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

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

REF 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.

approaches but the most interesting one, from our point of view, is based on Support Vector Machines (SVMs). SVMs yield similar or even better performance 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

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

REF 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.

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.

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.

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

REF 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.

REF 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.

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.

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:

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

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

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

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

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

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

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

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 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.

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

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,
- 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 forc-

ing 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 transform (2.17) as follows:

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

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.

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

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

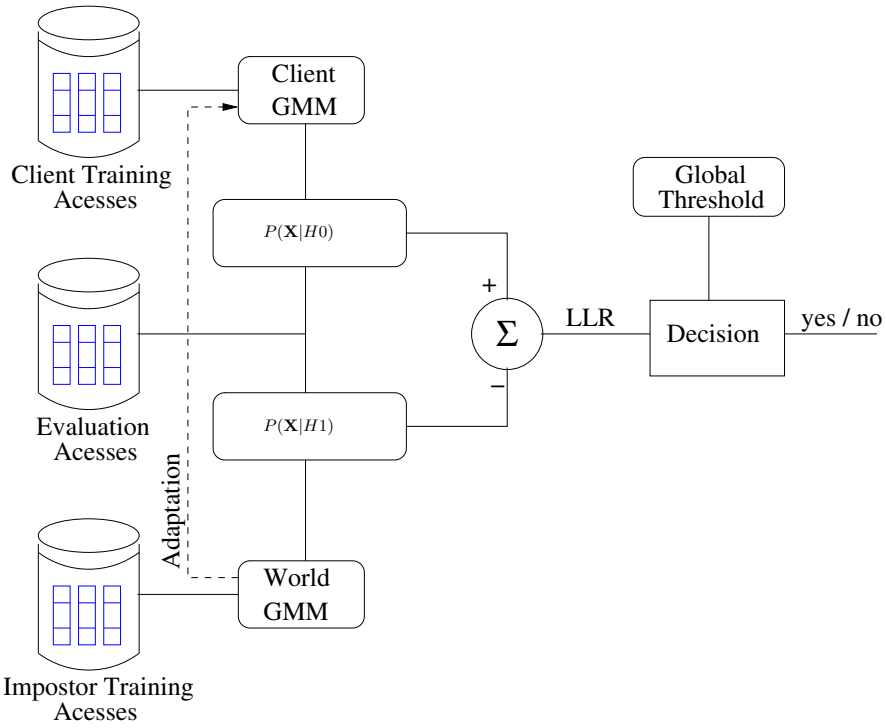

Figure 2.1. A summary of a state-of-the-art GMM based system.

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 sub-part 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 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.

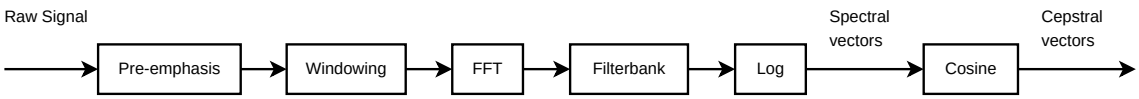

Figure 2.2. Modular Representation of a Filter-bank-based Cepstral parameterization.

REF 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.

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 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:

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

REF 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.

REF 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.

where $\widehat{\text{llr}}(\mathbf{X})$ is the new normalized score, $\text{llr}(\mathbf{X})$ 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 $\mathbf{X}_0$:

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

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

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:

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

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:

$$\mu_J = \frac{1}{J}\sum_j \text{llr}_{\text{S}_i}(\text{X}_j) \tag{2.26}$$

$$\sigma_J = \sqrt{\frac{1}{J}\sum_j (\text{llr}_{\text{S}_i}(\text{X}_j) - \mu_J)^2} \tag{2.27}$$

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.

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 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

REF 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.

REF 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.

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

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.

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

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

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

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.

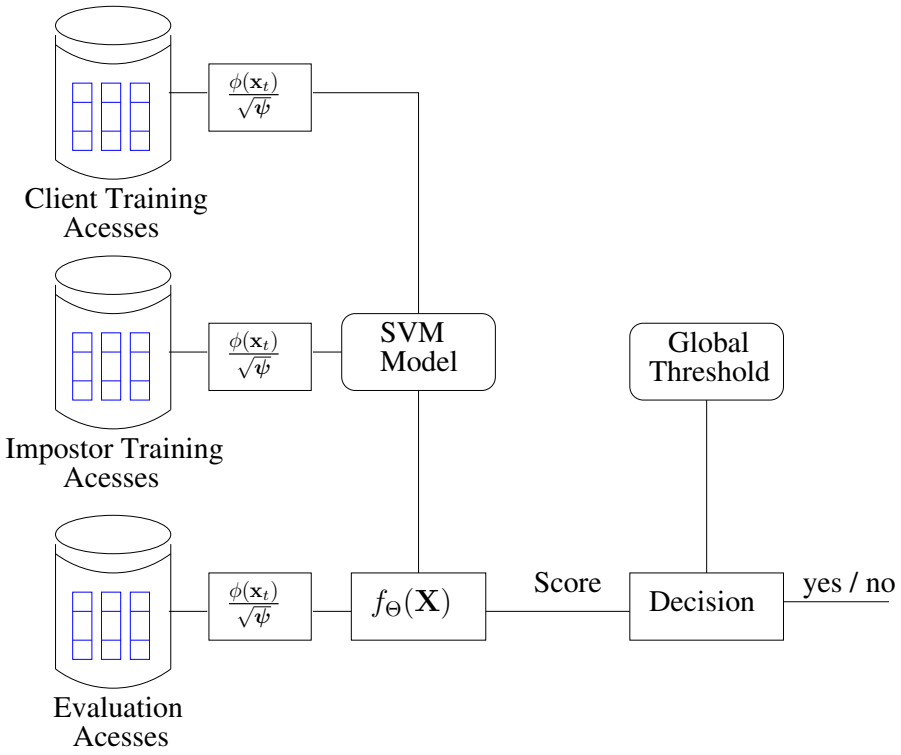

Figure 2.3. A summary of a state-of-the-art GLDS SVM based system.

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.