A Max Kernel For Text-Independent Speaker Verification Systems

Johnny Mariéthoz and Samy Bengio

IDIAP Research Institute, CP 592, 1920 Martigny, Switzerland and
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland,
{marietho,bengio}@idiap.ch

Abstract

In this paper, we present a principled SVM based speaker verification system. A general approach is developed that enables the use of any kernel at the frame level. An extension of his approach using the Max operator is then proposed. The new system is then compared to state-of-the-art GMM and other SVM based systems found in the literature on the Polyvar database. It is found that the new system outperforms, most of the time, the other systems, statistically significantly.

1 Introduction

Speaker verification systems are increasingly often used to secure personal information, particularly for mobile phone based applications. Furthermore, text-independent versions of speaker verification systems are the most used for their simplicity, as they do not require complex speech recognition modules. The most common approach using machine learning algorithms are based on Gaussian Mixture Models (GMMs) [14], which do not take into account any temporal information. They have been intensively used thanks to their good performance, especially with the use of the Maximum A Posteriori (MAP) [9] adaptation algorithm. This approach is based on the density estimation of an impostor data distribution, followed by its adaptation to a specific client data set. As the estimation of these densities is not the true goal of speaker verification systems, but rather to discriminate the client and impostor classes, discriminative models seem more appropriate.

As a matter of fact, Support Vector Machine (SVM) based systems have been the subject of several recent publications in which they obtain similar or even better performance than GMMs on several text-independent speaker verification tasks. One of these systems, based on an explicit polynomial expansion [5] has obtained good results during the NIST 2003 evaluation [6], but suffers from a lack of theoretical interpretation and justification. Moreover the approach precludes the use of the so-called kernel trick, which is at the heart of the flexibility of SVM based approaches. We thus propose in this paper a more principled SVM based speaker verification system that can make use of the kernel trick. Furthermore, a kernel based on the Max operator is proposed and compares favorably against the state-of-the-art approaches

The outline of this paper goes as follows. In Section 2, we present the problem of text-independent speaker verification, including a description of the GMM and SVM based system, the measures and the databases used in the experimental part. The new proposed approach is then presented in Section 3, and is compared to similar approaches found in the literature. The Max kernel is then proposed in section 4. Some improvements are also proposed at the end of this section. Results on a speaker verification tasks are presented in Section 5, while conclusion and future work are proposed in Section 6.

2 Text-Independent Speaker Verification

Person authentication systems are in general designed in order to let genuine clients access a given service while forbidding it to impostors. In this paper, we consider the problem from a machine learning point of view and we treat it independently for each speaker. The problem can thus be seen as a two class classification task and is defined as follows. Given a sentence $\mathbf{X}$ pronounced by a speaker S_i, we are searching for a parametric function $f_{\Theta_{S_i}}()$ and a decision threshold Δ_{S_i} such that

$$f_{\Theta_{S_i}}(\mathbf{X}) > \Delta_{S_i} \approx \Delta \qquad (1)$$

for all accesses $\mathbf{X}$ coming from S_i and only for them. 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. In this paper, we will use two kind of set of functions $f_\Theta()$.

2.1 GMM Based Systems

State-of-the-art speaker verification are based on GMMs, with one client GMM model compared to an impostor GMM model. In this paper, the impostor model (called *world* model) is the same for all clients. The client model is adapted from the world model using a Maximum A Posteriori adaptation algorithm. The decision function is called *log likelihood ratio* and is given by:

$$f_\Theta(\mathbf{X}) = \frac{1}{T}\sum_t \log \frac{\sum_{n=1}^{N} w_n \cdot \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n)}{\sum_{n=1}^{\bar{N}} \bar{w}_n \cdot \mathcal{N}(\mathbf{x}_t; \bar{\boldsymbol{\mu}}_n, \bar{\boldsymbol{\sigma}}_n)} > \log \Delta$$

where T is the number of frames for a given sentence $\mathbf{X}$, $\mathbf{x}_t$ is the t^{th} frame of $\mathbf{X}$, N is the number of Gaussians of the client model, $\bar{N}$ is the number of Gaussians of the world model, $\Theta_+ = \{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, w_n\}$ are the GMM parameters for the client model and $\Theta_- = \{\bar{\boldsymbol{\mu}_n}, \bar{\boldsymbol{\sigma}_n}, \bar{w_n}\}$ are the GMM parameters for the world model. Note that $\frac{1}{T}$ is an empirical normalization factor added to be independent of the length of the sentence.

