Apprentissage de métriques et méthodes à noyaux appliqués à la reconnaissance de personnes dans les images

Alexis Mignon

université de Caen
Basse-Normandie

UNIVERSITÉ DE CAEN BASSE NORMANDIE
U.F.R. de Sciences
ÉCOLE DOCTORALE SIMEM

T H È S E

Présentée par

M. Alexis MIGNON

et soutenue le
13 Décembre 2012

en vue de l'obtention du

DOCTORAT de l'UNIVERSITÉ de CAEN

Spécialité : Informatique et applications

Arrêté du 07 août 2006

Titre :

Apprentissage de métriques et méthodes à noyaux appliqués à la reconnaissance de personnes dans les images

MEMBRES du JURY :

M. Stéphane CANU	Professeur des Universités	INSA de Rouen	(***Rapporteur***)
M. Christophe GARCIA	Professeur des Universités	INSA de Lyon	(***Rapporteur***)
Mme Marinette REVENU	Professeur émérite des Universités	Université de Caen	
M. Frédéric JURIE	Professeur des Universités	Université de Caen	(***Directeur de thèse***)

UNIVERSITÉ DE CAEN BASSE NORMANDIE
U.F.R. de Sciences
ÉCOLE DOCTORALE SIMEM

T H È S E

Présentée par
M. Alexis MIGNON

et soutenue le
13 Décembre 2012

en vue de l'obtention du

DOCTORAT de l'UNIVERSITÉ de CAEN

Spécialité : Informatique et applications

Arrêté du 07 août 2006

Titre :

Apprentissage de métriques et méthodes à noyaux appliqués à la reconnaissance de personnes dans les images

MEMBRES du JURY :

M. Stéphane CANU	Professeur des Universités	INSA de Rouen	(***Rapporteur***)
M. Christophe GARCIA	Professeur des Universités	INSA de Lyon	(***Rapporteur***)
Mme Marinette REVENU	Professeur émérite des Universités	Université de Caen	
M. Frédéric JURIE	Professeur des Universités	Université de Caen	(***Directeur de thèse***)

Table des matières

CHAPITRE 1

La reconnaissance de personnes dans les images

LA reconnaissance des personnes dans les images suscite un vif attrait dans la communauté scientifique, à un tel point qu'il serait difficile de faire une revue exhaustive de l'ensemble des travaux sur le sujet. Ceci s'explique d'une part par l'énorme intérêt applicatif (aussi bien en matière de sécurité, vidéosurveillance, etc. que pour les applications multimédia grand public) mais aussi de par le défi que cela représente pour les algorithmes de vision artificielle. En effet, ceux-ci doivent être capables de faire face à la grande variabilité des visages eux-mêmes tout autant qu'aux variations des paramètres de prise de vue (pose, éclairage, coupe de cheveux, expression, arrière-plan, etc.).

Dans ce chapitre, nous présentons de manière générale les différentes tâches que recouvre la notion de reconnaissance des personnes dans les images, expliquons pourquoi ces tâches sont difficiles, présentons les différentes bases du domaine public utilisable pour expérimenter des algorithmes de reconnaissance, expliquons comment les méthodes à noyaux peuvent apporter à la reconnaissance de personnes, et, finalement, présentons les contributions de la thèse.

Sommaire

1.1 La reconnaissance de personnes dans les images : besoins et difficultés

La reconnaissance de personnes dans des images est un terme qui couvre différents besoins applicatifs. Nous présentons dans la suite de cette section ce que sont ces besoins, et expliquons pourquoi la variabilité des apparences des personnes dans les images rend ces tâches difficiles. Nous concluons cette section en présentant les bases d'images du domaine publique utilisable pour évaluer les technologies de reconnaissance de personnes.

1.1.1 Les différents types de besoins

Les dispositifs de reconnaissance des personnes peuvent être divisés en 2 grandes catégories : les systèmes *coopératifs* et les systèmes *non-coopératifs*.

La plupart des dispositifs biométriques sont à classer dans la catégorie des systèmes coopératifs car ils nécessitent que le sujet à reconnaître effectue une action spécifique prévue par un protocole d'identification. Le cas de l'identification de personnes par une analyse de l'iris, ou par mesure de la forme de la main sont deux exemples typiques de système coopératifs.

Dans cette thèse, nous nous intéresserons plutôt aux systèmes non-coopératifs ou faiblement coopératifs pour lesquels les données sont capturées dans des situations réelles, dans des conditions de prise de vue non contrôlées, avec des poses quelconques des personnes, des occlusions partielles, etc. Toutefois, cette distinction n'est pas toujours si tranchée : par exemple, dans le cas de la reconnaissance de personnes dans des photographies prises entre amis, les sujets ont souvent posé, et ont tendance à se présenter face à la caméra, visage visible, avec un éclairage convenable. Ce contexte diffère du cas où des personnes sont photographiées à leur insu par un système de surveillance. C'est ce que nous entendons par dispositifs *faiblement coopératifs* : les sujets apparaissent dans des conditions favorables sans qu'ils n'aient à effectuer d'actions particulières.

