine when to update the target model. Three NIST training conditions, 10sec4w, 1conv4w, and 8conv4w, all with the 1conv4w test condition are used for experiments with the NIST 2005 SRE. MinDCF values for the three training conditions are reduced from 0.0708 to 0.0277 for 10sec4w, from 0.0385 to 0.0199 for 1conv4w, and from 0.0264 to 0.0176 for 8conv4w using the supervised adaptation compared to the baseline. For the unsupervised adaptation, minDCF values were reduced to 0.0590, 0.0302, and 0.0210 for the respective training conditions.

1. Introduction

It is well known that the performance of speaker recognition systems tends to improve when larger amounts of training data are used to build speaker models. On the other hand, obtaining these larger amounts of training data can be problematic for many applications. In applications where users enroll in the system (such as in telephone banking), the users generally do not like having to give large amounts of training data. In applications where someone (other than a user) has to select some data to build a speaker model, finding appropriate segments to use as additional training data can be a time-consuming, tedious process, especially for speakers that do not appear very often. For these reasons and others having to do with robustness to intersession variability and voice aging [1,2], it is appealing to be able to automatically update/adapt speaker models as more training data are obtained, and it is especially desirable to be able to automatically obtain the additional training data with no additional training burden on users and no time-consuming, manual search by system developers.

One way that has been proposed for obtaining additional training data for speaker models is to use data from on-line test cases that score highly against the respective claimant models [1–8]. This unsupervised adaptation procedure has received considerable attention for text-dependent applications [1–6] and for applications in which the number of impostor trials is considerably lower than the number of true claimant trials [4–6].

Two notable studies [7, 8] have considered scenarios derived from NIST SRE[1] databases, which involve text-independent verification with a large ratio (approximately 10:1) of impostor tests to true claimant tests. In both [7,8], the NIST 2002 SRE database was used to synthetically create the adaptation testing paradigm, and both efforts showed a benefit from using unsupervised adaptation. Since 2004, NIST has provided test control files to allow for the possibility of running systems in an unsupervised adaptation mode in the annual SRE without the need for synthetically-generated test control files. In addition to using the 2002 database, [8] also used the 2004 database, but no improvement was found from using unsupervised adaptation with the 2004 data. One possible explanation for the disparity between the use of the 2002 and 2004 databases is that the 2002 (limited-data, one-speaker detection) database used only cellular telephone data and involved only English speech, whereas the 2004 database used data from both cellular and landline telephones and involved some non-English speech. As the work in [8] used the 2002 data for setting some of the parameters of the system, this might have contributed to the lack of improvement seen with the 2004 data.

The work reported here is similar in spirit to the work of [7, 8] in that we consider unsupervised adaptation in the NIST SRE context; however, we consider the NIST 2005 SRE database for testing and use the 2004 database for setting some of the system parameters. The 2005 database is similar to the 2004 database with a mix of telephone handset and channel types and some non-English speech, although the amount of non-English speech appears to be less in the 2005 database than in the 2004 database.

Opinions, interpretations, and conclusions are those of the authors and are not necessarily endorsed by the United States Air Force.

[1] See: http://www.nist.gov/speech/tests/spk/

With an on-line, unsupervised method of model adaptation, two important issues to address are determining when and how to update models. The work reported here focuses on the latter question of how to update models. In particular, we focus on various techniques for adapting speaker models in a standard speaker recognition system using mel-frequency cepstral coefficients (MFCCs) and Gaussian mixture models (GMMs) [13]. Previous work in [7, 8] used a standard adaptation method for GMMs involving trying a number of different settings for a relevance factor; however, we show that setting the adaptation factor based on the ratio of the number of speech frames in the test file to the sum of the number of speech-frames in the test file and the number used to build the current model, and setting a floor and ceiling on this ratio, yields more performance improvement on the three training conditions investigated than adapting based on a relevance factor. Without the floor and ceiling operations, the proposed adaptation method would be similar to that used in [3].

With regard to the problem of determining when to update a model, much of the previous work has used a speaker-independent threshold, which does not have to be the same as that used for the decision threshold (*e.g.*, see [4,5,7,8]). Our work also uses a speaker-independent threshold for determining when to adapt a model, and we set it equal to the decision threshold that we determined from the NIST 2004 database. Notable work that has considered other methods besides speaker-independent thresholds includes [6, 7, 10].