2.2 SVM Based Systems

Support Vector Machines (SVMs), as proposed by [15], are more and more often used in machine learning applications. Even if the speaker verification task can be seen as a two-class classification problem, SVMs can not be applied directly: examples are sequence and classical SVMs can only work with fixed size vectors. Nevertheless, we review here the SVM model. The underlying decision function can be written as:

$$f_\Theta(\mathbf{x}) = b + \mathbf{w} \cdot \Phi(\mathbf{x}) \tag{2}$$

where $\mathbf{x}$ is the current example, $\Theta = \{b, \mathbf{w}\}$ are the model parameters and $\Phi()$ is an "a priori" chosen function that maps the input data into some high dimensional space. It can be shown that solving the SVM problem allows to express the decision function as an hyper-plane defined by a linear combination of training examples in the feature space $\Phi()$. We can thus express (2) using the dual formulation as:

$$f_\Theta(\mathbf{x}) = b + \sum_{l=1}^{L} \alpha_l y_l \Phi(\mathbf{x}_l) \cdot \Phi(\mathbf{x}).$$

We call *support vector* a training example for which $\alpha_l \neq 0$. As $\Phi()$ only appears in dot products, we can replace them by a kernel function as follows:

$$f_\Theta(\mathbf{x}) = b + \sum_{l=1}^{L} \alpha_l y_l k(\mathbf{x}_l, \mathbf{x}).$$

This so-called "kernel trick" helps to reduce the computational time and also permits to project $\mathbf{x}_l$ into virtually infinite dimensional feature spaces without the need to compute anything in that space. The two most well known kernels are the Radial Basis Function (RBF) kernel,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right) \tag{3}$$

where σ is a hyper-parameter than can be used to tune the capacity of the model, and the polynomial kernel,

$$k(\mathbf{x}_i, \mathbf{x}_j) = (a\mathbf{x}_i \cdot \mathbf{x}_j + b)^p \tag{4}$$

where p, b, a are hyper-parameters that control the capacity.

Several SVM based approaches have been proposed recently to tackle the speaker verification problem [16, 6]. While this task is mainly a two-class classification problem for each client, it differs from the classical problem by the nature of the examples, which are variable size sequences. Since classical SVMs can only deal with fixed size vectors as input, two approaches can be considered. Either work at the frame level and merge the frame scores in order to obtain only one score for each sequence; or try to convert the sequence into a fixed size vector. The first approach is probably not ideal, because we try to solve a problem which is more difficult than the original one: indeed, each frame contains only few discriminant information and some even contain no information (like silence frames). Most solutions are thus based on the second approach, such as the so-called Fisher scores or the explicit polynomial expansion.

Fisher score based systems [10] compute the derivative of the log likelihood of a generative model with respect to its parameters and use it as input to an SVM. This provides a nice theoretical framework, but is very costly for GMM based generative models with large observation space (which yield more than 10 000 parameters in general for speaker verification) and furthermore still needs to train generative models.

The explicit polynomial expansion approach [6] expands each frame of a sequence using a polynomial function and averages them over the whole sequence in the feature space. The resulting fixed size vector is used as input to a linear SVM ($\Phi(\mathbf{x}) = \mathbf{x}$). The method is quite fast and robust, but is a bit tricky to tune. In this paper we propose a new approach with a better framework from a machine learning point of view that generalizes the polynomial approach and extends it to any kernel function.

2.3 Measures

In this paper, we used *a posteriori* measures such as Equal Error Rates (where the threshold Δ is chosen such that (FAR=FRR) and DET curves [12] which present FRR as a function of FAR by varying Δ to tune and analyze systems. On the other hand, to fairly compare models on unseen data, we used *a priori* measures such as Half Total Error Rate (HTER) $\frac{(\text{FAR}_\Delta + \text{FRR}_\Delta)}{2}$ and the Expected Performance Curves [2] which show HTER on the test set as a function of some trade-off parameter α of a convex combination of FAR and FRR used to select Δ on a separate development set:

$$\Delta^* = \arg\min_{\Delta}\left(\alpha \text{FAR}_\Delta + (1-\alpha)\text{FRR}_\Delta\right). \qquad (5)$$

Finally, we have also added for both curves and values a confidence interval of 95% using a modified version of the standard proportion test [1].

