IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

# Decision fusion in a multi-modal identity verification system using a multi-linear classifier

Patrick Verlinde * Gilbert Maître † Eddy Mayoraz ‡

IDIAP–RR 97-06

September 1997

Dalle Molle Institute for Perceptive Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

* Royal Military Academy (RMA), Electrical Engineering and Telecommunications Department (ELTE), B1000 Brussels, Belgium, e-mail: patrick.verlinde@tele.rma.ac.be

† Dalle Molle Institute for Perceptive Artificial Intelligence (IDIAP), CH1920 Martigny, Switzerland, e-mail: gilbert.maitre@idiap.ch

‡ Dalle Molle Institute for Perceptive Artificial Intelligence (IDIAP), CH1920 Martigny, Switzerland, e-mail: eddy.mayoraz@idiap.ch

IDIAP Research Report 97-06

# Decision fusion in a multi-modal identity verification system using a multi-linear classifier

Patrick Verlinde Gilbert Maître Eddy Mayoraz

September 1997

**Abstract.** This paper presents the use of a multi-linear classifier allowing to fuse the results of several modalities in a multi-modal person identity verification context. In the considered verification system, each of the $d$ modalities forms an autonomous bloc that produces a score, which is not only supposed to be monotone but also to have a value between zero and one. The fusion module that we are discussing here takes a binary decision: accept or reject the identity claimed by the person, based on the whole of the scores given in parallel by all $d$ modalities. To realize this fusion module we have developed a classifier that, on the one hand, accepts the monotonicity hypothesis and, on the other hand, is based on separating the classes (accept - reject) by a combination of half-spaces, a technique from which it derived its name. The classifier is trained using couples formed by extracting an example from each class and the half-spaces are determined by maximizing a global separability measure of the thus formed couples. Afterwards, each region of the partition of the $d$ dimensional space, generated by the half-spaces, is labeled with the corresponding class, using the Logical Analysis of Data (LAD) method. The performance of the developed multi-linear classifier has been evaluated on multi-modal experimental data and the obtained results are presented.
Keywords: *decision fusion, multi-linear classifier, multi-modal identity verification.*

# Contents

# List of Figures

# List of Tables

# 1 Introduction

The automatic identification/verification is rapidly becoming an important tool in several applications such as controlled access to restricted (physical and virtual) environments. Just think about secure tele-shopping, accessing the safe room of your bank, .... A number of different readily available techniques, such as passwords, personal (magnetic) cards and PIN-numbers are already widely used in this context, but the only thing - if any - they really verify, is the correct restitution of a character and/or digit combination. As is well known, this can very easily lead to abuses, induced for instance by the loss or theft of a personal card. Therefore a new kind of methods is emerging, based on so called *biometric* measures, such as vocal (speech) or visual (face, profile, ... ) information of the person to be identified. Biometric measures in general, and user-friendly (vocal, visual) biometric measures in particular, are very attractive because they have of course the huge advantage that one can not lose or forget them, since they are based on a physical appearance measure. We can start using these user-friendly biometric measures now, thanks to the progress made in the field of automatic speech analysis and artificial vision. In general these new applications use a "classical" technique (password, etc ... ) to claim a certain identity which is then verified using one or more biometric measures.

If one uses only a single user-friendly biometric measure, the results obtained are not good enough. This is due to the fact that these user-friendly biometric measures tend to *vary with time* for one and the same person and to make it even worse, the importance of this variation is itself very variable from one person to another. This especially is true for the vocal (speech) modality, which shows an important *intra-speaker variability*. One possible solution to try to cope with the problem of this *intra-person* variability is *to use more than one user-friendly biometric measure*. In this specific case, each biometric measure is also called a *modality*. In this new *multi-modal* context, it is thus becoming important to be able to combine (or *fuse*) the outcomes of different modalities. There is currently a significant international interest in the topic. Also the European project M2VTS (*Multi-Modal Verification for Tele-services and Security applications*), in the framework of which this research work has been performed, is concerned with this combination of verification modalities.

Combining the outcomes of different modalities can be done by using classical data fusion techniques [Ant95, Das94, Kle93, WL90], but the major drawback of the bulk of all these methods is their rather high degree of complexity, which is expressed - amongst else - by the fact that these methods tend to incorporate a lot of parameters that have to be estimated. If this estimation is not done properly (*i.e.* using enough training data), this places a serious constraint on the ability of the system to correctly generalize [Ben95]. But actually a major difficulty of this particular fusion problem is the scarcity of multi-modal training data. Indeed, to keep the system user friendly, the enrollment of a (new) client should not take too much time, and as a direct consequence from this, the amount of client training data tends to be limited. To try to deal with this lack of training data, one possibility is to develop *simple* classifiers (*i.e.* for instance classifiers that use only few parameters), so that their parameters can be estimated using only limited amounts of training data. The development of such a simple classifier is discussed in Section 3. This *multi-linear classifier* can be situated in the class of Parametric Templates according to the taxonomy proposed by [WL90].

In Section 2 we show the main characteristics of a multi-modal identity verification system. Section 3 contains the development and Section 4 explains the implementation of the multi-linear classifier. The performance of this classifier has been evaluated using multi-modal experimental data coming from the M2FDB database [PV, PV97a]. The test protocol and the obtained results are presented and discussed in Section 5. Conclusions and future work are presented in the Sections 6 and 7 respectively.

# 2 Multi-modal identity verification system

## 2.1 Characterization of an identity verification system

The verification of the identity of a person is typically a two-class problem: either the person is the one (in this case he is called a *client*), or is not the one (in that case he is called an *impostor*) he claims to be. When dealing with binary hypothesis testing (and that is exactly what a classifier needs to do in a two class problem), it is trivial to understand that the classifier can make two kind of errors [Tre68]. Applied to this problem of the verification of the identity of a person, these two errors are called:

- False Rejection (FR): *i.e.* when an actual *client* is rejected as being an *impostor*;
- False Acceptance (FA): *i.e.* when an actual *impostor* is verified as being a *client*.

The performance of an identity verification system is usually characterized only by global error rates computed during tests: the False Rejection Rate [FRR = (number of FR)÷(number of client accesses)] and the False Acceptance Rate [FAR = (number of FA)÷(number of impostor accesses)]. A unique measure can be obtained by combining these two errors into the Total Error Rate [TER = (number of FA + number of FR)÷ (total number of accesses)] or its complimentary, the Total Success Rate [TSR = 1 - TER]. A perfect identity verification (FAR=0 and FRR=0) is in practice unachievable. However, as shown by the study of binary hypothesis testing [Tre68], any of the two FAR, FRR can be reduced as close to zero as desired, with the drawback of increasing the other one. The Equal Error Rate (EER), *i.e.* the FAR and FRR when they are the same, is often used as the only performance measure of an identity verification method.

Mono-modal verification systems are usually built arranging two main modules in cascade: (1) a module which compares the recorded data from the person under test with a reference client model and outputs a scalar number, followed by (2) a decision module realized by a thresholding operation. In such a system the scalar number, which we call *score*, is assumed to be a monotone measure of identity correctness. Formally this property can be stated as: given the two scores $s_1 \leq s_2$, if *accept* is the better decision for $s_1$, then *accept* is the better decision for $s_2$, and if *reject* is the better decision for $s_2$, then *reject* is the better decision for $s_1$.

## 2.2 Multi-modal architecture

The straightforward way of building a multi-modal verification system from $d$ mono-modal systems is to input the $d$ scores provided in parallel into a fusion module which has to take the decision *accept* or *reject*. However, two alternatives remain for the fusion module: dependence or independence of person identity. For the sake of simplicity we have opted for an identity independent fusion module. The architecture of the multi-modal verification system is represented in Figure 1.

## 2.3 Decision fusion as a particular classification problem

In a verification system as the one represented in Figure 1 with $d$ modalities, the fusion module has to realize a mapping from $\mathbb{R}^d$ into the set $\{\text{rejected}, \text{accepted}\}$. Such a mapping characterizes a classifier having a $d$-dimensional input vector and two classes: $\{\text{rejected}, \text{accepted}\}$. The fusion module can therefore be designed as a classifier, however with some application specific constraints (see also Figure 2):

**Monotonicity** The monotonicity property of scores (see section 2.1) states a monotonicity constraint for the classifier. Formally this property can be stated as: given the two sets of scores $(s_1^1, s_2^1, \ldots, s_d^1)$ and $(s_1^2, s_2^2, \ldots, s_d^2)$ such that $\forall i : s_i^1 \leq s_i^2$, if the decision for $(s_1^1, s_2^1, \ldots, s_d^1)$ is *accept*, then the decision for $(s_1^2, s_2^2, \ldots, s_d^2)$ is *accept*, and if the decision for $(s_1^2, s_2^2, \ldots, s_d^2)$ is *reject*, then the decision for $(s_1^1, s_2^1, \ldots, s_d^1)$ is *reject*.

**Scarcity of training data** In an operational verification system, large amounts of impostor accesses can be simulated with the recordings of other persons. In most applications, client accesses however are scarce since clients would not accept performing long training sessions.

**Tunable FAR/FRR trade-off** As described in section 2.1, any of the two errors, FAR and FRR, can be reduced as close to zero as desired, with the drawback of increasing the other one. In

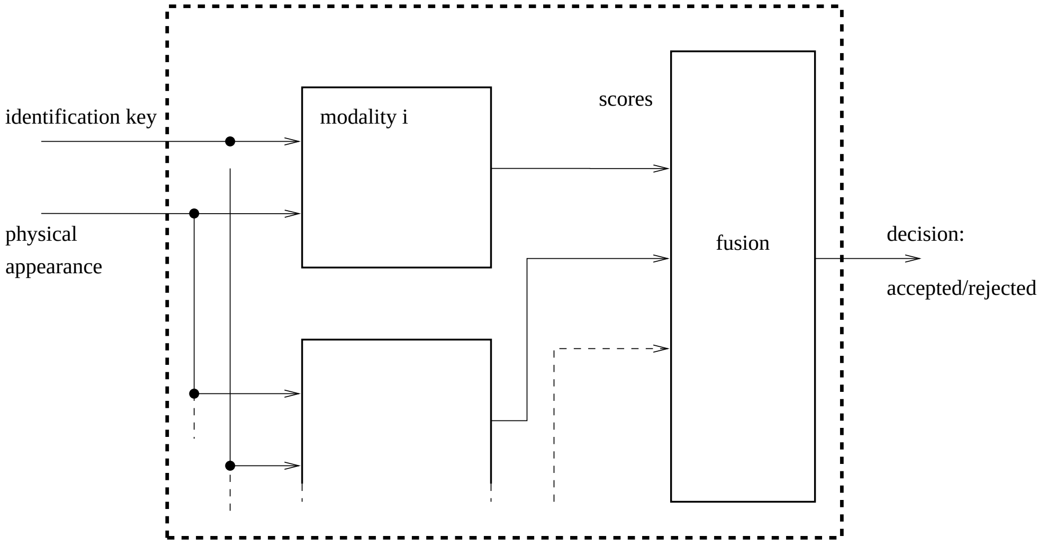


Figure 1: Multi-modal architecture

certain applications security is preferred (FAR small), in others client comfort (FRR small). A parameter to tune the FAR/FRR trade-off is therefore desired.

