

SCORE CALIBRATING FOR SPEAKER RECOGNITION BASED ON SUPPORT VECTOR MACHINES AND GAUSSIAN MIXTURE MODELS

Marcel Katz, Martin Schafföner, Sven E. Krüger, Andreas Wendemuth
IESK-Cognitive Systems
University of Magdeburg, Germany
email: marcel.katz@e-technik.uni-magdeburg.de

ABSTRACT

In this paper we investigate three approaches of calibrating and fusing output scores for speaker verification. Today's speaker recognition systems often consist of several subsystems that use different generative and discriminative classifiers. If subsystems like Gaussian Mixture Models (GMMs) and Support Vector Machines (SVMs) are used to obtain a final score for decision a probabilistic calibration of single classifier scores is important. Experiments on the NIST 2006 evaluation dataset show a performance improvement compared to the single subsystems and the standard un-calibrated fusion methods.

KEY WORDS

score calibration, score fusion, speaker recognition, support vector machine

1 Introduction

State of the art text-independent speaker verification systems are based on modeling acoustic features by Gaussian Mixture Models (GMMs) [1]. Most systems use the Universal Background Model (UBM) approach with specific speaker models adapted from this background model [2]. During the last years several discriminative kernel classifiers like the Support Vector Machine (SVM) have shown a good performance on different classification tasks. Also, in speaker recognition there have been several approaches of integrating SVMs into standard GMM systems. Especially the supervector concept, proposed in [3], shows good performance on different speaker recognition tasks.

The fusion of different verification systems often results in a performance increase, especially if different levels of speaker characteristics, like acoustic and phonotactic features are modeled [4]. In this paper the subsystems are based on different feature extraction methods that are common in the field of speech and speaker recognition. However, if we want to combine or to fuse the outputs of the single systems it is important that the scores are distributed in the same range [5]. This can be realized by one of the three presented calibration methods for discriminative and generative classifiers, namely a GMM fitting of the scores, the sigmoid fitting proposed by Platt [6] and the isotonic regression [7].

The paper is organized as follows: We first describe

the standard methods in speaker recognition in section 2 and give a short overview of kernel-based methods and SVMs in section 3. Section 4 explains the calibration methods for unmoderated output scores. In section 6 we present the experimental results and finally we conclude this paper with a short discussion in section 7.

2 GMM Based Speaker Verification

State of the art speaker verification systems are based on Gaussian Mixture Models. A GMM is the weighted sum of M Gaussian densities given by:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^M c_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (1)$$

where c_i is the weight of the i 'th component and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is a multivariate Gaussian with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$.

In the UBM approach the mixture model is trained on a large amount of background data by standard methods like the Expectation Maximization (EM) algorithm. For each client of the system a speaker-dependent GMM is then derived from the background model by adapting the parameters of the UBM using a Maximum A-Posteriori (MAP) approach [2]. The decision of detecting a client is usually based on the ratio between the summed log-likelihoods of the specific speaker models and the background model. Defining the probability $P(\mathbf{X}|\lambda_k)$ as the probability of client C_k producing the sentence \mathbf{X} and $P(\mathbf{X}|\Omega)$ as the probability of the background model, the client is detected or accepted if the ratio is above a speaker-independent threshold δ :

$$\log \frac{P(\mathbf{X}|\lambda_k)}{P(\mathbf{X}|\Omega)} > \delta. \quad (2)$$

This results in two possible detection error probabilities: $P_{Miss|Target}$, the speaker is the claimed client but the resulting likelihood-ratio of equation (2) is lower than the threshold.

$P_{FalseAlarm|NonTarget}$, the speaker is not the claimed one but the likelihood-ratio is higher than δ and the speaker is detected.

