SYSTEM COMBINATION USING AUXILIARY INFORMATION FOR SPEAKER VERIFICATION

Luciana Ferrer[1,2] *Martin Graciarena*[2] *Argyris Zymnis*[1] *Elizabeth Shriberg*[2]

[1]Department of Electrical Engineering, Stanford University, Stanford, CA, USA
[2]Speech Technology and Research Laboratory, SRI International, Menlo Park, CA, USA

ABSTRACT

Recent studies in speaker recognition have shown that score-level combination of subsystems can yield significant performance gains over individual subsystems. We explore the use of auxiliary information to aid the combination procedure. We propose a modified linear logistic regression procedure that conditions combination weights on the auxiliary information. A regularization procedure is used to control the complexity of the extended model. Several auxiliary features are explored. Results are presented for data from the 2006 NIST speaker recognition evaluation (SRE). When an estimated degree of nonnativeness for the speaker is used as auxiliary information, the proposed combination results in a 15% relative reduction in equal error rate over methods based on standard linear logistic regression, support vector machines, and neural networks.

Index Terms— Speaker recognition, System combination, Auxiliary Information, Nonnative speech, Logistic Regression

1. INTRODUCTION

We consider the task of text-independent speaker verification; i.e., given a sample from a speaker and a claimed identity we need to decide whether the claim is true or false. In the last few years a common approach to speaker verification has been to use different knowledge sources by modeling them separately, and then combining them at the score level. Score-level combination produces a final score that is later thresholded to obtain a decision. Examples of systems for which a score-level combination has resulted in significant gains over the best individual system can be found in [1, 2, 3, 4, 5]. Common combination procedures include neural networks [1, 2], support vector machines (SVM) [2], linear logistic regression [3], and weighted summation using empirically determined weights [5]. In our experience, linear combiners tend to outperform or at least match other types of combiners, and the particular method used to obtain the weights is not very relevant. Results in Section 4.3 support this observation.

The combiners mentioned above use only the output of the subsystems to perform the combination. By using auxiliary information about the trials as extra input to the combiner it may be possible to improve the overall performance. In [6] we proposed a method that uses auxiliary information from the train and test utterances, such as the length of utterances, estimated channel type (e.g. cellphone, carbon, electret), gender of the speaker, and so on, to aid the combination. The auxiliary information for train and test conversations from each trial was clustered, and separate combiners were trained for each resulting cluster. Solewicz [7, 8] simultaneously proposed a very similar method to perform combination using attributes obtained from the utterances. Significant improvements were achieved by both research teams.

In this paper we propose a system combination method in which auxiliary information is used to affect the weights of a linear logistic regression (LLR) combiner. This method is similar to that presented in [6] with the exception that the earlier work used weighted least squares to train each combiner, whereas the novel approach presented here uses LLR and a regularization method to control the complexity of the model. Furthermore, the most useful auxiliary feature explored here, the nonnativeness score, has never been tried before.

2. PROPOSED SYSTEM COMBINATION METHOD

Consider a training set with M samples, $S = \{(x_i, s_i, y_i); i = 1, ..., M\}$, where $x_i \in \mathcal{R}^d$ is the vector of scores from the individual systems for sample i, $s_i \in \{l_1, \ldots, l_L\}$ is the auxiliary information label and $y_i \in \{-1, +1\}$ is the class corresponding to the sample (in the case of speaker verification the classes are *impostor* and *target speaker*). Here, we assume that the auxiliary information is represented as a categorical value. If the original auxiliary information corresponds to a vector of continuous random variables, this vector can be quantized to obtain the labels s_i.

Our goal is to find a function $f(x_i, s_i)$ such that $I(f(x_i, s_i) > h)$ is a predictor for the class of the sample, where $I(a)$ is the indicator function (which is 1 if a is true and 0 otherwise), and h is a tunable threshold. In the linear logistic regression model such a function is the estimated posterior of the class given the input features. We propose a modification to this model that makes use of auxiliary information to achieve better predictions.

2.1. Standard Linear Logistic Regression

Linear logistic regression assumes the following model for the posterior probability of the class given the input features:

$$P(y_i|x_i) = \frac{1}{1 + e^{-y_i w^t x_i}} \quad (1)$$

where w is a vector of weights that has to be estimated. In order to simplify notation let vector x_i contain an additional component, $x_{i,0}$, which is always equal to 1, while $x_{i,1}, \ldots, x_{i,d}$ are the actual features: in our case, the outputs of the individual systems that we wish to combine. This way, the bias term is given by w_0 and is included in the scalar product $w^t x_i$.

