A Kernel Trick For Sequences Applied to Text-Independent Speaker Verification Systems

Johnny Mariéthoz * Samy Bengio

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

Abstract

This paper present a principled SVM based speaker verification system. We propose a new framework and a new sequence kernel that can make use of any Mercer kernel at the frame level. An extension of the sequence kernel based on the Max operator is also proposed. The new system is compared to state-of-the-art GMM and other SVM based systems found in the literature on the Banca and Polyvar databases. The new system outperforms, most of the time, the other systems, statistically significantly. Finally, the new proposed framework clarifies previous SVM based systems and suggests interesting future research directions.

Key words: support vector machines, Gaussian mixture models, sequence kernel, text-independent speaker verification

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) (Reynolds et al., 2000), 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

* Corresponding author. Tel. +41 27 721 77 44, fax: +41 27 721 77 12.

A Posteriori (MAP) (Gauvain and Lee, 1994) 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 (Campbell, 2002) has obtained good results during the NIST 2003 evaluation (Campbell et al., 2005), 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.

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 framework, the measures and the databases used in the experimental part. In Section 3, we provide a brief introduction to SVMs. The new proposed approach is then presented in Section 4, and is compared to similar approaches found in the literature. Some improvements are also proposed at the end of this section. Results on two speaker verification tasks are then 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

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

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. To select the best function, we need to define a set of functions $f_\Theta()$ parameterized by Θ and make use of a set of sentence examples called the "training set":

$$Tr = \left\{ (\mathbf{X}_l, y_l) | \mathbf{X}_l \in \mathbb{R}^{d \times T_l}, y_l \in \{-1, 1\} \right\}_{l=1..L}$$

where $\mathbf{X}_l$ is an input sequence of T_l frames of d dimensions with a corresponding target y_l equal to 1 for a true client sequence and -1 otherwise, L is the total number of sequences in the training set. We are searching for parameters Θ of a parametric function $f_\Theta : \mathbb{R}^{d \times T_l} \mapsto \mathbb{R}$ that minimize a loss function $Q()$ which returns low values when $f_\Theta(\mathbf{X}_l)$ is near y_l and high values otherwise:

$$\Theta^* = \arg\min_{\Theta} \sum_{(\mathbf{X}_l, y_l) \in Tr} Q(f_\Theta(\mathbf{X}_l), y_l).$$

The loss function usually accounts for the training errors as well as some constraints that are known to yield better generalization performance (for example maximizing the margin, as is the case for SVMs). Note that the overall goal is not to obtain zero error on Tr but rather on unseen examples drawn from the same probability distribution as those of Tr.

Depending on whether the underlying $f_\Theta()$ is based on probabilities or not, two frameworks can be considered and are presented in this section.

2.1 Statistical Framework

Most state-of-the-art speaker verification systems are based on statistical models. In that framework, the system has to decide whether a sentence $\mathbf{X}$ was pronounced by a speaker S_i or by any other person $\bar{S}_i$. It accepts a claimed speaker as a *client* only if:

$$P(S_i|\mathbf{X}) > P(\bar{S}_i|\mathbf{X}). \tag{2}$$

Using Bayes theorem, we can rewrite (2) as follows:

$$\frac{p(\mathbf{X}|S_i)}{p(\mathbf{X}|\bar{S}_i)} > \frac{P(\bar{S}_i)}{P(S_i)} = \Delta_{S_i} \approx \Delta \tag{3}$$

where Δ_{S_i} represents the ratio of the prior probabilities of being or not being the client. In this paper this threshold will be replaced by a client independent decision threshold Δ. The left part of equation (3) is the parametric function $f_\Theta()$ in (1), but as we use two probability estimators, $f_\Theta()$ is decomposed as follows:

$$f_\Theta(\mathbf{X}) = \frac{f_{\Theta_+}(\mathbf{X})}{f_{\Theta_-}(\mathbf{X})} = \frac{p(\mathbf{X}|S_i)}{p(\mathbf{X}|\bar{S}_i)}$$