3 Support Vector Machines

We now give a short overview of SVMs and refer to [8] for more details and further references. Support Vector Machines (SVM) were first introduced by Vapnik, derived from the theory of Structural Risk Minimization (SRM) [9]. SVMs are linear classifiers that can be generalized to non-linear classification by the so-called kernel trick. Instead of applying the linear methods directly to the input space \mathbb{R}^d , they are applied to a higher dimensional feature space \mathcal{F} which is nonlinearly related to the input space via the mapping $\Phi : \mathbb{R}^d \rightarrow \mathcal{F}$. Instead of computing the dot-product in \mathcal{F} explicitly, a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ satisfying Mercer's conditions is used to compute the dot-product. Possible kernel functions are the linear

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^t \mathbf{x}_j \quad (3)$$

and the Gaussian radial basis function (RBF) kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right). \quad (4)$$

Assume that we have a training set of input samples $\mathbf{x} \in \mathbb{R}^d$ and corresponding targets $y \in \{1, -1\}$. The SVM tries to find an optimal separating hyperplane in \mathcal{F} by solving the quadratic programming problem:

$$W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \quad (5)$$

under the constraints $\sum_{i=1}^N \alpha_i y_i = 0$ and $0 < \alpha_i < C \forall i$. The parameter C allows us to specify how strictly we want the classifier to fit to the training data.

The output of the SVM is a distance measure between the test pattern and the decision boundary:

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b \quad (6)$$

where the pattern \mathbf{x} is assigned to the positive class if the sign of $f(\mathbf{x})$ is positive.

4 Score Calibration Methods

In this section we describe the three calibration methods for mapping SVM outputs to posterior probabilities. As already mentioned in the last section, the output $f(\mathbf{x})$ of the SVM is a distance to the decision boundary and is located in large range. To interpret this distance as a posterior probability, the SVM output has to be transformed into the $[0, 1]$ interval by a rescaling method.

In the first calibration method a Gaussian Mixture Model is used to calibrate the SVM output. The GMM is fitted to the SVM output $f(x)$ by a maximum likelihood estimation of the GMM parameters given labeled data samples.

The second parametric approach proposed by Platt [6] transforms the SVM outputs to posterior probabilities by a sigmoid function. This is the most popular method of transforming unmoderated SVM outputs to posterior probabilities. For the posterior class probability we have to model the distribution $P(y = +1|f(x))$ of the positive class, given the unmoderated SVM output $f(x)$. This probability is assumed to be a sigmoid function

$$P(y = +1|\mathbf{x}) = \frac{1}{1 + \exp(Af(x) + B)} \quad (7)$$

where the parameters A and B are computed by a maximum likelihood estimation.

A more general calibration of the SVM outputs is given by the isotonic regression. This method is just restricted on the assumption that the calibration function is an isotonic function dependent on the SVM output $f(x)$. An isotonic function implies a strict increase of function values. The isotonic regression is computed by the pair-adjacent violators (PAV) algorithm [10] that finds a stepwise constant solution according to the mean-squared error criterion.

Assuming training scores \mathbf{x}_i and corresponding targets $y_i \in \{0, 1\}$, we first sort the training set according to the scores. and define function values $g(\mathbf{x}_i) = y_i$ that give the values to be learned the PAV algorithm estimates the isotonic regression \hat{g} . If g is not already isotonic, it follows that $g(\mathbf{x}_{i-1}) \geq g(\mathbf{x}_i)$ where the examples \mathbf{x}_{i-1} and \mathbf{x}_i are called pair-adjacent violators. All these pair-adjacent violators are now replaced by their average $\hat{g}(\mathbf{x}_{i-1}) = \hat{g}(\mathbf{x}_i) = (g(\mathbf{x}_{i-1}) + g(\mathbf{x}_i))/2$ to fulfill the isotonic assumption. Finally, the algorithm returns a stepwise constant function, such that $\hat{g}(\mathbf{x}_{i-1}) \leq \hat{g}(\mathbf{x}_i)$.

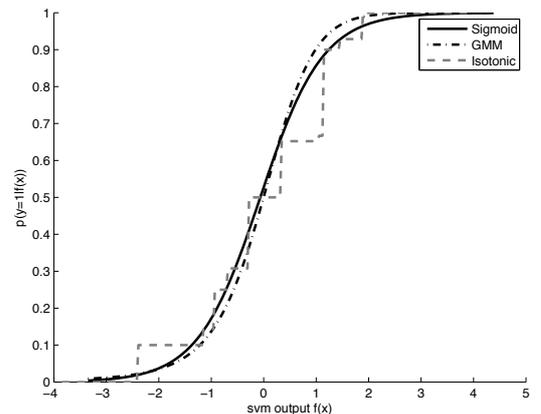


Figure 1. Comparison of the Gaussian, the sigmoid and the isotonic score calibration on an artificial dataset.

