MOBIO

Mobile Biometry

http://www.mobioproject.org/

Funded under the 7th FP (Seventh Framework Programme)
Theme ICT-2007.1.4
[Secure, dependable and trusted Infrastructure]

D4.8: Advanced Model Adaptation System

Due date: 30/06/2010 **Submission date:** 15/05/2010
Project start date: 01/01/2008 **Duration:** 36 months
WP Manager: Norman Poh **Revision:** 1

Author(s): Norman Poh, Niklas Johansson and Chris McCool

Project funded by the European Commission in the 7th Framework Programme (2008-2010)		
Dissemination Level		
PU	Public	Yes
RE	Restricted to a group specified by the consortium (includes Commission Services)	No
CO	Confidential, only for members of the consortium (includes Commission Services)	No

D4.8: Advanced Model Adaptation System

Abstract:

The effectiveness of a biometric system strongly relies on the quality of the enrolment data. Unfortunately, during enrolment, biometric samples are often of high quality whereas during the operational phase, biometric sample quality can very significantly. In this deliverable, we examine two degrading factors: changing acquisition environment and sensor mismatch.

A promising way to compensate for the lack of variability of the enrolment sample is actively acquire samples during the operational phase and use the confidently labeled samples to update the biometric reference. In the machine-learning community, this learning paradigm is called *semi-supervised learning*.

This deliverable examines two important settings in semi-supervised learning, applied to biometrics; they are online versus offline adaptation, and self-training versus co-training. In online adaptation, correctly matched query samples are used immediately for adaptation whereas in the offline setting, this process takes place at a latter stage, often after accumulating a number of samples.

In self-training, a biometric system labels query samples and incorporates them into training, without relying on any external source of knowledge. In contrast, co-training relies on the knowledge of another biometric system.

Our experimental results on the BANCA bimodal (talking) face and video modalities show that co-training systematically outperforms self-training, and that the online adaptation setting remains to be a difficult strategy, although some improvement has been observed. This is due to the highly dynamic nature of the online adaptation process, i.e., models keep evolving over time. Despite this nature, by using multiple models, we show that online adaptation can still be an effective strategy when the correct threshold is used.

Contents

1 Introduction

1.1 Motivation

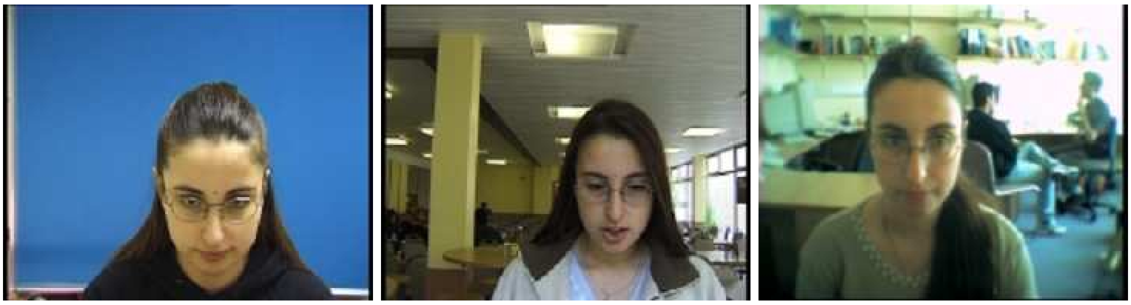
Biometric person authentication remains a challenging problem for two key reasons. Firstly, there are very few enrolment samples to train the model for a particular user. Secondly, there is often significant variation between the samples used for enrolment and the samples that are used to authenticate the user (the test samples). This problem is sometimes referred to as a train-test mismatch.

This mismatch (or large variation) occurs between the enrolment and test samples occurs for several reasons. One reason is that the data acquisition process is vulnerable to these variations. For instance, face images can easily be effected changes in illumination while speech signals can be corrupted by environmental noise such as passing cars or other people speaking. Another reason is that biometric traits can alter temporarily or permanently due to aging, diseases or treatment to a disease.

An important consequence of the above factors is that a reference model cannot be expected to fully and automatically cope with all possible sources of variation. A promising learning paradigm to solve the above problem is known as *semi-supervised learning*. In this paradigm a biometric system is initialised with correctly labelled samples and then (as a classifier) attempts to label the test samples and considers these samples as potential training samples; the initialisation is the only part that is supervised, hence the name semi-supervised. If the samples are labelled correctly, the system can indeed capture the variation of the test conditions. On the other hand, if an impostor's sample is labelled as being genuine, the resultant system may perform significantly worse. While there exists a large body of literature on semi-supervised learning [Zhu05] and concept drift dealing with general adaptive pattern recognition systems [WK96, EP09, Tsy04], the biometric problem deserves a dedicated treatment of its own, considering that a biometric system is potentially rolled out in very large scale and may persist through the life time of a person.

There are, in general, two settings for the semi-supervised experimental framework: *online* and *offline*. In the *online* framework, a biometric system (acting as a classifier here) has to decide whether or not to update a sample immediately after matching. This generally assumes that it is no possible to store the sample due to limited memory storage. In contrast, the *offline* adaptation assumes that there is a memory buffer that can be used to store possible candidate samples. The model parameters can then be updated with the set of samples at a later time, e.g., this happens only when the memory buffer is full.

Another important distinction in semi-supervised learning is the training strategy which can be either *self-training* or *co-training*. In *self-training*, a unimodal biometric system (face or speech) attempts to update its parameters, after observing a test sample, without considering the decision made by another biometric system. In *co-training*, there is necessarily more than one biometric system and the decision of one system can impact on the update strategy of another system.



(a) Controlled (b) Adverse (c) Degraded

Figure 1: The three scenarios of the BANCA database.

1.2 Objectives

The objectives of this deliverable are two-fold: to examine the effect of *online* versus *offline unsupervised adaptation*, and to examine the merit of self-training versus co-training (both of which are also *unsupervised adaptation*).

This deliverable differs significantly with D4.2 as this deliverable examines semi-supervised adaptation whereas D4.2 examined supervised adaptation. Supervised adaptation provides the most optimistic scenario where all test data points are known *a priori* and are used to adapt the client model. This is opposed to *semi-supervised adaptation* where the identity of the biometric sample is not known.

It is expected that the performance of supervised adaptation will be much better than the baseline non-adaptive approach and that the performance of the semi-supervised adaptation will be somewhere between the two. Thus, the supervised adaptation provides the *upper bound* of the achievable performance. Our objective is therefore to assess how well unsupervised adaptation fairs with this performance upper bound.

We validated our experiments using the existing bimodal face and speech BANCA database [BBBB$^+$03b]. This database contains three acquisition conditions, namely controlled, adverse and degraded conditions. With reference to the controlled conditions, the adverse ones are due to acquisition in a noisy environment whereas the degraded ones are due to the use of a different acquisition device. The impact of these three conditions are clearly visible in Figure 1.

There are several important findings that will be shown in this study. Firstly, in Section 3 we show that if the variability of operational biometric samples are captured during enrolment or through supervision (by manually labelling of the operational data), then the system performance systematically improves. This stresses the importance of adapting on operational data with varying quality. Secondly, in Section 4 we found empirically that when performing online adaptation it is important to maintain several references (templates) and to keep updating the models when necessary. Also, the success of online adaptation relies on many factors such as: the order in which query samples arrive, the variability that exists (whether caused by sensor or by the environment) in a data set, and the chosen adaptation threshold. Thirdly, in Section 5 we show that by relying on two (or more) biometric modalities, co-training systematically leads to better generalization performance than self-training.

1.3 Organization

This report is organized into three major technical sections and a database section. We first described the database in Section 2, followed by the three technical sections. Section 3 analyzes the impact of matching and mismatched conditions on the system performance. Section 4 examines the online adaptation setting. Section 5 examines self-training, co-training and their variations, exploiting the bimodal nature of the video sequence. Finally, Section 6 concludes the report.

2 Database and Baseline Systems

2.1 Database

In order to be consistent with the previous deliverables (D3.1 and D3.2), we use the same database, experts and adopt similar experimental protocols. However, the original BANCA protocols cannot be used because in adaptive experimental design it is important to have a separate partition of data for adaptation.

The database used here is the BANCA database [MKS+04a]. This is a bimodal database recorded using a camcorder. It consists of 52 people reading text-prompted sentences as well as answering short questions. The sample images, for all three conditions are shown in Figure 1.

A consequence of this BANCA database setting is that the face verification problem becomes extremely challenging, compared to the speaker verification problem. This is because in both the adverse and degraded conditions, the noise due to the environmental conditions affecting the speech modality, which are all indoor recordings, is still relatively unimportant in comparison with the face modality.

A novel aspect concerning the usage of this database, unlike precedent efforts in [MKS+04a] or [MKS+04b], is that *video sequences* are actually used here, rather than *still images* extracted from the video sequence.

2.2 BANCA protocols

Our experiments were performed on the BANCA (English) database. The database consists of 26 females and 26 males users. Every user were captured in three different conditions; four sessions in each condition. The first four sessions belongs to the *Controlled (Co)* condition, the next four belonged to the *Degraded (De)* condition and the final four sessions belonged to the *Adverse (Ad)* condition. An example of the three conditions are given in Figure 1 and a summary of the conditions and sessions is given in Table 1.

Existing conditions	Session number
Controlled	1 2 3 4
Degraded	5 6 7 8
Adverse	9 10 11 12

Table 1: The twelve sessions split over three conditions

In addition, the BANCA database is split in two equal groups (mixed gender) and they act as validation sets for each other. More precise, the threshold that with the best HTER for group 1 was used as global threshold for group 2 during testing, and vice verse.

When performing verification on the the BANCA database, errors were measured in *Half Total Error Rate (HTER)*

$$HTER = \frac{FRR + FAR}{2} \tag{1}$$

where FRR is the ratio of true clients that were falsely rejected and FAR is the ratio of impostors that were falsely accepted.

2.3 Baseline Systems

The face and speaker verification baseline systems (also referred to as experts) are Bayesian classifiers whose class-conditional densities are approximated using Gaussian Mixture Models (GMMs) with the Maximum *a posteriori* adaptation [RQD00]. This is a long-standing state-of-the-art classifier for the speaker verification, but since then, has also been successfully used for the face verification problem [CSM03]. The face verification problem can benefit from this approach mainly thanks to parts-based local feature descriptors, as illustrated in Figure 2. The parts-based approach first divides an image into overlapping or non-overlapping blocks of image. For each block of image, its texture is described using a *local feature descriptor*. The local feature descriptors used here are based on a post-processed subset of Discrete Cosine Transform features called "DCTMod2" [SP02].