Dans la vie de tous les jours, l'information visuelle que nous utilisons le plus pour reconnaître les personnes est bien évidemment le visage. C'est sur lui que se concentre notre attention et le cerveau humain est particulièrement efficace dans cette tâche. Si le visage est d'une importance capitale, d'autres indices visuels tels que la démarche, la silhouette, la tenu vestimentaire, peuvent nous permettre de correctement reconnaître une personne lorsque le visage n'est pas visible.

Les expériences réalisées dans cette thèse se focalisent principalement, sur la reconnaissance des visages, et dans une moindre mesure sur la reconnaissance de personnes basée sur l'aspect global de la personne. Le terme de *reconnaissance de personnes* est un terme générique qui recouvre en réalité plusieurs tâches différentes qu'il est nécessaire de différencier.

Supposons que nous disposions d'un ensemble $\mathcal{E}$ de N personnes connues, il est possible d'envisager différentes tâches de reconnaissance. Nous en distinguons trois.

L'authentification. Étant donné l'image d'une personne (typiquement un visage de cette personne), l'authentification consiste à prédire si la personne appartient à l'ensemble $\mathcal{E}$ des personnes connues. Une application pratique de l'authentification est le contrôle d'accès. Cette tâche peut être vue comme une tâche de classification binaire.

La vérification. Dans cette tâche, deux images sont présentées à l'entrée du système, qui doit prédire si ces images représentent ou pas la même personne. Cette tâche peut aussi être vue comme une tâche d'authentification mais avec un ensemble $\mathcal{E}$ ne contentant qu'une seule image de visage. En terme d'application, il s'agit par exemple de vérifier qu'une personne correspond bien à la photographie présente sur sa pièce d'identité.

L'identification. Cette tâche consiste à décider si une personne appartient à l'ensemble $\mathcal{E}$ des personnes connues et à déterminer, parmi les personnes représentées dans $\mathcal{E}$, de laquelle il s'agit. Cette tâche complète la tâche d'authentification avec en plus une information d'identité. En pratique, une reconnaissance complète de visages pourrait remplacer ou compléter les dispositifs de contrôle d'accès existants.

La ré-identification. La ré-identification consiste à retrouver une même personne capturée par des caméras depuis des points de vue différents. Une application typique pourrait être le suivi de personnes dans les aéroports/gares à partir des enregistrements des caméras de surveillance. Elle peut utiliser des images de visages, ou, lorsque le visage n'est pas visible ou que l'information n'est pas exploitable (sur une photographie prise d'une distance trop grande ou de trop mauvaise qualité, par exemple), utiliser l'apparence globale de la personne.

1.1.2 La représentation des visages

Comme nous l'avons signalé plus haut, notre travail sur la reconnaissance des personnes porte essentiellement sur la reconnaissance de visages. De plus nous nous limitons aux informations contenues dans des images 2D. C'est pourquoi nous nous focalisons, dans la suite de ce chapitre, à la représentation des visages dans les images.

Nous supposons nous trouver ici dans la situation où deux visages ont été détectés dans deux images différentes, et où les images ont été approximativement alignées. L'alignement [66, 68] consiste à fixer une boite englobante autour du visage, ce qui fixe la position du visage et son échelle dans l'image. Force est de reconnaitre que même s'il s'agit de la même personne dans les deux images et que l'alignement est réalisé de manière parfaite, les pixels contenus dans les deux boites peuvent être très différents. Ces différences proviennent principalement (a) des variations d'apparence provoquées par des changements de pose 3D, (b) celles provoquées par des changements d'illumination, (c) celles provoquées par les changements d'expression.

Prise en compte des variations de pose. Quelle que soit la tâche envisagée, la tâche va reposer sur une mesure de similarité ou un classifieur. Une approche possible consiste à spécialiser ces mesures de similarité ou classifieurs à une pose particulière [92] – dans ce cas, seuls les visages de poses similaires sont comparés – ou à une paire de poses [96, 61, 106] . Ces approches nécessitent toutefois plusieurs images d'un même visage dans plusieurs poses différentes, ainsi qu'une méthode d'estimation précise de la pose d'un visage à partir d'une image.

D'autres approches telles que les modèles déformables [68] ou les modèles d'apparence active [32] utilisent des modèles 2D du visage qui sont alignés sur les images par déformation, pouvant ainsi compenser des variations de pose. En pratique, ces techniques ne sont généralement capables de traiter qu'une palette restreinte de variations. Elles sont, en particulier, sensibles aux problèmes d'auto-occultation liés à la nature tridimensionnelle des visages.

Plus récemment, des techniques basées sur des modèles 3D des visages ont été développées. Elles consistent à estimer la pose 3D d'un modèle 3D du visage à partir d'une représentation 2D, voire à reconstruire le visage à partir d'une image et d'un modèle générique de visage [17]. Les visages sont ensuite placés dans une pose normalisée [17] avant d'être comparés. Les paramètres des modèles 3D peuvent également être comparés directement entre eux. Un inventaire des méthodes de comparaison de visages en 3D peut-être trouvée dans [22]. L'ajustement d'un modèle 3D sur une image 2D et la comparaison de modèles 3D restent cependant des problèmes non résolus.