We obtain an isotonic regression output for a new example by finding the interval i in which the score fits. Then we assign $\hat{g}(\mathbf{x}_i)$ as the corresponding probability estimate.

In figure 1 we show the behavior of the three different calibration methods on an artificial dataset. While the parametric assumption of the Gaussian and the sigmoid fitting leads to the sigmoid shape of the curves, the isotonic fit shows that the algorithm just transforms the data to increasing values. For the PAV we used a MATLAB implementation made available by [11].

5 System Description

In this section we briefly describe our current speaker verification system for the upcoming NIST evaluation. This system is based on our submission for the 2006 NIST speaker recognition evaluation [12] and consists of two subsystems based on different feature extraction methods. While the first subsystem uses Mel Frequency Cepstral Coefficients (MFCCs), the second system is based on Linear Predictive Cepstral Coefficients (LPCCs).

5.1 Front End Processing

The speech data was band-limited to the frequency range 300Hz-3400Hz. Feature vectors were extracted from speech by using a 20ms Hamming window and a window shift of 10ms. Energy-based speech detection was performed on feature vectors to discard frames containing low energy.

For the first subsystem we computed mel-cepstral (MFCC) feature vectors with 13 coefficients per vector. From these static MFCC features the first and second order temporal differences were computed and appended to each vector. Additionally, the first and second order temporal differences of the frame energy were computed and also appended to each feature vector. This results in a 41 dimensional feature vector.

The second subsystem is based on modeling the vocal tract transfer function by an all-pole filter, where the model coefficients were estimated by linear prediction. Then 19 cepstral coefficients were obtained from the linear prediction coefficients and the first and second order derivatives of the coefficients and the energy were appended to form the final LPCC feature vector. Afterwards, the MFCC as well as the LPCC feature vectors were normalized to zero mean and unit standard deviation on the remaining speech data. The whole feature extraction was done using the SPRO-Toolkit [13].

5.2 The GMM system

The GMM system is based on a Universal Background Model (UBM) approach. Instead of our original system presented at the NIST 2006 evaluation [12], we used the well known ALIZE toolkit [14] for the GMM-UBM modeling. For each subsystem two gender dependent UBMs were trained on background data taken from the NIST 2000, the Switchboard Cellular and NIST 2004 datasets. Each

UBM consists of 512 mixture components and was trained via the Expectation Maximization (EM) algorithm. The speaker specific models were derived from the UBMs using a one step Maximum A-Posteriori adaptation [2]. Only the means of the mixtures were adapted with a relevance factor $\tau = 16$. During the detection test only the N -best decoded mixture components with respect to the background model were used for scoring. In our GMM-UBM experiments we set $N = 10$.

Due to the fact that different handset types (e.g., carbon, electret) and channels (e.g., land-line, cellular) are used in the evaluation, a normalization of the feature vectors was performed by Feature Mapping [15]. First, the root-UBM was trained on data from several different channels and handsets. Secondly, we adapted seven channel dependent GMMs (carbon, electret, gsm, cdma, cordless, cellphone, regular) from the root-UBM by MAP adaptation. Each utterance was then classified by the channel dependent (dep) GMMs and each feature vector \mathbf{x}_{dep} was mapped into the channel independent space via the top 1 decoded Gaussian mixture μ_{dep} of the channel dependent model. The resulting channel independent feature vectors were then used for the MAP adaptation of the speaker models.

To compensate different shifts and scales of the log-likelihood scores during testing, the so called Test Normalization (T-Norm) was applied to the ratio scores [16]. The T-Norm is defined as:

$$S(\mathbf{X}|\lambda) = \frac{\log P(\mathbf{X}|\lambda) - \mu_x}{\sigma_x} \quad (8)$$

with the mean μ_x and the standard deviation σ_x of log-likelihood scores of the test sentence against the set of background speakers. The T-Norm models were trained on impostor speakers of the NIST 2004 dataset.

