# SVM SPEAKER VERIFICATION USING A NEW SEQUENCE KERNEL

*Jérôme Louradour, Khalid Daoudi*

Institut de Recherche en Informatique de Toulouse - CNRS UMR 5505
118, route de Narbonne, 31062 Toulouse cedex 04, FRANCE
phone: +33 (0)5 61 55 72 01 - fax: +33 (0)5 61 55 62 58
email: {louradou,daoudi}@irit.fr - web: www.irit.fr/recherches/SAMOVA/

## ABSTRACT

Using the framework of Reproducing Kernel Hilbert Spaces, we develop a new sequence kernel that measures similarity between sequences of observations. We then apply it to a text-independent speaker verification task using the NIST 2004 Speaker Recognition Evaluation database. The results show that incorporating our new sequence kernel in an SVM training architecture not only yields performance significantly superior to those of a baseline UBM-GMM classifier but also outperforms the Generalized Linear Discriminant Sequence (GLDS) Kernel classifier. Moreover, our kernel maps to a relatively low dimensional feature space while allowing a large choice for the kernel function.

## 1. INTRODUCTION

The goal of speaker verification is to determine whether a test speech utterance was produced by a target speaker, referred to as client. The majority of text independent automatic speaker verification systems in use today are based on statistical classification methods, most of the time involving Gaussian Mixture Models [1]. Nevertheless, for classification tasks, discriminant methods such as Support Vector Machines (SVMs) can achieve better performance than generative classifiers, while requiring significantly less training data. Even if the core algorithms of such kernel methods are now mastered [2], and their properties have been widely studied, finding the optimal way to represent real-world data as input to these algorithms remains an open problem.

In the case of speaker verification, the decision is binary (claimant speaker / impostor), and items to be classified are variable length sequences. In the following, we will call *input space* the space in which each vector of the sequence is observed : in our experiments, these acoustic vectors are extracted from a cepstral analysis. In practice, training data of a given speaker (class +1), besides the noise corruption, are scattered within the cloud of background vectors, which are considered in the case of verification as impostor data (class -1). Moreover, the problem geometry is made complex by the mismatch between training and testing conditions : different types of handsets and channels, background noise.

In order to apply SVM to a speaker verification task, the simplest idea is to train with vectors in the input space and to compute the mean of SVM's output on frames in order to assign a score to a sequence, as it was done in [3]. But because of the previously mentioned reasons, such a frame-level discriminative approach gives poor performance. In addition, the implementation is awkward : as it is common to collect a large amount of background data (typically telephone conversations) to feed the training of the classifier with impostor instances, the training algorithms become intractable unless a data clustering is used.

Moreover, in speaker verification, the goal is to minimize classification errors on sequences, not on speech frames. That is why a sequence-based learning approach seems more appropriate.

Several recent studies dealt with the conception of sequence kernels, the challenge being to overcome the length variability. Such kernels aim at quantifying the similarity between sequences in relation to a problem. One can distinguish three kinds of approaches :

- A first trend (to compare two sequences) consists in training statistical models from these sequences, and defining the kernel as a similarity between the two estimated distributions (Kullback divergence [4], Bhattacharyya affinity [5], $\chi_2$ distance, etc.). In our case, the usual shortness of test sequences prevents a robust estimation of distribution parameters.
- A second trend recommends to work in the probabilistic score space given some appropriate client and impostor models [6], or in the derivate space via the Fisher kernel [7, 8]. These methods are computationally heavy and their performance are comparable to those of an UBM-GMM method.
- Another strategy to construct sequence kernels, recently developed by W.Campbell in [9, 10], is based upon training on one sequence and testing on another one. This theoretical process leads to efficient and powerful speaker verification systems. The new approach developed in the present paper uses the same philosophy, but leads to a more flexible sequence kernel.

The last approach [9] amounts to mapping explicitly sequences to a fixed-dimension space using a polynomial expansion, and to performing a dot product (linear kernel) in this space. Let $d$ the dimension of the input space, and $k$ the maximal polynomial order, the dimension of the mapping is $M = \frac{(d+k)!}{d!k!}$. In practice, $d$ is about 25, and $M$ becomes too large when $k > 3$ (*e.g.* $M = 23,751$ when $d = 25$ and $k = 4$). So the approach is limited to polynomial expansion with orders equal or lower than three.

In this paper, we develop a new sequence mapping which not only supports polynomials of any degree, but also any other kernel function.

