Chapitre 3

Méthodes de réduction et classification.

3.1 Introduction

Tout système de reconnaissance biométrique comporte une phase très importante basée sur la réduction d'espace. Pour cela nous consacrons un chapitre pour présenter les différentes techniques de projection d'espace. La classification n'est pas à négliger. Nous essayons de présenter l'essentiel des approches utilisées.

Dans les différents domaines de recherches scientifiques, le développement technologique et le besoin de superviser des systèmes de plus en plus complexes nécessitent l'analyse de bases de données de taille importante (signaux, images, documents, ...). Toutefois, si dans cette accumulation de données, on est sûr d'avoir une information complète et utile, celle-ci risque d'être "noyée" dans la masse. Ceci pose les problèmes de la structuration des données et de l'extraction des connaissances. En effet, les bases de données sont en général définies par des tableaux à deux dimensions correspondant aux données et aux attributs les caractérisent. Ces deux dimensions peuvent prendre des valeurs très élevées, ce qui peut poser un problème lors du stockage, de l'exploration et de l'analyse. Pour cela, il est important de mettre en place des outils de traitement permettant l'extraction des connaissances sous-jacentes. L'extraction des connaissances s'effectue selon deux directions, la catégorisation des données (par regroupement en classes) et/ou la réduction de la dimension de l'espace de représentation de ces données (par sélection ou extraction d'attributs). La classification vise à découvrir la structure intrinsèque d'un ensemble de données en formant des groupements qui partagent des caractéristiques similaires.

La réduction de la dimension se pose comme une étape primordiale dans le processus de prétraitement des données (filtrage, nettoyage, élimination des points aberrants, etc.). En effet, pour des données appartenant à un espace de grande dimension, certains attributs n'apportent aucune information voire expriment du bruit, d'autres sont redondants ou corrélés. Ceci rend les algorithmes de décision complexes, inefficaces, moins généralisables et d'interprétation délicate. Les méthodes de réduction de la dimension de l'espace de représentation peuvent être divisées en méthodes d'extraction d'attributs et méthodes de sélection d'attributs. L'extraction d'attributs transforme l'espace d'attributs de départ en un nouvel espace formé de la combinaison linéaire ou non linéaire des attributs initiaux. La sélection d'attributs

choisit les attributs les plus pertinents selon un critère donné. Les données sont alors analysées après projection dans un espace de représentation composé des attributs les plus pertinents. Toutefois, l'interprétation des attributs extraits est plus délicate que l'interprétation des attributs sélectionnés. Le point clé de la sélection d'attributs est la définition d'un score mesurant la pertinence de chacun des attributs. Cette sélection s'appuie sur la connaissance explicite et implicite sur les données. Quand on ne dispose d'aucune information à priori sur le regroupement des données en classes, le contexte d'apprentissage est dit non supervisé. La pertinence d'un attribut est alors mesurée en évaluant ses capacités à préserver la structure des données. Pour de nombreuses applications, on dispose des informations à priori sur la répartition des données en classes. Ainsi, pour ces données, les labels des classes ont été fournis. Dans ce cas, la sélection supervisée consiste à mesurer la corrélation entre l'attribut et les labels des classes des données. Nous n'aborderons que ce dernier cas de classification car tous système biométrique est fondé sur une phase d'apprentissage où la création de signature des données est effectuée d'où la création d'une base de données indexée.

Notre **objectif** c'est l'**identification** des personnes dans un premier temps par la modalité visage en se basant sur une analyse globale en **2D**, puis par la couleur et profondeur en 3D et finalement une **fusion** des différentes régions d'intérêts en présence **d'expressions faciales**. Donc la fusion est prise en considération et nous lui consacrons le **chapitre 4** pour son étude.

3.2 Méthodes de réduction de dimension

L'**A**nalyse en **C**omposantes **P**rincipales (**PCA**) est l'une des méthodes les plus utilisées dans la reconnaissance de visages, elle a été proposée par **M.A.Turk et M.P. Pentland** [4]. Dans l'identification de visage basée sur la **PCA**, les images de visage **2D** sont transformées en vecteurs colonnes **1D**. Le calcul de la matrice de covariance à base de ces vecteurs est difficile à cause de la grande taille des vecteurs **1D** et le nombre important d'échantillons d'apprentissage. En général, le calcul des vecteurs propres d'une grande matrice de covariance prend beaucoup de temps. L'analyse discriminante linéaire (**LDA**) est née des travaux de **Belhumeur et al. en 1997.** La **LDA** effectue une véritable séparation de classe et cela en **minimisant** les **variations** entre les images d'un **même individu** tout en **maximisant** les **variations** entre les **images d'individus différents**. Néanmoins, lorsque le nombre d'individus est inférieur à la résolution de l'image, il est difficile d'appliquer la **LDA** qui peut faire apparaître des matrices de dispersions singulières. Afin de contourner ce problème, certains algorithmes basés sur la **LDA** ont été proposées, le plus connu est la **RLDA** (**R**egularized **LDA**).

Roli en 2002 [205], ont remarqué que la **LDA** et la **PCA** ne sont pas corrélées car la LDA génère un espace propre significativement différent de la **PCA**. Les expériences effectuées dans [205] montrent que la ***fusion de la LDA et de la PCA ont donné de bons résultats.***

Dans [206]**,** il a été prouvé expérimentalement que la **PCA** et la **LDA** peuvent être appliquées sur un nombre réduits de coefficients **DCT** pour réaliser une meilleure reconnaissance avec un gain en temps de calcul et en espace mémoire. Afin d'améliorer le taux de reconnaissance de la **PCA** et **LDA**, leur fusion est proposée par **G. L. Marcialis** et **al.** La réduction de dimensionnalité employant la **PCA** ou la **LDA** nécessite un temps prohibitif lorsque la dimension et le nombre d'échantillons d'apprentissage sont importants. Pour cette raison, la réduction de la complexité informatique est fortement exigée. Pour cela, la transformée en cosinus discrète (**DCT**) a été utilisée dans l'identification de visages pour la réduction de dimension.

Dans ce chapitre nous nous attachons à décrire plusieurs méthodes de réduction de dimension. La réduction de dimension consiste à transformer des données représentées dans un espace de grande dimension en une représentation dans un espace de dimension plus faible. Idéalement, la nouvelle représentation a une dimension égale au nombre de paramètres nécessaires pour décrire les données observées [207]. La réduction de dimension est importante dans de nombreux domaines étant donné qu'elle facilite la classification, la visualisation ou encore la compression de données de grande dimension. Elle permet également souvent de limiter l'effet de la malédiction de la dimension et d'autres propriétés non désirées des espaces de grande dimension [208].

Récemment, un grand nombre de méthodes de réduction de dimension ont été proposées [208, 209,210,211,212,100,213,214,215]. Ces techniques sont capables de traiter des problèmes complexes non linéaires et ont souvent été proposées comme une alternative aux techniques linéaires classiques telles l'analyse en composantes principales (**ACP**) ou l'analyse discriminante linéaire (**LDA**).