Let $\mathbf{X} \equiv \{\mathbf{x}_i | i = 1, \ldots, N\}$ be a sequence of N feature frames and each feature frame is denoted by $\mathbf{x}_i$ (for the i-th frame). For the face modality, a feature frame is a vector containing the DCT coefficients of a block of image. For the speech modality, a feature frame contains Mel-scale Cepstral Coefficients [RJ93]. These features are a short-term representation of spectral envelopes filtered by a set of filters motivated by the human auditory system.

Let $p(\mathbf{x}|\omega_o)$ be the likelihood function of the world or background model and $p(\mathbf{x}|\omega_j)$ be the model for the claimed identity $j \in \{1, \ldots, J\}$. In parts-based face or speaker verification, both $p(\mathbf{x}|\omega_o)$ and $p(\mathbf{x}|\omega_j)$, for any j, are estimated using a Gaussian Mixture Model (GMM) [Bis99]. The world model is first obtained from a large pool of sequences

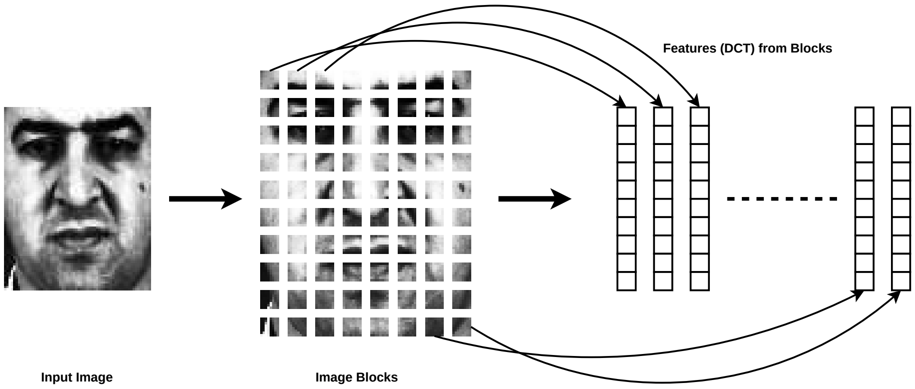


Figure 2: A flow chart of describing the extraction of feature vectors from the face image for the parts-based approach.

$\{\mathbf{X}\}$ contributed by a large and possibly separate population of users (possibly from an external database than the one used for enrollment/testing). Each client-specific model is then obtained by adapting the world model upon the presentation of the enrollment data of a specific user/client.

If the score y is greater than a pre-specified threshold, one declares that the query data $\mathbf{X}$ belongs to the model j. Hence, this will result in an acceptance decision. Otherwise, one rejects the hypothesis and hence rejects the identity claim.

The speaker verification classifier used here differs from the face one in the following ways. First, the variability across sessions are removed thanks to now a standard technique called factor analysis [KBD05, VBS05, MSFB07]. This technique is applied to all training and test data prior to building a (client-specific) GMM model.

3 Preliminary analysis on conditional mismatch

As the face is captured without the interaction of the user, it becomes difficult to control the conditions (illumination variations, different head poses and facial expressions). Therefore, face recognition methods need to be more robust to a variety of different conditions than those biometrics that are more controlled (iris).

In this section we will show experimentally that if there is a mismatch between conditions (for training and testing) there will be a significant performance loss. Because of this performance loss, it is important to come up with a method to deal with the conditional mismatch between train and test. One way to do this is be to continuously update (adapt) the user model and incorporate more information about the user in different conditions.

3.1 Protocols

Our aim is to evaluate conditional mismatches for face recognition. A well used database for face recognition that include different condition is the BANCA database. However, none of the seven testing protocols in BANCA (MC, MD, MA, UD, UA, P and G [BBBB$^+$03a]) deal exhaustively with conditional mismatches. Therefore, we created a new “mismatched protocol” that was close to the other seven protocols but was extended to cover all combinations of conditions.

To derive the mismatched protocol we picked one session from each condition (sessions 1, 5 and 9) and used combinations of these to enrol the user models. By doing so, we formed seven different configurations that are presented in Table 2. The remaining nine sessions (2, 3, 4, 6, 7, 8, 10, 11, 12) were used for testing our system, the nine sessions spanned all the three conditions.

Name	Sessions used to enrol
A	1
B	1 5
C	1 5 9
D	1 9
E	5
F	5 9
G	9

Table 2: The seven different system configurations. For each system configuration we indicate the sessions used to enrol client models.

Name	Controlled	Degraded	Adverse
A	*matched*	**mismatched**	**mismatched**
B	*matched*	*matched*	**mismatched**
C	*matched*	*matched*	*matched*
D	*matched*	**mismatched**	*matched*
E	**mismatched**	*matched*	**mismatched**
F	**mismatched**	*matched*	*matched*
G	**mismatched**	**mismatched**	*matched*

Table 3: A more detailed presentation of which system configuration was matched or mismatched for a given condition when running our experiments.

3.2 Results and discussion

We evaluated the performance across different conditions and estimated the loss in performance when not training and testing in the same condition. For our experiments we used the part-based Gaussian Mixture Model framework, as presented in Section 2.3. We define mismatch as the case where we have not seen a condition during training that we later see during test. All systems that are mismatched are in dark gray while the systems that are matched are in light gray.

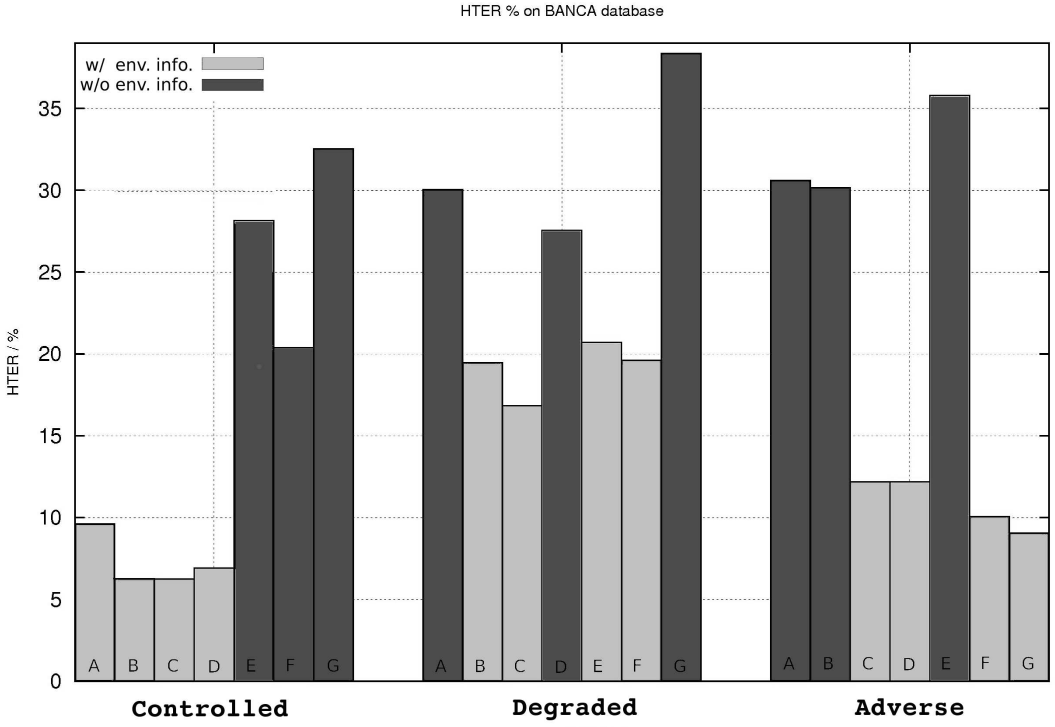


Figure 3: Evaluation of models that contain samples from the testing condition (bars in light grey) compared to the models that do not contain samples from the testing condition (bars in dark grey).

In each of the three conditions in Figure 3 (controlled, degraded and adverse), there are four systems that perform better than the other three. These four systems correspond exactly to the system configurations that are matched. More precisely, the system configurations that contain samples in training from the condition that is later being tested in, performs better.

It can therefore be concluded that there is a need to handle conditional mismatches. One way to do this would be to continuously update or adapt the biometric templates (client models). We are going to present two different types of unsupervised adaptation in the Section 4 and Section 5.

4 Online Adaptation

There are two different types of adaptation when adapting a user model with a given set of samples: either we know which of the samples from the set to use, or we have to use a method to select the "correct" samples. If there are labels associated with all the samples it is called supervised adaptation, whereas if we have to infer the labels it is called unsupervised adaptation. As one knows the true labels in supervised adaptation, supervised adaptation will always perform better than unsupervised adaption. This means that supervised adaptation will form a lower bound (error) for unsupervised adaptation.

When performing supervised adaption there are two different approaches. The first approach refers to *on-line adaptation* and consists of taking the decision to adapt or not the model directly as soon as a new sample is given. The second approach refers to *off-line adaptation* and consists of taking the decision to adapt or not the model taken later in time as opposed to on-line model adaptation. In both cases, it should be noted that multiple samples can be accumulated and considered for adaptation. Our aim is to evaluate a system that is on-line and unsupervised.

We will begin by presenting the multi-model system that was used during our evaluation. We will then describe the method we used to trigger adaptation along with the database and protocols used for our experiments. Finally, we will present and discuss the results for the best method of *on-line adaptation*.

4.1 Single- vs. Multi-model system

We chose to use a multi-model framework in our evaluations. This meant that each user had a collection of models instead of just one as is the normal case. We chose the multi-model approach because it gave us more flexibility and the ability to have "experts". However, since a single-model system is a special case of a multi-model system we were able to evaluate both multi- and single-model systems in the same framework. When adapting with new samples we either created a new model and added it to the collection, or we selected a model and modified it.

There are two main issues when using a set of multiple models: first we need a scheme to score against a collection of models and second we need a scheme to include new information (data) to modify (add or replace) the models in our collection. We will expand on those two issues in the following paragraphs. To describe our multi-model system we use the notation presented in Table 4.

Notation	Meaning
$\mathbf{x}$	samples from access (input)
Ω	collection of models
ω	a model in the collection Ω ($\omega \in \Omega$)
ω_{world}	the world model (sometimes called the universal background model)
$\Lambda(\mathbf{x}\|\omega)$	score for samples $\mathbf{x}$ given model ω
$\mathbb{S}(\omega)$	all samples used when creating model ω
ω^*	the best scoring model during test. ($\omega^* = \underset{\omega\in\Omega}{\operatorname{argmax}}\,\Lambda(\mathbf{x}\|\omega)$)

Table 4: Notation to describe our multi-model system for one user.

