Learning the Decision Function for Speaker Verification

Samy Bengio [1] Johnny Mariéthoz [2]

IDIAP–RR 00-40

January 24, 2001

published in
IEEE International Conference on Acoustic, Speech, and Signal Processing, 2001

[1] IDIAP, CP 592, 1920 Martigny, Switzerland, bengio@idiap.ch
[2] IDIAP, CP 592, 1920 Martigny, Switzerland, marietho@idiap.ch

IDIAP Research Report 00-40

Learning the Decision Function for Speaker Verification

Samy Bengio Johnny Mariéthoz

January 24, 2001

published in
IEEE International Conference on Acoustic, Speech, and Signal Processing, 2001

Abstract. This paper explores the possibility to replace the usual thresholding decision rule of log likelihood ratios used in speaker verification systems by more complex and discriminant decision functions based for instance on Linear Regression models or Support Vector Machines. Current speaker verification systems, based on generative models such as HMMs or GMMs, can indeed easily be adapted to use such decision functions. Experiments on both text dependent and text independent tasks always yielded performance improvements and sometimes significantly.

1 Introduction

The goal of speaker verification is to decide whether a given speech utterance has been pronounced by a claimed client or by an impostor. A good introduction to the field can be found in [1]. Different scenarios can take place in this framework: in *text dependent* applications, the machine knows the lexical content of the utterance used for verification, while in *text independent*, the machine does not know what the client will say.

In any case, most state-of-the-art methods start by creating a generative model for each client as well as a generative model for impostors (this will be explained in section 2). Every time a client tries to access the system, the decision is then based on a ratio between the probability that the client has pronounced the sentence and the probability that anybody else has pronounced it. When this ratio is higher than a given threshold, the access is permitted, otherwise it is refused.

In this paper, we propose to enhance the decision process, replacing this thresholding rule by a more powerful decision function, based for instance on Support Vector Machines. In the next section, we give an overview of the main steps of speaker verification. Then, we explain how the decision function could be modified into a more general function and propose two different forms of functions, namely Linear Regression and Support Vector Machines. Finally, in the experiment section, we show that these decision functions always yield better verification performance than the simple thresholding rule and sometimes significantly.

2 Baseline Speaker Verification System

Classical speaker verification models are based on statistical models. We are interested in $P(S_i|X)$ the probability that a speaker S_i has pronounced the sentence X. Using Bayes theorem, we can write it as follows:

$$P(S_i|X) = \frac{P(X|S_i)P(S_i)}{P(X)}. \tag{1}$$

To decide whether or not a speaker S_i has pronounced a given sentence X, we compare $P(S_i|X)$ to the probability that any other speaker has pronounced X, which we write $P(\bar{S}_i|X)$. When $P(\bar{S}_i|X)$ is the same for all clients S_i, which is the case in this paper, we replace it by a speaker independent model $P(\Omega|X)$ where Ω represents the *world* of all the speakers. The decision rule is then:

$$\text{if } P(S_i|X) > P(\Omega|X) \text{ then } X \text{ was generated by } S_i. \tag{2}$$

Using equation (1), inequality (2) can then be rewritten as:

$$\frac{P(X|S_i)}{P(X|\Omega)} > \frac{P(\Omega)}{P(S_i)} = \delta_i \tag{3}$$

where the ratio of the prior probabilities is usually replaced by a threshold δ_i since it does not depend on X. Taking the logarithm of (3) leads to the *log likelihood ratio*:

$$\log P(X|S_i) - \log P(X|\Omega) > \log \delta_i = \Delta_i. \tag{4}$$

To implement this, we need to create a model of $P(X|S_i)$ for every potential speaker S_i, as well as a *world model* $P(X|\Omega)$, and then we need to estimate the threshold Δ_i for each client S_i. Alternatively, it is often more convenient (because of a lack of data available for each client) to search for a unique threshold Δ that would be client independent. This is the approach taken in this paper. Other researchers have proposed other *a priori* methods to select Δ_i (see [2] for a comparison) or adaptive methods in order to account for the long-term variability of people's voice [3].

Depending on the task, the models can be estimated using different statistical tools. When we know the sentence that each speaker is supposed to pronounce, we can use a specific Hidden Markov

Model (HMM) tailored to the given sentence. When we know nothing about the sentence pronounced by the speakers, we can use either an ergodic HMM or a Gaussian Mixture Model (GMM) to represent $P(X|S_i)$ and $P(X|\Omega)$.

In any case, we have to decide the correct architecture (number of Gaussians for GMMs, number of states, topology, and number of Gaussians per state for HMMs) using methods such as cross-validation.

Finally, when all the client models and the world model are created, we still need to find the threshold Δ_i of the decision rule (4).