De précédentes études ont en effet montré que les *approches non linéaires surpassent les méthodes linéaires* sur des jeux de *données artificiels* hautement non linéaires. Cependant, les succès de réduction de dimension avec les *méthodes non linéaires sur des jeux de données naturelles sont plutôt rares.* Dans la suite, nous décrivons celles qui sont les plus proches de notre modèle des techniques linéaires classiques telles :

l'**A**nalyse en **C**omposantes **P**rincipales (**ACP**) [216], la **F**actorisation **N**on négative de **M**atrices (**NMF**) [48], l'Analyse en **C**omposantes **I**ndépendantes (**ICA**) [217] et l'**A**nalyse

Discriminante **L**inéaire (**LDA**) [217], ainsi que dix méthodes non linéaires (le nom de chaque méthode n'a volontairement pas été traduit) :

Multi **D**imensional **S**caling (**MDS**) [218,219], Isomap[200,214], **K**ernel **PCA** (**KPCA**) [221 ,100], **D**iffusion **M**aps [211,222], **M**ulti **L**ayer **A**uto **E**ncoders[223,210], **L**ocally **L**inear **E**mbedding (**LLE**) [212], **L**aplacian **E**igenmaps [208], **H**essian **LLE** [209], **L**ocal **T**angent **S**pace **A**nalysis (**LTSA**) [215] et **L**ocally **L**inear **C**oordination (**LLC**) [213].

D'autres techniques non linéaires ont été proposées, telles que :

Principal **C**urves [224], **G**eneralized **D**iscriminant **A**nalysis [225], **K**ernel **M**aps [226], **M**aximum **V**ariance **U**nfolding [227], **C**onformal **E**igen**M**aps [257], **L**ocality **P**reserving **P**rojections [229], **L**inear **L**ocal **T**angent **S**pace **A**lignment [230], **S**tochastic **P**roximity **E**mbedding [231], **F**ast**M**ap [232], **G**eodesic **N**ull**S**pace **A**nalysis [234].

La plupart d'entre elles sont des variantes des dix méthodes énoncées plus haut, et ne seront donc pas décrites ici. Nous nous intéressons à celles qui ont fait leurs preuves dans le domaine de la reconnaissance de visages et d'après l'état de l'art établit dans le **chapitre 2**. Il en sort que la **PCA, DPCA, ICA, KPCA, EFM** sont les plus sollicitées. Nous essayons de les présenter dans ce qui suit.

3.2.1 La réduction de dimension

Supposons qu'un jeu de données soit décrit par la matrice X de taille (n x D) où n est le nombre de vecteurs x_i de dimension D. Ce jeu de données possède une dimension propre (ou intrinsèque) d, où d < D voire d << D. En termes mathématiques, la dimension intrinsèque signifie que le jeu de données repose sur une variété de dimension d, contenu dans un espace de plus grande dimension D. Une technique de réduction de dimension transforme le jeu de données X en un nouvel ensemble Y de dimension d, en gardant au maximum l'essentiel de l'information de l'ensemble de départ. Généralement, ni la géométrie de la variété, ni la dimension d sont connus. Les techniques de réduction de dimension peuvent être classées en plusieurs groupes (voir la figure **3.1**). Le principal critère de classement est l'aspect linéaire ou non des méthodes. Les méthodes linéaires supposent que les données reposent sur une variété linéaire de l'espace de grande dimension. Les méthodes non linéaires ne reposent pas sur cette hypothèse et sont capables de caractériser des variétés plus complexes. Cette hypothèse et sont capables de caractériser des variétés plus complexes.

Figure 3.1 Taxonomie des techniques de réduction de dimension.

3.2.2 Méthodes linéaires de réduction de dimension

Nous décrivons ici quatre des méthodes linéaires les plus couramment utilisées : l'**A**nalyse en **C**omposantes **P**rincipales (**ACP**), la **F**actorisaion de **M**atrices **N**on négatives (**NMF**), l'**A**nalyse en **C**omposantes **I**ndépendantes (**ICA**) et l'**A**nalyse **D**iscriminante **L**inéaire (**LDA** ou **FLD** : **F**isher **L**inear **D**escriminant) et l'**EFM** (**E**nhanced **F**isher **L**inear **D**iscriminant **M**odel ou **M**odèle de **F**isher **A**mélioré).

3.2.2.1 L'Analyse en Composantes Principales

L'Analyse en Composantes Principales (**ACP**) [235], aussi connue sous le nom de transformée de **Karhunen–Loève** [236] est une méthode très utilisée en statistique. Introduite par **Pearson** [237] puis plus tard par **Hotelling** [216], sa principale idée est de **réduire** la **dimension** d'un jeu de données tout en gardant un **maximum d'informations**. Cela est réalisé grâce à une **projection** qui **maximise la variance** tout en **minimisant l'erreur quadratique moyenne** de la reconstruction pour plus de détails, voir [238,235,239,240].

Pour la dérivation **Hotelling** définit l'**ACP** comme une projection orthogonale maximisant la variance dans l'espace projeté. Étant donné n échantillons $x_i \in R^D$ et $u \in R^D$ tel que :

$$\|u\| = u^T u = 1 \qquad (3.1)$$

soit un vecteur ortho normal de projection. Un échantillon x_i est projeté sur u par : $a_i = u^T x_i$

La variance de l'échantillon peut d onc être estimée :

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x}) \qquad (3.2)$$

où xmoy est la moyenne des projetés des échantillons de la base :

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \quad \text{d'où} \quad \overline{a} = u^T \overline{x} \qquad (3.3)$$

Ainsi la variance du projeté est donnée par :

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (a_i - \overline{a}) \qquad (3.4)$$

$$= \frac{1}{n-1} \sum_{i=1}^{n} (u^T x_i - u^T \bar{x}) \quad (3.5)$$

$$= u^T C u \quad (3.6)$$

Où $C \in R^{DxD} = \frac{1}{n-1} \sum_{i=1}^{n} ((x_i - \bar{x})(x_i - \bar{x})^T)$ (3.7)

est la matrice de covariance de $X = [x_1; \ldots\ldots ; x_n] \in R^{Dxn}$. Le problème de maximisation de la variance dans l'espace projeté peut donc s'écrire : $max\ u^T C\ u$ avec $u^T u = 1$

Le calcul de la solution optimale peut être réalisé grâce au multiplicateur de Lagrange :

$$f(u,\lambda) = u^T CU + \lambda(1-u^T u) \quad (3.8)$$

Par dérivation partielle selon u : $\frac{\partial f(u,\lambda)}{\partial u} = 2Cu - 2\lambda u = 0$ (3.9)

on obtient : $Cu = \lambda u$ (3.10)

Ainsi, le maximum pour le multiplicateur de Lagrange est obtenu si λ est une valeur propre et u un vecteur propre de C. Ainsi la variance décrite par le vecteur de projection u est donnée par λ. D'autres méthodes de **dérivation** de l'**ACP** sont données dans **[**241,235]. Pour une vue probabiliste de la dérivation de l'**ACP**, voir [84,85].