5.3 The supervector SVM system

The SVM system is based on the supervector approach presented by Campbell [3]. For all background and target speakers specific GMMs were adapted from the UBM and all resulting means of each GMM were concatenated into a single supervector. Using 512 mixture components and 41 acoustical features, this results in a 20992 dimensional supervector. The SVMs were trained with the supervectors of the background and target speakers using the linear kernel of equation (3). While we have as many negative training examples as speakers in the background set, we only have one positive example for each target speaker. To speed up the classification of the test data it is possible to compact the kernel machine of equation (6) by pre-calculation of the sum of all weighted support vectors:

$$f(x) = \left(\sum_{i=1}^N \alpha_i y_i \mathbf{x} \right)^t \mathbf{x} + b \quad (9)$$

So we only need to calculate a single dot-product in the evaluation.

In addition to feature mapping in the acoustic space we compensated the channel and session variabilities in the GMM-space by another projection. This projection is called the Nuisance Attribute Projection (NAP), introduced by [17], which tries to remove the subspaces containing nuisance variabilities by the projection:

$$\mathbf{x}^* = (\mathbf{I} - \mathbf{S}\mathbf{S}^t)\mathbf{x} \quad (10)$$

where $(\mathbf{I} - \mathbf{S}\mathbf{S}^t)$ is the complementary PCA projection with matrix \mathbf{S} containing the eigenvectors of the PCA eigenvalue problem and the identity matrix \mathbf{I} . The NAP matrix was constructed by the supervectors of all the NIST 2004 data. For each speaker the data of different sessions were collected and the speaker mean was subtracted from each vector.

5.4 Performance Measure

Detection Error Tradeoff (DET) curves and Equal Error Rates (EERs) were used as performance measures. They were obtained by pooling all scores from the speaker and impostor trials. In addition to DET curves and EERs, Decision Cost Function (DCF) was also used as performance measure. The DCF is defined as a weighted sum:

$$C_{det} = C_{Miss} \times P_{Miss|Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times P_{NonTarget} \quad (11)$$

with the prior probabilities P_{Target} and $P_{NonTarget} = 1 - P_{Target}$ of target and impostor speakers, respectively. The relative costs of detection errors in this function are the costs of miss C_{Miss} and false alarm errors $C_{FalseAlarm}$. Following NIST's recommendation [18], these parameters were set as follows: $P_{Target} = 0.01$, $C_{Miss} = 10$ and $C_{FalseAlarm} = 1$.

6 Experiments

The core task of the NIST 2006 speaker recognition evaluation [18] is a speaker detection task that contains a two-channel conversation of each speaker of approximately five minutes duration in the training and also two-channel conversations in the test. The evaluation set of the core test consists of 354 male and 462 female speaker sentences for the model training. There are more than 50000 trials in the 1con4w test containing different languages as well as different transmission channels.

For the background model training, the development test and the evaluation test we used four different datasets in the NIST 2006 evaluation. The background models were trained on parts of the SWITCHBOARD Cellular dataset, the NIST 2000 evaluation set and a part of the NIST 2004 evaluation set. For the development test and the T-Norm models we also used a part of the NIST 2004 data.

Figure 2 shows the detection error tradeoff (DET) curves of the baseline GMM and SVM subsystems of the

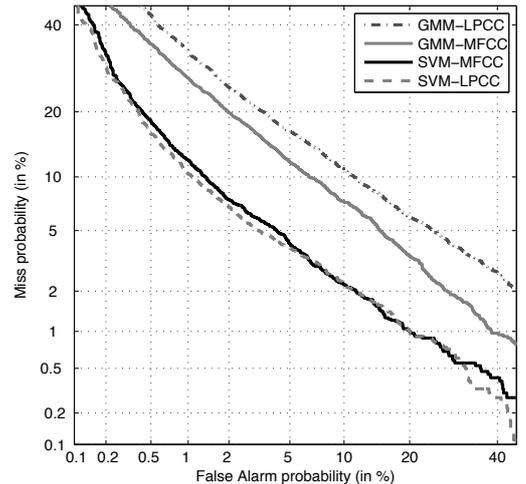


Figure 2. Performance comparison of GMM and SVM systems with T-Norm on the NIST 2006 Evaluation corpus.

current speaker detection system. As can be seen in table 1 the best results were achieved by the supervector SVM_LPCC system with an Equal Error Rate (EER) of 4.36% compared to 4.59% of the SVM_MFCC system. These results are comparable to the state of the art systems of [4][19].