2.4 Experimental Setup and Polyvar Database

The *Polyvar* telephone database [7], contains two sets (called hereafter *development* and *test* sets) of 19 clients (12 men and 7 women) as well as another population of 56 speakers (28 men and 28 women) used to train 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 client has 17 models, one for each word, and only 5 sequences are available to train each model. As in the original protocol, only same word accesses are done. The development set of this database is used to analyze the systems presented in this paper.

Each sentence was parameterized using 24 *Linear Filter Cepstral Coefficients* (LFCC) [13] of order 16, complemented by their first derivative (delta) and delta-energy, for a total of 33 coefficients. All frames were normalized in order to have zero mean and unit standard deviation per sequence. A simple silence detector based on an unsupervised bi-Gaussian model was also used to remove all silence frames [11].

A state-of-the-art GMM based text-independent speaker verification system was used as a baseline to assess the various proposed systems. Two gender dependent world models were trained using Expectation Maximization with a Maximum Likelihood criterion. A lower bound of the variances of the Gaussians was used to control the capacity and was fixed to a certain percentage of the total variance of the data. The final world model was then obtained by merging the two gender dependent models. For each client, a model was then created by adapting the final world model using a MAP algorithm [14]. Only the mean parameters of the client model were adapted using the following update rule.

All hyper-parameters of the baseline system, such as number of Gaussians, variance flooring factor and MAP adaptation factor, were selected on the development set.

3 A Principled Approach to Sequence Kernels for Speaker Verification

One particularity of the speaker verification problem is that inputs are sequences. This requires, for SVM based approaches, a kernel that can deal with variable size sequences. A simple solution, which does not take into account any temporal information, as in the case of GMMs, is the following:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{T_i T_j} \sum_{t_i=1}^{T_i} \sum_{t_j=1}^{T_j} k(\mathbf{x}_{t_i}, \mathbf{x}_{t_j}) \qquad (6)$$

where $\mathbf{X}_i$ is a sequence of size T_i and $\mathbf{x}_{t_i}$ is a frame of $\mathbf{X}_i$. We thus apply a kernel $k()$ to all possible pairs of frames coming from the two input sequences $\mathbf{X}_i$ and $\mathbf{X}_j$. This will be referred to in the following as the Mean operator approach (as we are averaging all possible kernelized dot products of frames).

This kind of kernel has already been applied successfully in other domains such as object recognition [3]. It has the advantage that all forms of kernels can be used for $k()$ and the resulting kernel $K()$ respects all Mercer conditions [4] which make sure that for all possible training sets the resulting Hessian is semi-positive which makes the problem convex. Two forms of kernels $k()$ are used in this paper: an RBF kernel (3) and a polynomial kernel (4). For the polynomial kernel, we fixed a and b to $p!^{-\frac{1}{2}p}$ in order to avoid numerical problems for large values of p. The degree p of the polynomial kernel and the standard deviation σ of the RBF kernel are thus the only hyper-parameters tuned over the development set.

3.1 Comparison with Campbell's Polynomial Approach

[5] recently proposed a new approach using SVMs for speaker verification based on an explicit polynomial expansion. He proposed a new kernel called GLDS (Generalized Linear Discriminant Sequence) of the form:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \Phi(\mathbf{X}_i)\mathbf{\Gamma}^{-1}\Phi(\mathbf{X}_j) \qquad (7)$$

where $\mathbf{\Gamma}$ is a matrix derived by the metric of the feature space induced by $\Phi()$. This matrix is usually a diagonal approximation γ of the covariance matrix computed over all the training data. He furthermore defines:

$$\Phi(\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T}\phi(\mathbf{x}_t)$$

and

$$\phi'(\mathbf{x}_t) = \frac{\phi(\mathbf{x}_{t_i})}{\sqrt{\gamma}}$$

where $\phi'()$ is the normalized version of $\phi()$, and can thus rewrite (7) as:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{T_i}\sum_{t_i=1}^{T_i}\phi'(\mathbf{x}_{t_i}) \cdot \frac{1}{T_j}\sum_{t_j=1}^{T_j}\phi'(\mathbf{x}_{t_j})$$

