IDIAP RESEARCH REPORT

Compensating User-Specific Information with User-Independent Information in Biometric Authentication Tasks

Norman Poh [a] Samy Bengio [a]

IDIAP–RR 05-44

July 2005

submitted for publication

[a] IDIAP, CP 592, 1920 Martigny, Switzerland

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

IDIAP Research Report 05-44

Compensating User-Specific Information with User-Independent Information in Biometric Authentication Tasks

Norman Poh Samy Bengio

July 2005

submitted for publication

Abstract. Biometric authentication is a process of verifying an identity claim using a person's behavioral and physiological characteristics. This is in general a binary classification task because a system either accepts or rejects an identity claim. However, a biometric authentication system contains many users. By recognizing this fact, better decision can be made if *user-specific* information can be exploited. In this study, we propose to combine user-specific information with user-independent information such that the performance due to exploiting both information sources *does not* perform worse than either one and in some situations can *improve significantly* over either one. We show that this technique, motivated by a standard Bayesian framework, is applicable in two levels, i.e., fusion level where multiple (multimodal or intramodal) systems are involved, or, score normalization level, where only a single system is involved. The second approach can be considered a novel score normalization technique that combines both information sources. The fusion technique was tested on 32 fusion experiments whereas the normalization technique was tested on 13 single-system experiments. Both techniques that are originated from the same principal share a major advantage, i.e., due to prior knowledge as supported by experimental evidences, *few or almost no free parameter* are actually needed in order to employ the mentioned techniques. Previous works in this direction require at least 6 to 10 user-specific client accesses. However, the proposed technique requires as few as 2 user-specific client accesses, hence overcoming the learning problem with extremely few user-specific client samples. Finally, but not the least, a non-exhaustive survey on the state-of-the-arts of incorporating user-specific information in biometric authentication is also presented.

1 Introduction

1.1 Fundamentals

Biometric authentication (BA) is a problem of verifying an identity claim using a person's behavioral and physiological characteristics. Although it is an important alternative to traditional authentication methods such as keys ("something one has", i.e., by possession) or PIN numbers ("something one knows", i.e., by knowledge), it is sensitive to various factors. Some of these factors are the deformable nature of biometric traits, corruption by environmental noise, variability of biometric traits over time, the state of users (especially behavioral biometrics) and occlusion by the user's accessories.

Due to inherent properties in each biometric and external manufacturing constraints in the sensing technologies, no single biometric trait can achieve 100% authentication performance. This problem can be alleviated by combining two or more biometric traits, also known as the field of multimodal biometric authentication. In the literature, there are several methods to combine multimodal information. These methods are known as *fusion techniques*. From an architectural point of view, fusion can be done either at the *feature level* (extracted or internal representation of the data stream) or *score level* (output of a single system). Among the two, the latter is most commonly used in the literature. Some studies further categorize three levels of score level fusion [6], namely, fusion using the scores directly, using a *set of most probable* category labels (called abstract level) or using the *single most probable* category label (called decision level). We will focus on the score for two reasons: the last two cases can be derived from the score and more importantly, by using only labels instead of scores, precious information is lost.

1.2 Terminology Referring to User-specific Information

BA is in general considered a two-class problem, i.e., a BA system is designed to accept a genuine access request or to reject an imposture attempt. However, such task is composed of many users. Hence, BA is actually an *aggregated* set of two-class tasks. This implies that if a BA system can make the accept/reject decision on a *per user* basis, the system performance may be improved. We shall call this strategy *user-specific* strategy.

Due to the evolving nature of this fields, several terms have been introduced by different authors and none dominates the others. To avoid confusion, we will adopt some terms at least consistent in this paper. Suppose that each authorized person is identified by a unique identity claim $j \in \mathcal{J} \equiv \{1, \dots, J\}$ and there are J identities. In the following list, terms in bold are used in this paper and terms separated by "/" are considered synonyms. These terms are also reported in the literature [12, 35] (to name a few).

- **Client**/user: (noun) a person that should be granted access by the system and is assigned a unique identity j.
- **Impostor**/abuser: (noun) a person that should be denied access by the system because he/she is not the person being assigned identity j.
- **User-specific**/client-dependent/local: (adjective) on a per client basis.
- **User-independent**/client-independent/global/common: (adjective) indifferent to each client.
- **User-adapted**: (adjective) that makes use of *both* user-specific and user-independent information.
- **Client-centric**/target-centric: (adjective) that makes use of user-specific client accesses only.
- **Impostor-centric**: (adjective) that makes use of user-specific impostor accesses only.
- **Client-impostor centric**/target-impostor centric: (adjective) that makes use of both user-specific client and impostor accesses.

1.3 User-Specific Techniques in Biometric Authentication

In the literature, there are several studies that rely on user-specific information, at different levels of a BA architecture. They can further be divided into two categories, i.e., at feature level or score level. They are listed below with the first two being user-specific techniques at the feature level and the rest being techniques at the score level:

- **User-specific feature set**. In [8], it was demonstrated that the performance of a speaker verification task can be enhanced by using a subset of features for each client. These features are chosen using a feature selection technique.

- **User-specific model/template**. This is a standard approach whereby a BA system builds a feature template or a model on a *per client* basis. A feature template can be the originally scanned biometric data or representation extracted from it. A model, on the other hand, refers to a set of parameters/numbers derived to represent the feature set of a particular user. For instance, the-state-of-the-art approach in speaker verification is based on user-specific model [32]. Recent techniques in face verification also follow the same trend, e.g., [9] using local features and [37] using client-specific Fisher's projection.

- **User-specific score (normalization)**. This approach is also known as score normalization. The most representative example is called Z-norm and first proposed by [13], which relies on user-specific impostor scores to carry out the normalization. In [33], a similar version of Z-Norm but using only user-specific client scores was reported. However, this technique requires much more user-specific client accesses. The authors' experiments were based on 5 accesses per client. Since the first work by [13], the form of normalization has not been changed much although the context of application is extended beyond that of mitigating user-induced score variations, such as T-Norm [1] (aiming at extenuating the mismatch during test), H-norm [16] (aiming at extenuating the mismatch due to the use of different handsets) or D-Norm [3] (aiming at reducing model-induced variations and is specific to Gaussian Mixture Models). A recent form of normalization which makes use of both user-specific client and impostor scores is called F-norm [31]. In the terms used in [12], Z-norm is impostor-centric because the normalization is carried out with respect to the impostor distributions (or more precisely scores derived from user-specific impostor data), whereas F-norm is *client-impostor centric* because it makes use of both user-specific client and impostor scores.

- **User-specific fusion**. This technique was proposed by [18] using a linear weighing scheme to weigh the outputs of several BA multimodal systems while a non-linear version, achieved via Multi-Layer Perceptron (MLP) was reported in [21]. In [10], a Support Vector Machine (SVM) classifier was used to construct a user-specific fusion function while in [11], a Bayesian classifier was used.

- **User-specific threshold**. This class of techniques is commonly applied to speaker verification tasks [13, 33, 23, 28, 7], for instance.

The score-level user-specific techniques are by far more explored than the feature-level ones as far as user-specific information is concerned. Often, the score-level techniques are used together with the feature-level techniques. For instance, the state-of-the-art speaker verification technique based on adapted Gaussian Mixture Model [32, 1] use both user-specific model and user-specific score normalization. Other similar techniques include [12] applied to signature verification or [34] applied to face verification.

User-specific score and user-specific threshold are actually not two separate techniques but closely linked together. It was shown [30] that if a user-specific threshold includes a global threshold, then the same user-specific threshold procedure can *equivalently* be implemented by a user-specific score normalization technique.

A recent study [35] proposed a new paradigm consisting of two dichotomies: user-specific/user-independent fusion (called "local/global learning" by the author) and user-specific/user-independent threshold (called "local/global decision"). These two dichotomies thus give four categories of methods to incorporate user-specific information, at the score level. Our study here separates the different levels where user-specific information can be incorporated and thus generalizes the paradigm. It was claimed that combining both user-specific fusion and threshold techniques that they called "local learning with local decision" achieved the best performance. Yet in [25], another possible combination was proposed, i.e., between user-specific score normalization (based on Z-norm) and user-independent fusion. It is probable that this framework has been reported elsewhere in the literature. In [30], a similar framework was employed except that F-norm was used in place of Z-norm. Employing F-norm in place of Z-norm gives an additional advantage, i.e., an added discrimination ability between the client and impostor classes, because F-norm is *client-impostor centric* while Z-norm is *impostor centric*. It should be cautioned that when there is no user-specific variation which in turn affects the scores, applying user-specific normalization techniques may deteriorate the performance. This is especially true in a text-independent speaker verification task whereby linguistic contents (as captured by a speaker's model) from a user may be different from another user. Finally, all user-specific strategies applied at the score level may suffer from lack of user-specific client data (scores). This can be a major obstacle. The following section reviews several works that can compensate this problem and we will focus on *user-specific fusion*.