The problem of scoring against a set of multiple models was covered in detailed by Kittler et al. [Kit98]. It was concluded in that paper that the *sum rule* outperformed other methods (min, max, product). However, since we might have a different number of models for each user we modify the sum rule into the average rule,

$$\Lambda(\mathbf{x}|\Omega, \omega_{world}) = \frac{1}{|\Omega|} \sum_{\omega\in\Omega} \Lambda(x|\omega) - \Lambda(x|\omega_{world}), \tag{2}$$

where $|\Omega|$ is the number of models in our collection and $\Lambda(\mathbf{x}|\omega)$ is the score against an individual model ω. $\Lambda(\mathbf{x}|\Omega, \omega_{world})$ is therefore the average score for a collection of models Ω and a world model ω_{world}.

We will now present four different adaptation strategies which we will call **MM**, **MM+**, **SM** and **SM+**, where MM is short for multiple-model, SM is short of single-model and the "+" is to indicate "extended version". They have much in common but we will present each of them in detail below.

In the first strategy **MM** we took the new data and our input samples **x** and performed *maximum a posterior (MAP)* adaption using the world model (ω_{world}). Then we added the newly created model, called ω_{new}, into the collection of already existing models Ω. This can be written more formally as:

$$\begin{aligned} \text{Adapt}: &\quad x, \omega_{world} &\longrightarrow \omega_{new} \\ \text{Add}: &\quad \omega_{new}, \Omega &\longrightarrow \Omega := \Omega \cup \omega_{new} \end{aligned}$$

The second strategy, called **MM+**, is a modification of the MM strategy. In addition to using the input samples **x** when adapting we included samples from another model. The extra samples were taken from the model that scored the best at test time (we will refer to this model as ω^*). The **MM+** strategy can be formally written as:

$$\begin{aligned} \text{Select best model}: &\quad \omega^* = \underset{\omega \in \Omega}{\operatorname{argmax}}\, \Lambda(\mathbf{x}|\omega) & \\ \text{Adapt}: &\quad x, \mathbb{S}(\omega_*), \omega_{world} &\longrightarrow \omega_{new} \\ \text{Add}: &\quad \omega_{new}, \Omega &\longrightarrow \Omega := \Omega \cup \omega_{new} \end{aligned}$$

Both strategies above are so called multiple-model strategies where we chose to include the new data **x** by creating a new model. To broaden our evaluation we also chose to use two single-model strategies. These two strategies either aggregated the new information (**x**) into the existing model or tried to replace the existing model altogether.

The first of the two single-model strategies, called **SM+**, can be presented formally as follows:

$$\begin{aligned} \text{Select best model}: &\quad \omega^* = \underset{\omega \in \Omega}{\operatorname{argmax}}\, \Lambda(\mathbf{x}|\omega) & \\ \text{Adapt}: &\quad x, \mathbb{S}(\omega_*), \omega_{world} &\longrightarrow \omega_{new} \\ \text{Replace}: &\quad \omega_{new}, \Omega &\longrightarrow \Omega := \omega_{new} \end{aligned}$$

and is very similar to the MM+ strategy. However, instead of keeping all the previous models we only kept one (the latest).

Our last strategy is called **SM** and was included in our tests to evaluate the need to aggregate data, old and new. It can formally be presented as:

$$\begin{aligned} \text{Adapt :} \quad & x, \omega_{world} && \longrightarrow \omega_{new} \\ \text{Replace :} \quad & \omega_{new}, \Omega && \longrightarrow \Omega := \omega_{new} \end{aligned}$$

which means that no old information is kept when we include the new information ($\mathbf{x}$). We summarise our four different strategies in Table 5.

Name	Brief summary
MM	A multi-model system that creates a new model with the new information
MM+	Same as MM but uses additional information when creating the new model
SM	A single-model system that only uses the newest information (samples)
SM+	A single-model system that aggregates all information, new and old, into one model

Table 5: Brief summary of the four adaption strategies used in our evaluation of on-line adaptation.

Above we have describe *how* the different adaption strategies that we used during our evaluations. Next we will describe *when* an adaptation was triggered.

4.2 Adaptation threshold

In on-line adaption we are forced to directly take the decision whether to use the new samples (input samples $\mathbf{x}$) or not. In our evaluations we used a threshold which we refer to as the *adaptation threshold*. Adaptation is then triggered when the average score of the input samples ($\mathbf{x}$) is greater than the *adaptation threshold*. Using the same notation as in Table 4, this can be formally written as:

$$\Lambda(\mathbf{x}|\Omega, \omega_{world}) = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \Lambda(x|\omega) - \Lambda(x|\omega_{world}) \tag{3}$$

$$\begin{cases} \Lambda(\mathbf{x}|\Omega, \omega_{world}) > \mu & \text{adapt} \\ \Lambda(\mathbf{x}|\Omega, \omega_{world}) \leq \mu & \text{do not adapt} \end{cases} \tag{4}$$

The purpose of this adaptation threshold is to ensure that we can effectively adapt the user's model. This can be considered as the trade-off between accepting new (difficult) samples to adapt with while ensuring that we do not adapt with data from another user (an impostor).

The adaptation threshold is central to the performance of the adaptation system and so several thresholds were chosen for evaluation. To choose the adaptation threshold we focused on the impostor distribution as we had more samples for this distribution. Having the adaptation threshold close to the impostor mean would result in more impostor samples being used for adaptation. Using this as our starting point, we tested thresholds at a fixed interval from the impostor mean. In order for our technique to be reproducible we decided to use the impostor variance as our step size from the impostor mean. In total five different adaptation thresholds were evaluated, these adaptation thresholds are presented in Table 6.

Adapting threshold	1	2	3	4	5
Threshold	μ +0.5 σ	μ +1.0 σ	μ +1.5 σ	μ +2.0 σ	μ +2.5 σ

Table 6: The 5 different adaptation thresholds used for on-line adaptation, where μ and σ are the mean and standard deviation of the impostor distribution (derived independently on the specified development set).

4.3 Protocol for on-line adaptation

In order to evaluate our on-line adaptation we created four different testing scenarios. The two key aspects we wanted to investigate were the effect of:

1. the composition of testing environments, and
2. the order of client and impostor trials.

The composition of testing environments refers to the order of the testing environments, for instance will the system be robust if the user continuously changes environments. While the order of the client and impostor trials is important because if the model is easily adapted then it would suggest that if the impostor trials are before the client trials then the user's model could quickly diverge to better match the impostor.

We based our *on-line adaptations* protocol on BANCA protocol P. According to this protocol, we enrol on session 1 which is a controlled environment, then test on sessions $2, 3, 4$ which are all in controlled environments. We then test against sessions $6, 7, 8$ which are considered to be degraded environments and finally we test against sessions $10, 11, 12$ which are all in the adverse condition.

Normally the order of the sessions does not matter. However for an adapting system it is very important and will change the end result (this is the problem of the composition of testing environment). Therefore we created two orders to test in: the "straight" testing order and the "mixed" testing order. For the "straight" testing order we tested in increasing session order. This means that we test our system in one condition at the time (controlled, degraded and adverse). For the "mixed" testing order we cycled over the testing condition, which meant that each test was performed in a different condition. These two testing orders are presented in detail in Table 7.

Task	Session
Enrol	1
Test client access	2
Test impostor access	2
Test client access	3
Test impostor access	3
Test client access	4
Test impostor access	4
Test client access	6
Test impostor access	6
Test client access	7
Test impostor access	7
Test client access	8
Test impostor access	8
Test client access	10
Test impostor access	10
Test client access	11
Test impostor access	11
Test client access	12
Test impostor access	12

Task	Session
Enrol	1
Test client access	2
Test impostor access	2
Test client access	6
Test impostor access	6
Test client access	10
Test impostor access	10
Test client access	3
Test impostor access	3
Test client access	7
Test impostor access	7
Test client access	11
Test impostor access	11
Test client access	4
Test impostor access	4
Test client access	8
Test impostor access	8
Test client access	12
Test impostor access	12

Table 7: On left and right we present two out of our four on-line adaptation protocols. On the left we present the "straight" protocol where we test the environments in groups (first the controlled, then the degraded and finally the adverse). On the right we present the "mixed"protocol where we alternate between the different environments.

We created two more protocols so that we could test effect of putting impostor trials before client trials. To test this effect we simply changed the protocols in Table 7 so that the impostor accesses occurred first. This gives use a total of four cases to evaluate with and are summarised in Table 8.

	Session order	Access order
1.	Straight order	Client first
2.	Mixed order	Client first
3.	Straight order	Impostor first
4.	Mixed order	Impostor first

Table 8: The four different on-line adapting protocols used for our experiments.

4.4 Results and Discussion

In order to keep the results simple we have chosen to only present the results for the best adaptation method (*MM+*) at the best adaptation threshold ($\mu + 0.5\sigma$). The rest of the results, for all the five adaptation thresholds and for all the different protocols are available in Appendix C. Moreover, since the main interest is unsupervised adaptation (harder than supervised), we have placed all the results for supervised adaptation in the Appendix and only kept the results for unsupervised adaptation.

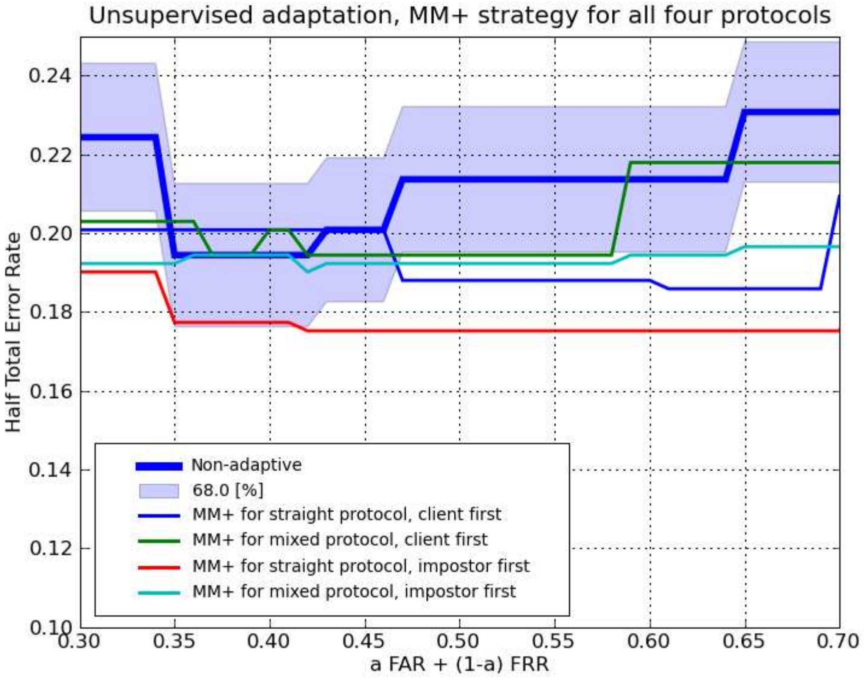


Figure 4: In this figures we compare unsupervised adaptation with a system that does not adapt (comparable to BANCA P system). Results are shown for the multi-model system that use extra samples (from another model) when creating the new model. This adaptation strategy was called *MM+* in our description.