where $\phi'()$ maps the example $\mathbf{x}_t \in \mathbb{R}^d \rightarrow \mathbb{R}^K$, $K = \frac{(d+p-1)!}{(d-1)!p!}$ is the dimension of the feature space, d is the dimension of each frame augmented by a new coefficient equal to 1, p is the degree of the polynomial expansion and each value $k \in \{1, ..., K\}$ of the expanded vector corresponds to a combination of $r_1, r_2, ..., r_d$ as follows:

$$\phi'_{k(r_1,r_2,...,r_d)}(\mathbf{x}_t) = \frac{1}{\sqrt{\gamma_k}} x_1^{r_1} x_2^{r_2} ... x_d^{r_d} \tag{8}$$

for all possible combinations of $r_1, r_2, ..., r_d$ such that $\sum_{i=1}^{d} r_i = p$ and $r_i \geq 0$.

Campbell proposed a method to normalize each expanded coefficient using γ computed over all concatenated impostor sequences. Once all vectors are computed and normalized, they can be used as input to a linear SVM.

While this approach yielded good performance on NIST 2003, it has some drawbacks. First no kernel trick can be applied: it seems not possible to include the normalization $\frac{1}{\sqrt{\gamma_k}}$ into it. And since we need to project explicitly the data into the feature space, only finite space kernels are applicable (an RBF kernel could not be used for instance).

The second main problem of this approach is related to the capacity [15]. For a polynomial kernel "a la Campbell" the only available parameter is the degree p of the polynomial, but this parameter is hardly tunable: for respectively $p =$1, 2, 3 and 4 the resulting feature space dimensions are 33, 595, 7140 and 66045. It is then difficult to correctly set the capacity. Moreover, as the best value is $p = 3$ for the considered databases, the dimension seems quite huge if we consider that a few hundred examples only are used for training.

In the following, we will try to answer questions such as: why is a normalization step required? Does taking the average of the $\phi()$ values over all frames make any sense?

We will first show that our proposed approach solves almost all drawbacks of the explicit polynomial approach and still includes the solution proposed by Campbell. Let us start by rewriting (6) as follows:

$$\begin{aligned} K(\mathbf{X}_i, \mathbf{X}_j) &= \frac{1}{T_i T_j}\sum_{t_i=1}^{T_i}\sum_{t_j=1}^{T_j}\phi(\mathbf{x}_{t_i}) \cdot \phi(\mathbf{x}_{t_j}) \\ &= \frac{1}{T_i}\sum_{t_i=1}^{T_i}\phi(\mathbf{x}_{t_i}) \cdot \frac{1}{T_j}\sum_{t_j=1}^{T_{t_j}}\phi(\mathbf{x}_{t_j}). \end{aligned}$$

Let us define $k(\mathbf{x}_i, \mathbf{x}_j)$ of (6) as a polynomial kernel of the form $(\mathbf{x}_i \cdot \mathbf{x}_j)^p$, where p is the degree of the polynomial. In order to perform an explicit expansion with the standard polynomial kernel we need to express the corresponding $\phi()$ function [4] in a similar way to (8). Each value of the extended vector is thus given by:

$$\begin{aligned} \phi_{k(r_1,r_2,...,r_d)}(\mathbf{x}_t) &= \sqrt{c_k} x_1^{r_1} x_2^{r_2} ... x_d^{r_d}, \\ \sum_{i=1}^{d} r_i &= p, \quad r_i \geq 0 \\ \text{where} \quad c_k &= \frac{p!}{r_1! r_2! ... r_{d+1}}, \quad k \in \{1, ..., K\} \end{aligned} \tag{9}$$

and each input frame is augmented by a new coefficient equal to 1.

When we compare equations (9) and (8) the difference only lies in the polynomial coefficients: each term is multiplied by a coefficient $\sqrt{c_k}$ in the proposed approach while the explicit expansion needs a normalization factor $\frac{1}{\sqrt{\gamma_k}}$ that disables the kernel trick. Empirically, when we compared the coefficient values for each term in the proposed approach with the normalization vector obtained by the explicit method they look very similar. In fact, the performance obtained on the development set of Polyvar are very similar, as shown by the DET curves given in Figure 1 and Equal Error Rates provided in Table 1. Figure 1 and Table 1 also provide results using an RBF kernel to show that it now becomes possible to change the kernel, even if, in that case, the best kernel was still polynomial.

