7 Sequence Kernel Based Speaker Verification

In the previous chapters, we proposed several new discriminant approaches for text-independent speaker verification, including the use of SVMs operating on some informations extracted from GMMs. In this chapter, we consider the use of SVMs with sequences of feature vectors as inputs.

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, called GLDS kernel, described in Chapter 2 and 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 chapter a more principled SVM based speaker verification system that can make use of the kernel trick.

We also present some improvements of the new proposed kernel in order to enhance the HTER performance, but also to make this new kernel usable for long sequences.

The outline of this chapter goes as follows. The new proposed approach is presented in Section 7.1, and is compared to similar approaches found in the literature. A new Max operator based kernel is described in Section 7.2. A smoothing version of the new kernel is then proposed in Section 7.5. Finally, in

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

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

order to reduce the complexity of the Max operator based kernel, we describe in Section 7.6 a solution using clustering techniques.

7.1 Mean Operator Kernel

SVMs have been designed to work on any type of data, as long as a kernel $K(\mathbf{X}_i, \mathbf{X}_j)$ comparing two examples $\mathbf{X}_i$ and $\mathbf{X}_j$ is defined. One specificity 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:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{T_i T_j} \sum_{t_i=1}^{T_i} \sum_{t_j=1}^{T_j} k(\mathbf{x}_{t_i}, \mathbf{x}_{t_j}) \tag{7.1}$$

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 kernels 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 Gram matrix is positive semidefinite which makes the problem convex. Given a set V of m vectors (points in $\mathbb{R}^n$), the Gram matrix G is the matrix of all possible inner products of V (definition taken from http://mathworld.wolfram.com). Two forms of kernels $k()$ are used in this thesis: an RBF kernel (2.14) and a polynomial kernel (2.15). For the polynomial kernel or order p, we fixed a and b to $p!^{-\frac{1}{2}p}$ in order to avoid overflow 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.

Comparison with GLDS Kernel Approach

Although the GLDS kernel based approach yielded good performance during the NIST campaigns, it has some drawbacks. First no kernel trick can be

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

REF C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):1–47, 1998.

applied: it seems not possible to include the normalization vector $\frac{1}{\sqrt{\psi_n}}$ in (2.32) 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 of the model (Vapnik, 2000). Empirically, we have seen that for various databases the optimal value for C in equation (2.9) which governs the tradeoff between a large margin and training errors, 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 GLDS based polynomial kernel 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, when considering 33 dimensional input vectors, are 33, 595, 7 140 and 66 045. It is then difficult to correctly set the capacity. Moreover, as the best value is empirically $p = 3$ for the considered databases, the dimension seems quite huge if we consider that a few hundred examples only are used for training.

Let us now consider again (7.1) and see how it relates to the GLDS approach. Let us start by rewriting (7.1) 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 (7.1) 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 the GLDS expansion, given in (2.32). Each value of the extended vector is thus given by:

$$\phi_{n(r_1,r_2,\ldots,r_d)}(\mathbf{x}_t) = \sqrt{c_n} x_1^{r_1} x_2^{r_2} \ldots x_d^{r_d}, \quad \sum_{i=1}^{d} r_i = p, \quad r_i \geq 0 \tag{7.2}$$

REF V. N. Vapnik. *The nature of statistical learning theory.* Springer, second edition, 2000.

REF C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):1–47, 1998.

$$\text{where} \quad c_n = \frac{p!}{r_1!r_2!...r_{d+1}!}, \quad n \in \{1, ..., N_f\}$$

and each input frame of dimension d is augmented by a new coefficient equal to 1 and N_f is the dimension of the expanded vector.

When we compare equations (7.2) and (2.32), the difference only lies in the polynomial coefficients: each term is multiplied by a coefficient $\sqrt{c_n}$ in (7.2) while the explicit expansion needs a normalization factor $\frac{1}{\sqrt{\psi_n}}$ that disables the kernel trick. We compared in Figure 7.1 the coefficient values for each term in (7.2) with the normalization vector obtained by the explicit GLDS method as estimated on Banca and Polyvar using a polynomial expansion of degree 3. As can be seen, 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 7.2 and Equal Error Rates provided in Table 7.1. Figure 7.2 and Table 7.1 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.