Calcul de l'ACP pour la mise en œuvre de méthodes : il est supposé que le jeu de données d'entraînement est disponible en entier. Ainsi nous avons un ensemble de n observations $x_i \in R^D$ organisés sous forme matricielle $X = [x_1;\ldots\ldots;x_n] \in R^{Dxn}$**.** L'estimation de la base de projection de l'**ACP** revient donc à estimer les éléments propres de la matrice de covariance C de X. Le calcul requiert d'abord l'échantillon moyen : $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ (3.11)

Puis les échantillons sont normalisés par rapport à la moyenne x_{moy} : $\hat{x} = \bar{x}_i = x_i - \bar{x}$

pour former la nouvelle matrice $\hat{x} = [\hat{x}_1;\ldots\ldots;\hat{x}_n]$. La matrice de covariance $C \in R^{DxD}$ est ensuite calculée par : $C = \frac{1}{n-1} \hat{x}\ \hat{x}^T$ (3.12)

La recherche des éléments propres de C conduit à l'obtention de la base de vecteurs propres $u_i \in R^D$, pour lesquels, à chacun d'eux, est associée une valeur propre λ_i. Généralement triés par ordre décroissant de valeur propre associée, les premiers vecteurs propres forment alors une base dans laquelle la plupart de l'information du jeu de données d'entraînement est gardée.

ACP pour des données de grande dimension.: La dimension de la matrice de covariance dépend de la dimension D des vecteurs du jeu de données, qui peut être relativement grande pour certains types de données (typiquement des images). La méthode décrite plus haut devient alors difficile à appliquer, essentiellement à cause de la recherche des éléments

propres de la matrice de covariance *C*. En effet, pour des images de taille 100x100 par exemple, la matrice de covariance *C* à inverser est de taille 10000x10000. Cependant, il est connu que pour toute matrice *X*, les produits matriciels XX^T et X^TX partagent les mêmes valeurs propres différentes de zéro.

Ainsi, le calcul des éléments propres de $C = XX^T$ $C \epsilon R^{DxD}$ peut se ramener au calcul des éléments propres de la matrice $M \epsilon R^{nxn}$ où $M = X^TX$. Soit e_i les vecteurs propres de *M* associés aux valeurs propres δ_i. On a donc : $X^TXe_i = \delta_i e_i$ (**3.13**)

En multipliant à gauche par *X* les deux côtés de l'équation, on obtient ainsi :

$$X(X^TXe_i) = X(\delta_i e_i) \quad (\mathbf{3.14})$$

$$X X^T(Xe_i) = \delta_i(X e_i) \quad (\mathbf{3.15})$$

On voit donc que Xe_i est vecteur propre de XX^T et que δ_i est la valeur propre associée, d'où

$$\begin{cases} ui = Xei \\ \lambda i = \delta i \end{cases} \quad (\mathbf{3.16})$$

La matrice *M* étant beaucoup plus petite que la matrice *C* (typiquement, on passe d'une complexité de l'ordre de la dimension des échantillons à une complexité de l'ordre du nombre d'échantillons d'apprentissage), les calculs sont donc plus efficaces. L'algorithme de l'**A**nalyse en **C**omposantes **P**rincipales est résumé à l'Algorithme ci-dessous.

Des variantes de l'**ACP** ont été proposées. Ainsi plusieurs méthodes ont été proposées pour extraire des axes principaux robustes notamment au bruit contenu dans les images d'apprentissage [242,243], ou des méthodes basées sur une formulation Espérance–Maximisation de l'**ACP** [84,244,85]. Dans le cas où les données d'apprentissage ne sont pas toutes disponibles au départ (cas de vidéos par exemple), des versions incrémentales de l'**ACP** ont été mises au point **[**245,246,247,248]. Des méthodes combinant l'aspect incrémental et robuste ont également été proposées dans [249,250].

Algorithme : Calcul de l'ACP

Entrées : matrice X

Sorties : vecteur moyen $\overline{x}$, base de vecteurs propres U, valeurs propres associées λ_i.

Calcul du vecteur moyen :

$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$

Normalisation des images d'entrées :

$\hat{x} = x_i - \overline{x}$

$\hat{x} = [\hat{x}_1; \ldots\ldots; \hat{x}_n]$.

si Données de grande dimension **alors**

$M = \frac{1}{n-1} \hat{x}\ \hat{x}^T$

Calcul des éléments propres de M :

$E = [e_1 ; \ldots\ldots ; e_n]$

$\delta = [\delta_1; \ldots\ldots ; \delta n]$

Calcul des éléments finaux :

$u_i = Xe_i \qquad U = [u_1 ; \ldots\ldots ; u_n]$

$\lambda_i = \delta_1 \qquad \lambda = [\lambda_1 ; \ldots\ldots ; \lambda_n]$

sinon

$C = \frac{1}{n-1} \hat{x}\ \hat{x}^T$

Calcul des éléments propres de C :

$U = [u_1 ; \ldots\ldots ; u_n]$

$\lambda = [\lambda_1 ; \ldots\ldots ; \lambda_n]$

retourner $\overline{x}$,U, λ

3.2.2.2 Factorisation de Matrice Non Négative

La factorisation de matrice non–négative(ou **NMF** pour **N**on **N**egative **M**atrix **F**actorization) a été proposée dans [251] et [252]. Introduite dans le cadre de la vision par ordinateur dans [253], cette technique, contrairement à l'**ACP**, ***n'autorise pas de valeurs négatives*** dans les *vecteurs de base* ni dans les *vecteurs de projection*. Les vecteurs de base sont donc additifs et représentent des structures locales. Plus formellement, la méthode peut être décrite ainsi : Étant donnée une matrice $V \in R^{nxm}$ positive contenant les images vectorisées, le but est de trouver les matrices non–négatives $W \in R^{nxr}$ et $H \in R^{rxm}$ qui approximent la matrice V :

$$V \approx WH \qquad (3.17)$$

Les deux matrices W et H doivent être estimées itérativement en considérant le problème d'optimisation suivant :

$$min \left\| V - WH \right\|_2^2 \quad \text{avec } W;H > 0 \qquad (3.18)$$

Les règles de mise à jour pour les matrices W et H sont alors : $H_{i;j} \leftarrow H_{i;j} \frac{(W^T V)_{i,j}}{(W^T WH)_{i,j}}$ (3.19)

$$W_{i;j} \leftarrow W_{i;j} \frac{(VH^T)_{i,j}}{(WHH^T)_{i,j}} \qquad (3.20)$$

Plus de détails sur la dérivation de la méthode ainsi que sur des descriptions de l'algorithme peuvent être trouvées dans [254] et [255]. De plus, pour améliorer la rapidité de l'algorithme ainsi que pour s'assurer que la solution trouvée soit le minimum global (le problème d'optimisation n'est en effet pas convexe en W ni en H), plusieurs extensions ont été proposées [256,257]. Elles considèrent une contrainte additionnelle de parcimonie et reformulent le problème en un problème convexe.

