A COMPARISON BETWEEN SEQUENCE KERNELS FOR SVM SPEAKER VERIFICATION

Khalid Daoudi

IRIT-CNRS. France.

Jérôme Louradour

University of Montreal. Canada

ABSTRACT

We present a comparative study of several SVM speaker verification (SV) systems based on sequence kernels: the GMM-supervectors kernel, the Fisher kernel, the Generalized Linear Discriminant Sequence (GLDS) kernel, our Feature Space Normalized Sequence (FSNS) kernel and a "novel" sequence kernel in SV, the Correlation kernel. We also compare these SVM systems to the conventional generative UBM-GMM. We carry out experiments on the NIST'2005 SRE evaluation set. The results show that the FSNS system yields comparable performances to UBM-GMM and significantly outperforms GLDS. They also show that the GMM-supervectors system outperforms all the others. Finally, they show that the best performances are achieved by fusing the FSNS and the GMM-supervectors systems.

Index Terms- Speaker verification, Sequence kernels, SVM

1. INTRODUCTION

Support Vector Machines are an interesting alternative to the traditional Gaussian Mixture Models (GMM) for speaker verification using acoustic features, as they are well suited to separate rather complex regions in binary classification problems, through an optimal nonlinear decision boundary. A challenge however in applying SVM to monitor conversations in a communication network, such as in NIST SRE (Speaker Recognition Evaluation) campaigns, is to deal with the huge amount of data available. Thus, in order to exploit a rich database involving various types of low quality cell phones with an SVM training algorithm, the frame-based approach needs to be adapted to make a tractable training and testing procedure. A solution could be to use clustering methods to reduce the size of the training. On the other hand, the problem in speaker verification is to classify sequences of vectors. It is then more natural to conceive kernels that measure similarity between sequences and use them in an SVM architecture.

The first contribution of this paper is to present a comparative study of four state-of-the-art SVM speakers verification system based on kernels between sets of vectors (that we call sequence kernels for simplicity). The first one is the *Generalized Linear discriminant Sequence* (GLDS) kernel that has been developed in [2] and was actually the first sequence kernel used (with success) in SVM speaker verification. The second one is the *Feature Space Normalized Sequence* (FSNS) kernel that we developed recently in [8]. This kernel is actually an extension of the GLDS kernel that overcomes theoretically and practically the limitations of the latter. The third one is the *GMM-supervectors* sequence kernel that has been developed recently in [3]. This kernel has been shown to be among the best SVM-based systems in the recent NIST SRE campaigns, in term of individual performances. The fourth one is the *Fisher* kernel that has been extensively studied in [13].

The second contribution is to introduce the *correlation* kernel [9] as a "novel" sequence kernel, in the sense it has never been applied to speaker verification (to the best of our knowledge). This kernel is indeed attractive in such an application because it has a closed form for GMM, and easy to compute for the particular GMM we deal with in speaker verification. We also compare all these SVM systems to the conventional generative UBM-GMM (Universal Background Model-GMM) system. The experiments are carried out on the NIST'2005 SRE evaluation set.

2. SEQUENCE KERNELS IN SVM SPEAKER VERIFICATION

Sequence kernels can be classified in 3 categories: Mutual Information (MI) kernels, kernels between distributions and combination of vector kernels. Roughly speaking, the basic idea behind MI kernels [11] is to draw a similarity measure from a prior parametric density. The popular Fisher kernel [5] can be actually seen as a MI kernel. The general principle of kernels between distributions is to estimate a probability distribution on each input sequence, and then to compute a kernel between these distributions. Probability product kernels [6] are a well known example of such kernels. As for the last category, it simply consists in considering a functional of inter- and/or intra-sequence vector kernels. The GLDS and FSNS kernels are examples of such kernels. We refer to [7] for a more detailed description of the different sequence kernels families. From now on, we note $\mathcal{X} = \{\mathbf{x}_t\}_{t=1\dots T_\mathcal{X}}$ and $\mathcal{Y} = \{\mathbf{y}_t\}_{t=1\dots T_\mathcal{Y}}$ two sequences of d-dimensional vectors.