In the next section we present a classifier (fusion module) designed to take into account the constraints mentioned above.

# 3 Monotone two-class multi-linear classifier

## 3.1 Principle

We have developed a classifier determining regions in the $d$-dimensional space corresponding to the two classes, based on a combination of half-spaces. We call this classifier *multi-linear classifier* in reference to the use of several half-spaces, each one building a linear classifier.

The classifier training consists of a supervised phase in which the different half-spaces are determined by hyper-planes which optimally separate pairs of points of either class and where the regions generated by these half-spaces are labeled with the class identifier (accept, reject).

At testing, each data point from the test set is simply given the class label of the region it is belonging to.

## 3.2 Training

### 3.2.1 Overview

Given examples of the two classes, the goal is to find hyper-planes separating optimally all pairs of points of either class and to label the generated regions with the corresponding class identifier. Let's describe the samples available for training by the two sets:

- the set of *positive points* (representing the client claims) $\{\boldsymbol{a}^k\}_{k\in K} \in I\!R^d, |K|$ being the total number of positive points;

- the set of *negative points* (representing the client claims) $\{\boldsymbol{a}^l\}_{l\in L} \in I\!R^d, |L|$ being the total number of negative points;

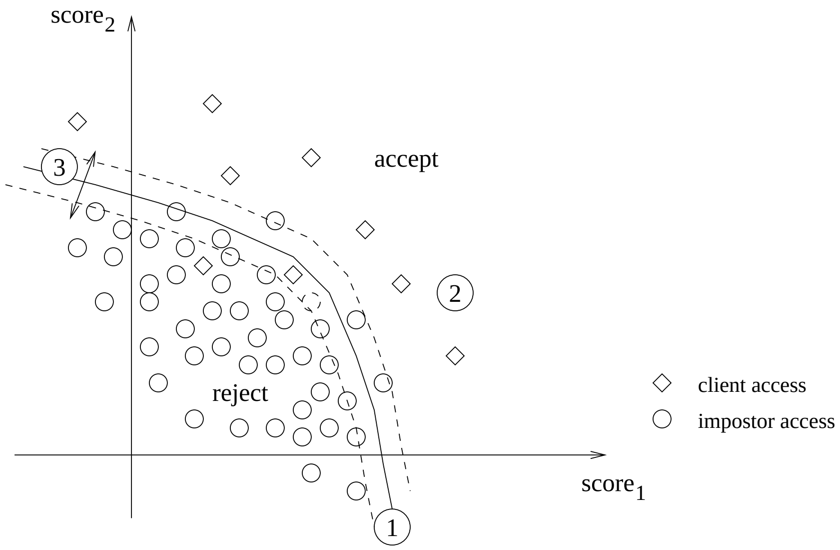


Figure 2: Particular classification problem: (1) monotonicity, (2) scarcity of client accesses for training, (3) tunable FAR/FRR trade-off

The training of the multi-linear classifier consists of:

**First step** Reduction of training samples;

**Second step** Determination of half-spaces;

**Third step** Class attribution to intersections of half-spaces.

Each of these steps is going to be detailed separately hereafter.

#### 3.2.2 Reduction of training samples

In a first step the classifier reduces the number of data points in the two classes by using the monotonicity hypothesis. In this specific case, the constraint of monotonicity implies that a given linear separator (*i.e.* a half-space):

- has a positive normal vector, which can be formally expressed as $w_i^s \geq 0, \forall i = 1, \ldots, d$;
- is considered to separate a particular pair of points only if the positive point (client) is on the positive side of the separator, and the negative point (impostor) on the negative side.

This monotonicity hypothesis allows a preprocessing of the input of the problem as follows: if there exists two points $\boldsymbol{x}$ and $\boldsymbol{y}$ in the positive set (respectively negative set) such that $x_i \leq y_i, \forall i = 1, \ldots, d$, the point $y$ (respectively $x$) can be suppressed from the input set.

As a result of this data reduction, only data points situated along the separation surface of the two classes are maintained. This technique reduces thus also the number of couples that can be formed consisting of one point from each class. These couples are the ones used in the next step.

#### 3.2.3 Determination of half-spaces

**Principle** The half-spaces are determined by **maximizing** a separability (discrimination) measure of point pairs.

The goal is thus to determine a set of $|S|$ half-spaces given by $(\boldsymbol{w}^s, w_0^s) \in I\!R^d \times I\!R, s \in S$, with the following property for the discrimination between two points. A given half-space $(\boldsymbol{w}^s, w_0^s)$ discriminates between two points $\boldsymbol{x}$,$\boldsymbol{y} \in I\!R^d$ if $(\boldsymbol{x}\boldsymbol{w}^s - w_0^s)$ and $(\boldsymbol{y}\boldsymbol{w}^s - w_0^s)$ are both non-zero and of opposite signs. Because of the monotonicity constraint, it will be considered that $(\boldsymbol{w}^s, w_0^s)$ discriminates between $\boldsymbol{x}$,$\boldsymbol{y}$ only if $\boldsymbol{y}\boldsymbol{w}^s - w_0^s < 0 < \boldsymbol{x}\boldsymbol{w}^s - w_0^s$, and in this case, the quality of this discrimination is given by the minimum of the module of these two values, *i.e.* by $\min\{\boldsymbol{x}\boldsymbol{w}^s - w_0^s, \, -\boldsymbol{y}\boldsymbol{w}^s + w_0^s\}$.

The total discrimination for the *whole* set of separators for each pair of points is simply the sum of the discrimination obtained for each separator. This total discrimination for a pair is defined as $\Delta$. A reference value for $\Delta$ is given by $\Delta_0$, defined as half of the minimal Euclidean distance between a pair of positive/negative points $(\boldsymbol{x}, \boldsymbol{y})$. This is the discrimination for $(\boldsymbol{x}, \boldsymbol{y})$ that would obtain a single half-space cutting orthogonally and at the middle, the segment $[\boldsymbol{x}, \boldsymbol{y}]$. It is obvious that the total number $|S|$ of half-spaces thus obtained will (amongst else) strongly depend on this user-defined value of $\Delta$. The greater this value of $\Delta$ is chosen to be, the greater the total number of half-spaces in the set will be. This dependency of the number of separators in the set on the choice of $\Delta$, can be observed in the example of Section 3.4 and its impact on the verification results is discussed in Section 5.5.

As we already have announced, we wish to be capable to introduce a *bias* in the classifier. This can be achieved by weighting differently the separation towards positive and negative points. Therefore the previous discrimination measure will be replaced by $\min\{\alpha(\boldsymbol{x}\boldsymbol{w}^s - w_0^s), \, -\boldsymbol{y}\boldsymbol{w}^s + w_0^s\}$, where $\alpha$ is any non-zero constant. It is clear that the value of $\alpha$ determines the bias that show the half-spaces with respect to a certain class. In Section 3.4 this "attraction tendency" can be observed and the impact of the choice of $\alpha$ on the verification results is studied in Section 5.5. The reference value for $\alpha$ is "1", which corresponds to no bias at all.

One can thus see that the number $|S|$ of half-spaces generated and the bias they show towards one of either classes, are governed by two user-defined parameters respectively called $\Delta$ and $\alpha$.

**Proposed hybrid approach** To solve this formal problem, we propose to use two successive phases: an iterative one followed by a global one. The purpose of the iterative phase is to generate iteratively a set of $|S|$ linear separators (coarse tuning). The subsequent global phase is then used to locally optimize this set of $|S|$ half-spaces (fine tuning).

**The iterative phase** In this phase the total separability $\Delta$ to be achieved is fixed (by the user) and using this value a first half-space is calculated. Subsequently, half-spaces are continued to be inserted iteratively, until the total discrimination $\Delta$ is reached *for each pair of points.* At each iteration $u$ the following problem has to be solved: given the two sets of points $\{\boldsymbol{a}^k\}_{k\in K}$, $\{\boldsymbol{a}^l\}_{l\in L} \subset I\!R^d$ and the half-spaces $(\boldsymbol{w}^s, w_0^s) \in I\!R^d \times I\!R, s = 1, \dots, u-1$ already determined before the current iteration, find $(\boldsymbol{w}^u, w_0^u)$, maximizing the following *iterative goal function*:

$$\text{maximize} \sum_{k\in K, l\in L} \min\{\Delta \, , \sum_{s\in S} \Delta_{kls}\} \tag{1}$$

$$\text{where} \quad \Delta_{kls} = \max\{0, \min\{\alpha(\boldsymbol{a}^k\boldsymbol{w}^s - w_0^s), -\boldsymbol{a}^l\boldsymbol{w}^s + w_0^s\}\} \tag{2}$$

$$\text{under the normalization conditions} \quad -1 \le w_i^s \le 1, \forall s \in S, \forall i = 0, \dots, d \tag{3}$$

$$\text{and with } S = \{1, \dots, u\}.$$

The advantage of this method is that the number of half-spaces need not to be fixed a priori. The disadvantage is that the different half-spaces are added sequentially to the total set of separators and once they have been entered they are not altered (fine-tuned) any more by the subsequent iterations.

As can be seen in equation (1), the maximal quality of the discrimination for a certain pair is limited to $\Delta$. This has explicitly been done to limit the influence of distant pairs (which are easy to separate) on the determination of the current half-space.

The iterative phase has been implemented using a gradient descent method. The computation of the gradient of the iterative goal function is detailed in Appendix A. The initial points for this method are obtained in a hybrid manner. Some of the initial points are, as is usually the case, chosen at random. However, a certain number of those initial points are found using a heuristic approach, *i.c.* by calculating a half-space that separates the $n$ worst discriminated pairs at a certain moment. These half-spaces are calculated using one of two simple classical linear classifiers: either a Fisher or a nearest-mean linear classifier [The89, DH73], depending on the convergence of the Fisher classifier. The number of random initialization points and the number $n$ of worst discriminated pairs at a certain moment can both be varied by the user, to allow for the generation of a bigger and/or different set of initialization half-spaces.