3.2.2.3 Analyse en Composantes Indépendantes

L'**A**nalyse en **C**omposantes **I**ndépendantes (ou **ICA** pour **I**ndependant **C**omponent **A**nalysis) a été introduite par **Hérault**, **Jutten** et **Ans** dans [258,259] et [260] dans le contexte de la neurophysiologie. Elle devint populaire lors de son utilisation dans le domaine du traitement du signal pour la séparation de sources aveugles dans [261] et [262]. Le but est d'exprimer un ensemble de n variables aléatoires $x_1;\ldots\ldots;\ x_n$ comme une combinaison linéaire de n variables aléatoires statistiquement indépendantes s_j :

$$x_j = a_{j;1}s_1 + \ldots\ldots + a_{j;n}s_n \quad \forall\, j \qquad (3.21)$$

ou sous forme matricielle : $x = As$ (3.22)

où $x = [x_1;\ \ldots\ldots\ ;\ x_n]^T$, $s = [s_1;\ \ldots..\ ;\ s_n]^T$ et A est une matrice contenant les coefficients a_{ij}. Le but de l'Analyse en Composantes Indépendantes est l'estimation des composantes originales s_i, ou de manière équivalente des coefficients a_{ij}. Par définition, les variables aléatoires s_i sont mutuellement indépendantes et la matrice de mélange est donc inversible. Ainsi le problème de l'**ICA** peut être formulé [263] : $u = Wx = WAs$ (3.23)

Plusieurs fonctions objectives ont été proposées, ainsi que des méthodes efficaces de résolution : **I**nfo**M**ax [264] ou **F**ast**ICA** [265]. Pour plus de détails sur la théorie et les applications possibles de l'Analyse en Composantes Indépendantes, voir [266]. Pour l'application de l'**ICA** à la **reconnaissance de visages** [89] et [267] proposent **deux architectures**. Dans la **première**, les images sont considérées comme un **mélange linéaire** d'**images** de base statistiquement indépendantes. Dans la **seconde,** le but est de trouver **des coefficients statistiquement indépendants représentant l'image d'entrée**.

Pour ces deux architectures, une Analyse en Composantes Principales est **appliquée en prétraitement.**

3.2.2.4 Analyse Discriminante Linéaire

Si les données d'apprentissage sont labélisées, ces informations peuvent être utilisées pour l'apprentissage du sous–espace. Ainsi, pour assurer une classification plus efficace,

l'**A**nalyse **D**iscriminante **Linéaire** de **Fisher (LDA** pour **L**inear **D**iscriminant **A**nalysis) a pour but de maximiser la distance entre les classes tout en minimisant la variance intra-classe. Plus formellement, soient $\{x_1; \ldots\ ;\ x_n\}$ n échantillons appartenant à une classe parmi c $\{X_1; \ldots\ ; X_c\}$. L'Analyse Discriminante Linéaire calcule une fonction de classification $g(x) = W^T x$, où la matrice W est choisie comme la projection linéaire minimisant la variance intra–classe : $S_B = \sum_{j=1}^{c} n_j (\overline{x_j} - \overline{x})(\overline{x_j} - \overline{x})^T$ (**3.24**)

tandis que la variance inter–classe est maximisée :

$$S_W = \sum_{j=1}^{c} \sum_{x_k \epsilon X_j} (x_k - \overline{x}_j)(x_k - \overline{x}_j)^T \quad (\mathbf{3.25})$$

où $\overline{x}$ est le vecteur moyen de tous les échantillons, $\overline{x_j}$ est le vecteur moyen des échantillons appartenant à la classe j, et n_j est le nombre d'échantillons de la classe j. Le calcul de la projection est ainsi obtenu en **maximisant** le **critère de Fisher** :

$$W_{opt} = \arg\max \frac{|(W^T S_B W)|}{|(W^T S_w W)|} \quad (\mathbf{3.26})$$

La solution optimale à ce problème d'optimisation est donnée par la résolution du problème généralisé des valeurs propres : $S_B w = \lambda S_W w$ (**3.27**)

ou en calculant directement les vecteurs propres de $S_w^{-1} S_B$. Le rang de $S_w^{-1} S_B$ est au plus c - 1. Ainsi, pour de nombreuses applications, cette matrice est singulière et le problème des valeurs propres ne peut être résolu. Ce problème est souvent appelé le problème des échantillons de petite taille (small sample size problem). Pour surmonter ce problème, plusieurs solutions ont été proposées [83,268,269]. De plus, de nombreuses variantes de la **LDA** ont été introduites telles la classification robuste [270], ou la **LDA** incrémentale [271]. La **LDA** est étroitement liée à l'**ACP**, du fait que toutes les deux recherchent des combinaisons linéaires des variables qui représentent au mieux les données. La **LDA** essaye explicitement de modeler la différence entre les classes des données. L'**ACP** quand à elle, ne tient pas compte des différences entre les classes.

Chaque visage, qui se compose d'un grand nombre de pixels, est réduit à un plus petit ensemble de combinaisons linéaires avant la classification. Chacune des nouvelles dimensions est une combinaison linéaire des valeurs de pixels. Les combinaisons linéaires obtenues en utilisant **FLD** s'appellent les Fisherfaces, en analogie avec les visages propres (EigenFaces) [272,55]. La **LDA** est une technique qui cherche les directions qui sont efficaces pour la discrimination entre les données. La **figure 3.2** représente un exemple de classification de deux nuages de points. L'axe principal de la **LDA** est l'axe de projection qui maximise la

séparation entre les deux classes. Il est clair que cette projection est optimale pour la séparation des deux classes par rapport à la projection sur l'axe principal calculé par **ACP**.

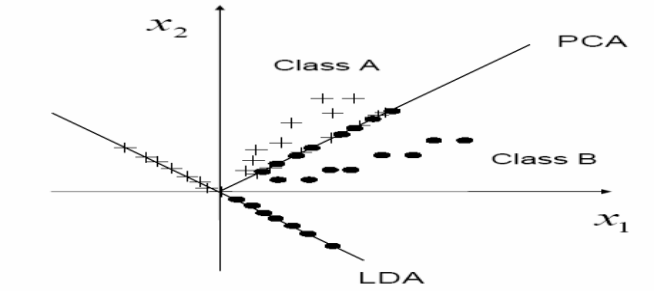

Figure 3.2 Comparaison entre les projections de deux classes de points ("class 1" et "class 2") sur les axes principaux construits par **ACP** et par **LDA.**

Les étapes de l'analyse discriminante linéaire **LDA** sont :

1) Calcul des moyennes

Nous calculons la moyenne des images dans chaque classe (m_i) et la moyenne de toute les images m

$$m_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_j^i \;\; ; \;\; i = 1,2,.....C \qquad (3.28)$$

$$m = \frac{1}{P}\sum_{i=1}^{P} x_i \;\; ; \qquad (3.29)$$

P : Le nombre d'images d'apprentissage.

n_i : Le nombre d'images dans chaque classe.