Prise en compte des variations d'illumination. A l'échelle du pixel, les variations dues aux différences d'illumination du visage peuvent être beaucoup plus importantes que les différences entre

deux personnes distinctes dans les mêmes conditions d'éclairage [4]. Les travaux théoriques de Belhumeur et Kriegman [11] ont montré que l'ensemble des variations dans l'espace des pixels dues aux différences d'illumination reposent sur une variété de faible dimension qu'ils ont appelée *le cône d'illumination*. Basri et Jacobs [10] ont, par exemple, montré comment calculer un sous-espace de faible dimension pour obtenir une approximation du cône d'illumination. Le passage de la théorie aux applications pratiques reste un vaste sujet de recherche.

En pratique, des solutions empiriques sont souvent utilisées pour faire face à ces changements d'illumination. Une solution simple consiste à effectuer une normalisation de la distribution des niveaux de gris dans l'image [105]. D'autres auteurs ont proposé l'utilisation de méthodes de filtrage passe-bande adaptées [109], ou l'utilisation de descripteurs visuels sensibles aux gradients de l'image, moins sensibles aux changements d'illumination [83, 5, 38].

Prise en compte des variations d'expression. Les modèles déformables 2D ou 3D [68, 17] ou des modèles d'apparence active [32] peuvent être utilisés également pour capturer l'expression des visages. Comme pour la pose, les visages peuvent être ramenés à une expression normalisée de laquelle des caractéristiques indépendantes de l'expression peuvent être extraites. Les calculs impliqués dans l'ajustement de tels modèles sont néanmoins lourds et peu robustes, en particulier en présence de variations de pose et d'éclairage.

Les modèles multilinéaires ou les *tensorfaces* [76, 120] utilisent, quant à eux, une généralisation de la décomposition en valeurs singulières pour les tenseurs multidimensionnels . Ces méthodes sont capables d'apprendre des sous-espaces différents pour chaque type de variations, mais nécessitent une quantité importante d'images et contenant les variations d'expressions désirées, pour chacune des personnes.

Certains systèmes ignorent simplement la zone de la bouche qui est la plus sujette à des modifications lors des variations d'expression. Un tour d'horizon plus complet des différents problèmes rencontrés en reconnaissance de visages et de leur traitement peut être trouvé dans [69].

1.1.3 Bases de données publiques utilisées

De nombreuses bases de données de visages sont disponibles pour la recherche sur la reconnaissance de personnes/visages. Ces bases d'images varient de par leur taille et leur objectif. Un certain nombres de ces bases ont été créé pour la reconnaissance de visages dans des conditions expérimentales particulières. Constituer une telle base de données sur une courte période de temps et à un endroit particulier peut présenter des avantages pour certains types de recherches. Notamment, l'expérimentateur a un meilleur contrôle sur la variabilité des paramètres dans la base.

Toutefois, lorsque l'on veut étudier des problèmes de reconnaissance de visages plus génériques, il devient nécessaire de tester les algorithmes sur des bases de données comportant un grand nombre de visages différents avec des paramètres de prise de vue variés. La création d'une telle base en laboratoire permet de contrôler précisément les paramètres de prise de vue. C'est une opération très coûteuse en temps et il est ensuite difficile de déterminer quelles distributions de paramètres seront les plus utiles pour avoir une base de données pertinente :

- Quel pourcentage de sujets doit porter des lunettes de soleil ou une barbe ?
- Combien doivent sourire ?
- Quel type d'arrière-plan doit être utilisé ?

1.1.3.1 La base Multi PIE

La base CMU Multi PIE [57] est née de la prise en compte des principales causes de variabilité de l'apparence des visages. Le «PIE» signifie «Pose Illumination et Expression». La base fournit donc

des séries de visages de plusieurs personnes prises sous différents points de vue, sous différentes conditions d'éclairage et avec différentes expressions.

Les images ont été capturées en laboratoire avec un système complexe d'appareils photos et de flash positionnés autour de la personne à photographier. La figure 1.1 montre l'arrangement spatial des appareils photos par rapport au sujet ainsi que les étiquettes utilisées pour identifier les appareils et donc le point de vue associé. Quatre sessions d'acquisition ont été effectuées sur une durée de 5 mois.

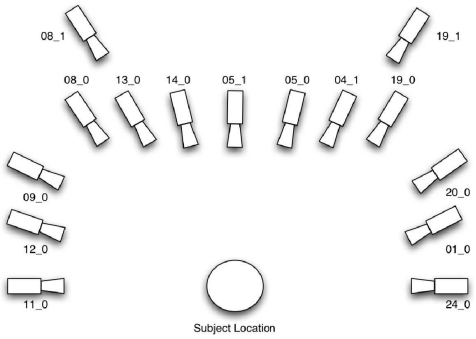


FIGURE *1.1: Multi PIE : étiquettes et localisation des appareils photographiques utilisés pour capturer les images. 13 appareils sont localisés à hauteur de la tête et positionnés tous les 15°. Deux appareils supplémentaires (08_1 and 19_1), sont placés au-dessus du sujet simulant des caméras de vidéo-surveillance. Figure reproduite d'après [57].*

1.1.3.2 Labeled Faces in the Wild (LFW)

Contrairement à la base Multi PIE, la base de visages ***Labeled Faces in the Wild*** (LFW) [66] a été créée volontairement à partir de photographies n'ayant pas été prises spécifiquement pour le problème de la reconnaissance de visages automatique. Elle a pour but de proposer une base de visages présentant une large gamme de variations telles que celles que l'on rencontre dans la vie courante. Cela inclut des variations de pose, d'éclairage, d'expression, d'arrière-plan, d'ethnicité, d'âge, de sexe, d'habillement, de coupe de cheveux, de qualité de prise de vue, de saturation des couleurs, de mise au point, et d'autres paramètres encore. La base LFW a été conçue pour traiter le problème de la *vérification* (appelée également parfois *correspondance de paires*), mais peut tout à fait être utilisée pour les autres types de problèmes énoncés précédemment.

