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 proclaim-

ing 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:

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

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

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:

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

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:

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

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.

REF 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.

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 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

REF 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.

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 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:

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

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:

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

REF 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.

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:

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

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

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

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 $T(g)/T$.

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:

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

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.

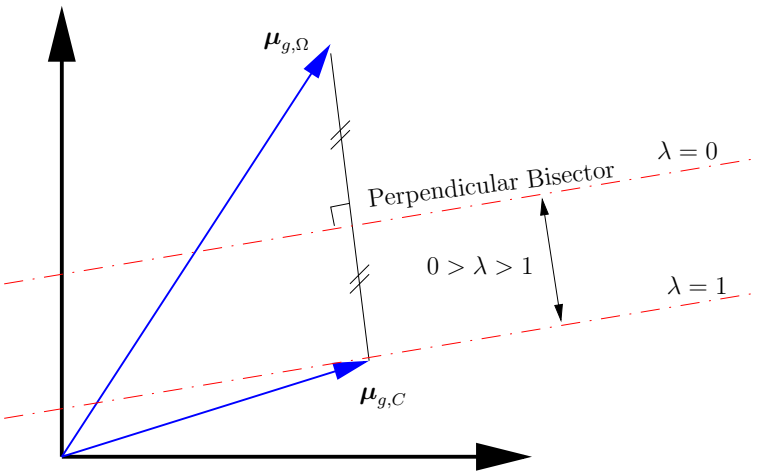

Figure 5.1. Perpendicular bisector interpretation.

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 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	$\pm$0.84	$\pm$0.82	$\pm$0.89	$\pm$0.82

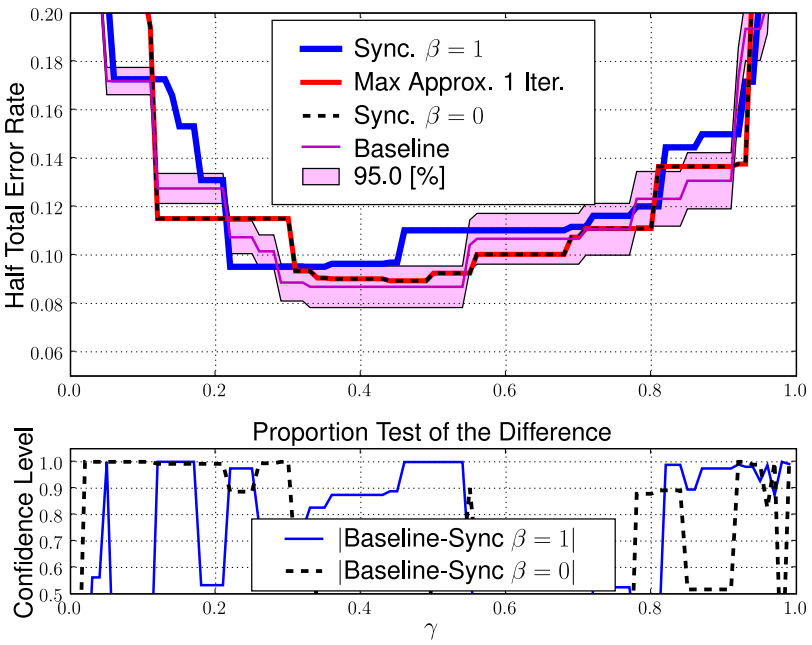

Figure 5.2. Results on the NIST2002 database: GMM baseline results, max approximation with only one iteration for EM training, synchronous alignment on client and on world model.

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.

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

REF 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.

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

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 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:

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

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

Taking the logarithm, we finally obtain:

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

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

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

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

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:

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

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

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

REF 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.

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

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

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

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:

$$\sigma_M = \sigma_C = \sigma_{\bar{C}} \ . \quad (5.40)$$

We thus obtain:

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

If we now define the means as:

$$\begin{aligned} \mu_C &= \text{llr} \\ \mu_{\bar{C}} &= \mu_M \end{aligned} \quad (5.42)$$

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:

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

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:

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

$$\mu_J = \frac{1}{J}\sum_j \text{llr}(\text{X}_j) \tag{5.44}$$

$$\sigma_J = \sqrt{\frac{1}{J}\sum_j (\text{llr}(\text{X}_j) - \mu_J)^2} \tag{5.45}$$