The reason why we have chosen this hybrid form of initialization is to be able to cope with the following phenomenon. After only a few iterations, the iterative goal function in equation (1) rapidly degenerates in this sense that it doesn't stay a smooth surface, where one can easily use a gradient descent method starting from a randomly chosen initialization point. Instead of the smooth initial surface, there soon appear very scarce and local peaks in the goal function. This is due to the very brutal non-linear behavior of our goal function, showing indeed a succession of *max* and *min* operators, which introduces discontinuities of the first kind. So to have more chances to place the initialization points at least somewhere in the neighborhood of the slopes of (one of) these peaks, the aforementioned heuristic with respect to the separation of the $n$ worst separated points is used. The appearance of these peaks can be clearly observed in the simple two dimensional example of Section 3.4.2. This simple heuristic approach guarantees by no means that the *global* optimum (maximum) of the iterative goal function for the current iteration is going to be reached at each iteration.

**Comment w.r.t. a smoother version of the iterative goal function** We did try to improve the degree of smoothness of our iterative goal function by replacing the max/min operators in equation (1)by a sigmoidal function such as the *atanh*, but this only improves the smoothness of the slopes of the peaks and it doesn't change at all the highly undesirable fact that this goal function rapidly shows very large plateaus where the gradient descent method has absolutely no chance of working. So this sigmoidal like function didn't improve the behavior of the iterative goal function drastically, but it did increase the computing time severely, so we decided to fall back to the original max/min type of goal function, adding the heuristic approach for finding useful initial points.

**The global phase** In this global phase, the number $|S|$ of separators is fixed a priori and the purpose of this phase is then to globally maximize the discrimination over all pairs of points by locally acting on all $|S|$ half-spaces at the same time (fine-tuning). The following *global goal function* has to be optimized: find $(\boldsymbol{w}^u, w_0^u)$, which

$$\text{maximize} \min_{k\in K, l\in L} \sum_{s\in S} \Delta_{kls} \tag{4}$$

where $\Delta_{kls}$is defined as in (2), under the constraints described in (3).

To try to optimize the set of $|S|$ half-spaces that has been found during the iterative phase, the global phase uses a new goal function, as can be seen when comparing equations (4) and (1). The main difference is that in the global phase we are optimizing the global separation for all pairs, which for a single pair is not limited any longer to the value of $\Delta$, as it was the case during the iterative phase.

The global phase has also been implemented using a gradient descent method. The computation of the gradient of the global goal function is detailed in Appendix B. An initial pair of points is selected at random at each iteration; if the total separation of this pair is above the current minimum, nothing is changed, otherwise, the parameters $w_i^s$ are modified in the direction of the gradient of the objective function given in (4). This gradient is calculated in the point where the goal function in equation (4) is minimal. If there are $|N|$ such points instead of one, then the gradient is calculated in each point. But in this global approach we can use only one general direction for optimizing all $|S|$ half-spaces at the same time. To be able to find this best direction, the following problem needs to be solved:

$$\text{Using all } |N| \text{ global gradient vectors: } \nabla^S_{glob_1}, \dots, \nabla^S_{glob_{|N|}} \in I\!R^{(d+1).|S|},$$

find an $\boldsymbol{x}^S \in I\!R^{(d+1).|S|}$ such that $\forall n \in N : \boldsymbol{w}^S + \eta.\boldsymbol{x}^S$ maximizes expression (4) for all minimal pairs.

Where $d$ is the number of modalities to be combined, $|S|$ is the number of half-spaces in the set,

$\boldsymbol{w} \in I\!R^{(d+1).|S|}$ is the vector that contains all $|S|$ half-spaces,

$|N|$ is the number of pairs with minimal separation and $\eta$ is any positive number.

To be able to solve this problem in an easy way, we have transformed it into an alternative form. In Appendix C it is shown that the preceding problem is equivalent with the following one:

$$\text{Find } \boldsymbol{x}^S \in I\!R^{(d+1).|S|}$$

$$\text{Such that } \forall n \in N,\ \boldsymbol{x}^S.\nabla^S_{glob_n} > 0 \text{ is maximal.}$$

This problem can now be solved easily using linear programming, since all constraints are purely linear [PTVF92].

#### 3.2.4 Class attribution to intersections of half-spaces

The resulting set of $|S|$ half-spaces after training induces a partition of the $d$ dimensional space. Each region of this partition is then coded by a word of $|S|$ bits, indicating its membership to each half-space. A "1" means the considered region is lying on the positive side of the considered separator and a "0" means on the negative side. Afterwards the label of one of either classes is attributed to each region, using the Logical Analysis of Data (LAD) [BHI+96] method.

One possibility offered by the flexibility of LAD is to attribute a "?" to a certain bit instead of a "1" or a "0", to express a certain doubt with respect to the classification. In our case we have decided to do this for the regions lying very close to (*i.e.* in a small zone determined by $\Delta_0$ along both sides of) a certain separator.

**Coding of training samples** In the binarization phase, a data point of the training set is characterized by a word of $|S|$ bits, according to its membership of a certain region of the partition.

**Labeling of the partition** After the binarization phase, one of either classes needs to be attributed to each region of the partition and this is done using LAD.

## 3.3 Testing

During testing the membership of each data point w.r.t. the $|S|$ half-spaces is calculated and each data point receives simply the class label of the region of the hyper-space it is lying in.

## 3.4 Synthetic two-dimensional example

### 3.4.1 Representation of the two classes

Figure 3 shows the two (synthetic) classes of positive and negative points that are used to explain the basic ideas and mechanisms explained so far.

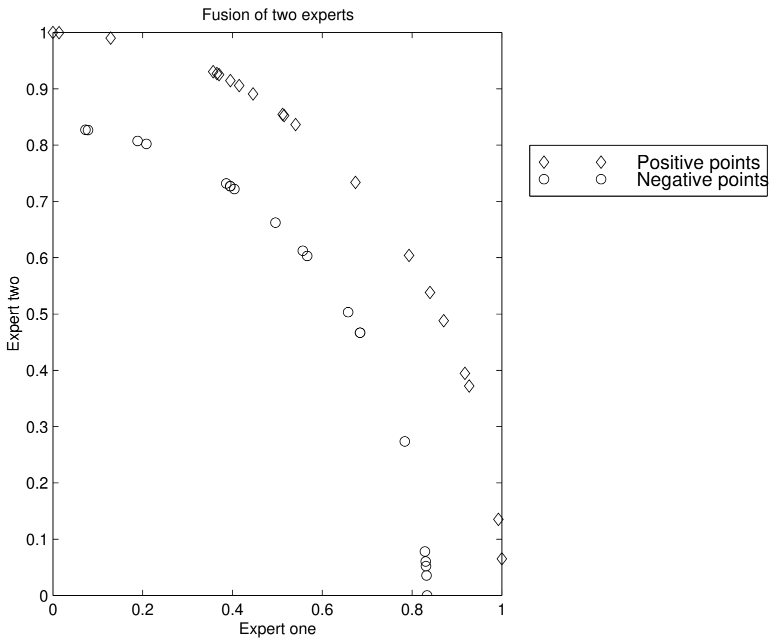


Figure 3: Simple two dimensional two class problem

### 3.4.2 Appearance of the goal-function

Figures 4 and 5 show the appearance of the iterative goal function (1) in the case of this simple example, after respectively two and five iterations. To be able to represent the three components $w_0^s, w_1^s$and $w_2^s$ of a half-space, the two dimensional components $w_1^s$and $w_2^s$ have been represented onto one axis by using the transformation "angle" $= \arccos(w_1^s) = \arcsin(w_2^s)$, the other axis being $w_0^s$. This transform has also the advantage that it satisfies automatically the normalization constraints (3).

It can be clearly seen by comparing Figures 4 and 5 that already after five iterations the goal function has enormous plateaus in which the classical gradient descent doesn't work.

### 3.4.3 Determination of half-spaces

Figure 6 shows the set of half-spaces that the multi-linear classifier has found in the case of this example ($|S| = 5$). This set of half-spaces has been found for the reference values for $\alpha$ and $\Delta$ and will be used as a reference case to be compared with the following Figures..