This paper is organized as follows. The next section discusses the experimental setup, including the NIST 2005 SRE database and the performance measures to be considered. Section 3 describes the baseline system and the various model adaptation methods investigated. Section 4 discusses the experimental results of using the different adaptation methods in supervised adaptation and of applying the best model adaptation scheme for the unsupervised adaptation task. Finally, Section 5 presents the discussion and conclusions.

2. Experimental Setup

2.1. Database

All experiments were run on the NIST 2005 SRE [11] database using conversational speech collected over telephone channels. Under the NIST 2005 SRE framework, five training conditions and four testing conditions were outlined resulting in 20 different conditions to evaluate. Experiments run for this paper focused on three of these conditions which were the: 10-second training (10sec4w), one-conversation training (1conv4w), and eight-conversation training (8conv4w) conditions, with the testing condition held constant at one-conversation (1conv4w). The "4w" designates the four-wire case, where both sides of the conversation are given as separate channels in the speech files, but only one side (*i.e.*, channel) is used for the training or the testing. The length of time for each condition is the approximate amount of speech given to either build an initial model or to evaluate a model. The 1conv4w and 8conv4w conditions use one and eight conversation sides, respectively, where a conversation is five minutes long and should result in nominally 2–2.5 minutes of speech for each speaker assuming roughly equal turn taking. However, we have found that a number of conversation sides yield significantly less than two minutes of speech; hence, our proposed adaptation method is based on numbers of speech frames as opposed to nominal amounts of speech as was considered in [1]. For these adaptation experiments, an index file is supplied by NIST which is processed sequentially, and for each model-test file pair, the model could be updated with the test file for future trials.

2.2. Performance Measures

NIST compares system performance in two major ways. First, NIST uses a detection cost function, C_D, defined as a weighted sum of miss and false alarm probabilities:

$$C_D = C_M P_{M|T} P_T + C_{FA} P_{FA|NT}(1 - P_T),$$

where C_M is the cost of a miss (chosen by NIST as 10), C_{FA} is the cost of a false alarm (chosen by NIST as 1), P_T is the *a priori* probability of a target (chosen by NIST as 0.01), $P_{M|T}$ is the probability of a miss given a target trial, and $P_{FA|NT}$ is the probability of a false alarm given a non-target trial. $P_{M|T}$ and $P_{FA|NT}$ are a function of system performance and the chosen detection threshold. For a given system, chosen costs, and *a priori* target probability, there is a threshold that yields a minimum value of C_D; we refer to this minimum value of C_D as the minDCF value. Second, NIST uses plots of $P_{M|T}$ versus $P_{FA|NT}$, called Detection Error Trade-off (DET) plots [12], to show how system performance varies for a wide range of operating points. In addition to these two presentations of performance, we will also use the equal error rate (EER), the value of $P_{M|T}$ (or $P_{FA|NT}$) when $P_{M|T} = P_{FA|NT}$.

3. Baseline System and Adaptation Methods

This section describes the MFCC/GMM system and discusses four methods for adapting the target models with test data.

3.1. The Baseline System

The baseline system was based on Version 2.1 of the MIT Lincoln Laboratory (MIT-LL) MFCC/GMM system [13] with $M = 2048$ mixtures per model and diagonal covariance matrices for each mixture. Nineteen MFCCs

were computed in the bandwidth of 300–3138 Hz every 10 msec. RASTA filtering [14] was applied to the MFCCs and deltas were then calculated. Only frames labeled as speech by a speech activity detector were used, and these were further processed with the feature mapping method of [15] for channel compensation. The mapped features were then normalized to have zero mean and unit variance. For a test file, the set of mapped and normalized features, F_T, were scored against an hypothesized speaker's GMM, λ_S, and a background GMM, λ_{BKG}, using the "top-5" scoring method as outlined in [13], and the system log-likelihood score was formed as

$$\Lambda(F_T, \lambda_S) = \log(p(F_T|\lambda_S)) - \log(p(F_T|\lambda_{BKG})).$$

The system used a speech activity detector which worked in three stages. The first stage utilized a two-state speech/non-speech Hidden Markov Model (HMM) with MFCCs as the features. The second stage refined the HMM output by applying an energy-based detector. The final stage post-processed the output by reclassifying as non-speech any segments labeled as speech that were less than 20 msec in duration. The MFCC/HMM portion of the SAD was built using HTK from Cambridge University[2] using 64 mixtures per state. The energy-based detection was performed using the MIT-LL *xtalk* program from their MFCC/GMM speaker recognition system.