1.4 Existing Techniques in User-specific fusion

Often, due to user-friendliness aspect of a BA system, at least at the initial phase of system operation, very few biometric data samples are acquired from a new client. Hence, most of the precious data samples are used to train/construct the user-specific model (with possibly adaptation of features from other persons) while few samples are left to compute the unbiased user-specific score normalization. One way to counter the lack of scores is to use cross-validation so that the biometric samples that are not used to train the model are used to output the scores. These scores are subsequently used to implement any of the three score-level user-specific techniques mentioned in the previous section. Even with the cross-validated procedure, one is still very often left with *extremely few* user-specific client accesses (in the order of one to three) as compared to the user-specific impostor accesses (in the order of hundreds). Hence, in reality, implementing any of the three score-level user-specific techniques is non-trivial. Consequently, most of the reported results in the previous section rely on *more than three* user-specific client score accesses to implement, for instance as many as 10 in [35] and 5 in [14][1]. In [11], it was shown that the user-adapted technique requires *at least* 6 user-specific samples before the adaptation procedure achieves the performance comparable to that of the user-independent mean operator (the sum rule). One exception is F-norm [31], which was designed to work with as few as two score accesses on a single-modal BA system.

A similar work in this direction can be found in [10], whereby, a standard SVM was used in a somewhat novel way, i.e., an SVM was constructed using a user-independent set of scores plus a user-specific set of scores. Each of these sets of scores contain both client and impostor classes of scores. This strategy was called "adapted user-dependent fusion" by the author. This is to be distinguished from "user-independent fusion" whereby no user-specific data is used, or "user-dependent fusion" whereby only user-specific (client and impostor) scores are used (while ignoring the existence of user-independent client and impostor scores). The mentioned novelty in the said study is the use of the C parameter in SVM [36]. This parameter rates the *relative influence* of each example. When included in the support vectors (i.e., examples falling in the margin), the relatively high C parameters of these examples can change the decision boundary drastically. In [10], two C values are assigned to two sets of scores, i.e., one for the user-specific scores and one for the user-independent scores. In order for the adapted fusion to be effective a *greater* C value has to be associated to the precious user-specific

[1] Techniques that are *impostor centric* do not suffer from the lack of user-specific client data as they do not rely on this data but instead rely on user-specific impostor data.

scores as compared to the C value of the user-independent scores. It was demonstrated empirically that when C was tuned *a posteriori* on the test set (due to lack of available data for tuning the C parameter), the adapted fusion was potentially beneficial as compared to either user-independent or user-specific fusion. Since the additional free parameter C was tuned *a posteriori*, hence providing an additional degree of freedom to fit the data, the experimental results are thus *biased* towards the adapted fusion strategy.

Another similar idea using Bayesian adaptation (instead of using SVM) was reported by the same author in [11], also using the same multimodal database. The architecture employed is similar to the Gaussian Mixture Model (GMM) with Maximum *A Posteriori* (MAP) adaptation, the current state-of-the-art system in speaker verification [32]. However, a single Gaussian component with a diagonal covariance matrix was used[2]. According to our understanding, the justification for using a single Gaussian component is that there are just too few user-specific client scores to adapt (from two to three, depending on bootstrap samples). Similar to the C parameter in SVM, the GMM-MAP algorithm also has a free parameter called a "relevance factor". This factor is crucial in that it balances the right mix between the user-specific and user-independent information. In other words, both C and relevance factor play the same role in this context. Again, the relevance factor was tuned *a posteriori* and thus inevitably reporting *biased* performance towards the GMM-MAP algorithm. Ideally, any free parameter should be tuned on a separate validation set.

1.5 Contribution and Organization

In this paper, we are motivated by the fact that user-independent information is directly dependent on user-specific information, especially when the number of clients is small. However, as the number of clients increases, the user-independent information is no longer dependent on the user-specific information. In other words, when there are very few clients, any one of the clients may contribute to the user-independent information. However, if there are many clients, and all clients have equal probability of using a BA system, then the user-independent information cannot be dominated by the information specific to any of the clients.

Our approach is best explained and implemented using a Bayesian framework. This hence shares a similarity with [11]. However, different from the latter, the proposed approach does not suffer from the requirement that the scores have to be normally distributed. Thus the proposed method actually relaxes the Gaussian assumption. From a methodological point of view, although our approach also has a free parameter as in [10, 11], this parameter can be *automatically* tuned via standard algorithms, thus in a way, mitigating the small sample-size problem phenomenon. In comparison to [35], our approach does not require noise injection because the uncertainty about the scores are inherent in the Bayesian framework.

On top of unique advantages not shared by previously proposed techniques, this work contributes to the state-of-the-art in user-specific information in the following way:

- it provides an overview of techniques where user-specific information can be introduced in a BA system.

- it gives a Bayesian motivated justification for compensating between user-specific and user-independent information (such justification has not been somewhat emphasized previously).

- by incorporating some general prior knowledge about BA, the proposed Bayesian framework was turned into a practical algorithm and its performance was empirically measured.

- by extending the above-mentioned technique to a single modal biometric system, this technique can be considered a novel *user-adapted* score normalization procedure, which does not suffer from

[2] In the context of speaker verification, the use of GMM with a diagonal matrix per Gaussian component is fine since a full covariance matrix does not necessarily provide better performance. On the other hand, in the context of score-level fusion, a single Gaussian component with a full covariance matrix may be more appropriate, if the covariance information is *believed to be* valuable. Unfortunately, no comparative study was reported in this regard.

the traditional *impostor-centric* score normalization technique (Z-Norm), i.e., Z-Norm could degrade the performance with respect to the performance when not performing any normalization.

The rest of the paper is organized as follow: Section 2 describes the proposed technique in details; Section 3 presents the database and evaluation tools used, Section 4 presents the experimental results and this is followed by conclusions in Section 5.

2 Towards Compensating User-Independent and User-Specific Information

2.1 Preliminary

Suppose that each authorized person is identified by a unique identity claim $j \in \mathcal{J} \equiv \{1, \ldots, J\}$ and there are J identities. We sometimes call these users as clients to oppose a set of other unauthorized persons known as impostors. Hence, a biometric authentication system is aimed at distinguishing clients from impostors, which is an *aggregated* two-class problem, i.e., a two-class problem with J distinctive users. In this problem, it is common to represent a user by his/her feature template or *model*, i.e, a set of parameters derived from the features. Suppose that the output due to comparing a user model C_j to a feature X is $y(j)$. For each client model C_j, there is a corresponding impostor model I_j. Lacking a proper definition[3], the impostor model is often *naively* defined as the model of other finite clients $\forall_{j'} | j' \in \mathcal{J} - j$. To decide whether to accept or reject the access request represented by feature X claiming identity j, one can evaluate the *posterior probability ratio* in logarithmic domain (called log-posterior ratio, LPR):

$$\begin{aligned} \mathrm{LPR}_j &\equiv \log\left(\frac{P(C_j|X)}{P(I_j|X)}\right) = \log\left(\frac{P(X|C_j)P(C_j)}{P(X|I_j)P(I_j)}\right) \\ &= \underbrace{\log\frac{P(X|C_j)}{P(X|I_j)}}_{} + \underbrace{\log\frac{P(C_j)}{P(I_j)}}_{} \equiv y(j) - \Delta_j, \end{aligned} \tag{1}$$

where we introduced the term $y(j)$ (also called a Log-Likelihood Ratio, LLR) and a threshold Δ to handle the case of different priors. This constant also reflects the different *costs* of false acceptance and false rejection. In both cases, the threshold Δ has to be fixed *a priori*. The decision of accepting or rejecting an access is then:

$$\mathrm{decision}(\mathrm{LPR}_j) = \begin{cases} \text{accept} & \text{if } \mathrm{LPR}_j > 0 \\ \text{reject} & \text{otherwise.} \end{cases} \tag{2}$$

or

$$\mathrm{decision}(y(j)) = \begin{cases} \text{accept} & \text{if } y(j) > \Delta_j \\ \text{reject} & \text{otherwise.} \end{cases} \tag{3}$$

Although both forms are equivalent, the explicit presence of a threshold in the second decision function shows that the log-prior ratio can be adjusted *separately* from the score $y(j)$. Note that $y(j)$ is a direct function of X and the model associated to it (say θ_j), i.e., $y(j, X) = f_{\theta_j}(X)$. However, for simplicity, we write $y(j)$[4]. We use the function f with parameter θ to explicitly represent the *functional relationship* between $y(j)$ and X. Suppose that $y(j)$ is an instance of the variable $Y(j)$ and is drawn from the distribution $\mathcal{Y}(j)$. The decision function in Eqn. (3) then implies that $E_{\mathcal{Y}(j)|C_j}[Y(j)] >$

[3] Ideally, this impostor model should be the world population minus the user j. In terms of computation and data collection effort, this is not feasible and in practice not necessary.

[4] We are interested in the variability in a user-specific way. Hence, the functional relationship between $y(j)$ and X is not important in this study. Effectively, the variability due to X can be modeled as a random additive noise to otherwise deterministic score $y(j)$.

$E_{\mathcal{Y}(j)|I_j}[Y(j))]$, where $E_{\mathcal{Z}}[Z]$ is the expectation of Z under the law $\mathcal{Z}$. Typically, on a per client basis, one has an *extremely* few number of samples (depending on the protocols, there are 2 or 3 in the XM2VTS database [27], and 1 or 3 in the BANCA database [2]) to estimate the client distribution $\mathcal{Y}(j)|C_j$ whereas one has a relatively large number of samples (typically in the order of hundreds) to estimate the impostor distribution $\mathcal{Y}(j)|I_j$.