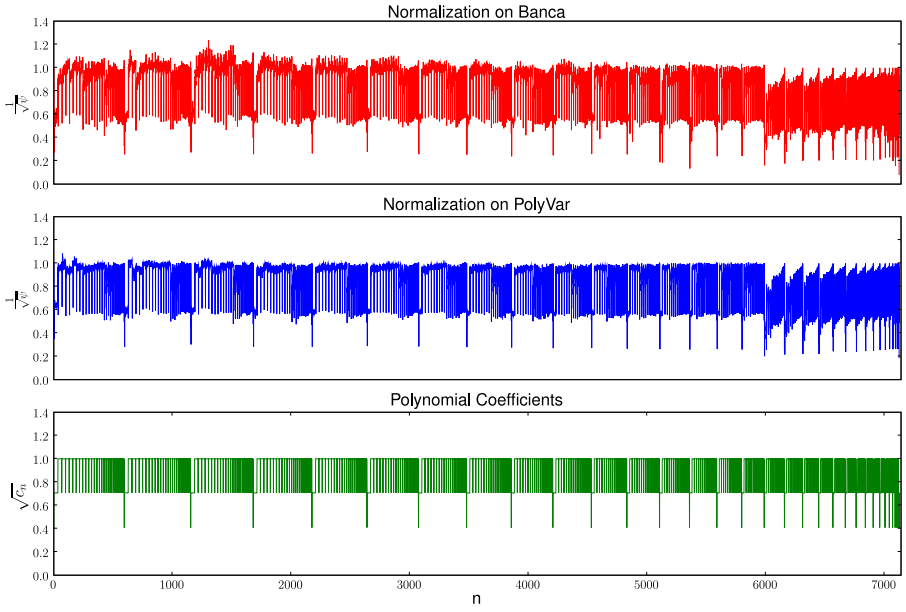


Figure 7.1. Coefficient values $\frac{1}{\sqrt{\psi_n}}$ of polynomial terms in the GLDS kernel, as computed on Banca and Polyvar, compared to the $\sqrt{c_n}$ polynomial coefficients of equation (7.2).

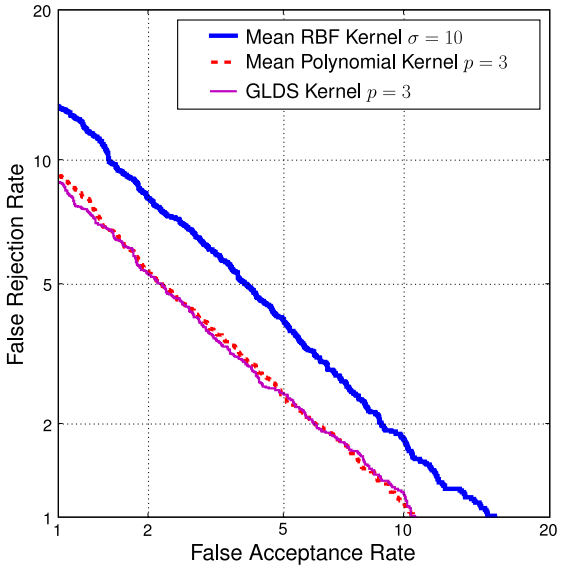


Figure 7.2. DET curves on the development set of the Polyvar database comparing the explicit polynomial expansion (GLDS based kernel), the principled polynomial kernel and an RBF kernel (using the Mean operator).

Table 7.1. Comparison of EERs (the lower the better) on the development set of the Polyvar database between the explicit polynomial expansion and a principled polynomial kernel applying the Mean operator over all pairs of frames. 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.

	GLDS $p = 3$	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

The drawback of (7.2), however, is the computational complexity for long sequences. If S is the number of speakers, NP the number of positive examples per speaker, NN the number of negative examples, and T the average number of frames per example, then the training time complexity is given by:

$$O(S\,T^2(\mathrm{NP}^2 + \mathrm{NN}\cdot\mathrm{NP}) + T^2\,\mathrm{NN})$$

while the equivalent complexity for GLDS kernel would be the same except that all T^2 would be replaced by T, hence becoming linear in the length of the sequence instead of quadratic for (7.2).

Long sequences are thus very costly. This is not a problem for databases such as Polyvar and Banca, especially, because 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 $O(N_{sv}T^2)$ where N_{sv} is the number of support vectors. Even in the test phase, computing scores for long sequences can be too time consuming. This problem can probably be addressed using clustering techniques and is treated in the following.

7.2 Max Operator Kernel

In equation (7.1), 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 (7.1), since we take the average. To illustrate this, let us create a sequence $\mathbf{X}_j$ containing 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 (7.1). In that case, one can easily obtain $K(\mathbf{X}_i, \mathbf{X}_j) \geq K(\mathbf{X}_i, \mathbf{X}_i)$.

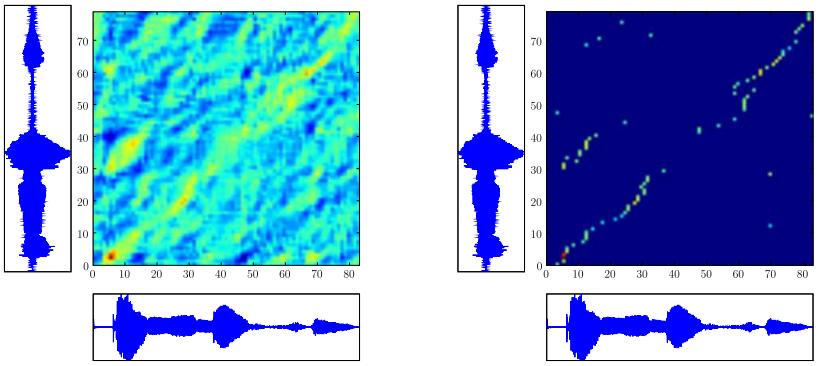
We thus propose here an alternative to taking the average over all frames. We consider, for each frame of sequence $\mathbf{X}_i$, the similarity measure of the closest corresponding frame in sequence $\mathbf{X}_j$. We thus propose to take a symmetric Max operator of the form:

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

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 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 semidefinite Gram matrix in which

case the training will converge perfectly well (Burges, 1998). 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)).

Figure 7.3 illustrates the main idea of the Max operator based kernel. Each subfigure represents all kernel evaluation values for two sequences from the same speaker pronouncing the same word; the blue color represents low values and the red color high values. Except for the silence part, we would thus like the diagonal to be higher in Figure 7.3(a). Indeed, having exactly two same accesses should produce a perfect diagonal. Figure 7.3(b) shows only the max values. Even if the correspondence is not perfect, the approximation seems good. Let us now compare the performance of the new Max operator based kernel.