Gender-dependent T-norm [16] was applied to the system log-likelihood scores (using 120 models for each gender), with the exception that gender-independent T-norm (with 240 models) was used in the 10sec4w training conditions. For the 10sec4w-10sec4w training/testing condition, T-norm models were built from 30 seconds of data. For the other training conditions, T-norm models were built using approximately two minutes of data.

The background model data consisted of approximately 16 hours of speech from a variety of sources, including the NIST 2001–2003 evaluations (for carbon button land line data, electret microphone land line data, and digital cellular data) and the OGI National Cellular Database[3] (for analog cellular data). The background model data were balanced for gender and the four previously mentioned channel types, and these channels were the ones used in the feature mapping. The T-norm model data came from NIST 2001–2003 evaluation data.

Initial target and T-norm models were built by adapting from the background model using MAP adaptation to adapt only the mixture means from those of the background model [13]. Let $X = \{\bar{x}_1, \ldots, \bar{x}_T\}$ be a set of T feature vectors to be used in adapting a model, then the probabilistic alignment of the t^{th} vector into the m^{th} prior mixture component is given by

$$\Pr(m|\bar{x}_t) = \frac{w_m b_m(\bar{x}_t)}{\sum_{j=1}^{M} w_j b_j(\bar{x}_t)}, \quad (1)$$

where w_m is the mixture weight for the m^{th} mixture and $b_m(\bar{x}_t)$ is the m^{th} uni-modal Gaussian mixture evaluated at $\bar{x}_t$. Let $n_m(X)$ be the probabilistic count of the vectors of X aligned with the m^{th} prior mixture, then

$$n_m(X) = \sum_{t=1}^{T} \Pr(m|\bar{x}_t). \quad (2)$$

Define $E_m(X)$ as

$$E_m(X) = \frac{1}{n_m(X)} \sum_{t=1}^{T} \Pr(m|\bar{x}_t)\bar{x}_t, \quad (3)$$

then the adapted mean of the m^{th} mixture, $\bar{\mu}'_m$, is given as

$$\bar{\mu}'_m = \alpha_m E_m(X) + (1 - \alpha_m)\bar{\mu}_m, \quad (4)$$

where $\bar{\mu}_m$ is the prior mean of the m^{th} mixture and α_m is the adaptation factor for the m^{th} mixture. In adapting T-norm and initial target models from the background model, α_m is typically defined as

$$\alpha_m = \frac{n_m(X)}{n_m(X) + r}, \quad (5)$$

where r is known as the relevance factor. Following the work of [13], we set $r = 16$.

3.2. Adaptation Methods

This subsection describes four methods that were investigated for on-line adaptation of target models in the experiments to be discussed in Section 4. In all cases, only mixture means were adapted using the same procedure as outlined in Equations 1–4 for adaptation from the background model. The difference was in the form of the adaptation factor, α_m.

3.2.1. Adaptation Factor 1

The first adaptation factor considered was the same one used to adapt the initial target models from the background model—namely, Equation 5. The relevance factor was chosen to be fixed at 16 as was done in adapting initial models from the background model. Setting a fixed relevance factor follows the adaptation process used in [7, 8]. Note that this adaptation factor only depends on the relevance factor and the probabilistic count of the vectors aligned with each mixture. Thus, this adaptation factor can rapidly shift a mixture's mean even if a large number of vectors have gone into determining the mixture's prior mean.

3.2.2. Adaptation Factor 2

Let the number of speech frames in a test file to be used for adaptation be T_T and let the total number of speech frames that have gone into making a model be T_M. The

[2] Available at: http://htk.eng.cam.ac.uk/

[3] See: http://cslu.cse.ogi.edu/corpora/corpCurrent.html

second adaptation factor considered was to set the adaptation factor for all of the mixtures to be: $\alpha = T_T/T_M$, where if $\alpha > 1$, then α is set to 0.75. This method works well for training conditions that start with large amounts of training data, thereby resulting in a smaller value of α and preserving the history. When the amount of data initially used to train a model is small and the amount of new data is large, α swings toward its upward bound and heavily weights the new data. Such a large shift in the means of the model can shift the model away from the background model and create problems for the fast "top-5" scoring method.