where $f_{\Theta_+}()$ is a function estimated with the positive examples and $f_{\Theta_-}()$ is a function estimated with the negative examples. The loss function used to train $f_{\Theta_-}()$ is the negative log likelihood and can be express as:

$$\Theta_-^* = \arg\min_{\Theta_-} \sum_{(\mathbf{X}_l)\in Tr_-} -\log p(\mathbf{X}_l|\Theta_-)$$

where Tr_- is the subset of examples of Tr where $y_l = -1$. As generally few positive examples are available, the loss function used to train $f_{\Theta_+}()$ is based on a Maximum A Posteriori (MAP) (Gauvain and Lee, 1994) adaptation scheme and can be written as follows:

$$\Theta_+^* = \arg\min_{\Theta_+} \sum_{(\mathbf{X}_l)\in Tr_+} -\log\Big(P(\mathbf{X}_l|\Theta_+)P(\Theta_+)\Big)$$

where Tr_+ is the subset of examples of Tr where $y_l = 1$. This MAP approach puts some prior about the distribution of Θ_+ in order to constrain them to some reasonable values.

We thus need to create an impostor model of $p(\mathbf{X}|\bar{S}_i)$, called *world* or *background model* if it is common for all speakers S_i, as well as a *client* model $p(\mathbf{X}|S_i)$ for every potential speaker. The two generative models are often estimated by Gaussian Mixture Models, which transforms (3) as follows:

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

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, $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\sigma})$ is the density of $\mathbf{x}$ according to a Normal distribution of mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$, $\Theta_+ = \{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, w_n\}$ 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 $\frac{1}{T}$ does not follow from (3) and is an empirical normalization factor added to be independent of the length of the sentence.

In the context of GMM based speaker verification systems, MAP adaptation broadly translates into forcing Θ_+ to be near Θ_- as the latter are assumed to be better estimated than the former. See for instance (Reynolds et al., 2000) for a practical implementation.

2.2 A Score Based Framework

If instead of relying on models generating probabilities, we want to use discriminative models such as SVMs, as described in the remainder of this paper, the framework described at the beginning of this section can be applied directly and no probabilistic interpretation need be given to $f_\Theta()$. Section 3 describes in detail the parametric form of function $f_\Theta()$ and the lost function $Q()$ used by SVMs.

2.3 Measures

Instead of the usual classification error rate often found in the machine learning literature, the speaker verification community uses a weighted version of it, as follows.

One can consider two kinds of errors. Rejecting a genuine client (*False Rejection*, FR) or accepting an impostor (*False Acceptance*, FA). All measures used in this paper are based on the corresponding error rates: the *False Acceptance Rate* (FAR), which is the number of FAs divided by the number of client accesses and the *False Rejection Rate* (FRR) which is the number of FRs divided by the number of impostor accesses.

Unfortunately, in the literature, most of the results are reported through "a posteriori" measures in the sense that the decision threshold Δ is selected to optimize a given criterion on the test set. In order to obtain unbiased results, one should rely instead on "a priori" measures, where the decision threshold Δ is first selected on a separate development set, and then applied to the test set.

Often used a posteriori measures include Equal Error Rates (where the threshold Δ is chosen such that (FAR=FRR) and DET curves (Martin et al., 1997) which present FRR as a function of FAR by varying Δ. They are normally used to tune and analyze systems. A priori measures, on the other hand, include Half Total Error Rate (HTER) $\frac{(\mathrm{FAR}_\Delta+\mathrm{FRR}_\Delta)}{2}$ and the Expected Performance Curves (EPC) (Bengio et al., 2005) 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:

$$\Delta^* = \arg\min_{\Delta}\left(\alpha \text{FAR}_{\Delta} + (1-\alpha)\text{FRR}_{\Delta}\right). \tag{4}$$

Finally, in this paper, we have also added for both curves and values a confidence interval of 95% using a modified version of the standard proportion test (Bengio and Mariéthoz, 2004).