La base LFW contient 13233 images de 5479 personnes différentes collectées sur le site de ***Yahoo ! News***. Parmi ces personnes, 1680 ont au moins deux images les représentant. Les autres personnes n'en ont qu'une. Les images sont organisées en deux jeux de données appelés ***vues***, conçus pour formaliser et faciliter les expérimentations.

Construction de la base. Une fois les images collectées, elles ont été soumises à l'algorithme de détection de visages de Viola-Jones [1] [122]. Les doublons, c'est-à-dire les images provenant de la même photographie d'origine, sont éliminés. Chaque image est étiquetée manuellement d'après le nom de la personne qu'elle représente. Le texte accompagnant l'image lors de sa collecte est utilisé comme aide pour déterminer l'identité de la personne. Enfin, le résultat de l'algorithme de détection de visages est utilisé pour recadrer et redimensionner chaque image de sorte qu'elle occupe un carré de 250×250 pixels.

1. Plus précisément son implémentation fournie par la bibliothèque OpenCV.

Une fois redimensionnées, les images sont enregistrées au format ***JPEG*** dans des fichiers dont le nom est constitué à partir du nom de la personne et du numéro de l'image. Par exemple, pour la 12ème image représentant George W. Bush, le nom de fichier serait `George_W_Bush_0012.jpg` et pour la 8ème image de John Kerry, on aurait `John_Kerry_0008.jpg`.

Organisation de la base.

Vue 1 : Destinée à l'expérimentation des différents modèles et au développement des algorithmes, elle se compose de deux sous-ensembles : un pour l'entraînement, un pour le test. Les données d'entraînement se composent de 1100 paires d'images pour lesquelles chaque élément de la paire représente une même personne et de 1100 paires d'images pour les lesquelles les éléments de la paire représentent des personnes différentes. Le jeu de données de test est composé de 500 paires d'images représentant une même personne et de 500 paires d'images représentant deux personnes différentes.

Vue 2 : Destinée à la mesure finale des performances, elle est supposée n'être jamais utilisée pendant la conception et l'ajustement des algorithmes. Elle se compose de 10 séries de 300 paires d'images positives et 300 paires d'images négatives, soit 6000 paires en tout. Les dix séries sont utilisées pour réaliser une validation croisée.

Méthodes d'apprentissage restreintes ou non aux images. Les vues sont données sous la forme de listes de paires d'images. Comme indiqué précédemment, chaque personne visible dans la base est identifiée par son nom. Les images sont identifiées par le nom de la personne qu'elles représentent et un numéro. Les vues sont simplement une liste où chaque ligne est de la forme :

```
name n1 n2
```

dans le cas d'une paire positive. `name` est le nom de la personne et `n1` et `n2` les numéros des images constituant la paire.

Pour une paire négative, la ligne sera de la forme :

```
name1 n1 name2 n2
```

Ce qui signifie que la paire est constituée de l'image `n1` de la personne `name1` et de l'image `n2` de la personne `name2`.

Par exemple, il est possible de rencontrer les deux lignes suivantes :

```
George_W_Bush  10 24
```

et

```
George_W_Bush  12 John_Kerry 8
```

Dans les méthodes dites *restreintes aux images*[2], il n'est autorisé d'utiliser que les paires indiquées, sans prendre en compte l'information donnée par le nom de la personne. Toutefois, le nombre de paires des vues 1 et 2 étant limité, il peut être souhaitable de générer d'autres paires d'images à partir des images contenues dans une vue donnée. Ceci permet d'augmenter la taille des jeux d'entraînement et de test. Pour cela, on utilise l'information contenue dans le nom de la personne. Les méthodes utilisant l'information sur l'identité de la personne pour créer d'autres paires sont qualifiées de non-restreintes[3].