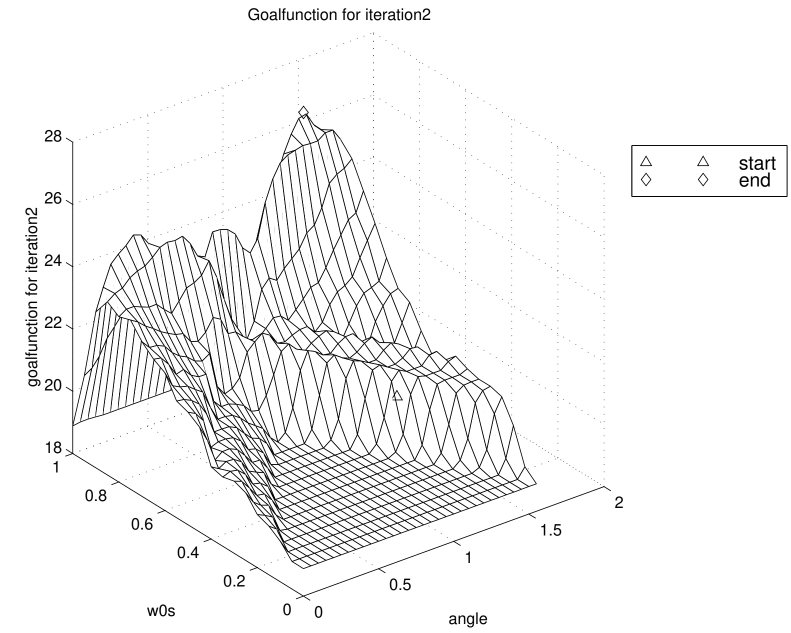


Figure 4: Example of the iterative goal function after two iterations

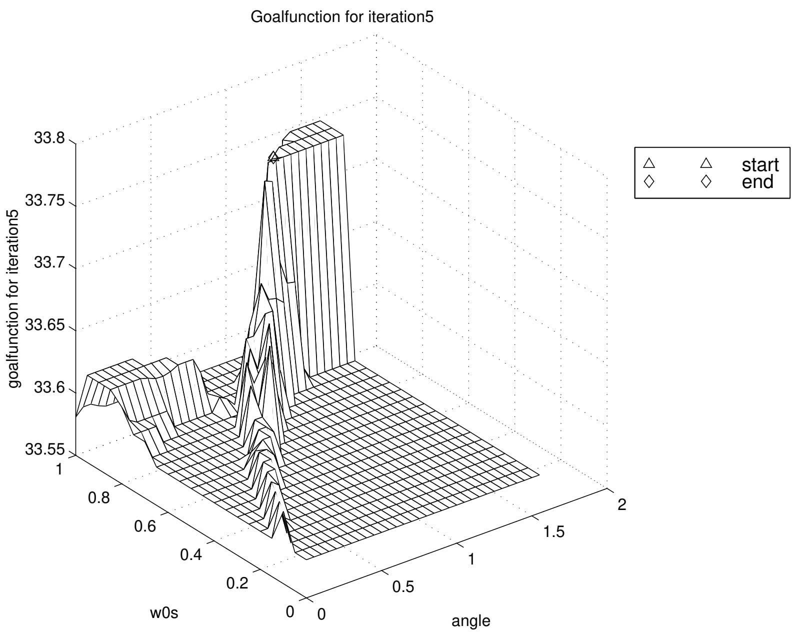


Figure 5: Example of the iterative goal function after five iterations

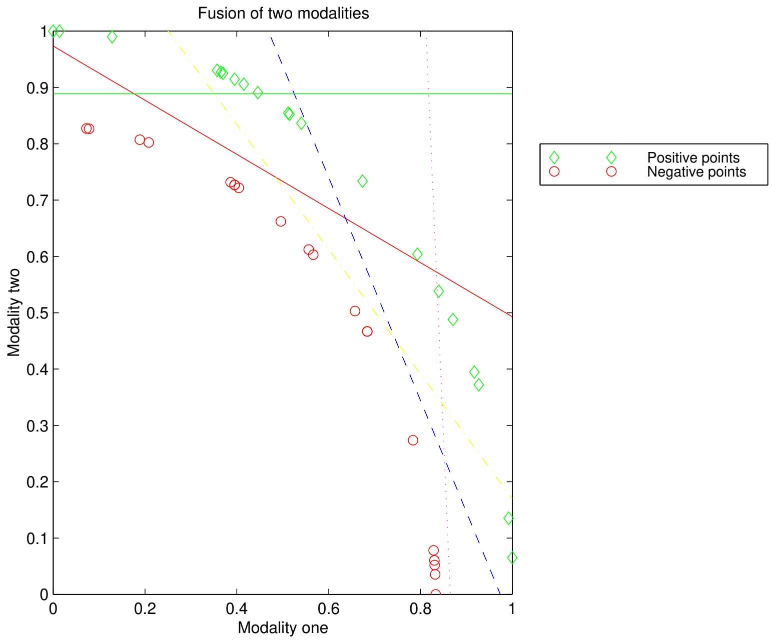


Figure 6: Set of half-spaces generated for $\alpha = 1$ and $\Delta = \Delta_0$

### 3.4.4 Influence of $\alpha$

Figure 7 and 8 show the influence of $\alpha$ on the *attraction tendency* of the set of half-spaces towards one of either classes. Figure 7 has been obtained for $\alpha = 0.9$ and Figure 8 for $\alpha = 1.1$. When $\alpha$ becomes smaller than the reference value, an attraction tendency towards the negative points can be observed and when $\alpha$ on the other hand becomes larger than the reference value, an attraction towards the positive points can be seen. Both these sets of half-spaces have been found for the reference value for $\Delta$. From the comparison of these two figures with our reference case, it can be seen that the value of $\alpha$ has also an influence on the *number* of half-spaces that are generated. When $\alpha$ is chosen smaller than the reference value, the number of half-spaces decreases w.r.t. our reference case ($|S| = 4 < 5$) and when $\alpha$ becomes larger than the reference, the number of half-spaces increases w.r.t. our reference case ($|S| = 6 > 5$).

This effect could have been expected since, when taking a closer look at equation (2), we see that $\alpha$ has a direct influence on the actually calculated discrimination. In the case the two classes are well separated, a value of $\alpha$ close to the reference value should generate the lowest number of half-spaces. The more $\alpha$ differs from the reference value, the more the half-spaces are approaching the points of one of either classes and the more half-spaces will therefore be needed to “zig-zag” around these points. This is a drawback of this method, since ideally spoken the number of half-spaces generated should only be influenced by $\Delta$. The interdependence of $\alpha$ and $\Delta$ makes it more difficult to fine-tune the method for a specific application.

### 3.4.5 Influence of $\Delta$

Figure 9 and 10 show the influence of $\Delta$ on the number of generated half-spaces. Figure 9 has been obtained for $\Delta = 0.25 * \Delta_0$ and Figure 10 for $\Delta = 4 * \Delta_0$. These sets of half-spaces have both been found for the reference value for $\alpha$. When $\Delta$ becomes smaller than the reference value, the number of half-spaces generated decreases w.r.t. our reference case ($|S| = 2 < 5$). When on the other hand $\Delta$ becomes larger than the reference value, the number of half-spaces generated increases w.r.t. our

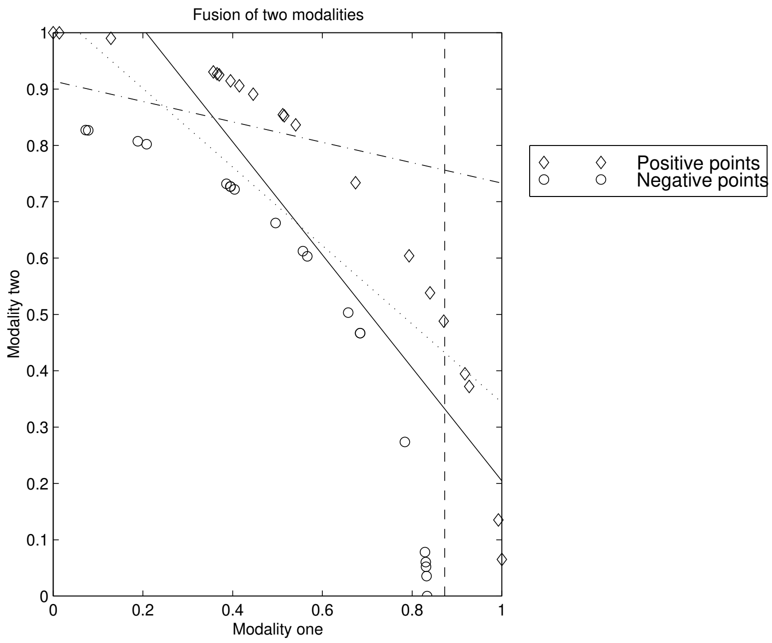


Figure 7: Set of half-spaces generated for $\alpha = 0.9$ and $\Delta = \Delta_0$

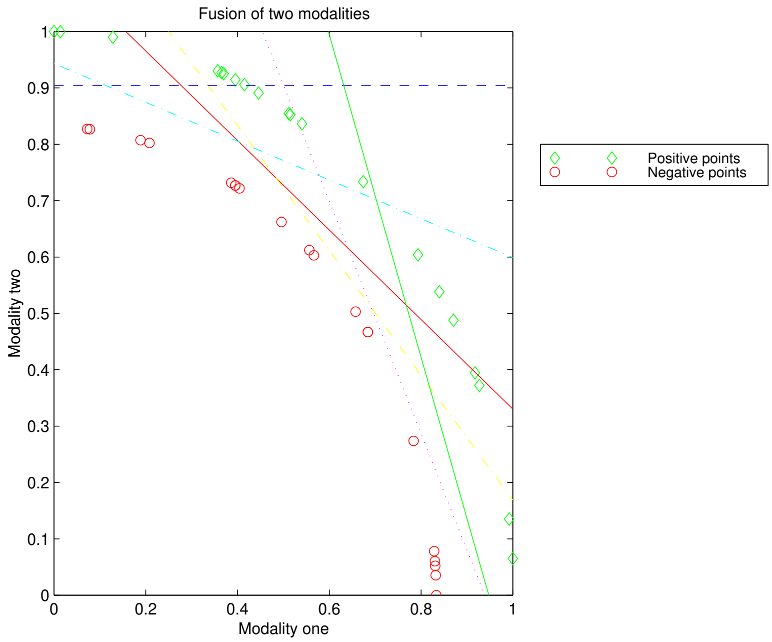


Figure 8: Set of half-spaces generated for $\alpha = 1.1$ and $\Delta = \Delta_0$

reference case ($|S| = 19 < 5$).

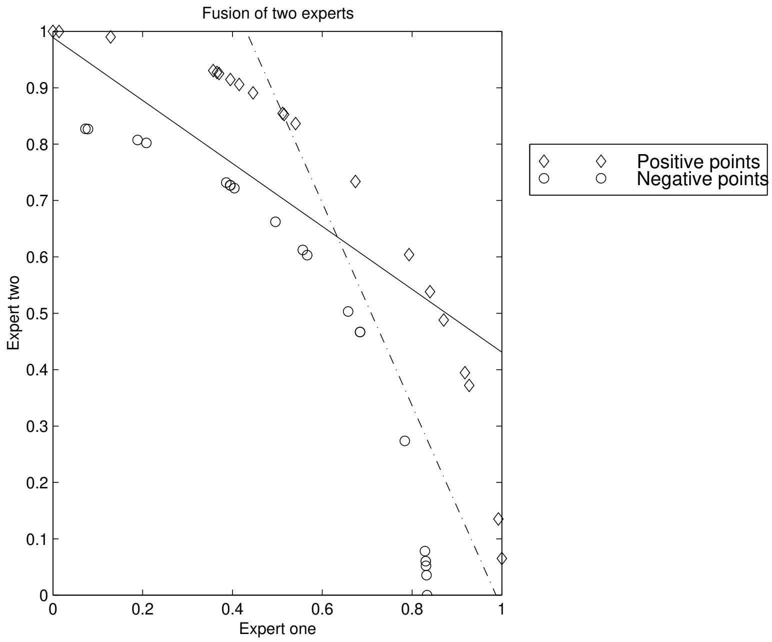


Figure 9: Set of half-spaces generated for $\alpha = 1$ and $\Delta = 0.25 * \Delta_0$

## 3.5 Discussion

It is important to realize that there shouldn't be too many half-spaces. Indeed, the ideal number of separators results from the classical trade-off between the *robustness* and the *sensitivity* of a classifier. In the specific case of our multi-linear classifier this compromise can be explained as follows. The more half-spaces there are, the larger the number of regions of the partition of the $d$ dimensional space. This means that the number of training data points that are likely to fall in a single region becomes smaller. This means that the attribution of the class label to the different regions is going to be more and more influenced by isolated training data points. This makes the classifier on the one hand more sensitive, but on the other hand at the same time also less robust. This duality is explained below:

**Greater sensitivity** If those isolated training data points are representative for the real (unknown) characteristics of the rest of the population then the classifier did a good job on capturing this level of detail.

**Smaller robustness** If those isolated training data points are outliers who's characteristics are only marginal related to those of the rest of the population, then the classifier is going to accumulate a lot of errors.