C : Le nombre de classes.

2) Centrer les images dans chaque classe

Soustraire la moyenne de chaque classe de l'image de cette classe

$$\forall x \in X_i, X_i \in X, \hat{x} = x - m_i \qquad (3.30)$$

3) Centrer les moyennes de chaque classe

Soustraire la moyenne totale des moyennes de classes.

$$\hat{m}_i = m_i - m \qquad (3.31)$$

4) Calculer la matrice de dispersion intra-classes S_W (within class scatter matrix**)**

La matrice de dispersion intra-classes **(S_W)** est la somme des matrices de dispersion des classes. Pour la i^éme^ classe la matrice de dispersion S_i est calculée par la somme des matrices de covariance des images centrées.

$$S_i = \sum_{x \in X_i} \hat{x}^T \hat{x} \qquad (3.32)$$

La matrice de dispersion intra-classes **(S_W)** est la somme de toutes les matrices de dispersion.

$$S_W = \sum_{i=0}^{C} S_i \qquad (3.33)$$

Où C est le nombre de classe.

5) Calculer de la matrice de dispersion inter-classes S_B (between class scatter matrix**)**

La matrice de dispersion inter-classes **(S_B)** est la somme de dispersion entre classes.

$$S_B = \sum_{i=1}^{C} n_i \hat{m}_i \hat{m}_i \qquad (3.34)$$

Où n_i : Le nombre d'images dans la classe.

$\hat{m}_i$: La moyenne des classes.

Le but est de maximiser les distances inter-classes tout en minimisant les distances intra-classes, ce qui revient à retrouver la matrice de transformation W qui maximise le critère :

$$\mathrm{J}(\mathrm{W}) = \frac{\mathrm{W}^{\mathrm{T}} \mathrm{S}_b \mathrm{W}}{\mathrm{W}^{\mathrm{T}} \mathrm{S}_w \mathrm{W}} \qquad (3.35)$$

donc W est optimale pour :

$$\mathrm{W}_{opt} = arg\max_{\mathrm{W}}\left(\frac{|\mathrm{W}^{\mathrm{T}} \mathrm{S}_b \mathrm{W}|}{|\mathrm{W}^{\mathrm{T}} \mathrm{S}_w \mathrm{W}|}\right) = [w_1, w_2, ..., w_m] \quad (3.36)$$

6) Résoudre le problème de valeurs propres généralisé

- Résoudre le problème généralisé de vecteurs propres **(V)** et des valeurs propres (Λ) de la matrice de dispersion S_W et la matrice de dispersion S_B.

$$S_B V = \Lambda S_W V \qquad (3.37)$$

La solution est rendue par calcul des vecteurs propres et des valeurs propres de la matrice $S_W^{-1} * S_B$.

- Ordonner les vecteurs propres par ordre décroissant des valeurs propres correspondantes. La matrice de transformation de la **LDA** est constituée par les premiers vecteurs propres.

Implémentation de l'Analyse Discriminante Linéaire (LDA) :

1. *Calcul la matrice de dispersion intra-classes S_W* (équation **(3.33)**).
2. *Calcul de la matrice de dispersion inter-classes S_B* (équation **(3.34).**
3. *Calcul des valeurs et vecteurs propres de la matrice $S_W^{-1} * S_B$.*
4. *Ordonner les vecteurs propres par ordre décroissant des valeurs propres correspondantes.*
5. *La matrice de transformation de la **LDA** est les m premiers vecteurs propres (U_{LDA}).*

- **Inconvénient**

La **FLD** exige un grand nombre d'échantillons de l'ensemble d'apprentissage pour la bonne généralisation. Quand un tel besoin n'est pas répondu, la **FLD** crée un problème de sur-ajustement aux données d'apprentissage et ceci s'apprête mal aux nouvelles données de test.[273,274,275]

- **Solution** [276]

  Le modèle **D**iscriminant **L**inéaire **A**mélioré de **F**isher (**E**nhanced **F**isher **M**odel **EFM).**

3.2.2.5 Le modèle discriminant linéaire amélioré de Fisher (EFM)

Ce modèle discriminant linéaire de **Fisher** améliore la capacité de généralisation de la **FLD** en décomposant la procédure **FLD** en diagonalisation simultanée des deux matrices de dispersion intra-classe et inter-classe [279]. La diagonalisation simultanée est une étape sagement équivalente à deux opérations comme l'a souligné **Fukunaga** [278]. Blanchiment de la matrice de dispersion intra-classe et l'application de l'**ACP** sur la matrice de dispersion intre-classe en utilisant les données transformées. Durant l'opération du blanchiment de la matrice de dispersion intra-classe apparaisse dans le dénominateur de la séparabilité des petites valeurs propres qui tendent à capturer du bruit [279]. Pour atteindre des performances améliorées **l'EFM** préserve un équilibre approprié entre la sélection des valeurs propres (correspondant à la composante principale de l'espace de l'image originale) qui tiennent compte de la plupart de l'énergie spectrale des données brutes, c'est à dire, représentation adéquate et l'exigence que les valeurs propres de la matrice de dispersion intra-classe (de l'espace **ACP** réduit) ne sont pas trop petites, c'est-à-dire une meilleure généralisation.

Le choix de rang des composantes principales (***m***) pour la réduction de la dimension, prend en compte de l'ordre de grandeur de l'énergie spectrale. Les valeurs propres de la matrice de covariance fournissent un bon indicateur pour répondre au critère de l'énergie. Il faut ensuite calculer les valeurs propres de la matrices de dispersion intra-classe dans l'espace **ACP** réduit pour faciliter le choix du rang des composantes principales de sorte que l'exigence de grandeur est respectée. A cette fin, on effectue la **FLD** par des étapes comme décrit ci-dessous. En particulier, ces étapes **FLD** permettent de trouver les valeurs propres et les vecteurs propres de $S_W^{-1}S_b$ comme résultat de la diagonalisation simultanée de S_W et S_b.

Les étapes de l'**EFM** sont présentées comme suit :

- Blanchissons d'abord la matrice de dispersion intra-classe :

$$S_W V = V\Lambda \quad \text{et} \quad V^T V = I \qquad (3.38)$$

$$\Lambda^{-1/2} V^T S_W V \Lambda^{-1/2} = I \qquad (3.39)$$

Où V , $\Lambda \in \mathbf{R}^{m\times m}$ sont la matrice des vecteurs propres et la matrice diagonale des valeurs propres de S_W respectivement.

Les valeurs propres de la matrice de dispersion intra-classe dans l'espace **ACP** réduit peut être obtenu en (équation (**3.37**).

Donc, **EFM** diagonalise en premier lieu la matrice de dispersion intra-classe S_W (**3.38**) et (**3.39**). Notez que V et Λ sont les matrices des vecteurs propres et des valeurs propres correspondants aux vecteurs caractéristiques.