2.1. The Fisher kernel

In their general form, MI kernels cannot be readily applied to speaker verification. However, the Fisher kernel (an approximation of a MI kernel) is easy to compute for GMM. If $\boldsymbol{\theta}_\circ = \{\omega_g, \boldsymbol{\mu}_g^\circ, \boldsymbol{\Sigma}_g \,|\, g = 1 \cdots G\}$ is the parameter set of the prior GMM (the UBM-GMM model), the Fisher kernel is given by: $\kappa_{\text{Fisher}}(\mathcal{X}, \mathcal{Y}) = \boldsymbol{\delta}(\boldsymbol{\theta}_\circ, \mathcal{X})^{\mathrm{T}}\ F^{-1}\ \boldsymbol{\delta}(\boldsymbol{\theta}_\circ, \mathcal{Y})$, where

- The Fisher mapping $\boldsymbol{\delta}(\boldsymbol{\theta}_\circ, \mathcal{X}) = \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\mathcal{X}|\,\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_\circ}$ contains the derivatives of the log-likelihood $p(\mathcal{X}|\boldsymbol{\theta})$

w.r.t. density parameters at $\boldsymbol{\theta}_{\mathrm{o}}$.

- The Fisher information matrix $F = \mathrm{E}\left[\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \cdot)\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \cdot)^{\mathrm{T}}\right]$ encodes the second moments of mappings $\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \cdot)$.

The Fisher mapping on a vector $\mathbf{x}$ has size $G\,(2d+1)$ and can be written as:

$$\begin{aligned}
\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{x}) &= \left[\, \boldsymbol{\delta}_1(\mathbf{x})\,, \cdots,\, \boldsymbol{\delta}_G(\mathbf{x}) \,\right]^{\mathrm{T}}, \text{where} \\
\boldsymbol{\delta}_g(\mathbf{x}) &= \gamma_g(\mathbf{x}) \begin{bmatrix} 1 \\ \mathbf{x} - \boldsymbol{\mu}_g^{\mathrm{o}} \\ \mathrm{diag}\left((x-\boldsymbol{\mu}_g^{\mathrm{o}})\,(x-\boldsymbol{\mu}_g^{\mathrm{o}})^{\mathrm{T}} - \boldsymbol{\Sigma}_g\right) \end{bmatrix} \\
\text{and} \quad \gamma_g(\mathbf{x}) &= \frac{\mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_g^{\mathrm{o}}, \boldsymbol{\Sigma}_g\right)}{\sum_{h\,=\,1}^{G} \omega_h\, \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_h^{\mathrm{o}}, \boldsymbol{\Sigma}_h\right)}\,.
\end{aligned}$$

Then, on a sequence, it is computed [13] as: $\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}\,, \mathcal{X}) = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{x}_t)$. When the total number of parameters is too high, a robust estimation of F would require too much background data. Thus in most applications, F is approximated diagonally.

2.2. Kernels between GMM

Generally speaking, kernels between distributions have an analytical expression for exponential family distributions (see [6] for probability product kernels, and [14] for probabilistic distances). For GMM densities, their computation is generally difficult. But if all GMM share the same weights ω_g and the same covariance matrices $\boldsymbol{\Sigma}_g$, some calculus of integrals can be simplified. In the following, we assume that GMM are trained on sequences by adapting means only, as it is commonly done in speaker verification. $\boldsymbol{\mu}_{\mathrm{X},g}$ denotes the g^{th} component of the GMM adapted on a sequence $\mathcal{X}$.

2.2.1. The correlation kernel

The only probability product kernel which have a closed form for GMM is the correlation kernel [9]:

$$\kappa_{\mathrm{corr}}(\mathcal{X}, \mathcal{Y}) = \int_{\mathbb{R}^d} p(\mathbf{z}|\boldsymbol{\theta}_{\mathrm{X}}) p(\mathbf{z}|\boldsymbol{\theta}_{\mathrm{Y}})\,\mathrm{d}\mathbf{z} \;=\; (2\pi)^{-\frac{d}{2}}\; \boldsymbol{\omega}^{\mathrm{T}}\, \boldsymbol{\Gamma}_{\mathrm{X,Y}}\, \boldsymbol{\omega}\,,$$

where the vector $\boldsymbol{\omega} = [\,\omega_1\,, \cdots,\, \omega_G\,]^{\mathrm{T}}$ contains GMM weights and $\boldsymbol{\Gamma}_{\mathrm{X,Y}}$ is the $G{\times}G$-symmetric matrix with element values

$$\Gamma_{g,h} = \frac{1}{\sqrt{\det(\boldsymbol{\Sigma}_g + \boldsymbol{\Sigma}_h)}}\, e^{-\frac{1}{2}(\boldsymbol{\mu}_{\mathrm{X},g} - \boldsymbol{\mu}_{\mathrm{Y},h})^{\mathrm{T}}\,(\boldsymbol{\Sigma}_g + \boldsymbol{\Sigma}_h)^{-1}\,(\boldsymbol{\mu}_{\mathrm{X},g} - \boldsymbol{\mu}_{\mathrm{Y},h})}$$

