Discrmininant Models for Text-independent Speaker Verification

Johnny Mariéthoz [1]

IDIAP–RR 06-70

November 10, 2006

[1] IDIAP Research Institute, CP 592, 1920 Martigny, Switzerland and Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, marietho@idiap.ch

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 06-70

Discrmininant Models for Text-independent Speaker Verification

Johnny Mariéthoz

November 10, 2006

Abstract

This thesis addresses text-independent speaker verification from a machine learning point of view. We use the machine learning framework to better define the problem and to develop new unbiased performance measures and statistical tests to compare objectively new approaches. We propose a new interpretation of the state-of-the-art Gaussian Mixture Model based system and show that they are discriminant and equivalent to a mixture of linear classifiers. A general framework for score normalization is also given for both probability and non-probability based models. With this new framework we better show the hypotheses made for the well known Z- and T- score normalization techniques.

Several uses of discriminant models are then proposed. In particular, we develop a new sequence kernel for Support Vector Machines that generalizes an other sequence kernel found in the literature. If the latter is limited to a polynomial form the former allows the use of infinite space kernels such as Radial Basis Functions. A variant of this kernel that finds the best match for each frame of the sequence to be compared, actually outperforms the state-of-the-art systems. As our new sequence kernel is computationally costly for long sequences, a clustering technique is proposed for reducing the complexity.

We also address in this thesis some problems specific to speaker verification such as the fact that the classes are highly unbalanced. And the use of a specific intra- and inter-class distance distribution is proposed by modifying the kernel in order to assume a Gaussian noise distribution over negative examples. Even if this approach misses some theoretical justification, it gives very good empirical results and opens a new research direction.

Keywords: Gaussian Mixture Models, Support Vector Machines, loss function, cost, text-independent speaker verification, unbalanced class problem, similarity measure, sequence kernel.

Contents

1 Introduction

There are more and more situations arising where people need a system to store or to exchange their personal informations. The most used solution in such a case, is to use secret codes or personal cards. This is the case for bank accounts, computer passwords, etc. The drawback of these traditional systems occurs when the secret code is lost or stolen. Who never wrote his personal identification number (PIN) code or password somewhere in order not to forget it? An alternative solution is to use biometric information, such as fingerprint, face, iris or voice, that are expected to be somehow unique to each individual, in order to restrict the access of a service to registered clients only. This is called "biometric authentication". In this thesis, we address the problem of biometric authentication using some pre-recorded human voices using an automatic system based on machine learning algorithms.

1.1 What is a Speaker Verification System?

A speaker verification system should verify the claimed identity of a person based on his voice. Basically, it has to accept as a *client* or reject as an *impostor* a speaker that claimed an identity. Different systems can be considered:

- Text-dependent systems: the phonetic content of the pronounced sentence is fixed. For example, the system can ask the speaker to pronounce a specific sentence.
- Text-independent systems: the phonetic content is free.

The former has the advantage to be robust to "replay" attacks (when an impostor plays back a pre-recorded sentence pronounced by the real speaker), but has the drawback that it needs more complex models and is very strict about the sentence pronounced by the speaker. In this thesis, we will consider only text-independent speaker verification systems that are the most used for their simplicity, as they do not require complex speech recognition modules and they are thus better adapted to various embedded applications (phone, personal digital assistant, etc.)

While speaker verification systems have been researched and developed in the last 20 years, it is only more recently that they have benefited from research in machine learning thanks to the computational power of modern computers. Before describing the objectives of this thesis, let us first explain what is machine learning.

1.2 What is Machine Learning?

Machine learning is a research domain at the crossroad of computer science and statistics that consists in developing algorithms that allow computers to improve, "learn", automatically through experience. In order to "learn" a solution to a problem, the algorithm needs some "training" examples corresponding to this problem for which the solution is known. The overall goal is then to find the best function over a selected set of functions, according to a given loss function applied to the training examples. The set of functions should be rich enough to contain a good solution but simple enough in order to "generalize" the concepts underlying the training examples to new, unseen, examples. The size of the chosen set of functions is directly related to a formal concept know as the *capacity* of a set of functions; the solution found by a machine learning algorithm is called a *model* and the set of training examples is called a *dataset*. The machine learning community developed several algorithms that can be applied to various problems, such as speaker verification, face detection, text categorization, etc.

1.3 Road Map of the Thesis

The aim of this thesis is to address the speaker verification problem from a machine learning point of view.

A common problem in machine learning is to classify examples into two categories, the so-called two-class classification problem (Bishop, 1995). The common approaches used to solve a two-class classification problem are either discriminant (trying to find an hyperplane that best separates the two given classes) or not, the latter being often implemented using generative models (that try to estimate separately the distribution of each class, then relying on Bayes rule to take a decision). According to Vapnik (2000), one should never try to solve a more complex task than the one at hand. Hence, discriminant models

C. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

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

should be favored in general. In this thesis, we consider speaker verification as a two-class classification problem for each client, with one class representing the client and the other representing the impostors.

First, we present the text-independent speaker verification systems in Chapter 2 as found in the literature and we note that unfortunately most results are usually presented using biased measures. Chapter 3 thus describes new unbiased measures and statistical tests in order to compare objectively the different proposed approaches. We used machine learning principles to design new speaker verification databases and protocols in order to produce unbiased results. In Chapter 4, we describe all benchmark databases used in this thesis to compare our new approaches to the state-of-the-art systems.

Looking at the current speaker verification literature, it is interesting to note that the dominant state-of-the-art model does not appear to be discriminant as it is based on generative models. In fact, the devil is in the details, as the speaker verification community proposed many modifications in order to reach state-of-the-art performance. When we analyze the resulting system more deeply, as done in Chapter 5, we can see that, due to these modifications, the state-of-the-art system becomes discriminant. But in this case, why not use directly discriminant models? In Chapter 5, we also propose a new generic framework that also includes discriminant models for speaker verification. We also extend this framework to score normalization techniques, that are used to make the decisions taken by a system more robust with respect to different recording conditions.

The speaker verification problem has some specificities that make the application of discriminant models difficult. First, the examples are encoded as variable length sequences of multi-dimensional vectors that depend on the phonetic content of the pronounced sentence and the speech rate of the user. Unfortunately, most discriminant models can only work on fixed size vectors. In Chapter 6, we address this problem by using informations taken from estimated densities in order to produce fixed size vectors that can be used by discriminant models. In Chapter 7, we then propose to use instead a particular discriminant model, called Support Vector Machine (SVM), which projects the examples into a high dimensional space before trying to discriminate the two classes. This projection is done using a specific mathematical function called “kernel” that can in theory be tailored to any kind of data structure. We thus propose in Chapter 7 a new kernel that can handle sequences. The recent speaker verification literature also proposed “sequence” kernels, called Generalized Linear Discriminant Sequence (GLDS) kernels that are limited to a polynomial form. Our approach gives a new enlightenment to these kernels and extend them to other kernel functions such as Gaussian kernels. As our approach is costly for long sequences and thus not applicable in some cases, we propose an approximate method to reduce its complexity.

An other particularity of the problem is that the number of positive examples (coming from the client) and the number of negative examples (coming from the impostors) are highly unbalanced. Indeed, each time the system enrolls a client, it needs records coming from this client. It is usually not realistic, from the application point of view, to ask a client to pronounce sentences several times per day during several months. We thus have only few (often only one) accesses to enroll a client. As we do not have “real” impostor accesses, the records coming from other speakers are used as negative examples, which can be several hundreds. In summary, we have a number of two-class classification problems equal to the number of clients to enroll, with about one positive example and hundred of negative examples for each problem. Fortunately we observed empirically that for all SVM based approaches the problem is separable and thus, as the SVM considers only examples in the margin, the ratio between negative and positive examples is reduced and the solution found by the SVM seems good. Instead, we address an other problem which we consider more important: the variance of the intra-client distance distribution is more peaky than the variance of the intra-impostors distance distribution. We thus propose, in Chapter 8, to create a new similarity measure by modifying the kernel by adding a Gaussian noise distribution around each negative example. Unfortunately, in order to obtain good performance, we have to modify a nice principled approach. Even if the final approach has not yet a good theoretical justification, it is a good starting point for future research.

2 Text-Independent Speaker Verification Systems

Speaker verification systems try to verify the identity of a claimed speaker given a recorded sentence. They are often used to secure personal information as a replacement for password or personal identification number (PIN) code based secure systems. These systems are also increasingly often used to secure personal information 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 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.

Feature extraction is also an important step in the speaker verification procedure. It basically transforms a mono dimensional speech signal into a sequence of multi-dimensional feature vectors. Largely inspired from the speech recognition domain, this is also aimed to discard non speaker frames, such as silence or noise, and keep as much as possible the speaker specific information.

Even if GMMs yield good performance, they try to estimate data density instead of solving the final task: find the decision boundary between a specific client and all possible impostors. Several researchers proposed discriminant approaches but the most interesting one, from our point of view, is based on Support Vector Machines (SVMs). SVMs yield similar or even better per-

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

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

formance than GMMs on several text-independent speaker verification tasks. One of these systems, based on an explicit polynomial expansion proposed by Campbell (2002) has obtained good results during the NIST 2003 evaluation (Campbell et al., 2005). We will retain this approach as a reference system with respect to our new SVM based algorithms.

The outline of this chapter goes as follows. In Section 2.1, we present the commonly used machine learning algorithms in text-independent speaker verification systems. In Section 2.2, a GMM based system, the most well-known, is presented. Section 2.3 is dedicated to the feature extraction procedure including a description of a speech/silence detector algorithm. In Section 2.4 the score normalization procedure is given to make scores robust to unmatched recording conditions. Finally, the SVM based system proposed by Campbell (2002) is described in Section 2.5.

2.1 Machine Learning Tools

Before defining the speaker verification problem and describing the state-of-the-art models, let us define some machine learning algorithms used in speaker verification.

Diagonal Covariance Gaussian Mixture Models

This is probably the most used algorithm to estimate a data density. Given a set of frames $\mathbf{X} = \{\mathbf{x}_1, .., \mathbf{x}_t, .., \mathbf{x}_T\}$, Gaussian Mixture Models can be defined as follows:

$$P(\mathbf{X}|\Theta) = \prod_t \sum_{g=1}^{N_g} w_g \cdot \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_g, \boldsymbol{\sigma}_g) \tag{2.1}$$

with

$$\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_g, \boldsymbol{\sigma}_g) = \frac{1}{\sqrt{2\pi\, \boldsymbol{\sigma}_g^2}} \exp -\frac{(\mathbf{x}_t - \boldsymbol{\mu}_g)^2}{2\, \boldsymbol{\sigma}_g^2} \tag{2.2}$$

where N_g is the number of Gaussians and $\Theta = \{w_g, \boldsymbol{\mu}_g, \boldsymbol{\sigma}_g\}_{g=1}^{N_g}$ are respectively the weight, the mean vector and the standard deviation vector of the g^{th} Gaussian of the mixture. Each off-diagonal element of the covariance matrix is set to zero, which is usually the case in speaker verification systems. Furthermore, all weights are positive and sum to one.

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.

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.

GMMs are generally trained using an iterative Expectation Maximization (EM) algorithm (Dempster et al., 1977) by Maximizing the Likelihood (ML) defined as follows:

$$\hat{\Theta} = \arg\max_{\Theta} P(\mathbf{X}|\Theta). \tag{2.3}$$

Alternatively, a GMM can be trained using a Maximum A Posteriori (MAP) criterion (Gauvain and Lee, 1994). This algorithm has the advantage to put some prior on the parameter distribution. It can be defined as follows:

$$\hat{\Theta} = \arg\max_{\Theta} P(\Theta|\mathbf{X}) = \arg\max_{\Theta} P(\mathbf{X}|\Theta)P(\Theta). \tag{2.4}$$

An implementation of MAP training for client model adaptation consists of using a global parameter to tune the relative importance of the prior distribution which is in this case represented by the generic model corresponding parameters estimated on a large dataset. The main idea of MAP adaptation is to force the adapted model parameters to be close to the prior generic model. The equations for adaptation of the parameters are:

$$\hat{w}_g = \lambda w_g + (1-\lambda)\sum_{t=1}^{T} P(g|\mathbf{x}_t) \tag{2.5}$$

$$\hat{\boldsymbol{\mu}}_g = \lambda \boldsymbol{\mu}_g + (1-\lambda)\frac{\sum_{t=1}^{T} P(g|\mathbf{x}_t)\mathbf{x}_t}{\sum_{t=1}^{T} P(g|\mathbf{x}_t)} \tag{2.6}$$

$$\hat{\boldsymbol{\sigma}}_g^2 = \lambda\left(\boldsymbol{\sigma}_g^2 + \boldsymbol{\mu}_g\boldsymbol{\mu}_g^{'}\right) + (1-\lambda)\frac{\sum_{t=1}^{T} P(g|\mathbf{x}_t)\mathbf{x}_t\mathbf{x}_t^{'}}{\sum_{t=1}^{'} P(g|\mathbf{x}_t)} - \hat{\boldsymbol{\mu}}_g\hat{\boldsymbol{\mu}}_g^{'} \tag{2.7}$$

where $\hat{w}_g$, $\hat{\boldsymbol{\mu}}_g$ and $\hat{\boldsymbol{\sigma}}_g$ are respectively the new weight, mean and covariance matrix of the g^{th} Gaussian, w_g, $\boldsymbol{\mu}_g$ and $\boldsymbol{\sigma}_g$ are the corresponding parameters in the generic model, $P(g|\mathbf{x}_t)$ is the posterior probability of the g^{th} Gaussian (from the client model at the previous iteration), $\lambda \in [0,1]$ is the adaptation factor chosen empirically on a separate validation set and $v^{'}$ denotes the transpose of vector v.

Note that in Equation (2.5) the new mean is simply a weighted sum of the prior mean and new statistics; $(1-\lambda)$ can hence be interpreted as the amount of faith we have in the new statistics.

A. P. Dempster, N. M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1(39):1–38, 1977.

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

Often used as density estimator or clustering algorithm, GMMs are wildly used in speaker verification. As we will see later, some modifications have nevertheless been applied to GMMs in order to reach state-of-the-art performances in speaker verification.

Support Vector Machines

Support Vector Machines (SVMs), as proposed by Vapnik (2000), are more and more often used in machine learning applications such as text classification (Joachims, 2002) and vision (Pontil and Verri, 1998). They have also been used successfully for regression (Kwok, 1998) and multi-class classification problems (Paugam-Moisy et al., 2000). In the context of two-class classification problems, the underlying decision function is:

$$f_\Theta(\mathbf{x}) = b + \mathbf{w} \cdot \Phi(\mathbf{x}) \tag{2.8}$$

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:

$$(\mathbf{w}^*, b^*) = \arg \min_{(\mathbf{w},b)} \frac{\| \mathbf{w} \|^2}{2} + C \sum_{l=1}^{L_{Tr}} \xi_l \tag{2.9}$$

under the constraints:

$$y_l(\mathbf{w}\phi(\mathbf{x}_l) + b) \geq 1 - \xi_l \quad \forall_l \tag{2.10}$$

$$\xi_l \geq 0 \quad \forall_l \tag{2.11}$$

where L_{Tr} is the number of training examples, y_l is the target class label in $\{-1, 1\}$ corresponding to input vector $\mathbf{x}_l$, C is a parameter that trades off the minimization of classification errors (represented by ξ_l) and the maximization of the margin (represented by $\frac{2}{\|\mathbf{w}\|}$), known to possess very good generalization properties. Maximizing the margin is very important in the context of speaker

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

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

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

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

H Paugam-Moisy, A. Elisseeff, and Y. Guermeur. Generalization performance of multiclass discriminant models. In *Int. Joint Conf. on Neural Networks (IJCNN)*, 2000.

verification, since in most cases very few positive examples are available, and the problem is often easily separable.

It can be shown that solving (2.9) enables the decision function to be expressed as a hyperplane defined by a linear combination of training examples in the feature space $\Phi()$. We can thus express (2.8) using the dual formulation as:

$$f_\Theta(\mathbf{x}) = b + \sum_{l=1}^{L_{Tr}} \alpha_l y_l \Phi(\mathbf{x}_l) \cdot \Phi(\mathbf{x}). \tag{2.12}$$

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:

$$f_\Theta(\mathbf{x}) = b + \sum_{l=1}^{L_{Tr}} \alpha_l y_l k(\mathbf{x}_l, \mathbf{x}). \tag{2.13}$$

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) and the polynomial kernels. The former can be defined as:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^2}\right) \tag{2.14}$$

where σ is a hyper-parameter than can be used to tune the capacity of the model, which is a formal measure of the complexity of the set of functions spanned by the SVM (Vapnik, 2000). The polynomial kernel can be defined as:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (a\mathbf{x}_i \cdot \mathbf{x}_j + b)^p \tag{2.15}$$

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

The difficulty to use SVMs for speaker verification is related to the nature of the data: they are variable length sequences. We will see in Chapter 5 which solution can be proposed in order to modify SVMs to accept sequences as input.

2.2 GMM Based System

Given a sentence $\mathbf{X}$ pronounced by a hypothesized speaker S_i, the aim of a text-independent speaker verification system is to decide whether $\mathbf{X}$ has been pronounced by S_i or not. The testing hypothesis is based on two alternatives:

- H0: $\mathbf{X}$ has been pronounced by S_i,

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

- H1: $\mathbf{X}$ has **not** been pronounced by S_i.

Using the Bayes decision rule, we obtain the likelihood ratio as follows:

$$\frac{p(\mathbf{X}|H0)}{p(\mathbf{X}|H1)} \geq \Delta, \text{ accept } H0 \tag{2.16}$$

where $p(\mathbf{X}|H0)$ is the probability density function of the observed speech segment $\mathbf{X}$ given the hypothesis $H0$, $p(\mathbf{X}|H1)$ is the probability density function of the observed speech segment $\mathbf{X}$ given the hypothesis $H1$ and Δ the decision threshold.

These two densities are most often estimated by two Gaussian Mixture Models with diagonal covariances. The model representing $H0$ is called *client model*. $H1$, the model representing the hypothesis that the sentence $\mathbf{X}$ has been pronounced by an impostor, is called *world model* when it is common to all clients. Note that this model is also often referred to as Universal Background Model (UBM) in the literature. This transforms (2.16) as follows:

$$\sum_t \log \frac{\sum_{g=1}^{N_g} w_g \cdot \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_g, \boldsymbol{\sigma}_g)}{\sum_{g=1}^{\bar{N}_g} \bar{w}_g \cdot \mathcal{N}(\mathbf{x}_t; \bar{\boldsymbol{\mu}}_g, \bar{\boldsymbol{\sigma}}_g)} > \log \Delta \tag{2.17}$$

where $\mathbf{x}_t$ is the t^{th} frame of $\mathbf{X}$, N_g is the number of Gaussians of the client model, $\bar{N}_g$ is the number of Gaussians of the world model, $\Theta_+ = \{\boldsymbol{\mu}_g, \boldsymbol{\sigma}_g, w_g\}$ are the GMM parameters for the client model and $\Theta_- = \{\bar{\boldsymbol{\mu}_g}, \bar{\boldsymbol{\sigma}_g}, \bar{w_g}\}$ are the GMM parameters for the world model.

In the context of GMM based speaker verification systems, ML is normally used to train the world model and MAP adaptation is used to train the client model (usually only the mean parameters are modified, weights and standard deviation are the same as for the world model) and 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.

Empirically some constraints have been added to the state-of-the-art. They can be seen somehow as “tricks” or “hacks” in the sense that it is difficult to justify their use other than empirically. They cannot be interpreted as regularization factors or generalization control parameters. There are basically three such “tricks” in baseline systems.

As we will see in more details in Chapter 5, the log likelihood ration (LLR) defined in (2.18) is normalized by the length of the sequence by adding empirically a normalization factor $1/T$. Removing this factor would increase drastically the final error of the system and thus seems to be crucial. This factor

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

transform (2.17) as follows:

$$\mathrm{llr} = \frac{1}{T}\sum_t \log \frac{\sum_{g=1}^{N_g} w_g \cdot \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_g, \boldsymbol{\sigma}_g)}{\sum_{g=1}^{\bar{N}_g} \bar{w}_g \cdot \mathcal{N}(\mathbf{x}_t; \bar{\boldsymbol{\mu}}_g, \bar{\boldsymbol{\sigma}}_g)} > \log \Delta. \tag{2.18}$$

During the estimation of the world model, the variances are constrained to a given minimum. Several methods are used for that purpose, but in our case the minimum is fixed to a given percentage of the the global variance of the data. Since a typical value for the variance flooring is between 10% to 60% of the global variance of the data for **each** Gaussian, it cannot be considered only as a regularization parameter to avoid numerical problem during the EM training. The estimated distribution is thus forced to be flatten, which is in contradiction with the density estimation hypothesis, but nevertheless gives very good performance.

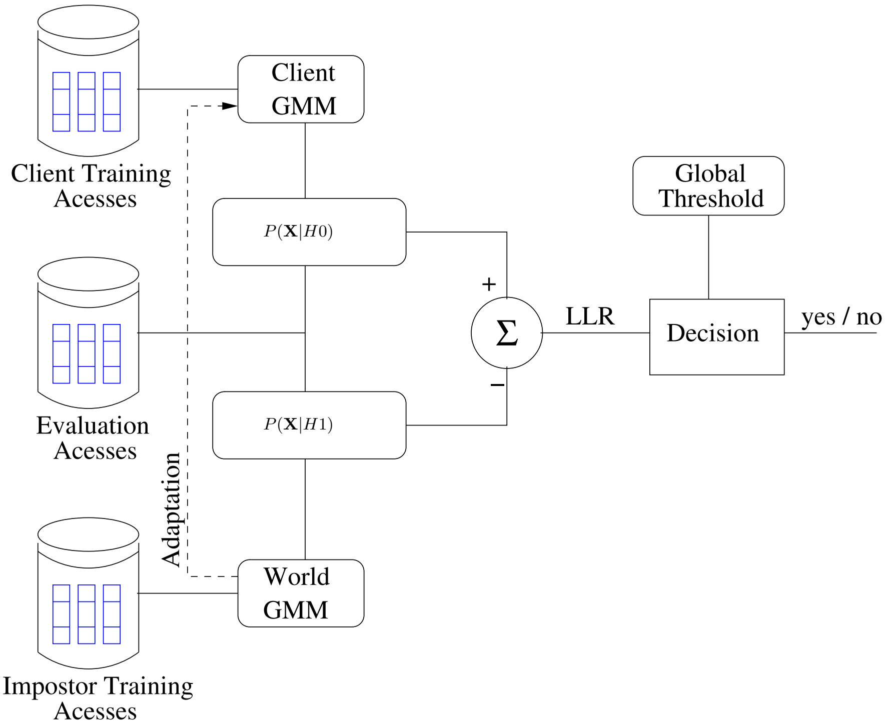


Figure 2.1. A summary of a state-of-the-art GMM based system.

Finally, the use of the MAP adaptation method is often justified by the fact that very few training examples are available for each client. Unfortunately, this justification is contradicted by the fact that MAP adaptation is still better than ML even when plenty of client training data is available, such as in the extended task of the NIST contest. As described in Chapter 5, our explanation is related to the fact that the “a priori” model used to adapt the client model is the same than the one used as normalization model in the decision function.

Figure 2.1 shows an overview of a state-of-the-art GMM based system.

2.3 Feature Extraction

The feature extraction step transforms a recorded speech signal into a set of feature vectors. The resulting data representation is more suitable for statistical models but probably also for discriminant models.

Inspired from the speech recognition domain, most choices of feature extraction parameters come from the last 10 years of experiments, done with HMMs or GMMs. Even if the parameters of the feature extraction procedure have been selected for statistical models, they can (and will) be also used on discriminant models, for simplicity reasons.

While, in this thesis, we refer to $\mathbf{X}$ as the sentence pronounced by the speaker, this is in fact a set of feature vectors obtained by the transformation described in the following.

Cepstral Parameters

In Figure 2.2 the feature extraction procedure is sketched. The aim is to convert a raw speech signal into a set of Cepstral Vectors. First, the speech signal is pre-emphasized. A filter is used to enhance the high frequency of the spectrum as follows:

$$\mathbf{x}_p(t) = \mathbf{x}(t) - a \cdot \mathbf{x}(t-1). \tag{2.19}$$

Values of a are generally between 0.95 and 0.98. As we would like apply a *Fast Fourier Transform* (FFT), the signal must be stationary. Thus we make the hypothesis that the signal is short-term stationary. We use a subpart of the signal by applying a sliding window. The length of this window is usually between 20 and 30 milliseconds. To smooth the windowing procedure, we overlap the window every 10 milliseconds typically. A vector computed for a given window will be called *frame*. As the FFT is sensible to side effects, Hamming window is preferred to rectangular window to smooth the transitions. The FFT is computed using typically 512 points and only the real part of it is retained. The resulting spectrum is composed of 256 points.

In order to reduce the size of the spectrum, it is multiplied by a filter bank. This is a series of band-pass filters, usually triangular. The center frequency of each filter is linearly distributed over the frequency scale. Some authors (see for instance Reynolds and Rose (1995)) use a Mel scale which

D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions On Speech and Audio Processing*, 3 (1), 1995.

corresponds to the auditory scale . In our case we chose 24 triangular filter-banks linearly distributed. To obtain the Spectral coefficients, we take the log of the spectral envelope and multiply each coefficient by 20. Finally a *Discrete Cosine Transform* DCT is applied as follows:

$$c_n = \sum_{i=1}^{N_{sp}} U_i \cos\left[n(i-\frac{1}{2})\frac{\pi}{N_{sp}}\right], n = 1, 2, ..., N_{cc} \tag{2.20}$$

where N_{sp} is the number of log-spectral coefficients, U_i are the log-spectral coefficients values, and N_{cc} is the number of Cepstral coefficients to calculate ($N_{cc} \leq N_{sp}$).

The log followed by a DCT is somehow an inverse FFT and usually makes the coefficients more suitable for Gaussian based models, such as GMMs.

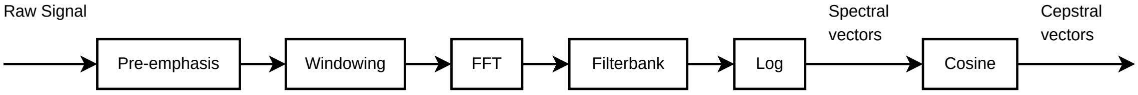


Figure 2.2. Modular Representation of a Filter-bank-based Cepstral parameterization.

Additional Transformations

The first Cepstral coefficient, often called c_0 is similar to the energy of the signal for a given window. In our case this coefficient is replaced by the log-energy.

Most of the models used in text-independent speaker verification do not use explicitly temporal information. However, it is possible to include short term temporal information by using dynamic features such as are the first derivative parameters computed as follows:

$$d_t = \frac{\sum_{i=1}^{W} W\,(c_{t+i} - c_{t-i})}{2\sum_{j=1}^{W} j^2} \tag{2.21}$$

where c_t are the Cepstral coefficients and W the window size to compute the derivative coefficients. A common value for W is 2. This is a polynomial approximation of the derivative. Some authors also use the second derivative coefficients, which can be obtained by re-applying the derivative transformation to the first derivative coefficients. In our experiments, this approach does not improve the results and thus will not be used. The d_t coefficients are simply concatenated to the c_t coefficients.

In order to compensate for the distortion of the acquisition system (channel effect), Cepstral Mean Subtraction (CMS) is often apply. CMS consists in removing the average value computed over the complete sequence for each coefficient. In addition to CMS, the Cepstral parameters can also be reduced: the variance over the complete sequence is equal to one. Note than the energy coefficient is not normalized. Its value is useful to discard silence frames and will be removed after the silence/speech detector, as it is more related to the distance between the speaker and the microphone than the speaker itself.

Silence/Speech Detector

A recording sequence contains some frames pronounced by the speaker and some frames containing noise. In order to take a robust decision, the silence frames must be discarded. Silences may appear before or after the sentence but also in between words. In order to decide whether a frame contains speaker information or not, several techniques can be used. The simplest is to fix a threshold and reject all frames for which the energy coefficient is lower that this threshold. From our point of view, this approach has some limitations: how to estimate the correct threshold. Why to limit this method to the energy coefficient?

Our approach is similar to that described in (Magrin-Chagnolleau et al., 2001) and consists in training a GMM with two Gaussians using the complete set of feature vectors. This training is unsupervised in the sense that we do not use any frame label (that would say whether a frame is silence or speech). Based on the hypothesis that the speech contains more energy than the silence, the Gaussian with the highest energy coefficient will be labelled as speech and the other as silence. This model is trained on each new sequence. An alternative consists to train a prior model using few sequences and adapt it using a MAP algorithm similar to (2.6) for each new sequence. To decide if a new frame is speech or not the ML criterion is used. This approach seems more robust compared to the simple energy based system.

After all these transformations, in our case, we obtain a variable length sequence of vectors of dimension 33 each.

2.4 Score Normalization

The last step of a speaker verification system is to compare the score to a decision threshold. If this score is higher than the decision threshold, the decision is “accept” otherwise “reject”. Estimating a good decision threshold is still an open problem and is generally tuned empirically. As very few client training accesses are available, the decision threshold Δ is common to all the

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

speakers. Thus the decision should be robust to the speaker and access variability. Several causes can make a pronounced sentence by a speaker variable:

The intra-sentence variability: phonetic contents, channel transmission effect.

The intra-speaker variability: quality of the training examples, emotion, state, health, time.

The inter-speaker variability: gender, age, speaking rate, accents.

Score normalization procedures try to increase the robustness to the access variability. Originally proposed by Li and Porter (1988), most normalization procedures are of the form:

$$\widehat{\mathrm{llr}}(\mathbf{X}) = \frac{\mathrm{llr}(\mathbf{X}) - \mu}{\sigma} \tag{2.22}$$

where $\widehat{\mathrm{llr}}(\mathbf{X})$ is the new normalized score, $\mathrm{llr}(\mathbf{X})$ is the original score, μ, σ some parameters to estimate.

Several normalization techniques to estimate μ and σ have been proposed in the literature. We propose to describe here the two most well known: the T-norm and the Z-norm.

T-norm

The T-norm, as introduced in (Auckenthaler et al., 2000) and (Navratil and Ramaswamy, 2003), estimates μ and σ as the mean and the standard deviation of LLRs using models of a subset of impostors, for a particular test access $\mathbf{X}_0$:

$$\mu_M = \frac{1}{M}\sum_m \mathrm{llr}_m(\mathbf{X}_0) \tag{2.23}$$

$$\sigma_M = \sqrt{\frac{1}{M}\sum_m (\mathrm{llr}_m(\mathbf{X}_0) - \mu_M)^2} \tag{2.24}$$

where M is the number of impostor models and llr_m is the score for the m^{th} impostor model for the particular access X_0. Using (2.23) we obtain:

$$\mathrm{llr}_{i_{T-norm}} = \frac{\mathrm{llr}_i - \mu_M}{\sigma_M} > \Delta\ . \tag{2.25}$$

Kung-Pu Li and J. E. Porter. Normalizations and selection of speech segments for speaker recognition scoring. In *Proceedings of the IEEE ICASSP*, pages 595–597, 1988.

R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10:42–54, 2000.