(a) Mean operator kernel (b) Max operator kernel

Figure 7.3. Gram matrices for two accesses of the female speaker F44 pronouncing the same word "annulation", extracted from the Polyvar database.

Figure 7.4 and Table 7.2 show that the Max approach outperforms the standard one on the development set of Polyvar. The RBF kernel yields results similar to the polynomial kernel when the Max operator is used. It is interesting to note that now the optimal value is $p = 1$ and thus the sequence kernel becomes a linear classifier. This is probably because the Max operator is more appropriate. And this value is reasonable because the input space dimension

REF C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):1–47, 1998.

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.

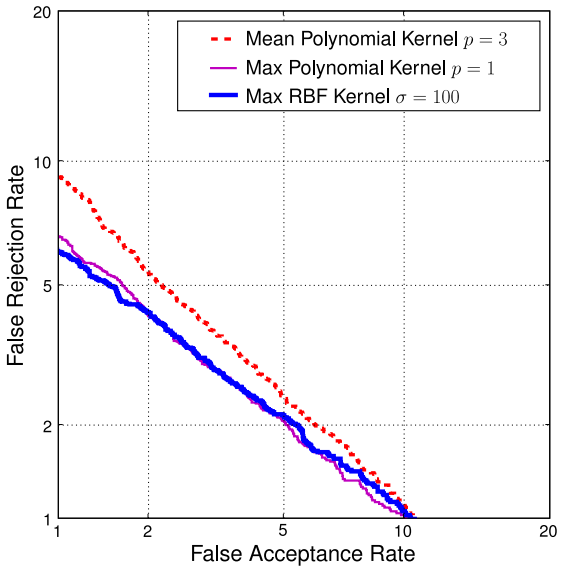


Figure 7.4. DET curves on the development set of the Polyvar database for Mean and Max operators for polynomial and RBF kernels.

Table 7.2. Results on the development set of the Polyvar database for Mean and Max operators for polynomial and RBF kernels.

	Mean $p = 3$	Max $p = 1$	Max $\sigma = 100$
EER [%]	3.46	2.99	3.06
95% Confidence	±0.28	±0.26	±0.26
# Support Vectors	87	73	99

7.3 Non-Mercer Kernels

The empirical results show that the Max operator based kernel yields good results (it will be also verified on other databases in the following), but it does not satisfy the Mercer conditions. We want here to study the consequences of

that potential problem. We first verify empirically that our kernel produces positive semidefinite Gram matrices. For the three NIST, Banca and Polyvar databases, we computed the eigenvalues of the Gram matrices obtained using the Max operator and various basic kernels (RBF, polynomial). All of them were positive except in one case: using the Max operator based kernel with polynomial kernel and $p = 1$ on Polyvar database. In that case, we obtained about 50 negative eigenvalues for about 900 positives eigenvalues. This is, nevertheless, one of the best kernel on the Polyvar database in term of performance. The obtained solution is thus good even if we have not solved the real SVM problem. Furthermore, using an RBF Max operator based kernel on the same database yields similar results. One can think that the found solution is close to the solution obtained if the eigenvalues would have been positive.

We also analyze the SVM implementation, here the Torch machine learning library (Collobert et al., 2002), and in particular the optimization algorithm. Solving the SVM problem is equivalent to solving a quadratic problem of the form $a\,x^2 + b\,x + c$ iteratively for two chosen examples of the training set (see detail in Collobert (2004), p.55). Having a positive semidefinite Gram matrix ensures that a kernel can be expressed by a dot product of $\phi()$ functions in some space. Normally only two cases can happen: $a > 0$ and $a = 0$. If the Gram matrix produces negative eigenvalues, then a can also be < 0. We verified this in our specific problem and it was never the case: thus the algorithm works. In order to prevent this for future training sets, we modified the implementation in order to solve the problem even when $a < 0$. For more details on the SVM optimization, the reader is referred to Collobert (2004). It is also known that adding some constant to the diagonal of the Gram matrix, makes the eigenvalues positive, which would be another way to be robust to this problem of negative eigenvalues. However doing this, we cannot make sure that the solution is close to the original problem.

7.4 Experimental Results on Polyvar and Banca Databases

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.

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

REF R. Collobert. *Large Scale Machine Learning*. PhD thesis, Université Paris VI, 28 June 2004.

Polyvar

Figure 7.5 presents the 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 shown in Table 7.3.