Although $y(j)$ is interpreted as an LLR here, many different machine-learning algorithms (e.g., Gaussian Mixture Models, Multi-Layer Perceptrons, Support Vector Machines) can be viewed as an approximation to this relationship, without necessarily giving it a probabilistic interpretation. This is to contrast with most Bayesian analyzes, e.g., [22, 20], that start by making an equivalence between $y(j)$ and $p(C_j|X)$, i.e., $y(j) \equiv p(C_j|X)$, such that the threshold $\Delta = 0.5$. Although this probabilistic interpretation is correct, it does not *explicitly* consider the fact that the threshold *changes* with the priors of client and impostor classes. In fact, the prior has already been integrated in $p(C_j|X) \propto p(X|C_j)p(C_j)$, which is the product between likelihood and the client prior. As a matter of fact, most biometric authentication systems crucially rely on this threshold to make the accept/reject decision. For instance, if the matching score $y(j)$ is based on a distance between a user template X_j and the submitted feature X, i.e., $y(j) \equiv dist(X, X_j)$, where $dist$ is a distance measure, then our Bayesian model above is still valid by interchanging between C_j and I_j, such that $E_{\mathcal{Y}(j)|C_j}[Y(j)] < E_{\mathcal{Y}(j)|I_j}[Y(j)]$. This distance measure simply cannot be interpreted in the probabilistic framework with $y(j) \equiv p(C_j|X)$.

Depending on the outcome of the decision (as a function of the threshold Δ_j), a biometric authentication system can commit two types of errors, namely, False Acceptances (FA) and False Rejections (FR). The error rates of FA and FR are defined as:

$$\begin{aligned}\mathrm{FAR}(\Delta_j) &= 1 - P(Y(j) \leq \Delta_j|I_j)\\ \mathrm{FRR}(\Delta_j) &= P(Y(j) \leq \Delta_j|C_j),\end{aligned}$$

where $P(Y(j) \leq \Delta_j|k_j)$ is the cumulative density function of the conditional variable $Y(j)$ within the range $[-\infty, \Delta_j]$ for each class k_j. Note that a unique point with Δ_j^* where $\mathrm{FAR}(\Delta_j^*) = \mathrm{FRR}(\Delta_j^*)$ is called Equal Error Rate (EER). EER is often used to characterize a system's performance. Another useful performance evaluation point for *any given threshold* Δ_j (not necessarily Δ_j^*) is called Half Total Error Rate (HTER) and is defined as the average of FAR and FRR, i.e.,:

$$\mathrm{HTER}(\Delta_j) = \frac{1}{2}(\mathrm{FAR}(\Delta_j) + \mathrm{FRR}(\Delta_j)).$$

The discussion until here concerns only a particular client indexed by j. In reality, one has extremely few examples of client accesses $y(j)|C_j$ and relatively large impostor accesses $y(j)|I_j$, as mentioned earlier. As a result, the estimation of the threshold Δ_j, i.e., user-specific threshold, is extremely unreliable. Hence, user-independent threshold is often used. This results in user-independent FAR, FRR and EER.

2.2 Compensating User-Specific Information with User-Independent Information

The analysis of LPR as in Eqn. (1) can be extended from feature to scores. Instead of considering the posterior probability $P(k_j|X)$, one can further consider $P(k_j|X, y(j), \theta_j)$, to emphasize the functional relationship $y(j) = f_{\theta_j}(X)$ and this relationship can be modeled by any appropriate machine-learning algorithm. For simplicity, one can write $P(k_j|y(j))$, instead. The corresponding LPR at the score-level of Eqn. (1) can be written as:

$$\mathrm{LPR}_j = \underbrace{\log\frac{P(y(j)|C_j)}{P(y(j)|I_j)}} + \underbrace{\log\frac{P(C_j)}{P(I_j)}}$$

Hence the first right-hand term corresponds to the score-level LLR whereas the second term corresponds to the same threshold as in Eqn. (1). At this point, it is useful to introduce an *aggregated score* y without the subscript j (due to not considering the fact the the scores $y(j)$ come from a particular client model j). Similarly, we use k instead of k_j to refer to the user-independent class label. The likelihood of the user-independent y is thus a result of accumulated likelihood of each of the user-dependent $y(j)$ for $j \in \mathcal{J}$, i.e.,

$$\begin{aligned} p(y|k) &\equiv \sum_{i \in \mathcal{J}} p(y|k, k_i) P(k_i) \\ &= \sum_{i \in \mathcal{J}-j} (p(y(i)|k, k_i) P(k_i)) + p(y(j)|k, k_j) P(k_j), \\ &= \sum_{i \in \mathcal{J}-j} (p(y(i)|k, k_i)) \left(1 - \frac{1}{J}\right) + p(y(j)|k, k_j) \left(\frac{1}{J}\right) \\ &\approx \sum_{i \in \mathcal{J}-j} (p(y(i)|k, k_i)) \text{ when } J \to \infty \end{aligned}$$

where $P(k_i)$ is the prior probably of client i using the system. Here, we assume that each client has equal probability of using the system so that $P(k_j) = P(k_i) = \frac{1}{J}$. We then separated the likelihood contribution of a particular client j from the rest of the clients. When the number of clients J is large, we see that $P(y|k)$ is independent of any specific client. Because of this phenomenon, we can model $p(y|k)$ by a mixture of user-independent (and hidden) (Gaussian) components which are independent of any client. Let the n-th component be denoted by c_n and there are N_c components. The user-independent likelihood can be estimated by:

$$p(y|k) \equiv \sum_{i=1}^{J} p(y|k, k_i) P(k_i) \approx \sum_{n}^{N_c} P(c_n) p(y|k, c_n) \tag{4}$$

where both $p(y|k, c_n)$ and $p(y|k, k_i)$ are each modeled by a Gaussian distribution. This phenomenon will be illustrated in Section 4.1 using real data.

The main idea of compensating the user-specific likelihood with user-independent likelihood is by mixing them into one of the two forms below:

$$p(y|k, k_j, w) = p(y|k)(1-w) + p(y(j)|k_j)w, \tag{5}$$

$$p(y|k, k_j, \gamma) = p(y|k)^{1-\gamma} p(y(j)|k_j)^{\gamma}, \tag{6}$$

where $\gamma, w \in [0, 1]$. The first form assumes that the two likelihoods are additive whereas the second form assumes that they are multiplicative. To emphasize the equivalence, we illustrate this in Figure 1. The top figure shows the likelihoods $P(y|k_j)$ of four different clients, $j \in \{1, 2, 3, 4\}$, together with the user-independent likelihood $P(y|k)$. The middle figure shows the likelihood $P(y(2)|k_2)^{\gamma} P(y|k)^{1-\gamma}$ for three γ values. The bottom figure shows the likelihood using the additive approach $P(y|k, k_j, r)$ as in Eqn. (5). For both the middle and bottom figures, the γ and r parameters were deliberately adjusted to give similar user-adapted likelihoods here.

Using the same formulation, the resultant multiplicative and additive LLR between $p(y|C, C_j)$ and $p(y|I, I_j)$ can be written as:

$$\begin{aligned} \text{LLR}_{mul} &\equiv \log \left\{ \left(\frac{P(y|C_j)}{P(y|I_j)} \right)^{\gamma} \left(\frac{P(y|C)}{P(y|I)} \right)^{1-\gamma} \right\} \\ &= \gamma \text{LLR}_j + (1-\gamma) \text{LLR}. \end{aligned} \tag{7}$$

and

$$\text{LLR}_{add} \equiv \log \left\{ \frac{p(y|C)(1-w) + p(y|C_j)w}{p(y|I)(1-w) + p(y|I_j)w} \right\}. \tag{8}$$