## 2. CONCEPTION OF THE NEW SEQUENCE KERNEL

This section introduces the theoretical material which lead to the formulation of our new sequence kernel.

### 2.1 Learning on a sequence in the Reproducing Kernel Hilbert Space

Consider we are given a training corpus $(\tau_n, s_n)_{n=1...N}$, with real vectors $\tau_n \in {}^d$ and binary labels $s_n \in \{0,1\}$, where $s_n = 0$ for a fixed set of background ***sparse*** data $C = (c_m)_{m=1...M}$ and $s_n = 1$ for all vectors of a sequence $A = (a_t)_{t=1...T_A}$ produced by a given target speaker. The process of finding a discriminant function $f : {}^d \rightarrow$ on this corpus can be written in the generalized form :

$$\min_{f\in\mathcal{H}} \left[ {}_{n=1}^{N} L(f(\tau_n), s_n) + \lambda J(f) \right] \tag{1}$$

where $L(y, f(x))$ is a loss function, $J(f)$ is a penalty functional, and $\mathcal{H}$ is the space of functions in which the search for $f$ is performed.

An important subclass of problems of the form (1) are generated by a positive definite kernel $K$, and the corresponding space of functions $\mathcal{H}_K$ is called *Reproducing Kernel Hilbert Space* [11]. Suppose that $K$ has an eigen-expansion

$$K(x,y) = {}_{i=1}^{D} \gamma_i^2 \ {}_i(x) \ {}_i(y) = \langle \gamma \ (x), \gamma \ (y) \rangle \tag{2}$$

with $\gamma = diag(\gamma_1, ..., \gamma_D)$ and $\ : {}^d \rightarrow {}^D$ is a $D$-dimensional vector function ($D$ can be infinite, *e.g.* with a radial basis kernel). Then each element of $\mathcal{H}_K$ have an expansion of the form $f(x) = {}_{i=1}^{D} \varepsilon_i \ {}_i(x)$.

In [12], it is shown that the solution to (1) has a finite-dimensional form :

$$f(x) = {}_{n=1}^{N} \omega_n K(x, \tau_n) \tag{3}$$

In the following, our sequence mapping for kernel computation (6,8,9) will be based on the basis functions $x \mapsto K(x, \tau_n)$, known as *representer of evaluation* at $\tau_n$ in $\mathcal{H}_K$. In order to achieve computationally efficiency and system stability, we want to conceive a sequence kernel which is independent of the target speaker. Thus, we will search for solutions of the form :

$$f(x) = {}_{n=1}^{M} \omega_m K(x, c_m) \tag{4}$$

In other words, the representers will be $K(., c_m)$ instead of $K(., \tau_n)$.

Considering regression with squared-error loss and no generalized ridge penalty ($\lambda = 0$ in (1)), the method of normal equations with some approximations (*cf.* [13] for details) gives the solution $\hat{\omega}_A = [\hat{\omega}_1, ..., \hat{\omega}_M]^{\mathrm{T}}$ :

$$\hat{\omega}_A = M.\mathbf{K}_C{}^{-2}\overline{\varphi_{\mathbf{C}}}(A) \tag{5}$$

where $\mathbf{K}_C = (K(c_m, c_n))_{(m,n)\in\{1...M\}^2}$ is the symmetric hessian matrix, and where we define the $M$-dimensional functions :

$$\begin{cases} \overline{\varphi_{\mathbf{C}}}(x_1, ..., x_T) = \frac{1}{T} \ {}_{t=1}^{T} \varphi_{\mathbf{C}}(x_t) \\ \varphi_{\mathbf{C}}(x) = [K(x, c_1), ..., K(x, c_M)]^{\mathrm{T}} \end{cases} \tag{6}$$

Each component of $\hat{\omega}_A$, which can be seen as a basis function, is indexed by a prototype $c_m$. If more flexiblity is desired in a particular region of the input space, then that region needs to be represented by more basis functions of the form $K(., c_m)$. By this way, we can control the complexity of the representation.