2. En anglais : ***image-restricted*** *methods*.
3. En anglais : ***unrestricted*** *methods*.

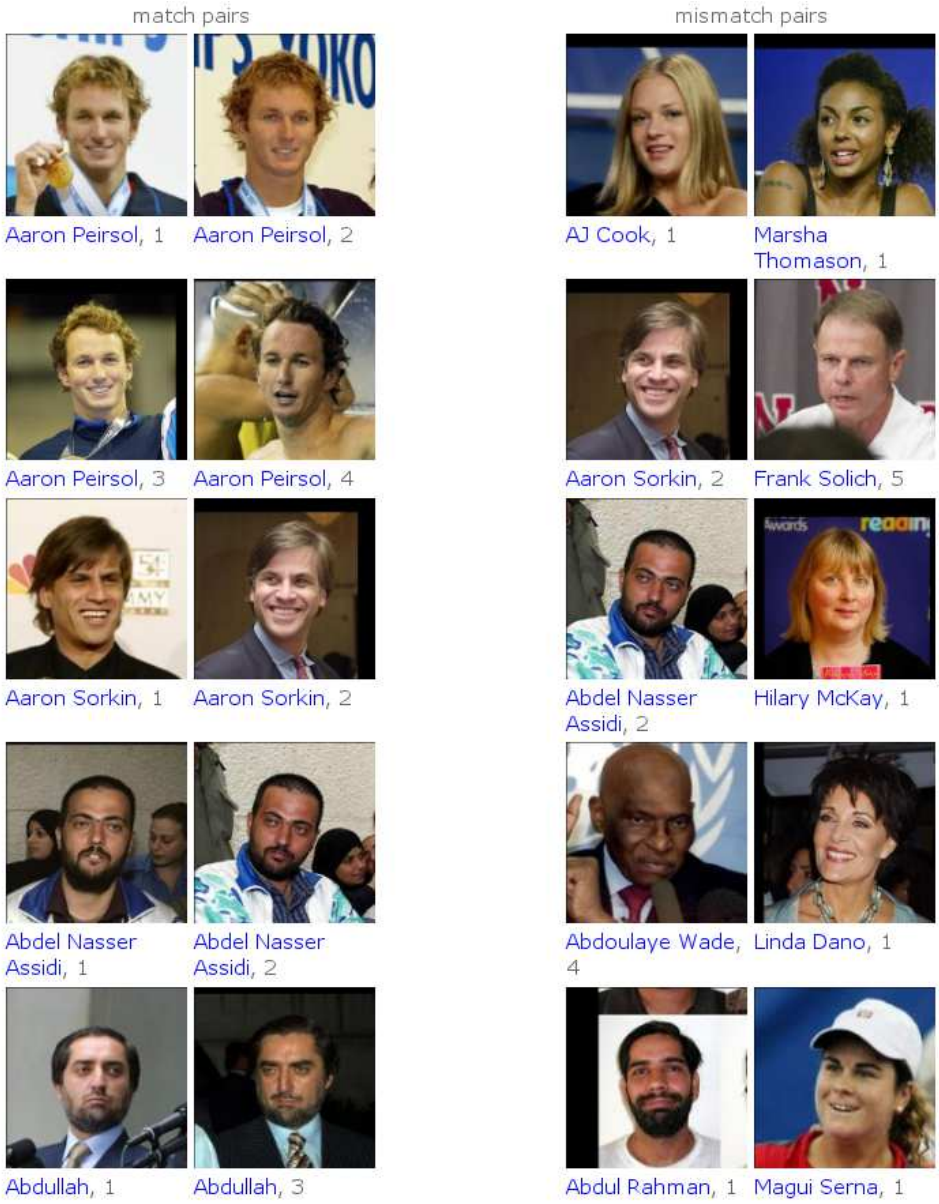


FIGURE *1.2: Exemples de paires d'images issues du jeu d'entraînement de la vue 1 de la base LFW. À gauche, des paires positives ; à droite, des paires négatives.*

1.1.3.3 La base CUFSF.

La base CHUK Face Sketch FERET (CUFSF) [125] inclut les photographies de 1194 personnes issues de la base de donnée FERET [93]. Pour chaque personne, une esquisse a été dessinée par un artiste avec l'intention d'exagérer certaines caractéristiques de leur apparence.

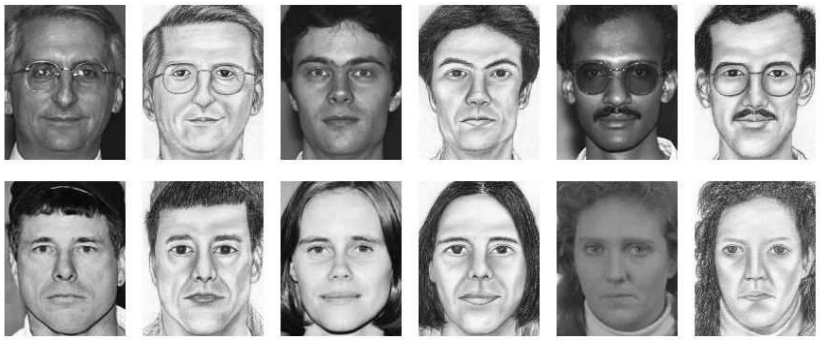


FIGURE *1.3: Exemple de paires d'image/esquisse de la base CUFSF*

Cette base est conçue pour la recherche sur les algorithmes de synthèse d'esquisses de visages et est, entre autre, utilisée pour tester les algorithmes permettant de retrouver la photographie d'une personne d'après son portrait-robot dans le cadre d'enquêtes de police.

Certaines images de la base de données FERET utilisée pour CUFSF sont absentes de la version de la base de données actuellement disponible. Pour cette raison, seules 860 paires photo/esquisse sont disponibles à l'heure actuelle.