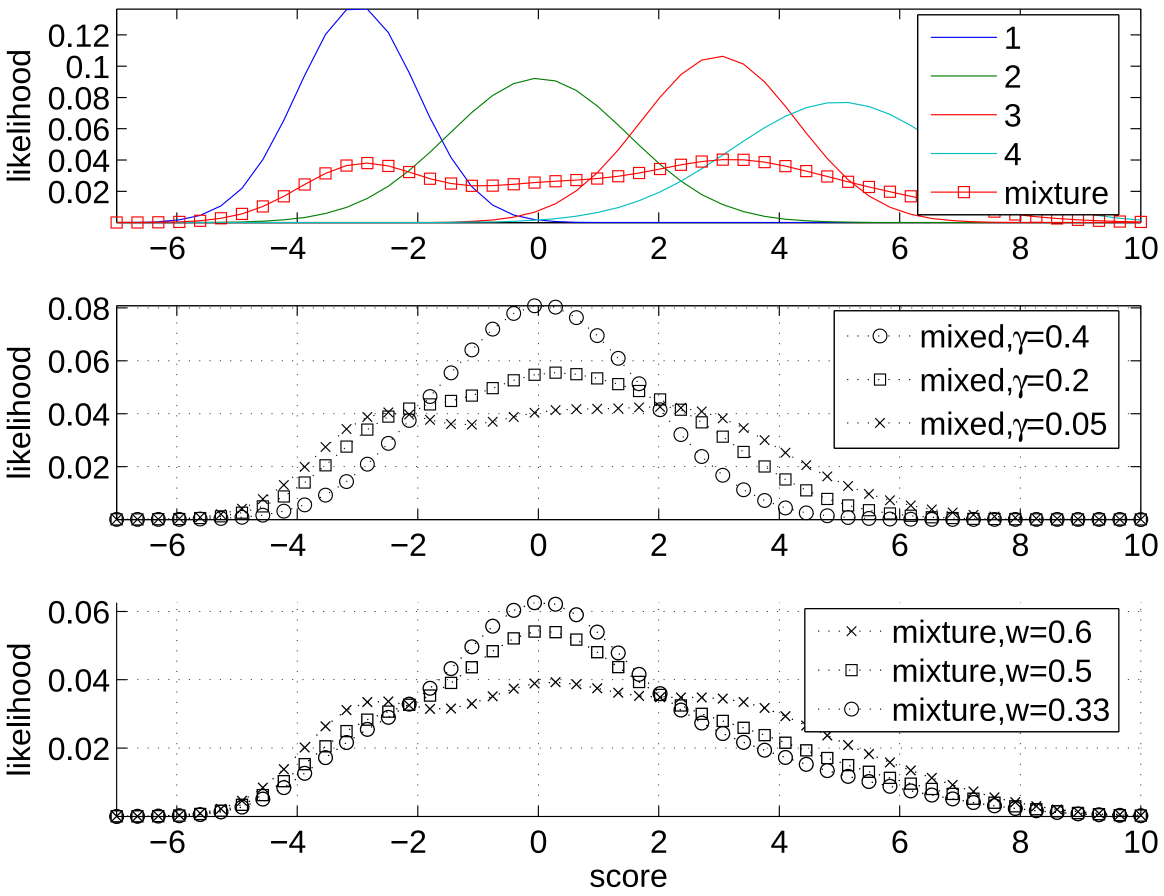
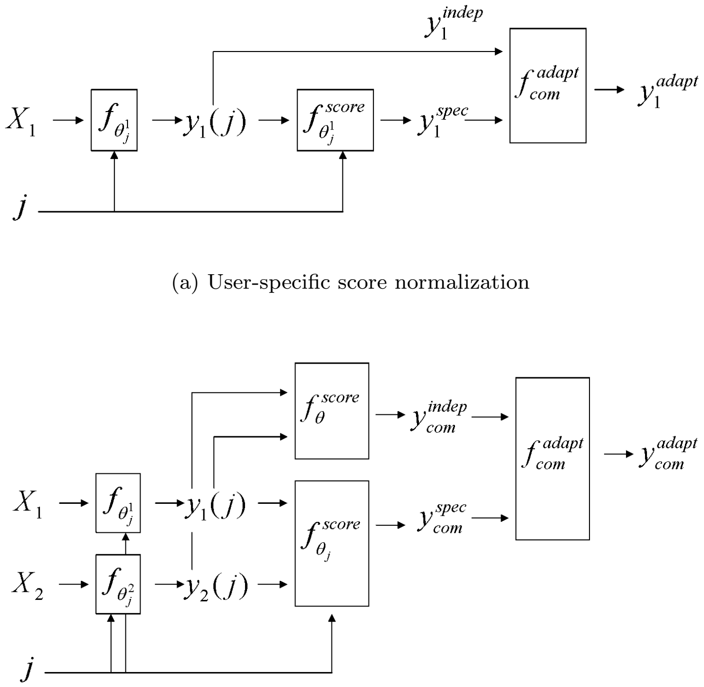


Figure 1: Top: The likelihood of four different clients and their user-independent mixture of (Gaussian) likelihood . Middle: The posterior likelihood of client 2 adapted using the idea of CD-posterior likelihood with different γ values. Bottom: The posterior-likelihood of client 2 obtained using the user-specific Gaussian mixture with different weight (prior) for client 2 (the rest of the weights for other clients are set to 1). All weights are then re-normalized to sum to one. Middle and bottom figures are adjusted such that the CD posterior likelihoods are proportional.

As can be seen, the former LLR can be decomposed whereas the latter cannot. The ability to decompose the two likelihoods in the former strategy is an advantage. This is because the user-specific LLR and user-independent LLR in Eqn. (7) can be estimated separately. Furthermore, the linear combination parameter γ can be estimated in a more elegantly, i.e., using standard algorithms, instead of using cross-validation. The rest of the discussion concerns the multiplicative strategy.

2.3 A Bigger Picture: Extending to Multiple Systems

Although the discussion in the previous section concerns a single system with a single output y or $y(j)$ (when considering the fact that the output is from client model j), the framework can easily be extended to cater to the case of a vector of scores, $\boldsymbol{y}(j) = [y_1(j), \ldots, y_N j]^T$, whose elements are each corresponding to a user-specific system and there are N such systems. The user-independent output vector $\boldsymbol{y}$ can be interpreted similarly. For the the single system case, this adaptation procedure can thus be considered a *user-adapted score normalization* procedure. For the multi-system case, the same adaptation procedure becomes a *user-adapted fusion* procedure. These two versions of the same



(a) User-specific score normalization

(b) User-specific Fusion

Figure 2: The architectures of (a) user-specific score normalization and (b) user-specific fusion

procedure are shown in Figure 2.

In each of these two settings, there are two important components: 1) user-specific and 2) users-independent components. By user-specific component, we mean that one has to create a function or module *for each user*. In contrast, only one global function or module is needed for the user-independent component. For the score normalization procedure (dealing with a single system; see Figure 2(a)), the user-specific component (denoted by the function $f_{\theta_j^1}^{score}$ takes $y_1(j)$ as input and outputs the score y_1^{spec} whereas the user-independent component outputs the score y_1^{indep}. In other words, y_1^{spec} models LLR$_j$ in Eqn. (7) whereas y_1^{indep} models LLR. Note that the user-independent component is a linear identity function which does not require any module. Then, both types of scores are then linearly combined via the function f_{com}^{adpat}. Figure 2(a) is thus a block diagram representation of Eqn. (7).

Similarly, Figure 2(b) is thus a block diagram representation of Eqn. (7) but in the context of fusion, where two or more scores of different systems are combined Its user-specific component takes in the vector $\boldsymbol{y}(j)$ and outputs a user-specific combined score. This score models the user-specific combined LLR in Eqn. (7). Note that one such fusion function has to be created for each client. The user-independent component (created only once and applied to all clients) also takes in the vector $\boldsymbol{y}(j)$ but outputs a user-independent combined score via the fusion function $f_{\theta_j}^{score}$. Both types of score are then linearly combined to obtain a user-adapted combined score.

We will first treat the *fusion procedure*, by handling its user-independent component in Section 2.4 and its user-specific in Section 2.5. Then for the *normalization procedure*, its user-specific component will be discussed in Section 2.6. The user-independent component counterpart is a linear identity function and thus does not require any processing. Finally, to combine both sources of information, linear procedures such as those to be discussed in Section 2.4 can be reused without modification.

2.4 User-Independent Fusion

Recall that an LLR is made up of two conditional likelihood functions, i.e., $\log\{p(\boldsymbol{y}|C)/p(\boldsymbol{y}|I)\}$. Each of these likelihoods can be modeled using a Gaussian Mixture Model (GMM), such that:

$$\hat{p}(\boldsymbol{y}|k) = \sum_{i}^{N_c} w_i^k \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_i^k, \boldsymbol{\Sigma}_i^k), \tag{9}$$

where, the i-th component of the class conditional (denoted by k) mean vector is $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_N]^T$ and its covariance matrix is $\boldsymbol{\Sigma}_i^k$ for each of the class $k \in \{C, I\}$. These parameters can be calculated using standard Expectation-Maximization algorithm [5, Chap. 2] since there are enough data in each category to do so. Another solution is to directly model the function $\text{LLR} = \log\{p(\boldsymbol{y}|C)/p(\boldsymbol{y}|I)\}$ by:

$$\hat{\text{LLR}}_j = \boldsymbol{w}^T \boldsymbol{y}. \tag{10}$$

As can be seen, there exists many standard algorithms to estimate LLR, such as logistic regression (often used in the statistics) [17], the perceptron algorithm or even its more advanced non-linear counterpart, Multi-Layer Perceptron (MLP) [5] or the Support Vector Machine (SVM) algorithm [36]. Strictly speaking, these algorithms *do not* produce LLR but in our context, they can be *viewed* as an approximation to LLR, i.e., their outputs correlate well with LLR.

2.5 Parameter Estimation of User-Specific LLR

Approximating user-specific LLR is more difficult than approximating user-independent LLR since few user-specific data points are available, especially the client scores (less serious for the impostor scores). We tackle this problem using the following rules-of-thumb: 1) using simple classifier model (with low degree of freedom); 2) using robust parameter estimation; and 3) relying on some prior knowledge such as user-independent distribution.

Because of few user-specific data points, the best that one can do is to assume that each class of user-specific scores is normally distributed. The first rule implies that using more than one Gaussian components as in the user-dependent case will probably result in overfitting.

Let us denote the class-conditional mean and covariance as $\boldsymbol{\mu}_j^k$ and $\boldsymbol{\Sigma}_j^k$, respectively, for $k = \{C, I\}$ and for client j. The user-independent parameters, $\boldsymbol{\mu}^k$ and $\boldsymbol{\Sigma}^k$, are defined similarly. The estimation of user-specific parameters $\boldsymbol{\mu}_j^k$ and $\boldsymbol{\Sigma}_j^k$ are discussed in the Appendix. The third rule can be realized via the following adaptation:

$$\boldsymbol{\mu}_{adapt,j}^k = \boldsymbol{\mu}_j^k \gamma_1^k + \boldsymbol{\mu}^k (1 - \gamma_1^k) \tag{11}$$

Similarly, the adapted covariance is defined as:

$$\boldsymbol{\Sigma}_{adapt,j}^k = \boldsymbol{\Sigma}_j^k \gamma_2^k + \boldsymbol{\Sigma}^k (1 - \gamma_2^k) \tag{12}$$

Both parameters γ_1^k and γ_2^k (for the first and second moments) are within the range $[0, 1]$. These form of adaptation is can be found in [15] and is called Maximum A Posteriori (MAP) adaptation by the authors. They balance between the user-specific estimate and the user-independent estimate of the two Gaussian parameters.