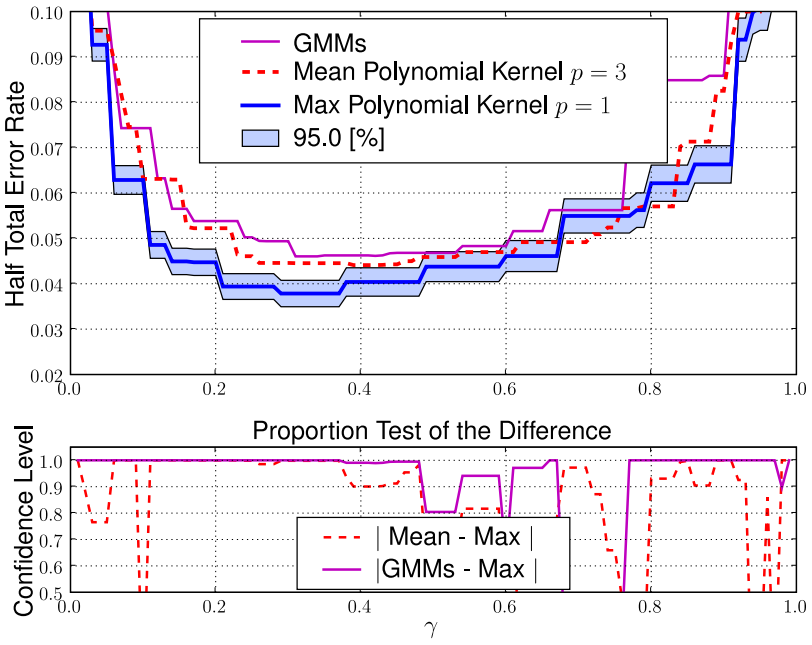


Figure 7.5. EPC curves on the test set of the Polyvar database for best Mean and Max operators for polynomial and RBF kernels.

The Max approach ($p = 1$) significantly outperforms GMMs for all values of γ with a confidence level greater than 99% most of the time. The Max approach ($p = 1$) also outperforms most of the time the Mean based system ($p = 3$) with a confidence level greater than 95%. The solution is also sparser in terms of number of support vectors. The Max RBF kernel yields results similar 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 7.3. Results on the test set of the Polyvar database for Mean and Max operators for polynomial and RBF kernels (SV = Support Vectors).

	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% Conf.	±0.34	±0.33	±0.32	±0.31	±0.32
# SV	-	62	87	73	99

Banca

Figure 7.6 and Table 7.4 present the performance of several systems on the Banca database. Once again, only the best systems for Max and Mean operators are presented.

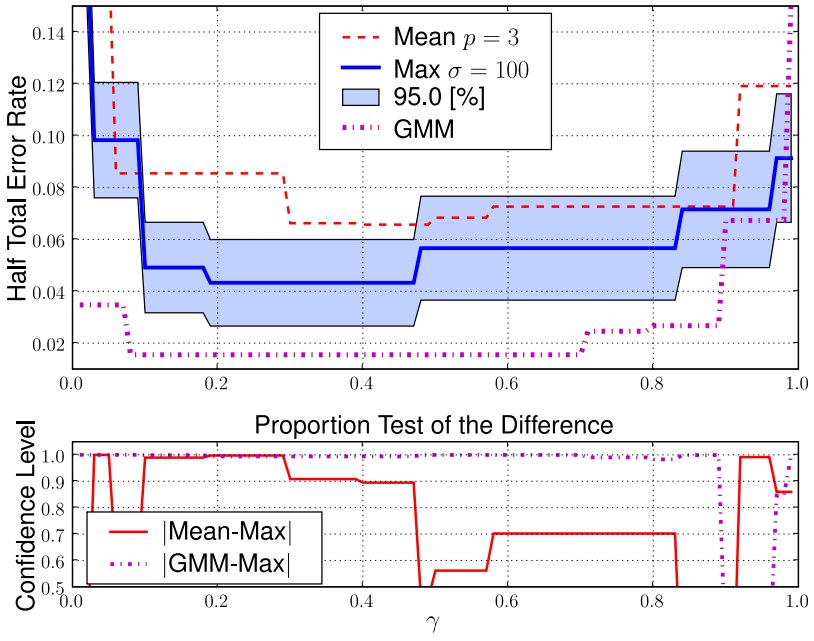


Figure 7.6. EPC curves on test set of the Banca database for best Mean and Max operator for polynomial and RBF kernels.

Table 7.4. Results on test set of the Banca database for Mean and Max operator for polynomial and RBF kernels (Support Vectors).

	GMM $N = 200$	Mean $\sigma = 8$ $C = \infty$	Mean $p = 3$ $C = \infty$	Max $p = 1$ $C = \infty$	Max $\sigma = 225$ $C = 130$
HTER [%]	2.72	8.71	6.41	5.98	4.70
95% Conf.	±1.42	±2.4	±2.08	±2.03	±1.78
# SV	-	18	27	42	17

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. Three recording conditions are used in this database: "controlled", "adverse" and "degraded". 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. Note however that (while this is not shown here) this difference is smaller on the development set than on the test set.

The Max approach ($\sigma = 225$) outperforms most of the time the Mean ($p = 3$) approach 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 (in terms of the number of support vectors) 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 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).

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

REF W.M. Campbell. Generalized linear discriminant sequence kernels for speaker recogni-

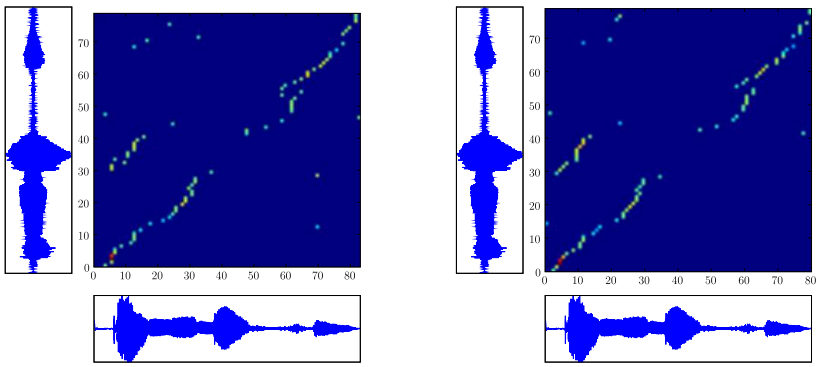
7.5 Smoothing the Max Kernel

Figure 7.3 shows that the maximum found by the Max operator based kernel is often in the diagonal of the Gram matrix for two same words, but it is still noisy. For text dependent speaker verification systems, a dynamic time warping (DTW) can be used, but it is not applicable in the context of text independent speaker verification. A simple solution consists in putting some local temporal constraints by applying a smoothing window that takes into account the frame context, as follows:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{T_i} \sum_{t_i} \max_{t_j} \sum_{h=0}^{H-1} k(\mathbf{x}_{t_i}, \mathbf{x}_{t_{j+h}}) + \frac{1}{T_j} \sum_{t_j} \max_{t_i} \sum_{h=0}^{H-1} k(\mathbf{x}_{t_{i+h}}, \mathbf{x}_{t_j})$$