2.4 Experimental Setup and Databases

In order to compare the systems presented here, two databases were used. The *Polyvar* telephone database (Chollet et al., 1996), 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, we kept only the impostor accesses containing the same word as the one chosen by the true client. The development set of this database is used to analyze the systems presented in this paper.

The English part of the *Banca* database (Bailly-Baillière et al., 2003) contains a development and a test set of 26 clients each (13 men and 13 women) as well as another population of 60 speakers (30 females and 30 males) used to train the world model. This database contains three recording conditions defined as controlled, degraded and adverse and is provided with 7 different protocols. We have chosen to use the protocol which we consider the most realistic [1]: only one controlled session is available to train the client model and 546 balanced test accesses in controlled, degraded and adverse conditions were used per population. In this paper, Banca is only used in the final comparison.

Table 1 shows a summary of the two Banca and Polyvar databases.

For both databases, each sentence was parameterized using 24 *Linear Filter Cepstral Coefficients* (LFCC) (Rabiner and Juang, 1993) 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 (Magrin-Chagnolleau et al., 2001).

[1] This corresponds to protocol P as defined in the Banca protocol

Table 1
Some statistics for the two Banca and Polyvar databases.

	Banca	Polyvar
# of client models on the dev set	26	323
# of client models on the test set	26	323
# of training impostor examples	60	592
# training client examples per model to train	1	5
# testing client examples for each set	234	6000
# testing impostor examples for each set	312	12000
# frames per example on average	1000	80

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 (Reynolds et al., 2000). Only the mean parameters of the client model were adapted using the following update rule:

$$\mu_n = \alpha\mu_n^{ML} + (1-\alpha)\bar{\mu}_n$$

where n is the Gaussian index, μ_n^{ML} the mean parameter vector estimated using the Maximum Likelihood criterion over the client data, $\bar{\mu}_n$ the mean parameter vector of the world model and α the MAP adaptation factor that represents the faith we have in the client data.

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 of each corresponding database and are given in Table 2.

Table 2
Summary of the hyper-parameters for GMM based systems.

Database	Number of Gaussians (N)	MAP Factor (α)	Variance Flooring Factor in [%]
Polyvar	100	0.2	0.1
Banca	200	0.5	0.6

3 Support Vector Machines

Support Vector Machines (SVMs), as proposed by Vapnik (1995), are more and more often used in machine learning applications such as text classification and vision (Joachims, 2002; Pontil and Verri, 1998). They have also been used successfully for regression and multi-class classification problem (Kwok, 1998). In the context of two-class classification problems, the underlying decision function is:

$$f_\Theta(\mathbf{x}) = b + \mathbf{w} \cdot \Phi(\mathbf{x}) \tag{5}$$

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.

Solving the SVM problem is equivalent to minimizing the following criterion:

$$(\mathbf{w}^*, b^*) = \arg\min_{(\mathbf{w},b)} \frac{\| \mathbf{w} \|^2}{2} + C \sum_{l=1}^{L} \xi_l \tag{6}$$

under the constraints:

$$y_l(\mathbf{w}\mathbf{x}_l + b) \geq 1 - \xi_l \quad \forall_l$$

$$\xi_l \geq 0 \quad \forall_l$$

where L is the number of training examples, y_l is the target class label in $\{-1, 1\}$ corresponding to $\mathbf{x}_l$, C is a parameter that trades off the minimization of classification errors (represented by ξ_l) and the maximization of the margin, known to possess very good generalization properties. Maximizing the margin is very important in the context of speaker verification, since in most cases very few positive examples are available, and the problem is often easily separable.

It can be shown that solving (6) enables the decision function to be expressed as a hyper-plane defined by a linear combination of training examples in the feature space $\Phi()$. We can thus express (5) as follows:

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

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 potentially 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,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right) \tag{7}$$

where σ is a hyper-parameter than can be used to tune the capacity (which represents the size of the set of possible functions $f_\Theta(\mathbf{x})$, as explained by Vapnik, 1995) of the model, and the polynomial kernel,

$$k(\mathbf{x}_i, \mathbf{x}_j) = (a\mathbf{x}_i \cdot \mathbf{x}_j + b)^p \tag{8}$$

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 (Wan and Renals, 2005; Campbell et al., 2005). 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 length 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 little 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 (Jaakkola and Haussler, 1998) 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 (Campbell et al., 2005; Wan and Renals, 2003) 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.