At first sight, having four free parameters γ_1^k, γ_2^k for $k \in \{C, I\}$ are too many if one considers that there are about a hundred user-specific impostor scores and about two user-specific client scores. In this case, using the first rule-of-thumb, one can fix these values *a priori*, i.e., $\gamma_1^I = 1$, $\gamma_2^I = 1$ (putting full confidence on the user-specific impostor estimates), $\gamma_1^C = 1$ and $\gamma_2^C = 0$ (putting zero confidence on the user-specific client covariance estimate since it is non-informative; see the Appendix). Hence, one is left with a single parameter $\gamma_1^C \in [0,1]$ to tune. The issue to tuning or not γ_1^C will be discussed after observing the real data by experimentation (see Section 4.2).

One can immediately recognize that this class of solution is called Quadratic Discriminant Analysis (QDA) when $\boldsymbol{\mu}^C_{adapt,j} \neq \boldsymbol{\mu}^I_{adapt,j}$ and Linear Discriminant Analysis (LDA) when $\boldsymbol{\Sigma}^C_{adapt,j} = \boldsymbol{\Sigma}^I_{adapt,j}$. Again, considering few user-specific client scores, we can derive the LDA solution from the QDA solution by enforcing the common covariance matrix to be that of impostor only (since the user-specific client covariance is not informative).

The formulations in Eqns. (11 and 12) are very similar to what is called "the relevance factor" mentioned in [11] or [32], whereby the parameter γ_i^k for all $i \in \{1,2\}$ and $k \in \{C,I\}$ is defined to be:

$$\gamma_i^k \equiv \frac{N^k(j)}{N^k(j)+r}, \tag{13}$$

where $N^k(j)$ is the number of user-specific training example (of client j) belonging to class k. The parameter r is the so-called *relevance factor* and it takes only positive values. In biometric authentication, where $N^I(j) >> N^C(j)$, r will give more weight to user-specific impostor information (or more precisely the two Gaussian parameters) whereas less weight to the user-specific client information. It should be recognized that relevance factor is also a form of constraint. Otherwise, a different r for each k or for each i would have meant that one has to still to tune the four parameters. In our case, we fixed these parameters *a priori* to further constrain the model fitting. Hence, these two forms are just some possible constraints and relevance factor is found to be particularly useful in speaker verification [32] in the context of mixture of Gaussians, instead of the single-component case as done here.

2.6 User-Specific Score Normalization

Using the formulation shown in Figure 2(a) of Section 2.3, we will describe here a particular implementation of user-specific score normalization. This implementation happens to be the well-known user-specific Z normalization.

Suppose that $p(y|k_j)$ is Gaussian distributed with mean μ^k and standard deviation σ_j^k for each class label $k \in \{C,I\}$. Let us further assume that both the conditional score distributions share the *same* standard deviation σ_j^I because the standard deviation of the user-specific client distribution is not reliable (due to few samples). The user-specific LLR, as in the first right-hand side term of Eqn. (7), can then be written as:

$$\begin{aligned}
\mathrm{LLR}_j &\equiv \log \frac{p(y|C_j)}{p(y|I_j)} \\
&= -\frac{1}{2(\sigma_j^I)^2}\left((y-\mu_j^C)^2 - (y-\mu_j^I)^2\right) \\
&\quad - \log \frac{\sqrt{2\pi(\sigma_j^I)^2}}{\sqrt{2\pi(\sigma_j^I)^2}} \\
&= \left(\frac{\mu_j^C - \mu_j^I}{(\sigma_j^I)^2}\right)\left(y - \frac{\mu_j^C + \mu_j^I}{2}\right),
\end{aligned} \tag{14}$$

after rearrangement. If one further assumes that $\mu_j^C = y$ because μ_j^C is unknown, then Eqn. (14) can further be written as:

$$\mathrm{LLR}_j = \frac{(y-\mu_j^I)^2}{2(\sigma_j^I)^2}$$

This form of equation is known as Z normalization. In practice, the following form is used:

$$f_j^z(y) = \frac{y - \mu_j^I}{\sigma_j^I} \tag{15}$$

Both forms are equivalent since $f_j^z(y) \propto \mathrm{LLR}_j$. In this way, it can be seen that Z-norm can be readily used to implement the user-specific score normalization. It should be noted that Z-norm is *impostor centric* due to the fact that the user-specific client distributions, as represented by μ_j^C and σ_j^C are not used at all. A more detailed discussion on this form of normalization is found in [19]. Another user-specific score normalization that is *client-impostor* centric can be found in [31].

2.7 Limitation and Solution to Gaussian Assumption

It should be warned that the user-specific adaptation mentioned in Sections 2.5–2.6 is founded on the assumption that scores can be modeled by a (multi-variate) Gaussian. Consider the case where the base-classifier output is an MLP which outputs posterior probability (within the range $[0, 1]$) typically due to using a logistic activation function or outputs scores within the range $[-1, 1]$ typically due to using hyperbolic tangent activation function. Then, one knows that the scores *cannot be adequately* modeled using Gaussian distributions because simply they are indeed *not* normally distributed. Using 1186 experiments, an empirical study in [29], for instance showed that the divergence from the Gaussian assumption of MLP outputs, when trained using the above activation functions, is much larger than that of GMM and SVM. Hence, ideally, one should *always* use the output just before applying any one of the two squashing functions mentioned. However, when this is not possible, (for instance, due to using a commercial-off-the-shelf product), reversing this process is possible, to a certain extent. The usual definition of sigmoid and tangent hyperbolic are:

$$\begin{aligned} \mathrm{sigmoid}(z) &= \frac{1}{1 + \exp(-z)}, \\ \tanh(z) &= \frac{\sinh(z)}{\cosh(z)}, \end{aligned}$$

respectively. If y is an output of a sigmoid or a hyperbolic tangent function, its inverse is:

$$\mathrm{sigmoid}^{-1}(y) = -\log\left(\frac{1}{y} - 1\right),$$
$$\tanh^{-1}(y) = \frac{1}{2}\log\left(\frac{1+y}{1-y}\right),$$

respectively. The above transformations are linear around the mid-range and non-linear towards the range limit. As a result, when y is near the range limits, its inversion will be limited to the machine precision represented by the value y. Hence, such remedial procedure cannot "reverse" perfectly the process. Fortunately, we now know that data points near the range limits cannot influence the decision function. On the other hand, only those data points near the decision boundary can influence the decision function [36]. This is because these data points are the ones found within the "margin" described in [36].

3 Database and Evaluation

3.1 XM2VTS Database and Its Fusion Benchmark

The XM2VTS database [27] contains synchronized video and speech data from 295 subjects, recorded during four sessions taken at one month intervals. On each session, two recordings were made, each consisting of a speech shot and a head shot. The speech shot consisted of frontal face and speech

Table 1: The Lausanne Protocols as well as the fusion protocol of XM2VTS database.

Data sets	Number of accesses		Fusion Protocols
	LP1	LP2	
LP Train client	3	4	NIL
LP Eval client	600 (3×200)	400 (2×200)	Fusion dev
LP Eval impostor	40,000 ($25 \times 8 \times 200$)		Fusion dev
LP Test client	400 (2×200)		Fusion eva
LP Test impostor	112,000† ($70 \times 8 \times 200$)		Fusion eva

†: Due to one corrupted speech file of one of the 70 impostors in this set, this file was deleted, resulting in 200 less of impostor scores, or a total of 111,800 impostor scores.

recordings of each subject during the recital of a sentence. The database is divided into three sets: a training set, an evaluation set and a test set. The training set (LP Train) was used to build client models, while the evaluation set (LP Eval) was used to compute the decision thresholds (as well as other hyper-parameters) used by classifiers. Finally, the test set (LP Test) was used to estimate the performance.

The 295 subjects were divided into a set of 200 clients, 25 evaluation impostors and 70 test impostors. There exists two configurations or two different partitioning approaches of the training and evaluation sets. They are called Lausanne Protocol I and II, denoted as **LP1** and **LP2** in this paper. In both configurations, the test set remains the same. Their difference is that there are three training shots per client for LP1 and four training shots per client for LP2. Table 1 is the summary of the data. More details can be found in [24]. The first column shows the data set, divided into training, evaluation and test sets. Columns two and three show the the partition of the data according to LP1 and LP2 whereas column four shows the partition of data for the fusion protocols that are *consistent* with the Lausanne Protocols. As far as fusion is concerned, there are only two data sets, labeled as “Fusion dev” (for development) and “Fusion eva” (for evaluation), since the data used in LP training sets are reserved to construct the base systems. Note that the fusion development set is used to calculate the parameters of fusion classifier as well as the optimal global threshold. They are then applied to the fusion evaluation set. Since the threshold is *calculated from the development set*, the reported performance obtained from the evaluation set is thus called an *a priori* performance. The performance measure is described in the next section.