En second lieu **EFM** procède à calculer la nouvelle matrice de dispersion inter-classe comme suit: $\Lambda^{-1/2}V^{T}S_{b}V\Lambda^{-1/2} = K_{b}$ (**3.40**)

- Diagonalisons maintenant la nouvelle matrice de dispersion inter-classe K_b :

$$K_bV_b=V_b\Lambda_b \quad \text{et } V_b^{t}V_b=I \qquad (3.41)$$

Où V_b , $\Lambda_b \in \mathbf{R}^{m\times m}$ sont la matrice des vecteurs propres et la matrice diagonale des valeurs propres de K_b respectivement.

- La matrice de transformation globale de l'EFM est définie comme suit :

$$U=V\Lambda^{-1/2}V_b \qquad (3.42)$$

Implémentation du modèle Discriminant Linéaire Amélioré de Fisher (EFM) :

1. *Calcul la matrice de dispersion intra-classes S_W* (équation (**3.33**))**.**
2. *Calcul de la matrice de dispersion inter-classes S_B* (équation (**3.34**)).
3. *Calcul des valeurs (Λ) et vecteurs (V) propres de la matrice S_W.*
4. *Calculer la nouvelle matrice de dispersion inter-classe $K_b = \Lambda^{-1/2}V^{T}S_bV\Lambda^{-1/2}$*
5. *Calcul des valeurs (Λ_b) et vecteurs (V_b) propres de la matrice K_b.*
6. *Calcul de la matrice $U = V\Lambda^{-1/2}V_b$.*
7. *La matrice de transformation de l'EFM est les m premiers vecteurs de U (U_{EFM}).*

3.2.3 Méthodes non linéaires de réduction de dimension

Nous décrivons ici les méthodes non linéaires de réduction de dimension. Les techniques non linéaires peuvent être catégorisées en trois principaux types : les techniques essayant de préserver les propriétés globales des données d'apprentissage dans l'espace de faible dimension, les techniques s'attachant à préserver les propriétés locales des données d'apprentissage, et les techniques réalisant un alignement global de modèles linéaires.

3.2.3.1 Méthodes globales

Les méthodes globales de réduction non linéaires de dimension essaient de préserver les propriétés globales des données d'apprentissage dans le nouvel espace de faible dimension. On peut citer les techniques : **MDS**, Isomap, Diffusion Maps et les AutoEncoders Multi–Couches. Nous nous limitons à la description de la **K**ernel **PCA** qui fait l'objet de notre application au système de reconnaissance de visages monomodale et multi algorithmiques. Notre choix est justifié par le fait que la **KPCA** est une extension de la **PCA** qui est à la base de tous nos travaux. Nous voulons par l'introduction de noyaux améliorer les performances de notre système et aussi soulever les limites de la **PCA**.

Kernel PCA L'**A**nalyse en **C**omposantes **P**rincipales à **N**oyaux (ou **KPCA** pour **K**ernel **P**rincipal **C**omponent **A**nalysis) est la reformulation non linéaire de la technique linéaire classique qu'est l'**A**nalyse en **C**omposantes **P**rincipales en utilisant des fonctions à noyaux

[280]. Depuis plusieurs années, la reformulation de techniques classiques à l'aide de l'astuce du noyau a permis l'émergence de nombreuses techniques comme les machines à support de vecteurs (ou **SVM** pour **S**upport **V**ector **M**achine) [281]. L'**ACP** à noyaux calcule les principaux vecteurs propres de la matrice de noyaux plutôt que la matrice de covariance. Cette reformulation de l'**ACP** classique peut être vue comme une réalisation de l'**ACP** sur l'espace de grande dimension transformée par la fonction noyau associée. L'**ACP** à noyaux permet ainsi de construire des **M**appings **N**on **L**inéaires.

L'**ACP** à noyaux calcule d'abord la **matrice de noyaux** K des points x_i dont les entrées sont définis par : $k_{ij} = k(x_i; x_j)$ (**3.43**)

où k est la fonction noyau [258]. Ensuite, la matrice de noyaux K est centrée :

$$k_{ij} = k_{ij} - \frac{1}{n}\sum_l k_{il} - \frac{1}{n}\sum_l k_{jl} + \frac{1}{n^2}\sum_{lm} k_{lm} \quad (\mathbf{3.44})$$

Cette opération correspond à la soustraction de la moyenne des vecteurs caractéristiques dans l'**ACP** linéaire classique.

Les d principaux vecteurs propres v_i de la matrice de noyaux centrée sont ensuite calculés. Il peut être montré que les vecteurs propres α_i de la matrice de covariance (dans l'espace de grande dimension) sont des versions mises à l'échelle des vecteurs propres v_i de la matrice de noyaux.

$$\alpha i = \frac{1}{\sqrt{\lambda_i}}\ v_i \quad (\mathbf{3.45})$$

La technique **KPCA** est une méthode basée sur les noyaux et ses performances dépendent alors grandement du choix de la fonction noyau k. Les noyaux classiquement utilisés sont le noyau linéaire (cela revient alors à effectuer une **AC** classique), le noyau polynomial ou encore le noyau gaussien [281].

L'**A**nalyse en **C**omposantes **P**rincipales à **N**oyaux a été appliquée avec succès à plusieurs problèmes comme la reconnaissance de la parole [282], ou la détection de nouveaux éléments d'un ensemble [283]. Un gros **défaut** de l'**A**nalyse en **C**omposantes **P**rincipales à **noyaux** est que la **taille de la matrice de noyaux est le carré du nombre d'échantillons** de l'ensemble d'**apprentissage** ce qui peut rapidement être prohibitif. Une approche permettant de résoudre ce problème peut être trouvé dans [284].

3.2.3.2 Méthodes locales

Les méthodes dites locales de réduction de la dimension essaient de préserver les propriétés dans le voisinage des points. Ce type de technique repose sur la supposition qu'en préservant les propriétés locales des données, les propriétés globales de la variété le seront tout autant. La plupart de ces techniques peuvent se ramener à une définition valide dans le

cadre de l'**ACP** à **Noyaux** à l'aide de noyaux locaux spécifiques [285,287]. Sont présentés dans [65] les méthodes **LLE**, **L**aplacian **E**igenmaps, **H**essian **LLE** et **LTSA**.

3.3 Classification

3.3.1 Classification par mesure de similarités

3.3.1.1 Comparaisons entre deux vecteurs

Lorsqu'on souhaite comparer deux vecteurs de caractéristiques issus du module d'extraction de caractéristiques d'un système biométrique, on peut soit effectuer une mesure de similarité (ressemblance), soit une mesure de distance (divergence).

La première catégorie de distances est constituée de distances Euclidiennes et sont définies à partir de la *distance de Minkowski d'ordre* p dans un espace euclidien $\boldsymbol{R^N}$ (N déterminant la dimension de l'espace euclidien).

Considérons deux vecteurs $X = (x_1, x_2, ..., x_N)$ et $Y = (y_1, y_2, ..., y_N)$, la *distance de Minkowski d'ordre* p notée $\boldsymbol{L_p}$ est définie par :

$$\mathbf{L}_p = \left(\sum_{i=1}^{N} |x_i - y_i|^p \right)^{1/p} \qquad (3.46)$$

Nous allons présenter quelques mesures de distance dans l'espace original des images puis dans *l'espace de Mahalanobis*.