### 2.2 Scoring on another sequence

Suppose a model $\hat{\omega}_A$ of the form (5) was learnt from a first sequence $A$. The similarity between an input vector $x$ and $A$ is given by the discriminant function :

$$f(x) = \langle \hat{\omega}_A, \varphi_C(x) \rangle \tag{7}$$

Then, extending this measure to another sequence $B = (b_t)_{t=1...T_B}$ can be done by computing the mean of each frame similarity :

$$\begin{aligned} similarity(B|A) &= \frac{1}{T_B} \ {}_{t=1}^{T_B} f(b_t) \\ &= M.\overline{\varphi_{\mathbf{C}}}(B)^{\mathrm{T}}\mathbf{K}_C{}^{-2}\overline{\varphi_{\mathbf{C}}}(A) \end{aligned}$$

Skipping the multiplicative factor $M$, this leads to our new symmetric sequence kernel :

$$\kappa(A,B) = \langle \overline{\Phi_C}(A), \overline{\Phi_C}(B) \rangle \tag{8}$$

where we define the sequence mapping :

$$\overline{\Phi_C}(x_1, ...x_T) = \mathbf{K}_C{}^{-1}\overline{\varphi_{\mathbf{C}}}(x_1, ..., x_T) \tag{9}$$

In [13], we show that this mapping is equivalent to a projection, in the $D$-dimensional feature space defined by in (2), of the average expansion $\frac{1}{T} \ {}_{t=1}^{T} \ (x_t)$, on the base of expanded background vectors $( \ (c_m))_{m=1...M}$.

## 3. PRACTICAL IMPLEMENTATION OF THE NOVEL SEQUENCE KERNEL

The computation of our sequence kernel (8) is done by mapping each sequence to a $M$-dimensional space, and by computing a dot product in this space. Thus, this computation would be intractable if $M$ is too high. This is why we assumed in 2.1 that the set of impostor data is sparse. However, in practice, the amount of background data is very high and they are not sparse at all. In our implementation, the set $C$ is obtained by a simple vector quantization of the original impostor data, thus the $(c_m)$ are vectors representative of these data.

In the following, we describe how to use this sequence kernel on a SVM scheme for a speaker verification task.

### 3.1 Notion of sequence

In our case, a sequence is defined as *a set of vectors produced by a same speaker under the same conditions* (handset, channel, language,...).

Experiments have shown that even when there is only one sequence available per speaker, there is no point in splitting this sequence into pieces. Such a trick provides no gain for our method as well as for the GLDS method in [9]. On the other hand, if several sequences from different recordings are available, they should not be combined, in order to preserve information about session variability.

### 3.2 Sequence mapping

The sequence mapping $\overline{\Phi_C}$ is entirely determined by a set of representers $C$ and a function $K$ satisfying the Mercer's conditions. The implementation of the mapping of a sequence $X = (x_1, ..., x_T)$ is shown in the Fig.1.

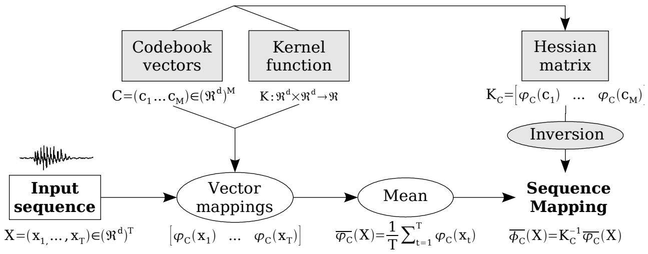
Codebook vectors
Kernel function
Hessian matrix
$C=(c_1 \ldots c_M)\in(\Re^d)^M$
$K:\Re^d\times\Re^d\to\Re$
$K_C=[\varphi_C(c_1) \ \ldots \ \varphi_C(c_M)]$
Inversion
Input sequence
Vector mappings
Mean
Sequence Mapping
$X=(x_1,\ldots,x_T)\in(\Re^d)^T$
$[\varphi_C(x_1) \ \ldots \ \varphi_C(x_T)]$
$\overline{\varphi_C}(X)=\frac{1}{T}\sum_{t=1}^{T}\varphi_C(x_t)$
$\overline{\Phi_C}(X)=K_C^{-1}\overline{\varphi_C}(X)$

Figure 1: Block diagram for the computation of the proposed Sequence Mapping $\overline{\Phi_C}$. Gray boxes indicate preliminary settings for the system.

### 3.3 SVM Model learning

Once the mapping is defined, we can pre-compute the mappings of impostor sequences from a background corpus, and then train one SVM model per target speaker with the procedure schematized in Fig.2. In our experiments, SVM models were trained with SVM Torch [14], using the SMO algorithm with a linear kernel. Note that each component of the mapping is normalized in order to keep the same variability for each real input. The normalization is given by :

$$\overline{\ }_C(X) \mapsto \frac{\overline{\ }_C(X)-\mu}{\sigma} \qquad (10)$$

where $\mu$ and $\sigma$ are respectively the mean and standard deviation estimates, on the background corpus, of the mapped feature vectors $\overline{\Phi_C}(X_{imp})$. This is a common precaution for SVM as they are not invariant to linear transformations.