The parameter vector w is usually estimated as the value that maximizes the log likelihood of the data assuming this model for the conditional probability, and assuming that samples are independent and identically distributed. The optimization problem is then given by

$$\text{minimize } \mathcal{L}(w) = \sum_{i=1}^{M} \log(1 + e^{-y_i w^t x_i}) \quad (2)$$

In this paper, we consider a modified version of this problem, in which we allow weights to be applied to each term in the sum. This approach is used to compensate for the priors observed in the training data if those priors are expected to be different from the priors that will be found in the test data. The modified LLR problem is given by

$$\text{minimize } \mathcal{L}(w) = \sum_{i=1}^{M} \alpha_{y_i} \log(1 + e^{-y_i w^t x_i}) \tag{3}$$

with

$$\alpha_{y_i} = \begin{cases} P/M_{-1} & \text{if } y_i = -1 \\ (1-P)/M_{+1} & \text{if } y_i = +1 \end{cases} \tag{4}$$

where P is the prior probability for class $y = -1$ expected in the test data, and M_{-1} and M_{+1} are the number of negative and positive samples found in the training data. The function $\mathcal{L}(w)$ is convex since it is a nonnegative weighted sum of convex functions [9]. Hence, the problem has a global optimum.

2.2. Auxiliary Information Conditioned LLR

In order to use the auxiliary information to obtain better combination performance, we model the posterior probability of the class given the features x_i and the auxiliary information label s_i, as follows

$$P(y_i|x_i, s_i) = \frac{1}{1 + e^{-y_i(w^t + w_{s_i}^t)x_i}} \tag{5}$$

where w_{s_i} is a weight vector that depends on the observed auxiliary information for sample i (recall that we are assuming that s_i is a categorical feature). Using the same assumptions as above we obtain a modified objective function

$$\mathcal{L}(w, w_{l_1}, \ldots, w_{l_L}) = \sum_{i=1}^{M} \alpha_{y_i} \log(1 + e^{-y_i(w^t + w_{s_i}^t)x_i}) \tag{6}$$

where $l_1, \ldots, l_L$ are the values s_i can take.

The new problem contains L additional weight vectors that need to be estimated. The obvious risk is that this could lead to overfitting of the training data. To address this potential problem, we introduce a regularization term in the objective function that aims to control the size (measured by the square norm) of the new weight vectors. It effectively shrinks the overall weight vector $w + w_{s_i}$ toward the global weight w. (Note that this is why we kept w in Equation (5), instead of simply replacing it by the new auxiliary information-dependent weights.)

Furthermore, we introduce a constraint on the sign of the overall weights applied to each feature (except for the bias term). We force them to be positive, thereby forcing the combination function to be monotonically increasing on the inputs. This is a reasonable assumption that could result in a more robust estimation of the weights when limited data is available for training.

The resulting optimization problem is given by

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{M} \alpha_{y_i} \log(1 + e^{-y_i(w^t + w_{s_i}^t)x_i}) + \sum_{j=1}^{L} \lambda_j w_{l_j}^t w_{l_j} \\ \text{subject to} \quad & w_k + w_{l_j,k} \geq 0; \quad j = 1, \ldots L; k = 1, \ldots d \end{aligned} \tag{7}$$

The value for λ_j is set to $\lambda(1 - M_j/M)$, where M_j is the number of samples with auxiliary information given by s_{l_j} and λ is a tunable parameter. That is, we penalize the growth of the norm of a weight vector depending on the number of samples available with the corresponding auxiliary information label.

We implemented a logarithmic barrier interior-point method to solve problem (7). This involves using Newton's method to solve a sequence of unconstrained problems that successively approximate the problem we want to solve. To create this sequence of problems, we augment the problem objective with a sum of logarithmic penalties. It can be shown that the complexity of this algorithm is polynomial in the problem's dimensions. For more on interior-point methods see [9, Chap. 11].

3. AUXILIARY INFORMATION

To test the proposed method we explore three types of auxiliary information that reflect characteristics of the signal, or the speaker found in the signal, and that we believed could be sucessfully used to condition the combiner parameters.

Number of phones: The purpose of this feature is to give an indication of the amount of useful information in the waveform. The output of an automatic speech recognizer (ASR) is used to obtain the total number of phones found in the waveform.

Rover posteriors: The posteriors generated by the rover step in ASR provide an indication of the confidence with which each word was recognized. We compute the geometric average of the word-level posteriors for each segment used by the recognizer and then average those values over all segments to obtain a single posterior for the unsegmented waveform.