3.2.3. Adaptation Factor 3

Let T_T and T_M be defined as for the second adaptation factor, then the third adaptation factor considered was to set the adaptation factor for all of the mixtures to be: $\alpha = T_T/(T_T + T_M)$. This ratio restricts α to be $0 \leq \alpha \leq 1$, but it can still result in values of α close to one if the amount of data used to build the initial model is small compared to the amount of new adaptation data. This adaptation factor has the same problems as mentioned in the discussion of the prior adaptation factor. Performance is good for conditions where the amount of new data is less than or equal to the amount of data used to build the initial model.

3.2.4. Adaptation Factor 4

The final adaptation factor considered is the same as the prior adaptation factor, but with two constraints: a ceiling of 0.5 and a floor of 0.1. This means that the new adaptation data will contribute at most half of the information in shifting the new mixture means and will contribute at least 1/10 of the information. When the number of speech frames used to build the model is small compared to the number of speech frames from the test file, this factor is almost always set to the ceiling of 0.5. On the other hand, even if the number of speech frames used in building a model is very large relative to the number of speech frames in the test file, there will still be some adaptation of the model. The next section shows that this addition of the floor and ceiling constraints significantly improves the performance.

4. Experimental Results

This section presents the experimental results for both supervised and unsupervised adaptation. The supervised adaptation results used the key for the NIST 2005 SRE to determine when to update the models. The unsupervised adaptation results used detection thresholds found from the NIST 2004 SRE database to determine when to update the models.

4.1. Supervised Adaptation

Figure 1 shows the results of using the four adaptation factors with supervised adaptation for the 10sec4w training condition. Adaptation with factors 2 and 3 outperforms the baseline system in the low false alarm region, but shows considerably worse performance in the high false alarm region. Adaptation with factors 1 and 4 performs better than the baseline system for false alarm rates up to a little less than 40%. The best performance came from adaptation factor 4 which resulted in a minDCF of 0.02772 compared to the baseline minDCF of 0.07079 and an EER of 13.4% compared to the baseline EER of 20.8%. This shows the large amount of improvement that might be gained if proper adaptation occurs.

Figure 2 shows the results of using the four adaptation factors with supervised adaptation for the 1conv4w training condition. Adaptation factor 2 performs better than the baseline for false alarm rates below 1%, while adaptation factor 1 outperforms the baseline for false alarm rates below 7%. Adaptation factors 3 and 4 show a performance increase across the entire visible region of the DET plot, with adaptation factor 4 performing the best. The minDCF of using adaptation factor 4 is 0.0199 compared to the baseline minDCF of 0.0385, while the EER is reduced to 6.9% from the baseline EER of 10.7%.

Figure 3 shows the results of using the four adaptation factors with supervised adaptation for the 8conv4w training condition. Adaptation factor 1 performs worse than the baseline for a wide range of false alarm rates. Adaptation factors 2, 3, and 4 all yield similar performance, significantly outperforming the baseline system over the entire visible region of the DET plot. The best performing method is adaptation factor 4, which reduces the minDCF from the baseline of 0.0264 to 0.0176 and reduces the EER from the baseline of 7.3% to 4.8%. The constraint of the floor of 0.1 contributes to the extra gain of adaptation 4 compared to the other methods by making sure a minimal amount of adaptation occurs.

4.2. Unsupervised Adaptation

Based on the fact that adaptation factor 4 performed the best for all of the supervised adaptation experiments, only this adaptation factor was used for unsupervised adaptation. The threshold used to determine when to adapt a model was chosen as the threshold for minDCF operation for the various conditions as found from the NIST 2004 SRE. For comparison purposes, Table 1 lists these thresholds per training condition and per year. One can see that the thresholds are similar between the 2004 and 2005 data for the 10sec4w and 1conv4w training conditions, but for the 8conv4w training condition, the threshold is much lower for the 2005 data. Based on these values, we might expect to miss correct model updates for the 8conv4w training condition by using the 2004 detec-

Training Condition	2004	2005
10sec4w	2.69	2.73
1conv4w	2.78	2.64
8conv4w	5.76	4.64

Table 1: *Threshold for minDCF operation for NIST 2004 and 2005 by training condition. The testing condition was 1conv4w.*

tion threshold as the threshold for determining when to update a model in unsupervised adaptation.