1) Distances Euclidiennes

- **Distance City Block (L_1)**

Pour $\boldsymbol{p = 1}$, on a :

$$\mathbf{L}_1(x, y) = \sum_{i=1}^{N} |x_i - y_i| \qquad (3.47)$$

- **Distance Euclidienne (L_2)**

Pour $\boldsymbol{p = 2}$, on a :

$$\mathbf{L}_2(x, y) = \sqrt{\sum_{i=1}^{N} |x_i - y_i|^2} \qquad (3.48)$$

2) Distances dans l'Espace de Mahalanobis

Soit $\boldsymbol{u}$ et $\boldsymbol{v}$ deux vecteurs propres de $\boldsymbol{J_m}$, issus de l'algorithme **PCA**, et $\boldsymbol{m}$ et $\boldsymbol{n}$ deux vecteurs de $\boldsymbol{E_{Mah}}$. Soit $\boldsymbol{\lambda_i}$ les valeurs propres associées aux vecteurs $\boldsymbol{u}$ et $\boldsymbol{v}$, et $\boldsymbol{\sigma_i}$ l'écart type, alors on définit $\boldsymbol{\lambda_i = \sigma_i^2}$. Les vecteurs $\boldsymbol{u}$ et $\boldsymbol{v}$ sont reliés aux vecteurs $\boldsymbol{m}$ et $\boldsymbol{n}$ à partir des relations suivantes :

$$m_i = \frac{u_i}{\sigma_i} = \frac{u_i}{\sqrt{\lambda_i}} \quad et \quad n_i = \frac{v_i}{\sigma_i} = \frac{v_i}{\sqrt{\lambda_i}} \qquad (3.49)$$

- **Mahalanobis L_1 ($MahL_1$)**

L_1 est définie par :

$$Mah_{\mathbf{L}_1}(u,v) = \sum_{i=1}^{N} |m_i - n_i| \qquad (3.50)$$

- **Mahalanobis L_2 ($MahL_2$)**

L_2 est définie par : $$Mah_{\mathbf{L}_2}(u,v) = \sqrt{\sum_{i=1}^{N} |m_i - n_i|^2} \qquad (3.51)$$

Par défaut, lorsqu'on parle de distance de Mahalanobis, c'est à cette distance que l'on doit se référer.

3.3.1.2 Comparaisons entre deux matrices

Dans **Yang** a proposé une nouvelle mesure de similarité au plus proche voisin pour la reconnaissance de visages. La distance de **Yang** se base sur la classification de matrices caractéristiques obtenues par l'**ACP2D**. Cette distance a été, également, adoptée par **Visani** et **al**. [Vis04] et **Bengherabi** [Ben08] elle est définie pour deux matrices caractéristiques réduites $\boldsymbol{Y_i} = [\boldsymbol{y_1^{(i)}} \boldsymbol{y_2^{(i)}} \dots \boldsymbol{y_{d_1}^{(i)}}]$ et $\boldsymbol{Y_j} = [\boldsymbol{y_1^{(j)}} \boldsymbol{y_2^{(j)}} \dots \boldsymbol{y_{d_1}^{(j)}}]$ comme suit :

$$d(Y_i, Y_j) = \sum_{k=1}^{d_1} \left\| y_k^{(i)} - y_k^{(j)} \right\|_2 \qquad (3.52)$$

Où $\left\| y_k^{(i)} - y_k^{(j)} \right\|_2$ désigne la distance Euclidienne entre les deux vecteurs $y_k^{(i)}$ et $y_k^{(j)}$d'où:

$$d(Y_i, Y_j) = \sum_{k=1}^{d_1} \left(\sum_{h=1}^{n} \left(y_{hk}^{(i)} - y_{hk}^{(j)} \right)^2 \right)^{1/2} \qquad (3.53)$$

Avec $\boldsymbol{y_k^{(i)}} = [\boldsymbol{y_{1k}^{(i)}} \boldsymbol{y_{21}^{(i)}} \dots \boldsymbol{y_{d_1 k}^{(i)}}]$ **et** $\boldsymbol{y_k^{(j)}} = [\boldsymbol{y_{1k}^{(j)}} \boldsymbol{y_{21}^{(j)}} \dots \boldsymbol{y_{d_1 k}^{(j)}}]$.

Dans [Zuo05], **Zuo** a proposé l'**A**ssembled **M**atrix **D**istance: **AMD** définie par:

$$d(Y_i, Y_j) = \left(\sum_{k=1}^{d_1} \left(\sum_{h=1}^{n} \left(y_{hk}^{(i)} - y_{hk}^{(j)} \right)^2 \right)^{p/2} \right)^{1/p} \text{ avec } p \succ 0 \qquad (3.54)$$

pour laquelle la distance de Yang est obtenue pour p = 1 et la distance de Frobenius pour p = 2.

3.3.2 Classification par Machine à Vecteurs de Support (SVM)

Le **SVM** (**S**upport **V**ector **M**achine) est une nouvelle technique d'apprentissage statistique, proposée par **V. Vapnik** en **1995**. Elle permet d'aborder des problèmes très divers comme le classement, la régression, la fusion, etc…

Depuis son introduction dans le domaine de la Reconnaissance de Formes (**RdF**), plusieurs travaux ont pu montrer l'efficacité de cette technique principalement en traitement d'image. L'idée essentielle consiste à projeter les données de l'espace d'entrée (appartenant à des

classes différentes) non linéairement séparables, dans un espace de plus grande dimension appelé espace de caractéristiques, de façon à ce que les données deviennent linéairement séparables. Dans cet espace, la technique de construction de l'hyperplan optimal est utilisée pour calculer la fonction de classement séparant les classes. Tels que les vecteurs appartenant aux différentes classes se trouvent de différents côtés de l'hyperplan.

Le but du **SVM** est donc de trouver une séparatrice qui minimise l'erreur de classification sur l'ensemble d'apprentissage mais qui sera également performante en généralisation sur des données non utilisées en apprentissage. Pour cela le concept utilisé est celui de marge (d'où le nom de séparateurs à vaste marge). La marge est la distance quadratique moyenne entre la séparatrice et les éléments d'apprentissage les plus proches de celle-ci appelés vecteurs de support (**figure 3.3**). Ces éléments sont appelés vecteurs de support car c'est uniquement sur ces éléments de l'ensemble d'apprentissage qu'est optimisée la séparatrice.