J. Navratil and Ganesh N. Ramaswamy. The awe and mystery of t-norm. In *Proc. of the European Conference on Speech Communication and Technology*, pages 2009–2012, 2003.

This method is often referred to as *utterance based* approach and tried to reduce the variability related to the test accesses. This approach provides usually good improvement, but is quite costly.

Z-norm

The basis of Z-norm (Auckenthaler et al., 2000) is to test a speaker model against example impostor utterances and use the corresponding LLR scores to estimate a speaker specific mean and standard deviation:

$$\begin{aligned} \mu_J &= \frac{1}{J}\sum_j \mathrm{llr}_{\mathrm{S}_i}(\mathrm{X}_j) && (2.26) \\ \sigma_J &= \sqrt{\frac{1}{J}\sum_j (\mathrm{llr}_{\mathrm{S}_i}(\mathrm{X}_j) - \mu_J)^2} && (2.27) \end{aligned}$$

where J is the number of impostor accesses and S_i the i^{th} speaker.

Z-norm is often referred to as *model based* approach and tried to be robust to the model variability. This approach is especially efficient when the training material for each client model is different. The parameters μ_J and σ_J can be estimated during the training phase and thus no additional time is needed during the client authentication.

2.5 SVM and GLDS Kernel

Several SVM based approaches have been proposed recently to tackle the speaker verification problem (Wan and Renals, 2005) and (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 even some contain no information (like silence frames). Most

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

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.

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 in training generative models.

The explicit polynomial expansion approach (Campbell, 2002) 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 ($\Phi(\mathbf{x}) = \mathbf{x}$). This kernel, called GLDS (Generalized Linear Discriminant Sequence), can be expressed as:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \Phi(\mathbf{X}_i)\boldsymbol{\Psi}^{-1}\Phi(\mathbf{X}_j) \tag{2.28}$$

where $\boldsymbol{\Psi}$ is a matrix derived by the metric of the feature space induced by $\Phi()$. This matrix is usually a diagonal approximation $\boldsymbol{\psi}$ of the covariance matrix computed over all the training data. We furthermore define:

$$\Phi(\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T}\phi(\mathbf{x}_t) \tag{2.29}$$

and

$$\tilde{\phi}(\mathbf{x}_t) = \frac{\phi(\mathbf{x}_t)}{\sqrt{\boldsymbol{\psi}}} \tag{2.30}$$

where $\tilde{\phi}()$ is the normalized version of $\phi()$, the fraction represents a term by term division of two vectors and the square root of a vector is a vector of the square root of its elements. We can thus rewrite (2.28) as:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{T_i}\sum_{t_i=1}^{T_i}\tilde{\phi}(\mathbf{x}_{t_i}) \cdot \frac{1}{T_j}\sum_{t_j=1}^{T_j}\tilde{\phi}(\mathbf{x}_{t_j}) \tag{2.31}$$

where $\tilde{\phi}()$ maps the example $\mathbf{x}_t \in \mathbb{R}^d \rightarrow \mathbb{R}^{N_f}$, $N_f = \frac{(d+p-1)!}{(d-1)!p!}$ 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

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

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.

value $n \in \{1, ..., N_f\}$ of the expanded vector corresponds to a combination of $r_1, r_2, ..., r_d$ as follows:

$$\phi'_{k(r_1,r_2,\ldots,r_d)}(\mathbf{x}_t) = \frac{1}{\sqrt{\psi_n}} x_1^{r_1} x_2^{r_2} \ldots x_d^{r_d} \tag{2.32}$$

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

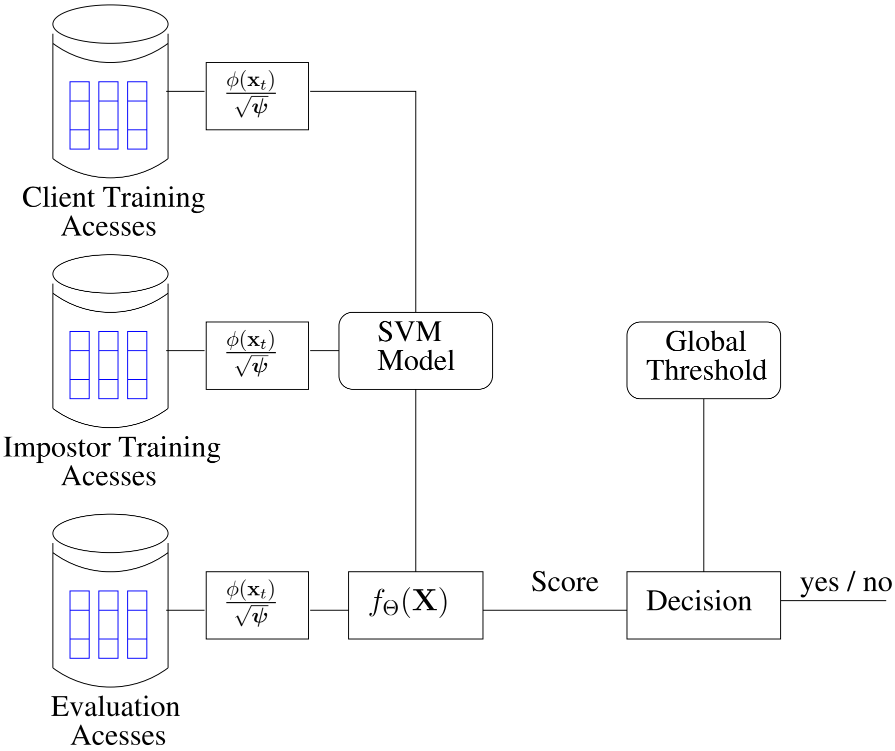

$\frac{\phi(\mathbf{x}_t)}{\sqrt{\psi}}$
Client Training Accesses
SVM Model
$\frac{\phi(\mathbf{x}_t)}{\sqrt{\psi}}$
Impostor Training Accesses
Global Threshold
$\frac{\phi(\mathbf{x}_t)}{\sqrt{\psi}}$
$f_\Theta(\mathbf{X})$
Score
Decision
yes / no
Evaluation Accesses

Figure 2.3. A summary of a state-of-the-art GLDS SVM based system.

Campbell proposed a method to normalize each expanded coefficient using $\boldsymbol{\psi}$ computed over all concatenated impostor sequences. Once all vectors are computed and normalized, they can be used as input to a linear SVM. The output of the SVM is compared to a decision threshold in order to accept or reject an access. This method is quite fast and robust, but is limited to the polynomial form.

Figure 2.3 summarizes the state-of-the-art GLDS SVM based system.

2.6 Conclusion

In this chapter, we have presented different state-of-the-art systems as found in the literature. In Chapter 4, we will present experimental results obtained by these models on the chosen benchmark databases. For a deeper analysis of these algorithms, we kindly invite the reader to go to Chapter 5.

At first, the performance measures are described in Chapter 3, because we think that they are especially important and often badly used in that domain. We thus dedicate a whole chapter to define new measures and to clearly explain how to use them in the speaker verification domain.

3 Performance Measures for Speaker Verification

Every time a researcher proposes a new idea or a new model to solve a given task, he needs to validate his approach using empirical data. In order to estimate the quality of a system, empirical measures such as numbers or curves are often used. They can be used for instance to estimate the expected performance on a new dataset coming from the same distribution as the one used to estimate the model, or to compare two different approaches.

In person authentication, several measures are commonly used as performance measures, such as equal error rate, half total error rate or detection cost functions. Even if the community made large efforts to make these measures standard in the speaker verification domain, for example during NIST evaluation (Martin and Przybocki, 2000), the published results in the scientific literature are most of the time optimistically biased. Too often, models are compared with some parameters estimated on the same examples as those used to estimate the performance measure. The estimation of these parameters are not trivial and the robustness of the models to the decision threshold for example, can be very variable. The machine learning framework proposes several tools to provide unbiased results, such as k-fold cross-validation or train - development - test set approaches. We will see in this chapter that this framework can be applied directly to performance measures such as half total error rate and also to new proposed curves called "expected performance curves".

Moreover, a single error value is difficult to assess without some form of confidence interval. In fact, as the quantity of available data to estimate the quality of a system is limited, the measures can vary depending on the size of the chosen dataset. It is thus important to give an interval around a given error, or a confidence value based on the hypothesis that two models are different, for example. Statistics provides tools such as proportion tests that can be used

A. Martin and M. Przybocki. The NIST 1999 speaker recognition evaluation - an overview. *Digital Signal Processing*, 10:1–18, 2000.

to compute these intervals. Strangely enough, almost nobody use this kind of tests in their research papers or if they do, the tests are often not correctly used. We thus provide a solution to apply a proportion test to the speaker verification domain.

The outline of this chapter goes as follows. In Section 3.1, we present the common measures in speaker verification and show their limitations. In Section 3.2, we present a new family of curves designed to compare systems. Section 3.3 is dedicated to the adaptation of the proportion test for speaker verification systems. Finally, in Section 3.4, we summarize the performance measures and the methodology used in this thesis.

3.1 Common Measures

A verification system has to deal with two kinds of events: either the person claiming a given identity is the one who he claims to be (in which case, he is called a *client*), or he is not (in which case, he is called an *impostor*). Moreover, the system may generally take two decisions: either *accept* the *client* or *reject* him and decide he is an *impostor*. From a machine learning point of view a client access can be labelled as 1 and an impostor as -1.

Let us thus consider two-class classification problems defined as follows: given a training set of examples (x_i, y_i) where x_i represents the input and y_i is the target class $\in \{-1, 1\}$, we are searching for a function $f(\cdot)$ and a threshold Δ such that

$$f(x_i) > \Delta \text{ when } y_i = 1 \text{ and } f(x_i) <= \Delta \text{ when } y_i = -1, \quad \forall i \,. \tag{3.1}$$

		Desired Class	
		1	-1
Obtained Class	1	TP	FP
	-1	FN	TN

Table 3.1. Types of errors in a 2-class classification problem.

The obtained function $f(\cdot)$ (and associated threshold Δ) can then be tested on a separate test data set and one can count the number of utterances of each possible outcome: either the obtained class corresponds to the desired class, or not. In fact, one can decompose these outcomes further, as exposed in Table 3.1, in 4 different categories: *true positives* (where both the desired and the obtained classes are 1), *true negatives* (where both the desired and the obtained classes is 1), *false positives* (where the desired class is -1 and the obtained class is 1), and *false negatives* (where the desired class is 1 and the obtained class is -1). Let TP, TN, FP and FN represent respectively the *number of utterances* of each of the corresponding outcomes in the data set.

Note once again that TP, TN, FP, FN and all other measures derived from them are in fact dependent both on the obtained function $f(\cdot)$ and the threshold Δ. In the following, we will sometimes refer to, say, FP by FP(Δ) in order to specifically show the dependency with the associated threshold.

In speaker verification, false positives and false negatives are respectively referred as *false acceptance* and *false rejection*.

Note that in most benchmark databases used in the authentication literature, there is a significant unbalance between the number of client accesses and the number of impostor accesses. This is probably due to the relatively higher cost of obtaining the former with respect to the latter. In order to be independent on the specific dataset distribution, the performance of the system is often measured in terms of rates of these two different errors, as follows:

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{FP{+}TN}} = \frac{\mathrm{FP}}{\mathrm{NN}}\ , \quad \mathrm{FRR} = \frac{\mathrm{FN}}{\mathrm{FN{+}TP}} = \frac{\mathrm{FN}}{\mathrm{NP}} \tag{3.2}$$

where NP is the number of true client (positive) examples, NN is the number of impostors (negative) examples, FAR is the false acceptance rate and FRR the false rejection rate. Based on these two kinds of errors, we need to define some measures to estimate the performance of a given system on unseen client and impostor accesses. These measures will be denoted hereafter “a posteriori” measures, when the decision threshold is set using the already seen examples and “a priori” measures when the decision threshold is set using unseen examples. The “a posteriori” measures should be used only for analysis purposes and not for comparison purposes.

A often used unique measure combines these two ratios into the so-called *detection cost function* (DCF) (Martin and Przybocki, 2000) as follows:

$$\mathrm{DCF} = \left\{ \begin{array}{l} \mathrm{Cost(FN)} \cdot P(\mathrm{client}) \cdot \mathrm{FRR} \\ +\mathrm{Cost(FP)} \cdot P(\mathrm{impostor}) \cdot \mathrm{FAR} \end{array} \right. \tag{3.3}$$

where P(client) is the prior probability that a client will use the system, P(impostor) is the prior probability that an impostor will use the system, Cost(FR) is the cost of a false rejection, and Cost(FA) is the cost of a false acceptance. These two costs depend on the application at hand.

A particular case of the DCF is known as the *half total error rate* (HTER) where the costs are equal to 1 and the probabilities are 0.5 each:

$$\mathrm{HTER} = \frac{\mathrm{FAR} + \mathrm{FRR}}{2}\ . \tag{3.4}$$

A. Martin and M. Przybocki. The NIST 1999 speaker recognition evaluation - an overview. *Digital Signal Processing*, 10:1–18, 2000.

Most authentication systems are measured and compared using HTER or variations of it.

In the literature, we also often encounter a measure called equal error rate (EER) which corresponds to the threshold nearest to a solution such that FAR = FRR, often estimated as follows:

$$\Delta^\star = \arg\min_{\Delta} |\mathrm{FAR}(\Delta) - \mathrm{FRR}(\Delta)| \text{ and } \mathrm{EER} = \mathrm{FAR}(\Delta) = \mathrm{FRR}(\Delta). \quad (3.5)$$

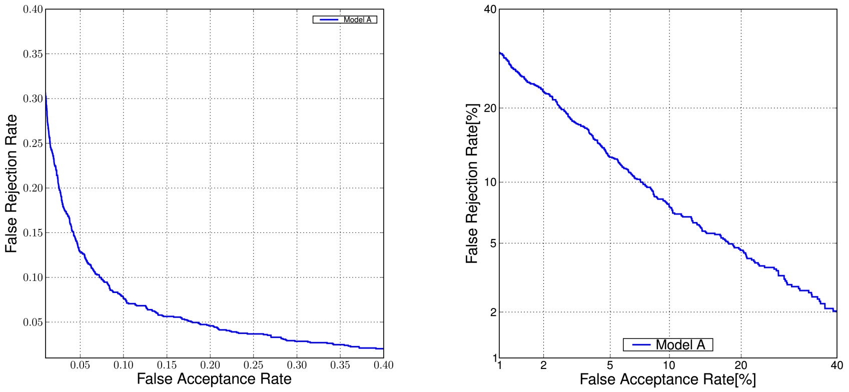
One has to note that this measure is an “a posteriori” measure and should only be used as a criterion to select a decision threshold and not to compare systems, because the exact decision threshold value that reaches the equal error rate in test (unseen) data cannot be known in advance. Only an estimation of it can be found and $\mathrm{FAR}_{test} \neq \mathrm{FRR}_{test}$. Often HTER and EER are similar and both measures are often used as criterion to select the threshold. However, as HTER can fall in a local minimum, EER seems to be more robust and will thus be used in the following.

In most cases, the system can be tuned using a decision threshold in order to obtain a compromise between either a small FAR or a small FRR. There is thus a trade-off which depends on the application: it might sometimes be more important to have a system with a very small FAR, for high security systems, while in other situations it might be more important to have a system with a small FRR, for domestic applications such as games for example. In order to see the performance of a system with respect to this trade-off, we usually plot the so-called Receiver Operating Characteristic (ROC) curve, which represents the FRR as a function of the FAR (Van Trees, 1968) (hence, the curve which is nearer the $(0, 0)$ coordinate is the best ROC curve). Figure 3.1(a) shows an example of a typical ROC. Other researchers have also proposed the DET curve (Martin et al., 1997), which is a non-linear transformation of the ROC curve in order to make results easier to be compared. The non-linearity is in fact a normal deviate, coming from the hypothesis that the scores of client accesses and impostor accesses follow a Gaussian distribution. If this hypothesis is true, the DET curve should be a line. Figure 3.1(b) shows an example of typical DET curve. Note that Figures 3.1(a) and 3.1(b) are computed for the same system. As we will see in the following, these curves make the implicit assumption that the decision threshold estimation is perfect. We can say that these curves are somehow “a posteriori” curves and thus cannot be use to

H. L. Van Trees. *Detection, Estimation and Modulation Theory, vol. 1.* Wiley, New York, 1968.

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

compare two systems; we thus propose instead a new kind of curve, called expected performance curves.



(a) A typical ROC curve.

(b) A typical DET curve.

Figure 3.1. Comparison between DET and ROC curve for the same system.

3.2 Expected Performance Curve

ROC curves are used in several domains, such as text categorization, biometric authentication, medical studies, etc. To be domain independent we need to redefine in a general framework the measures used in these domains.

Several tasks are in fact specific incarnations of 2-class classification problems. However, often for historical reasons, researchers specialized in these tasks have chosen different methods to measure the quality of their systems. In general the selected measures come by pair, which we will call generically here $V1$ and $V2$, and are simple antagonist combinations of TP, TN, FP and FN as defined in Table 3.1. Moreover, a unique measure (V) often combines $V1$ and $V2$. For instance,

- in the domain of person authentication (Verlinde et al., 2000) as we have already seen, the chosen measures are

$$V1 = \frac{\text{FP}}{\text{FP} + \text{TN}} \text{ and } V2 = \frac{\text{FN}}{\text{FN} + \text{TP}}. \tag{3.6}$$

  Several aggregate measures have been proposed, the simplest being the (HTER)

$$V = \frac{V1 + V2}{2} = \frac{\text{FAR} + \text{FRR}}{2} = \text{HTER} \; ; \tag{3.7}$$

[9] P. Verlinde, G. Chollet, and M. Acheroy. Multi-modal identity verification using expert fusion. *Information Fusion*, 1:17–33, 2000.

- in the domain of text categorization (Sebastiani, 2002),

  $$V1 = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \text{ and } V2 = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{3.8}$$

  and are called *precision* and *recall* respectively. Again several aggregate measures exist, such as the *F1* measure

  $$V = \frac{2 \cdot V1 \cdot V2}{V1 + V2} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} = F1 \; ; \tag{3.9}$$

- in medical studies,

  $$V1 = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \text{ and } V2 = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \tag{3.10}$$

  and are called *sensitivity* and *specificity* respectively (Zweig and Campbell, 1993).

In all the cases, in order to use the system effectively, one has to select the threshold Δ according to some criterion which is in general of the following generic form

$$\Delta^\star = \arg\min_\Delta g(V1(\Delta), V2(\Delta)) \; . \tag{3.11}$$

Examples of $g(\cdot,\cdot)$ are the HTER and *F1* functions already defined in equations (3.7) and (3.9) respectively. However, the most used criterion is called the *break even point* (BEP) also sometimes called equal error rate (EER) when $V1$ and $V2$ are error rates and corresponds to the threshold nearest to a solution such that $V1 = V2$, often estimated as follows:

$$\Delta^\star = \arg\min_\Delta |\mathrm{V1}(\Delta) - \mathrm{V2}(\Delta)| \; . \tag{3.12}$$

Note that the choice of the threshold can have a significant impact in the resulting system: in general Δ represents a trade-off between giving importance to $V1$ or $V2$. Hence, instead of committing to a single operating point, an alternative method is to present results by using ROCs. Note that the original ROC plots the true positive rate with respect to the false positive rate, but several researchers use the name ROC with various other definitions of $V1$ and $V2$.

Figure 3.2 shows an example of two ROC curves. Note that depending on the precise definition of $V1$ and $V2$, the best curve would tend to one of the four corners of the graph. In Figure 3.2, the best curve corresponds to the one nearest to the bottom left corner (corresponding to simultaneous small values of $V1$ and $V2$).

F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

M.H. Zweig and G. Campbell. ROC plots: a fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, 39(4):561–577, 1993.

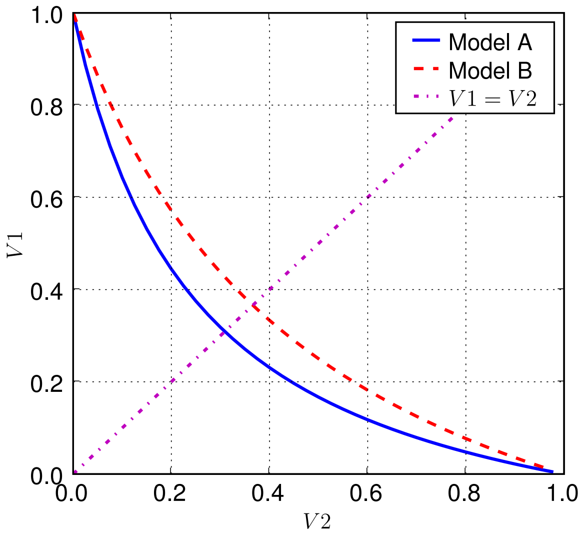


Figure 3.2. Example of two ROC curves with the BEP line.

Instead of providing the whole ROC, researchers often summarize it by some typical values taken from it; the most common summary measure is computed by using the BEP, already defined in equation (3.12), which produces a single value of Δ and to produce some aggregate value $V(\Delta)$ (such as *F1* or HTER). On Figure 3.2, the line intersecting the two ROCs is the BEP line and the intersections with each ROC correspond to their respective BEP point.

Cautious Interpretation of ROC and BEP

As explained above, researchers often use ROC and BEP to present and compare their results; for example, all results presented in (Sebastiani, 2002), which is a very good survey of text categorization, are presented using the BEP; a recent and complete tutorial on text independent speaker verification (Bimbot et al., 2004) proposes to measure performance through the use of DET curves, as well as the error corresponding to equal error rate, hence the BEP. We would like here to draw the attention of the reader to some potential risk of using ROC or BEP for comparing two systems, as it is done for instance in Figure 3.2, where we compare the test performance of models A and B. As can

F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovsk-Delacrétaz, and D. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Applied Signal Processing*, 4:430–451, 2004.

be seen on this Figure, and reminding that in this case $V1$ and $V2$ must be minimized, the best model appears to always be model A, since its curve is always below that of model B. Moreover, computing the BEP of models A and B yields the same conclusion.

Let us now remind that each point of the ROC corresponds to a particular setting of the threshold Δ. However, in real applications, Δ needs to be decided prior to seeing the test set. This is in general done using some criterion of the form of equation (3.11) such as searching for the BEP, equation (3.12), using some development data (obviously different from the test set).

Hence, assuming for instance that one decided to select the threshold according to (3.12) on a development set, the obtained threshold may not correspond to the BEP on the test set. There are many reasons that could yield such mismatch, the simplest being that assuming the test and development sets to come from the same distribution but be of fixed (non-infinite) size, the estimate of (3.12) on one set is not guaranteed to be the same as the estimate on the other set.

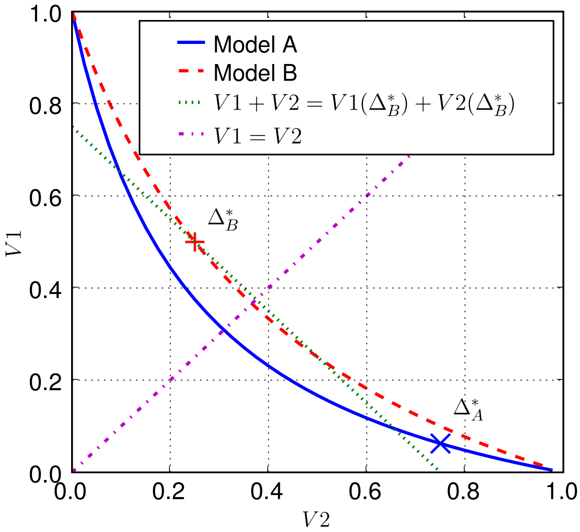
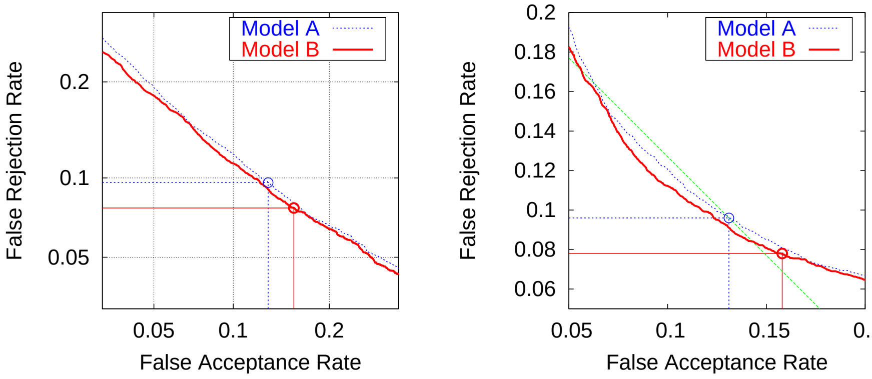


Figure 3.3. Two ROC curves of two different models with their own decision threshold learnt by minimizing the BEP.

Let us call $\Delta_A^\star$ the threshold estimated on the development set using model A and similarly for $\Delta_B^\star$. While the hope is that both of them should be aligned, on the test set, with the BEP line, there is nothing, in theory, that prevents them to be slightly or even largely far from it. Figure 3.3 shows such an example, where indeed,

$$V1(\Delta_B^\star) + V2(\Delta_B^\star) < V1(\Delta_A^\star) + V2(\Delta_A^\star) \tag{3.13}$$

even though the ROC of model A is always below that of model B, including at the intersection with the BEP line. One might argue that this may only rarely happen, but we have indeed observed this scenario several times in person authentication and text categorization tasks, including a text independent speaker verification application where the problem is described in more details in (Bengio and Mariéthoz, 2004). We replicate in the right side of Figure 3.4 the ROCs and in the left side, the DETs obtained on this task using two different models, with model B apparently always better than model A. However, when selecting the threshold on a separate validation set (hence simulating a real world life situation), the HTER of model A (0.111) becomes lower than that of model B (0.112), the graph shows the operating points selected for the two models.



(a) DET curves.

(b) ROCs curves.

Figure 3.4. Curves of two real models for a Text-Independent Speaker Verification task with their corresponding “a priori”operating points.

In summary, showing ROCs has potentially the same drawbacks and risks as showing the training error (indeed, one parameter, the threshold, has been implicitly tuned on the test data). One can expect that it reflects the expected generalization error, but this is true when the size of the data is huge, and false in the general case. Furthermore, real applications often suffer from an additional mismatch between training and test conditions which should be reflected in the used measure.

S. Bengio and J. Mariéthoz. The expected performance curve: a new assessment measure for person authentication. In *Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop*, 2004.

Expected Performance Curve: an “a priori” Performance Curve

We have seen in Section 3.1 that given the trade-off between $V1$ and $V2$, researchers often prefer to provide a curve that assesses the performance of their model for all possible values of the threshold. On the other hand, we have seen that ROCs can be misleading since selecting a threshold prior to seeing the test set (as it should be done) may end up in obtaining a different trade-off in the test set. Hence, we would like here to propose the use of new curves which would let the user select a threshold according to some criterion, in an unbiased way, and still present a range of possible expected performances on the test set. We shall call these curves Expected Performance Curves (EPC).

General Framework

The general framework of EPCs is the following. Let us define some parametric performance measure $\mathcal{C}(V1(\Delta, D), V2(\Delta, D); \gamma)$ which depends on a trade-off parameter γ as well as $V1$ and $V2$ computed on some data D for a particular value of the decision threshold Δ. Examples of $\mathcal{C}(\cdot, \cdot; \gamma)$ are the following:

- in person authentication, one could use for instance

$$\begin{aligned} &\mathcal{C}(V1(\Delta, D), V2(\Delta, D); \gamma) \\ &\quad = \mathcal{C}(\mathrm{FAR}(\Delta, D), \mathrm{FRR}(\Delta, D); \gamma) \\ &\quad = \gamma \cdot \mathrm{FAR}(\Delta, D) + (1 - \gamma) \cdot \mathrm{FRR}(\Delta, D) \end{aligned} \tag{3.14}$$

  which basically varies the relative importance of $V1$ (FAR) with respect to $V2$ (FRR); in fact, setting $\gamma = 0.5$ yields the HTER cost (3.7);

- in text categorization, since the goal is to maximize precision and recall, one could use

$$\mathcal{C}(V1(\Delta, D), V2(\Delta, D); \gamma) \tag{3.15}$$

$$= \mathcal{C}(\mathrm{Precision}(\Delta, D), \mathrm{Recall}(\Delta, D); \gamma)$$

$$= -(\gamma \cdot \mathrm{Precision}(\Delta, D) + (1 - \gamma) \cdot \mathrm{Recall}(\Delta, D)) \tag{3.16}$$

  where $V1$ is the precision and $V2$ is the recall; notice the negative sign in 3.16 as precision and recall are penalty measures and instead of costs.

- in general, one could also be interested in trying to reach a particular relative value of $V1$ (or $V2$), such as *I am searching for a solution with as close as possible to 10% false acceptance rate*; in that case, one could use

$$\mathcal{C}(V1(\Delta, D), V2(\Delta, D); \gamma) = |\gamma - V1(\Delta, D)| \tag{3.17}$$

or

$$\mathcal{C}(V1(\Delta, D), V2(\Delta, D); \gamma) = |\gamma - V2(\Delta, D)| \,. \tag{3.18}$$

Having defined $\mathcal{C}(\cdot, \cdot; \gamma)$, the main procedure to generate the EPC is to vary γ inside a reasonable range (say, from 0 to 1), and for each value of γ, to estimate Δ that minimizes $\mathcal{C}(\cdot, \cdot; \gamma)$ on a development set, and then use the obtained Δ to compute some aggregate value (say, V), on the test set. Algorithm 3.1 details the procedure, while Figure 3.5 shows an artificial example of comparing the EPCs of two models. Looking at this figure, we can now state that for specific values of γ (say, between 0 and 0.5), the underlying obtained thresholds are such that model B is better than model A, while for other values, this is the converse. This assessment is unbiased in the sense that it takes into account the possible mismatch one can face while estimating the desired threshold.

Algorithm 3.1 Method to generate the Expected Performance Curve

```
Let $devel$ be the development set
Let $test$ be the test set
Let $V(\Delta, D)$ be the value of $V$ obtained on the data set $D$ for threshold $\Delta$
Let $\mathcal{C}(V1(\Delta, D), V2(\Delta, D); \gamma)$ be the value of a criterion $\mathcal{C}$ that depends on
$\gamma$, and is computed on the data set $D$
for values $\gamma \in [a, b]$ where $a$ and $b$ are reasonable bounds do
  $\Delta^\star = \arg\min_\Delta \mathcal{C}(V1(\Delta, devel), V2(\Delta, devel); \gamma)$
  compute V$(\Delta^\star, test)$
  plot V$(\Delta^\star, test)$ with respect to $\gamma$
end for
```

Let us suppose that Figure 3.5 was produced for a person authentication task, where V is the HTER, $V1$ is the FAR, and $V2$ is the FRR. Furthermore let us define the criterion as in (3.14). In that case, γ varies from 0 to 1, and when $\gamma = 0.5$ this corresponds to the setting where we tried to obtain a BEP (or equal error rate, as it is called in this domain), while when $\gamma < 0.5$ it corresponds to settings where we gave more importance to false rejection errors and when $\gamma > 0.5$ we gave more importance to false acceptance errors.