Nonnativeness score: Nonnativeness scores are obtained automatically from the waveform as described in [10]. The system is trained using data from the Fisher corpus. Samples are labeled as belonging to either a native or a nonnative English speaker using 8-class MLLR features produced by an ASR system [11]. An SVM system is trained to classify samples into those two classes. The signed distance to the hyperplane is used as the score.

These three features are continuous measurements computed over waveforms, while the samples we refer to in Section 2 correspond to speaker verification trials, not waveforms. In the next section we will see how these values are transformed into the discrete per-trial features used to implement the method introduced above.

4. EXPERIMENTS

Experiments were conducted using data from the NIST speaker recognition evaluations (SRE) from 2004 and 2006. Data from the 2005 SRE were not used, because the distribution of nonnative speakers in that data is poorly matched to that in the 2006 data.

Each speaker verification trial consists of a test sample and a speaker model. The samples are one side of a telephone conversation with approximately 2.5 minutes of speech per side. We consider the 1-side training condition in which we are given a single conversation side to train the speaker model. This conversation corresponds to a single positive example when training the SVM model for the speaker. The negative examples are extracted from the Switchboard and Fisher databases.

The SRE04 and SRE06 1-side training condition tasks contain 15,317 and 24,013 samples, respectively. We report results in terms of both equal error rate (EER) and NIST's detection cost function (DCF). The DCF is defined as the Bayesian risk with the prior probability of the target equal to 0.01, the cost of a false alarm equal to 1, and the cost of a miss equal to 10. This is equivalent to having the target-to-impostor prior probability ratio be 1/10 and the costs of false alarms and misses be 1.

4.1. Individual systems

Below is a description of the systems included in the combination. The performance on SRE06 data is reported in parenthesis (10*DCF / EER%). For lack of space we do not list references for every system here. Please refer to [2] for such a list.

Cepstral GMM system: (0.276/6.15) This is a conventional cepstral Gaussian mixture model system adapted from a universal background model, using a 2048-component GMM.

Cepstral SVM system: (0.242/5.07) This system uses multiple projections of PCA-transformations of mean polynomial vectors of cepstral features. These features are modeled using SVMs, generating four separate scores that are combined with equal weight to produce the final score.

MLLR transform SVM system: (0.213/4.64) The features used by this system are the components of the maximum likelihood linear regression (MLLR) transforms used for speaker adaptation in SRI's speech recognition system. The transform coefficients are modeled by SVMs.

Word N-gram SVM system: (0.815/23.46) This system uses an SVM with a linear kernel with first-, second-, and third-order word N-gram frequencies as features.

SNERF system: (0.536/12.57) This system uses a set of prosodic features extracted from automatically estimated syllables. The modeling is done by SVMs. The system used here (as described in [12], but without intersession variability compensation) is an improvement over the one described in [2].

Duration systems: In this system, two sets of duration features, state- (0.705/16.02) and word-level (0.874/22.22), are modeled by GMMs.

4.2. Representing the auxiliary information

As explained in Section 3 all auxiliary features used in this paper are originally continuous values for each waveform. Our first job is to discretize these features. As explained below, this can be done by either using a clustering algorithm, or simply quantizing them into predetermined bins. The next step is to turn these waveform-level features into trial-level features. Each trial consists of a target model, in our case trained with a single waveform, and a test waveform. We transform the waveform-level auxiliary information into trial-level information simply by concatenating the labels corresponding to the train and the test waveforms. The resulting label is used as s_i in (5).

4.3. Results

Table 1 shows results on SRE04 and on SRE06 for several combination procedures. Results on SRE04 are obtained by cross-validation (training on half the data and testing on the remaining half and reversing the halves). The results are computed on the complete database after merging the results from the two halves. All parameters are tuned using this method. Results on SRE06 correspond to a combiner trained on all of SRE04. In this case, the actual DCF value is shown, in addition to the minimum DCF. The minimum DCF is the DCF value corresponding to the threshold that minimizes it on the test data, while the actual DCF is the DCF value obtained using the best threshold for the development data (in this case, SRE04).