In Figure 4 we show the performance of the MM+ strategy (our best performing adaptation strategy) for each of the four testing scenarios. In our tests we evaluated the strategy at five different adaptation threshold but only the best threshold for each testing scenario is presented in the figure. To compare our results we also include the performance of the baseline (non-adapted) system, this is equivalent to the baseline system on BANCA protocol P. This non-adaptive system will act as our baseline in our discussion of the results and is plotted with a thicker blue line and is surrounded with a 68% confidence interval in light blue colour. We will say that our results are significant if they perform better (lower error) than the baseline including this 68% interval.

A common operating point for verification systems is the equal error rate. This corresponds to when the *false acceptance rate (FAR)* is equal to the *false rejection rate (FRR)*, this occurs at point 0.5 on the x-axis of our plot. At this point, our adaptation strategy (MM+) is performing significantly better than the baseline for each of the four testing scenarios. As each of the four testing scenarios is different, this indicates that the MM+ adaptation strategy is robust to different orders of testing conditions and different mixes of impostors and client access attempts.

Comparing the four different testing scenarios, the MM+ strategy was best for the testing scenario that used the straight testing order with client accesses occurring before impostor accesses (session-by-session). In this testing scenarios the MM+ strategy was significantly better than the base line in the interval 0.3 to 0.7 (ratio between FAR and FRR). The reason that our strategy was more successful in this testing scenario than the other three is an area of future work.

In summary, the best on-line adaptation strategy **MM+** is performing better than the system that it did not adapt. It performs significantly better, in all of the four testing scenarios, at the most common operating point (where FAR equal FRR). At this operating point, the strategy can be called robust because it performs well in each the four testing scenarios.

5 Self-training, Co-training and Fusion-based Co-training in Offline Settings

5.1 Introduction

5.1.1 Problem characterization

This section differs from the Section 4 in two aspects.

First, the focus here is on *offline setting*, i.e., a memory buffer exists and is used to hold a number of query samples. The implication of this setting is that the sequence in which data samples arrive has little impact. In contrast, in the *online setting*, the adaptation problem becomes much harder if non-match (impostor) samples arrive *before* the genuine ones.

Second, in this section, the bimodal nature (face and speech) of videos is considered. When two modalities are available, one can employ the *co-training* algorithm [BM98]. In this algorithm, two (face and speech) systems attempt to label independently a (video) query sample. The labeled samples are then used to update each system. This strategy has been reported to be effective in multimodal biometrics [RMR08] as well as other domain

Algorithm 1 The Self-training algorithm

```
• Given: labelled data $\mathcal{L}$ and unlabelled data $\mathcal{U}$
• Loop until stop criterion satisfied:
    – Train g1 using $\mathcal{L}$
    – Label $\mathcal{U}$ using g1 to obtain $\mathcal{U}_*$
    – Add the highly confident self-labelled samples from $\mathcal{U}_*$ to $\mathcal{L}$
    – Remove the self-labelled examples from $\mathcal{U}$
```

problems [BM98].

5.2 Methodology

We shall first revisit self-training and co-training algorithm. Then, we will generalize the co-training algorithm to the more general case of *fusion-based co-training*. Furthermore, as a control experiment, we will also examine a novel case called "cross-training".

5.2.1 Revisiting self-training and co-training

In self-training, a biometric system, say g1, attempts to infer labels from an unlabeled data set $\mathcal{U}$. If the labels are inferred with sufficiently high confident, they are incorporated into a labeled data set, $\mathcal{L}$. The labeling process is repeated typically until no more labels can be inferred this way or stopped at a predetermined number of iterations. Algorithm 1 describes this procedure more formally.

In the original co-training algorithm [BM98], two biometric systems, say g1 and g2, attempt to infer labels from $\mathcal{U}$ independently. The confidently labeled samples are then added to $\mathcal{L}$. This procedure is described in Algorithm 2.

The main difference between Algorithms 1 and 2 is that the information between the two modality experts are not shared in self-training, whereas in co-training, the experts work collaboratively.

5.2.2 Fusion-based co-training

We observe that in the original co-training algorithm, the *union* of two inferred labels by two algorithms are used simultaneously. Hence, this operation can be interpreted as an OR fusion rule.

Rather than using the OR fusion rule, in multimodal biometrics, it is common to use a trainable classifier such as logistic regression. In essence, the OR rule is an example of decision-level fusion whereas logistic regression is an example of score-level fusion. Score-level fusion is generally better than decision-level fusion because the former considers the

Algorithm 2 The original co-training algorithm

```
• Given: labelled data $\mathcal{L}$ and unlabelled data $\mathcal{U}$
• Loop until stop criterion satisfied:
    – Train g1 using $\mathcal{L}$
    – Train g2 using $\mathcal{L}$
    – Label $\mathcal{U}$ using g1 to obtain $\mathcal{U}_*^1$
    – Label $\mathcal{U}$ using g2 to obtain $\mathcal{U}_*^2$
    – Add the highly confident self-labelled examples in $\mathcal{U}_*^1 \bigcup \mathcal{U}_*^2$ to $\mathcal{L}$
    – Remove the self-labelled examples from $\mathcal{U}$
```

Algorithm 3 The fusion-based co-training algorithm

```
• Given: labelled data $\mathcal{L}$ and unlabelled data $\mathcal{U}$
• Loop until stop criterion satisfied:
    – Train g1 using $\mathcal{L}$
    – Train g2 using $\mathcal{L}$
    – Train f using $\mathcal{L}$
    – Label $\mathcal{U}$ using f to obtain $\mathcal{U}_*$
    – Add the highly confident self-labelled examples in $\mathcal{U}_*$ to $\mathcal{L}$
    – Remove the self-labelled examples from $\mathcal{U}$
```

confidence of expert output, as reflected by the *absolute value* of the matching score. This piece of information is simply not taken into account in decision-level fusion.

Because of the above nature, combining a strong expert with a weak expert at the decision level is not always beneficial [Dau00]. In comparison, the score level fusion can still optimally draw the strength of both experts even with this unbalanced performance, in the Bayes sense [PK08b, PB05].

The fusion-based co-training algorithm is shown in Algorithm 3. The main difference in this algorithm compared to self-training and the original co-training is that the FB co-training requires an additional step in order to train the fusion classifier. This step is not needed if the fusion classifier used is based on fixed rules such as sum and product.

Since a trainable fusion classifier is used here, one has to ensure that that the data samples (scores) used to train the fusion classifier should not be the same as those used to train the baseline experts, since by so-doing, the resultant trained fusion classifier will be

Algorithm 4 The cross-training algorithm

```
• Given: labelled data $\mathcal{L}$ and unlabelled data $\mathcal{U}$
• Let $\mathcal{L}^1$=$\mathcal{L}$ and $\mathcal{L}^2$=$\mathcal{L}$
• Loop until stop criterion satisfied:
    – Train g1 using $\mathcal{L}^2$
    – Label $\mathcal{U}$ using g1 to obtain $\mathcal{U}_*^2$
    – Train g2 using $\mathcal{L}^1$
    – Label $\mathcal{U}$ using g2 to obtain $\mathcal{U}_*^1$
    – Add the highly confidently labelled samples from $\mathcal{U}_*^1$ to $\mathcal{L}^1$ and $\mathcal{U}_*^2$ to $\mathcal{L}^2$
    – Remove the examples just labelled from $\mathcal{U}$
```

overly optimistically biased (i.e., the expert outputs are more confident than they appear to be).

To avoid the positive bias, one can adopt k-fold cross validation. For instance, let k be two. One can then partition $\mathcal{L}$ into two non-overlapping sets, say $\mathcal{L}_1$ and $\mathcal{L}_2$. Then, one trains a base expert on $\mathcal{L}_1$ and generate scores using $\mathcal{L}_2$. Similarly, one trains another base expert on $\mathcal{L}_2$ and generate scores using $\mathcal{L}_1$. The union of the resultant scores obtained from $\mathcal{L}_1$ and $\mathcal{L}_2$ are used to train the fusion classifier “f”. In this way, from the fusion’s perspective, the training scores are unbiased.

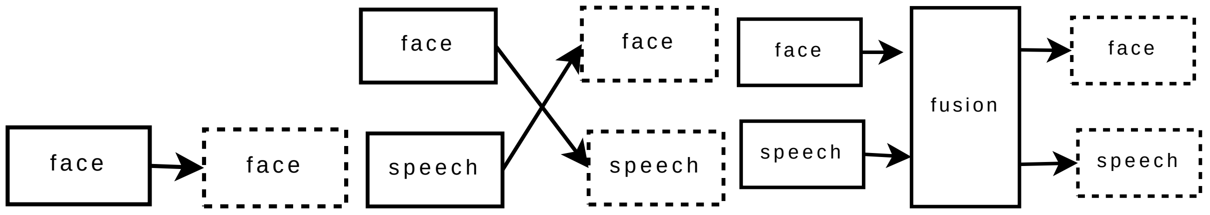
5.2.3 Cross-training

Since the difference between self-training and co-training is principally due to whether or not information is exchanged, it is instructive to study *how* this information is exchanged. A possible intermediate way of information exchange is to retrain one classifier by the labels inferred by another classifier. This gives rise to the *cross-training* algorithm, as shown in Algorithm 4.

Figure 5 shows the architecture of self-training, the original co-training and fusion-based co-training architecture. Although we show only the self-train face expert in Figure 5(a), it should be clear that the self-train speech expert proceeds in the same way.

5.3 Experimental Protocols

In order to compare adaptive versus non-adaptive (baseline) systems, we will use three partitions of data, each for enrollment, adaptation and testing. The first line in Table 9 shows the number of examples *for each client* in each partition of data. The second line shows the exact session numbers used to constitute the respective partition of data as well



(a) Self-training (b) Co-training (c) Fusion-based co-training

Figure 5: Three schemes of semi-supervised learning scenarios considered in this study. Boxes drawn with solid lines denote initial models whereas those drawn with dashed lines denote updated models. A solid arrow from module A to B means that the data samples whose labels are inferred by module A are used to update module B.

Table 9: Partition of data

Data sets	Enrollment			Adaptation			Test		
conditions	c	a	d	c	a	d	c	a	d
# match samples	1	0	0	0	1	1	3	3	3
session	$\{1\}$				$\{5\}$	$\{9\}$	$\{2,3,4\}$	$\{6,7,8\}$	$\{10,11,12\}$
# non-match samples	20 †	20 †	20 †	0	25	25	4	4	4
session	$\{1\}$	$\{5\}$	$\{9\}$		$\{5\}$	$\{9\}$	$\{2,3,4\}$	$\{6,7,8\}$	$\{10,11,12\}$

Note: c=controlled, a=adverse, d=degraded. Each row shows the number of match or non-match samples *for each client*. There are 26 clients in each group, and there are two groups according to the Banca protocols. †: the numbers indicated here are the *background* models.

as the conditions under which the data (video) sample is obtained (controlled, adverse or degraded).