In order to illustrate EPCs in real applications, we have generated them for both a person authentication task and a text categorization task. The resulting curves can be seen in Figures 3.6 and 3.7. Note that the graph reporting $F1$ seems inverted with respect to the one reporting HTER, but this is because we are searching for low HTERs in person authentication but high $F1$ in text categorization. Note also that the EPC of Figure 3.6 corresponds to the ROC and DET of Figure 3.4. Finally, note that we kindly provide a C++ tool that generates such EPCs. An EPC generator is available at http://www.Torch.ch/extras/epc as a package of the Torch machine learning library.

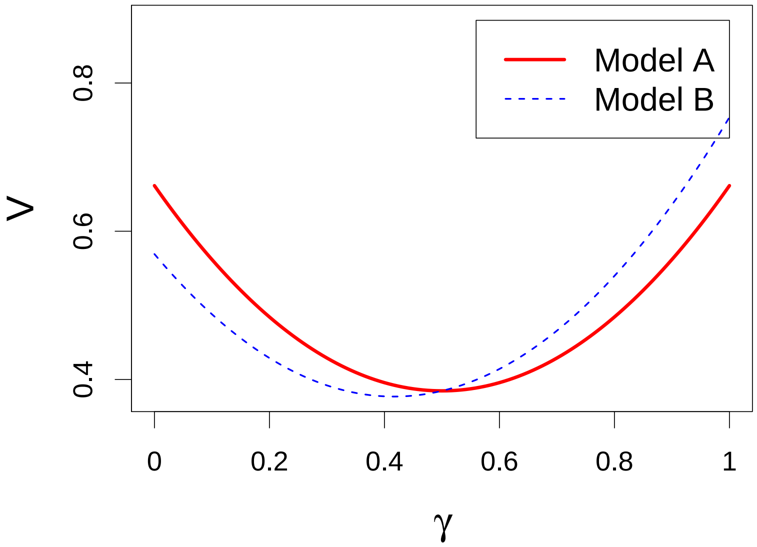


Figure 3.5. Example of two theoretical EPCs.

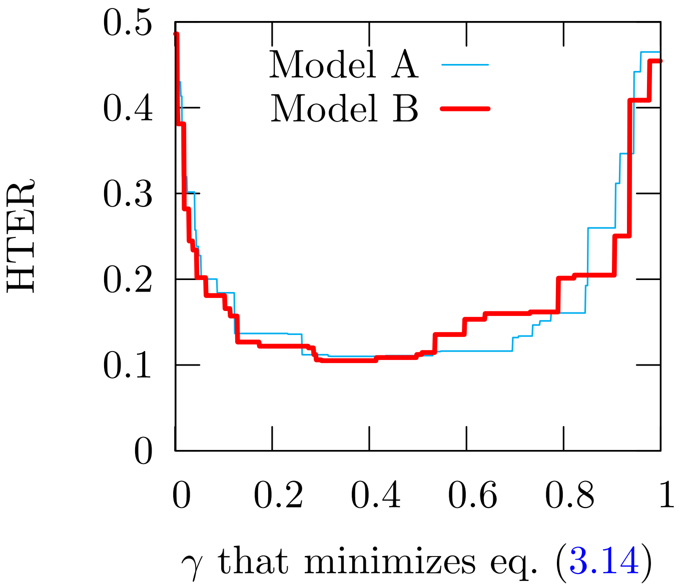


Figure 3.6. Expected Performance Curves for person authentication, where one wants to trade-off false acceptance rates with false rejection rates.

To compare the performance of two systems, we can use either numbers such as HTER with a decision threshold estimated "a priori" or curves such as EPC. Unfortunately, this might not be enough; as an error may be meaningless

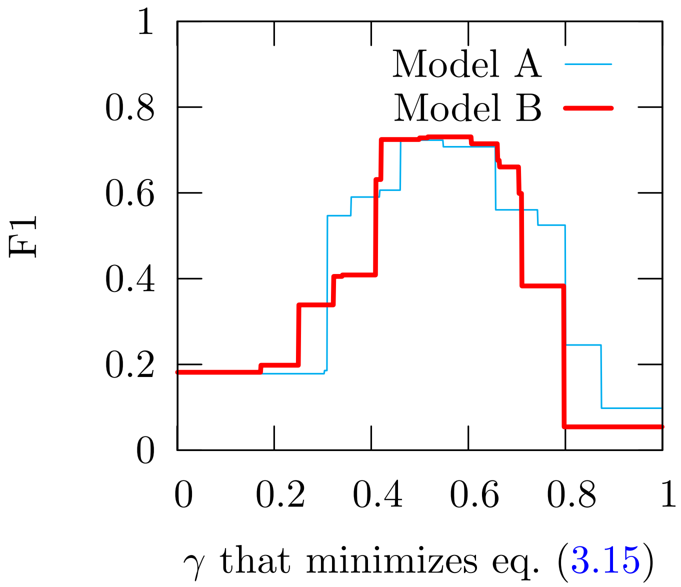


Figure 3.7. Expected Performance Curves for text categorization, where one wants to trade-off precision and recall and print the $F1$ measure.

if no confidence interval is given. In "biometric authentication", measures such as HTER are used instead of the classification error, thus, as will be shown in the next section, usual techniques to estimate the confidence interval cannot be used as is. We thus propose an adaptation of the z-test for speaker verification systems that can be applied to numbers such as HTER, DCF and also to EPCs.

3.3 Statistical Tests

Whenever one researcher wants to compare a novel model to an existing solution, using either one value such as HTER or using a curve such as EPC, a quick review of the current literature in person authentication shows that either no statistical test is used to assess the difference between models, or, worse, statistical tests are used incorrectly, which often ends up in over-optimistic results, tending to show, for instance, that the new model is statistically significantly better than the state-of-the-art while it might not be the case in fact.

In this section, we present a proper method to compute a simple statistical test, known as the *test of two proportions*, or *z-test*, adapted to the problem of aggregate measures such as HTER and DCF.

The Z-Test on Proportions

Several statistical tests are available in the literature. For standard classification tasks, a simple yet often used test is known as the *z-test*, or *test between* *two proportions.* The rationale of this test is the following: given a set of n examples, each drawn independently and identically distributed (i.i.d.) from an unknown distribution, a given system is going to take a decision for each example, and this decision will be correct or not. Let us now look at the distribution of the number of errors that will be made by the classification system. Since each decision is independent from the others and is binary, it is reasonable to assume that the random variable $\mathbf{X}$ representing the number of errors should follow a *Binomial* distribution $\mathcal{B}(n, p)$ where n is the number of examples and p is the percentage of errors. In this section we use the following notation: bold letters such as **FA** represent random variables, while normal letters such as FA represent a particular value of the underlying random variable.

Moreover, it is known that a Binomial $\mathcal{B}(n, p)$ can be approximated by a Normal distribution $\mathcal{N}(\mu, \sigma^2)$ with

$$\mu = np \quad \text{and} \quad \sigma^2 = np(1-p)$$

when n is large enough. A rule of thumb often used is to have $np(1-p)$ larger than 10.

Finally, if $\mathbf{X} \sim \mathcal{N}(np, np(1-p))$, then the distribution of the proportion of errors $\mathbf{Y} = \frac{\mathbf{X}}{n} \sim \mathcal{N}\left(p, \frac{p(1-p)}{n}\right)$.

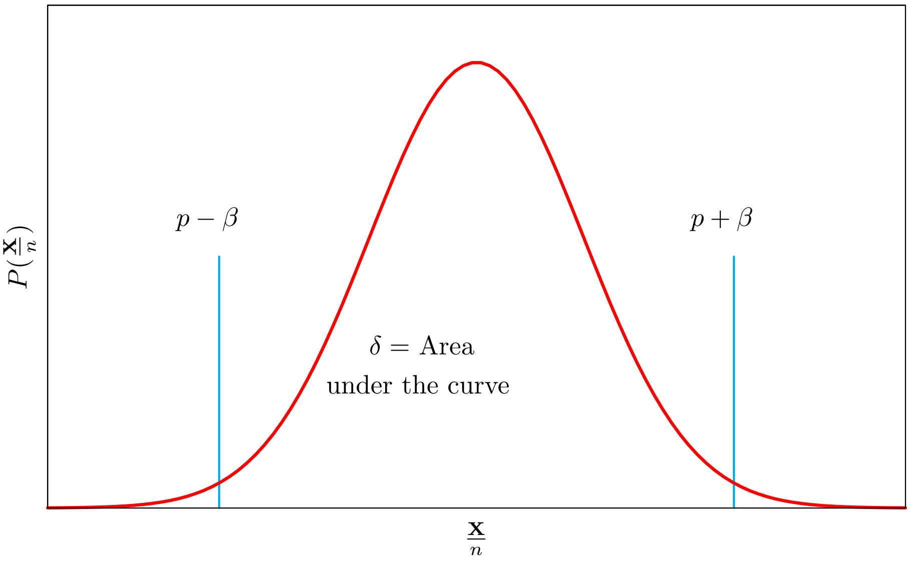


Figure 3.8. Confidence intervals are computed using the area under the Normal curve.

Confidence Intervals

In order to compute a confidence interval around p, we can search for bounds $\{p - \beta, p + \beta\}$ such that

$$P\left(p - \beta < \mathbf{Y} < p + \beta\right) = \delta \tag{3.19}$$

where δ represents our confidence. This is called a *two-sided* test since we are searching for two bounds around p. Fortunately, finding β in (3.19) for a given δ can be done efficiently for the Normal distribution. Figure 3.8 illustrates graphically the problem.

Difference Between Proportions

Alternatively, if one wants to verify whether a given proportion of errors p_A is statistically significantly different from another proportion p_B, a similar test can be performed. In the case where we already know that p_A cannot be lower than p_B, a *one-sided* test is used, otherwise we use a *two-sided* test. Noting respectively $\mathbf{Y}_A$ and $\mathbf{Y}_B$ the random variables representing the distribution of p_A and p_B, the *one-sided* test is based on

$$P(\mathbf{Y}_A - \mathbf{Y}_B < p_A - p_B) = \delta \tag{3.20}$$

while the *two-sided* test is based on

$$P(|\mathbf{Y}_A - \mathbf{Y}_B| < |p_A - p_B|) = \delta \tag{3.21}$$

which can be solved using the fact that the difference between two independent Normal distributions is a Normal distribution where the mean is the difference between the two Normal means and the variance is the sum of the two Normal variances, hence, if $\mathbf{Y}_A$ is not statistically different from $\mathbf{Y}_B$, then

$$\mathbf{Y}_A - \mathbf{Y}_B \sim \mathcal{N}\left(p_A - p_B, \frac{p_A(1 - p_A) + p_B(1 - p_B)}{n}\right) \tag{3.22}$$

and if δ is higher than a predefined value (such as 95%), then one can state that p_A is significantly different from p_B. Note that a better estimate of the variance of (3.22) can be obtained when assuming $p_A = p_B$ (which should be the case if they are not significantly different). In that case, equation (3.22) becomes

$$\mathbf{Y}_A - \mathbf{Y}_B \sim \mathcal{N}\left(0, \frac{2p(1 - p)}{n}\right) \tag{3.23}$$

with

$$p = \frac{p_A + p_B}{2} \ .$$

Note however that using this test to verify whether two models give statistically significantly different results on the same test database makes a wrong hypothesis, since $\mathbf{Y}_A$ and $\mathbf{Y}_B$ are not really independent as they correspond to decisions taken on *the same test set.*

Dependent Case

One possible solution proposed in (Snedecor and Cochran, 1989) is to only take into account the examples for which the two models disagree. Let p_{AB} be the proportion of examples correctly classified by model A and incorrectly classified by model B, and similarly p_{BA} be the proportion of examples correctly classified by model B and incorrectly classified by model A. In that case, the distribution $\mathbf{Y}_{|A-B|}$ of the difference between the proportions of errors committed by each model is still Normally distributed and, assuming the two models are not different from each other, should follow

$$\mathbf{Y}_{|A-B|} \sim \mathcal{N}\left(0, \frac{p_{AB} + p_{BA}}{n}\right) \tag{3.24}$$

with the corresponding two-sided test

$$P(\mathbf{Y}_{|A-B|} < |p_{AB} - p_{BA}|) = \delta \; . \tag{3.25}$$

This test is in fact very similar to the well-known McNemar test, based on a χ^2 distribution.

In the literature, most people adopt equation (3.23) and some adopt equation (3.24); remember that in order to use equation (3.24), one needs to have access to all the scores of both models, and not just the numbers of errors. When possible, we will look at both solutions here, for the case of person authentication.

$\mathrm{Z}_{\mathrm{HTER}}$-*Test: a Statistical Test for HTERs*

HTERs are not proportions, but they are an average of two well-defined proportions (FAR and FRR). In the following, we propose to extend the test between two proportions for the case of HTERs. We assume the distributions of FAR and FRR independent. This may look false since they are both linked by the same model and threshold, but in fact, *given a model and associated threshold* these two quantities are indeed most probably independent since they are computed on separate data (the client accesses and the impostor accesses), assuming the model was estimated on a separate training set, as it should be.

Confidence Intervals

Let the random variable **FP** represent the number of false positive. We can model it by a Binomial, and hence by a Normal, as follows:

G. W. Snedecor and W. G. Cochran. *Statistical Methods.* Iowa State University Press, 1989.

$$\begin{aligned}
\mathbf{FP} &\sim \mathcal{B}\left(\mathrm{NN}, \frac{\mathrm{FP}}{\mathrm{NN}}\right) \\
&\sim \mathcal{N}\left(\mathrm{NN}\cdot\frac{\mathrm{FP}}{\mathrm{NN}}, \mathrm{NN}\cdot\frac{\mathrm{FP}}{\mathrm{NN}}\cdot\left(1-\frac{\mathrm{FP}}{\mathrm{NN}}\right)\right) \\
&\sim \mathcal{N}\left(\mathrm{FP}, \mathrm{FP}\cdot(1-\mathrm{FAR})\right) \ . \qquad (3.26)
\end{aligned}$$

The random variable **FN** representing the number of false negative can be modeled accordingly:

$$\begin{aligned}
\mathbf{FN} &\sim \mathcal{B}\left(\mathrm{NP}, \frac{\mathrm{FN}}{\mathrm{NP}}\right) \\
&\sim \mathcal{N}\left(\mathrm{NP}\cdot\frac{\mathrm{FN}}{\mathrm{NP}}, \mathrm{NP}\cdot\frac{\mathrm{FN}}{\mathrm{NP}}\cdot\left(1-\frac{\mathrm{FN}}{\mathrm{NP}}\right)\right) \\
&\sim \mathcal{N}\left(\mathrm{FN}, \mathrm{FN}\cdot(1-\mathrm{FRR})\right) \ . \qquad (3.27)
\end{aligned}$$

We can now write the distribution of the random variable **FAR** representing the ratio of false acceptances:

$$\begin{aligned}
\mathbf{FAR} &\sim \mathcal{N}\left(\frac{\mathrm{FP}}{\mathrm{NN}}, \frac{\mathrm{FP}\,(1-\mathrm{FAR})}{\mathrm{NN}\cdot\mathrm{NN}}\right) \\
&\sim \mathcal{N}\left(\mathrm{FAR}, \frac{\mathrm{FAR}\,(1-\mathrm{FAR})}{\mathrm{NN}}\right) \qquad (3.28)
\end{aligned}$$

and similarly for the random variable **FRR**:

$$\begin{aligned}
\mathbf{FRR} &\sim \mathcal{N}\left(\frac{\mathrm{FN}}{\mathrm{NP}}, \frac{\mathrm{FN}\,(1-\mathrm{FRR})}{\mathrm{NP}\cdot\mathrm{NP}}\right) \\
&\sim \mathcal{N}\left(\mathrm{FRR}, \frac{\mathrm{FRR}\,(1-\mathrm{FRR})}{\mathrm{NP}}\right) \qquad (3.29)
\end{aligned}$$

Given the distribution of **FAR** and **FRR**, we can estimate the distribution of the random variable **HTER** as follows:

$$\begin{aligned}
\mathbf{FAR{+}FRR} &\sim \mathcal{N}\left(\mathrm{FAR{+}FRR}, \frac{\mathrm{FAR}\,(1-\mathrm{FAR})}{\mathrm{NN}} + \frac{\mathrm{FRR}\,(1-\mathrm{FRR})}{\mathrm{NP}}\right) \\
\frac{\mathbf{FAR{+}FRR}}{2} &\sim \mathcal{N}\left(\frac{\mathrm{FAR{+}FRR}}{2}, \frac{\mathrm{FAR}\,(1-\mathrm{FAR})}{4\cdot\mathrm{NN}} + \frac{\mathrm{FRR}\,(1-\mathrm{FRR})}{4\cdot\mathrm{NP}}\right) \\
\mathbf{HTER} &\sim \mathcal{N}\left(\mathrm{HTER}, \frac{\mathrm{FAR}(1-\mathrm{FAR})}{4\cdot\mathrm{NN}} + \frac{\mathrm{FRR}(1-\mathrm{FRR})}{4\cdot\mathrm{NP}}\right) \qquad (3.30)
\end{aligned}$$

Using this last definition, we can now compute easily confidence intervals around HTERs using the methodology summarized in Figure 3.9 for classical confidence values used in the scientific literature.

Moreover, the test can be easily extended to variations of HTER, such as the DCF in (3.3). For instance, in the case of the well-known NIST evaluations performed yearly to compare speaker verification systems, and which use the DCF measure described by equation (3.3) with Cost(FR) = 10, P(client) = 0.01, Cost(FA) = 1 and P(impostor) = 0.99, the underlying Normal becomes:

$$\mathbf{DCF} \sim \mathcal{N}\left(\mathrm{DCF}, \frac{\mathrm{FAR}\,(1-\mathrm{FAR})}{0.99^{-2}\cdot\mathrm{NN}} + \frac{\mathrm{FRR}\,(1-\mathrm{FRR})}{100\cdot\mathrm{NP}}\right)\ . \tag{3.31}$$

Difference Between HTERs

The distribution of the difference between two HTERs assuming *independence* between the two underlying distributions is

$$\mathbf{HTER}_A - \mathbf{HTER}_B \sim \mathcal{N}\left(0, \sigma^2_{\mathbf{INDEP}}\right) \tag{3.32}$$

with

$$\sigma^2_{\mathbf{INDEP}} = \begin{cases} \dfrac{\mathrm{FAR}_A\,(1-\mathrm{FAR}_A) + \mathrm{FAR}_B\,(1-\mathrm{FAR}_B)}{4\cdot\mathrm{NN}} \\ +\dfrac{\mathrm{FRR}_A\,(1-\mathrm{FRR}_A) + \mathrm{FRR}_B\,(1-\mathrm{FRR}_B)}{4\cdot\mathrm{NP}} \end{cases}$$

while the distribution of the difference between two HTERs assuming *dependence* between the two underlying distributions becomes

$$\mathbf{HTER}_A - \mathbf{HTER}_B \sim \mathcal{N}\left(0, \sigma^2_{\mathbf{DEP}}\right) \tag{3.33}$$

with

$$\sigma^2_{\mathbf{DEP}} = \frac{\mathrm{FAR}_{AB} + \mathrm{FAR}_{BA}}{4\cdot\mathrm{NN}} + \frac{\mathrm{FRR}_{AB} + \mathrm{FRR}_{BA}}{4\cdot\mathrm{NP}}$$

where $\mathrm{FAR}_{AB} = \frac{\mathrm{NN}_{AB}}{\mathrm{NN}}$ and NN_{AB} is the number of impostor accesses correctly rejected by model A and incorrectly accepted by model B, with similar definitions for FAR_{BA}, FRR_{AB}, and FRR_{BA}.

Hence, in summary, and using the standard confidence values used in the scientific literature, we obtain the simple methodology described in Figure 3.9 in order to compute statistical tests for person authentication tasks. Figure 3.9 represents a two-sided test and we thus use $Z_\alpha/2$ instead of Z_α. While this summary concerns HTERs, it should now be obvious to extend it to the general DCF function.

Other Statistical Tests

While several researchers have pointed out the use of the *z-test* to compute statistical tests around values such as FAR or FRR, see for instance (Wayman,

[0] J.L. Wayman. Confidence interval and test size estimation for biometric data. In *Proceedings of the IEEE AutoID Conference*, 1999.

The confidence interval (CI) around an HTER is HTER $\pm \sigma \cdot Z_{\alpha/2}$ with

$$\sigma = \sqrt{\frac{\mathrm{FAR}(1-\mathrm{FAR})}{4 \cdot \mathrm{NN}} + \frac{\mathrm{FRR}(1-\mathrm{FRR})}{4 \cdot \mathrm{NP}}}$$

$$Z_{\alpha/2} = \begin{cases} 1.645 & \text{for a } 90\% \text{ CI} \\ 1.960 & \text{for a } 95\% \text{ CI} \\ 2.576 & \text{for a } 99\% \text{ CI} \end{cases}$$

and similarly, HTER_A and HTER_B are statistically significantly different if $z > Z_{\alpha/2}$ with

$$z = \frac{|\mathrm{HTER}_A - \mathrm{HTER}_B|}{\sqrt{\frac{\mathrm{FAR}_A(1-\mathrm{FAR}_A) + \mathrm{FAR}_B(1-\mathrm{FAR}_B)}{4 \cdot \mathrm{NN}} + \frac{\mathrm{FRR}_A(1-\mathrm{FRR}_A) + \mathrm{FRR}_B(1-\mathrm{FRR}_B)}{4 \cdot \mathrm{NP}}}}$$

in the independent case, and

$$z = \frac{|\mathrm{FAR}_{AB} - \mathrm{FAR}_{BA} + \mathrm{FRR}_{AB} - \mathrm{FRR}_{BA}|}{\sqrt{\frac{\mathrm{FAR}_{AB} + \mathrm{FAR}_{BA}}{4 \cdot \mathrm{NN}} + \frac{\mathrm{FRR}_{AB} + \mathrm{FRR}_{BA}}{4 \cdot \mathrm{NP}}}}$$

in the dependent case.

Figure 3.9. Methodology for statistical tests around HTERs for a two-sided test.

1999), we are not aware, to the best of our knowledge, of any similar attempt for aggregate measures such as HTERs (or EER, or DCF). However, most people publishing results in verification use HTERs or DCF to assess the quality of their methods.

One simple solution could be to consider the classification error instead of the HTER and compute statistical tests around it. Since the classification error is a well-defined proportion, we can apply the *z-test* as well; Let **CLASS** be defined as the following random variable:

$$\mathbf{CLASS} = \frac{\mathbf{FP}+\mathbf{FN}}{\mathrm{NP}+\mathrm{NN}}$$

then, the corresponding underlying Normal becomes:

$$\mathbf{CLASS} \sim \mathcal{N}\left(\frac{\text{FP+FN}}{\text{NP+NN}}, \frac{\text{FP+FN}}{(\text{NP+NN})^2}\left(1-\frac{\text{FP+FN}}{\text{NP+NN}}\right)\right) \tag{3.34}$$

but remember that while this test is correct to assess models according to their respective classification error, it does not say anything on the confidence one has over the corresponding HTER, which is the measure of interest in person authentication. In fact, we will show in the next section that, under reasonable assumptions, the variance of **CLASS** in equation (3.34) is always smaller than the variance of **HTER** in equation (3.30), hence confidence tests using (3.34) will always result in over-confident statistical significance (or smaller confidence intervals). This will be explored further in the following section.

Another possible solution is to consider the HTER itself as a proportion (which it is not directly) and compute the statistical test on it. Let **NAIVE** be the random variable of this value; the underlying Normal becomes:

$$\mathbf{NAIVE} \sim \mathcal{N}\left(\text{HTER}, \frac{\text{HTER}(1-\text{HTER})}{\text{NP+NN}}\right) \tag{3.35}$$

Again, we will show in next section that under reasonable assumptions, the variance of **NAIVE** in equation (3.35) is always smaller than the variance of **HTER** in equation (3.30), hence confidence tests using (3.34) should always result in over-confident statistical significance (or smaller confidence intervals).

Yet another solution that has been proposed by some researchers, see for instance (Koolwaaij, 2000), is to compute a statistical test for FAR and FRR separately and then combine the results. The well-known NIST evaluation campaigns have also apparently recently investigated the use of the McNemar test to assess speaker verification methods, but have considered separately FARs and FRRs (Martin, 2004). For instance, in order to compute a confidence interval for HTER, one would average both upper bounds and both lower bounds found separately by the FAR and FRR tests. On top of the fact that there is no theoretical ground to justify such an approach, there is an evident problem with all approaches that consider separately FARs and FRRs. Two models could yield very similar HTERs but for some reason (linked to the choice of the threshold, which should be selected on a separate data set) one could be slightly biased toward FRRs and the other one slightly biased toward FARs. In such a case, these tests would consider them statistically significantly different while they would not be when considering globally their respective HTER instead. For this reason, we will not consider this solution further here.

J. Koolwaaij. *Automatic Speaker Verification in Telephony: a probabilitic approach.* PrintPartners Ipskamp B.V., Enschede, 2000.

A Martin. Personal communication. http://www.nist.gov/speech/staff/martinal.htm, 2004.

Analysis

We would like to compare in this section the use of the $\mathbf{Z_{HTER}}$-test with respect to the two other Class and Naive tests presented in the previous section. We will first show that under some reasonable conditions, increasing the ratio between NN and NP will increase the difference between the variance of the Normal of the $\mathbf{Z_{HTER}}$-test and the variance of the Normal of the other tests. Afterwards, we present two real case studies where the use of the $\mathbf{Z_{HTER}}$-test would have yielded a different conclusion with regard to the confidence intervals and the difference between the compared models.

Theoretical Analysis

Let us first look in which conditions $\sigma^2(3.30)$, the variance of **HTER** as written in equation (3.30) is higher than $\sigma^2(3.35)$, the variance of **NAIVE** as written in equation (3.35):

$$\sigma^2(3.30) > \sigma^2(3.35) \tag{3.36}$$

implies that

$$\frac{\text{FAR}\,(1-\text{FAR})}{4\,\text{NN}} + \frac{\text{FRR}\,(1-\text{FRR})}{4\,\text{NP}} > \frac{\text{HTER}(1-\text{HTER})}{\text{NP+NN}} \tag{3.37}$$

and assuming FAR is similar than FRR (again, when the threshold is chosen such that we have equal error rate (EER) on a separate validation set, as it is often done, this is reasonable), which can be simplified and yields

$$1 > \frac{1}{\text{NP+NN}} \tag{3.38}$$

which means that inequation (3.36) is always true under the assumption that FAR = FRR.

Let us now look in which conditions $\sigma^2(3.30)$ is higher than $\sigma^2(3.34)$, the variance of **CLASS**, representing the classification error:

$$\sigma^2(3.30) > \sigma^2(3.34) \tag{3.39}$$

implies that

$$\frac{\text{FAR}(1-\text{FAR})}{4\cdot\text{NN}} + \frac{\text{FRR}(1-\text{FRR})}{4\cdot\text{NP}} > \frac{\text{FP+FN}}{(\text{NP+NN})^2}\cdot(1-\frac{\text{FP+FN}}{\text{NP+NN}})$$

and assuming FAR is similar to FRR, it can be simplified into

$$1 > \frac{1}{\text{NP+NN}} \tag{3.40}$$

which is true as long as FAR = FRR. Note that (3.38) is equal to 3.40, because $\sigma^2(3.35) = \sigma^2(3.34)$ when FAR = FRR.

In order to verify these relations graphically, we have fixed some variables to reasonable values (FAR = 0.1, FRR = 0.2, NP = 100) and have varied NN, the number of impostor accesses. Figure 3.10 shows the relation between the standard deviation of the underlying Normal distributions and the ratio between NN and NP.

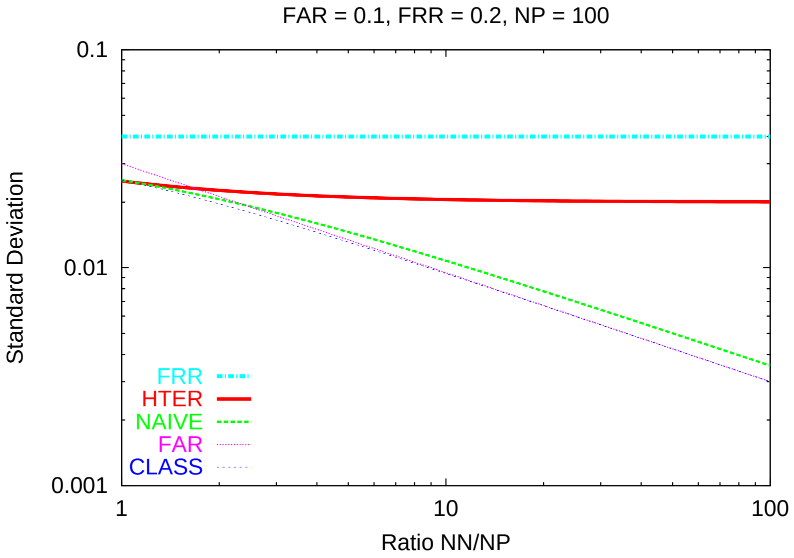


Figure 3.10. Standard deviation of the Normal distributions underlying the three different choices of distributions for a statistical test on HTERs. Also shown: standard deviations of both the **FAR** and **FRR** distributions. All curves are in log-log scale. The order in the legend corresponds to the order of the curves at the right of the figure.

As expected, the higher the ratio $\frac{\mathrm{NN}}{\mathrm{NP}}$, the bigger the difference between the standard deviation of the Normal distributions related to the three statistical tests. Moreover, we see that the standard deviation of the $\mathbf{Z_{HTER}}$-test distribution stays close to the one of the **FRR** distribution, which is mostly influenced by NP, the number of client accesses, and does not decrease with the increase of NN, contrary to the two other solutions. Since the size of the confidence interval is directly related to the standard deviation, this figure essentially shows that the confidence interval computed using the $\mathbf{Z_{HTER}}$-test will always be larger than that of the two other techniques. Hence two verification methods yielding two different HTERs could easily be considered statistically significantly different using one of the Class or Naive methods, while they would not be considered statistically significantly different using the $\mathbf{Z_{HTER}}$-test technique. In fact, the figure shows that the confidence interval is directly influenced by the minimum of NP and NN and not their sum.

In the next two subsections, we present two real case studies where the use of the $\mathbf{Z_{HTER}}$ statistical test would have yielded a different conclusion.

Empirical Analysis on XM2VTS

In the first case, the well-known text-independent audio-visual verification database XM2VTS (Lüttin, 1998) was used. In this database, the test set consists of up to 112000 impostor accesses and only 400 client accesses, for a total of 112400 accesses. In a recent competition (Messer et al., 2003), several models were compared on a face verification task and we will look here at the results of the best model, hereafter called *model A*, and the third best model, hereafter called *model B*, apparently significantly worse. Table 3.2 shows the difference of performance in terms of HTER between models A and B. Having up to 112400 examples, one could indeed expect the difference between the two models to be statistically significant.

While this is not the topic of this section (since it should apply to any data/model), people interested in knowing more about the problem tackled in this case study are referred to (Messer et al., 2003); we used results of the models of IDIAP and UniS-NC on the automatic registration task, using Lausanne Protocol I. Furthermore, note that the results of UniS-NC are slightly different from those published by Messer et al. (2003), but correspond to the list of scores provided by one of the authors of the method.

