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:

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

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

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.

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:

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

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 $C- = 10\, C+$.

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

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-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.

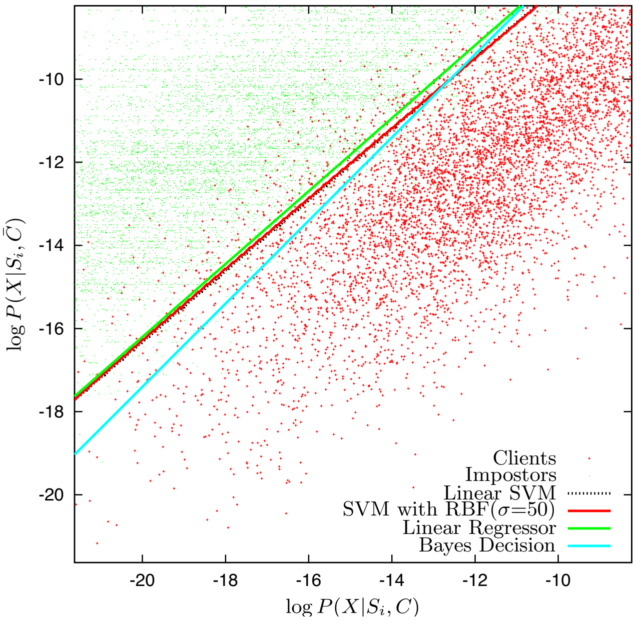

Figure 6.1. Different models to separate clients and impostors, in a text independent task on the Polyvar database.

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

REF 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.

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.

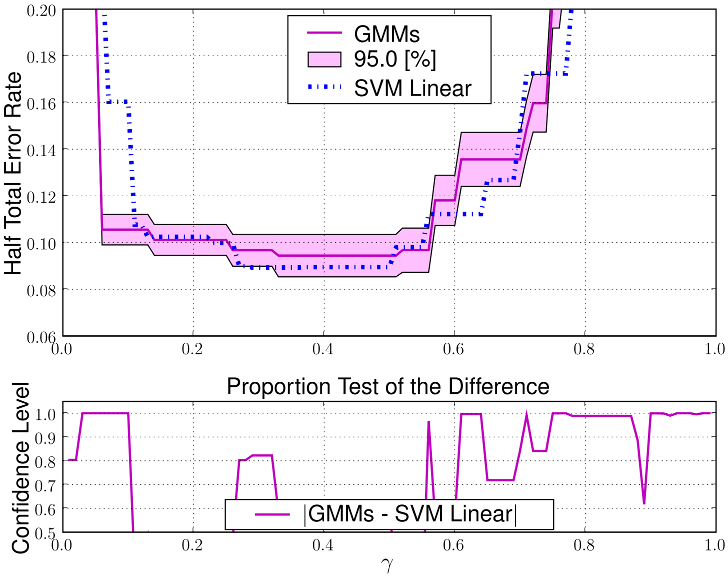

Figure 6.2. Results on the test set of the NIST database: GMMs vs linear SVM on LLR ($C+ = 3$ and $C- = 30$).

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 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:

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

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.

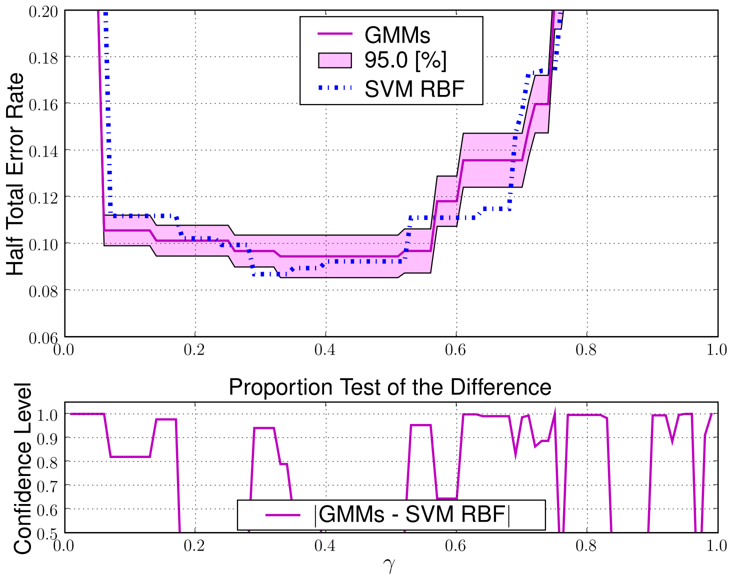

Figure 6.3. Results on the test set of the NIST database: GMMs vs RBF SVM on LLR ($C+ = 3$, $C- = 30$ and $\sigma = 5$)

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.