Recall that there are two groups of 26 clients in the original BANCA protocols. Hence, to get the total number of genuine (match) samples for each partition of data, each condition type (controlled, adverse or degraded) and for each group, one multiplies the numbers in Table 9 by 26 (row 2). The number of impostor attempts is obtained in a similar way (See row 3 of Table 9).).

The enrollment data partition normally does not contain non-match samples. However, the numbers indicated here (with †) are the number of samples used for training the background model or for feature extraction (e.g., principal component analysis), which comes with the BANCA English database.

The experiments are designed to compare five adaptive settings: no adaptation, self-training, cross-training, FB co-training and supervised adaptation. The training set of the non-adaptive system consists of only the enrollment partition of the data; the adaptive partition of data is not used at all. On the other hand, the training set of the supervised system consists of the enrollment and adaptive partitions of the data. In all five settings, the test partition is reserved uniquely for evaluating the system performance. Each of these settings are evaluated on the unimodal face, unimodal speech and bimodal fusion.

5.4 Threshold Determination

Before showing the result, there is another crucial aspect: threshold determination. One simple strategy is to use all possible thresholds. For this purpose, we choose three levels of threshold: a "loose", a "moderate" and a "stringent" threshold for adaptation.

A principled way of determining the three levels of threshold *by their confidence* is to map the threshold onto a probabilistic scale, e.g., the probability being a client given the expert output (say y):

$$confidence(y) = P(\mathrm{C}|y)$$

After this mapping process, also known as *score calibration*, we simply compare $confidence(y)$ with the adaptation threshold, Δ_{adpt}. If $confidence(y)$ exceeds the threshold, then the corresponding sample is added to the labeled set $\mathcal{L}$. Thus, the three levels of threshold can be taken as $\{0.25, 0.50, 0.75\}$, respectively for the loose, moderate and stringent thresholds, respectively.

Throughout our experiments, the posterior probability is estimated via logistic regression, which is trained on the held out group (i.e., when testing the BANCA g1 group of users, the data of g2 is used for training). The logistic regression is expressed by:

$$P(\mathrm{C}|y) = \frac{1}{1 + \exp\left(-(w_1 y + w_0)\right)}$$

where y is the output of the unimodal system (face or speech). For the bimodal fusion, we use

$$P(\mathrm{C}|y) = \frac{1}{1 + \exp\left(-(w_2 y_2 + w_1 y_1 + w_0)\right)}$$

instead, where y_i is expert output i and w_i is its associated weight. The weight parameters are estimated using the expectation maximization principle. The realized algorithm is known as "gradient-ascent" [HTF01].

5.5 Results

The experimental results are divided into two parts: unimodal (face and speech) systems and bimodal fusion. For each case, we shall plot only the half total error rate (HTER) and pooled DET curves (over G1 and G2 protocols). The HTER is the average of false acceptance rate and false rejection rate. For the assessment here, the threshold used here minimizes the Equal Error Rate on the development data set.

Figure 6 contains two panels, each showing the HTER of the face and speech unimodal systems. Below, we will explain the different face systems tested, as it is clear that the speech systems can be explained in exactly the same way.

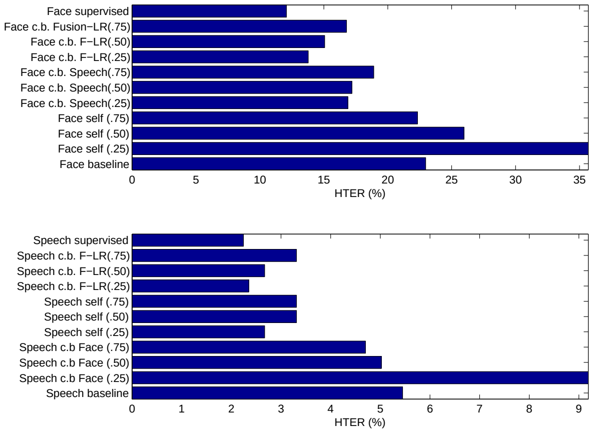


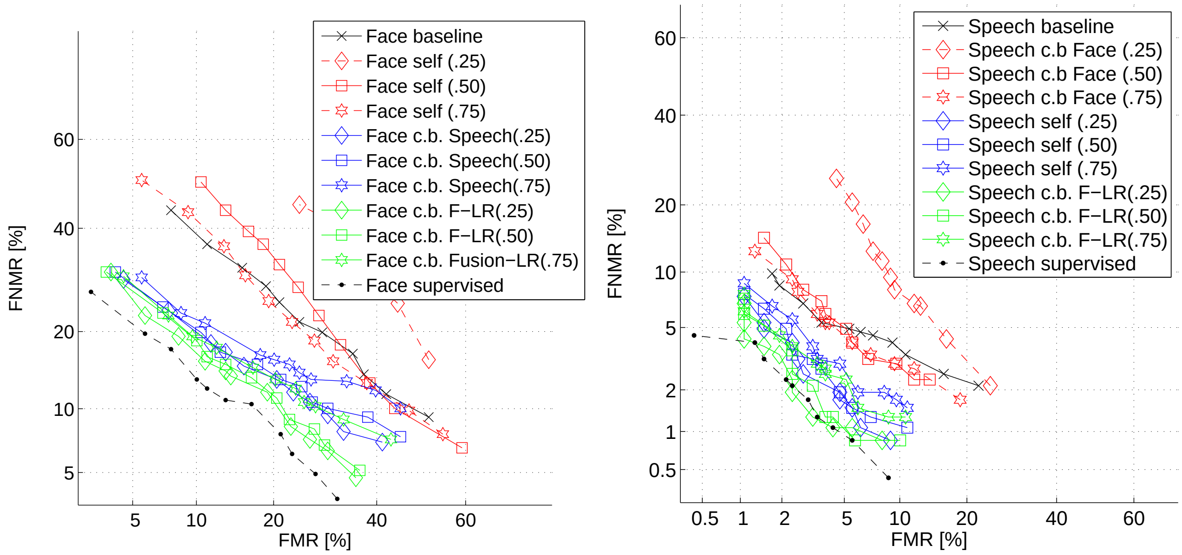
Figure 6: HTER of (a) face and (b) speech systems.

- Face supervised: this is a supervised system, where the labels of the adaptive partition of data is known.
- Face baseline: this is the original non-adaptive system; it effectively assumes that the adaptive partition of data simply does not exist.
- Face self-train: this is a self-training system that attempts to infer labels from the adaptive partition of data. The inferred data samples are used to augment the original enrollment partition of data for training.
- Face co-trained by speech: this is a cross-training setting where a face system is trained by the labels obtained from the speech system.
- Face co-trained by fusion: This system is trained by the labels obtained from the fusion system – logistic regression in our case.

The speech systems are obtained in a similar way. Therefore, for the "speech system co-trained by face", the data samples inferred by the face system are used to train the speech system.

Figure 7 shows the DET curves of the five systems, along with three levels of calibrated threshold (by its confidence), for the face and speech modality separately. We observe the following:

- The self-training speech system benefits from the loose threshold
- On the other hand, the self-training face system benefits from the more stringent threshold.
- The speech system that is co-trained by face degrades significantly in performance, compared to the baseline non-adaptive system



(a) Face

(b) Speech

Figure 7: DET curves of (a) face and (b) speech adaptive experiments.

- On the other hand, the face system benefits from co-training by the speech system.
- The fusion-based co-training performs most optimally with the loose threshold.

These observations are all consistent in supporting the case that fusion-based co-training is better than cross-training (face co-trained by speech or speech co-trained by face). It is interesting to observe that the self-training strategy degrades the face system but improves the speech system. This suggests that an already good system is likely to benefit from self-training whereas a weak system will further degrades in performance with self-training.

Figures 8 and 9 present the HTER and DET curves for the multimodal systems, respectively. Both figures again reveal a similar trend, except that the difference of performance between the supervised system and the best performing co-training system (at 0.25 adaptation threshld) is significantly larger here.

6 Conclusions

This deliverable examines several important issues regarding adaptive biometric systems. Our empirical findings suggest that

- In order for semi-supervised adaptation to be effective, the (unlabeled) data set must have the same conditions as the actual operational conditions.

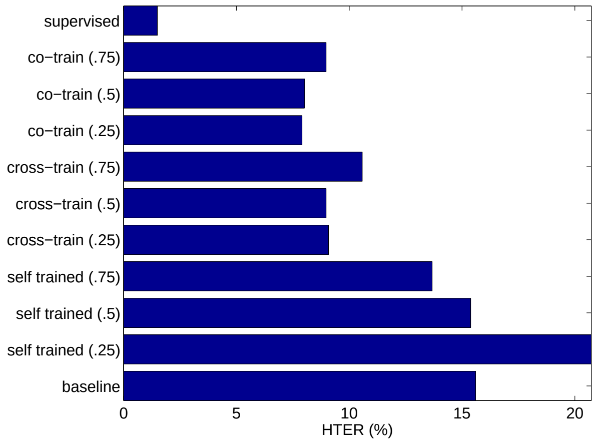


Figure 8: HTER

- Using multiple models clearly is important for online adaptation, especially in the presence of changing biometric sample quality/conditions.

- Fusion-based co-training clearly outperforms self-training and cross-training. This shows the importance of exchanging information among experts.

Our experiments also point out the need for further studying the dynamic behaviour of online adaptation. Since the system performance keeps involving over time, a time-dependent analysis of performance such as [PK07a] is necessary. Another future research direction is to study user-specific threshold rather than a global one, as done here. Since the system performance is client-dependent, it is reasonable to expect that user-specific threshold or normalization [PK07b, PK08a] will improve the system performance. Last but not least, for the multi-model online adaptive setting, it is unclear whether or not each model has successfully gauged an aspect of signal quality. In theory, if the degrading factor or factors have been modeled successfully, no further adaptation is needed. In short, our assessment exercise suggests that the online adaptation strategy deserves a much more extensive study.