Method	FAR (%)	FRR (%)	HTER (%)
Model A	1.15	2.50	1.82
Model B	1.95	2.75	2.35

Table 3.2. HTER Performance comparison on the test set between models A and B when the threshold was selected according to the Equal Error Rate criterion (EER) on a separate validation set.

Table 3.3 shows the size of the confidence intervals computed around the result (using HTER or the classification error) obtained by model A for the three methods for three different values of δ (90%, 95% and 99%). As we can

J Lüttin. Evaluation protocol for the the XM2FDB database (lausanne protocol). IDIAP-COM 05, IDIAP, 1998.

K. Messer, J. Kittler, M. Sadeghi, S. Marcel, C. Marcel, S. Bengio, F. Cardinaux, C. Sanderson, J. Czyz, L. Vandendorpe, S. Srisuk, M. Petrou, W. Kurutach, A. Kadyrov, R. Paredes, B. Kepenekci, F. B. Tek, G. B. Akar, F. Deravi, and N. Mavity. Face verification competition on the XM2VTS database. In *4th International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA*. Springer-Verlag, 2003.

δ	HTER eq (3.30)	NAIVE eq (3.35)	CLASS eq (3.34)
90%	1.285%	0.131%	0.105%
95%	1.531%	0.156%	0.125%
99%	2.013%	0.206%	0.164%

Table 3.3. Confidence intervals around results of model A, computed using three different hypotheses (and their respective equation).

see, for all values of δ, the size of the interval is about one order of magnitude larger for the $\mathbf{Z_{HTER}}$-test based method than for the two other methods.

	HTER DEP, eq (3.33)	HTER INDEP, eq (3.32)	NAIVE eq (3.35)	CLASS eq (3.34)
δ	69.2%	64.7%	100.0%	100.0%
σ	0.0052	0.0057	0.0006	0.0005

Table 3.4. Confidence value δ on the fact that model A is statistically significantly different from model B, according to their respective performance (HTER or classification error), and computed using four different hypotheses (and their respective equation). For each method, we also give σ, the standard deviation of the corresponding statistical test.

Table 3.4 verifies whether the HTER obtained by model A gives statistically significantly different results than the one obtained by model B, using the *two-sided* test of equation (3.21) for the independent cases and (3.25) for the dependent case. According to both proposed $\mathbf{Z_{HTER}}$-test based methods (independent and dependent cases), both models are equivalent (the confidence on their difference, δ is much less than, say, 90%), while according to both other methods, the models would be different (with 100% confidence!). Remember that there was only 400 client accesses during the test, hence it is reasonable that only one error on these accesses makes a visible difference in HTER while it cannot seriously be considered statistically significant. This is well captured by our technique, but not by the other ones. Moreover, in this case, the dependence/independence assumption did not have any impact on the final decision.

Empirical Analysis on NIST'2000

In the second case, the well-known text-independent speaker verification benchmark database NIST'2000 was used. Here, the test set consists of 57748 impostor accesses and 5825 client accesses, for a total of 63573 accesses. We compared the performance of two models hereafter called *models C* and *D*. Note that, while on XM2VTS the ratio between the number of impostor and client accesses was very high (280 times more), for the NIST database, the ratio is more reasonable, but still high (around 10). Once again, while this is not the topic of this section, people interested in knowing more about the problem tackled in this case study are referred to (Mariéthoz and Bengio, 2003).

Method	FAR (%)	FRR (%)	HTER (%)
Model C	13.1	9.6	11.4
Model D	15.8	7.8	11.8

Table 3.5. HTER Performance comparison on the test set between models C and D when the threshold was selected according to the Equal Error Rate criterion (EER) on a separate validation set.

δ	HTER	NAIVE	CLASS
	eq (3.30)	eq (3.35)	eq (3.34)
90%	0.676%	0.414%	0.436%
95%	0.805%	0.493%	0.519%
99%	1.058%	0.648%	0.682%

Table 3.6. Confidence intervals around results of model C, computed using three different hypotheses (and their respective equation).

	HTER	HTER	NAIVE	CLASS
	DEP, eq (3.33)	INDEP, eq (3.32)	eq (3.35)	eq (3.34)
δ	98.8%	89.1%	98.9%	100.0%
σ^2	0.0016	0.0028	0.0018	0.0019

Table 3.7. Confidence value δ on the fact that model C is statistically significantly different from model D, according to their respective performance (HTER or classification error), and computed using four different hypotheses (and their respective equation). For each method, we also give σ, the standard deviation of the corresponding statistical test.

We now present the same kinds of results as for the XM2VTS case. Table 3.5

J. Mariéthoz and S. Bengio. An alternative to silence removal for text-independent speaker verification. IDIAP-RR 51, IDIAP, Martigny, Switzerland, 2003.

shows the difference of performance in terms of HTER between models C and D; Table 3.6 shows the size of the confidence intervals computed around the result obtained by model C; as we can see, given a ratio of impostor and client accesses around 10 instead of 280, the difference between all the confidence intervals is less drastic but still exists; Table 3.7 verifies whether the HTER obtained by model C gives statistically significantly different results than the one obtained by model D. For each test, we show both the confidence value δ and the standard deviation σ of the corresponding statistical test.

As it can be seen, in the DEP case, σ is very small, even smaller than the NAIVE and CLASS solutions, hence obtaining a very high confidence that the two models are different. In order to explain this unexpected result, note than none of the tests takes into account the possible dependence existing between the compared *models*. Indeed, if the two models are based on the same technique (which is often the case; for instance, in speaker verification, most systems are often based on Gaussian Mixture Models, but trained with slightly different assumptions), then both systems will have a natural tendency to answer very correlated scores on the same example. In the case of the two models trained on the XM2VTS database, they were very different (one was based on a Gaussian Mixture Model, while the other one was based on Linear Discriminant Analysis and Normalized Correlation); while for the models trained on the NIST database, both were in fact variations of Gaussian Mixture Models, hence are probably very correlated. Unfortunately, there exist no test that take this dependency into account. Hence, for instance, the variance $\frac{p_{AB}+p_{BA}}{n}$ of equation (3.24) will be quickly very small simply because the models are correlated (and not just because the examples are the same). Using this equation will thus result in an underestimate of the true variance when models are very correlated, as empirically shown in Table 3.7.

On the other hand, the INDEP case does not take into account the dependency between the data, but somehow it is reasonable to expect that the effect of this error may be balanced by the fact that it does not take into account the dependency between the models neither. The correct solution probably lies somewhere between these two solutions, hence, one should probably favor the most difficult test so as to only assess statistical differences when both tests agree on this fact (hence, here, with only 89.1% confidence).

As we have seen, two tests can be used: the independent case and the dependent case. In the following, we will use the independent case because it is very simple to compute, only FAR and FRR are needed, and we make sure that its outcome is not optimistically biased. As we have defined several new concepts such as EPC and z-test for speaker verification systems, we now present a summary of the way we tend to present results in the rest of this document.

3.4 Methodology and Presentation of Results

In this thesis, we present results using numbers and curves. We chose to present HTER as number measure by setting the threshold with a criterion that minimizes the EER on some separate validation set. We also add a confidence interval using the algorithm described in Figure 3.9 using the independent case. Table 3.8 shows examples of results:

Table 3.8. Sample of Results.

	Model A	Model B
HTER [%]	4.9	4.58
95% Confidence	±0.33	±0.33

DET curves will be used only for analysis purpose, as we have seen in Section 3.2, that EPC are more appropriate to presents final results. Different kinds of curves can be used. We propose to use a linear combination of FAR and FRR in abscissa representing the variation of γ. In ordinate, we would like to present a combination of FAR and FRR; two choices are possible, the DCF or HTER. The DCF has the advantage to plot what we are optimizing: a linear combination of FAR and FRR. The main drawback of this measure is that each point of the same curve cannot be compared. We can use HTER instead and in this case all points are comparable between curves which can be useful to choose a good operation point for a specific application. Figure 3.11 shows a typical EPC curve as presented later in order to compare systems. The best curve has its own confidence interval, but we need to have a confidence of how two models are different. This is thus presented in the second part of the figure. Each time that the blue line is greater that 95%, we can consider the two models as different with 95% confidence.

3.5 Conclusion

In this chapter we have presented the common measures used in speaker verification. We pointed out some problems of the use of theses measures found in the literature. First, we reminded that measures such that EER, ROC and DET curves are "a posteriori" measures and should thus not be used to compare systems. As no previously defined curve, to the best of our knowledge, was taking into account the decision threshold estimation problem, we have proposed new kinds of curves called EPCs. This work has been published in:

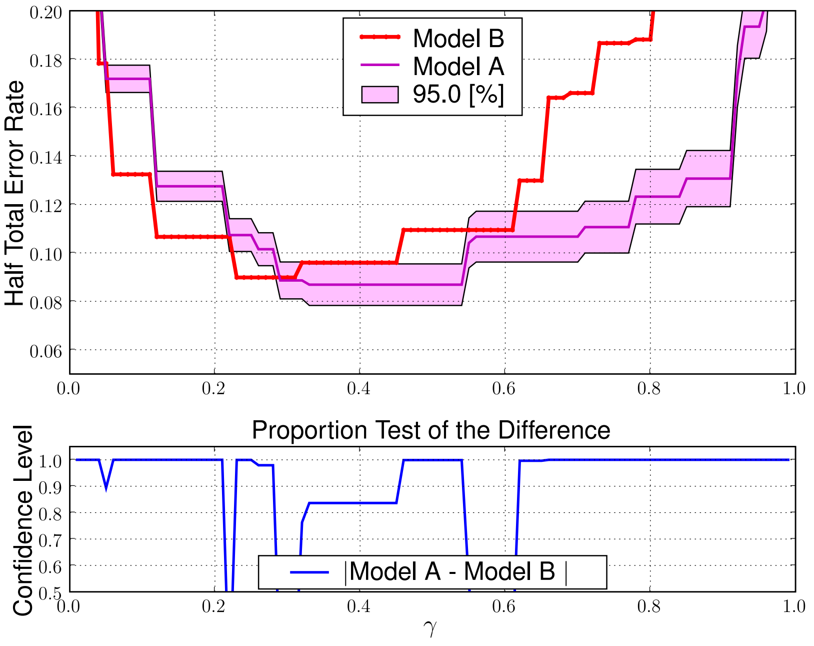


Figure 3.11. EPC curves using HTER with Confidence Intervals.

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

and more specifically for speaker verification in:

CONTRIB S. Bengio and J. Mariéthoz. The expected performance curve: a new assessment measure for person authentication. In *Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop*, 2004

Moreover as no statistical test, such as the Z-test, was applicable to the speaker verification problem, we proposed an adapted Z-test to give a confidence interval for speaker verification systems such as HTER and DCF. This work has been published in:

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

Finally, we have presented a typical example of results as presented later in this thesis.

Once we have defined the measures, we need data to estimate the quality of our new models. In the next chapter, we have chosen three well-known datasets and we have defined a new methodology to use them with discriminant models. Moreover, we present a new database called Banca with its own protocols and show that it is not easy to design a protocol to obtain unbiased results.

4 Experimental Methodology

In this chapter, we describe the methodology used to perform text-independent speaker verification experiments in this thesis. Three databases, Banca, Polyvar and NIST, are used in the following to compare systems. Two baseline models are considered: a GMM based system described in Section 2.2 and summarized in Figure 2.1 and an SVM with GLDS kernel based system described in Section 2.5 and summarized in Figure 2.3.

The outline of this chapter goes as follows. In Section 4.1, we describe the general methodology to use the databases. In Section 4.2, the databases are described and the baseline results are given for each of them.

4.1 Methodology

For both GMM and SVM based systems, the feature extraction, described in Section 2.3, is computed using the same procedure, as follows. The original waveforms are sampled every 10ms with a window size of 20ms. For all databases, each sentence is parameterized using 24 triangular band-pass filters with a DCT transformation, computed using (2.20), of order 16, complemented by their first derivative (delta), the log-energy and the delta-log-energy, for a total of 34 coefficients. A simple silence detector based on an unsupervised bi-Gaussian model is used to remove all silence frames. A bi-Gaussian model is learned using the ML criterion except for the NIST database. Since this database is noisy, the bi-Gaussian model is first learned on a random recording with land line microphone and adapted for each new sentence using the MAP algorithm with a MAP adaptation factor of $\lambda = 0.5$ in (2.6). All frames were normalized in order to have zero mean and unit variance. The NIST database being a telephone based database, the signal is thus band-pass filtered between 300 and 3400 Hz.

While the log energy is important in order to remove the silence frames, it is known not to be appropriate for the task of discrimination between clients and impostors. This feature was thus removed after removing the silences, but its first derivative was kept. Hence, the models are trained with 33 (34-1) features.

In order to select the various hyper-parameters (such as the number of Gaussians, the MAP adaptation factor, etc.), two different client populations are used: one for the development and one for the test set. We use the development set as follows; for each value of the hyper-parameter to tune, we train the client models using the training data available for each client. We then select the value of the hyper-parameter that optimizes the EER on the clients and impostors trials of the development set. Finally, we train the models on the test set using these hyper-parameters and measure the performance of the system.

All databases contain some accesses to enroll the world model. These accesses are also used as negative examples for discriminant models. The T-normalization models are the client models of the development set. When T-normalization is performed on the development set a leave-one-out cross-validation procedure (Devroye and Gyorfi, 1997) is applied in order not to bias the results: the model corresponding to the claimed identity is removed from the T-normalization model list.

4.2 Databases

In order to compare the systems presented here, three databases are used: Polyvar, Banca, NIST. All of the three databases have their own specificity that justifies their use.

Banca

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. The world model is the concatenation of two gender dependent world models. This database contains three recording conditions defined as controlled (acquired in an office with only one person), degraded (acquired in several offices of several people) and adverse (recorded in a public area) and is provided with 7 different protocols. We have chosen to use protocol P, which

L. Devroye and L. Gyorfi. *A Probabilistic Theory of Pattern Recognition.* Springer, 20 February 1997.

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

we consider the most realistic: 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. Even if this database is small, it is still interesting because of the several recording conditions, and because the impostors pronounce the same sentence as the client.

All hyper-parameters of the GMM based baseline system are tuned: the number of ML iterations to train the world model, the number of iterations for the MAP adaptation, the number of Gaussians, the variance flooring factor and the MAP adaptation factor. All were selected on the development set to minimize the EER and are given in Table 4.1. The hyper-parameters for the SVM GLDS kernel are given in Table 4.2. When we vary C, from a certain value up to ∞ we keep a maximum of support vectors (this corresponds to the optimal solution found on the development set for all databases). We will use in the following the notation $\rightarrow \infty$ to express this.

Table 4.1. Summary of the hyper-parameters for GMM based systems on the Banca database

# of ML Iterations	# of MAP Iterations	# of Gaussians	MAP Factor: λ in (2.6)	Variance Flooring in [%]
25	5	400	0.5	60

Table 4.2. Summary of the hyper-parameters for the SVM based system on the Banca database ($\rightarrow$ means "tends to").

Degree of the GLDS kernel	C in (2.9)
3	$\rightarrow \infty$

The SVM system is based on a GLDS kernel of degree 3, as originally proposed by Campbell et al. (2005). T-normalization was not performed because all recordings were done using only one microphone.

Table 4.3. Results on the Banca database: GMMs and SVMs

	GMMs	SVMs
HTER [%]	1.39	6.94
95% Confidence	±1.03	±2.15

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.

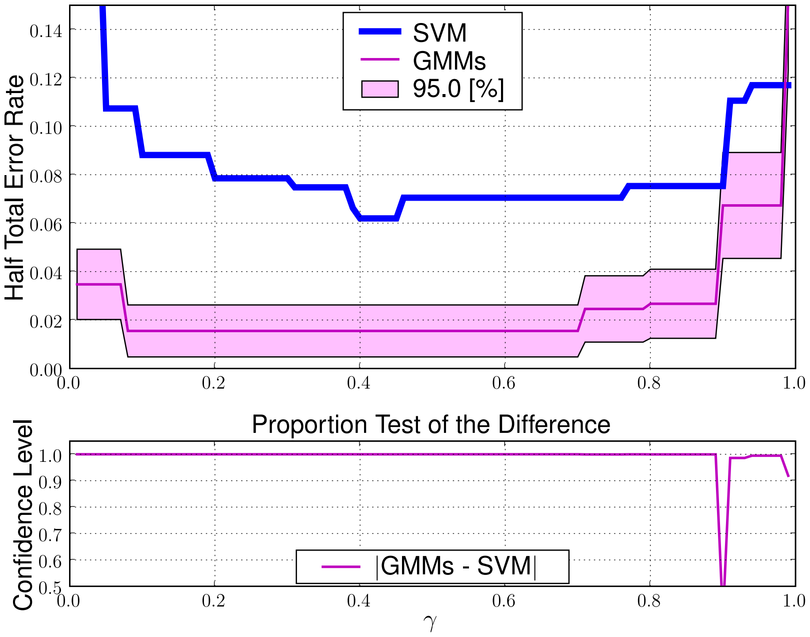


Figure 4.1. EPC curves on the test set of the Banca database: GMMs and SVMs.

Figure 4.1 and Table 4.3 show the results on the Banca database. We can see that the GMM based system outperforms significantly the SVM based system.

Polyvar

The Polyvar telephone database (Chollet et al., 1996), contains a development and a test set of 19 clients (12 men and 7 women) each, as well as another population of 56 speakers (28 men and 28 women) used to train the world model. The world model is the concatenation of two gender dependent world models. 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

G. Chollet, J.-L. Cochard, A. Constantinescu, C. Jaboulet, and P. Langlais. Swiss french polyphone and polyvar: telephone speech databases to model inter- and intra-speaker variability. IDIAP-RR 01, IDIAP, 1996.

protocol, only same word accesses are kept.

The hyper-parameters of GMM based systems where tuned using the same method as for the Banca database, minimizing the EER over the development set and Table 4.4 gives a summary of the obtained hyper-parameters.

Table 4.4. Summary of the hyper-parameters for GMM based systems on the Polyvar database

# of ML Iterations	# of MAP Iterations	# of Gaussians	MAP Factor: λ in (2.6)	Variance Flooring in [%]
25	5	200	0.2	10

The SVM system is, once again, based on a GLDS kernel of degree 3 originally proposed by Campbell et al. (2005). T-normalization was not performed because all recordings were done with the same kind of telephone (land line ISDN). The hyper-parameters of the SVM based system are the same as for the Banca database and are given in Table 4.2.

Table 4.5. Results on the Polyvar database: GMMs vs SVMs.

	GMMs	SVMs
HTER [%]	4.77	4.49
95% Confidence	±0.33	±0.32

Figure 4.2 and Table 4.5 show that SVMs and GMMs should be considered as equivalent for most values of $\gamma < 0.7$ while the SVM based system outperforms the GMM based system for most values of $\gamma > 0.7$.

NIST

The NIST database is a subset of the database that was used for the *NIST 2002 and 2003 Speaker Recognition Evaluation*, which comes from the second release of the cellular switchboard corpus, Switchboard Cellular - Part 2, of the Linguistic Data Consortium. This data was used as test set while the world model data and the development data comes from previous NIST campaigns. For both development and test clients, there were about 2 minutes of telephone speech used to train the models and each test access was less than 1 minute long. Only female data are used and thus only a female world model is used. The development population consisted of 100 females, while the test set is composed of 191 females. 655 different records are used to compute the world model or as

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.

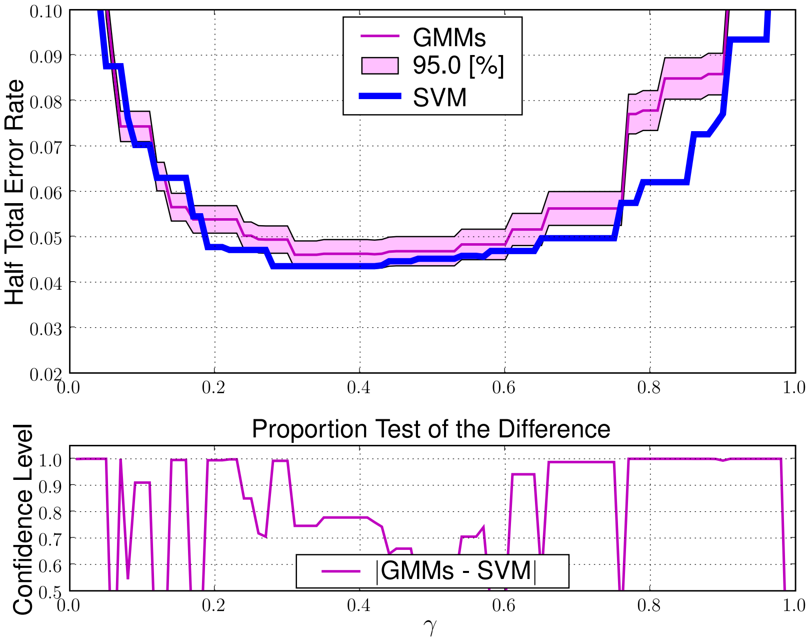


Figure 4.2. EPC curves on the test set of the Polyvar database: GMMs vs SVMs.

negative examples for the discriminant models. The total number of accesses in the development population is 3931 and 17578 for the test set population with a proportion of 10% of true target accesses. Only test accesses between 15 and 45 seconds are considered as the primary condition in the NIST campaign (see http://www.nist.gov/speech/tests/spk/2003 for the evaluation plan).

Table 4.6 gives a summary of the hyper-parameters used for GMM based experiments after selection based on minimizing EER on the development set. T-normalization is performed using (5.43) for the GMM based system. Figure 4.3 shows the improvement obtained by the T-normalization and justifies the use of score normalization for the GMM based system on NIST database. No score normalization procedure is applied for SVMs GLDS based kernel due to the computational cost and the small expected improvement as explained later in Chapter 5.

The hyper-parameters of SVMs based system are given in Table 4.2 and are once again the same as the two precedent databases.

Figure 4.4 and Table 4.7 show that the SVM based system outperforms the GMM based system for small values of γ and that the GMM based system

Table 4.6. Summary of the hyper-parameters for GMMs based systems on the NIST database

# of ML Iterations	# of MAP Iterations	# of Gaussians	MAP Factor: λ in (2.6)	Variance Flooring in [%]
25	5	100	0.5	60

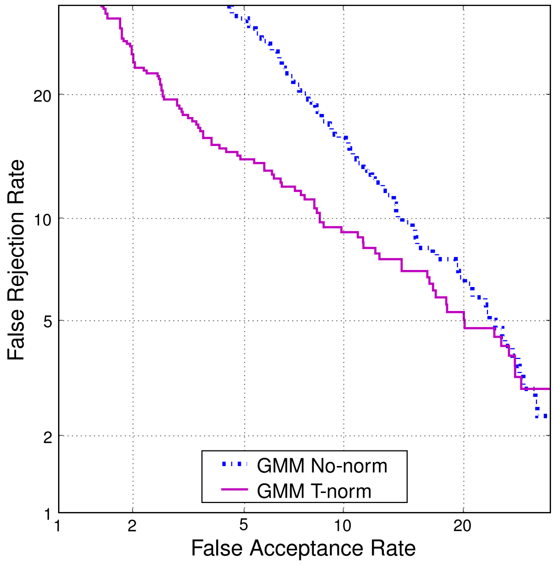


Figure 4.3. DET curves on the development set of the NIST database using, or not, the T-normalization procedure.

outperforms SVM based system for the other values of γ.

4.3 Conclusion

Except for the Banca database, both the SVM and GMM based systems are more or less equivalent. The SVM based system is easy to tune because the only hyper-parameters are the degree of the polynomial expansion and C in (2.9). In all cases the optimal value for degree was 3. We have also noted

Table 4.7. EPC curves on the test set of the NIST database: SVM v.s. GMM + T-norm

	GMMs + T-norm	SVM
HTER [%]	8.68	11.06
95% Confidence	±0.84	±1.05

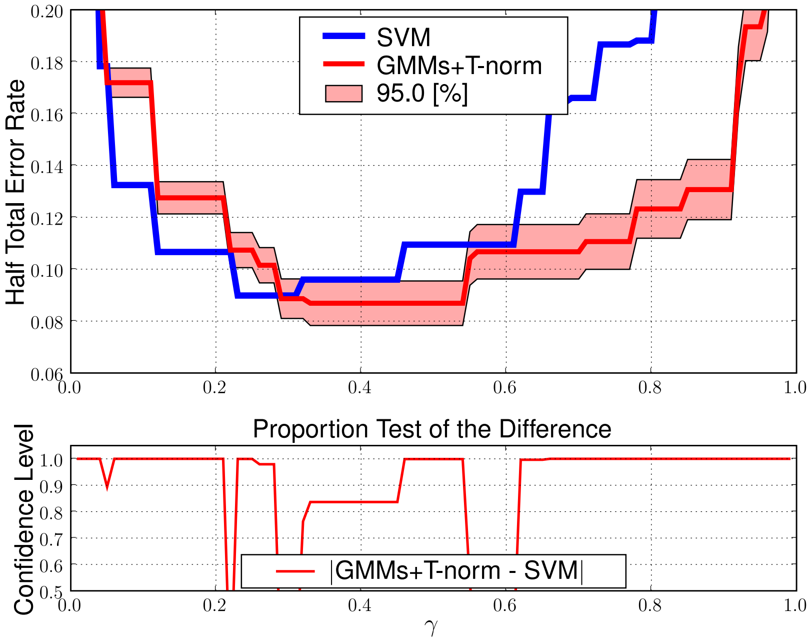


Figure 4.4. Results on the test set of the NIST database: GMMs + T-norm vs SVMs.

that the C value should be large. That means that SVMs maximize the margin without accepting examples in the margin. This can be explained by the fact that only few positive training examples are available and the cost function is not optimal for highly unbalanced class problem. In order to make use of C, the cost function should probably be modified. Even if it seems comfortable to have no hyper-parameter to tune, it also means there is no way to adjust the capacity of the SVM models, which can be important to expect improvements of the SVM performance.

The original Banca database and its protocol descriptions was published in:

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

The Polyvar database and its protocol descriptions was published in:

CONTRIB F. Bimbot, M. Blomberg, L. Boves, G. Chollet, C. Jaboulet, B. Jacob, J. Kharroubi, J. Koolwaaij, J. Lindberg, J. Mariéthoz, C. Mokbel, and H. Mokbel. An overview of the picasso project research activities in speaker verification for telephone applications. In *6th European Conference on Speech Communication and Technology — Eurospeech'99*, volume 5, pages 1963–1966, Budapest, Hungary, september 1999

The SVM based system never outperformed significantly the GMM based system.

- Does that mean that non-discriminant models are the best solution for speaker verification?
- Are GMM based systems really non-discriminant?
- Is the statistical framework applicable to SVMs?
- Is T-normalization also applicable to the SVMs based approaches?

In the next chapter we address these questions in order to have a good starting point to develop new discriminant approaches.

5 Text-Independent Speaker Verification: a Machine Learning Perspective

In order to propose new approaches based on discriminant models, we first need to define a general framework for speaker verification that would include several kinds of models: probabilistic models such as GMMs and non-probabilistic models such as SVMs. This framework should also enable the use of posterior probability models such as some kinds of multi-layer perceptrons (MLP). It is interesting to note that the normalization factor added empirically to GMMs will appear naturally for posterior probability based models.

The main purpose of this thesis is to use discriminant models for text-independent speaker verification. We should first try to give a definition of discriminant models. Moreover, GMMs are often used as state-of-the-art models and they are usually considered as non-discriminant. This is true in the sense that they try to estimate the data density of each positive and negative class independently. Here, we show that, after applying some modifications proposed by the speaker verification community in order to reach state-of-the-art performance, the models become discriminant and can be seen as a mixture of linear classifiers.

In this chapter, we also propose a unified framework that includes most score normalization techniques used in text-independent speaker verification. Furthermore, an implementation of two of the most common techniques, the so-called T- and Z-normalizations, are proposed in this novel framework. While the two approaches are not strictly equivalent, in practice they give similar results. In fact, this new framework can be used to understand the assumptions that are implicit when using T- and Z-normalization. Moreover, it can also been used to develop new normalization techniques.

The outline of this chapter goes as follows. In Section 5.1, we present a general framework to use probability and non-probability based models for speaker verification. In Section 5.2, we define what a discriminant model is and analyze whether GMMs are discriminant or not. Finally, Section 5.3 presents a new statistical framework for score normalization methods, such as T- and Z- normalizations.

5.1 Framework

Person authentication systems are in general designed in order to let genuine clients access a given service while forbidding it to impostors. In this thesis, we consider the problem from a machine learning point of view and we treat it independently for each speaker.

There are some specificities that make speaker verification different from a standard two-class classification problem. First, the input data are variable size sequences: indeed, the length of each sequence depends on the speaker rate and the phonetic content of the sentence. Furthermore, only few client training examples are available: in a real application, it is not possible to ask a client to speak during several hours or days in order to capture the entire variability of his voice. We have usually between one and three utterances of each sentence. Finally, the impostor distribution is not known: we have no idea of what an impostor is in a "real" application. In order to simulate impostor accesses, we normally use other speakers in the database. This implies that the intra-impostor distance distribution is the same as the impostor-client distance distribution. This also means that plenty of impostor accesses are usually available, often more than 1000, which makes the problem highly unbalanced. All these specificities are important and suggest that machine learning algorithms should be adapted to this specific task. Let us first define a general framework for this problem.

As we have already seen, this is a two-class classification task 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:

$$f_{\Theta_{S_i}}(\mathbf{X}) > \Delta_{S_i} \tag{5.1}$$

for all accesses $\mathbf{X}$ coming from S_i and only for them.

In order 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*:

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

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, N_{Tr} 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 minimizes a loss function $Q()$ which returns low values when $f_\Theta(\mathbf{X}_l)$ is near y_l and high values otherwise:

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

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.