This duality can also be expressed in terms of "over-training" and "under-training" the classifier. Over-training the classifier means that we are in fact modeling the noise on the training data, which leads to a bad generalization capability. This happens when we generate a lot of half-spaces but the isolated training points are in fact outliers or extreme values. Under-training the classifier means that we are not modeling enough significant variations in the training data, which means that we are generalizing too much. This happens when we generate only a small number of half-spaces such that meaningful isolated training data points are grouped with the bulk of the training data.

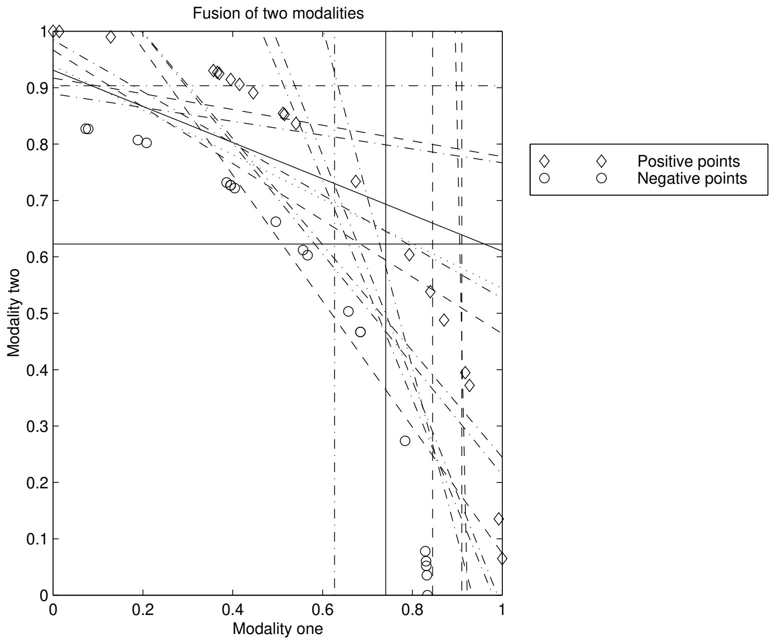


Figure 10: Set of half-spaces generated for $\alpha = 1$ and $\Delta = 4 * \Delta_0$

This compromise can be fixed using a supplementary data set (which can be seen as a kind of validation set).

# 4 Classifier implementation

The implementation of this method is done in Matlab5 and the algorithm effectively makes use of the multidimensional matrices that are one of the differences between Matlab4 and Matlab5. Without having analyzed the intrinsic complexity of the algorithm, we have observed that the execution time of the method is extremely long. A typical computing time on a LINUX PC (Pentium Pro at 200 MHz) for an application including three experts, ten initial points, three half-spaces which are generated, and typically 25 data points in the client class and 120 data points in the impostor class *after* the data reduction step, is one day.

# 5 Fusion experiments

To test and validate the concept of this multi-linear classifier, we have chosen to use to experiment with multi-modal data coming from the M2VTS project [PV, PV97a].

## 5.1 M2FDB audio-visual person database

The M2FDB multi-modal database comprises 37 different persons and provides 5 shots for each person. These shots were taken at one week intervals. During each shot, people were asked (1) to count from "0" to "9" in French (which was the native language for most of the people) and (2) to rotate their head from 0 to -90 degrees, back to 0 and further to +90 degrees, and finally back again to 0 degrees. The most difficult shot to recognize is the $5^{th}$ shot. This shot mainly differs from the others because of face "variations" (head tilted, eyes closed, different hair style, presence of a hat/scarf, ... ), voice variations or shot imperfections (poor focus, different zoom factor, poor voice signal to noise ratio, ... ).

Taking into account the specificity of our problem (*i.e.* combining outputs of several modalities) we are not going to use this $5^{th}$ shot, since we are not interested in developing individual powerful modalities that work well even under these extreme conditions as presented by shot number 5.

## 5.2 Individual verification modalities

### 5.2.1 Experimentation protocol

Due to the scarcity of the data, individual modalities were run on the M2FDB database according to the *leave-one-out* protocol [DK82]. In our case this means that three shots have been used for training purposes (one shot has been left out for test purposes) containing 36 persons (one person has been left out for impostor tests). Using this protocol, 37×4=148 different experiments can be defined.

To identify the different experiments, labeling conventions of the form DD_CC have been used, where:

- the first two digits DD specify the *shot* left out;
- the last two characters CC specify the initials of the *person* left out.

### 5.2.2 Modalities to be fused

In the fusion experiments the following three modalities have been combined:

- a voice modality from the *Dalle Molle Institute for Perceptual Artificial Intelligence (CH)* (IDIAP), using a text-dependent Hidden Markov Model method.
- a face (profile view) modality from the *Universite Catholique de Louvain (BE)* (UCL), based on a Chamfer profile matching method [PV97b].
- a face (frontal view) modality resulting from the collaboration between the *Aristotle University of Thessaloniki (GR)* (AUT) and the *Ecole Polytechnique Federale de Lausanne (CH)* (EPFL), based on a Morphological Dynamic Link Architecture method [KPF+97].

As it already has been stated in Section 2.2, all of these modalities produce a score which is supposed to be monotone and which lies in the interval [0,1].

The training and testing data sets contain the following information (example is given using the label 01_BP):

- the scores obtained on the training set consisting of the shots 2, 3 and 4 (shot 1 left out) which contain each 36 persons (person BP left out) and this training set allows us to generate:
  - 3 shots × 36 persons × 1 client claim per person and per shot = 108 client scores;
  - 3 shots × 36 persons × 35 impostor claims per person and per shot = 3780 impostor scores;
- the scores obtained on the test set consisting of the shot number 1 (which contains also the person BP) and this test set is used to perform:
  - 36 client claims on the same 36 persons contained in the training set (used for calculating the *FRR*);
  - 36 impostor claims using the BP images in shot 1 on the 36 persons contained in the training set (used for calculating the *FAR*).

## 5.3 Protocol of fusion experiments

Ideally, three different data sets are needed for training and testing the modalities and the fusion module. The first data set is a training set and is used by each modality to model the different persons. The second data set is used to train the fusion module and the third one is used to test the modalities and/or the fusion module. If there is not enough data available to make this possible, the following errors will be introduced:

- if the test data is the same as the training data, performances will be overestimated. This is true for both the modalities and the fusion module. This is of course due to the fact that the modalities and the fusion module will generate the best results for the same data they have been trained on.
- if the training data for the modalities is the same as for the fusion module, the fusion module will be under performing. The reason for this is that the fusion module doesn't get enough information. Indeed, in the extreme case of modalities that perform perfectly on their training data, the outcome of such a modality would be either 0 or 1, which leaves the fusion module with the arbitrary choice of setting the threshold somewhere in between.

In our case this last situation is occurring since we use the same data for training the different modalities and the fusion module.

Fusion has been performed for five different *experiments*. Four of these experiments have been chosen to allow each *shot* to be the one left out. To allow for some possible variation in the experiments due to an eventual influence of the composition of the training database, these four experiments were each having another *person* left out. To try to capture an eventual influence of the choice of this person left out, an additional fifth experiment has been introduced leaving out one of the four already left out persons, but this time leaving out another shot. To try to clarify this explicitly, these five experiments have been labeled as follows, using the convention introduced in Section 5.2.1:

1. 01_BP;
2. 01_XB;
3. 02_FO;
4. 03_BP;
5. 04_SP.

Several tests have been performed and each *test configuration* was specified by attributing specific values for both $\alpha$ and $\Delta$. For each test configuration the same five different experiments have been carried out.

## 5.4 Results

### 5.4.1 Single modalities

In order to have verification results from single modalities comparable with the verification results of the fusion experiment, the results for the different single modalities have been obtained using a *global* threshold (*i.e.* the same for all persons) that has been calculated on the training database using the *Equal Error Rate* (EER) criterion. This EER criterion defines the threshold as the one for which both error rates are the same (*i.e.* FRR = FAR). Please note that these results have been obtained on test sets containing only 36 data points and this as well for the client tests as for the impostor tests.

Table 1 shows the verification results for each experiment and for each modality. The last column gives, as an indication, for each modality the mean value over all five experiments. These results are represented in the classical way, *i.e.* by expressing FRR and FAR (both in %).

| Modality | 01_BP | | 01_XB | | 02_FO | | 03_BP | | 04_SP | | Mean value | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR |
| IDIAP | 27.8 | 0.0 | 30.6 | 0.0 | 25.0 | 0.0 | 27.8 | 0.0 | 36.1 | 0.0 | 29.5 | 0.0 |
| UCL | 11.1 | 27.8 | 11.1 | 36.1 | 11.1 | 38.9 | 11.1 | 27.8 | 11.1 | 22.2 | 11.1 | 30.6 |
| AUT & EPFL | 5.6 | 5.6 | 5.6 | 19.4 | 2.8 | 8.3 | 5.6 | 5.6 | 5.6 | 0.0 | 5.0 | 7.8 |

Table 1: Verification results in % on test set for single modalities

### 5.4.2 Fused modalities

Table 2 shows the verification results for each experiment and for different values of $\alpha$, while $\Delta$ was given the reference value $\Delta_0$. Table 3 shows the verification results for each experiment and for different values of $\Delta$, while $\alpha$ was given the reference value 1. Table 4 shows the verification results for each experiment and for different values of $\alpha$ and $\Delta$, both different from their respective reference values. Table 5 shows the number of half-spaces generated per test.

| $\alpha$ | $\Delta$ | 01_BP | | 01_XB | | 02_FO | | 03_BP | | 04_SP | | Mean value | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR |
| 0.80 | $1.00 * \Delta_0$ | 22.2 | 0.0 | 30.6 | 0.0 | 2.8 | 0.0 | 41.7 | 0.0 | 13.9 | 0.0 | 22.2 | 0.0 |
| 0.85 | $1.00 * \Delta_0$ | 16.7 | 0.0 | 25.0 | 0.0 | 22.2 | 0.0 | 19.4 | 0.0 | 11.1 | 0.0 | 18.9 | 0.0 |
| 0.90 | $1.00 * \Delta_0$ | 33.3 | 0.0 | 25.0 | 0.0 | 2.8 | 0.0 | 2.8 | 0.0 | 5.6 | 0.0 | 13.9 | 0.0 |
| 0.95 | $1.00 * \Delta_0$ | 38.9 | 0.0 | 5.6 | 0.0 | 25.0 | 0.0 | 47.2 | 0.0 | 22.2 | 0.0 | 27.8 | 0.0 |
| 1.00 | $1.00 * \Delta_0$ | 41.7 | 0.0 | 5.6 | 0.0 | 27.8 | 0.0 | 30.6 | 0.0 | 36.1 | 0.0 | 28.4 | 0.0 |
| 1.10 | $1.00 * \Delta_0$ | 38.9 | 0.0 | 33.3 | 0.0 | 27.8 | 0.0 | 13.9 | 0.0 | 36.1 | 0.0 | 30.0 | 0.0 |