A problem with the correlation kernel is that it yields values in a wide range: the mean vectors $\{\boldsymbol{\mu}_{\mathrm{X},g}\,,\, \boldsymbol{\mu}_{\mathrm{Y},g}\}$ can be very close, still $\kappa_{\mathrm{corr}}(\mathcal{X}, \mathcal{Y})$ ca be much lower than $\kappa_{\mathrm{corr}}(\mathcal{X}, \mathcal{X})$ and $\kappa_{\mathrm{corr}}(\mathcal{Y}, \mathcal{Y})$. If no precaution is taken, the classification model may over-fit. A simple way to avoid this problem is to compute the "spherically" normalized kernel

$$\mathring{\kappa}_{\mathrm{corr}}(\mathcal{X}, \mathcal{Y}) = \frac{\kappa_{\mathrm{corr}}(\mathcal{X}, \mathcal{Y})}{\sqrt{\kappa_{\mathrm{corr}}(\mathcal{X}, \mathcal{X})\,\kappa_{\mathrm{corr}}(\mathcal{Y}, \mathcal{Y})}}\,.$$

This normalization adds RBF-like properties to the kernel since $\mathring{\kappa}_{\mathrm{corr}}(\mathcal{X}, \mathcal{X}) = 1$ for all $\mathcal{X}$ (all mappings belong to a hypersphere in the feature space). It does not prevent from over-fitting, but resolves some numerical problems.

We underline here that we did not find in the literature any application of the correlation kernel in speaker verification. This will be carried out in this paper.

2.2.2. Kernel between GMM supervectors

A widely used measure to compare probability distributions is the KL divergence $\mathcal{D}_{\mathrm{KL}}$. In [4] it is shown that the KL divergence between two GMM is upper bounded by a simple analytic expression, which can be written for equally weighted mixtures of Gaussians $\mathcal{N}$ as:

$$\mathcal{D}_{\mathrm{KL}}\left(\sum_{g=1}^{G} \omega_g\, \mathcal{N}_{\mathrm{X},g}\,\middle\|\, \textstyle\sum_{g=1}^{G} \omega_g\, \mathcal{N}_{\mathrm{Y},g}\right) \leq \underbrace{\sum_{g=1}^{G} \omega_g\, \mathcal{D}_{\mathrm{KL}}(\mathcal{N}_{\mathrm{X},g} \| \mathcal{N}_{\mathrm{Y},g})}_{\mathcal{D}_{\mathrm{GMM}}(p_{\mathrm{X}}, p_{\mathrm{Y}})}\,,$$

where the (symmetric) KL divergence between two Gaussians $\mathcal{N}_{\mathrm{X},g}$ and $\mathcal{N}_{\mathrm{Y},g}$ with same covariance $\boldsymbol{\Sigma}_g$ is given by the Mahalanobis distance between mean vectors:

$$\mathcal{D}_{\mathrm{KL}}(\mathcal{N}_{\mathrm{X},g} \| \mathcal{N}_{\mathrm{Y},g}) = \left(\boldsymbol{\mu}_{\mathrm{X},g} - \boldsymbol{\mu}_{\mathrm{Y},g}\right)^{\mathrm{T}} {\boldsymbol{\Sigma}_g}^{-1} \left(\boldsymbol{\mu}_{\mathrm{X},g} - \boldsymbol{\mu}_{\mathrm{Y},g}\right)\,.$$

$\mathcal{D}_{\mathrm{GMM}}$ is actually the square of the euclidean distance between "GMM supervectors" $\boldsymbol{\phi}_{\mathrm{X}}$ [3]. These supervectors are the concatenation of G normalized mean:

$$\boldsymbol{\phi}_{\mathrm{X}} = \begin{bmatrix} \omega_1\, {\boldsymbol{\Sigma}_1}^{-1/2} \boldsymbol{\mu}_{\mathrm{X},1}\;, \\ \cdots, \\ \omega_G\, {\boldsymbol{\Sigma}_G}^{-1/2} \boldsymbol{\mu}_{\mathrm{X},G} \end{bmatrix} \tag{1}$$

Contrarily to the KL exponential embedding $e^{-\gamma \mathcal{D}_{\mathrm{KL}}}$, the GMM-supervector kernel $\kappa_{\mathrm{GMM}}(p_{\mathrm{X}}, p_{\mathrm{Y}}) = e^{-\gamma\, \mathcal{D}_{\mathrm{GMM}}(p_{\mathrm{X}}, p_{\mathrm{Y}})}$ satisfies Mercer conditions, since it is the RBF Gaussian kernel in the feature space defined by (1).