Because of a lack of data available for each client, it is not possible to search for a client dependent decision threshold Δ_{S_i} in (5.1). Let us first define a set of clients, called development set, different from the clients used for the test set and defined as:

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

where S_l is the claimed identity corresponding to the example $\mathbf{X}_l$ and N_{Dev} is the total number of sequences in the development set. We are searching for a client independent decision threshold $\Delta_{S_i} \approx \Delta$ that minimizes a loss function $Q_{thrd}()$, for example the EER as defined in (3.5):

$$\Delta^* = \arg\min_{\Delta} Q_{thrd}(Dev, \Delta). \tag{5.2}$$

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

Statistical Framework

State-of-the-art text independent speaker verification systems are based on statistical generative models. We are interested in $P(C|\mathbf{X}, S_i)$: the probability that a client C has pronounced the sentence $\mathbf{X}$ and claimed the identity S_i. Using Bayes theorem, we can write it as follows:

$$P(C|\mathbf{X}, S_i) = \frac{p(\mathbf{X}, S_i|C)P(C)}{p(\mathbf{X}, S_i)}. \tag{5.3}$$

In order to decide whether or not client S_i has indeed pronounced sentence $\mathbf{X}$, we compare $P(C|\mathbf{X}, S_i)$ to the probability that any other speaker proclaiming identity S_i has pronounced $\mathbf{X}$, which we write $P(\bar{C}|\mathbf{X}, S_i)$. We then accept the claimant if:

$$P(C|\mathbf{X}, S_i) > P(\bar{C}|\mathbf{X}, S_i). \tag{5.4}$$

Using (5.3), (5.4) can then be rewritten as:

$$\frac{p(\mathbf{X}, S_i|C)P(C)}{p(\mathbf{X}, S_i)} > \frac{p(\mathbf{X}, S_i|\bar{C})P(\bar{C})}{p(\mathbf{X}, S_i)}. \tag{5.5}$$

Rewriting (5.5) in order to isolate terms that do not depend on $\mathbf{X}$, we obtain:

$$\frac{p(\mathbf{X}, S_i|C)}{p(\mathbf{X}, S_i|\bar{C})} > \frac{P(\bar{C})}{P(C)}. \tag{5.6}$$

Using the conditional probabilities law, we get:

$$\frac{p(\mathbf{X}|S_i, C)P(S_i|C)}{p(\mathbf{X}|S_i, \bar{C})P(S_i|\bar{C})} > \frac{P(\bar{C})}{P(C)}. \tag{5.7}$$

Once again, isolating terms that do not depend of $\mathbf{X}$, we get:

$$\frac{p(\mathbf{X}|S_i, C)}{p(\mathbf{X}|S_i, \bar{C})} > \frac{P(\bar{C})P(S_i|\bar{C})}{P(C)P(S_i|C)}. \tag{5.8}$$

Using Bayes rule, we finally obtain likelihoods:

$$\frac{p(\mathbf{X}|S_i, C)}{p(\mathbf{X}|S_i, \bar{C})} > \frac{P(\bar{C}|S_i)}{P(C|S_i)} \approx \Delta \tag{5.9}$$

where the ratio of probabilities on the right hand side of the equation can be replaced by the decision threshold Δ.

From (5.9), one can derive two approaches, one based on likelihood models and one based on posterior models.

GMM Based Approach

A statistical framework can be defined using the following general form:

$$f_{\Theta_{S_i}}(\mathbf{X}) = \frac{f_{\Theta^+_{S_i}}(\mathbf{X})}{f_{\Theta^-_{S_i}}(\mathbf{X})} = \frac{p(\mathbf{X}|S_i, C)}{p(\mathbf{X}|S_i, \bar{C})}$$

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

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

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 $f_{\Theta^+_{S_i}}()$ is based on a Maximum A Posteriori (MAP) adaptation scheme and can be written as follows:

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

where Tr_+ is the subset of examples of Tr where $y_l = 1$. This MAP approach puts some prior on the distribution of $\Theta_{S_i}^{+}$ in order to constrain them to some reasonable values.

We thus need to create a world model of $p(\mathbf{X}|S_i, \bar{C})$, as well as a client model $p(\mathbf{X}|S_i, C)$ for every potential speaker.

Posterior Probability Models

Multi Layer Perceptron (MLP) are known to be good posterior probability estimators (Lippmann, 1992). In order to try to use them directly as discriminant models, we derive the equation of the probabilistic framework in order to obtain a posterior probability form. Using (5.9) and making the assumption that all T frames $\mathbf{x}_t$ of $\mathbf{X}$ are independent, as is done with GMMs, we obtain:

$$\prod_{t=1}^{T} \frac{p(\mathbf{x}_t|S_i, C)}{p(\mathbf{x}_t|S_i, \bar{C})} > \frac{P(\bar{C}|S_i)}{P(C|S_i)}. \tag{5.10}$$

Using the conditional probability law, we get:

$$\prod_{t=1}^{T} \frac{p(\mathbf{x}_t, S_i, C)P(S_i, \bar{C})}{p(\mathbf{x}_t, S_i, \bar{C})P(S_i, C)} > \frac{P(\bar{C}|S_i)}{P(C|S_i)}. \tag{5.11}$$

Using the conditional probability law again, we get:

$$\prod_{t=1}^{T} \frac{P(C|\mathbf{x}_t, S_i)P(\bar{C}|S_i)}{P(\bar{C}|\mathbf{x}_t, S_i)P(C|S_i)} > \frac{P(\bar{C}|S_i)}{P(C|S_i)}. \tag{5.12}$$

Regrouping identical terms, we obtain:

$$\prod_{t=1}^{T} \frac{P(C|\mathbf{x}_t, S_i)}{P(\bar{C}|\mathbf{x}_t, S_i)} > \frac{P(C|S_i)^{T-1}}{P(\bar{C}|S_i)^{T-1}}. \tag{5.13}$$

Taking the log, we obtain:

$$\frac{1}{T-1}\sum_{t=1}^{T} \log \frac{P(C|\mathbf{x}_t, S_i)}{P(\bar{C}|\mathbf{x}_t, S_i)} > \log \frac{P(C|S_i)}{P(\bar{C}|S_i)}. \tag{5.14}$$

We can normally assume that $P(\bar{C}|\mathbf{x}_t, S_i) = 1 - P(C|\mathbf{x}_t, S_i)$; we thus obtain:

$$f_{\Theta_{S_i}}(\mathbf{X}) = \frac{1}{T-1}\sum_{t=1}^{T} \log \frac{P(C|\mathbf{x}_t, S_i)}{1 - P(C|\mathbf{x}_t, S_i)} > \Delta. \tag{5.15}$$

REF[0] R. P. Lippmann. Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities. *Neural Computation*, 3:461–483, 1992.

where the ratio of log probabilities is usually replaced by the decision threshold Δ.

In practice, with generative models, we normalize the LLR by the number of frames T in order to be independent of the length of the access. Here, this factor appears naturally from the equations.

In this case (5.15) is directly our scoring function $f_{\Theta_{S_i}}(\mathbf{X})$. When the model used is an MLP with a single output passed through a sigmoid function, the decision function can be simplified as:

$$f_{\Theta_{S_i}}(\mathbf{X}) = \frac{1}{T-1}\sum_{t=1}^{T} g(\mathbf{x}_t) \tag{5.16}$$

where $g(\mathbf{x}_t)$ is the input of the sigmoid function. The loss function used to train $f_{\Theta_{S_i}}(\mathbf{X})$ can simply be to minimize the mean squared error or better, the cross-entropy:

$$\Theta^*_{S_i} = \arg\min_{\Theta_{S_i}} \sum_{(\mathbf{X}_l, y_l)\in Tr} \sum_{t=1}^{T_l} \log\Big(1 + \exp(-y_l f_{\Theta_{S_i}}(\mathbf{x}^l_t))\Big). \tag{5.17}$$

A Score Based Framework

If instead of relying on models generating probabilities, we want to use non-statistical models such as SVMs, as described in the remaining of this thesis, the framework described at the beginning of this section can be applied directly and no probabilistic interpretation need to be given to $f_{\Theta_{S_i}}()$. In Chapter 2 the parametric form of function $f_{\Theta_{S_i}}()$ and the loss function $Q()$ used by SVMs have been described in details. Using the trick described by Platt (2000), one can force SVMs to output probabilities. However, this only approximates probabilities, but one cannot consider SVMs to be probabilistic models.

5.2 Are GMMs Discriminant?

As we have already seen in this thesis, one of the state-of-the-art models is based on GMMs. In the speaker verification domain, most researchers use the term “generative models”, opposing them to “discriminant models”. By definition, a generative model can generate data but nothing prevent it to be discriminant. Conversely, a “diabolo” neural network (trained to reconstruct the inputs) for example, cannot generate data but is non-discriminant in the sense that it is trained using only one class of examples. In this thesis, we

J. C. Platt. Probabilities for SV machines. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

consider a model as discriminant if the parameters of this model are trained using the examples of more than one class, typically using client and impostor data. Conversely, a model is considered non-discriminant only if its parameters are trained using examples of only one class. Basically, the cost function decides if a model is discriminant or not. Given this new definition, can we say whether a GMM based system is discriminant or not?

When $P(\mathbf{X}|S_i, C)$ and $P(\mathbf{X}|S_i, \bar{C})$ are trained separately using an ML criterion, the two models are independent and thus we can consider the resulting model as non-discriminant. However, as explained in Chapter 2, several modifications have been used to reach state-of-the-art performance, and some of them may suggest that the resulting model is not optimized to have a good data density estimation. Especially the use of a MAP adaptation procedure seems to make the GMMs discriminant. In (Mariéthoz and Bengio, 2002), we tried to use different kinds of adaptation methods, but only MAP adaptation seems to be so efficient. As a matter of fact, using MAP, the client parameters are a linear combination of the world model parameters and the new observed data. Thus, at least, the client model should be considered as discriminant. Given these intuitions, we can now try to make some simplifications on the GMM based system in order to have an interpretation of the resulting decision function.

GMMs: a Mixture of Linear Classifiers

As we know, GMMs are often used as data density estimators, but also as clustering algorithms. The EM training algorithm can be seen as a soft version of the well-known K-Means clustering algorithm. In the case of speech frames, one can thus expect that each Gaussian represents somehow a sub-unit of speech. Moreover, the LLR between the world and the client model is used to take the decision and the client model parameters are adapted from the world model and thus each Gaussian in the world model has its own corresponding Gaussian in the client model. Applying some approximations, such as forcing each frame to be represented by only one Gaussian, GMM based systems can thus be seen as performing the verification in two steps: first the frames are clustered into sub-units of speech; then the classification is done using a local classifier composed of a couple of Gaussians (one from the client model, and the corresponding one in the world model). In order to consider couples of Gaussians, we first need to enforce an exact correspondence between the world and client Gaussians. This is in fact already the case when MAP adaptation is

J. Mariéthoz and S. Bengio. A comparative study of adaptation methods for speaker verification. In *International Conference on Spoken Language Processing ICSLP*, pages 581–584, Denver, CO, USA, September 2002. IDIAP-RR 01-34.

used to train client models. More precisely, we chose to adapt only the mean parameters of the world model Ω, as usually done in speaker verification, using the following MAP equation (same as (2.6)):

$$\hat{\boldsymbol{\mu}}_g = \lambda \boldsymbol{\mu}_{g,\Omega} + (1-\lambda)\boldsymbol{\mu}_{g,C}. \tag{5.18}$$

Let us now assign each frame $\mathbf{x}_t$ to only one Gaussian as follows: let $g^*_{t,\theta}$ be the Gaussian in model Θ that best represents $\mathbf{x}_t$:

$$g^*_{t,\theta} \quad = \quad \arg\max_g \log w_g \; p(\mathbf{x}_t|\Theta, g) \tag{5.19}$$

where w_g is the weight corresponding to the Gaussian g.

We can compute the corresponding approximation of llr (2.18) as follows:

$$\mathrm{llr}_v = \frac{1}{T}\sum_t \log \frac{p(\mathbf{x}_t|S_i, C, g^*_{t,\Theta_{S_i}})}{p(\mathbf{x}_t|S_i, \Omega, g^*_{t,\Theta_\Omega})} \; . \tag{5.20}$$

Note that there is no constraint in (5.20) that guarantees that a given frame is assigned to the same Gaussian index in the client and world models. In order to enforce this, a synchronous alignment procedure, originally applied for HMMs (Mariéthoz et al., 1999), can be used:

$$g^*_t = \arg\max_g \;\; \beta \log w_g \; p(\mathbf{x}_t|S_i, \Omega, g) + (1-\beta) \log w_g \; p(\mathbf{x}_t|S_i, C, g) \tag{5.21}$$

where β is a trade-off between placing our confidence in the world or the client model. Using this synchronous alignment, we define a new score llr_s as follows:

$$\mathrm{llr}_v \cong \mathrm{llr}_s = \frac{1}{T}\sum_t \log \frac{p(\mathbf{x}_t|S_i, C, g^*_t)}{p(\mathbf{x}_t|S_i, \Omega, g^*_t)} \; . \tag{5.22}$$

We can now express (5.22) as a sum over all couples of Gaussians as follows:

$$\mathrm{llr}_s = \sum_g \frac{T(g)}{T} \, \mathrm{llr}_s(g) \;\; \text{where} \;\; \mathrm{llr}_s(g) = \frac{1}{T(g)} \sum_{t=1}^{T(g)} \log \frac{p(\mathbf{x}_{r_g(t)}|S_i, C, g)}{p(\mathbf{x}_{r_g(t)}|S_i, \Omega, g)} \; . \tag{5.23}$$

where $T(g)$ is the number of frames assigned to the couple of Gaussians g, and $r_g(t)$ returns the index of the t^{th} frame assigned to the cluster g. This can be seen as a mixture of classifiers where the weight assigned to each expert is $T(g)/T$.

J. Mariéthoz, Dominique Genoud, Frédéric Bimbot, and Chafik Mokbel. Client / world model synchronous alignement for speaker verification. In *6th European Conference on Speech Communication and Technology — Eurospeech'99*, Budapest, Hungary, September 1999.

It is interesting to analyze more deeply the local classifier for each frame $\mathbf{x}_t$. If we train the client model using MAP by adapting only the **mean** parameters keeping variances and weights the **same** as the world model, and if we force the EM algorithm to perform only **one** iteration we obtain:

$$\begin{aligned}
\mathbf{llr}_s(g, \mathbf{x}_{t(g)}) &= \log \frac{p(\mathbf{x}_{t(g)}|S_i, C, g)}{p(\mathbf{x}_{t(g)}|S_i, \Omega, g)} && (5.24)\\
&= \log \frac{1}{\sqrt{2\pi\boldsymbol{\sigma}_g^2}} - \left(\frac{\mathbf{x}_{t(g)} - \hat{\boldsymbol{\mu}}_g}{2\,\boldsymbol{\sigma}_g}\right)^2 - \log \frac{1}{\sqrt{2\pi\boldsymbol{\sigma}_g^2}} + \left(\frac{\mathbf{x}_{t(g)} - \boldsymbol{\mu}_{g,\Omega}}{2\,\boldsymbol{\sigma}_g}\right)^2 \\
&= \frac{\hat{\boldsymbol{\mu}}_g - \boldsymbol{\mu}_{g,\Omega}}{\boldsymbol{\sigma}_g^2}\left(\mathbf{x}_{t(g)} - \frac{\hat{\boldsymbol{\mu}}_g + \boldsymbol{\mu}_{g,\Omega}}{2}\right). && (5.25)
\end{aligned}$$

We can see in (5.25) that $\boldsymbol{\sigma}_t^2$ can be factorized easily and appears in the weight of each expert. More formally we obtain:

$$\mathrm{llr}_s = \sum_g^{N_g} \frac{T(g)}{\boldsymbol{\sigma}_g^2\, T}\mathrm{llr}_s(g) \;\text{ where }\; \mathrm{llr}_s(g) = \frac{1}{T(g)}\sum_{t(g)}^{T(g)}(\hat{\boldsymbol{\mu}}_g - \boldsymbol{\mu}_{g,\Omega})\left(\mathbf{x}_{t(g)} - \frac{\hat{\boldsymbol{\mu}}_g + \boldsymbol{\mu}_{g,\Omega}}{2}\right). \tag{5.26}$$

Remember (from Chapter 2) that until now it was difficult to interpret the use of the variance flooring in the context of density estimation. Indeed, the actual value of this hyper parameter is so huge in practice (between 10% and 60% of the global variance of the data) that it makes the distribution nearly uniform. On the other hand, interpreting the LLR as a mixture of linear classifier, variance flooring can be interpreted as pushing the weights of every experts to be equal. That tends to make the weight of each local classifier independent of the variance of the corresponding sub-acoustic unit. This suggests that we could learn these weights using a discriminant cost function.

Including (5.18) to (5.25), we obtain:

$$\frac{\boldsymbol{\mu}_{g,C} - \boldsymbol{\mu}_{g,\Omega}}{\boldsymbol{\sigma}_g}\left(\frac{\mathbf{x}_{t(g)}}{\boldsymbol{\sigma}_g} - \left[(1-\lambda)\frac{\boldsymbol{\mu}_{g,C} + \boldsymbol{\mu}_{g,\Omega}}{2\boldsymbol{\sigma}_g} + \lambda\,\frac{\boldsymbol{\mu}_{g,\Omega}}{\boldsymbol{\sigma}_g}\right]\right) \tag{5.27}$$

Figure 5.1 shows that the corresponding decision function is a perpendicular bisector. The adaptation factor λ affects only the bias while the slope of the decision function is still the same. The adaptation factor varies the decision function between the perpendicular bisector and the line passing by the non-adapted mean vector.

Experimental Results

In order for this interpretation to be valid, we need to make several simplifications as already explained: training the client model by adapting only the mean

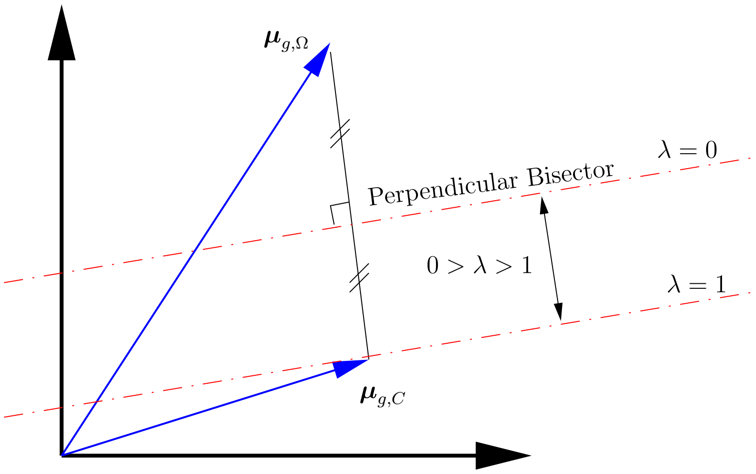


Figure 5.1. Perpendicular bisector interpretation.

vector for only one EM iteration, plus some approximations of the LLR as detailed in (5.19) - (5.22). To verify whether these simplifications are reasonable, we performed some experiments described as follows.

First a GMM based system using several iterations of EM during the MAP adaptation procedure is referred to as the baseline system. Then the approximation done using (5.19) and with only one EM iteration is performed to validate the max approximation. Finally the synchronous alignment experiments are done to validate the approximation given by (5.22). Two values of β in (5.21) are given: aligning on the world model ($\beta = 1$), or aligning on the client model ($\beta = 0$). All the results are performed on the NIST database described in Chapter 4 and are presented in Table 5.1 and Figure 5.2.

Table 5.1. Results on the NIST database: GMM baseline results, max approximation with only one iteration of EM training, synchronous alignment on client and on world model.

	Baseline	Max. 1 Iter.	Sync. $\beta = 1$	Sync. $\beta = 0$
HTER [%]	8.68	8.88	**9.72**	8.68
95% Confidence	±0.84	±0.82	±0.89	±0.82

All the simplifications seem reasonable as all approaches give similar results except the synchronous alignment using the world model $\beta = 1$.

We show results with the alignment on the world model only because it can be useful to speed up the testing procedure when the T-normalization is used. Indeed in this case, we have to compute the best Gaussian only once for all T-norm models. Unfortunately, even if this is feasible, the performance is significantly degraded.

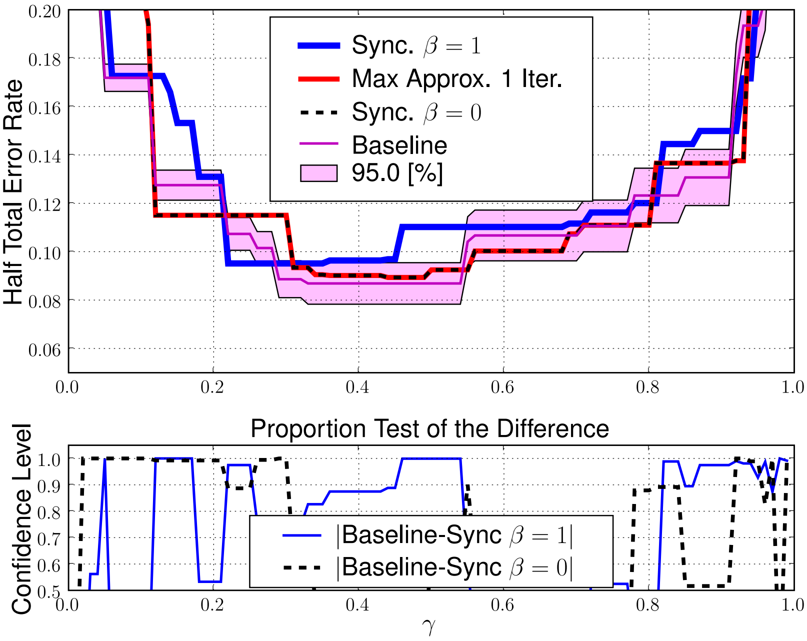


Figure 5.2. Results on the NIST2002 database: GMM baseline results, max approximation with only one iteration for EM training, synchronous alignment on client and on world model.

Note that, when T-normalization is applied to the max approximation with an alignment only on the client model, the performance is exactly the same because the world model contribution is canceled due to the T-normalization.

Discussion

We have shown that a GMM based state-of-the-art system can be seen as a mixture of linear classifiers. It is interesting to note that all the “tricks” used to make these generative models work now have a new meaning: (1) the normalization factor added empirically to be independent of the length of the sequence appears naturally in the discriminant framework; (2) the variance flooring that makes the new density estimation quasi uniform in the generative model transforms the weight of each local expert to be uniform and suggests to use a discriminant criterion to be chosen correctly; (3) finally, the MAP adaptation factor represents the bias of each local expert and can thus be seen as a generalization factor. It is particularly true given the fact that no impostor distribution is really available and thus the confidence on this estimation can be represented by the MAP adaptation factor.

5.3 Score Normalization

Text-independent speaker verification systems have evolved through time (Bimbot et al., 2004). The first systems had reasonable performance only in controlled conditions (no noise, same channel, same gender, etc). Over the years, researchers have improved their systems for unmatched conditions, thanks largely to score normalization techniques. Here, we propose a unified framework that explains several score normalization techniques used in text-independent speaker verification. Furthermore, an implementation of two of the most common techniques, the so-called T- and Z-normalization (Auckenthaler et al., 2000), is proposed here in this novel framework. While the two approaches are not strictly equivalent, in practice they give similar results. In fact, this new framework can be used to understand the assumptions that are implicit when using T- and Z-normalization. Moreover, it can also be used to develop new normalization techniques.

Unified Framework for Score Normalization

Most state-of-the-art text-independent speaker verification systems use linear score normalization functions of the form:

$$\mathrm{llr}_{norm} = \frac{\mathrm{llr} - \mu}{\sigma} > \Delta \tag{5.28}$$

where μ and σ are respectively the mean and the standard deviation of a normal distribution of LLRs. These parameters are then estimated differently for each type of score normalizations. We propose a unified framework for all kinds of normalization of the form of (5.28), and also other non-linear functions. We further propose an implementation for the two well-known T- and Z-normalization techniques.

We have seen that in text-independent speaker verification we are interested in the probability that a speaker S_i has pronounced a sentence X. Let us now consider the LLR as an additional random variable, and let us introduce it in the original framework by looking at $P(C|\mathrm{llr}, \mathrm{X}, S_i)$, the probability that a speaker S_i has pronounced a sentence X and obtained an LLR of llr. Using

F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovsk-Delacrétaz, and D. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Applied Signal Processing*, 4:430–451, 2004.

R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10:42–54, 2000.

the same approach as in Section 5.1, we obtain:

$$P(C|\mathrm{llr}, \mathrm{X}, S_i) > P(\bar{C}|\mathrm{llr}, \mathrm{X}, S_i). \tag{5.29}$$

Applying the conditional law of probabilities, we obtain:

$$P(C, \mathrm{llr}, \mathrm{X}, S_i) > P(\bar{C}, \mathrm{llr}, \mathrm{X}, S_i). \tag{5.30}$$

Applying the conditional law of probabilities, we obtain:

$$p(\mathrm{llr}|C, \mathrm{X}, S_i)p(\mathrm{X}, C, S_i) > p(\mathrm{llr}|\bar{C}, \mathrm{X}, S_i)p(\mathrm{X}, \bar{C}, S_i). \tag{5.31}$$

Applying the conditional law of probabilities on the second term of each part of the inequation, we obtain:

$$p(\mathrm{llr}|C, \mathrm{X}, S_i)p(\mathrm{X}|C, S_i)P(C|S_i) > p(\mathrm{llr}|\bar{C}, \mathrm{X}, S_i)p(\mathrm{X}|\bar{C}, S_i)P(\bar{C}|S_i) \tag{5.32}$$

$$\frac{p(\mathrm{llr}|C, \mathrm{X}, S_i)p(\mathrm{X}|C, S_i)}{p(\mathrm{llr}|\bar{C}, \mathrm{X}, S_i)p(\mathrm{X}|\bar{C}, S_i)} > \frac{P(\bar{C}|S_i)}{P(C|S_i)}. \tag{5.33}$$

Taking the logarithm, we finally obtain:

$$\mathrm{llr}_{norm} = \log \frac{p(\mathrm{llr}|C, \mathrm{X}, S_i)}{p(\mathrm{llr}|\bar{C}, \mathrm{X}, S_i)} + \mathrm{llr} > \log \frac{P(\bar{C}|S_i)}{P(C|S_i)} \approx \Delta\ . \tag{5.34}$$

Comparing equation (5.34) of this new framework with the original equation (2.18) shown in Chapter 2, we can see that a new term appears. It is the log of the ratio of two likelihoods estimated by two score distributions. The numerator represents the distribution of LLRs for a given access X and for client S_i. The denominator represents the distribution of LLRs for a given access X and for all impostors $\bar{C}$. We will see that, depending on how these two distributions are estimated, we can obtain classical score normalization techniques such as T-norm (when estimated on a test access) or Z-norm (when estimated for each client S_i).

Relation to Existing Normalization Techniques

T-norm

The T-norm, as introduced in (Auckenthaler et al., 2000) and (Navratil and Ramaswamy, 2003), estimates μ and σ as the mean and the standard deviation

R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10:42–54, 2000.

J. Navratil and Ganesh N. Ramaswamy. The awe and mystery of t-norm. In *Proc. of the European Conference on Speech Communication and Technology*, pages 2009–2012, 2003.

of the log likelihood ratios (LLRs) using models of a subset of impostors, for a particular test access X.

$$\mu_M = \frac{1}{M}\sum_m \mathrm{llr}_m(\mathrm{X}) \tag{5.35}$$

$$\sigma_M = \sqrt{\frac{1}{M}\sum_m (\mathrm{llr}_m(\mathrm{X}) - \mu_M)^2} \tag{5.36}$$

where M is the number of impostor models and llr_m is the score for the m^{th} impostor model for the particular access X. Using (5.28) we obtain:

$$\mathrm{llr}_{T-norm} = \frac{\mathrm{llr} - \mu_M}{\sigma_M} > \Delta\ . \tag{5.37}$$

Let us now show how it is possible to perform T-normalization using our new framework under reasonable assumptions.

Given (5.34), we must define two distributions, which will be here defined as Normal, as follows:

$$\hat{p}(\mathrm{llr}|C, \mathrm{X}, S_i) = \mathcal{N}(\mathrm{llr}; \mu_C, \sigma_C) \tag{5.38}$$

$$\hat{p}(\mathrm{llr}|\bar{C}, \mathrm{X}, S_i) = \mathcal{N}(\mathrm{llr}; \mu_{\bar{C}}, \sigma_{\bar{C}}) \tag{5.39}$$

where μ_C, σ_C are the parameters of the client distribution and $\mu_{\bar{C}}, \sigma_{\bar{C}}$ are the parameters of the impostor distribution. To obtain the T-norm we make the assumption that the standard deviations are equal:

$$\sigma_M = \sigma_C = \sigma_{\bar{C}}\ . \tag{5.40}$$

We thus obtain:

$$\begin{aligned}\log\frac{\hat{p}(\mathrm{llr}|C,\mathrm{X},S_i)}{\hat{p}(\mathrm{llr}|\bar{C},\mathrm{X},S_i)} &= -\frac{1}{2\sigma_M^2}\left((\mathrm{llr}-\mu_C)^2 - (\mathrm{llr}-\mu_{\bar{C}})^2\right) - \log\frac{\sqrt{2\pi\sigma_M^2}}{\sqrt{2\pi\sigma_M^2}} \\ &= \frac{\mu_C - \mu_{\bar{C}}}{\sigma_M^2}\left(\mathrm{llr} - \frac{\mu_C + \mu_{\bar{C}}}{2}\right)\ .\end{aligned} \tag{5.41}$$

If we now define the means as:

$$\begin{aligned}\mu_C &= \mathrm{llr} \\ \mu_{\bar{C}} &= \mu_M\end{aligned} \tag{5.42}$$

when $llr > \mu_M$. Otherwise, a reasonable thing to do is to reject directly without any normalization a claimed speaker if its obtained LLR is smaller than the average of LLRs over a subset of impostors.

We finally obtain:

$$\mathrm{llr}_{unified-T-norm} = \mathrm{llr} + \frac{(\mathrm{llr} - \mu_M)^2}{2\sigma_M^2} > \Delta\ . \tag{5.43}$$

Z-norm

The basis of Z-norm (Auckenthaler et al., 2000) is to test a speaker model against example impostor utterances and to use the corresponding LLR scores to estimate a speaker specific mean and standard deviation:

$$\begin{aligned} \mu_J &= \frac{1}{J}\sum_j \mathrm{llr}(\mathrm{X}_j) && (5.44)\\ \sigma_J &= \sqrt{\frac{1}{J}\sum_j (\mathrm{llr}(\mathrm{X}_j) - \mu_J)^2} && (5.45) \end{aligned}$$

where J is the number of impostor accesses.

Using a similar approach to T-normalization, the estimate of the two distributions needed for the proposed unified framework becomes:

$$\begin{aligned} \hat{p}(\mathrm{llr}|C, \mathrm{X}, S_i) &= \mathcal{N}(\mathrm{llr}; \mu_C, \sigma_C) && (5.46)\\ \hat{p}(\mathrm{llr}|\bar{C}, \mathrm{X}, S_i) &= \mathcal{N}(\mathrm{llr}; \mu_{\bar{C}}, \sigma_{\bar{C}}) && (5.47) \end{aligned}$$

with, again, the same standard deviation, $\sigma_J = \sigma_C = \sigma_{\bar{C}}$.

If we now define the means as follows:

$$\begin{aligned} \mu_C &= \mathrm{llr}\\ \mu_{\bar{C}} &= \mu_J && (5.48) \end{aligned}$$

when $llr > \mu_J$. Otherwise, we reject directly without any normalization a claimed speaker if its obtained LLR is smaller than the average of LLRs over a subset of impostors.

Then using (5.48) and (5.41) we obtain:

$$\mathrm{llr}_{unified-Z-norm} = \mathrm{llr} + \frac{(\mathrm{llr} - \mu_J)^2}{2\sigma_J^2} > \Delta\ . \tag{5.49}$$

R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10:42–54, 2000.

Discussion

In order to implement the standard T- and Z-norm using the new score normalization framework, we made some strong assumptions to fix the score distribution parameters. One can consider the choice of the mean parameters reasonable. At the opposite, fixing the standard deviation parameter to be the same for both the client and impostor score distributions seems less obvious. Indeed the variability of the impostor scores should be bigger than the variability of the client scores because the variability of the impostor accesses is obviously bigger than the variability of the client accesses. Even if usually only too few client accesses are available to have a good estimate for each client, one can imagine to use a set of other clients to estimate a client independent standard deviation as it is usually done for the decision threshold as explained in Section 5.1.