4 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:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{T_i T_j} \sum_{t_i=1}^{T_i} \sum_{t_j=1}^{T_j} k(\mathbf{x}_{t_i}, \mathbf{x}_{t_j}) \tag{9}$$

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 (Boughorbel et al., 2004). It has the advantage that all forms of kernels can be used for $k()$ and the resulting kernel $K()$ respects all Mercer conditions (Burges, 1998) which make sure that for all possible training sets the resulting Hessian is semi-positive; these conditions make the problem convex. Two forms of kernels $k()$ are used in this paper: an RBF kernel (7) and a polynomial kernel (8). For the latter, we fixed a and b to $p!^{-\frac{1}{2}}$ which makes the maximum value of the polynomial coefficients equal to one 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.

4.1 Comparison with Campbell's Polynomial Approach

Campbell (2002) 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:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \Phi(\mathbf{X}_i)\mathbf{\Gamma}^{-1}\Phi(\mathbf{X}_j) \tag{10}$$

where $\mathbf{\Gamma}$ is a matrix derived by the metric of the feature space induced by $\Phi()$. This matrix is usually a diagonal approximation $\boldsymbol{\gamma}$ 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

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

where $\phi'()$ is the normalized version of $\phi()$, and can thus rewrite (10) 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:

$$\phi'_{k(r_1,r_2,...,r_d)}(\mathbf{x}_t) = \frac{1}{\sqrt{\gamma_k}}x_1^{r_1}x_2^{r_2}...x_d^{r_d} \tag{11}$$

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 $\boldsymbol{\gamma}$ 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 (Vapnik, 1995). Empirically, we have seen that for various databases the optimal value for C in equation (6) becomes ∞. This is in general due to the use of an incorrect cost function. As often in speaker verification, only few positive examples (even only one) are available. Furthermore, the ratio between the number of positive and negative examples is very different between the training and the test accesses. As C cannot be used to tune the capacity of the system (since it always end up being ∞), we can rely only on the hyper-parameters of the chosen kernel. 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, 7 140 and 66 045. 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 (9) as follows:

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

Let us define $k(\mathbf{x}_i, \mathbf{x}_j)$ of (9) 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 (Burges, 1998) in a similar way to (11). Each value of the extended vector is thus given by:

$$\phi_{k(r_1, r_2, \ldots, r_d)}(\mathbf{x}_t) = \sqrt{c_k} x_1^{r_1} x_2^{r_2} \ldots x_d^{r_d}, \quad \sum_{i=1}^{d} r_i = p, \quad r_i \geq 0 \tag{12}$$

$$\text{where} \quad c_k = \frac{p!}{r_1! r_2! \ldots r_{d+1}!}, \quad k \in \{1, \ldots, K\}$$

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

When we compare equations (12) and (11) 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. We compared in Figure 1 the coefficient values for each term in the proposed approach with the normalization vector obtained by the explicit method as estimated on Banca and Polyvar using a polynomial expansion of degree 3. As can be seen, they look very similar: all of them show high (resp. low) values at the same time. In fact, the performance obtained on the development set of Polyvar are very similar, as shown by the DET curves given in Figure 2 and Equal Error Rates provided in Table 3. Figure 2 and Table 3 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.