3.2 Evaluation Using Pooled EPC Curve

Perhaps the most commonly used performance visualizing tool in the literature is the Decision Error Trade-off (DET) curve [26]. It has been pointed out [4] that two DET curves resulted from two systems are not comparable because such comparison does not take into account how the thresholds are selected. It was argued [4] that such threshold should be chosen *a priori* as well, based on a given criterion. This is because when a biometric system is operational, the threshold parameter has to be fixed *a priori*. As a result, the Expected Performance Curve (EPC) [4] was proposed. We will adopt this evaluation method, which is also in coherence with the original Lausanne Protocols defined for the XM2VTS database. The criterion to choose an optimal threshold is called weighted error rate (WER), defined as follows:

$$\mathrm{WER}(\alpha, \Delta) = \alpha \mathrm{FAR}(\Delta) + (1 - \alpha)\ \mathrm{FRR}(\Delta), \tag{16}$$

where FAR and FRR are False Acceptance Rate and False Rejection Rate, respectively. Note that WER is optimized for a given $\alpha \in [0, 1]$. Let Δ^*_α be the threshold that *minimizes* WER on a *development set* for a given α. The performance measured on an *evaluation set* at a given Δ^*_α is called

Half Total Error Rate (HTER), which is defined as:

$$\mathrm{HTER}(\alpha) = \frac{\mathrm{FAR}(\Delta^*_\alpha) + \mathrm{FRR}(\Delta^*_\alpha)}{2}. \tag{17}$$

The EPC curve simply plots HTER versus α, since different values of α give rise to different values of HTER. The EPC curve can be interpreted in the same manner as the DET curve, i.e., the lower the curve is, the better the performance but for the EPC curve, the comparison is done at a given cost (controlled by α). One advantage of EPC curve is that it can plot a pooled curve from several experiments. For instance, to compare two methods over M experiments, only one pooled curve is necessary. This is done by calculating HTER at a given α point by taking into account all the false acceptance and false rejection accesses over all M experiments. The pooled FAR and FRR across $j = 1, \ldots, M$ experiments for a given $\alpha \in [0, 1]$ is defined as follow:

$$\mathrm{FAR}^{pooled}(\alpha) = \frac{\sum_{j=1}^{M} \mathrm{FA}(\Delta^*_\alpha(j))}{NI \times M}, \tag{18}$$

and

$$\mathrm{FRR}^{pooled}(\alpha) = \frac{\sum_{j=1}^{M} \mathrm{FR}(\Delta^*_\alpha(j))}{NC \times M}, \tag{19}$$

where $\Delta^*_\alpha(j)$ is the optimized threshold at a given α, NI is the number of impostor accesses and NC is the number of client accesses. FA and FR count the number of false acceptance and the number of false rejection at a given threshold $\Delta^*_\alpha(j)$. The pooled HTER is defined similarly as in Eqn. (17).

4 Experiments

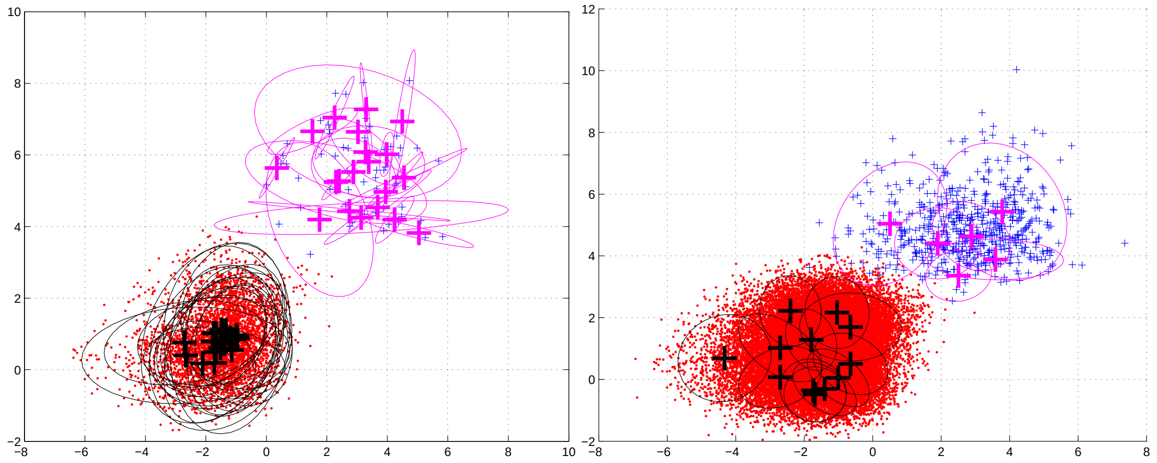
4.1 Preliminary Experiments on User-Independence Assumption

The aim of this experiment is to answer the question: "Is user-independent LLR *really* independent of the LLR of any specific client?". In other words, we would like to verify Eqn. (4). This is clearly not true because the user-independent conditional likelihood is *functionally* dependent on the user-dependent conditional likelihood. Figure 3 illustrates that the number of Gaussian components in $P(y|k)$ when estimated using a GMM model, does not need to be equivalent to the number of clients. In fact, much less number of components are needed, e.g., in this case, 200 clients versus 12 impostor components and 6 client components.

4.2 Preliminary Experiment on Tuning Parameter γ_1^C

This experiment verifies the so-called "prior knowledge" mentioned in Section 2.5, that constrains the parameters γ_i^k for $i = \{1, 2\}$ and $k = \{C, I\}$, as discussed in Section 2.5. By the nature of the problem, we fix $\gamma_i^I = 1$ for $i = \{1, 2\}$, hence making full confidence on user-specific impostor Gaussian parameters. We fix $\gamma_2^C = 1$ because the user-specific covariance is likely to be non-informative. Hence, this left us with one tunable parameter, γ_1^C.

As a preliminary experiment, we used two settings: 1) $\gamma_1^C = 1$ and $\gamma_2^C = 1$, and 2) $\gamma_1^C = 1$ and $\gamma_2^C = 0$. For both settings, the impostor $\gamma_i^I|\forall_i$'s are set to one. These two settings are applied to one of the 21 fusion tasks mentioned in Section 3.1. The results for setting 1 are shown in Figure 4. The pair of user-independent and user-specific LLRs representing an access, (LLR and LLR$_j$), are plotted in Figure 4(a) for the development (training) and in Figure 4(b) for the evaluation (test) set. As can be observed, with $\gamma_2^C = 1$, the user-specific LLR overfits the training set – the separation between client and impostor accesses are *perfect* but when applied to the test set, the user-specific LLR performs much worse. If this setting was to be used, the second adaptation classifier (f_{adapt} in Figure 2) would never generalize.



(a) User-dependent

(b) User-independent

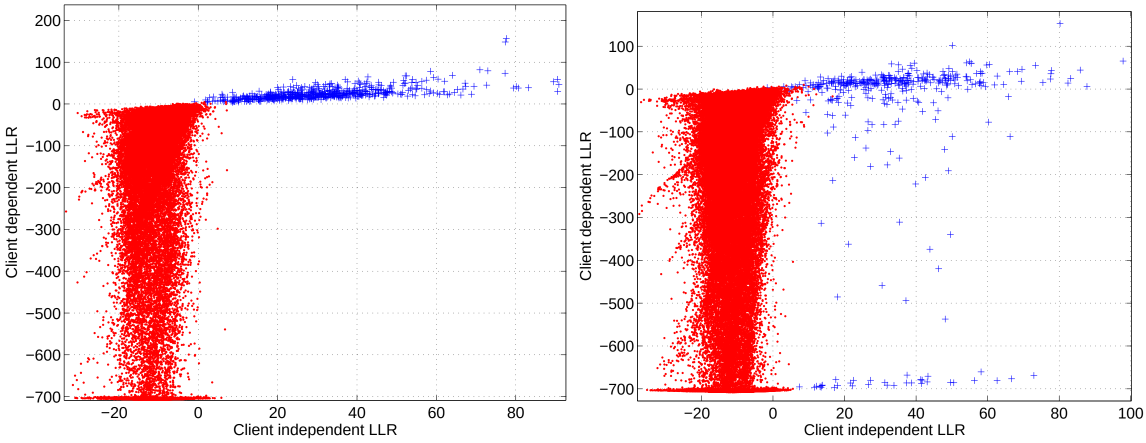
Figure 3: (a) User-dependent and (b) user-independent Gaussian fits, whose centers are represented by a big plus sign. For both figures, the X- and Y-axes are the output score-space of a face and speech experts, respectively. The upper right clusters are client accesses whereas the lower left clusters are impostor accesses. In (a), only 20 class-conditional Gaussian components are fitted on each of class of scores belonging to each of 20 client models. In (b), the client and impostor accesses of all 200 clients are plotted directly on the mixture of Gaussian components. Plotting all 200 clients in (a) will be too cluttered. Obviously, in (b), less Gaussian components are needed. In this case, only 12 Gaussian components are needed to represent the (user-independent) impostor accesses and 6 are needed for the client accesses, despite the fact that these accesses are from 200 clients and their corresponding impostor accesses. The number of Gaussian components are optimized using standard Expectation Maximization algorithm such that the conditional likelihood of the corresponding validation set is maximized.

This indicates that using an *unreliable* user-specific covariance for the client accesses can result in overfitting. For this experiments, there are actually three user-specific client accesses and two multimodal experts: one is a face and the other is a speech expert. In the second setting where $\gamma_2 = 0$, both the scatter plot of development and evaluation data sets are similar to Figure 4(b), implying no overfitting. Hence, $\gamma_2 = 0$ will be used through out the experiments, whereas γ_1 will be tuned by cross-validation.