Comparison Between New and Classical Z- and T-norm

Here, we show the difference between the T-norm implementation found in the literature and our implementation using a unified framework. This demonstration can also be applied to Z-normalization.

The new implementation is given by:

$$\mathrm{llr}_{unified-T-norm} = \mathrm{llr} + \frac{(\mathrm{llr} - \mu_M)^2}{2\sigma_M^2} > \Delta \tag{5.50}$$

The classical method to implement T-norm is equivalent to the second term of the left side of (5.50) since:

$$\begin{aligned} \frac{(\mathrm{llr} - \mu_M)^2}{2\sigma_M^2} &> \Theta \\ (\mathrm{llr} - \mu_M)^2 &> \Theta\, 2\sigma_M^2 \\ (\mathrm{llr} - \mu_M)^2 - 2\Theta\, \sigma_M^2 &> 0 \\ \left[(\mathrm{llr} - \mu_M - \sqrt{2\Theta}\,\sigma_M)\cdot(\mathrm{llr} - \mu_M + \sqrt{2\Theta}\,\sigma_M)\right] &> 0 \end{aligned} \tag{5.51}$$

and if $\mathrm{llr} > \mu_M$ then we can simplify (5.51) further into:

$$\begin{aligned} \mathrm{llr} - \mu_M - \sqrt{2\Theta}\,\sigma_M &> 0 \\ \frac{\mathrm{llr} - \mu_M}{\sigma_M} &> \sqrt{2\Theta}\,. \end{aligned} \tag{5.52}$$

This inequation has a real solution only when $\Theta > 0$, which is true if $\mathrm{llr} > \mu_M$. This assumption is reasonable: we do not want to accept an access if the LLR on the client model is smaller than the average LLR obtained over a subset of impostors. Given this reasonable assumption we can see the standard T-norm as a simplification of the T-norm using our new unified framework.

Experiments

The goal of these experiments is to show that the proposed framework can indeed be used to perform T-norm or Z-norm while obtaining the same performance as the original methods, and, gaining some insight about the underlying assumptions.

Experimental Results

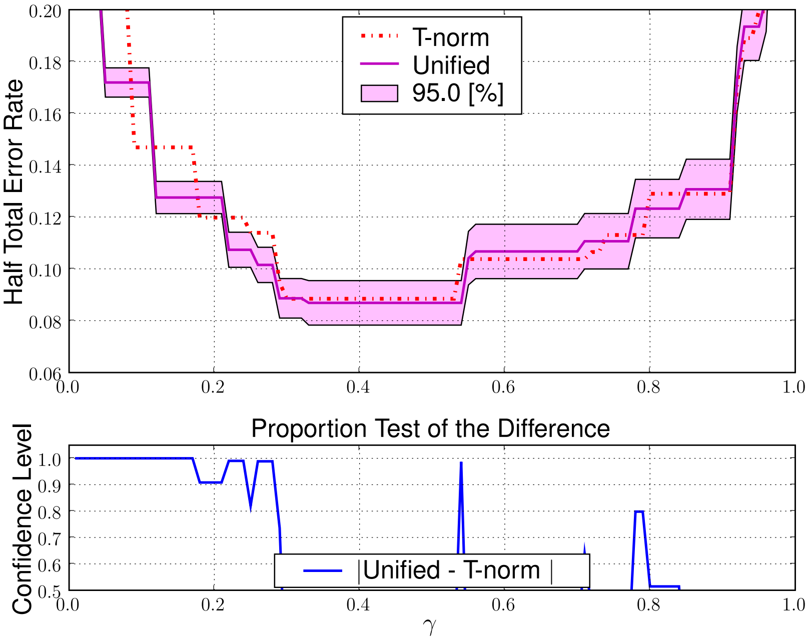


Figure 5.3. EPC curves on the NIST 2002 test set for the T-norm and unified framework T-norm systems.

To verify the validity of our framework and the underlying assumptions, we first compared the standard T-normalization and the version derived from the proposed framework. Figure 5.3 presents the results on the NIST database. On this database, the T-normalization is important since speakers have been recorded through different types of microphones. As can be seen, the two curves are most of the time not significantly different. These results show that the two approaches are equivalent. In fact they are perfectly equal if we remove llr in (5.43) and (5.49). Note that in (Mariéthoz and Bengio, 2005), we draw

[9] J. Mariéthoz and S. Bengio. A unified framework for score normalization techniques applied to text independent speaker verification. *IEEE Signal Processing Letters, Volume*

the same conclusions for the Z-normalization, but using an older version of the NIST database.

T-norm for SVM

Similarly to GMM based systems, it can also be useful to have a channel compensation procedure for SVM based systems. Channel compensation techniques try to compensate the difference of distortion produced by an acquisition system: microphone-compression-transmission. Indeed, some of the benchmark databases contain recordings using several kinds of channel transmission: land line, GSM, etc. Solomonoff et al. (2004) have proposed a channel compensation method by mapping the input vector data to a high dimensional space in order to perform the compensation in that space. This approach needs data to estimate the mapping and is not a score normalization technique as T-normalization.

If we want to perform T-norm using a score normalization approach, a naive approach consists of:

$$f_{\Theta_{S_i}}(\mathrm{X})_{T-norm-naive} = \frac{f_{\Theta_{S_i}}(\mathrm{X}) - \mu_M}{\sigma_M} \tag{5.53}$$

where $f_{\Theta_{S_i}}(\mathrm{X})$ is the output score of the SVM, while μ_M and σ_M are the mean and the standard deviation estimated using M impostor models.

Unfortunately SVMs are not able to output probabilities and the unified framework proposed before is thus not valid. Let us extend this framework to SVMs. Starting from (5.31) and replacing llr by the output score of the SVM and applying then the conditional probabilities law we get:

$$p(f_{\Theta_{S_i}}(\mathrm{X})|C, \mathrm{X}, S_i)p(C|\mathrm{X}, S_i) > p(f_{\Theta_{S_i}}(\mathrm{X})|\bar{C}, \mathrm{X}, S_i)p(\bar{C}|\mathrm{X}, S_i). \tag{5.54}$$

It has been proposed by Platt (2000) that one can transform an SVM score into probabilities by plugging it into a sigmoid function of the form:

$$\frac{1}{1 + \exp(-a\, f_{\Theta_{S_i}}(\mathrm{X}) + b)} \tag{5.55}$$

where a and b are parameters to be tuned. Note that one could tune a and b separately for each speaker but we choose to tune them globally, as for the

12, 12, 2005. IDIAP-RR 04-62.

A. Solomonoff, C. Quillen, and W.M. Campbell. Channel compensation for svm speaker recognition. In *Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop*, pages 57–62, 2004.

J. C. Platt. Probabilities for SV machines. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

threshold Δ in (5.2). This allows to have an estimated posterior probability. Using $p(C|\mathrm{X}, S_i) = 1 - p(\bar{C}|\mathrm{X}, S_i)$ we obtain:

$$\frac{p(f_{\Theta_{S_i}}(\mathrm{X})|C, \mathrm{X}, S_i)}{p(f_{\Theta_{S_i}}(\ \mathrm{X}\ )|\bar{C}, \mathrm{X}, S_i)} \exp(af_\Theta(\mathrm{X}) + b) > 1. \tag{5.56}$$

Taking the log, we get:

$$\log \frac{p(f_\Theta(\mathrm{X})|C, \mathrm{X}, S_i)}{p(f_\Theta(\mathrm{X})|\bar{C}, \mathrm{X}, S_i)} + af_\Theta(\mathrm{X}) > -b \approx \Delta\ . \tag{5.57}$$

If we use the same hypothesis than those made for GMMs, we obtain:

$$f_\Theta(\mathrm{X})_{unified-T-norm} = af_\Theta(\mathrm{X}) + \frac{(f_\Theta(\mathrm{X}) - \mu_M)^2}{2\sigma_M^2} > \Delta\ . \tag{5.58}$$

Note that (5.58) is valid only when $f_\Theta(\mathrm{X}) > \mu_M$. A reasonable thing to do is to reject directly without any normalization a claimed speaker if its obtained SVM output is smaller than the average of SVM outputs over a subset of impostors. The consequence of this on the T-norm equation is to force the threshold Δ in (5.58) to be positive.

Experiments

We verified empirically this framework using the GLDS based SVM system described in Chapter 2 on the NIST database. Table 5.2 and Figure 5.4 show the results for SVMs without score normalization, with the naive T-normalization approach given by (5.53) and with the new unified T-norm given by (5.58). The results show that the naive approach degrades the performance significantly for small values of γ of the EPC. The parameter a, here tuned to minimize the EER ($a = 0.2$) on the development set should perhaps be tuned for each value of γ in (3.14). As explained in (Grandvalet et al., 2005), the precision of the probability estimator depends on the cost of each type of errors, $Cost(FN)$ and $Cost(FP)$ in (3.3). Moreover, the solution given by the unified approach correspond to the naive solution when $a = 0$ and corresponds to the SVM without score normalization solution when $a \to \infty$. Anyway, the solution found by the unified T-norm corresponds approximatively to the minimum of the two other systems.

Due to the computational cost of the T-normalization method and the relative small performance improvement, T-normalization will not be used for SVM based systems in the following experiments.

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.

Table 5.2. Results on the NIST test set for the T-norm and unified framework T-norm systems

SVM	No-norm	T-norm Naive	T-norm Unified
HTER [%]	11.06	10.54	9.11
95% Confidence	±1.05	±0.81	±0.85

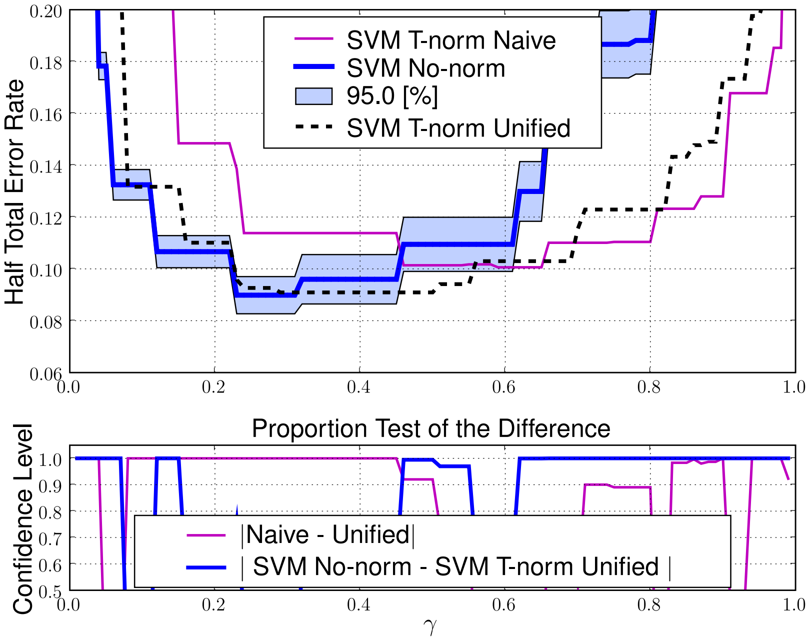


Figure 5.4. EPC curves on the NIST test set for the T-norm and unified framework T-norm systems.

5.4 Conclusion

In this chapter we tried to analyze state-of-the-art models used in speaker verification. As the main purpose of this thesis is to use discriminant models, we defined a general framework to use this kind of models. This framework was originally presented in:

> CONTRIB J. Mariéthoz and S. Bengio. A kernel trick for sequences applied to text-independent speaker verification systems. IDIAP-RR 77, IDIAP, 2005

Before proposing new discriminant models, we first showed that a GMM based system is discriminant and can be interpreted as a mixture of linear classifiers. Several adaptation methods where compared and this comparison was published in:

> CONTRIB J. Mariéthoz and S. Bengio. A comparative study of adaptation methods for speaker verification. In *International Conference on Spoken Language Processing ICSLP*, pages 581–584, Denver, CO, USA, September 2002. IDIAP-RR 01-34

It shows that MAP adaptation is the best one and suggests that it can be the best only because it makes the models more discriminant.

To interpret GMMs as mixtures of experts, we used an algorithm called "synchronous alignment", published in:

> CONTRIB J. Mariéthoz, Dominique Genoud, Frédéric Bimbot, and Chafik Mokbel. Client / world model synchronous alignement for speaker verification. In *6th European Conference on Speech Communication and Technology — Eurospeech'99*, Budapest, Hungary, September 1999

We also used a max approximation of the log likelihood ratio proposed in:

> CONTRIB J. Mariéthoz and S. Bengio. An alternative to silence removal for text-independent speaker verification. IDIAP-RR 51, IDIAP, Martigny, Switzerland, 2003

Finally, score normalization is often used to compensate unmatched conditions between data used to train the model and test accesses. A generalized score normalization framework was proposed. It enlights the hypothesis implicitly done when T- and Z- normalization are used and can be used to develop new normalization procedures. This work was published in:

> CONTRIB J. Mariéthoz and S. Bengio. A unified framework for score normalization techniques applied to text independent speaker verification. *IEEE Signal Processing Letters, Volume 12*, 12, 2005. IDIAP-RR 04-62

This chapter thus provided some tools and intuitions to develop new discriminant approaches either as complementary to GMMs or independently by solving some problem specific to the speaker verification domain such as the use of sequences. The next chapters will be dedicated to the presentation of new discriminant models for speaker verification.

6 GMMs and Discriminant Models

In the previous chapter, we have seen that GMM based systems are discriminant due to some modifications proposed by the speaker verification community in the last ten years. In this chapter we propose to use common discriminant models of the machine learning community such as SVMs. Unfortunately, standard SVMs cannot directly use variable size sequences of acoustic feature vectors. Before addressing this problem, we can use GMMs as a pre-processing for SVMs.

We first propose to replace the Bayes decision function of state-of-the-art GMM based systems, which can be seen as a linear function of two log likelihoods with a fixed slope equal to one, by learning a discriminant decision function with an SVM.

Several other values could be provided to an SVM. First, we propose client and world model scores, respectively the numerator and the denominator of the LLR in (2.18). Secondly, we can enrich this representation with local LLRs for each Gaussian in order to increase the size of the input vector. After analyzing the results, we conclude that having only one discriminant model for all clients seems to be a limitation. We thus propose to use GMM posteriors to enroll a specific discriminant model for each client. The obtained results on the NIST database show that all the proposed approaches are interesting but none increase significantly the performance of the baseline system. The next step, which is treated in Chapter 7, is to train discriminant models on acoustic feature vectors.

The outline of this chapter goes as follows. In Section 6.1, we propose to replace the Bayes decision by learning the decision function with discriminant models. In Section 6.2, we describe how to change the cost function in order to minimize the HTER instead of the usual classification error. Sections 6.3 and 6.4 describe different ways to produce inputs for a client independent SVM. Finally, Section 6.5 proposes a solution to build one discriminant model per client using GMM Gaussian posteriors.

6.1 Learning the Decision Function

While most state-of-the-art methods for speaker verification are based on non-discriminant models (such as HMMs or GMMs), a better solution should be in theory to use a discriminant framework, see (Vapnik, 2000) for a discussion on discriminant versus non-discriminant models.

A simple way to add some discriminant power to these generative models is to use discriminant decision rules. In our case the generative models are GMMs and the standard decision function, as given in (5.9), can be written as:

$$\frac{p(\mathbf{X}|S_i,C)}{p(\mathbf{X}|S_i,\bar{C})} > \frac{P(\bar{C}|S_i)}{P(C|S_i)} \approx \Delta \tag{6.1}$$

where $\mathbf{X}$ is a sentence pronounced by a client C or an impostor $\bar{C}$ given the claimed identity S_i.

It can be rewritten as follows:

$$y = \log p(X|S_i,C) - \log p(X|S_i,\bar{C}) - \Delta \tag{6.2}$$

such that the sign of y gives the decision. The goal can thus be to find a value of Δ that optimizes a given criterion over the decision. If the probabilities are perfectly estimated, which is usually not the case, then the Bayes decision is optimal and Δ should be near the log ratio of priors.

In this chapter, we are interested in the case where the probabilities are not perfectly estimated and where the Bayes decision might not be the optimal solution. We thus propose to explore other forms of decisions, based either on linear functions or on more complex functions such as SVMs. In this case we can generalize (6.1) using:

$$f_{\Theta_{S_i}}(g(\mathbf{X})) > \Delta \tag{6.3}$$

where $g(\mathbf{X})$ is a vector of features extracted from the models of $p(X|S_i,C)$ and $p(X|S_i,\bar{C})$.

If the decision function is common to all clients, then it becomes:

$$f_{\Theta}(g(\mathbf{X})) > \Delta\,. \tag{6.4}$$

In the following, we propose to further enhance the decision function given by (6.3) or (6.4) using more powerful models, such as the SVMs. Different $g(\mathbf{X})$ features are studied.

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

To measure the performance of these several approaches, the experiments are performed on the NIST database, using the development set to tune the hyper-parameters and the test set to measure the performance. Each system is compared to a GMM based system. The T-normalization should be used, but as the T-normalization is applied to the LLR and as we do not use directly the LLR as SVMs inputs, it does not make sense to use it for these SVM based approaches, and we will not use it neither for the baseline. On the other hand, standard score normalization approaches can be adapted specifically for the new proposed approaches and can be a part of further research to improve such models.

6.2 HTER Cost Function

In classical speaker verification systems, when no prior information is given on the cost of the different kinds of errors, the Bayes decision rule is applied by selecting the value of Δ in (6.2) that minimizes the HTER.

Note that this cost function changes the relative weight of client and impostor accesses in order to give them equal weight, instead of the one induced by the training data.

It is important to note that the training criterion used in SVMs is related to the number of classification errors. In order to optimize the HTER cost (3.4), the relative weight of each example (Lin et al., 2002) in the normal SVM formulation is changed. The cost function (2.9) is modified by splitting the C parameter as follows:

$$(\mathbf{w}^*, b^*) = \arg\min_{(\mathbf{w},b)} \frac{\| \mathbf{w} \|^2}{2} + \sum_{l=1}^{L} C_l |1 - y_l(\mathbf{w}\phi(\mathbf{x}_l) + b)|_+ \tag{6.5}$$

where $C_l = \begin{cases} C+ & \text{when } y_l > 0 \\ C- & \text{otherwise} \end{cases}$ where $C+$ is the trade-off parameter for the positive examples and $C-$ the trade-off parameter for the negative examples. For the NIST database, we have NN $=$ 10 NP; the number of negative examples are ten times more than the number of positive examples, then $C- = 10\,C+$.

6.3 GMM LLR

Let us now consider the simplest SVM model, using the two log probabilities as inputs. The resulting function is given by (6.4) where $g(\mathbf{X})$ would be a two-

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in non-standard situations. *Machine Learning*, 46:191–202, 2002.

dimensional vector containing $\log p(\mathbf{X}|S_i, C)$ and $\log P(\mathbf{X}|S_i, \bar{C})$, the client and world model scores.

Figure 6.1, originally published in (Bengio and Mariéthoz, 2001) on the Polyvar database, shows the decision function found for Bayes, a linear SVM and an RBF SVM. Each green point represents an impostor access and each red point represents a client access. Each line represents a specific decision function where all points above the line are rejected and all points below the line are accepted. We can observe visually that the decision function seems to be linear even for the RBF based SVM. Note that the slope of the Bayes decision is fixed to one by definition.

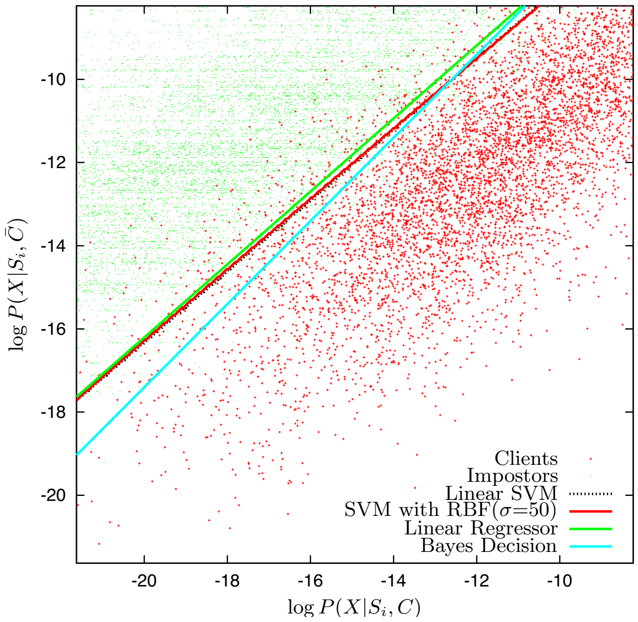


Figure 6.1. Different models to separate clients and impostors, in a text independent task on the Polyvar database.

Figure 6.2 shows the results on the test set of NIST database comparing a GMM based system with an SVM using a linear kernel. We can see that both systems are similar. Instead of using a linear kernel, we can use an RBF kernel; in Figure 6.3, we see that it does not help.

6.4 GMM Gaussian LLR

As we have seen in Chapter 5, the GMM decision function can be interpreted as a mixture of linear classifiers under some hypotheses. (5.23) expressed the

S. Bengio and J. Mariéthoz. Learning the decision function for speaker verification. In *IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP*, Salt Lake, City, USA, 2001. IDIAP-RR 00-40.

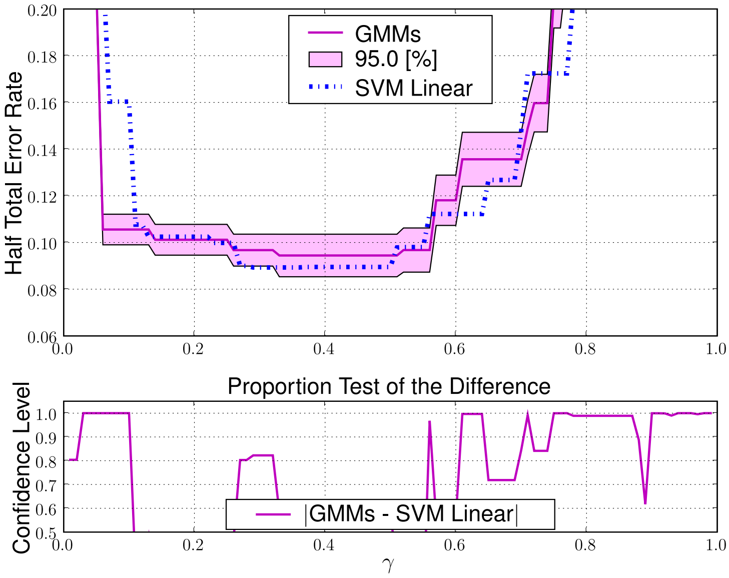


Figure 6.2. Results on the test set of the NIST database: GMMs vs linear SVM on LLR ($C+ = 3$ and $C- = 30$).

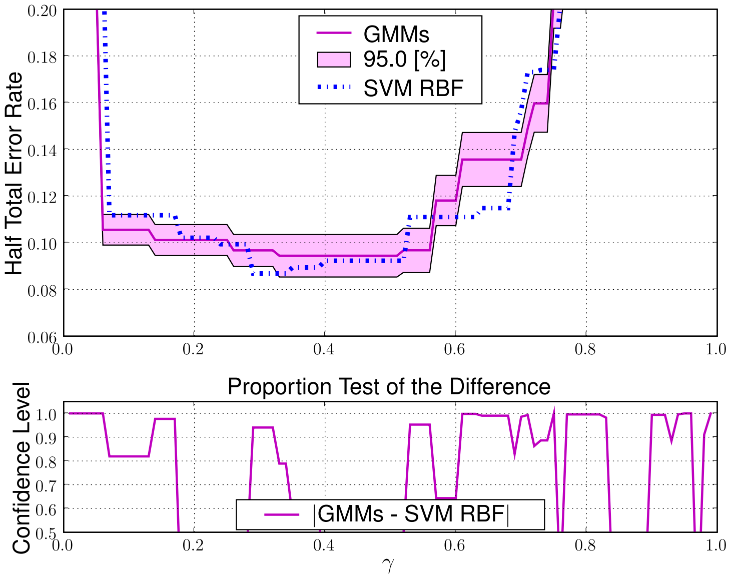


Figure 6.3. Results on the test set of the NIST database: GMMs vs RBF SVM on LLR ($C+ = 3$, $C- = 30$ and $\sigma = 5$)

LLR as a sum of local LLRs over all Gaussians of GMMs. These values can be used to increase the number of SVM inputs by taking the average of the LLRs over all frames for each Gaussian, given by:

$$\text{llr}_s(g) = \frac{1}{T(g)} \sum_{t=1}^{T(g)} \log \frac{p(\mathbf{x}_{r_g(t)}|S_i, C, g)}{p(\mathbf{x}_{r_g(t)}|S_i, \bar{C}, g)} \tag{6.6}$$

where $T(g)$ is the number of frames assigned to the couple of Gaussians g, and $r_g(t)$ returns the index of the t^{th} frame assigned to the cluster g.

We obtain for each sequence a fixed sized vector of size equal to the number of Gaussians in the GMM. This vector corresponds to $g(\mathbf{X})$ in (6.4) and can be used as input to an SVM classifier.

In Figure 6.4, we can see that most of the time the new SVM system is similar to or even worse than the GMM based system.

Note, however, that this approach gave good results for the task of removing silence frames automatically without using a silence/speech detector, see (Mariéthoz and Bengio, 2003) for more details.

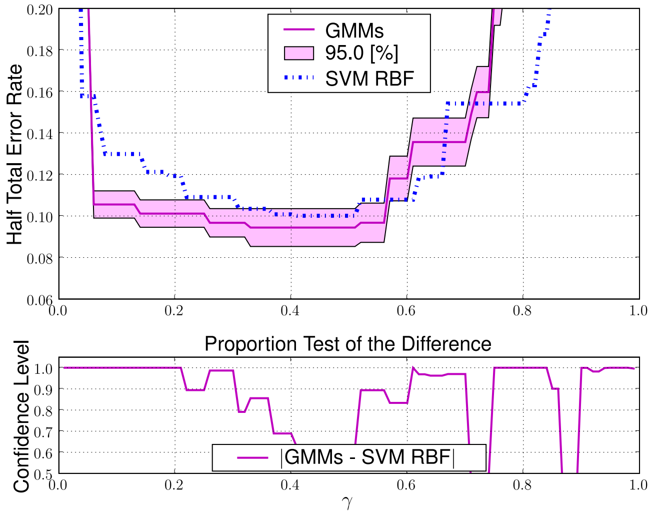


Figure 6.4. Results on the test set of the NIST database: GMM vs RBF SVM on Gaussian LLR ($C+ = 20, C- = 200$ and $\sigma = 800$).

Having only one discriminant model for all speakers seems to be a limitation for the use of discriminant models for speaker verification. Let us now explore a solution where a specific discriminant model is trained for each speaker.

J. Mariéthoz and S. Bengio. An alternative to silence removal for text-independent speaker verification. IDIAP-RR 51, IDIAP, Martigny, Switzerland, 2003.

6.5 Posterior Based Approach

The main idea here is to use some information from already trained GMMs in order to learn a discriminant model for each client. Unfortunately, we cannot use the LLR scores directly because only very few client accesses are available: only one for the NIST database for example. Indeed, we need at least one example to train the SVM model. When only one access is available for training and since this example was normally already used to enroll the client GMM model, the LLR of this particular access is optimistically biased. This also explains why it is not possible to learn a decision threshold for each client. We thus need to use client independent GMM parameters. A solution consists in using the posterior probability of each Gaussian from a generic GMM model.

Consider the average over all frames of the posterior probability of each Gaussian of a generic GMM model (the world model in a GMM based system for example):

$$P(g|\mathbf{X}) = \frac{P(g|\Theta)p(\mathbf{X}|g,\Theta)}{p(\mathbf{X}|\Theta)}$$

where g is the Gaussian index and Θ is the set of parameters. Using a GMM as estimator with $\Theta = \{w_g, \boldsymbol{\mu}_g, \boldsymbol{\sigma}_g\}_{g=1}^{N_g}$, we obtain:

$$P(g|\mathbf{X}) \approx \sum_{t=1}^{T} \log \frac{w_g \, \frac{1}{\sqrt{2\pi\,\boldsymbol{\sigma}_g^2}} \, \exp - \frac{(\mathbf{x}_t - \boldsymbol{\mu}_g)^2}{2\,\boldsymbol{\sigma}_g^2}}{\sum_{j=1}^{N_g} w_j \, \frac{1}{\sqrt{2\pi\,\boldsymbol{\sigma}_j^2}} \, \exp - \frac{(\mathbf{x}_t - \boldsymbol{\mu}_j)^2}{2\,\boldsymbol{\sigma}_j^2}}$$

where $\mathbf{x}_t$ is the t^{th} frame of the sequence $\mathbf{X}$.

Normalizing by the length of the sequence as commonly done in that domain and as explained in Chapter 5, we finally obtain:

$$P_{norm}(g|\mathbf{X}) = \frac{1}{T} \sum_{t} \log \frac{w_g \, \frac{1}{\sqrt{2\pi\,\boldsymbol{\sigma}_g^2}} \, \exp - \frac{(\mathbf{x}_t - \boldsymbol{\mu}_g)^2}{2\,\boldsymbol{\sigma}_g^2}}{\sum_{j=1}^{N_g} w_j \, \frac{1}{\sqrt{2\pi\,\boldsymbol{\sigma}_j^2}} \, \exp - \frac{(\mathbf{x}_t - \boldsymbol{\mu}_j)^2}{2\,\boldsymbol{\sigma}_j^2}}.$$

All N_g values of $P_{norm}(g|\mathbf{X})$ are concatenated in order to have a vector of size number of Gaussians. This is similar to the Fisher score based approach proposed by Jaakkola and Haussler (1998), which consists in computing the derivative of the log likelihood of a generative model with respect to its parameters and use it as inputs to an SVM. In our case it corresponds to the Fisher score based approach by taking only the GMM weights as parameters. Using the mean and the variance parameters of the GMM makes the system impractical to train (a typical GMM has around 10^4 to 10^5 parameters).

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

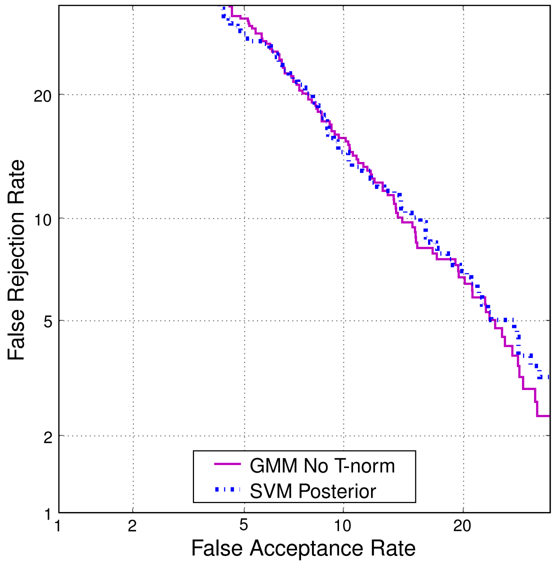


Figure 6.5. Results on the development set of the NIST database: GMM without T-norm vs SVM trained on posteriors. ($C+ = C- \to \infty$ and $N_g = 1000$).