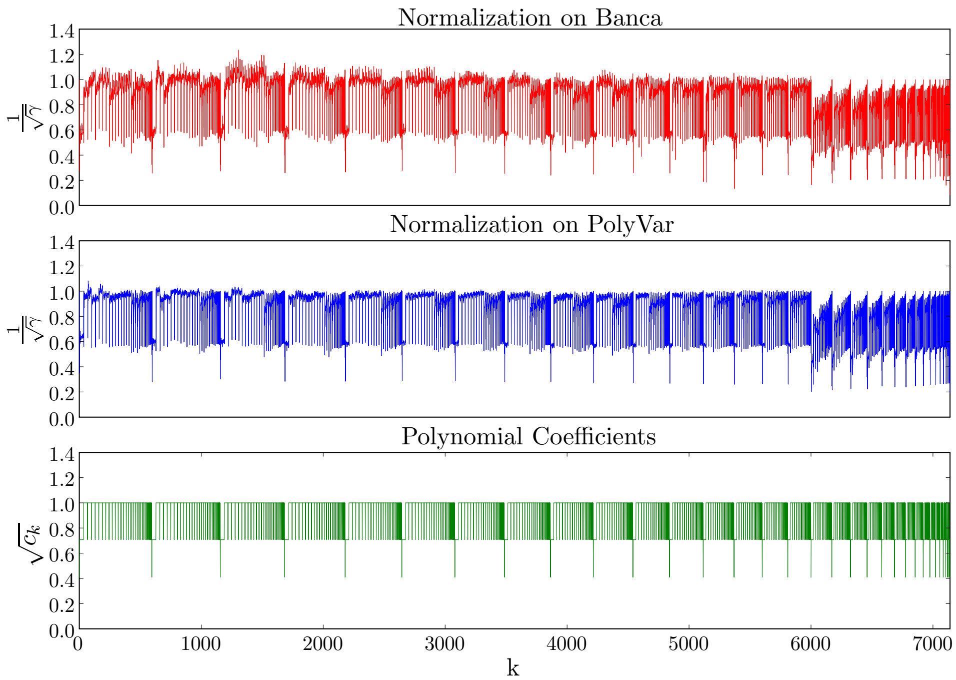

Fig. 1. Coefficient values of polynomial terms, as computed on Banca and Polyvar, compared to the c_k polynomial coefficients.

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:

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

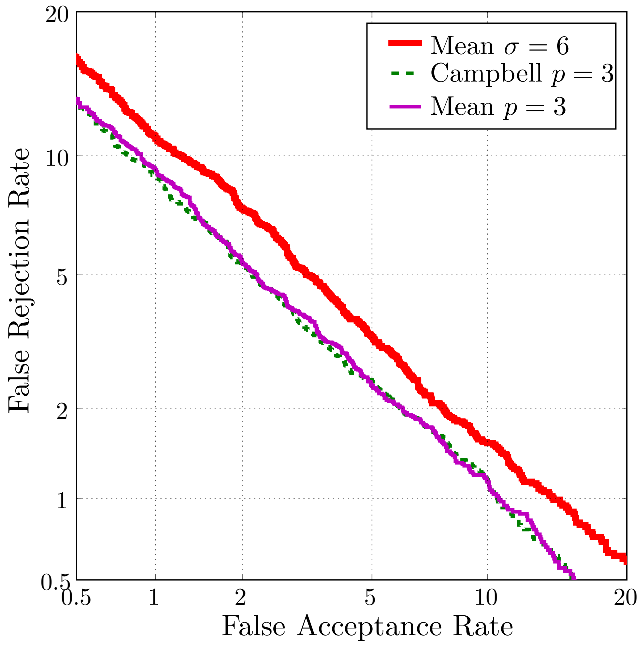

Fig. 2. DET curves on the development set of the Polyvar database comparing the explicit polynomial expansion (noted as "Campbell $p = 3$ in the legend), the principled polynomial kernel (noted "Mean $p = 3$") and an RBF kernel using the Mean operator (noted "Mean $\sigma = 6$").

Table 3
Comparison of EERs (the lower the better) on the development set of the Polyvar database between the explicit polynomial expansion (noted "Campbell") and two principled kernels (polynomial and RBF) applying the mean operator over all pairs of frames (noted respectively "Mean $p = 3$" and "Mean $\sigma = 3$"). 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.

	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

Long sequences are thus very costly. This is not a problem for databases such as Polyvar and Banca, 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:

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

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.