### 3.4 Utterance testing

Modulo a decision threshold, the output of a SVM on a sequence $Y$ has the form :

$$score(Y) = \underset{i}{\ } \alpha_i y_i \langle \overline{\Phi_C}(Y), \overline{\Phi_C}(S_i) \rangle \qquad (11)$$

where $(\alpha_i)$ are positive weights learnt during SVM training, $(S_i)$ are training sequences, and $y_i = \pm 1$ is the corresponding class label (+1 for speaker sequences, −1 for impostor sequences).

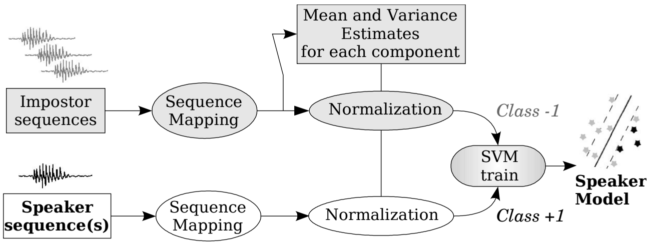

Figure 2: Block diagram of the SVM training scheme.

Given the linearity of the dot product, this score computation can be simplified :

$$score(Y) = \langle \overline{\Phi_C}(Y), \underset{i}{\ } \alpha_i y_i \overline{\Phi_C}(S_i) \rangle = \langle \overline{\Phi_C}(Y), \ _{sp} \rangle \qquad (12)$$

As in [9], collapsing all the support sequences in a single model $_{sp}$ allows memory saving for speaker model storing, and time saving during testing.

The binary decision is taken by comparing the sequence score to a fixed threshold.

## 4. EXPERIMENTS

### 4.1 Database

We applied our new sequence kernel to the 2004 NIST Speaker Recognition Evaluation, in the core test condition [15]. In this condition, one sequence with more or less two minutes of speech is available to train each speaker model. For the background corpus, about one thousand impostor sequences per gender were extracted from the 2001 NIST SRE database.

### 4.2 Front-end Processing

To obtain acoustic vectors from a speech utterance, 12 MFCC are extracted on 16ms Hamming window, processing at a 10ms rate. 12 derivative coefficients and the derivative of the energy logarithm are also added. Then, a speech activity detector discard low-energy frames. Finally, the 25-dimensional input vectors are warped over 300 frame windows [16]. Such a normalization reduce the combined effects of slowly changing additive noise and channel effects.

### 4.3 Reference systems

In order to validate our new system, we compare its performance to two state-of-the-art systems. For fair comparison, exactly the same development and test data where used for all systems.

The first one, that we presented at NIST 2004 SRE, is a UBM-GMM system that estimates two gender-dependent background models. 512 components Gaussian Mixture Models (GMM) with diagonal covariance matrices are estimated with a EM algorithm. Each target speaker GMM is derived from the appropriate background model, by adapting mean vectors with a MAP criterion. During the decision phase, a sequence score is computed as the mean log-likelihood ratio. For better

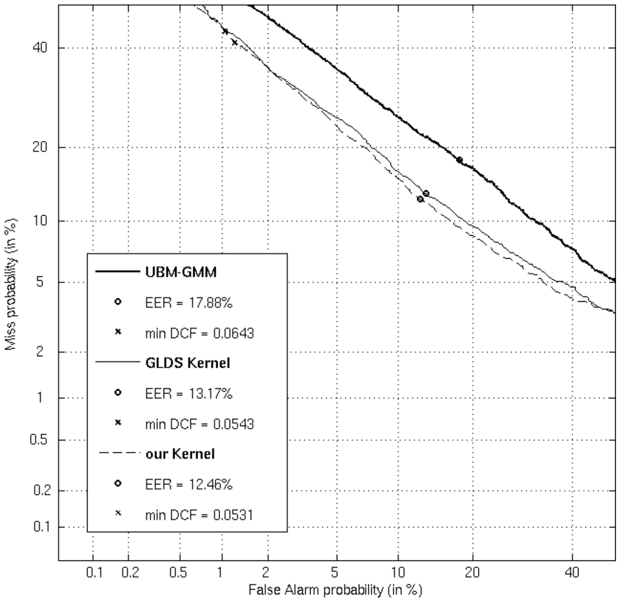

Figure 3: DET plots showing a comparison of the new approach with state-of-the-art approaches. Circles show Equal Error Rates and x-marks show operating points corresponding to the minimum of the DCF defined in (13).

performances, only the 10-best scoring components are used to calculate each frame log-likelihood ratio.