Table 1. Comparison of the EER and the DCF for single subsystems on the NIST 2006 SRE task.

System	No Norm		T-Norm	
	EER	DCF	EER	DCF
1 GMM.MFCC	8.97	0.0449	8.39	0.0365
2 GMM.LPCC	9.86	0.0467	10.49	0.0426
3 SVM.MFCC	5.15	0.0247	4.59	0.0212
4 SVM.LPCC	5.01	0.0238	4.36	0.0208

In a first experiment the un-calibrated scores were fused by a simple average and a weighted sum. The fusion parameters for the weighted sum were obtained on the development set using a minimum mean-squared-error criterion.

The fusion results are given in table 2. As can be seen in the table, the weighted sum fusion of the GMM and SVM subsystems increases the performance of the whole system compared to the corresponding best subsystem in table 1. However, the fusion of all eight subsystems does not increase the fusion of the single GMM or SVM subsystems. In all fusion experiments we can observe that the systems normalized by the T-Norm achieve better results than the not normalized ones. Interestingly, the fusion results of the average and the weighted sum for the SVM subsystems are nearly the same, 3.77% and 3.76% respectively.

Table 2. Comparison of the EER and the DCF for un-calibrated fusion of the four subsystems with and without T-Norm on the NIST 2006 SRE task.

Fusion	fused Systems	No Norm		T-Norm	
		EER	DCF	EER	DCF
Average	1/2	9.03	0.0447	8.52	0.0365
	3/4	4.37	0.0220	3.77	0.0179
	1/2/3/4	6.45	0.0332	4.46	0.0212
Weighted Sum	1/2	9.00	0.0444	8.19	0.0359
	3/4	4.40	0.0223	3.76	0.0176
	1/2/3/4	5.81	0.0300	4.21	0.0199

Table 3. Comparison of the EER and the DCF for calibrated fusion of the four subsystems with and without T-Norm on the NIST 2006 SRE task.

Calibration	System	No Norm		T-Norm	
		EER	DCF	EER	DCF
Gaussian	1/2	9.14	0.0482	8.96	0.399
	3/4	4.78	0.0252	3.91	0.0184
	1/2/3/4	5.91	0.0299	4.63	0.0258
Sigmoid	1/2	8.97	0.0443	8.03	0.0357
	3/4	4.35	0.0215	3.71	0.0178
	1/2/3/4	4.76	0.0251	4.29	0.0207
Isotonic	1/2	8.80	0.0438	7.87	0.0352
	3/4	4.18	0.0211	3.63	0.0174
	1/2/3/4	4.21	0.0229	3.96	0.0189

For the calibration experiments we transformed the scores of the subsystems by the described calibration methods and fused the calibrated scores by the weighted sum. Again we used the normalized and un-normalized scores for fusion. The parameters for the calibration methods and for the weighted sum were again estimated on the development set.

The results of the calibrated and fused subsystems are presented in table 3. The Gaussian calibration decreases the performance of the systems compared to the un-calibrated fusion of table 2. The sigmoid fitting and the isotonic regression outperform the un-calibrated systems and show a decrease of the EER. If we use un-normalized scores the EER of the SVM systems decrease from 4.40% to 4.35% for the sigmoid fitting and to 4.18% for the isotonic regression compared to the un-calibrated results of the weighted sum fusion.

Using the normalized scores for fusion the sigmoid calibration method results in an EER of 3.71%, which is only a slight decrease. The isotonic calibration method achieves an EER of 3.63%. And also the minimum DCF of the sigmoid and the isotonic calibration achieve a very small gain in performance for the normalized SVM systems compared to the un-calibrated ones.

Finally, we combined the calibrated scores of the

Table 4. Comparison of the EER and the DCF for combination of the normalized and un-normalized scores of the calibrated fusions on the NIST 2006 SRE task.