where H represents the size of the smoothing window and is an hyper-parameter to tune using a development set.

Figure 7.7 shows the result of the smoothing procedure. One can see that smoothing yields max values that are closer to the diagonal, which is what we expect when the speaker pronounces the same sentence.



(a) Max operator based kernel.

(b) Smoothed Max operator based kernel with $H = 4$.

Figure 7.7. Gram matrices, Max and smooth Max operator based kernel, for two accesses of the female speaker F44 pronouncing the same word “annulation”, extracted from the Polyvar database.

tion. In *Proc IEEE International Conference on Audio Speech and Signal Processing*, pages 161–164, 2002.

Figure 7.8 and Table 7.5 show the results of the new smoothing kernel compared to the Mean and Max operator kernels. The new smoothing kernel outperforms statistically significantly the Mean operator kernel for all values of γ and outperforms statistically significantly the Max operator kernel for some value of γ. Note that the smoothing method gives also a smaller number of support vectors.

Table 7.5. Results on the test set of the Polyvar database for Mean, Max and smooth Max based kernels.

	Mean $p=3$ $C=\infty$	Max $p=1$ $C=\infty$	Smooth Max $p=1$ $C=\infty$ $H=4$
HTER [%]	4.47	3.9	3.40
95% Confidence	±0.32	±0.31	±0.28
# Support Vectors	87	73	48

7.6 Clustering Techniques

Even if the new proposed kernels seem promising, the underlying computational complexity makes their use not realistic for long sequences such as those of the NIST database. Let us remind the non-symmetric Max operator based kernel:

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

For each kernel $K()$, we have to compute a local kernel $k()$ between all T_i frames of the first sequence $\mathbf{X}_i$ and all T_j frames of the second sequence $\mathbf{X}_j$. Hence, in order to compare two sequences, $T_i T_j$ local kernel evaluations are needed. In order to avoid to compute the max over all the T_j frames for a given $\mathbf{x}_i$ frame of the first sequence, we can try first to cluster the frames of the two sequences and search the max only into a subset of frames of $\mathbf{X}_j$ that share the same cluster as $\mathbf{x}_i$. Unfortunately, this approach does not work empirically. In our preliminary experiments, neither using K-Means clustering nor GMM clustering, the results were good. Our explanation is that those methods are hard clustering techniques (a frame belong to only one cluster) and the hard constraint is too strong.

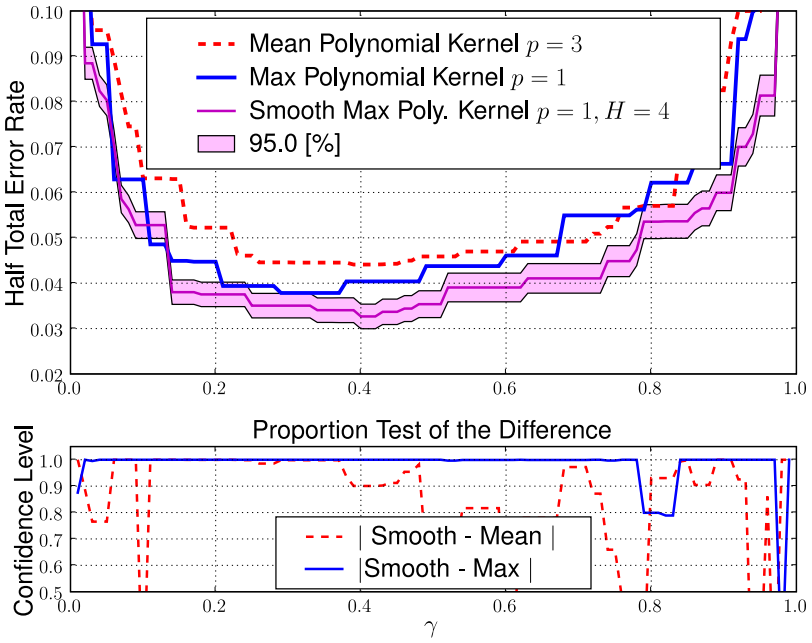


Figure 7.8. EPC curves on the test set of the Polyvar database for best Mean, Max and smooth Max operators for polynomial kernels.

In order to relax the hard constraint, we propose to use a soft clustering model based on HMM contextual posterior values, as proposed by (Ketabdar et al., 2005), and often called gamma values in the literature. They represent $p(q_t = s|\mathbf{X})$, the posterior probability of being in HMM state s at time t, given the whole sequence $\mathbf{X}$. Note that these posteriors can be efficiently estimated using a well-known recursion used in the EM training algorithm for HMMs.