2.3. The GLDS kernel

The GLDS kernel [2] involves a polynomial expansion $\boldsymbol{\phi}_q$, with monomials (between each combination of input vector components) up to a given degree q. For example, if $q = 2$ and $\mathbf{x} = [x_1, x_2]^{\mathrm{T}}$ is a 2-dimensional input vector, then $\boldsymbol{\phi}_q(\mathcal{X}) = [x_1,\; x_2,\; {x_1}^2,\; x_1 x_2,\; {x_2}^2]^{\mathrm{T}}$. The GLDS kernel is given by:

$$\kappa_{\mathrm{GLDS}}(\mathcal{X}, \mathcal{Y}) = \left[\frac{1}{T_{\mathrm{X}}} \sum_{t=1}^{T_{\mathrm{X}}} \boldsymbol{\phi}_q(\mathbf{x}_t)\right]^{\mathrm{T}} {\mathbf{S}_q}^{-1} \left[\frac{1}{T_{\mathrm{Y}}} \sum_{s=1}^{T_{\mathrm{Y}}} \boldsymbol{\phi}_q(\mathbf{y}_s)\right] \tag{2}$$

where $\mathbf{S}_q$ is the second moment matrix of polynomial expansions $\boldsymbol{\phi}_q$ estimated on some background population, or its diagonal approximation for better efficiency. In practice, the GLDS kernel allows only expansion with monomials up to degree 3 because the size of the explicit polynomial expansion $\boldsymbol{\phi}_q$ becomes intractable for polynomial expansions with maximal degree q higher than 3.

2.4. FSNS kernels

An interesting problem then is to find a tractable way to compute or approximate (2) for any q. A more general problem is to provide a finite-dimensional (and tractable) form of (2) for any expansion $\boldsymbol{\phi}$ including infinite ones. By this way, radial basis function (RBF) kernels, which have been proved very efficient in most kernel learning methods, could also be used.

We addressed this problem in [8] by proposing an extension of the GLDS kernel for any expansion $\boldsymbol{\phi}$. This led to the formulation of a rich family of sequence kernels, which we referred to as *Feature Space Normalized Sequence* (FSNS) kernels. Given a vector kernel k, the main step in computing a FSNS kernel is to operate an Incomplete Cholesky Decomposition on the Gram matrix of the background data. This yields a (relatively) small size codebook $\mathcal{C} = \{\mathbf{b}_{I_1}, \ldots, \mathbf{b}_{I_m}\}$ of background data. Then, the FSNS kernel is defined as:

$$\kappa_{\text{FSNS}}(\mathcal{X}, \mathcal{Y}) = \overline{\boldsymbol{\psi}}_{\mathcal{C}}(\mathcal{X})^{\text{T}} \; \mathbf{R} \; \overline{\boldsymbol{\psi}}_{\mathcal{C}}(\mathcal{Y})$$

where $\mathbf{R}$ is a normalization matrix and $\overline{\boldsymbol{\psi}}_{\mathcal{C}}$ is the sequence empirical map on $\mathcal{C}$:

$$\overline{\boldsymbol{\psi}}_{\mathcal{C}}(\mathcal{X}) = \begin{bmatrix} \frac{1}{T_{\text{X}}} \sum_{t=1}^{T_{\text{X}}} k(\mathbf{b}_{I_1}, \mathbf{x}_t) \\ \ldots \\ \frac{1}{T_{\text{X}}} \sum_{t=1}^{T_{\text{X}}} k(\mathbf{b}_{I_m}, \mathbf{x}_t) \end{bmatrix}$$

We refer to [8] for a full description of the theoretical and practical aspects of FSNS kernels.