Table 2: Verification results in % on test set for fused modalities as a function of $\alpha$

| $\alpha$ | $\Delta$ | 01_BP | | 01_XB | | 02_FO | | 03_BP | | 04_SP | | Mean value | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR |
| 1.00 | $2.00 * \Delta_0$ | 41.7 | 0.0 | 27.8 | 0.0 | 33.3 | 0.0 | 38.9 | 0.0 | 35.4 | 0.0 | 35.4 | 0.0 |
| 1.00 | $1.00 * \Delta_0$ | 41.7 | 0.0 | 5.6 | 0.0 | 27.8 | 0.0 | 30.6 | 0.0 | 36.1 | 0.0 | 28.4 | 0.0 |
| 1.00 | $0.67 * \Delta_0$ | 19.4 | 0.0 | 33.3 | 0.0 | 19.4 | 0.0 | 16.7 | 0.0 | 13.9 | 0.0 | 20.5 | 0.0 |
| 1.00 | $0.50 * \Delta_0$ | 5.6 | 0.0 | 19.4 | 0.0 | 11.1 | 0.0 | 19.4 | 0.0 | 38.9 | 0.0 | 18.9 | 0.0 |
| 1.00 | $0.40 * \Delta_0$ | 30.6 | 0.0 | 30.6 | 0.0 | 38.9 | 0.0 | 47.2 | 0.0 | 16.7 | 0.0 | 32.8 | 0.0 |
| 1.00 | $0.25 * \Delta_0$ | 22.2 | 0.0 | 44.4 | 0.0 | 41.7 | 0.0 | 25.0 | 0.0 | 16.7 | 0.0 | 30.0 | 0.0 |

Table 3: Verification results in % on test set for fused modalities as a function of $\Delta$

## 5.5 Result analysis

### 5.5.1 General

Analyzing the results obtained *after fusion*, one can see that the FAR is in our case always equal to zero. This could indicate that the generated hyper-planes are lying too close to the client prototypes.

| $\alpha$ | $\Delta$ | 01_BP | | 01_XB | | 02_FO | | 03_BP | | 04_SP | | Mean value | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR | FRR | FAR |
| 0.85 | $0.67 * \Delta_0$ | 13.9 | 0.0 | 5.6 | 0.0 | 5.6 | 0.0 | 5.6 | 0.0 | 5.6 | 0.0 | 7.3 | 0.0 |
| 0.85 | $0.50 * \Delta_0$ | 13.9 | 0.0 | 8.3 | 0.0 | 13.9 | 0.0 | 11.1 | 0.0 | 5.6 | 0.0 | 10.6 | 0.0 |
| 0.90 | $0.67 * \Delta_0$ | 5.6 | 0.0 | 8.3 | 0.0 | 5.6 | 0.0 | 2.8 | 0.0 | 11.1 | 0.0 | 6.7 | 0.0 |
| 0.90 | $0.50 * \Delta_0$ | 2.8 | 0.0 | 19.4 | 0.0 | 33.3 | 0.0 | 19.4 | 0.0 | 44.4 | 0.0 | 23.9 | 0.0 |
| 0.95 | $0.80 * \Delta_0$ | 22.2 | 0.0 | 33.3 | 0.0 | 2.8 | 0.0 | 5.6 | 0.0 | 5.6 | 0.0 | 13.9 | 0.0 |
| 0.95 | $0.67 * \Delta_0$ | 11.1 | 0.0 | 5.6 | 0.0 | 2.8 | 0.0 | 2.8 | 0.0 | 5.6 | 0.0 | 5.6 | 0.0 |
| 0.95 | $0.71 * \Delta_0$ | 16.7 | 0.0 | 5.6 | 0.0 | 19.4 | 0.0 | 13.9 | 0.0 | 5.6 | 0.0 | 12.2 | 0.0 |
| 0.95 | $0.50 * \Delta_0$ | 2.8 | 0.0 | 16.7 | 0.0 | 16.7 | 0.0 | 11.1 | 0.0 | 5.6 | 0.0 | 10.6 | 0.0 |
| 1.00 | $1.00 * \Delta_0$ | 41.7 | 0.0 | 5.6 | 0.0 | 27.8 | 0.0 | 30.6 | 0.0 | 36.1 | 0.0 | 28.4 | 0.0 |

Table 4: Verification results in % on test set for fused modalities as a function of $\alpha$ and $\Delta$

| $\alpha$ | $\Delta$ | 01_BP | 01_XB | 02_FO | 03_BP | 04_SP | Mean value |
|---|---|---|---|---|---|---|---|
| | | $S$ | $S$ | $S$ | $S$ | $S$ | $S$ |
| 0.80 | $1.00 * \Delta_0$ | 3 | 3 | 3 | 4 | 4 | 3.4 |
| 0.85 | $1.00 * \Delta_0$ | 4 | 3 | 4 | 4 | 3 | 3.6 |
| 0.85 | $0.67 * \Delta_0$ | 3 | 2 | 2 | 2 | 2 | 2.2 |
| 0.85 | $0.50 * \Delta_0$ | 2 | 2 | 3 | 2 | 2 | 2.2 |
| 0.90 | $1.00 * \Delta_0$ | 4 | 3 | 2 | 2 | 2 | 2.6 |
| 0.90 | $0.67 * \Delta_0$ | 1 | 2 | 2 | 2 | 2 | 1.8 |
| 0.90 | $0.50 * \Delta_0$ | 1 | 2 | 2 | 2 | 4 | 2.2 |
| 0.95 | $1.00 * \Delta_0$ | 4 | 3 | 4 | 5 | 3 | 3.8 |
| 0.95 | $0.80 * \Delta_0$ | 3 | 3 | 2 | 2 | 2 | 2.4 |
| 0.95 | $0.67 * \Delta_0$ | 2 | 2 | 2 | 2 | 2 | 2.0 |
| 0.95 | $0.71 * \Delta_0$ | 2 | 2 | 2 | 2 | 2 | 2.0 |
| 0.95 | $0.50 * \Delta_0$ | 2 | 2 | 3 | 2 | 2 | 2.2 |
| 1.00 | $2.00 * \Delta_0$ | 6 | 6 | 5 | 7 | 6 | 6.0 |
| 1.00 | $1.00 * \Delta_0$ | 4 | 3 | 4 | 4 | 4 | 3.8 |
| 1.00 | $0.67 * \Delta_0$ | 2 | 3 | 2 | 2 | 2 | 2.2 |
| 1.00 | $0.50 * \Delta_0$ | 1 | 2 | 2 | 2 | 2 | 1.8 |
| 1.00 | $0.40 * \Delta_0$ | 2 | 2 | 2 | 2 | 2 | 2.0 |
| 1.00 | $0.25 * \Delta_0$ | 1 | 2 | 2 | 2 | 3 | 2.0 |
| 1.10 | $1.00 * \Delta_0$ | 4 | 3 | 4 | 2 | 3 | 3.2 |

Table 5: The number $S$ of half-spaces generated as a function of $\alpha$ and $\Delta$

Another general observation is that there is a great spread in the results obtained over the five experiments in almost each configuration. This can be, amongst else, attributed to the initialization procedure. The starting points found by the used heuristics do not always include the best possible initial point. We have observed that, performing several times the same experiment using different initialization points, sometimes one of the randomly chosen starting points gives the best optimization of the goal function. This means that the number of iterations and thus the number of half-spaces that are generated depend on the initial conditions. This fact alone is already a major drawback for this method. Other factors are of course the differences between persons and the fact that using only a very limited number of tests, the confidence intervals for the errors are relatively large.

#### 5.5.2 Influence of $\alpha$

The influence of $\alpha$ is shown in Table 2. For values of $\alpha$ greater than one we observe an increase of the FRR, as could be expected. By using values of $\alpha$ smaller than one we first see as expected a decrease of the FRR, but when $\alpha$ reaches the value of 0.85, the FRR starts increasing again. Normally when the half-spaces lean more and more towards the impostor prototypes, we would expect a further decrease of the FRR and a gradual increase of the FAR. That this doesn't happen might be due to the fact that a change in $\alpha$ not only modifies the position, but also the number of half-spaces (see section 3.4.4). Other possible explanations are:

- the impostor and/or client prototypes determined during the training are not really representative for the client and/or impostor accesses made during testing;
- the monotonicity hypothesis is not valid.

#### 5.5.3 Influence of $\Delta$

The influence of $\Delta$ is shown in Table 3. Intuitively we could expect an increase in performance when the number of half-spaces becomes larger, since the LAD method seems to get more information for labeling the different section of the partition of the hyper-space. But as the number of half-spaces increases, the number of sections in that partition also increases and since in this method we are using only very few training data, the population of these sections will be getting sparse very rapidly (see Section 3.5). This phenomenon is in our opinion at the basis of what we observe in Table 3. When $\Delta$ increases, the FRR also increases and when $\Delta$ decreases the FRR decreases until a certain point $(0.50 * \Delta_0)$ from where on the FRR starts increasing again. This last phenomenon can be explained by the fact that when $\Delta$ gets too small, the corresponding half-space(s) don't have to be "very good" (the stop criterion for each iteration is reached sooner), which obviously will lead to more errors.

#### 5.5.4 Combined influence of $\alpha$ and $\Delta$

The combined influence of $\alpha$ and $\Delta$ is shown in Table 4. These results indicate that this method can have a very high performance. Indeed, the best results of the multi-linear classifier outperform those of the best single modality. The main problem with this is the fact that these "best results" are obtained for specific values of the parameters $\alpha$ and $\Delta$ and unfortunately there is no way of knowing *a priori* which values to give to these parameters for optimizing the classifier in a certain application. Furthermore the method seems to be rather sensitive to the "correct" values for these two parameters. This means that the only possibility to find "good" values for $\alpha$ and $\Delta$ is to try out a relatively large number of different values for both parameters, observing results on a *validation set*. However one should take into account that this iterative "trial and error" procedure will be time consuming and is not easy to conduct since there exists an interdependency between the two parameters.

# 6 Conclusions

Looking at the best results obtained, the multi-linear classifier intrinsically has the possibility to become a performing fusion method. When analyzing these results one should however not forget to take into account the following facts which have an unfavorable effect on the results:

- we have used the same set of half-spaces for each person (*i.e.* a global approach);
- the performances obtained here are a lower bound of what can be obtained in a real application since in this report we have used the same training data for the modalities as for the decision module, which implies that the decision module doesn't get the maximal amount of information.