The drawback of our method, however, is the computational complexity for long sequences. If S is the number of speakers, N_+ the number of positive examples per speaker, N_- the number of negative examples, and M the average number of frames of an example, then the training time complexity is given by:

$$O(S(N_+^2 M^2) + N_- M^2).$$

Long sequences are thus very costly. This is not a problem for databases such as Polyvar, especially, because $N_+ << N_-$ and negative examples are shared between all clients and can thus be cached in memory. It is still unfortunately intractable for other databases such as NIST, in its present form. The test complexity for each access is:

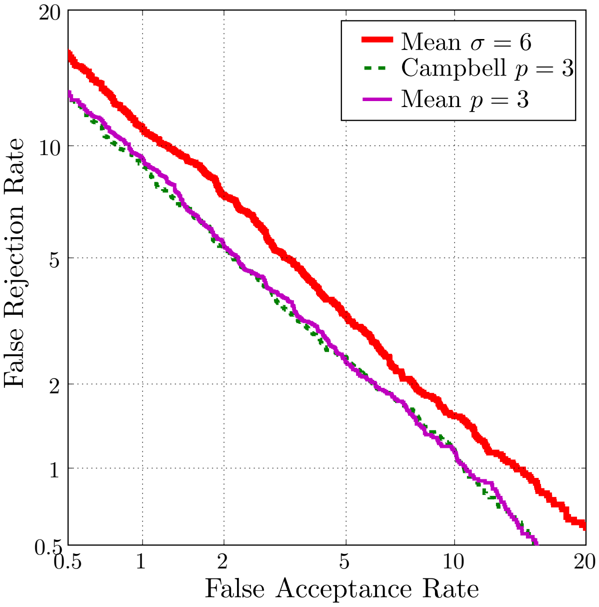

Figure 1. DET curves on the development set of the Polyvar database comparing the explicit polynomial expansion (from Campbell), the principled polynomial kernel and an RBF kernel (using the Mean operator).

$$O(X_l^2 M^2)$$

where X_l is the number of support vectors. Even for the test, computing scores for long sequences can take too long. This problem can certainly be addressed using clustering techniques and will be in a future work.

4 Max Approach

In equation (6), we can see that all frames of two sequences are compared with each other. Does this make sense? Is it a good idea to compute a similarity measure (which is what a kernel does) between frames coming from different sub-acoustic units? The answer is probably “no”. Moreover, we expect a similarity between two identical sequences to be maximum, which is not necessarily the case with equation (6), since we take the average. To illustrate this, let us create a sequence $\mathbf{X}_j$ contains exactly one frame taken from another sequence $\mathbf{X}_i$ that gives the maximum value of $k(\mathbf{x}_{t_i}, \mathbf{x}_{t_j})$ in (6). In that case, one can easily obtained $K(\mathbf{X}_i, \mathbf{X}_j) \geq K(\mathbf{X}_i, \mathbf{X}_i)$.

	Campbell $p = 3$	Mean $p = 3$	Mean $\sigma = 3$
EER [%]	3.38	3.46	4.08
95% Confidence	±0.27	±0.28	±0.3
# Support Vectors	68	87	62

Table 1. Comparison of EERs on the development set of the Polyvar database between the explicit polynomial expansion and a principled polynomial kernel applying the mean operator. The second line provides a 95% confidence interval of the EERs while the third line provides the resulting average number of support vectors for each client model.

We thus propose here an alternative to taking the average over all frames. We consider, for each frame of sequence $\mathbf{X}_i$, the similarity measure of the closest corresponding frame in sequence $\mathbf{X}_j$. We thus propose to take a symmetric Max operator of the form:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{T_i} \sum_{t_i} \max_{t_j} k(\mathbf{x}_{t_i}, \mathbf{x}_{t_j}) + \frac{1}{T_j} \sum_{t_j} \max_{t_i} k(\mathbf{x}_{t_i}, \mathbf{x}_{t_j}).$$

The main idea is that, instead of comparing frames coming from different acoustic events, we compare close frames only. Unfortunately, the resulting function does not satisfy Mercer’s conditions anymore. In practice however, even if a function does no satisfy Mercer’s conditions, one might still find that a given training set results in a positive semi-definite Hessian in which case the training will converge perfectly well [4]. The empirical results provided here and in Section 5 show that the Max operator based kernel [1] gives good results.