3. EXPERIMENTS

3.1. Database and front-end processing

All sequences used for development and evaluation come from the NIST'2005 SRE database in "core test" condition, limited to the female population. They include about two minutes of telephone speech pronounced by a same speaker. The development protocol was defined by the Biosecure project [12] and involves distinct sets of speakers by means of audio sequences from NIST 2003 and 2004 SREs. The *background* database includes 283 sequences, that correspond to about 9 hours of speech. Besides, 113 additional sequences which involve other speakers are available: they can be used to compute statistics for score normalization, or to increase impostor accesses for discriminative training. The validation corpus consists in 7062 trials that involve 181 target speaker and 368 test sequences. After having tuned the system to perform as well as possible on the validation set, NIST SRE 2005 is used to measuring the actual Detection Cost Function (DCF). This criterion to minimize is a weighted sum of False Rejection and False Alarm rates: DCF = 0.1FR% + 0.99FA%. We insist on the fact that development, validation and evaluation involve non-overlapping sets of speakers.

We employed a classical front-end processing for speaker verification. To extract acoustic vectors from a speech sequence, 12 MFCC and their first order time derivatives are extracted on 16ms window, at a 10ms frame rate. The derivative of the energy logarithm is also added. Then, a speech activity detector discards silence frames, using an unsupervised bi-Gaussian modeling on the energy level. Finally, the 25-dimensional input vectors are normalized by *feature warping* [10] over 3 seconds windows.

3.2. Implementation of the different systems

The baseline UBM-GMM system is based on the *Alize* speaker verification software [1]. The front-end processing of this generative system is a bit different from the one we use for the SVM systems, since the optimal settings for the two types of systems are different. The cepstral features extracted from the speech signal are also MFCC but are normalized differently: instead of the feature warping, we use a mean/variance normalization on the sequence (so that each vector component to has a zero mean unit variance on the sequence). We also used 2048 components in the GMMs with variance flooring during training, as well as 10-best scoring with T-Norm. As for the SVM systems, we implemented the GLDS system using the algorithm described in [2]. For the FSNS and the other systems (Fisher and Correlation), we refer to [8] and [7] respectively for further details. For the GMM-supervectors system, the observations sequence of each utterance, in train and test, is used to estimate its corresponding GMM density. This is done by MAP (Maximum A Posteriori) adaptation of the initial UBM-GMM. Then the parameters of this adapted GMM are used to build the supervector (1) which is then used as input to the SVM classifier.

3.3. Evaluation

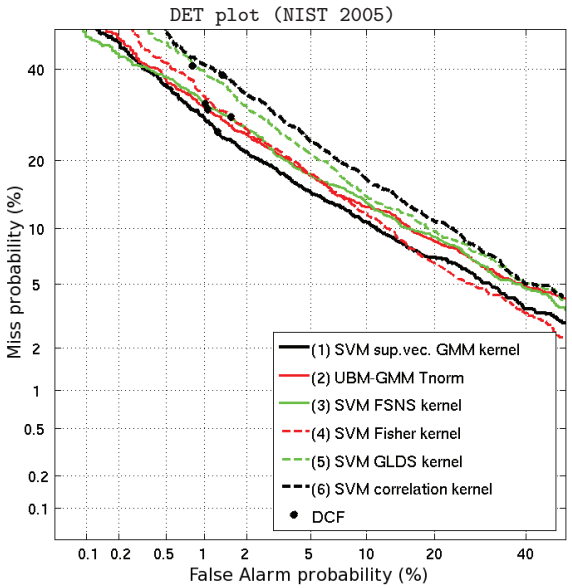
The individual performances of each system are displayed in Fig.1. We first note that the FSNS kernel, which is an extension of the GLDS kernel, leads indeed to better performances than GLDS. We also mention that we compared the GLDS system (with degree $q = 3$) with the FSNS one using the polynomial $k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^3$ as the vector kernel, we obtained the same performances. This shows that our extension is not only theoretically sound, but practically also. Moreover, the results show that FSNS yields similar performances to the conventional UBM-GMM system while it does not exploit any generative probabilistic modeling.

Second, it is clear from the results that the GMM-supervectors system significantly outperforms all the others. This comes to confirm that exploiting generative modeling in a discriminative framework can be very beneficial. It is also clear that the GLDS, Fisher and Correlation systems yield relatively poor performances as compared to the others. We can not however conclude in a strict manner that they are always less performing than the others. We can not indeed claim that we implemented all the systems in the best/optimal way. We can however say that we tried to be as close as possible to state-of-the-art literature.