4.2 Max Approach

In equation (9), 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 (9), 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 (9). In that case, one can easily obtained $K(\mathbf{X}_i, \mathbf{X}_j) \geq K(\mathbf{X}_i, \mathbf{X}_i)$.

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 the 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 (Burges, 1998). The empirical results provided here and in Section 5 show that the Max operator based kernel [2] gives good results on at least two speaker verification databases.

Figure 3 and Table 4 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

[2] 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 (Burges, 1998))

compared to the number of examples. Thus we need very small capacity, and the plain dot product seems sufficient.

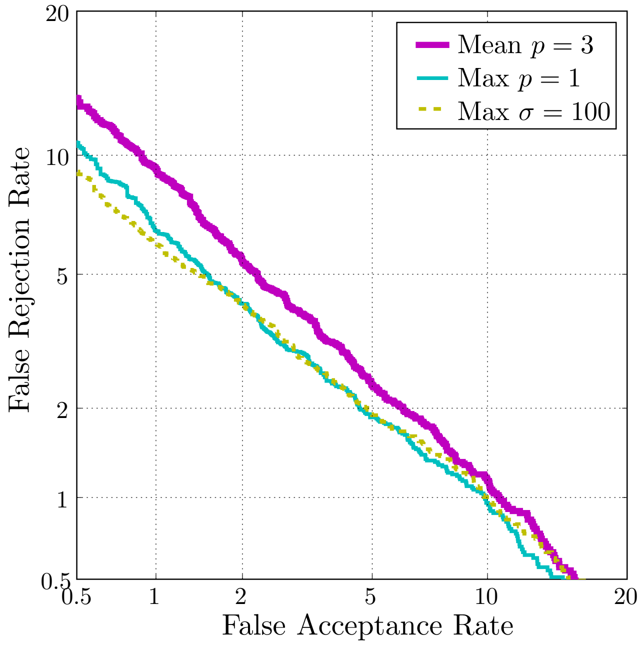

Fig. 3. DET curves on the development set of the Polyvar database for Mean and Max operators for polynomial (noted "Max $p = 1$") and RBF kernels (noted "Max $\sigma = 100$").

Table 4
Results on the development set of the Polyvar database for Mean and Max operators for polynomial (noted "Max $p = 1$") and RBF (noted "Max $\sigma = 100$") kernels.

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

5 Experimental Results

We provide in this section performance results comparing the various speaker verification systems over the test sets of both the Polyvar and the Banca databases.

Figure 4 presents the final performance on the test set of the Polyvar database. Only the best systems (according to the development set) for Max and Mean operator based kernels are presented. Complementary results are presented in Table 5. 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 (4), 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.