3 Learning the Decision Function

Even if most state-of-the-art methods for speaker verification are based on generative models (such as HMMs or GMMs), a better solution should in theory be to use a *discriminant* framework (see [4] for a discussion on discriminant versus generative models). A simple way to add some discriminant power to these generative models is to use discriminant decision rules. Let us rewrite (4) as follows:

$$y = \log \hat{P}(X|S_i) - \log \hat{P}(X|\Omega) - \Delta_i \tag{5}$$

such that **the sign of y gives the decision**. Note that $\hat{P}()$ means that we are using the estimates of $P()$ obtained by our models. The goal is thus to find a value of Δ_i that optimizes a given criterion over the decision. If the probabilities are perfectly estimated (which is usually not the case), then the Bayes Decision is the optimal decision.

3.1 Bayes Decision Rule and HTER Cost Function

In classical speaker verification systems, when no prior information is given on the cost of the different kinds of errors, the Bayes Decision rule is applied by selecting the value of Δ_i in (5) that minimizes the *Half Total Error Rate*:

$$\text{HTER} = \frac{1}{2}(\%\text{FA} + \%\text{FR}) \tag{6}$$

where $\%FA$ is the rate of false acceptances and $\%FR$ is the rate of false rejects. Note that this cost function changes the relative weight of client and impostor accesses in order to give them equal weight, instead of the one induced by the training data.

In this paper, we are interested in the case where the probabilities are not perfectly estimated and where the Bayes Decision might not be the optimal solution. We thus propose to explore other forms of decisions, based either on linear functions or on more complex functions such as Support Vector Machines.

3.2 Linear Regression

A general linear decision rule could be:

$$y = a_i \log \hat{P}(X|S_i) + b_i \log \hat{P}(X|\Omega) + c_i \tag{7}$$

where parameters a_i, b_i, and c_i are learned by minimizing the Mean Squared Error (MSE) over a training set where we tag $y = 1$ for clients and -1 for impostors. In fact, in order to optimize the HTER cost, we used a weighted MSE cost to give the same relative weight between the set of clients and the set of impostors. The decision is then to accept clients if

$$\log \hat{P}(X|S_i) + \frac{b_i}{a_i} \log \hat{P}(X|\Omega) > -\frac{c_i}{a_i}. \tag{8}$$

3.3 Support Vector Machines

In the following, we propose to further enhance the decision function using more powerful models, such as the recently proposed Support Vector Machines (SVMs) [4]. These have been applied to many problems in classification and regression tasks, generally yielding good performance compared to other algorithms. The resulting function is of the form

$$y = \sum_{j=1}^{l} y_j \alpha_j K(x, x_j) + b \tag{9}$$

where x would be the two-dimensional vector containing $\log \hat{P}(X|S_i)$ and $\log \hat{P}(X|\Omega)$), the client and world model scores, and x_j the corresponding vector for sentence X_j. $y_j = 1$ when X_j was pronounced by S_i and -1 otherwise. l is the number of training sentences, the α_j and b are the parameters of the model, and $K(x, x_j)$ is a kernel function that can have different forms; the simplest is the dot product kernel:

$$K(x, x_j) = x\ x_j \tag{10}$$

which leads to a Linear SVM, while a more general kernel is the Radial Basis Function (RBF) kernel:

$$K(x, x_j) = \exp\left(\frac{-\|x - x_j\|^2}{\sigma^2}\right) \tag{11}$$

with σ being a parameter that needs to be selected (by cross-validation for instance) and that determines the complexity of the function.

Without going into details on the specific method to train SVMs[1], it is important to note that the training criterion used in SVMs is related to the number of classification errors, as opposed to the MSE used in Linear Regression models. Moreover we had to change the normal SVM formulation in order to optimize the HTER cost (6) by changing the relative weight of each example [6]. Note that in the resulting solution (9), most α_j are equal to 0, and the examples with non-zero α_j are called *support vectors*. Note however that the training complexity of SVMs is quadratic on the number l of examples, which makes the use of SVMs for large datasets difficult.

4 Experimental Results

4.1 Settings

In order to compare different decision functions, we have used the *PolyVar* telephone database [7, 8], that contains two sets (called hereafter *data1* and *data2*) of 19 clients (12 men and 7 women) as well as another population of 56 speakers (28 men and 28 women) used for the world model. For each client, a training set contains 5 repetitions of 17 words (composed of 3 to 12 phonemes each), while a separate test set contains on average 18 repetitions of the same 17 words, for a total of 6000 utterances, as well as on average 12000 impostor utterances. Each sentence was parameterized using 12 LPCC coefficients of order 16 as well as the energy, complemented by their first (delta) and second (delta-delta) derivatives, for a total of 39 coefficients. A simple silence detector based on an unsupervised bi-Gaussian model was also used to remove all silence frames.