Acknowledgments

We would like to thank the following partners for their significant contributions to this deliverable:

- Sébastien Marcel, Chris McCool and Niklas Johansson (IDIAP)

- Driss Matrouf (LIA)

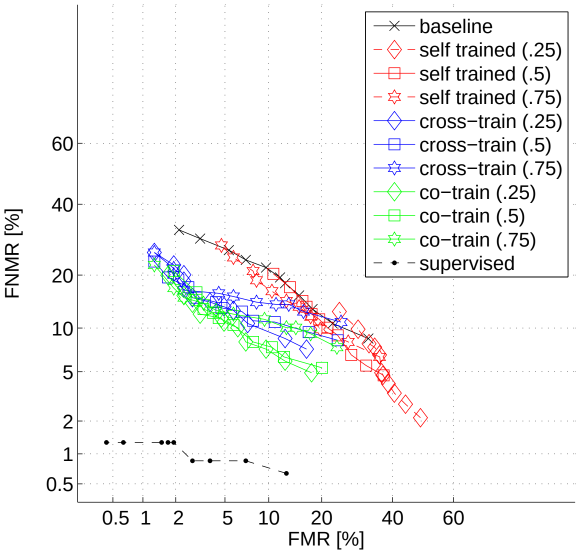


Figure 9: Pooled DET curves the 11 fusion systems

References

[BBBB+03a] E. Bailly-Bailliere, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Porée, et al. *Audio-and Video-Based Biometric Person Authentication.* Springer New York, 2003.

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

[BBF+04] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska, and D. A. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Applied Signal Processing, Special issue on biometric signal processing*, 2004.

[Bis99] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1999.

[BM98] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, pages 92–100, 1998.

[CL01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[CSM03] F. Cardinaux, C. Sanderson, and S. Marcel. Comparison of MLP and GMM classifiers for face verification on XM2VTS. In *Proc. International Conference on Audio- and Video-based Biometric Person Authentication*, pages 1058–1059, 2003.

[Dau00] J. Daugman. Biometric decision landscapes. Technical Report TR482, University of Cambridge Computer Laboratory, 2000.

[DPMR00] G. Doddington, M. Przybocki, A. Martin, and D. Reynolds. The NIST speaker recognition evaluation — overview, methodology, systems, results, perspective. *Speech Communication*, 31(2-3):225–254, 2000.

[EP09] R. Elwell and R. Polikar. Incremental learning of variable rate concept drift. In *Multiple Classifier Systems (MCS 2009)*, pages 142–151, Iceland, 2009.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[KBD05] P. Kenny, G. Boulianne, and P. Dumouchel. Eigenvoice Modeling With Sparse Training Data. *IEEE Transactions on Speech and Audio Processing*, 13(3):345, 2005.

[Kit98] J. Kittler. Combining classifiers: A theoretical framework. *Pattern Analysis & Applications*, 1(1):18–27, 1998.

[LC04] S. Lucey and T. Chen. A GMM parts based face representation for improved verification through relevance adaptation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 855–861, 2004.

[LG96] C. Lee and J. Gauvain. Bayesian adaptive learning and map estimation of hmm. In C.-H. Lee, F. Soong, and K. Paliwal, editors, *Automatic speech and speaker recognition : Advanced topics*, pages 83–107. Kluwer Academic Publishers, Boston, Massachusetts, USA, 1996.

[MKS$^+$04a] K. Messer, J. Kittler, M. Sadeghi, M. Hamouz, A. Kostin, F. Cardinaux, S. Marcel, S. Bengio, C. Sanderson, N. Poh, Y. Rodriguez, J. Czyz, L. Vandendorpe, C. McCool, S. Lowther, S. Sridharan, V. Chandran, R. P. Palacios, E. Vidal, L. Bai, L-L. Shen, Y. Wang, Chiang Yueh-Hsuan, H-C. Liu, Y-P. Hung, A. Heinrichs, M. Muller, A. Tewes, C. vd Malsburg, R. Wurtz, Zg. Wang, Feng Xue, Yong Ma, Qiong Yang, Chi Fang, Xq. Ding, S. Lucey, R. Goss, , and H. Schneiderman. Face authentication test on the banca database. In *Int'l Conf. Pattern Recognition (ICPR)*, volume 4, pages 523–532, 2004.

[MKS$^+$04b] K. Messer, J. Kittler, M. Sadeghi, M. Hamouz, A. Kostyn, S. Marcel, S. Bengio, F. Cardinaux, C. Sanderson, N. Poh, Y. Rodriguez, K. Kryszczuk, J. Czyz, L. Vandendorpe, J. Ng, H. Cheung, and B. Tang. Face authentication competition on the banca database. In *Intl. Conf. Biometric Authentication*, pages 8–15, 2004.

[MSFB07] D. Matrouf, N. Scheffer, B. Fauve, and J-F. Bonastre. A straightforward and efficient implementation of the factor analysis model for speaker verification. In *INTERSPEECH Conference, Antewerp, Belgium*, 2007.

[PB05] N. Poh and S. Bengio. How Do Correlation and Variance of Base Classifiers Affect Fusion in Biometric Authentication Tasks? *IEEE Trans. Signal Processing*, 53(11):4384–4396, 2005.

[PK07a] N. Poh and J. Kittler. A Method for Estimating authentication Performance Over Time, with Applications to Face Biometrics. In *12th IAPR Iberoamerican Congress on Pattern Recognition (CIARP)*, pages 360–369, Via del Mar-Valparaiso, Chile, 2007.

[PK07b] N. Poh and J. Kittler. On the Use of Log-likelihood Ratio Based Model-specific Score Normalisation in Biometric Authentication. In *LNCS 4542, IEEE/IAPR Proc. Int'l Conf. Biometrics (ICB'07)*, pages 614–624, Seoul, 2007.

[PK08a] N. Poh and J. Kittler. Incorporating Variation of Model-specific Score Distribution in Speaker Verification Systems. *IEEE Transactions on Audio, Speech and Language Processing*, 16(3):594–606, 2008.

[PK08b] N. Poh and J. Kittler. On Using Error Bounds to Optimize Cost-sensitive Multimodal Biometric Authentication. In *Proc. 19th Int'l Conf. Pattern Recognition (ICPR)*, 2008.

[PM93] W. B. Pennebaker and J. L. Mitchell. *JPEG still image data compression standard.* New York: Van Nostrand Reinhold, 1993.

[Rey97] D. A. Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *Proc. European Conference on Speech Communication and Technology (EuroSpeech)*, pages 963–966, Rhodes, Greece, September 1997.

[RJ93] L. Rabiner and B-H Juang. *Fundamentals of Speech Recognition.* Oxford University Press, 1993.

[RMR08] A. Rattani, G.L. Marcialis, and F. Roli. Capturing large intra-class variations of biometric data by template coupdate. In *6th IEEE Biometric Symposium*, Tampa,USA, 2008.

[RQD00] D. A. Reynolds, T. Quatieri, and R. Dunn. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10(1–3):19–41, 2000.

[SP02] C. Sanderson and K. K. Paliwal. Fast feature extraction method for robust face verification. *Electronic Letters*, 38(25):1648–1650, 2002.

[Tsy04] A. Tsymbal. The problem of concept drift: Definitions and related work. Technical report, Department of Computer Science, Trinity College, Ireland, 2004.

[VBS05] R. Vogt, B. Baker, and S. Sridharan. Modelling Session Variability in Text-Independent Speaker Verification. In *Proc. European Conference on Speech Communication and Technology (EuroSpeech)*, 2005.

[WK96] G. Widmer and M. Kubat. Learning in the presence of concepts drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996.

[Zhu05] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

A Parts-Based Gaussian Mixture Model (PB-GMM) for Face Verification

The first face verification baseline model implementation presented in this report combines part-based approaches and GMM modeling. Parts-based approaches divide the face into blocks, or parts, and treats each block as a separate observation of the same underlying signal (the face). According to this technique, a feature vector is obtained from each block by applying the Discrete Cosine Transform (DCT) and the distribution of these feature vectors is then modelled using GMMs. Several advances have been made upon this technique, for instance, Cardinaux *et al.* [CSM03] proposed the use of background model adaptation while Lucey and Chen [LC04] examined a method to retain part of the structure of the face utilising the parts-based framework as well as proposing a relevance based adaptation.

Feature Extraction

The feature extraction algorithm is described by the following steps. The face is normalised, registered and cropped. This cropped and normalised face is divided into blocks (parts) and from each block (part) a feature vector is obtained. Each feature vector is treated as a separate observation of the same underlying signal (in this case the face) and the distribution of the feature vectors is modelled using GMMs. This process is illustrated in Figure 2.

The feature vectors from each block are obtained by applying the DCT. Even advanced feature extraction methods such as the DCTmod2 method [SP02] use the DCT as their basis feature vector; the DCTmod2 feature vectors incorporate spatial information within the feature vector by using the deltas from neighbouring blocks. The advantage of using only DCT feature vectors is that each DCT coefficient can be considered to be a frequency response from the image (or block). This property is exploited by the JPEG standard [PM93] where the coefficients are ranked in ascending order of their frequency.

Feature Distribution Modelling

Feature distribution modelling is achieved by performing background model adaptation of GMMs [CSM03, LC04]. The use of background model adaptation is not new to the field of biometric authentication; in fact, it is commonly used in the field of speaker verification [DPMR00]. Background model adaptation first trains a world (background) model Ω_{world} from a set of faces and then derives the client model for the i^{th} client Ω^{i}_{client} by adapting the world model to match the observations of the client.

Two common methods of performing adaptation are mean only adaptation [Rey97] and full adaptation [LG96]. Mean only adaptation is often used when there are few observations available because adapting the means of each mixture component requires fewer observations to derive a useful approximation. Full adaptation is used where there are sufficient observations to adapt all the parameters of each mode. Mean only adaptation is the method chosen for this work as it requires fewer observations to perform adaptation, this is the same adaptation method employed by Cardinaux *et al.* [CSM03].

Verification

To verify an observation, $\mathbf{x}$, it is scored against both the client (Ω^{i}_{client}) and world (Ω_{model}) model, this is true even for methods that do not perform background models adaptation [SP02]. The two models, Ω^{i}_{client} and Ω_{world}, produce a log-likelihood score which is then combined using the log-likelihood ratio (LLR),

$$h(\mathbf{x}) = \ln(p(\mathbf{x} \mid \Omega^{i}_{client})) - \ln(p(\mathbf{x} \mid \Omega_{world})), \tag{5}$$

to produce a single score. This score is used to assign the observation to the world class of faces (not the client) or the client class of faces (it is the client) and consequently a threshold τ has to be applied to the score $h(\mathbf{x})$ to declare (verify) that $\mathbf{x}$ matches to the i^{th} client model Ω^{i}_{client}, i.e if $h(\mathbf{x}) \geq \tau$.

B Gaussian Mixture Model-Support Vector Machine Based Speaker Verification

The use of GMM in a GMM-UBM framework has been a standard approach in the speaker verification [BBF$^+$04]. In addition to this framework, the Latent Factor Analysis (LFA) is systematically applied for all systems in training and testing [KBD05, VBS05, MSFB07]. From the resulting session compensated model it is possible to extract supervectors by concatenating Gaussian means. These supervectors can be used directly in a SVM classifier. This association between the factor analysis and SVM allows to benefit from the FA decomposition power and SVM classification power. The implemented baseline system uses Z-T-norm for score normalization.