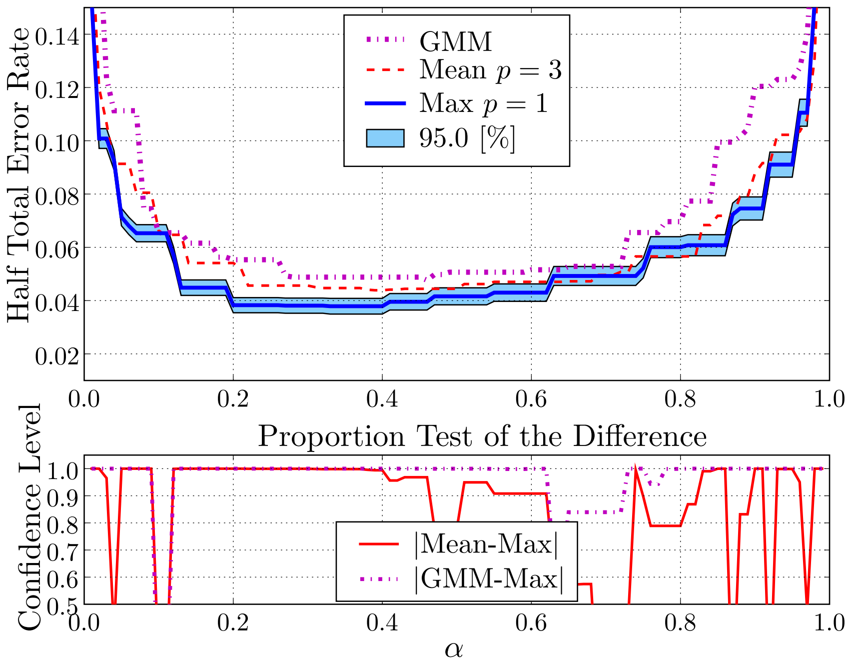

Fig. 4. EPC curves on the test set of the Polyvar database for GMM and best Mean and Max operators for polynomial and RBF kernels.

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.

Table 5
Results on the test set of the Polyvar database for GMM, Mean operator for polynomial (noted "Mean $p = 3$") and RBF (noted "Mean $\sigma = 6$) kernels and Max operator for polynomial (noted "Max $p = 1$") and RBF (noted "Max $\sigma = 100$") kernels.

	GMM $N = 100$	Mean $\sigma = 6$ $C = \infty$	Mean $p = 3$ $C = \infty$	Max $p = 1$ $C = \infty$	Max $\sigma = 100$ $C = \infty$
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

5.2 Banca

Figure 5 and Table 6 present the final performance of several systems on the Banca database. Once again, only the best systems for Max and Mean operators are presented.

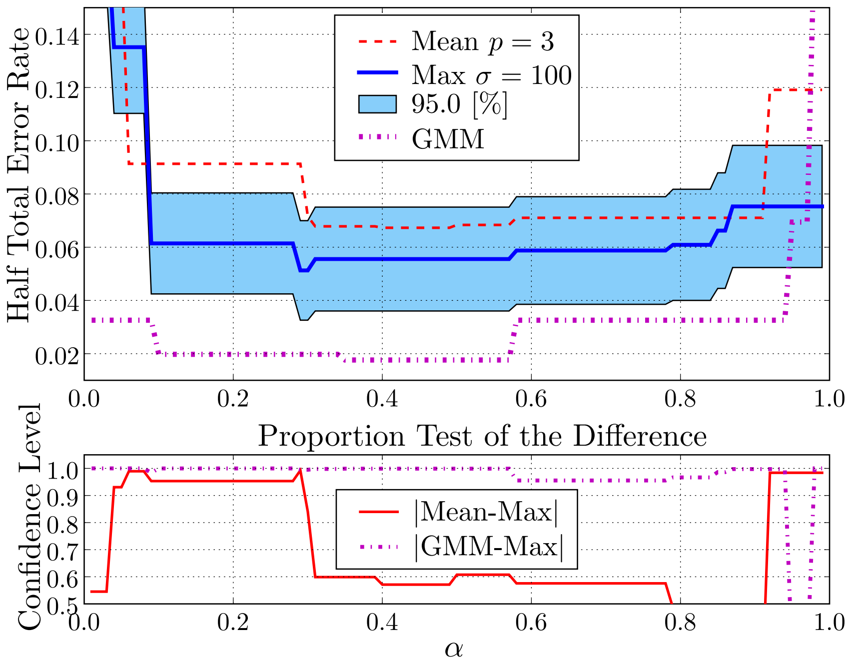

Fig. 5. EPC curves on test set of the Banca database for GMM and best Mean and Max operator for polynomial and RBF kernels.

The first conclusion is that, for this database, the GMM based system outperforms all the SVM based systems. The particularity of this database is the unmatched conditions. Only one "controlled" training session per speaker is available and all conditions are used during the test. SVMs might be less robust than GMMs for unmatched conditions.

Table 6
Results on test set of the Banca database for GMM, Mean operator polynomial (noted "Mean $p = 3$") and RBF (noted "Mean $\sigma = 8$") kernels and Max operator for polynomial (noted "Max $p = 1$") and RBF (noted "Max $\sigma = 200$") kernels.