Two different kinds of generative models for clients and world were created: one using GMMs, used for text independent experiments, and one using HMMs, used for text dependent experiments. All models were trained using the Maximum Likelihood criterion with the EM algorithm.

[1]See [5] for a good introduction on SVMs.

4.2 Text Independent Experiments

To determine the number of Gaussians for the world model, we used a simple validation technique, training on 90% of the available training set for the world model and selecting the model with the highest likelihood over the remaining 10%. This led us to a GMM with 1000 Gaussians.

To determine the number of Gaussians for the client models, we used a 10-fold cross-validation technique: every client model is trained with a given number of Gaussians on nine tenths of the training set, and the test likelihood is computed on the rest of the data, and this is done for all 10 partitions of the training set. The model that gave the highest overall test likelihood had 200 Gaussians.

Having fixed the client and world models, we trained different decision rules using test data of every client in *data1* and recorded their performances over every client in *data2*. We also did the converse (train with *data2* and test with *data1*) and computed the average. This is thus a 2-fold cross-validation technique.

Table 1 gives the results of these experiments. We compared a Linear SVM as well as an SVM with an RBF kernel using σ=50. Note that this value of σ was found by cross-validation, although it was observed that its actual value (ranging from 1 to 100) did not yield significant differences. These two SVM models were compared to a Linear Regression model as well as the classical Bayes Decision rule. We can see that usual Bayes Decision rule yielded the worst results, while both SVM gave similar but significantly better results. The relative improvement between SVM with RBF(σ=50) and Bayes Decision is equal to 14.8%, which is particularly significant.

Figure 1 shows the different decision rules trained on *data1* (the lines or curves) as well as the scores of clients and world model (the big and small dots) for *data2* (hence we see the test performance of each decision rule). We can see that the SVM with RBF(σ=50) found a slightly non-linear solution that gives it a small but significant advantage with respect to the Bayes Decision.

Model	# params	HTER Error (%)	
		Train	Test
SVM with RBF(σ=50)	2204	4.77	4.73
Linear SVM	2	4.78	4.75
Linear Regression	2	5.04	5.28
Bayes Decision	1	5.17	5.55

Table 1: Results for a text independent task. With 95% confidence, the SVM with RBF(σ=50) and the Linear SVM are significantly better than Bayes Decision (over 36186 tests).

4.3 Text Dependent Experiments

For text dependent experiments, we used HMMs with the same topology for the world and the client models. This topology was based on previous experiments on another dataset, *CAVE* [9], and was thus not optimized for *PolyVar*. The resulting architecture is an HMM for each word with 2 states per phoneme and 3 Gaussians per state.

Having trained the client and world models, we used the same protocol as for the text independent experiments to train different decision rules: the 2-fold cross-validation technique previously described.

Table 2 gives the results of these experiments. We compared the same models as for text independent experiments. This time, Bayes Decision yielded competitive results, and both SVM gave similar results, but the difference between all models shrinked: the relative improvement between SVM with RBF(σ=50) and Bayes Decision is only 2.70%. This is probably due to the fact that HMMs are better generative models that GMMs for this dataset. Indeed, if the models were perfectly estimated, the Bayes Decision with a threshold set *a posteriori* should give the optimal solution, but this is not a