Figure 2 and Table 2 show that the Max approach outperforms the standard one on the development set of Polyvar. The RBF kernel gives similar result to the polynomial kernel when the Max operator is used. It is interesting to note that now the optimal value is $p = 1$. This is probably because the Max operator is more appropriate. And this value is reasonable because the input space dimension of each sequence $\mathbf{X}$ is given by $T_i T_j d$ which is already huge compared to the number of examples. Thus we need very small capacity, and the plain dot product seems sufficient.

5 Experimental Results

Figure 3 presents the final performance on the test set of the Polyvar database. Only the best systems (according

[1] Note that in the following we will continue to call such a function a kernel even if it does not satisfy Mercer’s conditions, as it is often done in the literature (see for instance [4])

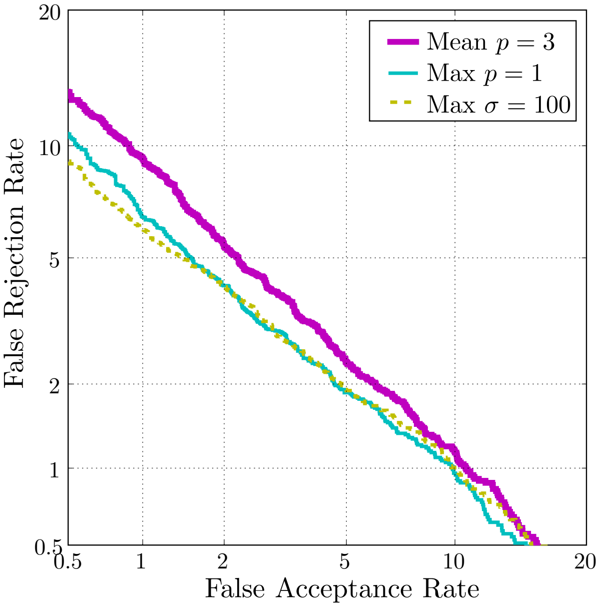

Figure 2. DET curves on the development set of the Polyvar database for Mean and Max operators for polynomial and RBF kernels.

Table 2. Results on the development set of the Polyvar database for Mean and Max operators for polynomial and RBF kernels.

	Mean	Max	Max
	$p = 3$	$p = 1$	$\sigma = 100$
EER [%]	3.46	2.99	2.95
95% Confidence	±0.28	±0.26	±0.26
# Support Vectors	87	73	99

to the development set) for Max and Mean operator based kernels are presented. Complementary results are presented in Table 3. The figure is composed of two graphs. The first one represents an EPC providing the HTER as a function of the parameter α of a convex combination of FAR and FRR, as given by equation (5), which was used to set the threshold on a development set. Thus, the lower the curve, the better the performance. The second part provides the confidence level for each value of α. The higher the curve, the more confident we can be on the statistical significance of the difference in performance between the two compared models.

The first conclusion is that the SVM based systems outperform the GMM based system. Furthermore, the Max approach significantly outperforms GMMs for all values of α with a confidence level greater than 99% most of the time. The Max approach also outperforms most of the time the Mean based system (equivalent to the "Campbell" approach for polynomial kernels) with a confidence level greater than 95%. The solution is also sparser in terms of number of support vectors. The Max RBF kernel gives similar results to the Max polynomial kernel. It is also interesting to note that the optimal degree for the Max polynomial kernel is equal to 1.

6 Conclusions

We have proposed a new method to use SVMs for speaker verification. It allows the use of all kinds of kernels, generalizes the explicit polynomial approach and outperforms SVM based state-of-the-art approaches for the tested database.

We have also proposed a new Max operator instead of averaging the kernel values over all pairs of frames. It makes more sense and outperforms the standard approach. Unfortunately it does not satisfy the Mercer conditions but still converges very well for the studied databases.

The main drawback of our proposed method is the large complexity for long sequences. This can probably be alleviated using some clustering techniques.

Acknowledgments

This research has been partially carried out in the framework of the Swiss NCCR project (IM)2. It was also supported in part by the IST Program of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, funded in part by the Swiss OFES. All experiments were performed using the *Torch* package [8]. We would also like to thank Jérôme Louradour and David Barber for fruitful discussions.