	GMM $N = 200$	Mean $\sigma = 8$ $C = \infty$	Mean $p = 3$ $C = \infty$	Max $p = 1$ $C = \infty$	Max $\sigma = 200$ $C = 100$
HTER [%]	2.72	8.71	6.57	6.57	5.61
95% Confidence	±1.42	±2.4	±2.1	±2.1	±1.94
# Support Vectors	-	18	27	42	9

The Max approach outperforms most of the time the Mean system but the confidence level of the difference is low. This database is unfortunately too small to gives statistically significant results. However, it is interesting to note once again that the Max operator solution is sparser than the Mean operator solution. The optimal C value is not ∞ for the Max RBF kernel so in some cases it can still be interesting to tune this parameter. Empirically most of the time, the optimal value of the C parameter remains ∞. It is probably due to the SVM criterion: it has been designed to minimize the classification error rate, which is not optimal in our case and should be modified in order to deal with highly unbalanced data. This problem has already been investigated recently by Grandvalet et al. (2005).

Note also that, contrary to the Polyvar database, the optimal kernel is now the RBF kernel. This shows that it is important to provide an SVM approach where the kernel can be chosen according to the database, which was not the case in (Campbell, 2002).

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 two tested databases.

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.

We have also shown that the capacity parameter C influences the results using the Max operator which was not the case with the approach proposed by Campbell (2002). We still need to understand better how to modify the SVM criterion for unbalanced data as often found in speaker verification tasks. A big indicator of the problem is that using a polynomial kernel with a Max operator, the optimal degree is equal to 1. Thus we hope to be able to reduce the capacity by tuning the C value.

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 (Collobert et al., 2002). We would also like to thank Jérôme Louradour and David Barber for fruitful discussions.

References

Bailly-Baillière, E., Bengio, S., Bimbot, F., Hamouz, M., Kittler, J., Mariéthoz, J., Matas, J., Messer, K., Popovici, V., Porée, F., Ruiz, B., Thiran, J.-P., 2003. The BANCA database and evaluation protocol. In: 4th International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA. Springer-Verlag, pp. 625–638.

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

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

Boughorbel, S., Tarel, J. P., Fleuret, F., 2004. Non-mercer kernel for svm object recognition. In: British Machine Vision Conference.

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

Campbell, W., 2002. Generalized linear discriminant sequence kernels for speaker recognition. In: Proc IEEE International Conference on Audio Speech and Signal Processing. pp. 161–164.

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

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

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

Gauvain, J. L., Lee, C.-H., April 1994. Maximum a posteriori estimation for multivariate gaussian mixture observation of markov chains. In: IEEE Transactions on Speech Audio Processing. Vol. 2. pp. 291–298.

Grandvalet, Y., Mariéthoz, J., Bengio, S., 2005. A probabilistic interpretation of svms with an application to unbalanced classification. In: Advances in Neural Information Processing Systems, NIPS 15. IDIAP-RR 05-26.

Jaakkola, T., Haussler, D., 1998. Exploiting generative models in discriminative classifiers. Advances in Neural Information Processing 11, 487–493.

Joachims, T., 2002. Learning to Classify Text using Support Vector Machines. Kluwer Academic Publishers, Dordrecht, NL.

Kwok, J. T.-Y., 1998. Support vector mixture for classification and regression problems. In: 14th International Conf. on Pattern Recognition.

Magrin-Chagnolleau, I., Gravier, G., Blouet, R., June 2001. Overview of the 2000-2001 ELISA consortium research activities. In: A Speaker Odyssey. pp. 67–72.

Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M., 1997. The DET curve in assessment of detection task performance. In: Proceedings of Eurospeech'97, Rhodes, Greece. pp. 1895–1898.

Pontil, M., Verri, A., 1998. Support vector machines for 3-d object recognition. IEEE Transaction PAMI 20, 637–646.

Rabiner, L., Juang, B.-H., 1993. Fundamentals of speech recognition, 1st Edition. Prentice All.

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

Vapnik, V. N., 1995. The nature of statistical learning theory, 2nd Edition. Springer.

Wan, V., Renals, S., 2003. Support vector machine speaker verification methodology. In: IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP. pp. 221–224.

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