Figure 7.9 shows the contextual posterior (hereafter simply called posterior) values for an HMM of 50 fully connected states, with one Gaussian per state. Blue color represents low values and high values are represented by red color. We can see that the phoneme /a/ and /la/ are represented by the same state (number 7). It is also interesting to note that the posterior values are peaky, short time stationary and smooth.

REF H. Ketabdar, J. Vepa, S. Bengio, and H. Bourlard. Developing and enhancing posterior based speech recognition systems. In *9th European Conference on Speech Communication and Technology, Eurospeech-Interspeech*, 2005.

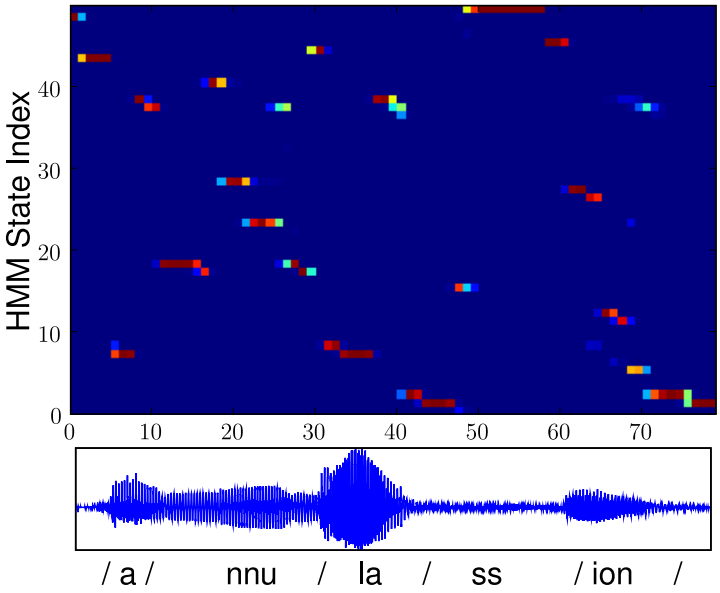


Figure 7.9. Posterior values for access "f4425w14" of Polyvar database.

Let us now describe an algorithm that uses posterior values to reduce the complexity of the Max operator based kernel. Let us consider the non-symmetric Max operator based kernel, but instead of comparing a given $\mathbf{x}_{t_i}$ to all frames of $\mathbf{X}_j$, we want to consider only a subset of $\mathbf{X}_j$, as follows:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{T_i} \sum_{t_i} \max_{t_j \in \{t\}^*} k(\mathbf{x}_{t_i}, \mathbf{x}_{t_j})$$

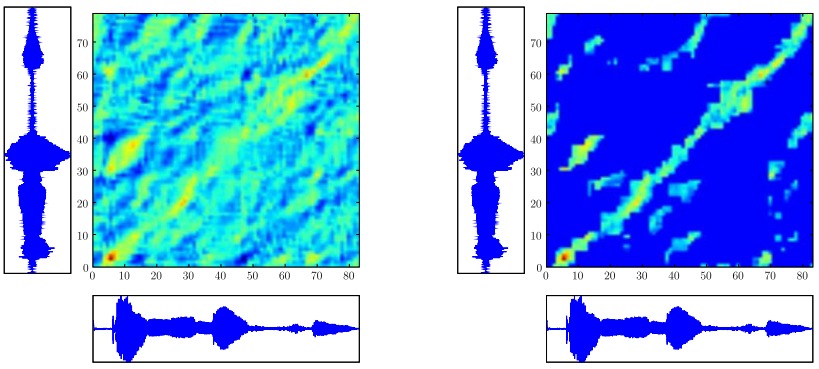
where $\{t\}^*$ is a subset of index frames of the sequence $\mathbf{X}_j$ given by:

$$\{t\}^* = \arg \operatorname*{nbest}_{\{t\}_1^{N_b}} p(q_t = s^*(t_i)|\mathbf{X}_j)$$

where $\operatorname{nbest}_{\{t\}_1^{N_b}}$ is a new operator that returns the N_b best values with respect to the posterior values of the state $s^*(t_i)$, computed as follows:

$$s^*(t_i) = \arg \max_s p(q_{t_i} = s|\mathbf{X}_i).$$

Figure 7.10 shows the Gram matrix. On Figure 7.10(b), all parts of the graphic with the dark blue color will not be considered by the kernel evaluations. We can see that the diagonal values are kept most of the time.



(a) Mean kernel.

(b) Mean kernel with posterior based clustering approach (50 states, $N_b = 10$). More than 80% of the kernel evaluations are saved.

Figure 7.10. Gram matrices for two accesses ("f4413w06" and "f4425w14") of the female speaker F44 pronouncing the same word "annulation", extracted from the Polyvar database.

In order to perform the clustering, we need to train an HMM, here using the world model population without using any transcription; the training is completely unsupervised with the EM procedure maximizing the data likelihood. All the hyper-parameters are tuned in order to minimize the ERR on the development set. The HMM used to perform NIST experiments has 50 states with only one Gaussian per state and a full transition probability matrix. The best value for N_b is 200. In fact, the error is quite stable from 100. For simplicity reason, the feature extraction procedure used to enroll the HMM is the same as the one used for the SVMs; this can be sub-optimal in the sense that these features should be able to discriminate between phonemes and not between speakers.

We tried to add a minimum duration constraint by replicating each HMM state, but it did not yield any improvement. Further analysis are needed to explain this, as intuitively the minimum duration should improve the results: we have seen that smoothing the kernel by putting local temporal constraints helps the system and thus we had the same hope for the minimum duration constraint.