1.1.3.4 Viewpoint Invariant Person Re-identification (VIPeR)

Contrairement aux bases de données que nous avons présentées jusqu'ici, la base VIPeR, est conçue pour la recherche sur la ré-identification de personnes. À notre connaissance, il s'agit de la plus grande base public disponible pour la ré-identification de personnes. Elle contient les représentations de 632 personnes capturées en extérieur avec deux images pour chaque personne. La principale source de variation d'apparence est due au changement de point de vue entre les deux caméras utilisées. La plupart des paires d'images contiennent une vue de face ou de dos et une vue de profil. Quelques exemples de paires issues de la base sont présentées à la figure 1.4.

1.2 Reconnaissance de visage et noyaux

Nous décrivons dans cette section le contexte scientifique de la thèse, qui est celui de l'utilisation de noyaux pour la reconnaissance de visages. Nous allons dans un premier temps établir quels sont les principes généraux de la reconnaissance de visages et montrer dans quelle mesure les noyaux jouent un rôle particulier. Nous donnerons ensuite les principes généraux des méthodes à noyaux et terminerons cette section en montrant comment nos contributions se placent dans ce contexte.

1.2.1 La reconnaissance de visages : principes et résultats récents

D'une manière générale, les algorithmes de reconnaissance de visages commencent par représenter les pixels du visage en un descripteur (appelé également signature ou vecteur de forme) qui va être utilisé dans un second temps par une mesure de similarité ou un classifieur.

FIGURE *1.4: Exemples de personnes capturées selon deux points de vue dans la base VIPeR.*

1.2.1.1 Les principaux descripteurs visuels utilisés pour représenter les visages

Descriptions globales des visages Les premiers travaux en reconnaissance de visages [114, 12] étaient pour la plupart basés sur des caractéristiques globales extraites de manière implicite par des méthodes de décomposition en sous-espaces. Par exemple, les visages propres (*Eigen Faces*) et les visages de Fisher (*Fisher Faces*) projettent l'ensemble du visage dans un sous-espace linéaire permettant de capturer les variations du visage.

Cependant, les méthodes actuelles les plus performantes, reposent majoritairement sur l'utilisation d'informations locales pour décrire les caractéristiques du visage. Des statistiques sur ces informations locales sont ensuite collectées sur des régions (voire la totalité) de l'image.

L'échantillonnage local. L'échantillonnage local consiste à utiliser comme descripteurs les valeurs de pixels, ou des interpolations entre des pixels, situés dans le voisinage d'un point considéré. La version la plus simple consiste à prendre des patchs [4] centrés sur le point considéré. Dans [24], Cao et al. échantillonnent des valeurs régulièrement réparties sur des cercles centrés sur le pixel de référence. Le descripteur résultant est simplement la concaténation des valeurs en niveau de gris des échantillons. Ces descripteurs locaux, calculés en chaque pixel, sont ensuite quantifiés à l'aide d'une méthode de partitionnement par arbre (mais il serait possible d'utiliser une méthode de k-moyennes). Puis, des histogrammes de codes (chaque code correspondant à une feuille de la quantification) sont calculés sur différentes régions de l'image.

Les filtres de Gabor. Les filtres de Gabor [46] représentent une famille d'ondelettes souvent utilisée en vision par ordinateur [4, 124] et inspiré des systèmes visuels biologiques. Ce sont des filtres linéaires constitués d'une gaussienne modulée par une sinusoïde. L'enveloppe gaussienne leur donne des propriétés d'analyse locale, tandis que la sinusoïde permet une analyse fréquentielle. Dans [94, 35], les auteurs utilisent une batterie de filtres de Gabor, localisés spatialement, dans un système multi-couche.

Les motifs locaux. Dans [91], Ojala *et al.* introduisent le concept de motifs binaires locaux (**LBP** pour *Local Binary Patterns*). Le principe est le suivant : dans une image (ou une région d'image) en

4. nous appelons patchs des petits morceaux d'images

niveau de gris, les valeurs v_i (éventuellement interpolées) sont échantillonnées en P points régulièrement espacés sur un cercle de rayon R centré sur le pixel de référence. Chacune de ces P valeurs est comparée à la valeur du pixel central v_c produisant ainsi une représentation binaire.

$$s_i = \begin{cases} 0 & \text{si } v_i - v_c \leq 0 \\ 1 & \text{si } v_i - v_c > 0 \end{cases}$$

L'ensemble des P valeurs binaires s_i mises les unes à la suite des autres $[s_1, \ldots, s_P]$ constituent un motif binaire. Des histogrammes d'apparition de ces motifs sur l'ensemble de l'image ou une région de l'image peuvent ensuite être calculés. [91] présente également certaines variantes des LBP invariantes à la rotation ou spécialisées sur les contours, facilement calculées à partir des motifs ordinaires à l'aide de simples tables d'association.

Les motifs ternaires locaux [109] (**LTP** pour *Local Ternary Patterns*) sont une extension des LBP introduite pour leur plus grande résistance au bruit. Au lieu de binariser les valeurs échantillonnées, celles-ci peuvent prendre trois valeurs selon leur distance à la valeur du pixel central :

$$t_i = \begin{cases} -1 & \text{si } v_i - v_c < -\tau \\ 0 & \text{si } |v_i - vc| \leq \tau \\ 1 & \text{si } v_i - v_c > \tau \end{cases}$$

où τ est un paramètre de seuil.