The second one uses the GLDS kernel described in [9]. The principle is similar to our approach, but the sequence mapping is different. It consists of an average polynomial expansion, followed by a normalization which parameters are also determined from the background corpus. The maximal polynomial order is set to 3, the size of the mapping is $\frac{(25+3)!}{25!3!} = 3276$.

### 4.4 Results

The Detection Error Tradeoff (DET) curves for both reference systems and our system are shown in Fig.3 (each point of the curve corresponds to a decision threshold). The Detection Cost Function, that has to be minimized, was defined by NIST SRE [15] as a weighted sum of miss and false alarm probabilities (*resp.* $P_{miss}$ and $P_{fa}$) :

$$DCF = (0.1 \times P_{miss}) + (0.9 \times P_{fa}) \tag{13}$$

For our system, we chose a polynomial kernel of degree 7 : $K(x,y) = (1 + \langle x,y \rangle)^7$. RBF kernels also achieve similar performances. $M = 2048$ representers was estimated by a vector quantization of the background corpus. Experiments have shown that increasing the number $M$ of representers generally improves performances.

One can see that our system outperforms the others at all operating points. Experiments on NIST 2003 SRE with exactly the same technical choices confirm this statement. In comparison with the GLDS Kernel method, our mapping has in addition a lower dimension (2048 instead of 3276).

## 5. CONCLUSION

We introduced a novel sequence kernel to build a new SVM speaker verification system. All the experiments we carried out show that this system significantly outperforms the classical UBM-GMM classifier. Moreover, it outperforms the powerful GDLS method while using lower-dimensional feature space. In addition, the flexibility of the new kernel offers good perspectives for further improvements and can be applied to other classification tasks.

## REFERENCES

[1] D. Reynolds, T. Quatieri, and R. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital Signal Processing*, 2000.

[2] B. Schölkopf and A.J. Smola, *Learning with kernels : Support Vector Machines, regularization, optimization and beyond*, MIT Press, 2001.

[3] M. Schmidt and H. Gish, “Speaker identification via support vector machines,” in *Proc. ICASSP*, 1996.

[4] P. Moreno and P. Ho, “A new svm approach to speaker identification and verification using probabilistic distance kernels,” in *Proc. Eurospeech*, 2003.

[5] R. Kondor and Jebara T., “A kernel between sets of vectors,” in *Proc. ICML*, 2003.

[6] Q. Le and S. Bengio, “Client dependent gmm-svm models for speaker verification,” in *Proc. Networks, ICANN*, 2003.

[7] T. Jaakkola and D. Haussler, “Exploiting generative models in discriminative classifiers,” *Advances in Neural Information Processing Systems 11*, 1998.

[8] V. Wan and S. Renals, “Speaker verification using sequence discriminant support vector machines,” *IEEE Trans. on Speech and Audio Processing*, 2004.

[9] W.M. Campbell, “Generalized linear discriminant sequence kernels for speaker recognition,” in *Proc. ICASSP*, 2002.

[10] W. Campbell, J. Campbell, D. Reynolds, D. Jones, and T. Leek, “Phonetic speaker recognition with support vector machines,” in *Proc. NIPS*, 2003.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, chapter 5. Basis Expansions and Regularization, Springer, 2001.

[12] G. Wahba, *Applied Mathematics*, vol. 59, chapter Spline Models for Observational Data, CBMS-NSF Regional Conference Series, 1990.

[13] J. Louradour, “a new sequence kernel and its application to speaker verification,” Irit research report, www.irit.fr/∼jerome.louradour/papers/, 2005.

[14] R. Collobert and Bengio, “Svmtorch : Support vector machines for large-scale regression problems,” *Journal of Learning Machine Research*, 2001.

[15] “Nist speaker recognition 2004 evaluation plan,” http://www.nist.gov/speech/tests/spk/, 2004.

[16] J. Pelecanos and S. Sridharan, “Feature warping for robust speaker verification,” in *Proc. Speaker Odyssey*, 2001.