Figure 7.11 shows the results for a Max operator based kernel without the use of the posterior clustering approach (needs several weeks to run) and with

the posterior clustering approach (needs less than 2 days to run). We can see that the approximation is reasonable and gives similar results. These results have been estimated on a previous campaign of the NIST database.

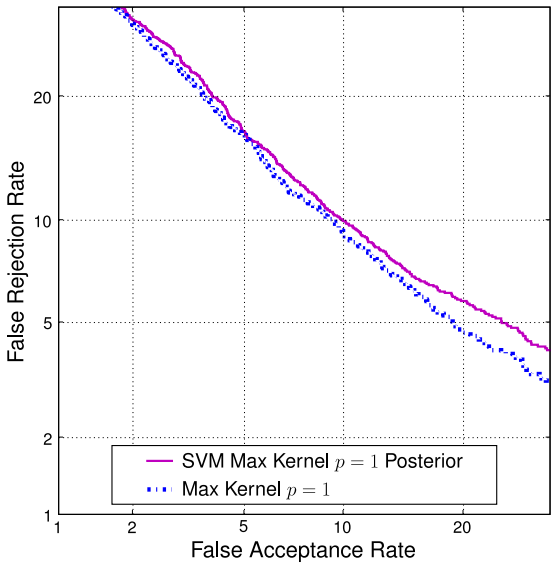


Figure 7.11. Results on the development set of a previous version of the NIST database: Max operator based kernel $p = 1$ with and without posterior based approximation.

7.7 Experimental Results on the NIST Database

Due to Max operator kernel complexity, it was too costly to run this new kernel on the NIST database. Using the posterior clustering approach, we can presents the results for the NIST database.

Figure 7.12 shows the results for the GLDS based kernel approach with $p = 3$ and a Mean operator polynomial kernel with $p = 3$. Even if they are comparable for most values of γ, we can see that they are not really equivalent and the polynomial approach outperforms the GLDS based kernel for some values of γ. As it does not need the computation of a normalization vector $\frac{1}{\sqrt{\psi_n}}$ in (2.32), this approach seems preferable. Note that the Mean operator kernel can be computed with the same complexity as the GLDS approach for a polynomial form.

The Max operator based kernel is compared to the Mean operator based

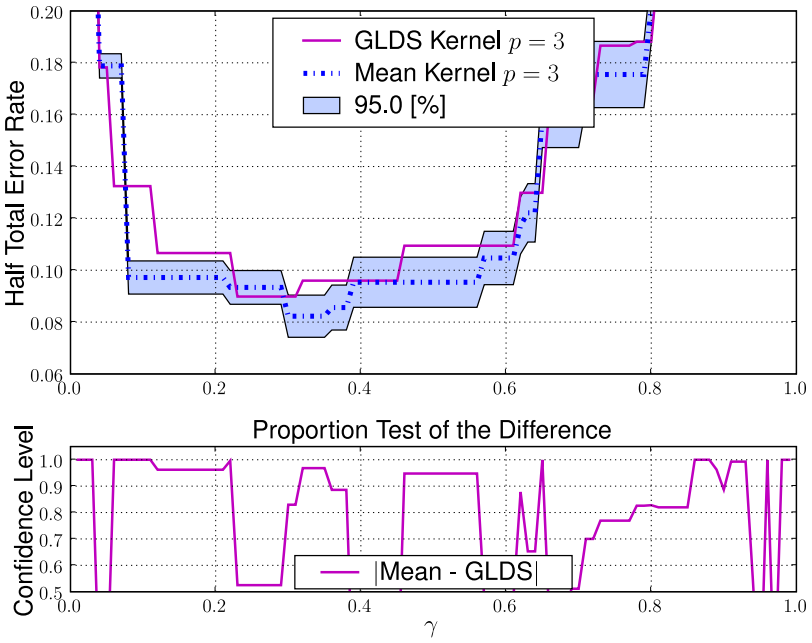


Figure 7.12. Results on the test set of the NIST database: GLDS Kernel $p = 3$ vs Mean operator Kernel $p = 3$.

kernel on Figure 7.13 and Table 7.6. Unfortunately, the improvement observed on the two Banca and Polyvar databases does not appear on the NIST database for all values of γ. Moreover, for small values of γ the Max operator based kernel is worse than the standard Mean operator kernel. Even if it needs deeper analysis to be explained, intuitively the longer the sequence is, the bigger the risk of confusion is when the max is taken. It can thus be important to add some local temporal smoothing procedure. For example, one can take the N best frames instead of the single best as with the Max operator based kernel. One can also use the HMM posterior values, as in Figure 7.9. We can see that these values cut the sequence into short segments. One can use this information to create a new kernel that compares segments instead of frames.