The first block of results corresponds to system combination performed using only the available scores, without any auxiliary information. Several standard combination methods were explored: a single-layer neural network (NN), support vector machines with a linear or a radial basis function (RBF) kernel, weighted least squares (WLS), and linear logistic regression (LLR). In the case of the NN, several parameter settings were tried, with different numbers of layers and different numbers of internal nodes. For the SVM, polynomial kernels with different parameters were also tested. These results are not shown since they resulted in worse performance than those in the table. In the case of WLS, the weights are chosen such that the overall weight given to the positive samples divided by that given to the negative samples equals 1/10 (this is a common procedure to optimize the DCF value). Similarly, $P = 0.91$ (such that $(1-P)/P \approx 1/10$) is used in Equation (4). Results show that all these procedures give very similar performance. WLS leads to a significantly better EER value, but its actual DCF is worse than that of the SVM and LLR methods.

The next three blocks of results correspond to combiners that use the three different types of auxiliary information. For each type we present results using the method introduced in [6], here called auxiliary information conditioned (AIC) WLS, where separate combiners are trained for each value of the auxiliary information using WLS. In this method, auxiliary information at the waveform level is first clustered into two clusters, and a Gaussian model for each cluster is used to obtain the posterior probabilities of each cluster given a waveform. Then, for each trial, the probability of each possible combination of train and test clusters is computed as the product of the probabilities of the individual clusters. All samples are then used to train all combiners, weighting each sample by those probabilities. The second line in each block corresponds to the method presented in this paper, the auxiliary information conditioned (AIC) LLR, using the optimal parameter settings on SRE04 data. The first parameter is t, a threshold used to discretize the auxiliary information into two bins. Instead of learning the clusters based on data, as in the previous method, here we find the best value for the threshold empirically using SRE04 data. This has the advantage of finding bins that are optimal for the combination task, but the disadvantage of having to make hard decisions about the value of the auxiliary information label rather than using probabilities. As we explain below, these differences are not inherent to the methods. The second parameter is the λ value used for regularization. The effect of this parameter is more noticeable when few samples are available for a particular value of the auxiliary feature, since in that case the parameter prevents overfitting of the weights corresponding to that value. Furthermore, larger values of λ generally lead to better estimation of the actual DCF. The third parameter, pc, indicates whether or not we are imposing the positivity constraint on the weights. In our experiments, the unconstrained weights were already positive in most cases, except when little data was available for a particular value of the auxiliary information. In these cases, the positivity constraint usually resulted in a more robust estimation of the parameters, although differences in performance were always small. The third parameter, rb, indicates whether the bias term is regularized along with the other weights. In some cases, not regularizing the bias terms that depend on the auxiliary information results in better performance.

In general we see that even though both combination methods that consider auxiliary information lead to similar performance, the new method seems to be more robust at estimating the optimal score threshold for the DCF, resulting in better actual DCF values. The performance improvement obtained with either of the two methods using the nonnativeness scores as auxiliary information is large, between 15% and 20% (depending on which baseline we compare against, the WLS or the LLR), 15% for minimum DCF and between 13% and 20% in actual DCF. These gains are highly significant, with p-values much smaller than 1% in all cases. In the case of the other two auxiliary features the gains are much smaller and generally not statistically significant.

Auxiliary feature	Combination method	SRE04		SRE06		
		EER	MinDCF	EER	MinDCF	ActDCF
None	NN, no hidden layer	4.42	0.205	3.88	0.175	0.203
	SVM, linear	4.29	0.190	3.99	0.175	0.181
	SVM, RBF, $\gamma = 0.001$	4.29	0.190	3.99	0.175	**0.180**
	WLS	4.29	0.198	**3.67**	**0.173**	0.209
	LLR	4.35	0.190	3.88	**0.173**	**0.180**
Number of phones	AIC WLS	4.15	0.188	**3.45**	**0.170**	0.187
	AIC LLR, t=450, λ=20, pc=1, rb=1	3.93	0.189	3.78	0.172	**0.180**
Rover posteriors	AIC WLS	4.29	0.198	**3.67**	0.173	0.192
	AIC LLR, t=0.92, λ=200, pc=1, rb=0	4.08	0.188	**3.67**	**0.164**	**0.175**
Nonnativeness Score	AIC WLS	3.79	0.161	**3.13**	**0.146**	0.168
	AIC LLR, t=-0.6, λ=100, pc=0, rb=0	3.58	0.155	**3.13**	**0.146**	**0.156**

Table 1. Results on SRE04 with the combiner trained on SRE04 using a cross-validation procedure, and on SRE06 with the combiner trained on all of SRE04, for several different combination procedures. The best results in each block for each performance measure are highlighted in bold. t: threshold used to quantize auxiliary information, pc: positivity constraint applied, rb: regularize the bias term.