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.

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

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

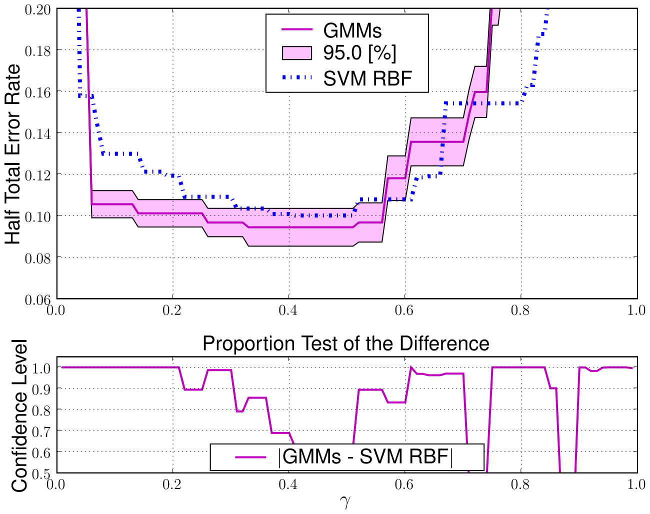

Figure 6.4. Results on the test set of the NIST database: GMM vs RBF SVM on Gaussian LLR ($C+ = 20, C- = 200$ and $\sigma = 800$).

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).

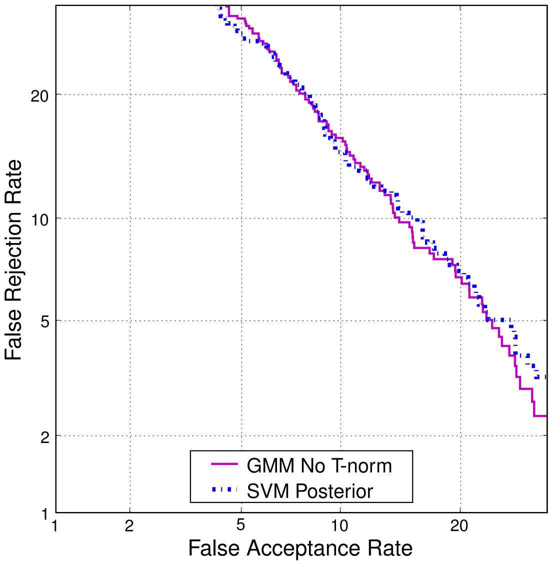

Figure 6.5. Results on the development set of the NIST database: GMM without T-norm vs SVM trained on posteriors. ($C+ = C- \rightarrow \infty$ and $N_g = 1000$).

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

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

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.

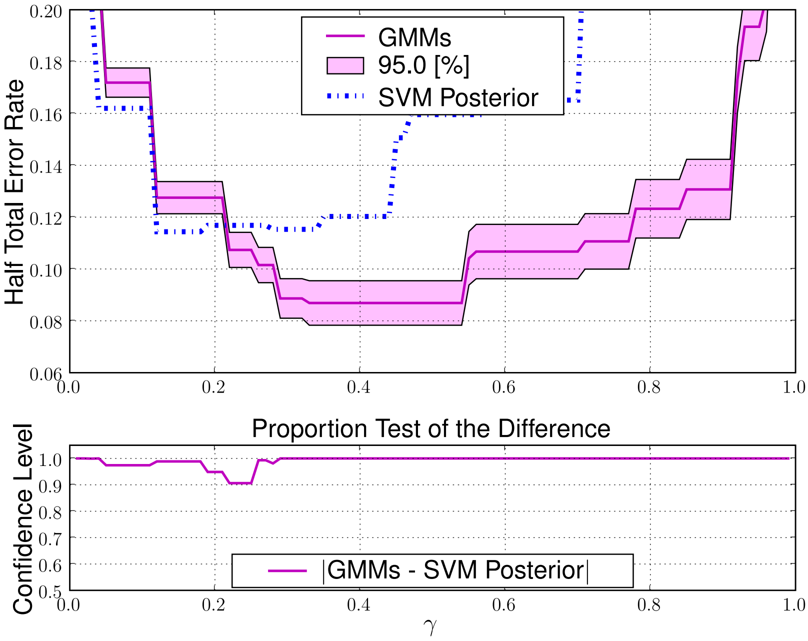

Figure 6.6. Results on the test set of the NIST database: GMM vs SVM Posterior. ($C+ = C- = \infty$ and $N_g = 1000$).

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 produced 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:

> 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.

REF 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.