4.3 Pooled Fusion Experiments

In the previous sections, preliminary experiments are conducted in order to understand the characteristics of the problems at hand. In this and the next section, generalization performance will be objectively measured in terms of HTER. The experimental results of user-specific *fusion* procedure are reported here whereas those of user-specific *normalization* procedure are reported in Section 4.4. The fusion experiments are further divided into three categories: multimodal fusion (21 experiments), face intramodal fusion (7 experiments) and speech intramodal fusion (4 experiments). They are shown in Figure 5.

Having run all the experiments, we checked the tuned γ_1^C values and found that most values are exactly 1 and the smallest value is 0.7.. As a control, we indeed repeated the *same* set of experiments but fixing $\gamma_1^C = 1$. Similar results are obtained as in Figure 5. The probabilistic inversion procedure as discussed in Section 2.7 was employed to estimate the user-specific fusion function using QDA



(a) development set

(b) evaluation set

Figure 4: User-dependent LLR versus user-independent LLR using (a) a development set and (b) an evaluation set. The client accesses are plotted with plus signs whereas the impostor accesses are plotted with dots. As can be seen, there is a clear sign of overfitting for the user-dependent LLR when $\gamma_2 = 1$ and γ_1 is optimized on the data set by validation. For most experiments, it was found that γ_1 is *always* close to 1.

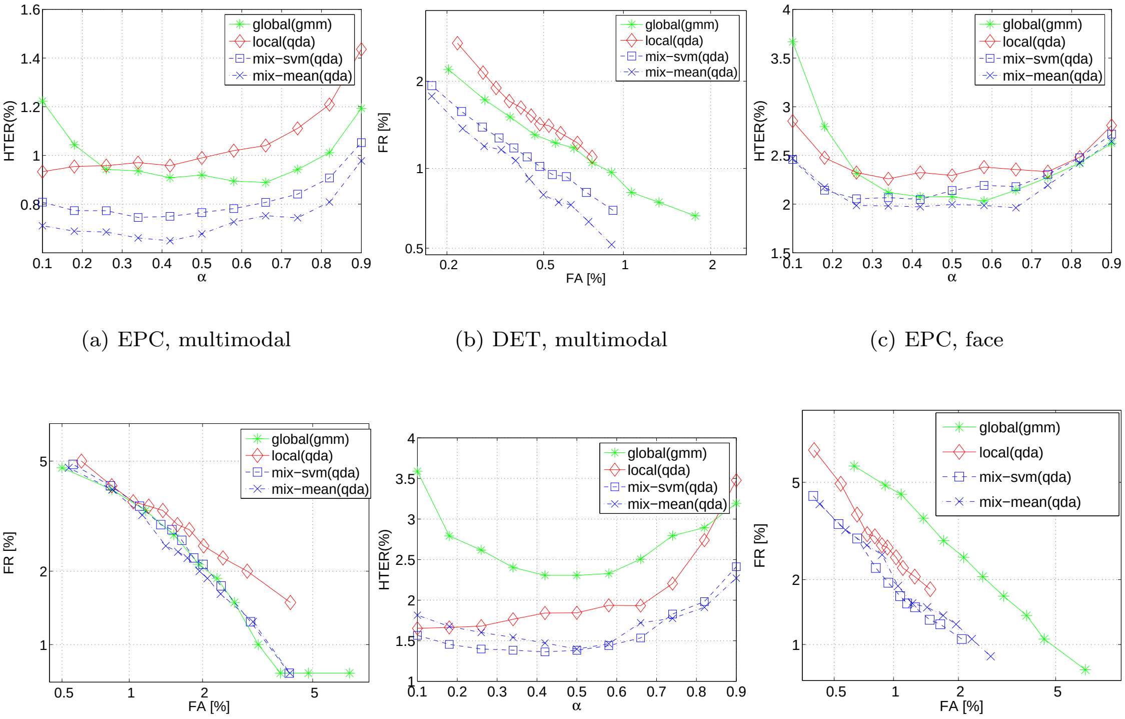
whenever the output of an expert is an MLP using either of the activation functions as the final output layer. As another set of control experiments, we also used carried out the same experiments, but this time, *not using* the probabilistic inversion procedure (not shown here). The results are that for the multimodal and speech intramodal settings, the improvement was reduced. As for the face intramodal experiments, no improvement was observed. Therefore, we can conclude that the probabilistic inversion procedure is an important factor to guarantee the successful use of QDA-based user-specific fusion.

4.4 Pooled User-Specific Score Normalization Experiments

In this section all the 7 face and 6 speech baseline systems are used to test the user-adapted normalization procedure proposed in Section 2.3 and shown in Figure 2(a). The results are shown in Figure 6. As can be seen from the face experiments, directly applying user-specific Z-norm will only worsen the generalization performance. However, the proposed user-adapted normalization procedure does not suffer from this drawback because it relies on both sources of information. On the other hand, for the speech systems, user-specific Z-norm improves the performance significantly. Furthermore, the user-adapted normalization procedure even boosts the generalization performance further. Based on these 13 experiments, we can conclude that the user-adapted score-normalization procedure takes the best of both user-specific and user-independent information.

5 Conclusions

In this study, we proposed to categorize different approaches to using user-specific information in a general biometric authentication framework. Although the review in this domain is not exhaustive, we believe that it represents the state-of-the-art as long as the use of user-specific information in this application is concerned. In particular, this study generalizes the paradigm proposed by [35], whereby two user-specific techniques are combined: user-specific fusion and user-specific threshold.



(a) EPC, multimodal

(b) DET, multimodal

(c) EPC, face

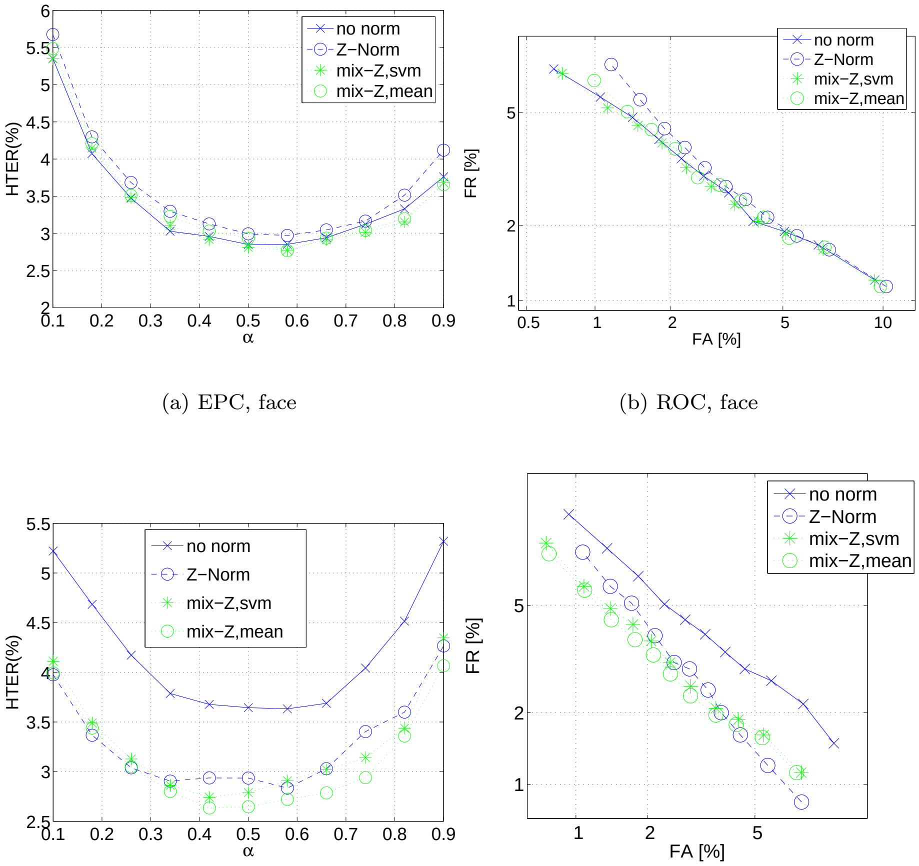
(d) DET, face

(e) EPC, speech

(f) DET, speech

Figure 5: Fusion experiments: Comparison of fusion using user-independent information (denoted as "global", implemented by using a GMM), user-specific information (denoted as "local", implemented using user-specific QDA), and both information sources (denoted as "mix"), also called adapted strategy. The adapted strategy was implemented using two methods: mean and SVM, applied to multimodal datasets (a) and (b) (evaluated over 21 experiments), intramodal face datasets (c) and (d) (over 7 experiments) and intramodal speech datasets (e) and (f) (over 4 experiments), each pair depicting EPC and DET curves, respectively. The γ_1 of each experiment was tuned on a validation set. A similar set of experiments are repeated but using $\gamma_1 = 1$ gives the similar results as in (a)–(f) and thus not shown here.