Récemment Hussain *et al.* [116] ont introduit les motifs locaux quantifié (**LQP** pour *Local Quantized Patterns*). Ceux-ci permettent d'utiliser les LBP ou des LTP sur des voisinages plus complexes c'est-à-dire des voisinages constitués de plusieurs cercles concentriques et avec un plus grand nombre d'échantillons. Pour éviter l'explosion combinatoire du nombre de codes, une étape supplémentaire de quantification vectorielle est ajoutée.

Dans [6, 109, 130, 115] des histogrammes de motifs locaux (LBP, LTP et LQP) sont utilisés pour la reconnaissance de visage. Dans [88, 123, 138], des histogrammes de LBP sont extraits sur des images d'orientation obtenues par l'application de différents types de filtres d'orientation tels que les filtres de Gabor.

Les histogrammes d'orientation de gradient. Pour calculer ce type de descripteurs, une carte de gradient est d'abord calculée. Pour chaque pixel, la direction du gradient est quantifiée en un certain nombre de canaux (*bins*) d'orientation. Des histogrammes sont calculés sur des régions de l'image où chaque pixel contribue au canal d'orientation correspondant proportionnellement à la magnitude du gradient. Dans le descripteur **SIFT** [83] (*Scale Invariant Feature Transform*), les histogrammes sont calculés dans les cellules d'une grille centrée sur un point d'intérêt après normalisation de l'image en fonction de l'échelle de détection et de l'orientation principale du point d'intérêt. Les SIFT denses calculent le descripteur SIFT sur les nœuds d'une grille fine, avec une échelle et une orientation fixes. Dans les descripteurs **HOG** [38] (*Histograms of Oriented Gradients*), les histogrammes sont calculés pour un ensemble de cellules couvrant l'image (potentiellement avec recouvrement).

Dans [58], Guillaumin *et al.* utilisent des descripteurs SIFT à plusieurs échelles calculés sur des points d'intérêts détectés automatiquement.

Pour obtenir de meilleures performances, les méthodes récentes [24, 75, 35, 109, 130, 136] utilisent des combinaisons de plusieurs descripteurs locaux. Les méthodes utilisées pour les combinaisons vont de la simple moyenne à l'utilisation de méthodes d'apprentissage de noyaux multiples (MKL for *Multiple Kernel Learning*).

1.2.1.2 Mesure de similarité entre descripteurs

Dans certaines situations, une simple analyse en composante principale blanchie (*whiten PCA*) peut suffire à obtenir de très bons résultats [24, 115].

Mais en général ce n'est pas suffisant. Aussi un nombre important de techniques ont été développées pour apprendre des mesures de similarité ou des mesures de distance spécifiques à nue tâche. Dans [75], Kumar apprennent des représentations de plus haut niveau en entraînant des classifieurs à discriminer la présence d'attributs relatifs à la couleur de peau, l'âge, le sexe etc. Ils utilisent également des parties de visages de référence pour déterminer des mesures de similarité. La représentation de haut niveau est donc constituée de la réponse des classifieurs d'attributs et des similarités des parties du visage par rapport aux visages de référence (le nez de Jennifer Anniston, la bouche de Barack Obama, etc.). La caractéristique de cette méthode tient au fait que la mesure de similarité est construite de manière indirecte, sans essayer de l'optimiser directement à partir des données d'entraînement.

Pourtant l'approche consistant à optimiser directement une mesure de similarité à partir des descripteurs bruts semble naturelle puisqu'il s'agit de déterminer si des visages se ressemblent ou non. C'est pourquoi de nombreuses méthodes récentes utilisent des techniques d'apprentissage de distance ou de métrique [108, 58, 87, 65, 139, 136].

Quoique nous ayons été amenés à développer notre propre descripteur local (le descripteur HOLD décrit au chapitre 2), les travaux présentés dans cette thèse suivent principalement cette ligne de recherche [85, 86]. Par ailleurs, les méthodes proposées reposent majoritairement sur l'utilisation des noyaux comme mesure de similarité brute.

1.2.2 Les méthodes à noyaux et le « truc du noyau »

Les mesures de similarité ou leur contrepartie, les mesures de distances jouent un rôle essentiel dans les algorithmes d'analyse de données ou d'apprentissage automatique en général, et de reconnaissance de visage en particulier. En effet, pour extraire des structures à partir des données (agrégats, variétés), il faut pouvoir comparer les objets étudiés.

Dans de nombreux cas pratiques, les données sont représentées sous la forme de vecteurs de nombres, ou peuvent être ramenées sous cette forme moyennant certaines transformations. Un point de donnée, pourra donc être représenté sous la forme d'un vecteur $\boldsymbol{x} \in \mathbb{R}^d$ à d dimensions.

Pour pouvoir comparer les objets, il est utile d'avoir une mesure de la similarité entre les objets et/ou une mesure de distance. Intuitivement, la distance entre des objets similaires doit être petite alors que la distance entre des objets très différents doit être grande.