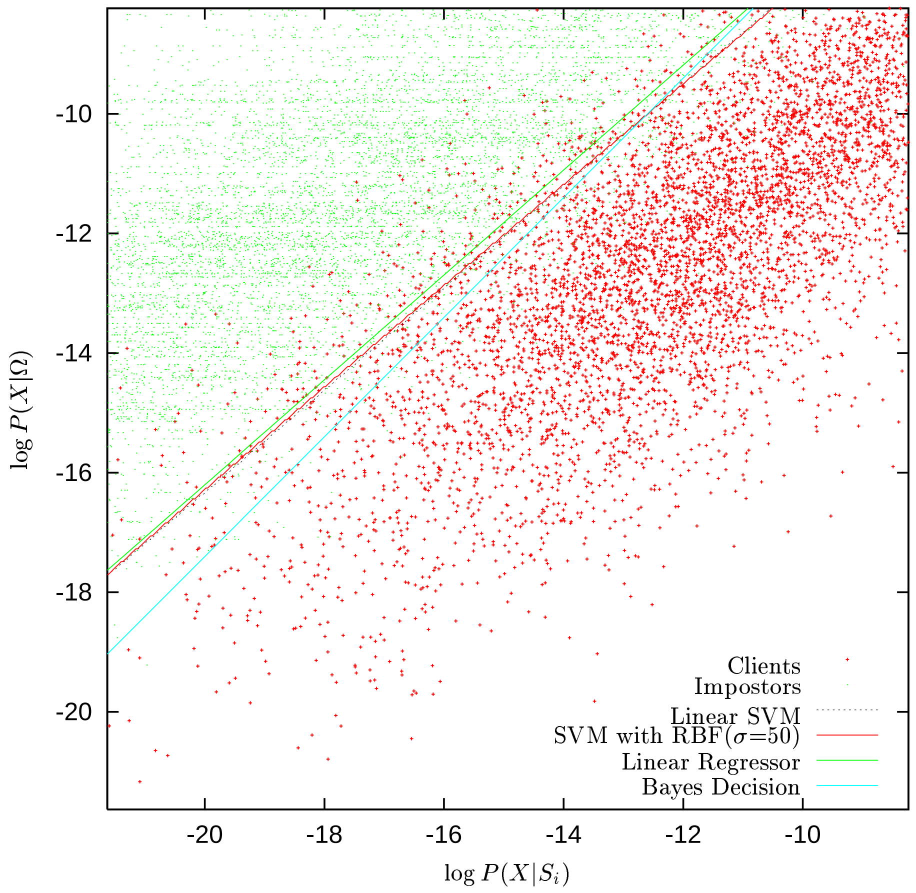

Figure 1: Different models to separate clients and impostors, in a text independent task.

realistic scenario as it is in general impossible to estimate perfectly the models and the threshold given the available data.

5 Conclusion

In this paper, we have presented a simple technique to replace the classical thresholding rule used to accept or reject a speaker by a more powerful decision function, based on Linear Regression models or Support Vector Machines. Such functions are trained in a discriminative way, using as inputs the scores of client and world models previously trained using classical generative models such as GMMs and HMMs. Experiments on both text dependent and text independent tasks always yielded performance improvements and they were significant in text independent. Further experiments using additional inputs, or using client dependent discriminative models, are currently under investigation and already gave very promising results.

6 Acknowledgments

This research has been partially carried out in the framework of the European BANCA project, funded by the Swiss OFES project number 99-0563-1. The authors would like to thank Hervé Bourlard for fruitful discussions. The experiments using SVMs have been realized using *SVMTorch* [10].

Model	# params	HTER Error (%)	
		Train	Test
SVM with RBF(σ=50)	1750	3.28	3.34
Linear SVM	2	3.34	3.33
Linear Regression	2	3.59	3.60
Bayes Decision	1	3.38	3.42

Table 2: Results for a text dependent task. With 95% confidence, there was no significant difference between Bayes Decision and the other models (over 36186 tests).

References

[1] S. Furui, "Recent advances in speaker recognition," in *Audio- and Video-based Biometric Person Authentication*, Springer, Ed., 1997, pp. 237–252.

[2] J.-B. Pierrot, J. Lindberg, J. Koolwaaij, H.-P. Hutter, D. Genoud, M. Blomberg, and F. Bimbot, "A comparison of a priori thresholds settings procedures for speaker verification in the CAVE project," in *Proc. of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, 1998, vol. I, pp. 125–128.

[3] T. Matsui and S. Furui, "Robust methods of updating model and a priori threshold in speaker verification," in *Proc. of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, 1996, vol. I, pp. 97–100.

[4] V. N. Vapnik, *The nature of statistical learning theory*, Springer, second edition, 1995.

[5] C. Burges, "A tutorial on support vector machines for pattern recognition," *Knowledge Discovery and Data Mining*, vol. 2, no. 2, 1998.

[6] Y. Lin, Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," Tech. Rep. 1016, Department of Statistics, University of Wisconsin, 2000, Available at http://www.stat.wisc.edu/p/stat/ftp/pub/yilin/tr1016.ps.

[7] G. Chollet, J.-L. Cochard, A. Constantinescu, C. Jaboulet, and P. Langlais, "Swiss french polyphone and polyvar: telephone speech databases to model inter- and intra-speaker variability," IDIAP-RR 01, IDIAP, 1996, Available at ftp://www.idiap.ch/pub/reports/1996/rr96-01.ps.gz.

[8] J. Mariéthoz and C. Mokbel, "Synchronous alignment," IDIAP-RR 06, IDIAP, 1999, Available at ftp://www.idiap.ch/pub/reports/1999/rr99-06.ps.gz.

[9] F. Bimbot, H-P. Hutter, C. Jaboulet, J. Koolwaaij, J. Lindberg, and J-B. Pierrot, "Speaker verification in the telephone network: Research activities in the CAVE project," in *Proc. of the European Conference on Speech Communication and Technology*, 1997.

[10] R. Collobert and S. Bengio, "Support vector machines for large-scale regression problems," IDIAP-RR 17, IDIAP, 2000, Available at ftp://www.idiap.ch/pub/reports/2000/rr00-17.ps.gz.