Although the idea of compensating user-specific information with user-independent information is not new [10, 11, 32], we proposed a novel way to combining both information sources, founded on our interpretation of the Bayesian framework. We showed that user-specific information can be used at the fusion level (resulting in a user-adapted fusion strategy) and at the score level (resulting in a user-adapted normalization procedure). The challenge to devising a useful user-specific/user-adapted procedure is to make use of the *extremely few* user-specific client features/scores for training. One major conclusion from both experiments is that not only that balancing both sources of information *does not deteriorate* the performance (especially when relying solely on user-specific information can deteriorate the performance), it can also improve the generalization performance compared to using either one information source. Finally, in view of limited user-specific client scores, prior knowledge is definitely important in order to build a user-specific fusion classifier. Empirical results show that indeed the free parameters involved could be and should be fixed *a priori* for best generalization performance. In our case, the proposed user-specific fusion classifier can be trained *without fixing* any free parameter (the four γ_i^k parameters, for $i \in \{1, 2\}$ and $k \in \{C, I\}$). Furthermore, the adaptation



(a) EPC, face

(b) ROC, face

(c) EPC, speech

(d) ROC, speech

Figure 6: Score normalization experiments: Comparison of user-specific Z-norm, user-independent score (no normalization) and mixing both information sources using mean and SVM, applied to each of the face datasets (a) and (b), and speech data sets (c) and (d). The EPC and DET curves are pooled over 7 face experiments and 6 speech experiments.

classifier that combines both the user-specific (approximated) LLR and user-independent (approximated) LLR can be trained (as in the case of using SVM) or without training (as is the case of using the mean operator) without the *risk of overfitting*. In comparison to the previous works [10, 11], this is an important advantage. This is because due to few user-specific samples, cross validation can be very unreliable. This advantage compares favorably with the work proposed in [35] whereby due to the same problem, noise is injected to increase the number of user-specific client scores. Our proposed approach handles this via the standard Bayesian framework that includes such uncertainty in a natural way.

Acknowledgement

The authors thank Sebastien Marcel and Fabien Cardinaux for providing raw score data sets. This work was supported in part by the IST Program of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, funded in part by the Swiss Federal Office for Education and Science (OFES) and the Swiss NSF through the NCCR on IM2. This publication only reflects the authors' view.

References

[1] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score Normalization for Text-Independant Speaker Verification Systems. *Digital Signal Processing (DSP) Journal*, 10:42–54, 2000.

[2] E. Bailly-Baillière, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Porée, B. Ruiz, and J.-P. Thiran. The BANCA Database and Evaluation Protocol. In *Springer LNCS-2688, 4th Int. Conf. Audio- and Video-Based Biometric Person Authentication, AVBPA'03*. Springer-Verlag, 2003.

[3] M. Ben, R. Blouet, and F. Bimbot. A Monte-Carlo Method For Score Normalization in Automatic Speaker Verification Using Kullback-Leibler Distances. In *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 689–692, Orlando, 2002.

[4] S. Bengio and J. Mariéthoz. The Expected Performance Curve: a New Assessment Measure for Person Authentication. In *The Speaker and Language Recognition Workshop (Odyssey)*, pages 279–284, Toledo, 2004.

[5] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1999.

[6] R. Brunelli and D. Falavigna. Personal Identification Using Multiple Cues. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(10):955–966, 1995.

[7] K. Chen. Towards Better Making a Decision in Speaker Verification. *Pattern Recognition*, 36(2):329–346, 2003.

[8] A. Cohen and Y. Zigel. On Feature Selection for Speaker Verification. In *Proc. COST 275 workshop on The Advent of Biometrics on the Internet*, pages 89–92, Rome, November 2002.

[9] Fabien Cardinaux, Conrad Sanderson, and Samy Bengio. User Authentication via Adapted Statistical Models of Face Images. IDIAP-RR 38, IDIAP, 2004. accepted for publication in *IEEE Trans. Signal Processing*, 2005.

[10] J. Fierrez-Aguilar, D. Garcia-Romero, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Exploiting General Knowledge in User-Dependent Fusion Strategies For Multimodal Biometric Verification. In *IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 617–620, Montreal, 2004.

[11] J. Fierrez-Aguilar, D. Garcia-Romero, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Bayesian Adaptation for User-Dependent Multimodal Biometric Authentication. *Pattern Recognition*, 38:1317–1319, 2005.

[12] J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Target Dependent Score Normalisation Techniques and Their Application to Signature Verification. In *Springer LNCS-3072, Int'l Conf. on Biometric Authentication (ICBA)*, pages 498–504, Hong Kong, 2004.

[13] S. Furui. Cepstral Analysis for Automatic Speaker Verification. *IEEE Trans. Acoustic, Speech and Audio Processing / IEEE Trans. on Signal Processing*, 29(2):254–272, 1981.

[14] D. Garcia-Romero, J. Gonzalez-Rodriguez, J. Fierrez-Aguilar, and J. Ortega-Garcia. U-Norm Likelihood Normalisation in PIN-Based Speaker Verification Systems. In *Springer LNCS-2688, 4th Int'l. Conf. Audio- and Video-Based Biometric Person Authentication (AVBPA 2003)*, pages 208–213, Guildford, 2003.

[15] J.L. Gauvain and C.-H. Lee. Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Obervation of Markov Chains. *IEEE Tran. Speech Audio Processing*, 2:290–298, 1994.

[16] G. Gravier, J. Kharroubi, and G. Chollet. On the Use of Prior Knowledge in Normalization Schemes for Speaker Verification. *Digital Signal Processing (DSP) Journal*, 10:213–225, 2000.

[17] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[18] A. Jain and A. Ross. Learning User-Specific Parameters in Multibiometric System. In *Proc. Int'l Conf. of Image Processing (ICIP 2002)*, pages 57–70, New York, 2002.

[19] Johnny Mariéthoz and Samy Bengio. A Bayesian Framework for Score Normalization Techniques Applied to Text Independent Speaker Verification. IDIAP-RR 62, IDIAP, 2004. to appear in Signal Processing Letters.

[20] J. Kittler, M. Hatef, R. P.W. Duin, and J. Matas. On Combining Classifiers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[21] A. Kumar and D. Zhang. Integrating Palmprint with Face for User Authentication. In *Workshop on Multimodal User Authentication (MMUA 2003)*, pages 107–112, Santa Barbara, 2003.

[22] L.I. Kuncheva. A Theoretical Study on Six Classifier Fusion Strategies. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(2):281–286, February 2002.

[23] J. Lindberg, J.W. Koolwaaij, H.-P. Hutter, D. Genoud, M. Blomberg, J.-B. Pierrot, and F. Bimbot. Techniques for a priori Decision Threshold Estimation in Speaker Verification. In *Proc. of the Workshop R̈econnaissance du Locuteur et ses Applications Commerciales et Criminalistiques(RLA2C)*, pages 89–92, Avignon, 1998.

[24] J. Lüttin. Evaluation Protocol for the XM2FDB Database (Lausanne Protocol). Communication 98-05, IDIAP, Martigny, Switzerland, 1998.

[25] M.W. Mak. A Two-Level Fusion Approach to Multimodal Biometric Verification. In *IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 485–488, Philadelphia, 2005.

[26] A. Martin, G. Doddington, T. Kamm, M. Ordowsk, and M. Przybocki. The DET Curve in Assessment of Detection Task Performance. In *Proc. Eurospeech'97*, pages 1895–1898, Rhodes, 1997.

[27] J. Matas, M. Hamouz, K. Jonsson, J. Kittler, Y. Li, C. Kotropoulos, A. Tefas, I. Pitas, T. Tan, H. Yan, F. Smeraldi, J. Begun, N. Capdevielle, W. Gerstner, S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. Comparison of Face Verification Results on the XM2VTS Database. In *Proc. 15th Int'l Conf. Pattern Recognition*, volume 4, pages 858–863, Barcelona, 2000.

[28] J.B. Pierrot, J. Lindberg, J.W. Koolwaaij, H.P. Hutter, D. Genoud, M. Blomberg, and F.Bimbot. A Comparison of *a priori* Threshold Setting Procedures for Speaker Verification in the CAVE Project. In *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pages 125–128, Seattle, 1998.

[29] N. Poh and S. Bengio. How Do Correlation and Variance of Base Classifiers Affect Fusion in Biometric Authentication Tasks? Research Report 04-18, IDIAP, Martigny, Switzerland, 2004. accepted for publication in *IEEE Trans. Signal Processing*, 2005.

[30] N. Poh and S. Bengio. Improving Single Modal and Multimodal Biometric Authentication Using F-ratio Client Dependent Normalisation. Research Report 04-52, IDIAP, Martigny, Switzerland, 2004.

[31] N. Poh and S. Bengio. F-ratio Client-Dependent Normalisation on Biometric Authentication Tasks. In *IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pages 721–724, Philadelphia, 2005.

[32] D. A. Reynolds, T. Quatieri, and R. Dunn. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10(1–3):19–41, 2000.

[33] J.R. Saeta and J. Hernando. On the Use of Score Pruning in Speaker Verification for Speaker Dependent Threshold Estimation. In *The Speaker and Language Recognition Workshop (Odyssey)*, pages 215–218, Toledo, 2004.

[34] C. Sanderson and K.K. Paliwal. Likelihood Normalization for Face Authentication in Variable Recording Conditions. In *Int. Conf. on Image Processing*, pages 301–304, New York, 2002.

[35] K.-A. Toh, X. Jiang, and W.-Y. Yau. Exploiting Global and Local Decision for Multimodal Biometrics Verification. *IEEE Trans. on Signal Processing*, 52(10):3059–3072, October 2004.

[36] V. N. Vapnik. *Statistical Learning Theory*. Springer, 1998.

[37] Xiaojun Wu, Kittler Josef, Jingyu Yang, Messer Kieron, Shitong Wang, and Jieping Lu. On Dimensionality Reduction for Client Specific Discriminant Analysis with Application to Face Verification. In *Advances in Biometric Person Authentication: 5th Chinese Conference on Biometric Recognition, SINOBIOMETRICS, LNCS 3338*, pages 305–312, Guangzhou, 2004.