Most of the best performing systems in NIST'SRE campaigns are fusion of two or more systems. We thus tested all possible fusion combinations between the systems we implemented. The fusion we used is a linear combination of output scores where the weight parameters are set so as to minimize the DCF on the validation set. Improvement with such a simple fusion has been previously observed in speaker verification [2].

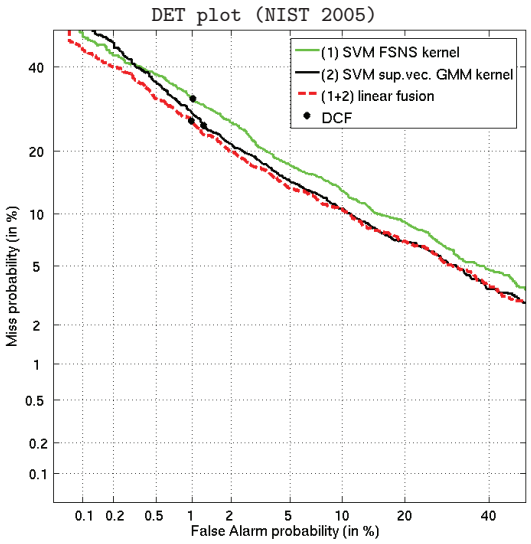
The only fusion that led to a non-negligible improvement is the one between the FSNS and GMM-supervectors systems and is depicted in Fig.2. In particular, the fusion of the UBM-GMM and the GMM-supervectors systems does not improve the performance. This is not surprising, in our opinion, given that both share the same "global" information for classification, that is the MAP adapted GMMs. On the opposite the fusion of the FSNS and GMM-supervectors

significantly improve the performance (particularly in term of DCF), which suggests that the two systems use complementary classification information. It is thus worth to continue the exploration of FSNS kernels in order to better exploit their discriminative potential and their complementarity with GMM-based systems[1].



	EER(%)	DCF(x10^{-3}) min	DCF(x10^{-3}) actual
(1)	10.40	37.4	37.7
(2)	12.06	40.5	40.6
(3)	11.91	41.4	41.6
(4)	11.90	42.5	44.0
(5)	12.54	48.5	48.8
(6)	13.92	50.5	52.1

Fig. 1. Performances of the SVM systems and UBM-GMM



	EER(%)	DCF(x10^{-3}) min	DCF(x10^{-3}) actual
(1)	11.91	41.4	41.6
(2)	10.40	37.4	37.7
(1+2)	10.28	35.3	36.1

Fig. 2. Performance of the linear fusion between FSNS and GMM-supervector

References

[1] J.-F. Bonastre, F. Wils, and S. Meignier. Alize, a free toolkit for speaker recognition. In *Proc. ICASSP*, 2005.

[2] W.M. Campbell and al. Support vector machines for speaker and language recognition. *Computer Speech and Language*, 20:210–229, 2006.

[3] W.M. Campbell, D. Sturim, and D. Reynolds. Support vector machines using gmm supervectors for speaker averification. *Signal Processing Letters*, 13, 2006.

[4] M.N. Do. Fast approximation of kullback-leibler distance for dependence trees and hidden markov models. *Signal Processing Letters*, 10(4):115–118, 2003.

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

[6] T. Jebara and al. Probability product kernels. *Journal of Machine Learning Research*, 5, 2004.

[7] J. Louradour and K. Daoudi. Sequence kernels for svm speaker verification. *Preprint*, 2008.

[8] J. Louradour, K. Daoudi, and F. Bach. Feature space mahalanobis sequence kernels: Application to svm speaker verification. *IEEE Trans. on Audio, Speech and Language Processing*, 15(8):2465–2475, 2007.

[9] S. Lyu. A kernel between unordered sets of data : the gaussian mixture approach. In *Proc. ECML*, 2005.

[10] J. Pelecanos and S. Sridharan. Feature warping for robust speaker verification. In *IEEE Odyssey*, 2001.

[11] M. Seeger. Covariance kernels from bayesian generative models. *Proc. NIPS*, 2002.

[12] Biosecure network of excellence : Biometrics for secure authentification. `http://www.biosecure.info`, 2005.

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

[14] S. Zhou and R. Chellappa. From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel hilbert space. *IEEE Trans. PAMI*, 28(6):917–929, 2006.

[1]Given the lack of space, we could not provide a conclusion. We do apologize for that.