Figure 6.5 shows a DET curve on the development set of the NIST database for a GMM based system without score normalization and an SVM using as inputs the posteriors of a generic GMM composed of 1000 Gaussians learned using the world population. We can see that the two systems appear similar. Unfortunately, these results are not confirmed on the test set as shown in Figure 6.6. The SVM Posterior based system is statistically significantly worse than the GMM T-norm system. We obtained the same kind of results on preliminary experiments over an old version of the NIST database: it yielded good results for the female population and poor results on the male population. In fact, in order to obtain good results, we need a rich generic model. Rich in the sense that we need a lot of Gaussians, but also a large diversity in terms of recording conditions and number of speakers. Probably, the world population is not enough representative of the test set. The development set comes from the same previous NIST campaigns as the world population, which may explain the good obtained performance. This is unfortunately not the case for the test set population.

Given that the posterior probability values represent more something like the phonetic content rather than the way a specific speaker pronounces a sentence, the obtained results are surprisingly good. Since this model produces phonetic information, it can be interesting to perform fusion with LLRs pro-

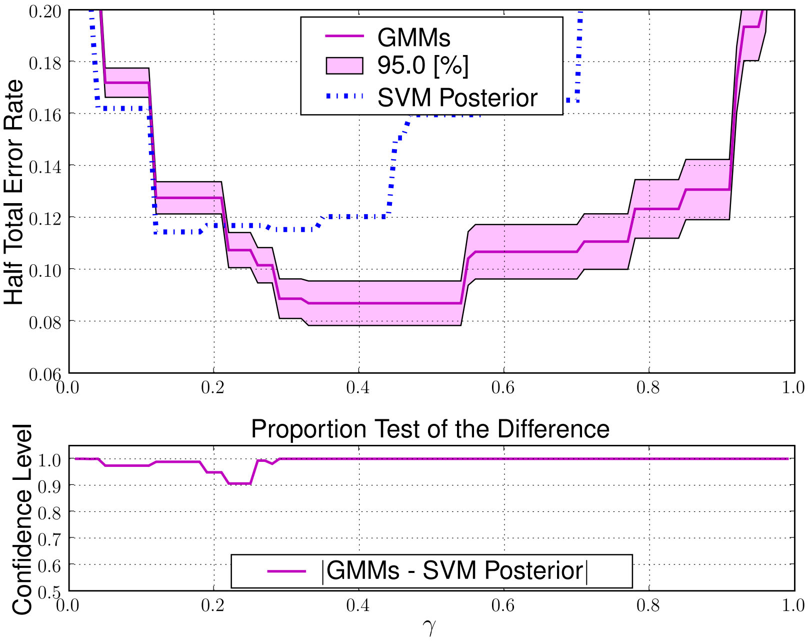


Figure 6.6. Results on the test set of the NIST database: GMM vs SVM Posterior. ($C+ = C- = \infty$ and $N_g = 1000$).

duced by a GMM based system, or with an SVM based system by appending the obtained vector to the explicit polynomial expansion of the GLDS kernel.

6.6 Conclusion

In this chapter, we proposed a few simple approaches to use discriminant models with GMM based speaker verification systems. The new approaches do not improve the performance over the baseline system. In fact, the GMM Gaussian posterior based systems need further research in order to become really efficient and similar approaches used in object recognition should be considered (Jurie, 2005).

Learning the decision function suggests that the discriminant models should be client dependent. This work was published in:

B. Jurie, F. and Triggs. Creating efficient codebooks for visual recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 604– –610, 17 October 2005.

CONTRIB S. Bengio and J. Mariéthoz. Learning the decision function for speaker verification. In *IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP*, Salt Lake, City, USA, 2001. IDIAP-RR 00-40

The use of discriminant models as a decision function and using a large vector of LLRs was proposed in:

CONTRIB J. Mariéthoz and S. Bengio. An alternative to silence removal for text-independent speaker verification. IDIAP-RR 51, IDIAP, Martigny, Switzerland, 2003

In this chapter, we studied the use of discriminant models using informations from already trained client GMM models. In the next chapter we will focus on discriminant models using directly acoustic feature vectors as input. This avoids first training a generative model and makes the entire system discriminant. More specifically we address the problem of sequences for SVMs.

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

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.

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.

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 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).

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

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

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:

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

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

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)

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

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

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.

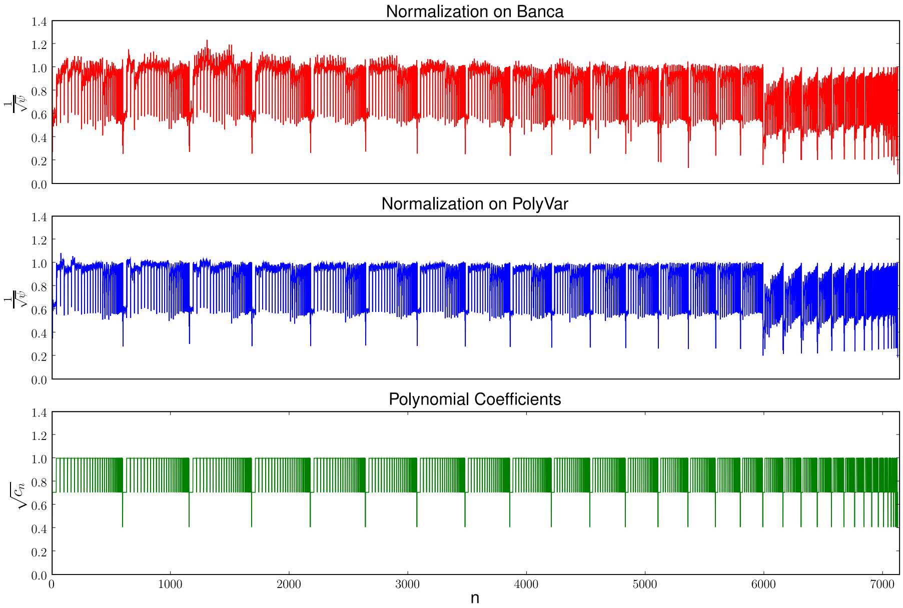


Figure 7.1. Coefficient values $\frac{1}{\sqrt{\psi_n}}$ of polynomial terms in the GLDS kernel, as computed on Banca and Polyvar, compared to the $\sqrt{c_n}$ polynomial coefficients of equation (7.2).

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).

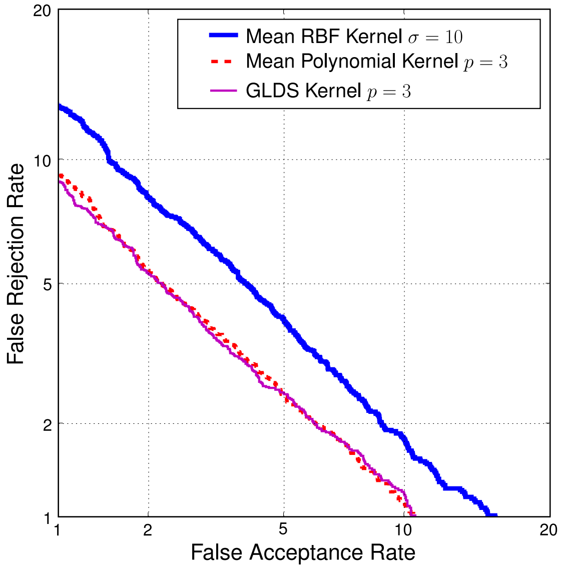


Figure 7.2. DET curves on the development set of the Polyvar database comparing the explicit polynomial expansion (GLDS based kernel), the principled polynomial kernel and an RBF kernel (using the Mean operator).

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

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 $K(\mathbf{X}_i, \mathbf{X}_j) \geq K(\mathbf{X}_i, \mathbf{X}_i)$.

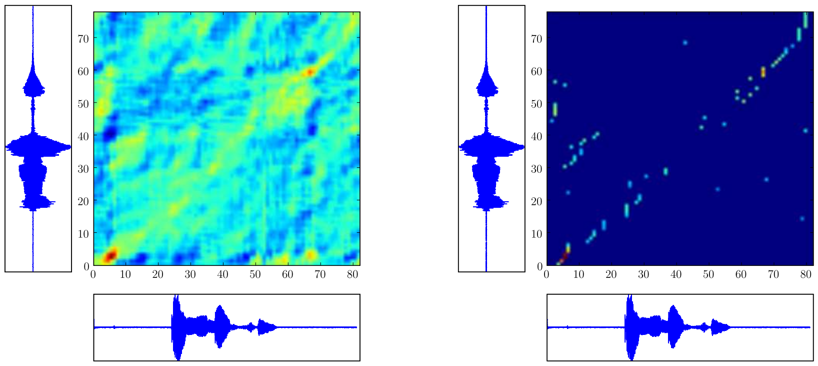
We thus propose here an alternative to taking the average over all frames. We consider, for each frame of sequence $\mathbf{X}_i$, the similarity measure of the closest corresponding frame in sequence $\mathbf{X}_j$. We thus propose to take a symmetric Max operator of the form:

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

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.

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



(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 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.

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

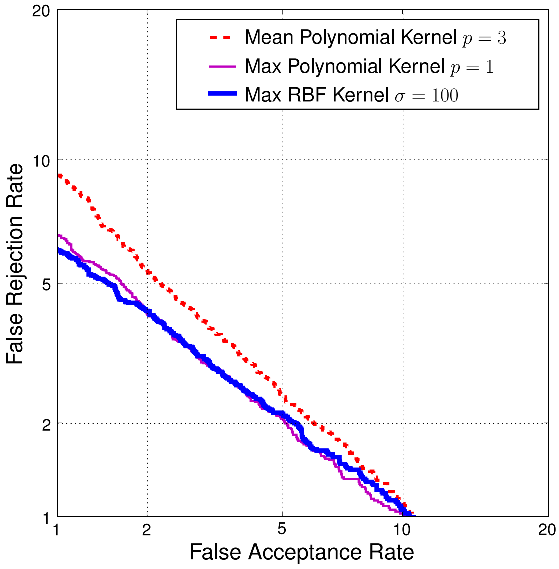


Figure 7.4. DET curves on the development set of the Polyvar database for Mean and Max operators for polynomial and RBF kernels.

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

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

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

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.

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.

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

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.

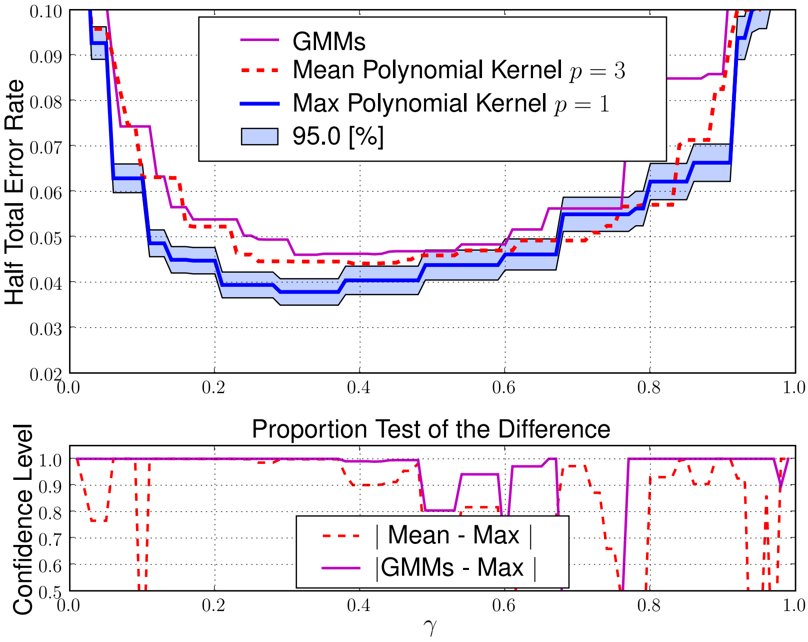


Figure 7.5. EPC curves on the test set of the Polyvar database for best Mean and Max operators for polynomial and RBF kernels.

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.

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:

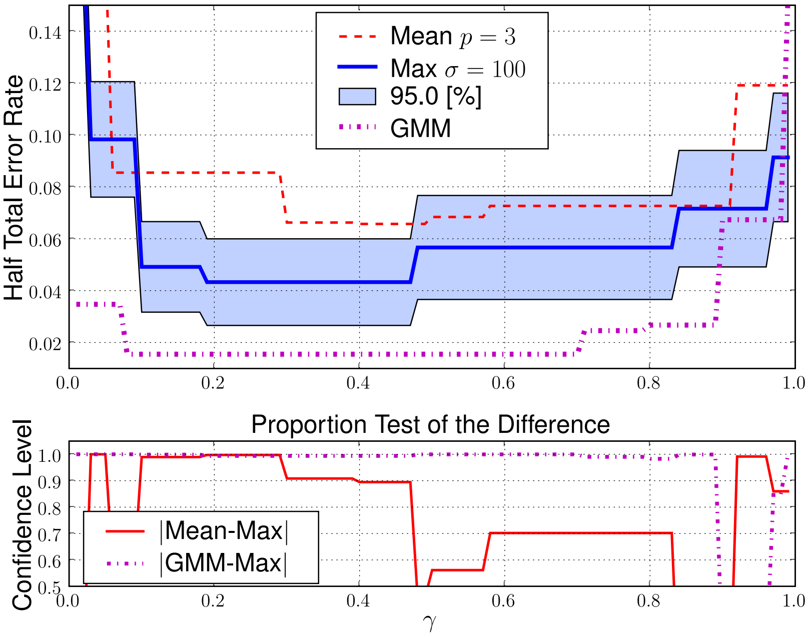


Figure 7.6. EPC curves on test set of the Banca database for best Mean and Max operator for polynomial and RBF kernels.

“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

[9] Y. Grandvalet, J. Mariéthoz, and S. Bengio. A probabilistic interpretation of svms with an application to unbalanced classification. In *Advances in Neural Information Processing*

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).

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.

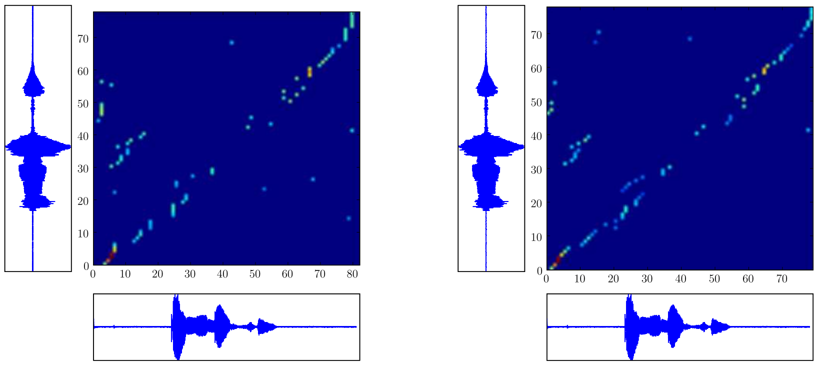
Figure 7.8 and Table 7.5 show the results of the new smoothing kernel compared to the Mean and Max operator kernels. The new smoothing kernel outperforms statistically significantly the Mean operator kernel for all values of γ and outperforms statistically significantly the Max operator kernel for some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

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:

Systems, NIPS 15, 2005. IDIAP-RR 05-26.

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.



(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.

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

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

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

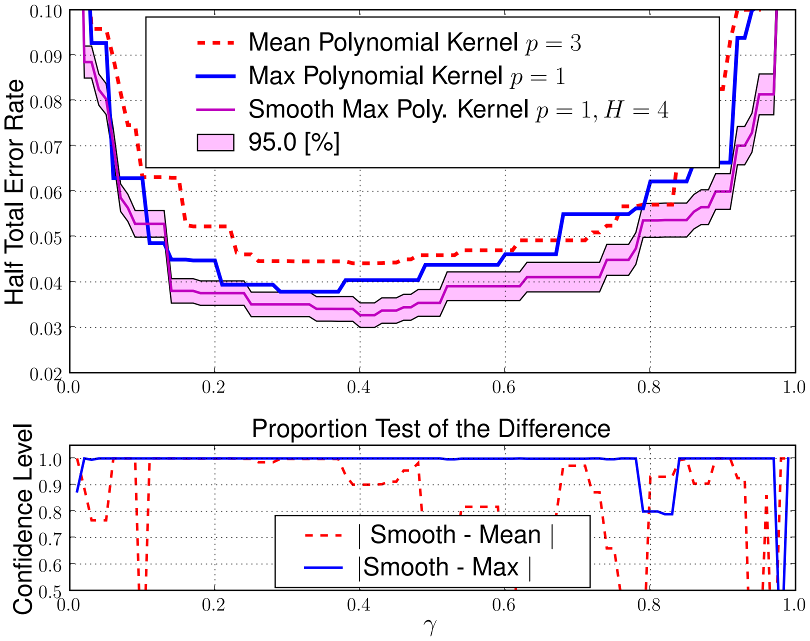


Figure 7.8. EPC curves on the test set of the Polyvar database for best Mean, Max and smooth Max operators for polynomial kernels.

constraint is too strong.

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.

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-

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.

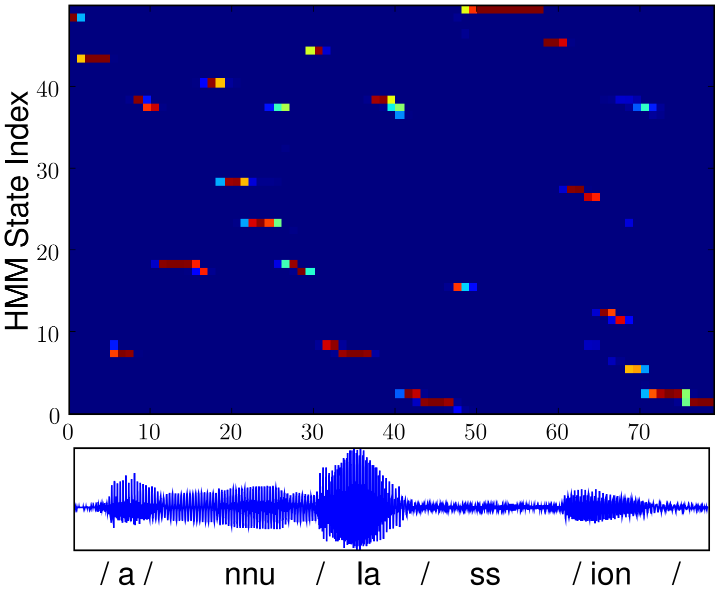


Figure 7.9. Posterior values for access “f4425w14” of Polyvar database.

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:

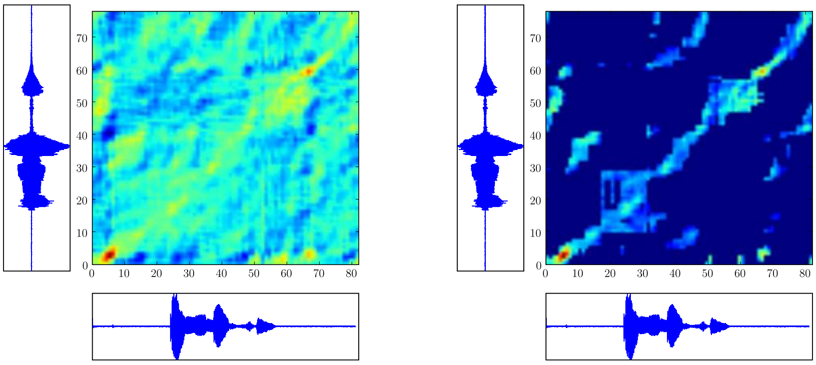
$$\{t\}^* = \arg \underset{\{t\}_1^{N_b}}{\text{nbest}}\ p(q_t = s^*(t_i)|\mathbf{X}_j)$$

where $\text{nbest}_{\{t\}_1^{N_b}}$ 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.

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



(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.

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.

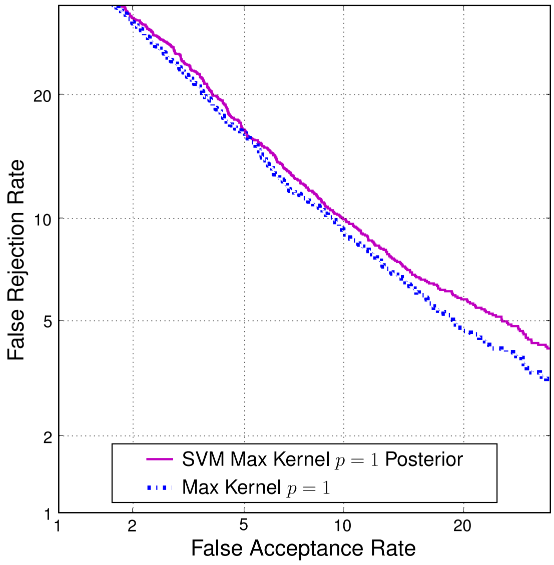


Figure 7.11. Results on the development set of a previous version of the NIST database: Max operator based kernel $p = 1$ with and without posterior based approximation.

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 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

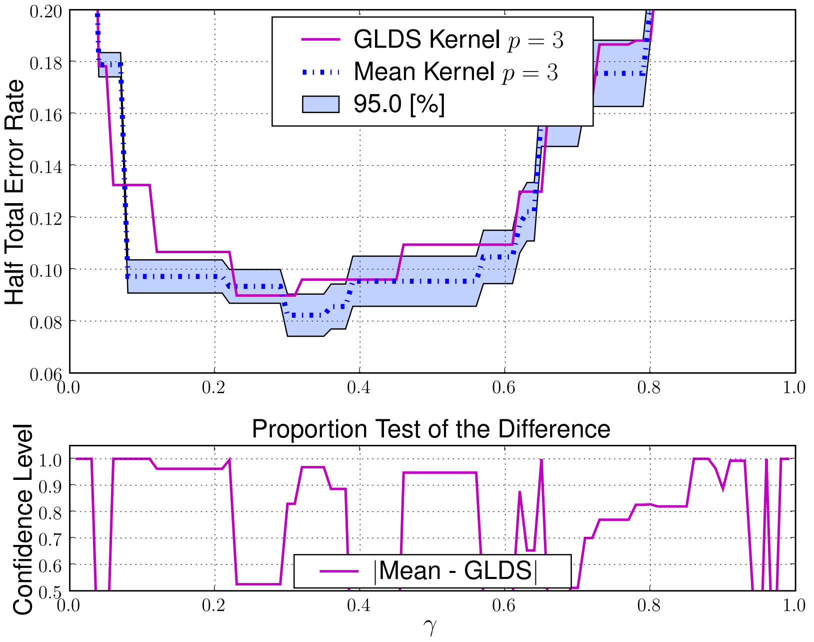


Figure 7.12. Results on the test set of the NIST database: GLDS Kernel $p = 3$ vs Mean operator Kernel $p = 3$.

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.

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.	±0.84	±1.05	±1.03	±1.04	±0.72
# SV	-	38	40	110	33

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

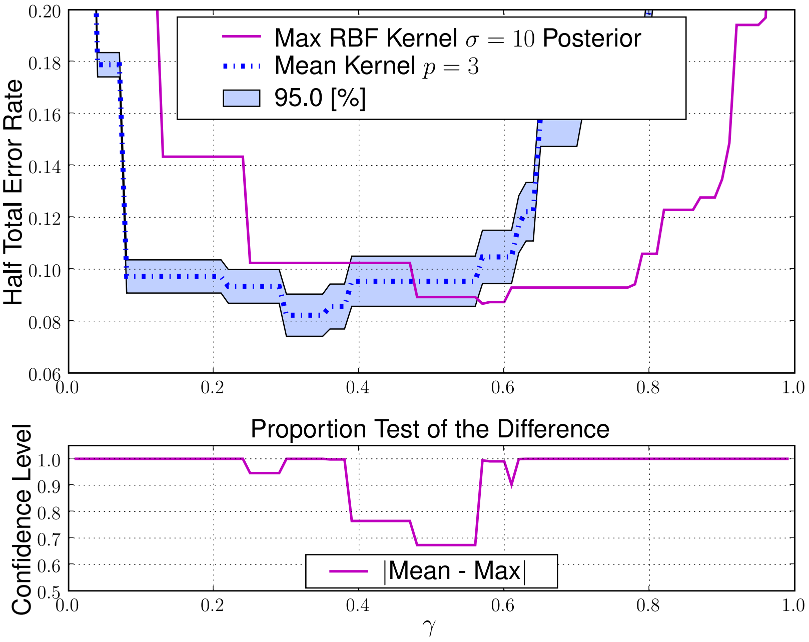


Figure 7.13. Results on the test set of the NIST database: Max operator RBF Kernel $\sigma = 10$ using posterior based approximation and Mean operator Kernel $p = 3$.

costs used by the NIST campaign: $\gamma \approx 0.909$. It is also interesting to note that in that case the RBF kernel outperforms the polynomial kernel.

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:

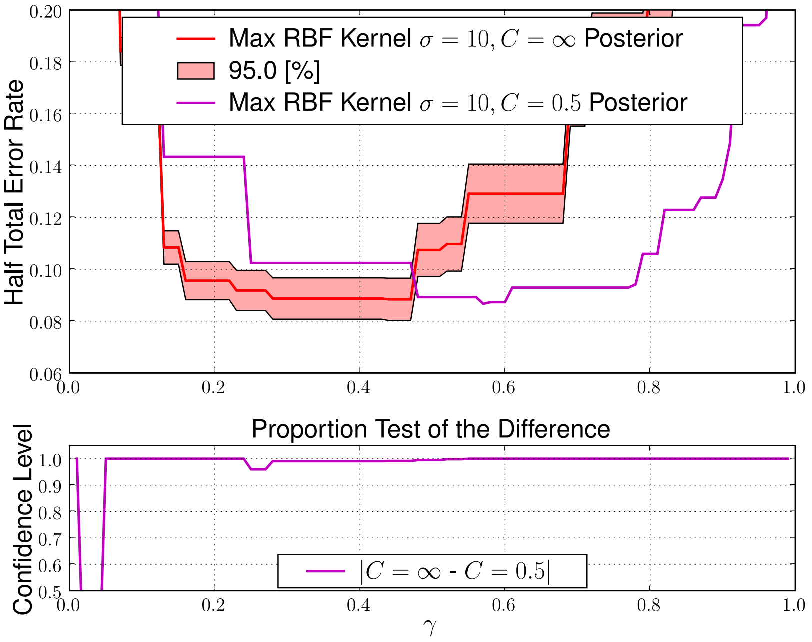


Figure 7.14. Results on the test set of the NIST database: Max operator RBF Kernel $\sigma = 10$ using posterior based approximation for two different values of C: ∞ and 0.5.

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.

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.

8 A New Perspective: Working on the Distance Measure

Speaker verification is a highly unbalanced two-class classification problem and it might be important to consider specific training criteria for such cases. Gradient based models (such as Multilayer Perceptrons) can easily accommodate various possible training criteria adapted to unbalanced datasets, and thus can be good candidates to solve this problem. Unfortunately, when using a large margin approach, the number of training iterations needed to converge to a good solution is huge. This has also been observed in Collobert and Bengio (2004). SVMs have usually faster convergence rates, so we will instead consider unbalanced criteria for SVMs.

After analyzing two already proposed criteria for this problem (Lin et al., 2002) and (Grandvalet et al., 2005), we note that they are useless in our case. Indeed, empirically we observed that for all SVM based sequence kernels that give reasonable performance, and for all client models, the problem is in fact linearly separable in the feature space and we can show that for such a problem these unbalanced criteria have no effect. Moreover, in the separable case the standard SVM solution is good because only examples in the margin are considered.

At the opposite, another specific speaker verification problem, which for us is more important, is addressed here: the intra-impostor distance distribution is different than the intra-client distance distribution. We thus propose to modify the SVM kernel by assuming a Gaussian noise on negative examples. Starting from a principled approach, and after some empirical modification, we show

R. Collobert and S. Bengio. Links between perceptrons, MLPs and SVMs. In *International Conference on Machine Learning, ICML*, 2004.

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. *Machine Learning*, 46:191–202, 2002.

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.

that the new system outperforms the baseline system.

The outline of this chapter goes as follows. In Section 8.1, we present the known unbalanced class criteria for SVMs and show they are useless for separable problems. Section 8.2 is dedicated to a new similarity measure that takes into account the difference between the intra-impostor and intra-client distance distributions.

8.1 Unbalanced SVM Criteria

SVMs are known to perform well in terms of misclassification error, but they also have been recognized to provide skewed decision boundaries for unbalanced classification losses, where the losses associated with incorrect decisions differ according to the true label. The mainstream approach used to address this problem was proposed in (Lin et al., 2002) and consists in using different costs for positive and negative examples using two smoothing parameters C_+, C_- instead of a single C as in (2.9). This solution was used, for instance, in Chapter 6 and is given in (6.5).

Another solution, proposed in Grandvalet et al. (2005) is based on a probabilistic interpretation of SVMs. The cost to optimize now becomes:

$$\begin{aligned}\arg\min_{(\mathbf{w},b)} \frac{\| \mathbf{w} \|^2}{2} \quad + \quad & C\left(\sum_{\{i|y_i=1\}} [-\log(P_0) - (1-P_0)(f(\mathbf{x}_i)+b)]_+ \right. \\ & \left. + \sum_{\{i|y_i=-1\}} [-\log(1-P_0) + P_0(f(\mathbf{x}_i)+b)]_+ \right)\end{aligned} \tag{8.1}$$

where $P_0 = \frac{C(\text{FP})}{C(\text{FP})+C(\text{FN})}$, $C(\text{FP})$ is the cost of a false positive and $C(\text{FN})$ is the cost of a false negative.

Even if these two approaches give good results on standard machine learning databases, as shown in (Grandvalet et al., 2005), they have no positive effect in our case. Indeed, empirically we can observe that for all sequence kernels that provide good performance, the problem is separable: all the training examples are well classified. It seems reasonable: the feature space dimension is greater than the number of training examples. Moreover most of the time the optimal value for C tends to ∞ and thus the criterion does not tolerate any error. This is probably because it cannot make an error on positive examples: they

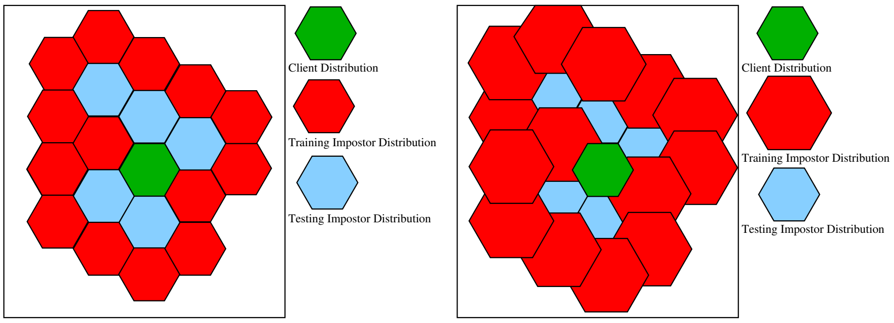
Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in non-standard situations. *Machine Learning*, 46:191–202, 2002.