At this moment the major drawbacks of this method still are:

- the dependency of the results on the initial conditions,
- the problems linked with the determination of the best values for the two parameters $\alpha$ and $\Delta$,
- the long computing time.

# 7 Future work

A lot of work remains to be done to completely analyze the behavior and the performance of this multi-linear classifier, compared with classical and popular classifiers such as for instance the Multi Layer Perceptron (MLP). The main axis for future work and research are the following ones:

- Analyze the intrinsic complexity of the algorithm;
- Minimize the computing time by compiling the Matlab m-files and by adapting the algorithm;
- Find better heuristics which would allow to avoid the use of random initialization points;
- Include a test on the data to check whether the monotonicity hypothesis on the individual modalities is verified;
- Analyze the validity of the monotonicity constraint for the fusion module by comparing the half-spaces found by the first part of a normally trained MLP (*i.e.* using back-propagation) and the half-spaces found by our multi-linear classifier;
- Analyze the trade-off between the robustness and the sensitivity of the multi-linear classifier (see the discussion in Section 3.5);
- Verify our assumption that the multi-linear classifier needs less training data than an MLP (comparison of verification results of both methods using the same (amount of) training data);
- Use Genetic Algorithms to find the global optimum of the goal function [SV96];
- Compare the multi-linear classifier with the first part of an MLP (the part containing the input layer and the hidden layer), using the same number of neurons in the hidden layer of the MLP than the number of half-spaces we used in our set;

# 8 Acknowledgment

This research has been carried out at the Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP), Switzerland, in the framework of the M2VTS project granted by the European Community ACTS program and by the Swiss Federal Office for Education and Science.

## References


[Ant95] R. T. Antony. *Principles of Data Fusion Automation.* Artech House Publishing, 1995.

[Ben95] Y. Bennani. A modular and hybrid connectionist system for speaker identification. *Neural Computation*, 7, 1995.

[BHI+96] E. Boros, P. L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, and I. Muchnik. An implementation of logical analysis of data. IDIAP-RR 5, Institut Dalle Molle d'Intelligence Artificielle Perceptive, Martigny, Switzerland, 1996.

[Das94] B. V. Dasarathy. *Decision Fusion.* IEEE Computer Society Press, 1994.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

[DK82] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice-Hall Inc., 1982.

[Kle93] L. A. Klein. *Sensor and Data Fusion Concepts and Applications*, volume 14 of *Tutorial Texts Series.* SPIE Optical Engineering Press, Washington, 1993.

[KPF+97] C. Kotropoulos, I. Pitas, S. Fischer, B. Duc, and J. Bigün. Face authentication using morphological dynamic link architecture. In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA'97)*, Lecture Notes in Computer Science. Springer Verlag, 1997.

[PTVF92] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C.* Cambridge University Press, second edition, 1992.

[PV] S. Pigeon and L. Vandendorpe. The M2VTS database . `http://www.tele.ucl.ac.be/M2VTS`.

[PV97a] S. Pigeon and L. Vandendorpe. The m2vts multi-modal face database. In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA'97)*, Lecture Notes in Computer Science. Springer Verlag, 1997.

[PV97b] S. Pigeon and L. Vandendorpe. Profile authentication using an optimized chamfer matching algorithm. In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA'97)*, Lecture Notes in Computer Science. Springer Verlag, 1997.

[SV96] J. Sklansky and M. Vriesenga. Genetic selection and neural modeling of piecewise-linear classifiers. *International Journal of Pattern Recognition and Artificial Intelligence*, 10(5):587–612, 1996.

[The89] C. W. Therrien. *Decision Estimation and Classification; An Introduction to Pattern Recognition and Related Topics.* Wiley, 1989.

[Tre68] H. L. Van Trees. *Detection, Estimation and Modulation Theory*, volume 1. John Wiley & Sons, New York, 1968.

[WL90] E. L. Waltz and J. Llinas. *Multisensor Data Fusion.* Artech House, 1990.

# A Derivation of the iterative goal function

The iterative goal function has been defined in expression (1). To simplify this expression, we introduce the following notations:

$$\tilde{\boldsymbol{w}}^s = \begin{pmatrix} \boldsymbol{w}^s \\ w_0^s \end{pmatrix},\ \tilde{\boldsymbol{a}}^k = \begin{pmatrix} \boldsymbol{a}^k \\ -1 \end{pmatrix},\ \tilde{\boldsymbol{a}}^l = \begin{pmatrix} \boldsymbol{a}^l \\ -1 \end{pmatrix} \in I\!R^{d+1}$$

$$\text{with } \boldsymbol{w}^s = \begin{pmatrix} w_1^s \\ \vdots \\ w_d^s \end{pmatrix},\ \boldsymbol{a}^k = \begin{pmatrix} a_1^k \\ \vdots \\ a_d^k \end{pmatrix},\ \boldsymbol{a}^l = \begin{pmatrix} a_1^l \\ \vdots \\ a_d^l \end{pmatrix} \in I\!R^d,\ s \in \{1, \ldots, |S|\}$$

In these expressions $d$ is the number of modalities, $|S|$ is the number of half-spaces in the set, $k$ is the number of positive and $l$ the number of negative points, $\tilde{\boldsymbol{w}}^s$ is the half-space being added during the current iteration $s$.

Furthermore we split $\Delta_{kls}$ into two parts: a first part $\Delta_{kl}$ that represents the separation on each pair $(k, l)$ obtained during the previous iterations $\{1, \ldots, s-1\}$ (note that this part is independent of the currently added half-space $\tilde{\boldsymbol{w}}^s$), taking into account the "clipping" effect induced by the max operator in the expression (1) and a second part $\delta_{kls}$ that represents the dynamic portion of the separation, added during the current iteration $s$ (this part depends of course of the current half-space $\tilde{\boldsymbol{w}}^s$). This convention can be written as follows:

$$\Delta_{kls}(\tilde{\boldsymbol{w}}^s) = \Delta_{kl} + \delta_{kls}(\tilde{\boldsymbol{w}}^s)$$

Using these new notations, the iterative goal function for the iteration $s$ becomes:

$$\text{goal}_{iter} = \sum_{k,l} \min\{\Delta, \Delta_{kl} + \delta_{kls}(\tilde{\boldsymbol{w}}^s)\},$$

$$\text{where } \delta_{kls}(\tilde{\boldsymbol{w}}^s) = \max\{0, \min\{\alpha.\delta_k(\tilde{\boldsymbol{w}}^s), -\delta_l(\tilde{\boldsymbol{w}}^s)\}\},$$

$$\text{with } \delta_x(\tilde{\boldsymbol{w}}^s) = \tilde{\boldsymbol{w}}^s.\tilde{\boldsymbol{a}}^x, x \in \{k, l\}.$$

Rewriting this goal function to eliminate the max and the min gives:

$$\text{goal}_{iter} = \left( \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} \geq \Delta}} \Delta + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta}} \Delta_{kl} + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta}} \delta_{kls}(\tilde{\boldsymbol{w}}^s) \right)$$

Replacing the first two terms (which, as already has been stated, are independent of $\tilde{\boldsymbol{w}}^s$) respectively by $A$ and $B$, the expression becomes:

$$\text{goal}_{iter} = \left( A + B + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta}} \delta_{kls}(\tilde{\boldsymbol{w}}^s) \right)$$

Replacing $\delta_{kls}(\tilde{\boldsymbol{w}}^s)$ by its expression and eliminating the max operator leads to the following expression:

$$\text{goal}_{iter} = \left( A + B + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } \min\{\alpha.\delta_k, -\delta_l\} > 0}} \min\{\alpha.\delta_k(\tilde{\boldsymbol{w}}^s), -\delta_l(\tilde{\boldsymbol{w}}^s)\} \right)$$

Eliminating the min operator gives us then:

$$\text{goal}_{iter} = \left( A + B + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < \alpha.\delta_k \leq -\delta_l}} \alpha.\delta_k(\tilde{\boldsymbol{w}}^s) + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < -\delta_l < \alpha.\delta_k}} -\delta_l(\tilde{\boldsymbol{w}}^s) \right)$$

Deriving with respect to $\tilde{\boldsymbol{w}}^s$ yields:

$$\nabla^s_{iter} = \left( 0 + 0 + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < \alpha.\delta_k \leq -\delta_l}} \alpha.\tilde{\boldsymbol{a}}^k + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < -\delta_l < \alpha.\delta_k}} -\tilde{\boldsymbol{a}}^l \right)$$

This gradient can then be rewritten in its final form as:

$$\nabla^s_{iter} = \left( \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < \alpha.\delta_k \leq -\delta_l}} \alpha.\tilde{\boldsymbol{a}}^k - \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < -\delta_l < \alpha.\delta_k}} \tilde{\boldsymbol{a}}^l \right)$$

# B Derivation of the global goal function

The global goal function has been defined in expression (4). To simplify this expression, we introduce the following new notations:

$$\tilde{\boldsymbol{w}}^S = \begin{pmatrix} \tilde{\boldsymbol{w}}^1 \\ \vdots \\ \tilde{\boldsymbol{w}}^{|S|} \end{pmatrix} \in I\!R^{(d+1).|S|}$$

$$\text{where} \quad \tilde{\boldsymbol{w}}^s = \begin{pmatrix} \boldsymbol{w}^s \\ w_0^s \end{pmatrix}, \; \tilde{\boldsymbol{a}}^k = \begin{pmatrix} \boldsymbol{a}^k \\ -1 \end{pmatrix}, \; \tilde{\boldsymbol{a}}^l = \begin{pmatrix} \boldsymbol{a}^l \\ -1 \end{pmatrix} \in I\!R^{d+1}$$

$$\text{and} \quad \boldsymbol{w}^s = \begin{pmatrix} w_1^s \\ \vdots \\ w_d^s \end{pmatrix}, \; \boldsymbol{a}^k = \begin{pmatrix} a_1^k \\ \vdots \\ a_d^k \end{pmatrix}, \; \boldsymbol{a}^l = \begin{pmatrix} a_1^l \\ \vdots \\ a_d^l \end{pmatrix} \in I\!R^d, \; \text{with } s \in \{1, \ldots, |S|\}$$

In these expressions $d$ is the number of modalities, $|S|$ is the number of half-spaces in the set, $k$ is the number of positive and $l$ the number of negative points, $\tilde{\boldsymbol{w}}^S$ is the vector containing all $|S|$ half-spaces.