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:

$$\hat{p}(\text{llr}|C, \text{X}, S_i) = \mathcal{N}(\text{llr}; \mu_C, \sigma_C) \tag{5.46}$$

$$\hat{p}(\text{llr}|\bar{C}, \text{X}, S_i) = \mathcal{N}(\text{llr}; \mu_{\bar{C}}, \sigma_{\bar{C}}) \tag{5.47}$$

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

If we now define the means as follows:

$$\begin{aligned} \mu_C &= \text{llr} \\ \mu_{\bar{C}} &= \mu_J \end{aligned} \tag{5.48}$$

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:

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

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:

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

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

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

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

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

This inequation has a real solution only when $\Theta > 0$, which is true if $\text{llr} > \mu_M$. 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

To verify the validity of our framework and the underlying assumptions, we first compared the standard T-normalization and the version derived from the

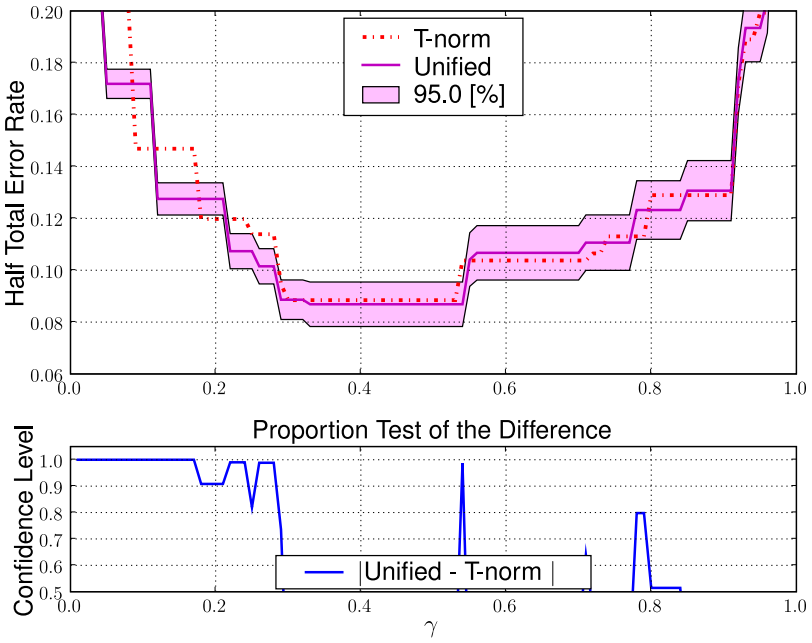

Figure 5.3. EPC curves on the NIST 2002 test set for the T-norm and unified framework T-norm systems.

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 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 tech-

REF 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.

niques 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 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:

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

REF 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.

REF 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.

Taking the log, we get:

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

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

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

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.

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

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

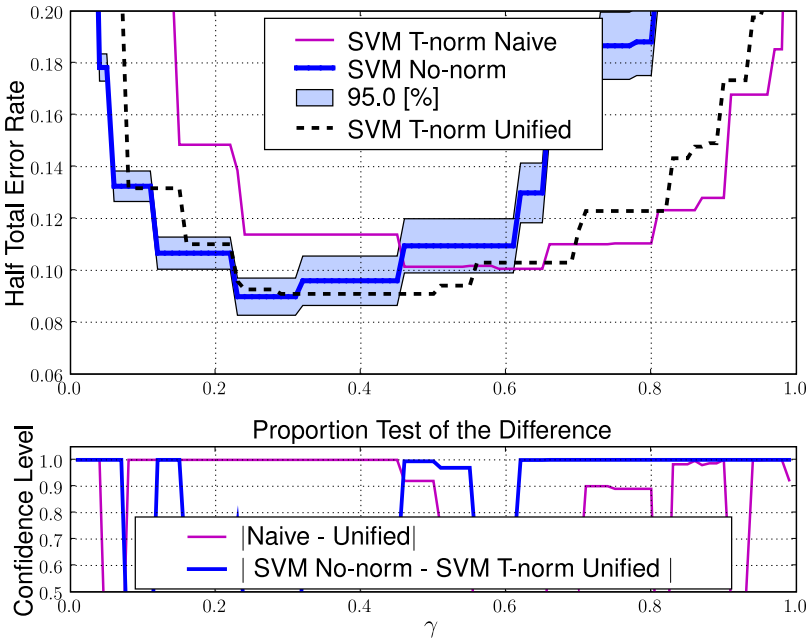

Figure 5.4. EPC curves on the NIST test set for the T-norm and unified framework T-norm systems.

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.