Calibration	System	EER	DCF
-	3/4	3.77	0.0176
Gaussian	3/4	3.91	0.0186
Sigmoid	3/4	3.68	0.0169
Isotonic	3/4	3.61	0.0165

normalized and un-normalized SVM subsystems of table 3. Compared to the weighted sum fusion without any calibration, the combination of the SVM subsystems with and without T-Norm results in an EER of 3.77%, see table 4. We achieved the best EER for the isotonic fitting, which increases the performance about 4% relatively compared to the weighted sum fusion of the normalized SVM systems in table 2. Also the minimum DCF of the isotonic calibration beats the un-calibrated ones and achieves a reduction of about 6.25% relatively.

7 Conclusions

In this paper we investigated three concepts of score calibrating for Speaker Verification based on Support Vector Machines and Gaussian Mixture Models. The results of the sigmoid and the isotonic calibrated scores outperform the best single baseline system as well as the fusion of un-calibrated scores.

Using calibrated scores for fusion does not always lead to an improvement of the verification process. If the scores of different subsystems are already normalized by the T-Norm there is only a slight increase of performance.

The results also show that fusing subsystems based on different feature extraction methods can complement one another.

In our future research we will investigate the calibration and fusion of additional subsystems like phone and word based speaker recognition systems.

References

- [1] D. Reynolds, "An overview of automatic speaker recognition technology," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 2002, pp. 4072–4075.
- [2] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.
- [3] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "Svm based speaker verification using a gmm supervector kernel and nap variability

- ity compensation,” in *International Conference on Acoustics, Speech, and Signal Processing*, 2006, pp. 97–100.
- [4] W. Campbell, D. Sturim, W. Shen, D. Reynolds, and J. Navratil, “The mit-ll/ibm 2006 speaker recognition system: High-performance reduced-complexity recognition,” in *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.
- [5] N. Brümmer and J. du Preez, “Application-independent evaluation of speaker detection,” in *Proceedings of ODYSSEY - The Speaker and Language Recognition Workshop*, 2004, pp. 33–40.
- [6] J. C. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in *Advances in Large-Margin Classifiers*, P. Bartlett, B. Schölkopf, D. Schuurmans, and A. Smola, Eds. Cambridge, MA, USA: MIT Press, oct 2000, pp. 61–74.
- [7] B. Zadrozny and C. Elkan, “Transforming classifier scores into accurate multiclass probability estimates,” in *International Conference on Knowledge Discovery and Data Mining*, 2002.
- [8] C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, jun 1998.
- [9] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed., ser. Information Science and Statistics. Berlin: Springer, 2000.
- [10] M. Ayer, H. Brunk, G. Ewing, W. Reid, and E. Silverman, “An empirical distribution function for sampling with incomplete information,” *Annals of Mathematical Statistics*, vol. 5, pp. 641–647, 1955.
- [11] L. Dümbgen, *Statistical Software (MATLAB)*, 2000. [Online]. Available: <http://www.imsv.unibe.ch/duembgen/software/>
- [12] M. Katz, M. Schafföner, E. Andelic, S. E. Krüger, and A. Wendemuth, “The iesk-magdeburg speaker detection system for the nist speaker recognition evaluation,” in *Proceedings of NIST Speaker Recognition Evaluation*, 2006.
- [13] SPro, *Speech Signal Processing Toolkit*. [Online]. Available: <http://www.irisa.fr/metiss/guig/spro.html>
- [14] J.-F. Bonastre, F. Wils, and S. Meignier, “Alize, a free toolkit for speaker recognition,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, pp. 737–740.
- [15] D. Reynolds, “Channel robust speaker verification via feature mapping,” in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2003, pp. 53–56.
- [16] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, “Score normalization for text-independent speaker verification systems,” *Digital Signal Processing*, vol. 10, pp. 42–54, 2000.
- [17] A. Solomonoff, W. Campbell, and I. Boardman, “Advances in channel compensation for svm speaker recognition,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, pp. 629–632.
- [18] M. Przybocki and A. Martin, “The nist year 2006 speaker recognition evaluation plan,” *NIST*, 2006. [Online]. Available: <http://www.nist.gov/speech/tests/spk/2006/>
- [19] P. Matejka, L. Burget, P. Schwarz, O. Glembek, M. Karafiat, J. Cernocky, D. van Leeuwen, N. Brümmer, A. Strasheim, and F. Grezl, “Stbu system for the nist 2006 speaker recognition evaluation,” in *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.