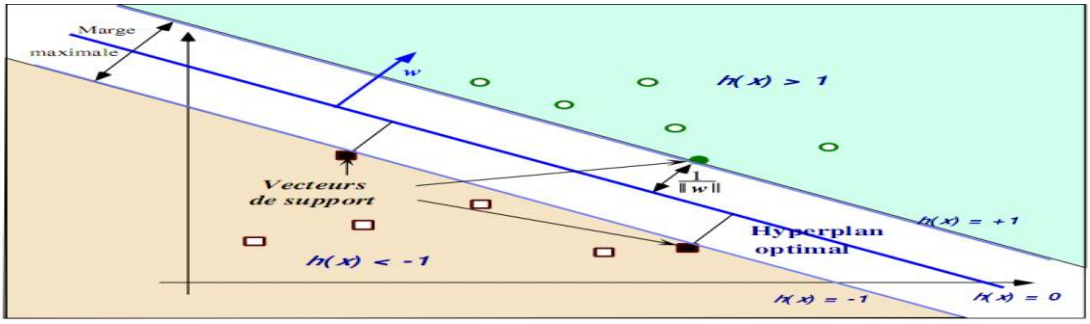

Figure 3.3 Principe de la technique **SVM**
(hyperplan optimal, vecteurs de supports, marge maximale).

Hyperplan optimal : est un Hyperplan qui classe correctement les données (lorsque c'est possible) et qui se trouve le plus loin possible de tous les exemples, on peut dire aussi que cet hyperplan maximise la marge**.**

Vecteurs de support : ce sont les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan.

La marge : est la distance entre l'hyperplan et les exemples. La marge est calculée à partir du produit scalaire entre les vecteurs situés la frontière de chaque classe et le vecteur unitaire normal de l'hyperplan séparateur. [287]

Tout classifieur a pour but de classifier un élément $\boldsymbol{x}$, ici $x = (s_1,\ldots s_N)$ est un vecteur de scores de dimension N, dans l'une des classes possibles. Dans notre problème il y a deux classes, Client ou Imposteur, dont l'étiquette sera noté $\boldsymbol{y}$ avec y = -1, 1, -1 correspondant à la classe des Imposteurs et 1 à la classe des Clients. Le classifieur a donc pour but de déterminer $\boldsymbol{f}$ telle que :

$$\boldsymbol{Y = f(x)} \qquad (3.55)$$

Le **SVM** a pour but de trouver la meilleure séparatrice linéaire (en terme d'émarge maximale, c'est à dire la meilleure généralisation) dans l'espace transformée par la fonction de noyau ***K***, c'est à dire de déterminer le vecteur ***w*** et la constante *b* tels que la séparatrice ait pour équation : $\boldsymbol{w.k(x)+b=0}$ (**3.56**)

La distance entre un point de l'espace x_i et l'hyperplan d'équation $w.K(x)+b = 0$ est égal à :

$$h(x_i) = \frac{w.K(x_i)+b}{\|w\|} \quad (3.57)$$

Pour maximiser la marge, il faut donc minimiser $\|w\|$ en maximisant $\boldsymbol{w.K(x_i)+b}$ pour les $\boldsymbol{x_i}$ définis comme vecteurs de support. Ces **vecteurs de supports** sont les x_i pour i = 1 : *m* de la base d'**apprentissage** tels que : $\boldsymbol{w.K(x_i)+b=\pm 1}$. (**3.58**)

La résolution de ce problème d'optimisation est faite par l'utilisation des multiplicateurs de Lagrange où le Lagrangien est donné par :

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i(w.K(x_i)+b-1)\right) \quad (3.59)$$

Avec les coefficients $\boldsymbol{\alpha_i}$ appelés multiplicateurs de Lagrange. Pour résoudre ce problème d'optimisation, il faut minimiser le Lagrangien par rapport à ***w*** et ***b*** et le maximiser par rapport à $\boldsymbol{\alpha}$. Dans la pratique, il est souvent impossible de trouver un séparateur linéaire (même dans l'espace transformé par la fonction noyau) car il y a toujours des erreurs de classification. Il a donc été introduit par **Vapnik** [288] la technique de marge souple. Ce principe de marge souple tolère les mauvais classements par l'introduction de variables ressorts $\boldsymbol{x_i}$ qui permettent de relâcher les contraintessur les éléments d'apprentissage qui ne doivent plus être à une distance supérieure ou égale à 1 de la marge (l'égalité correspondant aux vecteurs de support),mais à une distance supérieure ou égale à $1-\boldsymbol{\zeta_i}$, c'est à dire :

$$y_i(w.K(x_i)+b \geq 1-\xi_i \quad (3.60)$$

Avec $\boldsymbol{x_i} \geq 0$ pour i = 1 : *M*; *M* étant le nombre d'éléments de la base d'apprentissage.

Le problème d'optimisation est donc modifié et le Lagrangien devient :

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{M} \xi_i - \sum_{i=1}^{M} \alpha_i \left(y_i(w.K(x_i)+b-1)\right) \quad (3.61)$$

Où *C* est une constante strictement positive qui permet de régler le compromis entre le nombre d'erreurs de classification et la largeur de la marge. Cette constante est en général déterminée empiriquement par validation croisée sur l'ensemble d'apprentissage. [58]

3.4 La Décision

Pour estimer la différence entre deux images, il faut introduire une mesure de similarité. Il est important de noter que le système de vérification automatique de visage se base en sa totalité sur la méthode de localisation.

3.5 Conclusion

Dans ce chapitre, nous avons présenté un certain nombre de méthodes utilisées dans les cinq étapes d'un système de reconnaissance de visage. Nous nous sommes limités aux méthodes proches du modèle de notre approche. Toutes ces méthodes ont des avantages et des inconvénients selon la complexité, le besoin de données d'apprentissage ou les paramètres à optimiser. Le choix d'une méthode va donc dépendre de l'application visée selon le compromis entre performance, simplicité, confort et quantité de données d'apprentissage disponible. Toutes ces méthodes seront implémentées puis comparées sur la base de données **XM2VTS**, **Casia 3D** et **Casia (3D expression)** dans les **chapitres 5 et 6**. L'utilisation de systèmes biométriques multimodaux a été encouragée par la menace d'usurpation d'identité, où l'on estime qu'un système monomodal est insuffisant pour authentifier les individus. La multi modalité repose sur des techniques de fusion. La multi modalité est l'utilisation de plusieurs systèmes biométriques. La combinaison de plusieurs systèmes a pour objectif d'en diminuer les limitations. En effet, l'utilisation de plusieurs systèmes a pour premier but d'améliorer les performances de reconnaissance. En augmentant la quantité d'information discriminante de chaque personne, on souhaite augmenter le pouvoir de reconnaissance du système.

Plusieurs méthodes non linéaires de reconnaissance ont été présentées tout au long de ce chapitre tel que la théorie des noyaux reproduisant a permis le développement fulgurant d'une classe d'algorithmes de reconnaissance des visages dont la formulation ne dépend pas de la nature des données traitées, ni de l'espace de représentation adopté pour résoudre les problèmes. Au delà de ce caractère universel, celles que l'on range désormais sous le qualificatif de méthodes à noyau doivent également leur succès à l'essor de la théorie statistique de l'apprentissage, au sein de laquelle la prédiction de leurs performances en généralisation fait aujourd'hui encore l'objet d'études approfondies.