References

[1] S. Bengio and J. Mariéthoz. A statistical significance test for person authentication. In *Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop*, pages 237–240, 2004.

[2] S. Bengio, J. Mariéthoz, and M. Keller. The expected performance curve. In *International Conference on Machine Learning, ICML, Workshop on ROC Analysis in Machine Learning*, 2005.

[3] S. Boughorbel, J. P. Tarel, and F. Fleuret. Non-mercer kernel for svm object recognition. In *British Machine Vision Conference*, 2004.

[4] C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.

[5] W. Campbell. Generalized linear discriminant sequence kernels for speaker recognition. In *Proc IEEE International*

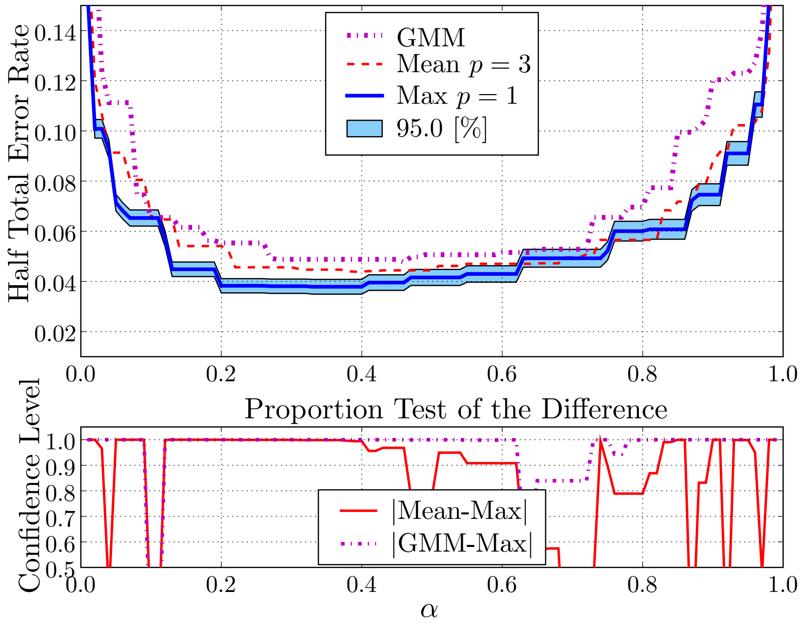

Figure 3. EPC curves on the test set of the Polyvar database for best Mean and Max operators for polynomial and RBF kernels.

Table 3. Results on the test set of the Polyvar database for Mean and Max operators for polynomial and RBF kernels.

	GMM	Mean	Mean	Max	Max
	$N = 100$	$\sigma = 6$	$p = 3$	$p = 1$	$\sigma = 100$
		$C = \infty$	$C = \infty$	$C = \infty$	$C = \infty$
HTER [%]	4.9	4.59	4.47	3.9	4.21
95% Confidence	±0.34	±0.33	±0.32	±0.31	±0.32
# Support Vectors	-	62	87	73	99

Conference on Audio Speech and Signal Processing, pages 161–164, 2002.

[6] W. Campbell, J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carrasquillo. Support vector machines for speaker and language recognition. *Computer Speech and Language*, 2005.

[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] R. Collobert, S. Bengio, and J. Mariéthoz. Torch: a modular machine learning software library. Technical Report IDIAP-RR 02-46, IDIAP, 2002.

[9] J. L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observation of markov chains. In *IEEE Transactions on Speech Audio Processing*, volume 2, pages 291–298, April 1994.

[10] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing*, 11:487–493, 1998.

[11] I. Magrin-Chagnolleau, G. Gravier, and R. Blouet. Overview of the 2000-2001 ELISA consortium research activities. In *A Speaker Odyssey*, pages 67–72, June 2001.

[12] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proceedings of Eurospeech'97, Rhodes, Greece*, pages 1895–1898, 1997.

[13] L. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*. Prentice All, first edition, 1993.

[14] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1–3), 2000.

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

[16] V. Wan and S. Renals. Speaker verification using sequence discriminant support vector machines. *IEEE Transactions on Speech and Audio Processing*, 13(2):203–210, 2005.