Figures 4–6 compare the baseline system performance to the performance obtained with adaptation factor 4 for both supervised and unsupervised adaptation for the three different training conditions. For all three training conditions, the unsupervised adaptation improves the performance in the low false alarm regions.

For the 10sec4w training condition shown in Figure 4, one can clearly see the potential of what the adaptation might provide if done properly. However, starting with 10 seconds of data to build initial models makes it difficult to find the true matches during unsupervised adaptation. One other significant problem for this training condition is using a fixed threshold for determining when to update a model. As can be seen in Table 1, as the training data increases, so generally does the minDCF threshold. In fact, it has been noted in [6, 9, 10] that as models are adapted, there tend to be shifts in both impostor and true claimant scores. A dynamic, speaker-dependent threshold that takes into account the amount of data in the model being evaluated may be useful [6, 9, 10].

Figure 5 shows the results for the 1conv4w training condition. The unsupervised adaptation never performs worse than the the baseline and provides good improvement over the baseline for false alarm rates less than 5%. Compared to the supervised adaptation, there is still room for improvement.

Figure 6 shows the results for the 8conv4w training condition. Starting with eight conversations of training data, fewer mistakes are made when adapting models with testing data, so the unsupervised adaptation system more closely approximates that of the supervised adaptation performance.

Tables 2 and 3 list the minDCF and EER, respectively, for each training condition and for the baseline, unsupervised, and supervised adaptation experiments. In all cases, whether supervised or unsupervised adaptation, there is improvement in the minDCF area compared to the baseline system. With respect to the EER, only the 10sec4w training condition fails to give an improvement using unsupervised adaptation.

Table 4 lists statistics on model adaptation for the different training conditions. Note the large number of falsely updated (*i.e.,* impostor-corrupted) models for the 10sec4w training condition and the large number of missed model updates across all the training conditions. As the length of the initial training data goes up, the false alarm and miss rates on model adaptation decrease; however, for any given condition, further improvement in the unsupervised adaptation performance might be achieved by better adjusting the adaptation decision threshold.

Training Condition	Baseline	Adapted	
		Unsupervised	Supervised
10sec4w	0.0708	0.0590	0.0277
1conv4w	0.0385	0.0302	0.0199
8conv4w	0.0264	0.0210	0.0176

Table 2: *MinDCF for the baseline, unsupervised adaptation, and supervised adaptation systems by training condition. All testing conditions were 1conv4w.*

Training Condition	Baseline	Adapted	
		Unsupervised	Supervised
10sec4w	20.75%	26.66%	13.38%
1conv4w	10.73%	10.51%	6.85%
8conv4w	7.32%	5.84%	4.80%

Table 3: *Equal error rates for the baseline, unsupervised adaptation, and supervised adaptation systems by training condition. All testing conditions were 1conv4w.*

5. Discussion and Conclusions

In this paper, various model adaptation techniques were implemented on the NIST 2005 SRE using a supervised adaptation scheme. Setting the adaptation factor based on the ratio of the number of speech frames in the test file to the sum of the number of speech frames in the test file and the number used to build the current model, and setting a floor and ceiling on this ratio, yielded more performance improvement on the three training conditions investigated than adapting based on a fixed relevance factor. MinDCF values for the three training conditions are reduced from 0.0708 to 0.0277 for 10sec4w, from 0.0385 to 0.0199 for 1conv4w, and from 0.0264 to 0.0176 for 8conv4w training conditions using the supervised adaptation compared to the baseline. Using the minDCF threshold derived from the NIST 2004 SRE, unsupervised adaptation was executed using this adaptation method and minDCF values were reduced to 0.0590 for 10sec4w, 0.0302 for 1conv4w, and 0.0210 for 8conv4w training conditions.