Other auxiliary features we experimented with were gender (automatically estimated from the signal) and the ratio of voiced frames to all speech frames. These did not lead to performance improvements. Many other auxiliary features can be considered. In [6] we showed significant gains when the MLLR features for each sample were used to cluster the trials. These features turned out to be useful only when that particular MLLR system was used. When other systems are considered for combination (changing that MLLR system for an improved one [11]), the gain vanishes.

Finally, we point out that the two auxiliary information-conditioned methods presented here can be more directly compared by performing some modifications. The WLS method can be modified so that each combiner is trained using only the samples corresponding to each value of the auxiliary features, as is done for the modified LLR method. If the clustering procedure is skipped and the discretization is performed as in the method proposed here, the only differences between the two methods are the models used to train each combiner and the regularization used in the LLR method. When this is done, results for the two methods are quite similar, although again, the LLR method generally leads to better actual DCF values. Furthermore, we could modify the LLR method to use all samples to train all weight vectors instead of choosing a single value of the auxiliary feature for each sample. This can be done by rewriting (5) as $P(y_i|x_i) = \sum_l P(y_i|x_i, s_i)P(s_i)$, where we consider s_i to be unknown, and consider the auxiliary information to be independent of the features. This is a direction that we plan to explore in the future.

5. CONCLUSIONS

We have proposed a modified linear logistic regression combination procedure that conditions weights on auxiliary information and that includes a regularization procedure to control the complexity of the extended model. On SRE06 data, the proposed combination results in a 15% relative reduction in equal error rate over methods based on standard linear logistic regression, support vector machines, and neural networks, when nonnativeness estimates are used as auxiliary features. We found similar gains using this auxiliary feature with our previously proposed class-dependent weighted least squares combiner. We explored several other, much simpler, auxiliary features, but did not obtain statistically significant gains. We expect that more complex auxiliary information (e.g. estimates of language or dialect, or speaking style) could yield gains in relevant applications.

6. ACKNOWLEDGMENTS

We thank Sachin Kajarekar, Nicolas Scheffer, and Andreas Stolcke for component system development and valuable input. This work was funded by SRI NSF IIS-0544682 via a subcontract to Stanford University, and through an SRI development contract with Sandia National Laboratories. The views herein are those of the authors and do not reflect the views of the funding agencies.

7. REFERENCES

[1] D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang, "The SuperSID project: Exploiting high-level information for high-accuracy speaker recognition," in *Proc. ICASSP*, Hong Kong, Apr. 2003, vol. 4, pp. 784–787.

[2] L. Ferrer, E. Shriberg, S. Kajarekar, A. Stolcke, K. Sönmez, A. Venkataraman, and H. Bratt, "The contribution of cepstral and stylistic features to SRI's 2005 NIST speaker recognition evaluation system," in *Proc. ICASSP*, Toulouse, May 2006, vol. 1, pp. 101–104.

[3] N. Brummer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 2072–2084, Sept. 2007.

[4] F. Huenupán, N. B. Yoma, C. Molina, and C. Garretón, "Speaker verification with multiple classifier fusion using Bayes based confidence measure," in *Proc. Interspeech*, Antwerp, Aug. 2007.

[5] N. Dehak, P. Kenny, and P. Dumouchel, "Continuous prosodic features and formant modeling with joint factor analysis for speaker verification," in *Proc. Interspeech*, Antwerp, Aug. 2007.

[6] L. Ferrer, K. Sönmez, and S. Kajarekar, "Class-dependent score combination for speaker recognition," in *Proc. Interspeech*, Lisbon, Sept. 2005.

[7] Y. Solewicz and M. Koppel, "Considering speech quality in speaker verification fusion," in *Proc. Interspeech*, Lisbon, Sept. 2005.

[8] Y. Solewicz and M. Koppel, "Using post-classifiers to enhance fusion of low- and high-level speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, Sept. 2007.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[10] E. Shriberg, L. Ferrer, S. Kajarekar, N. Scheffer, and A. Stolcke, "Detecting nonnative speech using speaker recognition approaches," in *Proc. Odyssey-08*, Stellenbosch, South Africa, Jan. 2008.

[11] A. Stolcke, L. Ferrer, and S. Kajarekar, "Improvements in MLLR-transform-based speaker recognition," in *Proc. Odyssey-06*, Puerto Rico, USA, June 2006.

[12] E. Shriberg and L. Ferrer, "A text-constrained prosodic system for speaker verification," in *Proc. Interspeech*, Antwerp, Aug. 2007.