It is interesting to note that now the C smoothing parameter has a positive influence. It reduces drastically the number of support vectors from 135 to 33 and Figure 7.14 shows that it reduces the HTER and also the DCF for the costs used by the NIST campaign: $\gamma \approx 0.909$. It is also interesting to note that

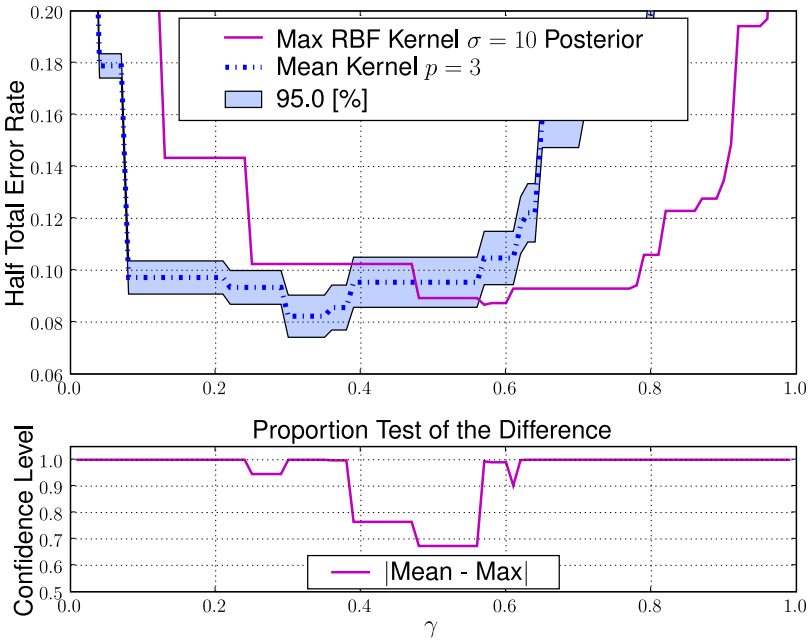


Figure 7.13. Results on the test set of the NIST database: Max operator RBF Kernel $\sigma = 10$ using posterior based approximation and Mean operator Kernel $p = 3$.

Table 7.6. Results on the test set of the NIST database for Mean and Posterior based Max operators for polynomial and RBF kernels(SV = Support Vectors).

	GMM $N = 100$	GLDS $p = 3$ $C = \infty$	Mean $p = 3$ $C = \infty$	Max $p = 1$ $C = \infty$	Max $\sigma = 10$ $C = 0.5$
HTER [%]	8.68	11.06	10.48	11.01	9.12
95% Conf.	$\pm$0.84	$\pm$1.05	$\pm$1.03	$\pm$1.04	$\pm$0.72
# SV	-	38	40	110	33

in that case the RBF kernel outperforms the polynomial kernel.

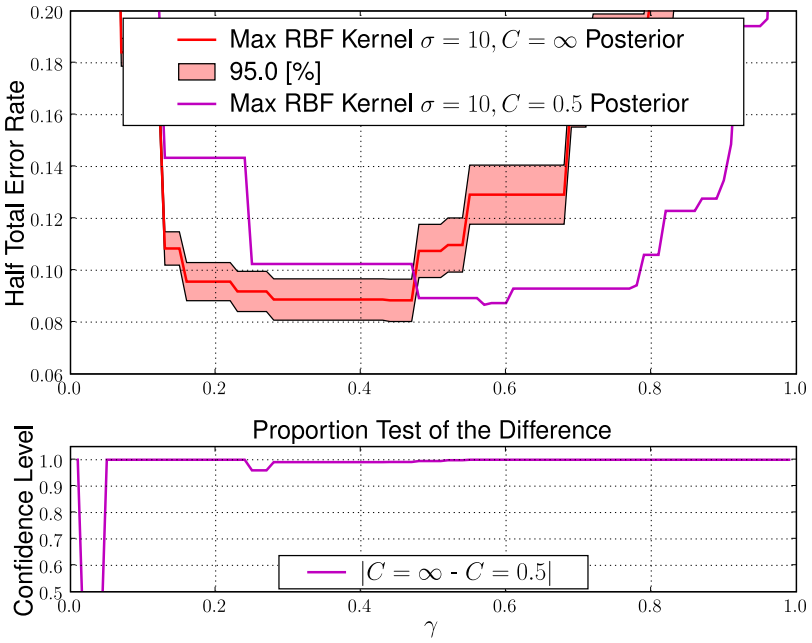


Figure 7.14. Results on the test set of the NIST database: Max operator RBF Kernel $\sigma = 10$ using posterior based approximation for two different values of C: ∞ and 0.5.

7.8 Conclusion

We have proposed in this chapter, 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 most of the time SVM based state-of-the-art approaches for the 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. This work was published in:

CONTRIB J. Mariéthoz and S. Bengio. A kernel trick for sequences applied to text-independent speaker verification systems. In *Second Workshop on Multimodal User Authentication, MMUA*, 2006. IDIAP-RR 05-77

A longer version of this paper has been submitted to the Patter Recognition journal.

We have also proposed a smoothing method to enforce local temporal constraints and show that it improves statistically significantly the baseline system.

The main drawback of our proposed method is the large underlying complexity for long sequences. We thus proposed new clustering methods based on HMM contextual posterior values in order to make the Max operator based kernel usable with long sequences. We performed some experiments on the NIST database and showed that the approximation was good and reduced the computing time from several weeks to less than two days. Unfortunately, while the Max operator based kernel outperformed the Mean operator based kernel for both Banca and Polyvar database; it was not the case for all possible decision thresholds on the NIST database. On the other hand, it allows for the first time the use of infinite dimensional kernels on the NIST database and opens some research directions to create new sequence kernels. In particular, we think that it should be interesting to consider methods to align speech segments using contextual posterior values in order to create a new sequence kernel.

We have also shown that the SVM 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 to properly handle unbalanced data, as is often the case in speaker verification tasks. A serious indicator of the problem is that using a polynomial kernel with a Max operator, the optimal degree is always equal to 1. Thus we hope to be able to reduce the capacity by being able to properly tune the C hyper-parameter.

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