A trend seen in previous research, [5–7, 9] is a shift in scores as models are adapted. These score shifts can negatively affect performance as they increase the chance of model corruption as updating continues. Various score normalization schemes have been investigated to combat this problem [10], but with mixed results. While score normalization was not the focus of this paper, results from the 10sec4w training condition further illus-

	10sec4w	1conv4w	8conv4w
Total trials	31,398	31,315	23,630
Correct updates	1,371	2,055	1,793
Correct rejections	28,329	28,388	21,340
False alarms	244	152	49
Misses	1,454	720	448
Num. target models	642	635	497

Table 4: *Model adaptation information per training condition for the unsupervised adaptation experiments.*

trate that score normalization would be useful. The most gain to be realized with adaptation is in the short training condition of 10sec4w; however this is precisely the case when it is more difficult initially to determine when to properly update the model. A very conservative threshold may need to be set initially to control for false alarms in model adaptation.

Future work will investigate when to update the target models with the test data as there is still a performance gap between supervised and unsupervised results. This work will include investigating dynamic thresholds or score normalization schemes to combat the score shifts that occur when updating the models. Also of interest is to investigate adjustments to the adaptation factor, such as in [4, 5], which more aggressively update a model when a test file scores very highly against it.

6. References

[1] W. Mistretta and K. Farrell, "Model adaptation methods for speaker verification," in *Proc. of ICASSP*, (Seattle WA), May 1998.

[2] K. Farrell, "Speaker verification with data fusion and model adaptation," in *Proc. of ICSLP*, (Denver CO), September 2002.

[3] C. Fredouille, *et al.*, "Behavior of a Bayesian adaptation method for incremental enrollment in speaker verification," in *Proc. of ICASSP*, (Istanbul, Turkey), June 2000.

[4] L. Heck and N. Mirghafori, "On-line unsupervised adaptation in speaker verification," in *Proc. of ICSLP*, (Beijing, China), October 2000.

[5] L. Heck and N. Mirghafori, "On-line unsupervised adaptation in speaker verification: Confidence-based updates and improved parameter estimation," in *Proc. of Adaptation in Speech Recognition*, (Sophia Antipolis, France), August 2001.

[6] N. Mirghafori and L. Heck, "An adaptive speaker verification system with speaker dependent a priori decision thresholds," in *Proc. of ICSLP 2002*, (Denver CO), September 2002.

[7] C. Barras, S. Meignier, and J. Gauvain, "Unsupervised online adaptation for speaker verification over the telephone," in *Proc. of Speaker Odyssey 2004: The Speaker and Language Recognition Workshop*, (Toledo, Spain), May–June 2004.

[8] D. A. van Leeuwen, "Speaker adaptation in the NIST speaker recognition evaluation 2004," in *Proc. of Interspeech 2005*, (Lisbon, Portugal), September 2005.

[9] A. Sankar and A. Kannon, "Automatic confidence score mapping for adapted speech recognition systems," in *Proc. of ICASSP 2002*, (Orlando FL), May 2002.

[10] N. Mirghafori and M. Hébert, "Parameterization of the score threshold for a text-dependent adaptive speaker verification system," in *Proc. of ICASSP 2004*, (Montreal, Canada), May 2004.

[11] NIST, *The NIST Year 2005 Speaker Recognition Evaluation Plan,* Version 6, 29 March 2005. (Available at: http://www.nist.gov/speech/tests/spk/2005/sre-05_evalplan-v6.pdf)

[12] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. of EuroSpeech '97,* (Rhodes, Greece), September 1997.

[13] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, nos. 1–3, pp. 19–41, 2000.

[14] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 578–589, October 1994.

[15] D. Reynolds, "Channel robust speaker verification via feature mapping," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),* (Hong Kong), April 2003.

[16] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing,* vol. 10, nos. 1–3, pp. 42–54, 2000.