What is particular for this global approach is that *all* half-spaces are used at the same time (this was not the case for the iterative approach, where only the last added half-space was used). With these new notations, the global goal function becomes:

$$\text{goal}_{glob} = \min_{k,l} \left( \sum_s \Delta_{kls}(\tilde{\boldsymbol{w}}^s) \right)$$

$$\text{where} \quad \Delta_{kls}(\tilde{\boldsymbol{w}}^s) = \max\{0, \min\{\alpha.\delta_k(\tilde{\boldsymbol{w}}^s), -\delta_l(\tilde{\boldsymbol{w}}^s)\}\},$$

$$\text{with} \quad \delta_x(\tilde{\boldsymbol{w}}^s) = \tilde{\boldsymbol{w}}^s.\tilde{\boldsymbol{a}}^x, x \in \{k, l\}.$$

This time however there is no need to split $\Delta_{kls}$ as we have done in the iterative approach, since in this global goal function, the separation for the pair $(k, l)$ is calculated directly for all $|S|$ half-spaces of the set, without having to deal with the "clipping" effect of the iterative goal function. Taking this into account and rewriting the global goal function replacing $\Delta_{kls}$ by its value, gives:

$$\text{goal}_{glob} = \min_{k,l} \left( \sum_s \max\{0, \min\{\alpha.\delta_k(\tilde{\boldsymbol{w}}^s), -\delta_l(\tilde{\boldsymbol{w}}^s)\}\} \right)$$

Eliminating the max operator gives us then the following expression:

$$\text{goal}_{glob} = \min_{k,l} \left( \sum_{\substack{s \\ s.t.\ 0<\min\{\alpha.\delta_k, -\delta_l\}}} \min\{\alpha.\delta_k(\tilde{\boldsymbol{w}}^s), -\delta_l(\tilde{\boldsymbol{w}}^s)\} \right)$$

Eliminating the min operator gives:

$$\text{goal}_{glob} = \min_{k,l} \left( \sum_{\substack{s \\ s.t.\ 0<\alpha.\delta_k \leq -\delta_l}} \alpha.\delta_k(\tilde{\boldsymbol{w}}^s) + \sum_{\substack{s \\ s.t.\ 0<-\delta_l<\alpha.\delta_k}} -\delta_l(\tilde{\boldsymbol{w}}^s) \right)$$

Deriving with respect to $\tilde{\boldsymbol{w}}^s$ yields then the gradient for the pair $(k, l)$:

$$\nabla^S_{glob} = \min_{k,l} \left( \sum_{\substack{s \\ s.t.\ 0 < \alpha.\delta_k \leq -\delta_l}} \alpha.\tilde{\boldsymbol{a}}^k - \sum_{\substack{s \\ s.t.\ 0 < -\delta_l < \alpha.\delta_k}} \tilde{\boldsymbol{a}}^l \right)$$

# C Proof of equivalence between two alternative problem formulations

To simplify the development of this proof, we introduce again the following notations:

$$\tilde{\boldsymbol{x}}^S = \begin{pmatrix} \tilde{\boldsymbol{x}}^1 \\ \vdots \\ \tilde{\boldsymbol{x}}^{|S|} \end{pmatrix}, \ \tilde{\boldsymbol{w}}^S = \begin{pmatrix} \tilde{\boldsymbol{w}}^1 \\ \vdots \\ \tilde{\boldsymbol{w}}^{|S|} \end{pmatrix} \in I\!R^{(d+1).|S|}$$

$$\text{where } \ \tilde{\boldsymbol{x}}^s = \begin{pmatrix} \boldsymbol{x}^s \\ x_0^s \end{pmatrix}, \tilde{\boldsymbol{w}}^s = \begin{pmatrix} \boldsymbol{w}^s \\ w_0^s \end{pmatrix}, \ \tilde{\boldsymbol{a}}^k = \begin{pmatrix} \boldsymbol{a}^k \\ -1 \end{pmatrix}, \ \tilde{\boldsymbol{a}}^l = \begin{pmatrix} \boldsymbol{a}^l \\ -1 \end{pmatrix} \in I\!R^{d+1}$$

$$\text{and } \ \boldsymbol{w}^s = \begin{pmatrix} w_1^s \\ \vdots \\ w_d^s \end{pmatrix}, \ \boldsymbol{a}^k = \begin{pmatrix} a_1^k \\ \vdots \\ a_d^k \end{pmatrix}, \ \boldsymbol{a}^l = \begin{pmatrix} a_1^l \\ \vdots \\ a_d^l \end{pmatrix} \in I\!R^d, \ \text{with } s \in \{1, \ldots, |S|\}$$

In this expression $d$ is the number of modalities, $|S|$ is the number of half-spaces in the set, $k$ is the number of positive and $l$ the number of negative points.

Using these conventions, the original problem can be stated as follows:

$$\text{Using all } |N| \text{ global gradient vectors: } \ \nabla^S_{glob_1}, \ldots, \nabla^S_{glob_{|N|}} \in I\!R^{(d+1).|S|},$$

find an $\boldsymbol{x}^S \in I\!R^{(d+1).|S|}$ such that $\forall n \in N : \boldsymbol{w}^S + \eta.\boldsymbol{x}^S$ maximizes expression (4) for all minimal pairs.

In this expression $|N|$ is the number of pairs with minimal separation and $\eta$ is any positive number.

$\forall n \in N$, we can rewrite the global goal function as follows:

For $k, l$ fixed, $k \in K, l \in L$:

$$\text{goal}_{glob}(\tilde{\boldsymbol{w}}^S) = \sum_{s=1}^{S} \Delta_{kls}$$

where $\Delta_{kls}$ is the same as in expression (2).

Rewriting this gives:

$$\text{goal}_{glob}(\tilde{\boldsymbol{w}}^S) = \sum_{s=1}^{S} \max\{0, \min\{\boldsymbol{w}^s.\boldsymbol{a}^k, -\boldsymbol{w}^s.\boldsymbol{a}^l\}\}$$

Returning to our problem, we can introduce $\boldsymbol{x}^S$ in the previous expression:

$$\text{goal}_{glob}(\tilde{\boldsymbol{w}}^S + \eta.\boldsymbol{x}^S) = \sum_{s=1}^{S} \max\{0, \min\{(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^k, -(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^l\}\}$$

We can eliminate the max by restricting the summation to only those half-spaces that separate the considered minimal pair. This gives us the following:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S) = \sum_{s\ separates\ k,l} \min\{(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^k, -(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^l\}$$

Since we want to maximize the additional separation on the minimal pair introduced by going in the direction $\boldsymbol{x}^S$), maximizing the previous expression is equivalent with the following expression:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S)\text{ - goal}_{glob}(\boldsymbol{w}^S) = \sum_{s\ separates\ k,l} \left(\min\{(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^k, -(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^l\} - \min\{\boldsymbol{w}^s.\boldsymbol{a}^k, -\boldsymbol{w}^s.\boldsymbol{a}^l\}\right)$$

This expression can be rewritten as:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S)\text{ - goal}_{glob}(\boldsymbol{w}^S) = \sum_{s\ separates\ k,l} \left(\min\{\boldsymbol{w}^s.\boldsymbol{a}^k + \eta.\boldsymbol{x}^s.\boldsymbol{a}^k, -\boldsymbol{w}^s.\boldsymbol{a}^l - \eta.\boldsymbol{x}^s.\boldsymbol{a}^l\} - \min\{\boldsymbol{w}^s.\boldsymbol{a}^k, -\boldsymbol{w}^s.\boldsymbol{a}^l\}\right)$$

Since $\eta$ is any positive number and since all other quantities involved are positive, the minimum of both expressions between brackets will not shift. This reduces the previous expression to:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S)\text{ - goal}_{glob}(\boldsymbol{w}^S) = \sum_{s\ separates\ k,l} \min\{\eta.\boldsymbol{x}^s.\boldsymbol{a}^k, -\eta.\boldsymbol{x}^s.\boldsymbol{a}^l\}$$

Since $\eta$ doesn't depend on $s$, we can place $\eta$ before the summation:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S)\text{ - goal}_{glob}(\boldsymbol{w}^S) = \eta. \sum_{s\ separates\ k,l} \min\{\boldsymbol{x}^s.\boldsymbol{a}^k, -\eta.\boldsymbol{x}^s.\boldsymbol{a}^l\}$$

As we want to maximize this expression and since $\eta$ is positive, this is equivalent with the following:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S)\text{ - goal}_{glob}(\boldsymbol{w}^S) = \sum_{s\ separates\ k,l} \min\{\boldsymbol{x}^s.\boldsymbol{a}^k, -\eta.\boldsymbol{x}^s.\boldsymbol{a}^l\}$$

Posing $\boldsymbol{x}^s.\boldsymbol{a}^k = \delta_k$ and $-\boldsymbol{x}^s.\boldsymbol{a}^l = -\delta_l$, and knowing that both these quantities are strictly positive (because we only consider the half-spaces that actually do separate $k, l$), we can rewrite the goal we are after as maximizing the RHS (Right Hand Side) of the previous expression. This gives us the following new formulation of the problem to be solved:

$$\text{Maximize} \sum_{\substack{s \\ s.t.\ 0<\delta_k, -\delta_l}} \min\{\boldsymbol{x}^s.\boldsymbol{a}^k, -\boldsymbol{x}^s.\boldsymbol{a}^l\}$$

Splitting the min function gives us the following:

$$\text{Maximize} \left( \sum_{\substack{s \\ s.t.\ 0<\delta_k\leq-\delta_l}} \boldsymbol{x}^s.\boldsymbol{a}^k - \sum_{\substack{s \\ s.t.\ 0<-\delta_l<\delta_k}} \boldsymbol{x}^s.\boldsymbol{a}^l \right)$$

Rewriting this expression using the complete vector of half-spaces $\boldsymbol{x}^S$, this expression becomes:

$$\text{Maximize}\ \ \boldsymbol{x}^S. \left( \sum_{\substack{s \\ s.t.\ 0<\delta_k\leq-\delta_l}} \boldsymbol{a}^{k^s} - \sum_{\substack{s \\ s.t.\ 0<-\delta_l<\delta_k}} \boldsymbol{a}^{l^s} \right)$$

The expression between brackets is nothing else than the gradient $\nabla^S_{glob_n}$ of the global goal function for the considered minimal pair. This is shown in Appendix B. Using that knowledge, the previous line gives:

$$\text{Maximize}\ \ \boldsymbol{x}^S.\nabla^S_{glob_n}$$

So the original problem can be restated as follows:

$$\text{Find } \boldsymbol{x}^S \in \mathbb{R}^{(d+1).|S|}$$

$$\text{Such that } \forall n \in N,\ \boldsymbol{x}^S.\nabla^S_{glob_n} > 0 \text{ is maximal.}$$