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.

are too few; and it can neither tolerate an error on a negative example: the coverage of the training negative examples is not good enough. Indeed, each negative example can cover its own variability but cannot cover the future testing negative examples (other impostors). As the training positive and negative examples do not correspond well enough to the test set, it can be interesting to use prior knowledge in the kernel: for instance we expect the variance of the intra-impostor distance distribution to be larger than that of the intra-client distance distribution.

8.2 Class Dependent RBF Kernel

When a two-class classification problem is separable, we can admit that a solution maximizing the margin is a good idea even if the problem is unbalanced. Indeed an SVM considers only examples in the margin and ignores other examples. Hence, the standard SVM criterion can be good also for separable unbalanced class problems. It still remains that, in the case of speaker verification, the distribution of the distance between two impostor accesses is larger than the client distance distribution: impostors are individual speakers and thus the intra-impostor distribution is more similar to the inter-class distance distribution than the intra-client distribution. In this case, it can be a good idea to change the kernel in order to make the negative examples closer. In other words, a negative example should cover its own variability (same speaker), but also unseen negative examples (other impostors).


Client Distribution
Training Impostor Distribution
Testing Impostor Distribution
Client Distribution
Training Impostor Distribution
Testing Impostor Distribution

(a) Normal (b) Enlarged

Figure 8.1. Client, training and testing impostor distributions.

Figure 8.1 shows that enlarging the negative example distribution, for instance by using a larger σ value for intra-negative RBF kernel evaluation, increases the coverage of the unseen impostor examples.

Vapnik (2000) proposed the use of vicinal risk minimization to learn a

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

decision function over distributions instead of points. One of several solutions he proposed is the soft vicinity function that uses a kernel over distributions. The main idea is to assume a Gaussian noise over each negative example. Using an RBF kernel with a Gaussian noise distribution, we have:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2\pi\sigma_i\sigma_j} \int\int \exp\left\{-\frac{(\mathbf{x}-\mathbf{x}')^2}{2\sigma^2} - \frac{(\mathbf{x}'-\mathbf{x}_i)^2}{2\sigma_i^2} - \frac{(\mathbf{x}-\mathbf{x}_j)^2}{2\sigma_j^2}\right\} d\mathbf{x}\, d\mathbf{x}' \tag{8.2}$$

where σ is the RBF kernel hyper-parameter, σ_i the noise standard deviation of example $\mathbf{x}_i$ and σ_j the noise standard deviation of example $\mathbf{x}_j$.

Vapnik (2000) then showed that (8.2) can be rewritten as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \frac{\sigma_i^2}{\sigma^2} + \frac{\sigma_j^2}{\sigma^2}\right)^{(-\frac{d}{2})} \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x}_j)}{2(\sigma^2 + \sigma_i^2 + \sigma_j^2)}\right\} \tag{8.3}$$

where d is the dimension of the input vector.

Let us now consider a Gaussian noise for the negative examples only, with variance $\tau\sigma^2$ where τ is a constant to tune, we obtain:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \exp -\frac{(\mathbf{x}_i-\mathbf{x}_j)^2}{2\,\sigma^2} & \text{if } y_i = y_j = 1 \\ (1+\tau)^{(-\frac{d}{2})} \exp -\frac{(\mathbf{x}_i-\mathbf{x}_j)^2}{2\,\sigma^2(1+\tau)} & \text{if } y_i \neq y_j \\ (1+2\tau)^{(-\frac{d}{2})} \exp -\frac{(\mathbf{x}_i-\mathbf{x}_j)^2}{2\,\sigma^2(1+2\,\tau)} & \text{if } y_i = y_j = -1. \end{cases} \tag{8.4}$$

In (8.4) we have a kind of RBF kernel with larger standard deviation if $y_i = y_j = -1$ than otherwise. This is what we expected: make the intra-negative distance smaller. Unfortunately, the constant $(1+2\tau)^{(-\frac{d}{2})}$ has the inverse effect and decreases faster that the exponential term. Moreover Vapnik (2000) said nothing about how to choose σ for a new test point (for which the class is obviously not known).

Even if this is not principled, we would like to propose some simplifications to Vapnik's approach, as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \exp -\frac{(\mathbf{x}_i-\mathbf{x}_j)^2}{\sigma_{++}^2} & \text{if } y_i = y_j = 1 \\ \exp -\frac{(\mathbf{x}_i-\mathbf{x}_j)^2}{\sigma_{+-}^2} & \text{if } y_i \neq y_j \\ \exp -\frac{(\mathbf{x}_i-\mathbf{x}_j)^2}{\sigma_{--}^2} & \text{if } y_i = y_j = -1 \end{cases} \tag{8.5}$$

with

$$\sigma_{++} = \sigma + - = \sigma_+ \tag{8.6}$$

$$\sigma_{--} = \sigma_- \tag{8.7}$$

$$\sigma_- > \sigma_+ \tag{8.8}$$

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

where σ_- and σ_+ are hyper-parameters to tune. The differences between (8.4) and (8.5) are that we remove the constants involving the dimension of the data d, and choose the same value for σ_{++} and σ_{+-}; in fact when we have only one positive example to train the model, any value for σ_{++} yields the same kernel value (equal to one). During test, we tried empirically several values of σ between σ_+ and σ_- and found that the best value is σ_+ for both Banca and Polyvar databases.

Figure 8.2 shows that the vicinity based method outperforms the Max operator based RBF kernel on the development set of the Polyvar database. This is also confirmed on the test set on Figure 8.3 and Table 8.1. The two models are statistically significantly different for most value of γ.

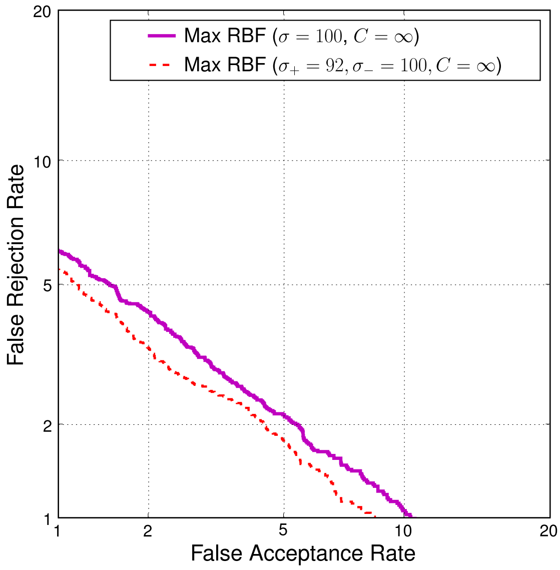


Figure 8.2. DET curves on the development set of the Polyvar database for the best σ, σ_+, σ_- Max RBF kernel.

Table 8.1. Results on the test set of the Polyvar database for Mean and Max operators for polynomial and RBF σ and σ_+, σ_- kernels.

	GMM $N_g = 100$	Mean $p = 3$ $C = \infty$	Max $\sigma = 100$ $C = \infty$	Max $\sigma_+ = 92$ $\sigma_- = 100$ $C = \infty$
HTER [%]	4.9	4.47	4.21	3.59
95% Confidence	±0.34	±0.32	±0.28	±0.32
# Support Vectors	-	87	99	76

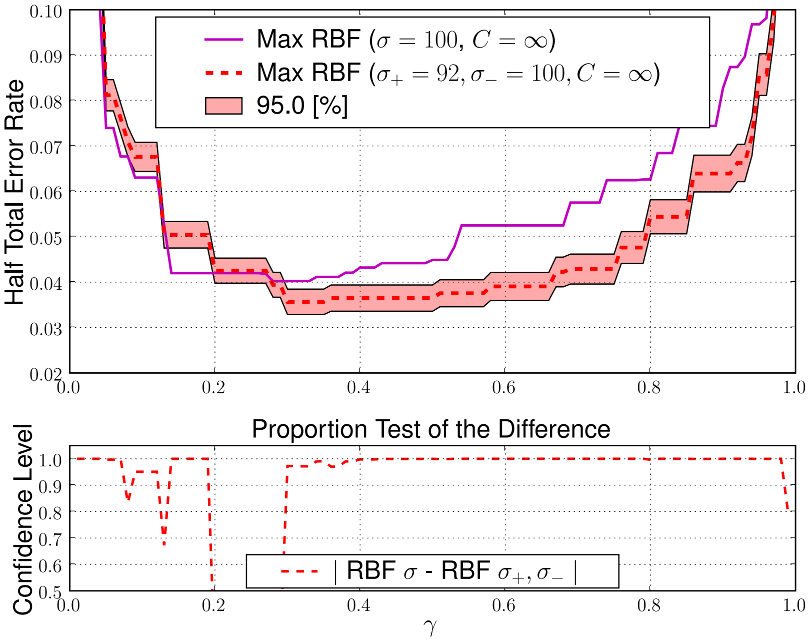


Figure 8.3. EPC curves on the test set of the Polyvar database for the best σ, σ_+, σ_- Max RBF kernel.

We also performed the same experiments on the Banca database and draw the same conclusion as shown in Figure 8.4, Figure 8.5 and Table 8.2. Even if on this database the results are not statistically significantly different due to the size of this database, the effect seems positive. Note also that, for this database, we are still far from the GMM based system, on the test set as seen in Table 8.2 and Figure 8.5 but it seems not be the case on the development set as seen in Figure 8.4.

Table 8.2. Results on the test set of the Banca database for Mean and Max operators for polynomial and RBF σ and σ_+, σ_- kernels.

	GMM $N_g = 200$	Mean $p = 3$ $C = \infty$	Max $\sigma = 225$ $C = 130$	Max $\sigma_+ = 125$ $\sigma_- = 225$ $C = 130$
HTER [%]	2.72	6.57	4.7	4.11
95% Confidence	±1.42	±2.1	±1.78	±1.66
# Support Vectors	-	27	17	13

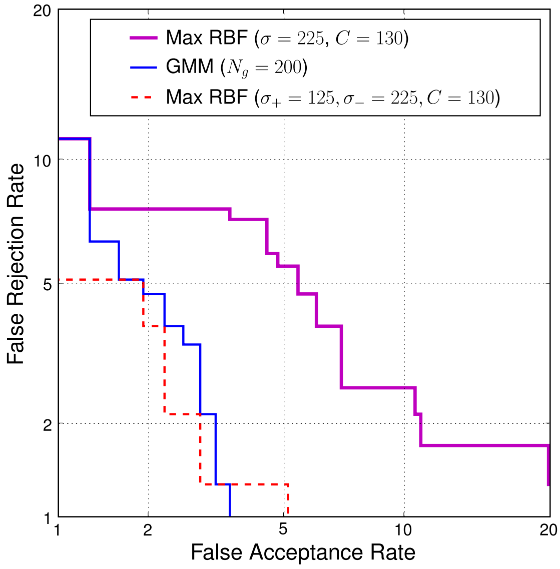


Figure 8.4. DET curves on the development set of the Banca database for the best σ, σ_+, σ_- Max RBF kernel and GMM based system.

8.3 Conclusion

In this chapter, we considered the unbalanced class problem underlying the speaker verification task. We tried to use modified criteria for SVM in order to deal with unbalanced datasets and observed that they have no effect on separable problems, which is the case for our speaker verification experiments. Indeed, we enlight the fact that for separable problems, the standard SVM criterion gives a good solution even with highly unbalanced task.

We proposed, instead, to work on new similarity measures. The intra-impostor distance distribution is larger than the intra-client distribution due to the problem itself. We thus proposed, based on the idea of the vicinity function proposed by Vapnik (2000), to add a Gaussian noise over the negative examples only. Unfortunately, we had to apply some empirical simplification in order to make this new approach feasible, which made it less principled. However, this suggests to modify the standard similarity measure, for example by adapting the kernel (Kwok and Tsang, 2003) or by learning a similarity measure, as done by Chopra et al. (2005) for face verification.

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

J. Kwok and I. Tsang. Learning with idealized kernels. In *International Conference on Machine Learning, ICML*, 2003.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Computer Society Confer-*

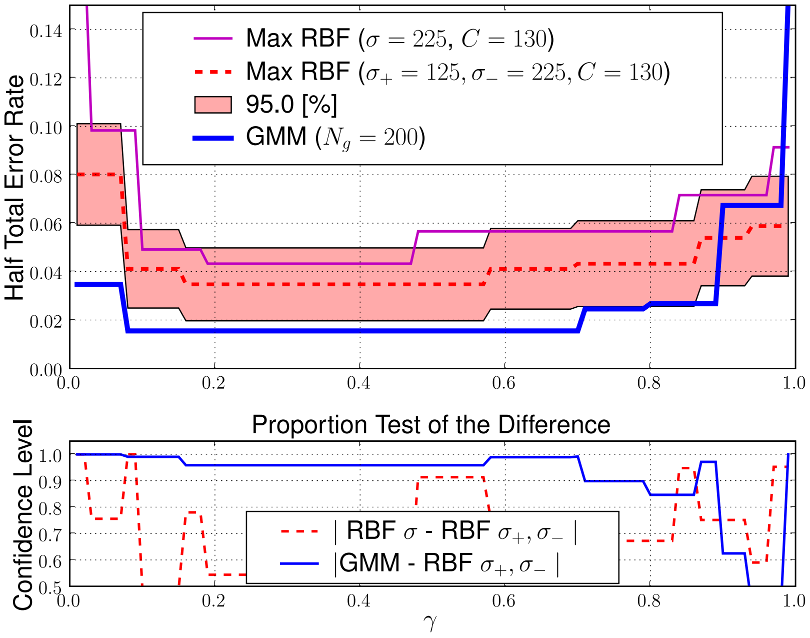


Figure 8.5. EPC curves on the test set of the Banca database for the best σ, σ_+, σ_- Max RBF kernel and GMM based system.

ence on Computer Vision and Pattern Recognition (CVPR), 2005.

9 Conclusion

In this thesis, we addressed the problem of text-independent speaker verification from a machine learning point of view. The main purpose was to consider this problem as a two-class classification problem for each speaker. As suggested by the machine learning theory, the model used to solve this kind of problems should be discriminant, while the current state-of-the-art text-independent speaker verification models are based on Gaussian Mixture Models (GMMs) which are not apparently discriminant.

9.1 Contribution of the Thesis

We first described the state-of-the-art models as found in the speaker verification literature. Unfortunately, the performance measures used to compare models are often biased, including Equal Error Rate and Detection Error Trade-off (DET) curves. We have thus proposed new kinds of curves called Expected Performance Curves (EPCs) that allow to compare fairly systems for a range of decision thresholds. This work was published in:

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

and more specifically for speaker verification in:

> CONTRIB S. Bengio and J. Mariéthoz. The expected performance curve: a new assessment measure for person authentication. In *Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop*, 2004

Moreover as no statistical test, such as Z-test, was applicable to the speaker verification problem, we adapted the Z-test in order to properly measure whether two systems were statistically significantly different in terms of Half Total Error Rate (HTER) and Detection Cost Function (DCF). This work was published in:

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

We have defined an experimental setup, including a protocol for the use of discriminant models. We performed experiments using three databases: Switchboard coming from the NIST campaign, the Banca database and the Polyvar database. The original benchmark Banca database and its protocol descriptions were published in:

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

The Polyvar database and its protocol descriptions were published in:

> CONTRIB F. Bimbot, M. Blomberg, L. Boves, G. Chollet, C. Jaboulet, B. Jacob, J. Kharroubi, J. Koolwaaij, J. Lindberg, J. Mariéthoz, C. Mokbel, and H. Mokbel. An overview of the picasso project research activities in speaker verification for telephone applications. In *6th European Conference on Speech Communication and Technology — Eurospeech'99*, volume 5, pages 1963–1966, Budapest, Hungary, september 1999

In order to propose new approaches based on discriminant models, a general framework has been developed for speaker verification that includes several kinds of models: probabilistic models such as GMMs and non-probabilistic models such as Support Vector Machines (SVMs). This framework was originally presented in:

> CONTRIB J. Mariéthoz and S. Bengio. A kernel trick for sequences applied to text-independent speaker verification systems. IDIAP-RR 77, IDIAP, 2005

This framework was then extended for the case of score normalization for both probabilistic and non-probabilistic based models. Score normalization is often used to compensate unmatched conditions between data used to train the model and test accesses. A generalized score normalization framework was proposed and enlights the hypothesis implicitly done when T- and Z- normalization are used. This new framework can be used to develop future score normalization procedures and is not limited to a Gaussian score distribution. This work was published in:

CONTRIB J. Mariéthoz and S. Bengio. A unified framework for score normalization techniques applied to text independent speaker verification. *IEEE Signal Processing Letters, Volume 12*, 12, 2005. IDIAP-RR 04-62

In order to better understand the state-of-the-art GMM based system, we analyzed it more deeply and mentioned several modifications suggested by the speaker verification community in order to reach the state-of-the-art performance. We showed that theses modifications make the GMM based model discriminant and is equivalent, using reasonable assumptions, to a mixture of linear classifiers. In order to interpret GMMs as mixtures of experts, we used an algorithm called “synchronous alignment”, published in:

CONTRIB J. Mariéthoz, Dominique Genoud, Frédéric Bimbot, and Chafik Mokbel. Client / world model synchronous alignement for speaker verification. In *6th European Conference on Speech Communication and Technology — Eurospeech'99*, Budapest, Hungary, September 1999

The Maximum A Posteriori (MAP) adaptation algorithm is also an important modification in order to obtain good performance. MAP adaptation methods where compared to other standard approaches and this comparison was published in:

CONTRIB J. Mariéthoz and S. Bengio. A comparative study of adaptation methods for speaker verification. In *International Conference on Spoken Language Processing ICSLP*, pages 581–584, Denver, CO, USA, September 2002. IDIAP-RR 01-34

We first tried to develop discriminant models using information coming from the GMM based system by replacing the Bayes decision function of state-of-the-art GMM based systems, which can be seen as a linear function of two log likelihoods with a fixed slope equal to one, by learning a discriminant decision function with an SVM. Learning the decision function suggests that the discriminant models should be client dependent. This work was published in:

> CONTRIB S. Bengio and J. Mariéthoz. Learning the decision function for speaker verification. In *IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP*, Salt Lake, City, USA, 2001. IDIAP-RR 00-40

Apart from log likelihoods, several other values could be inputted to an SVM. We can for instance enrich this representation with local log likelihood ratios (LLRs) for each Gaussian in order to increase the size of the input vector. After analyzing the results, we concluded that having only one discriminant model for all clients seems to be a limitation and it could be preferable to have a client dependent discriminant model that could be based on a whole sequence of feature vectors. The use of discriminant models as a decision function and using a large vector of LLRs was proposed in:

> CONTRIB J. Mariéthoz and S. Bengio. An alternative to silence removal for text-independent speaker verification. IDIAP-RR 51, IDIAP, Martigny, Switzerland, 2003

These models suggest that the SVM is a good candidate for the speaker verification problem, especially with its ability to maximize the margin. Indeed, to train one model per speaker, we have very few client accesses (often one) and hundreds of impostor accesses. As we observed for SVM based systems, the problem is separable and maximizing the margin guarantees a reasonable solution over all possible solutions that give zero training error. Unfortunately, default SVMs can handle only fixed size vectors and we thus had to propose new kernels that can handle variable length sequences of vectors. We first developed a new Mean operator sequence kernel that computes the average of all sub-kernels over all pairs of frames. We showed that it generalizes the GLDS kernel proposed by Campbell (2002) with the advantage to better control the capacity of the SVM model, while making possible the use of infinite space kernels, such as Radial Basis Functions (RBFs).

We also proposed a new Max operator sequence kernel that searches for each frame of one sequence, the frame of the other sequence that best matches. It makes more sense and outperforms the standard approach. Unfortunately it does not satisfy the Mercer conditions but still converges very well for the

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.

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 also proposed a method to smooth the Max operator based kernel. The good empirical results suggest that a more sophisticated method to enforce some temporal constraints can be a topic of future research.

Unfortunately, the Max operator method is computationally costly for long sequences. We thus proposed clustering techniques to make the algorithm tractable for long sequence based databases, such as Switchboard (NIST).

Finally, as speaker verification is a highly unbalanced two-class classification problem, it might be important to consider specific training criteria for such cases. As for most tested SVM kernels the problem is separable, the classical approach to compensate the unbalanced dataset are useless. We concluded that the solution found by the SVM is good even for highly unbalanced class examples.

A new SVM criterion that allows to deal with unbalanced class problems and interprets the output of an SVM as a probability has been published in:

> CONTRIB 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

We finally proposed a new research direction based on new distance measures. Such a measure should allow a training negative example to cover other unseen impostors. Our new approach is based on the vicinity function proposed by Vapnik (2000). The main idea is to assume a Gaussian noise on the negative examples. Even if this method is not principled, it gives good empirical results and suggests several extensions of our research for this problem. A Gaussian noise can also be added in order to capture the acquisition channel variability.

Overall, in this thesis, we used the machine learning theory to develop a good methodology and a good framework for the speaker verification problem. We proposed several new discriminant models that improve the HTER performance of the state-of-the-art systems, but more importantly that increase the

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

understanding of these models.

This opens several new research perspectives. For example, the score normalization framework allows the use of new score normalization procedures based on non-Gaussian score distribution estimation. The smoothing Max operator kernel suggests to consider some temporal constraints. It seems also very promising to develop a new similarity measure that includes some noise on the data distribution, either to allow a negative example to cover more unseen impostors but also to account for acquisition channel variation. An other general problem is that in "real" life we have no idea of what is a true impostor, which kind of strategy he/she can develop to break the system. Moreover, we cannot base our intuition on a human criterion: we showed in Mariéthoz and Bengio (2005) that current verification systems are robust to professional imitators while humans are not, while at the opposite automatic systems are less robust to noise than humans. In terms of applications, there is an evident need for mobile phone applications using some form of person identification. Thus speaker verification systems should be more and more robust to various recording conditions. Even if already existing solutions are robust for reasonable levels of noise, better robustness is still needed for high levels of noise. A potential solution could be the use of pre-processing methods such as selecting an audio source using a microphone array. An other interesting approach could be to use different biometric modalities such as speech, face, lips, etc. Existing approaches often simply fuse the scores obtained by each modality, but more principled approach to jointly consider all modalities during training are still needed.

9.2 Other contributions

All the algorithms developed in this thesis are based on a machine learning library called "Torch". This library is widely used by the machine learning community and is available at http://www.torch.ch. The author is one of the main contributor of this software.

During the course of this thesis, several other, yet related, scientific contributions were accepted for publications but not described here. They are:

CONTRIB S. Marcel, J. Mariéthoz, Y. Rodriguez, and F. Cardinaux. Bi-modal face and speech authentication: a biologin demonstration system. In *Workshop on Multimodal User Authentication (MMUA)*, 2006. IDIAP-RR 06-18

J. Mariéthoz and S. Bengio. Can a professional imitator fool a GMM-based speaker verification system? IDIAP-RR 61, IDIAP, 2005.

CONTRIB Y. Rodriguez, F. Cardinaux, S. Bengio, and J. Mariéthoz. Measuring the performance of face localization systems. *Image and Vision Computing*, 24(8):882–893, 2006

CONTRIB M. Liwicki, A. Schlapbach, H. Bunke, S. Bengio, J. Mariéthoz, and J. Richiardi. Writer identification for smart meeting room systems. In *Seventh IAPR Workshop on Document Analysis Systems, DAS*, 2006

CONTRIB J. Mariéthoz and S. Bengio. Can a professional imitator fool a GMM-based speaker verification system? IDIAP-RR 61, IDIAP, 2005

CONTRIB J. Mariéthoz and S. Bengio. A new speech recognition baseline system for numbers 95 version 1.3 based on torch. IDIAP-RR 16, IDIAP, 2004

CONTRIB Y. Rodriguez, F. Cardinaux, S. Bengio, and J. Mariéthoz. Estimating the quality of face localization for face verification. In *IEEE International Conference on Image Processing, ICIP*, 2004

CONTRIB C. Sanderson, S. Bengio, H. Bourlard, J. Mariéthoz, R. Collobert, M.F. BenZeghiba, F. Cardinaux, and S. Marcel. Speech & face based biometric authentication at idiap. In *International Conference on Multimedia and Expo, ICME*, 2003

CONTRIB S. Bengio, C. Marcel, S. Marcel, and J. Mariéthoz. Confidence measures for multimodal identity verification. *Information Fusion*, 3(4):267–276, 2002

Bibliography

R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10: 42–54, 2000.

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

S. Bengio and J. Mariéthoz. The expected performance curve: a new assessment measure for person authentication. In *Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop*, 2004.

S. Bengio and J. Mariéthoz. Learning the decision function for speaker verification. In *IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP*, Salt Lake, City, USA, 2001. IDIAP-RR 00-40.

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

S. Bengio, C. Marcel, S. Marcel, and J. Mariéthoz. Confidence measures for multimodal identity verification. *Information Fusion*, 3(4):267–276, 2002.

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

F. Bimbot, M. Blomberg, L. Boves, G. Chollet, C. Jaboulet, B. Jacob, J. Kharroubi, J. Koolwaaij, J. Lindberg, J. Mariéthoz, C. Mokbel, and H. Mokbel. An overview of the picasso project research activities in speaker verification for telephone applications. In *6th European Conference on Speech Communication and Technology — Eurospeech'99*, volume 5, pages 1963–1966, Budapest, Hungary, september 1999.

F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovsk-Delacrétaz, and D. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Applied Signal Processing*, 4:430–451, 2004.

C. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

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

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

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.

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.

G. Chollet, J.-L. Cochard, A. Constantinescu, C. Jaboulet, and P. Langlais. Swiss french polyphone and polyvar: telephone speech databases to model inter- and intra-speaker variability. IDIAP-RR 01, IDIAP, 1996.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

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

R. Collobert and S. Bengio. Links between perceptrons, MLPs and SVMs. In *International Conference on Machine Learning, ICML*, 2004.

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

A. P. Dempster, N. M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1(39):1–38, 1977.

L. Devroye and L. Gyorfi. *A Probabilistic Theory of Pattern Recognition.* Springer, 20 February 1997.

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

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.

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

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

B. Jurie, F. and Triggs. Creating efficient codebooks for visual recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 604– –610, 17 October 2005.

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.

J. Koolwaaij. *Automatic Speaker Verification in Telephony: a probabilitic approach.* PrintPartners Ipskamp B.V., Enschede, 2000.

J. Kwok and I. Tsang. Learning with idealized kernels. In *International Conference on Machine Learning, ICML*, 2003.

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

Kung-Pu Li and J. E. Porter. Normalizations and selection of speech segments for speaker recognition scoring. In *Proceedings of the IEEE ICASSP*, pages 595–597, 1988.

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in non-standard situations. *Machine Learning*, 46:191–202, 2002.

R. P. Lippmann. Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities. *Neural Computation*, 3:461–483, 1992.

M. Liwicki, A. Schlapbach, H. Bunke, S. Bengio, J. Mariéthoz, and J. Richiardi. Writer identification for smart meeting room systems. In *Seventh IAPR Workshop on Document Analysis Systems, DAS*, 2006.

J Lüttin. Evaluation protocol for the the XM2FDB database (lausanne protocol). IDIAP-COM 05, IDIAP, 1998.

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

S. Marcel, J. Mariéthoz, Y. Rodriguez, and F. Cardinaux. Bi-modal face and speech authentication: a biologin demonstration system. In *Workshop on Multimodal User Authentication (MMUA)*, 2006. IDIAP-RR 06-18.

J. Mariéthoz and S. Bengio. A comparative study of adaptation methods for speaker verification. In *International Conference on Spoken Language Processing ICSLP*, pages 581–584, Denver, CO, USA, September 2002. IDIAP-RR 01-34.

J. Mariéthoz and S. Bengio. An alternative to silence removal for text-independent speaker verification. IDIAP-RR 51, IDIAP, Martigny, Switzerland, 2003.

J. Mariéthoz and S. Bengio. A unified framework for score normalization techniques applied to text independent speaker verification. *IEEE Signal Processing Letters, Volume 12*, 12, 2005. IDIAP-RR 04-62.

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.

J. Mariéthoz and S. Bengio. A new speech recognition baseline system for numbers 95 version 1.3 based on torch. IDIAP-RR 16, IDIAP, 2004.

J. Mariéthoz and S. Bengio. A kernel trick for sequences applied to text-independent speaker verification systems. IDIAP-RR 77, IDIAP, 2005.

J. Mariéthoz and S. Bengio. Can a professional imitator fool a GMM-based speaker verification system? IDIAP-RR 61, IDIAP, 2005.

J. Mariéthoz, Dominique Genoud, Frédéric Bimbot, and Chafik Mokbel. Client / world model synchronous alignement for speaker verification. In *6th European Conference on Speech Communication and Technology — Eurospeech'99*, Budapest, Hungary, September 1999.

A Martin. Personal communication. http://www.nist.gov/speech/staff/martinal.htm, 2004.

A. Martin and M. Przybocki. The NIST 1999 speaker recognition evaluation - an overview. *Digital Signal Processing*, 10:1–18, 2000.

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

K. Messer, J. Kittler, M. Sadeghi, S. Marcel, C. Marcel, S. Bengio, F. Cardinaux, C. Sanderson, J. Czyz, L. Vandendorpe, S. Srisuk, M. Petrou, W. Kurutach, A. Kadyrov, R. Paredes, B. Kepenekci, F. B. Tek, G. B. Akar, F. Deravi, and N. Mavity. Face verification competition on the XM2VTS database. In *4th International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA*. Springer-Verlag, 2003.

J. Navratil and Ganesh N. Ramaswamy. The awe and mystery of t-norm. In *Proc. of the European Conference on Speech Communication and Technology*, pages 2009–2012, 2003.

H Paugam-Moisy, A. Elisseeff, and Y. Guermeur. Generalization performance of multiclass discriminant models. In *Int. Joint Conf. on Neural Networks (IJCNN)*, 2000.

J. C. Platt. Probabilities for SV machines. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

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

D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions On Speech and Audio Processing*, 3(1), 1995.

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

Y. Rodriguez, F. Cardinaux, S. Bengio, and J. Mariéthoz. Estimating the quality of face localization for face verification. In *IEEE International Conference on Image Processing, ICIP*, 2004.

Y. Rodriguez, F. Cardinaux, S. Bengio, and J. Mariéthoz. Measuring the performance of face localization systems. *Image and Vision Computing*, 24 (8):882–893, 2006.

C. Sanderson, S. Bengio, H. Bourlard, J. Mariéthoz, R. Collobert, M.F. Ben-Zeghiba, F. Cardinaux, and S. Marcel. Speech & face based biometric authentication at idiap. In *International Conference on Multimedia and Expo, ICME*, 2003.

F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

G. W. Snedecor and W. G. Cochran. *Statistical Methods*. Iowa State University Press, 1989.

A. Solomonoff, C. Quillen, and W.M. Campbell. Channel compensation for svm speaker recognition. In *Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop*, pages 57–62, 2004.

H. L. Van Trees. *Detection, Estimation and Modulation Theory, vol. 1*. Wiley, New York, 1968.

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

P. Verlinde, G. Chollet, and M. Acheroy. Multi-modal identity verification using expert fusion. *Information Fusion*, 1:17–33, 2000.

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

J.L. Wayman. Confidence interval and test size estimation for biometric data. In *Proceedings of the IEEE AutoID Conference*, 1999.

M.H. Zweig and G. Campbell. ROC plots: a fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, 39(4):561–577, 1993.