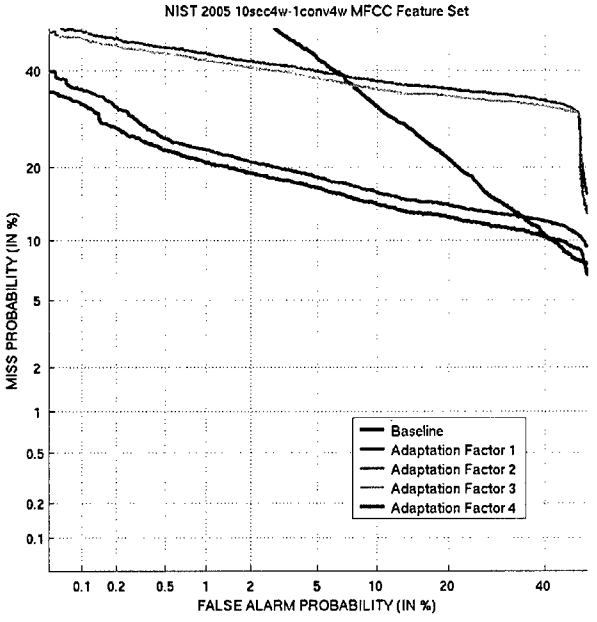

Figure 1: *NIST 2005 SRE 10sec4w training condition, 1conv4w testing condition, comparing four adaptation methods on supervised adaptation experiments. Adaptation factors 2 and 3 perform much worse that adaptation factors 1 and 4, with adaptation 4 performing the best.*

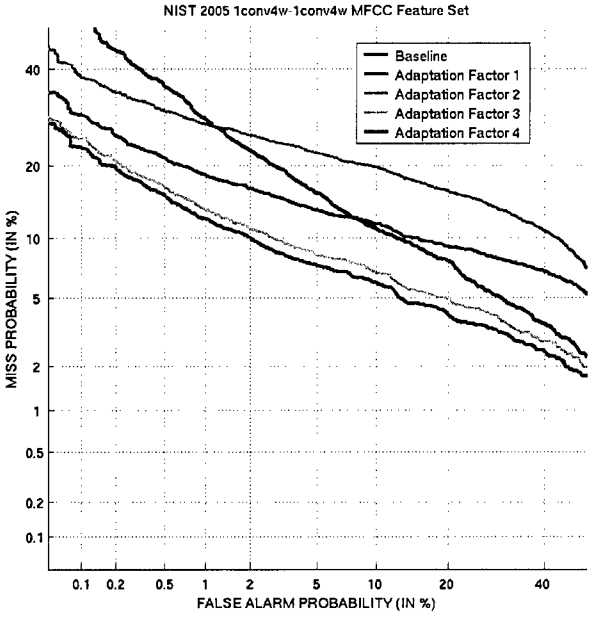

Figure 2: *NIST 2005 SRE 1conv4w training condition, 1conv4w testing condition, comparing four adaptation methods on supervised adaptation experiments. In increasing order of factor performance: 2, 1, 3, and 4.*

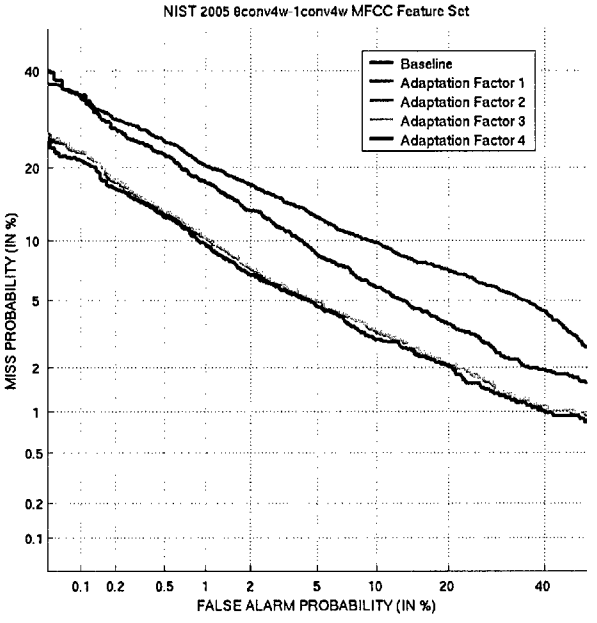

Figure 3: *NIST 2005 SRE 8conv4w training condition, 1conv4w testing condition, comparing four adaptation methods on supervised adaptation experiments. Factor 1 performed worse than the baseline, while the others resulted in similar performance gains over the baseline.*