Feature extraction

The signal is characterized by 50 coefficients including 19 linear frequency cepstral coefficients (LFCC), their first derivative, their first 11 coefficients of second derivatives and the delta-energy. They are obtained as follows: 24 filter bank coefficients are first computed over 20ms Hamming windowed frames at a 10ms frame rate. Bandwidth is limited to the 300-3400Hz range.

Here, the energy coefficients are first normalized using a mean removal and variance normalization in order to fit a 0-mean and 1-variance distribution. The energy component is then used to train a three component GMM, which aims at selecting informative frames. The most energized frames are selected through the GMM. Once the speech segments of a signal are selected, a final process is applied in order to refine the speech segmentation:
1- overlapped speech segments between both the sides of a conversation are removed,
2- morphological rules are applied on speech segments to avoid too short ones, adding or removing some speech frames.

Finally, the parameter vectors are normalized to fit a 0-mean and 1-variance distribution. The mean and variance estimators used for the normalization are computed file by file on all the frames kept after applying the frame removal processing.

World models

Two GMM world models are used, one for males and one for females. The two GMM are trained using Fisher English Training Speech Part 1 (LDC:LDC2004S13), and consists of about 10 million speech frames each for males and females.

Resulting world models are 512 gender dependent GMM's with diagonal covariance matrices. For a better separation of initial classes, frames are randomly selected among the entire learning signal via a probability followed by an iteration of the EM algorithm, to estimate the GMM parameters. During the estimation of the world model parameters, instead of using all the learning signals in their temporal order, 10% of frames is selected randomly at each new iteration. For the two last iterations, the entire signal is classically used in its temporal order. During all the process, a variance flooring is applied so that no variance value is less than 0.5.

Client, test and impostor models with Factor Analysis

A speaker model can be decomposed into three different components: world, a speaker dependent and session dependent components [KBD05, VBS05, MSFB07]. A GMM mean super-vector is defined as the concatenation of the GMM component means. In the following, (h, s) will indicate the session h of the speaker s. The latent factor analysis model, can be written as:

$$\mathbf{m}_{(h,s)} = \mathbf{m} + \mathbf{D}\mathbf{y}_s + \mathbf{U}\mathbf{x}_{(h,s)}, \tag{6}$$

where $\mathbf{m}_{(h,s)}$ is the session-speaker dependent super-vector mean, $\mathbf{D}$ is $S \times S$ diagonal matrix (S is the dimension of the supervector), $\mathbf{y}_s$ the speaker vector (its size equal S), $\mathbf{U}$ is the session variability matrix of low rank R (a $S \times R$ matrix) and $\mathbf{x}_{(h,s)}$ are the session factors, a R vector. Both $\mathbf{y}_s$ and $\mathbf{x}_{(h,s)}$ are normally distributed among $\mathcal{N}(0, I)$. $\mathbf{D}$ satisfies the following equation $\mathtt{I} = \tau \mathbf{D}^t \mathbf{\Sigma}^{-1} \mathbf{D}$ where τ is the *relevance factor* required in the standard MAP adaptation.

The client model is obtained by performing the decomposition of equation 6 and by retaining only the speaker dependent components:

$$\mathbf{m}_s = \mathbf{m} + \mathbf{D}\mathbf{y}_s, \tag{7}$$

The success of the factor analysis model relies on a good estimation of the $\mathbf{U}$ matrix, thanks to a sufficiently high amount of data, where a high number of different recordings per speaker is available. In these experiments the U matrix is trained by using about 240 speakers (120 males and 120 females) coming from NIST'04. For each speaker about 20 sessions are considered.

Kernel based scoring and SVM modeling

By using (7), the factor analysis model estimates supervectors containing only speaker information, normalized with respect to the session variability. A probabilistic distance kernel that computes a distance between GMM's, well suited for a SVM classifier. Let $\mathcal{X}_\mathbf{s}$ and $\mathcal{X}_{\mathbf{s}'}$ be two sequences of speech data corresponding to speakers $\mathbf{s}$ and $\mathbf{s}'$, the kernel formulation is given below.

$$K(\mathcal{X}_\mathbf{s}, \mathcal{X}_{\mathbf{s}'}) = \sum_{g=1}^{M} \left(\sqrt{\alpha_g} \mathbf{\Sigma}_g^{-\frac{1}{2}} \mathbf{m}_s^g\right)^t \left(\sqrt{\alpha_g} \mathbf{\Sigma}_g^{-\frac{1}{2}} \mathbf{m}_{s'}^g\right). \tag{8}$$

This kernel is valid when only means of GMM models are varying (weights and covariance are taken from the world model). $\mathbf{m_s}$ is taken here from the model in eq. 7, *i.e.* $\mathbf{m}_s = \mathbf{m} + \mathbf{D}\mathbf{y}_s$.

The LIA_SpkDet toolkit benefits from the LIBSVM [CL01] library to induce SVM and to classify instances. SVM models are trained with an infinite (very large in practice) C parameter thus avoiding classification error on the training data (hard margin behavior). The negative labeled examples are speakers from the normalization cohort.

C Additional results for on-line adaptation

In this section we present the complete results from our evaluation of on-line supervised adaptation and unsupervised adaptation. We will start by presenting the results for supervised adaption for our four different testing scenarios (see Table 8 for more details). Thereafter we will present the evaluation of the unsupervised adaptation for the four different testing scenarios at each of the five adapting thresholds (see Table 6 for more details).

C.1 Supervised Adaptation

In Figures 10 and 11 we present the EPC curves for our four different adapting system. We have chosen to show the EPC curves because the behaviors are very different ratios of *false acceptance rates (FAR)* and *false rejection rates (FRR)*.

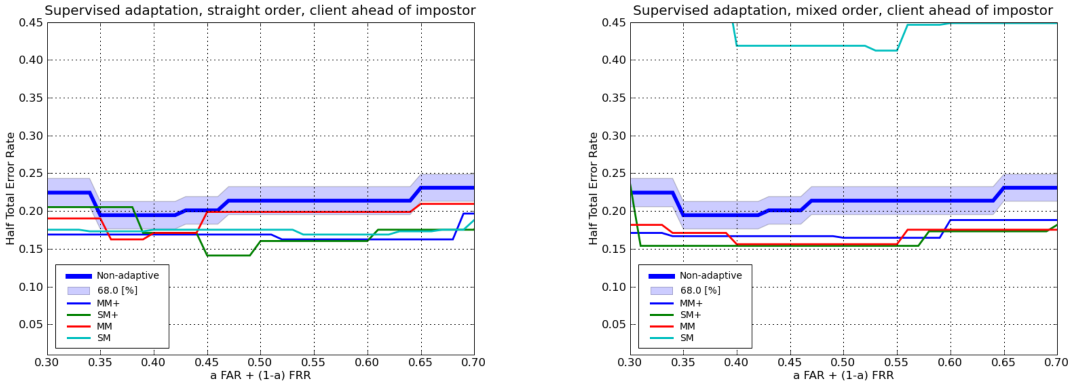


Figure 10: Supervised adaptation for the “straight” and “mixed” testing order, when testing clients ahead of impostors in each session.

C.2 Unsupervised Adaptation

We have decided to split our results for the unsupervised adaptation into four different parts, where each part is a different testing scenario. As with the results for supervised adaptation we present the EPC curves as the behavior changes dramatically at different ratios of *false acceptance rates (FAR)* and *false rejection rates (FRR)*.

As described in Table 6 we defined five different adaption thresholds for which we did our evaluations. The staring point and step size for the adaption thresholds were based on the impostor mean and standard deviation. Therefore, we calculated the mean and standard deviation for the impostor distributions for both groups (validation sets g1 and g2). *To be fair and correct we used the validation groups mean and standard variation to*

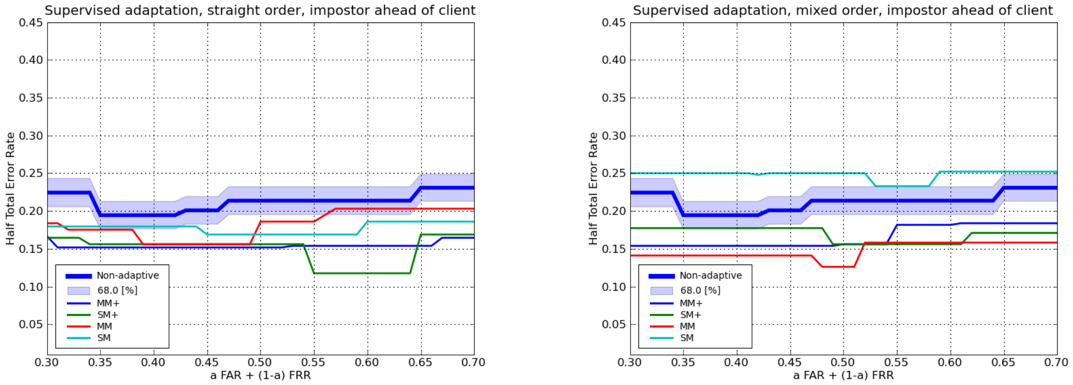


Figure 11: Supervised adaptation for the “straight” and “mixed” testing order, when testing impostors ahead of clients in each session.

calculate the adapting thresholds.

The adapting thresholds were chosen as shown in Table 11.

Group	Impostor mean μ	Impostor std σ
1	-0.274	0.308
2	-0.336	0.310

Table 10: Impostor distributions for group one and two

Group	μ +0.5 σ	μ +1.0 σ	μ +1.5 σ	μ +2.0 σ	μ +2.5 σ
1	-0.120	0.034	0.188	0.342	0.496
2	-0.181	-0.026	0.130	0.285	0.440

Table 11: The 5 operating points of unsupervised adaptation.

We start by presenting the results for the “straight” testing order when we tested the client before the impostor in each session.

C.2.1 Straight protocol - Client before impostor

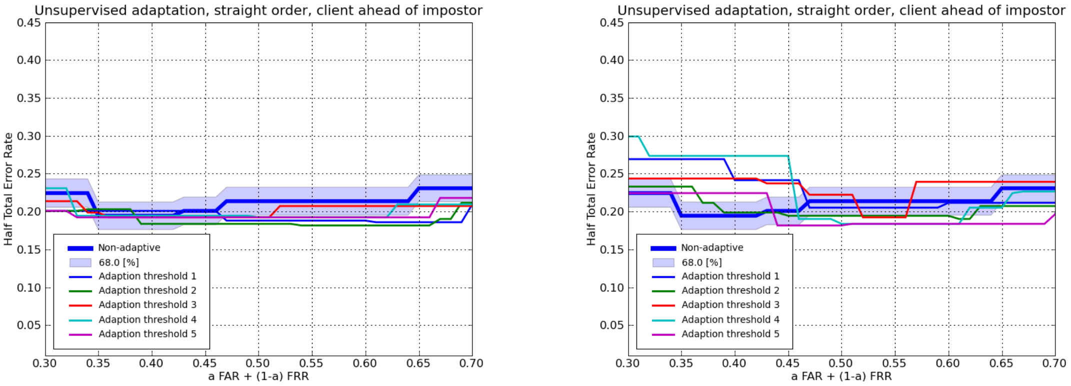


Figure 12: EPC curves for MM+ (left) and for SM+ (right)

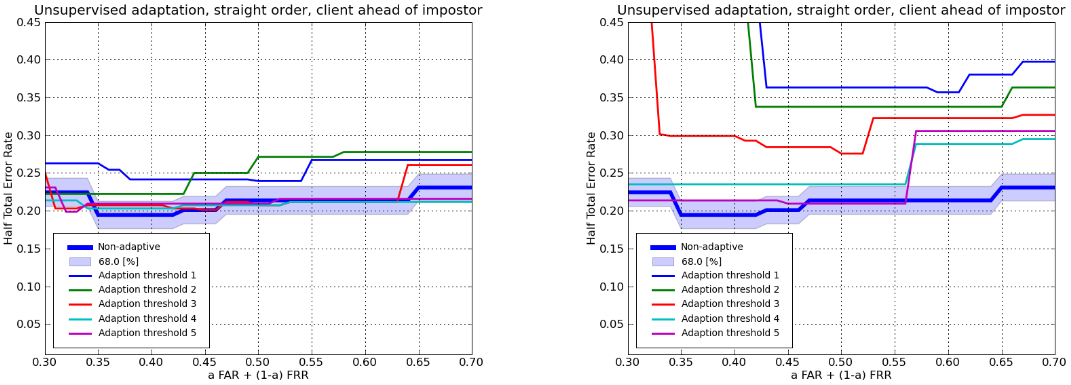


Figure 13: EPC curves for MM (left) and for SM (right)

C.2.2 Mixed protocol - Client before impostor

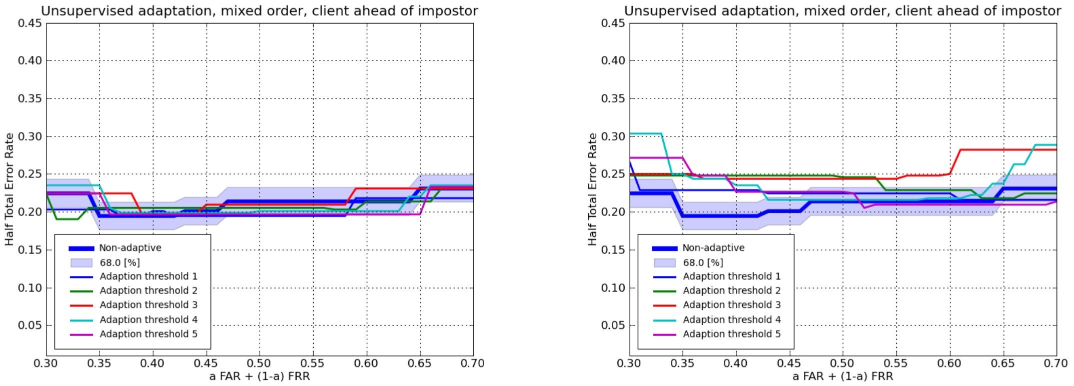


Figure 14: EPC curves for MM+ (left) and for SM+ (right)

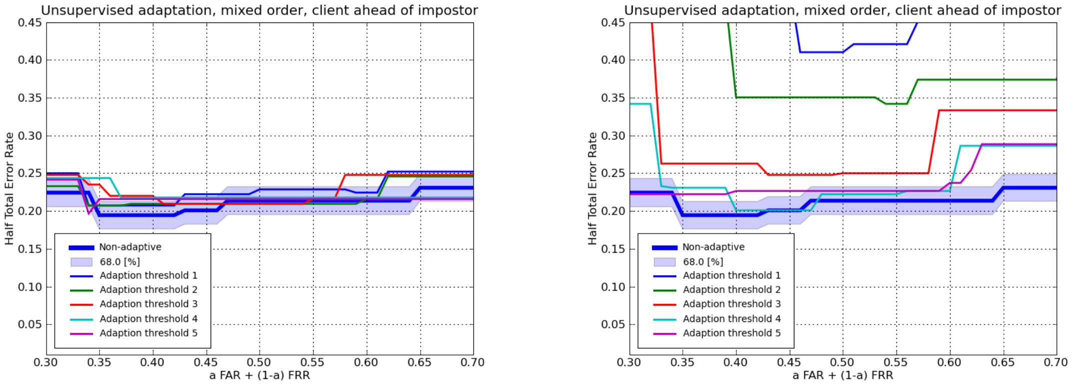


Figure 15: EPC curves for MM (left) and for SM (right)

C.2.3 Straight protocol - Impostor before client

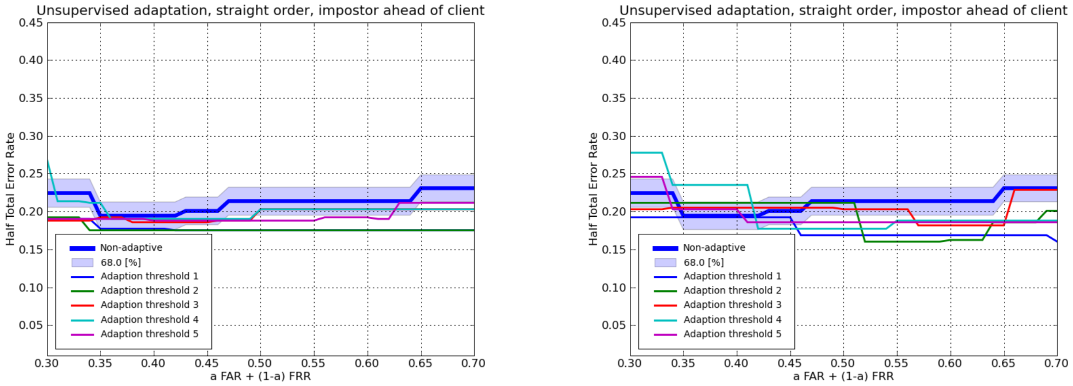


Figure 16: EPC curves for MM+ (left) and for SM+ (right)

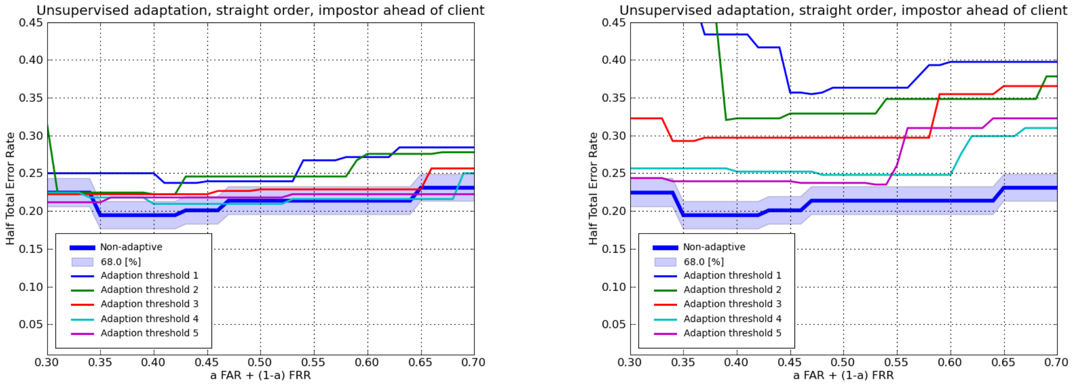


Figure 17: EPC curves for MM (left) and for SM (right)

C.2.4 Mixed protocol - Impostor before client

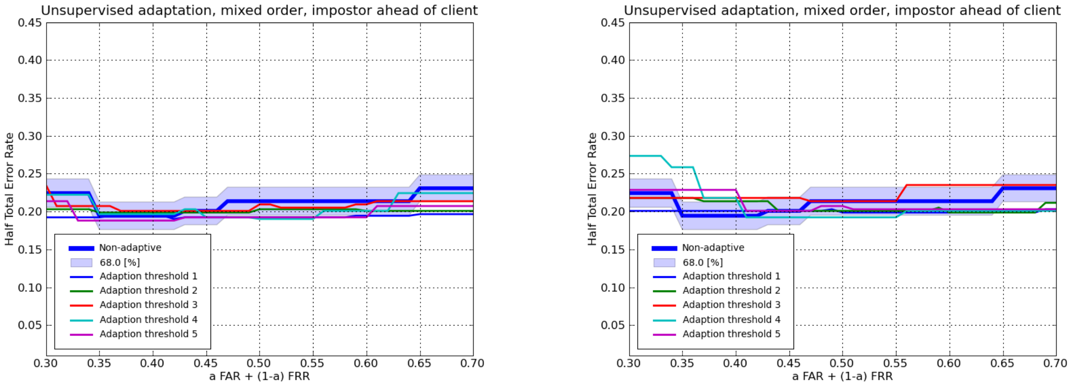


Figure 18: EPC curves for MM+ (left) and for SM+ (right)

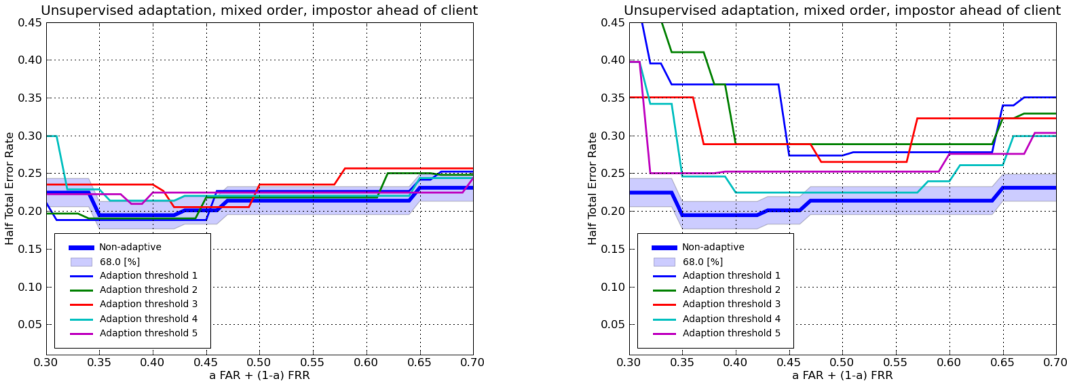


Figure 19: EPC curves for MM (left) and for SM (right)