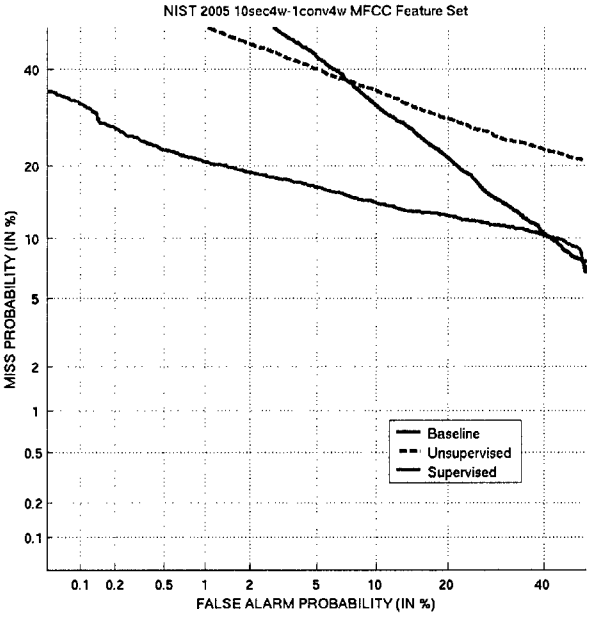

Figure 4: *NIST 2005 SRE 10sec4w training condition, 1conv4w testing condition, comparing the baseline system to supervised and unsupervised adaptation showing there is much to be gained if adaptation is done properly.*

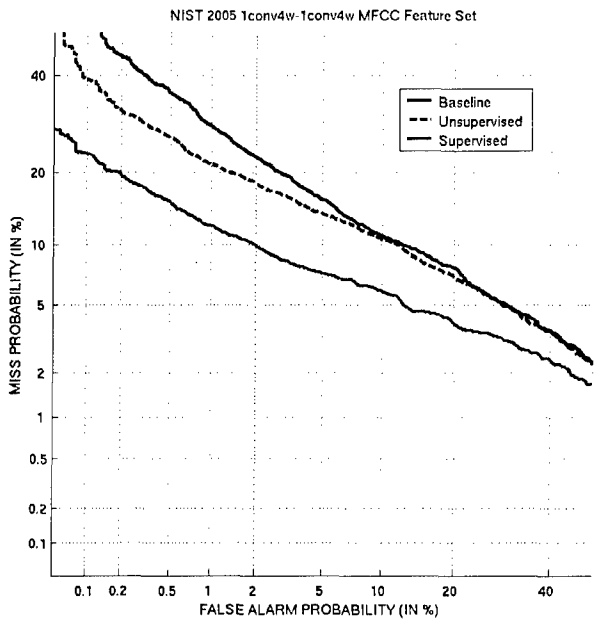

Figure 5: *NIST 2005 SRE 1conv4w training condition, 1conv4w testing condition, comparing the baseline system to supervised and unsupervised adaptation.*

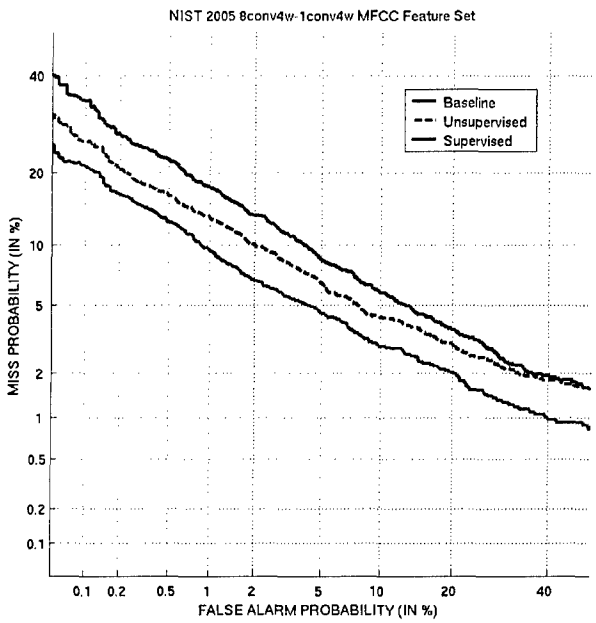

Figure 6: *NIST 2005 SRE 8conv4w training condition, 1conv4w testing condition, comparing the baseline system to supervised and unsupervised adaptation.*