1.2.2.1 Distance et produit scalaire

La distance euclidienne est un moyen intuitif et habituel pour mesurer la distance entre deux vecteurs de nombres. Soient $\boldsymbol{x}$ et $\boldsymbol{y}$ deux vecteurs de $\mathbb{R}^d$, le carré de la distance euclidienne est simplement :

$$D^2_{\text{eucl}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} (x_i - y_i)^2 \tag{1.1}$$

u_i désignant la i-ème composante du vecteur $\boldsymbol{u}$. On le voit, il s'agit simplement de la somme des différences au carré.

La distance euclidienne est intimement liée à la notion de produit scalaire. Le produit scalaire entre $\boldsymbol{x}$ et $\boldsymbol{y}$, noté $< \boldsymbol{x}, \boldsymbol{y} >$ (ou parfois $\boldsymbol{x}.\boldsymbol{y}$ ou encore $\boldsymbol{x}^T\boldsymbol{y}$ en considérant les vecteurs comme des matrices à une colonne) est défini par :

$$< \boldsymbol{x}, \boldsymbol{y} >= \sum_{i=1}^{d} x_i y_i \tag{1.2}$$

et il est facile de montrer que la distance euclidienne peut se réécrire :

$$D^2_{\text{eucl}}(\boldsymbol{x},\boldsymbol{y}) =< \boldsymbol{x},\boldsymbol{x} > + < \boldsymbol{y},\boldsymbol{y} > -2 < \boldsymbol{x},\boldsymbol{y} > \tag{1.3}$$

La norme (euclidienne) d'un vecteur $\|\boldsymbol{x}\|$ est définie par :

$$\|\boldsymbol{x}\|^2 =< \boldsymbol{x},\boldsymbol{x} > \tag{1.4}$$

Par ailleurs le cosinus de l'angle formé par deux vecteurs est donné par :

$$\cos(\boldsymbol{x},\boldsymbol{y}) = \frac{< \boldsymbol{x},\boldsymbol{y} >}{\sqrt{< \boldsymbol{x},\boldsymbol{x} >< \boldsymbol{y},\boldsymbol{y} >}} \tag{1.5}$$

Le cosinus représente une bonne mesure de la similarité entre deux vecteurs, en effet lorsque les vecteurs sont identiques la similarité vaut 1, lorsqu'ils sont orthogonaux elle vaut 0 et lorsqu'ils sont opposés elle vaut -1.

Remarquons que le cosinus de l'angle formé par deux vecteurs de norme unité est simplement le produit scalaire de ces deux vecteurs.

De nombreux algorithmes d'analyse de données ou d'apprentissage sont basés sur ces notions : les k-moyennes, les machines à vecteurs supports, la régression linéaire, l'analyse en composantes principales sont des exemples de méthodes reposant sur la notion de distance ou de produit scalaire.

1.2.2.2 Fonctions noyaux

Les fonctions noyaux définies positives peuvent être vues comme une généralisation de la notion de produit scalaire dans un espace euclidien. En voici la définition :

Définition 1.1. *Une fonction k à valeur dans $\mathbb{R}$ définie sur $\mathcal{X} \times \mathcal{X}$ est définie positive (DP) si et seulement si k est symétrique et si*

$$\sum_{i=1}^{n}\sum_{j=1} c_i c_j k(\boldsymbol{x}_i,\boldsymbol{x}_j) \geq 0$$

quels que soient $n \in \mathbb{N}$, $\boldsymbol{x}_i \in \mathcal{X}$, $i = 1,\ldots,n$ et pour tout $c_i \in \mathbb{R}$, $i = 1,\ldots,n$.

Cette propriété est notamment nécessaire pour s'assurer que la distance associée au noyau et donnée par :

$$D^2_k(\boldsymbol{x},\boldsymbol{y}) = k(\boldsymbol{x},\boldsymbol{x}) + k(\boldsymbol{y},\boldsymbol{y}) - 2k(\boldsymbol{x},\boldsymbol{y}) \tag{1.6}$$

soit toujours positive.

Il est facile de voir que le produit scalaire vérifie cette propriété :

Démonstration.

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j < \boldsymbol{x}_i,\boldsymbol{x}_j > &= \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \sum_{k=1}^{d} x_{ik}x_{jk} \\
&= \sum_{k=1}^{d}(\sum_{i=1}^{n} x_{ik})(\sum_{j=1}^{n} x_{jk}) \\
&= \sum_{k=1}^{d}(\sum_{i=1}^{n} x_{ik})^2 \geq 0
\end{aligned}$$

Le produit scalaire dans $\mathbb{R}^d$ est donc défini positif. □

1.2.2.3 Le «truc du noyau»

Le «truc du noyau» consiste à remplacer le produit scalaire, par une fonction noyau mieux adaptée à la comparaison des $\boldsymbol{x}$ et $\boldsymbol{y}$.

Par exemple, la distance euclidienne est mal adaptée à la comparaison d'histogrammes, et il lui est souvent préféré la distance du χ^2 :

$$D^2_{\chi^2}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{d} \frac{(x_i - y_i)^2}{x_i + y_i} \tag{1.7}$$

Cette distance correspond au noyau $k_{\chi^2} : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}$:

$$k_{\chi^2}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{d} 2\frac{x_i y_i}{x_i + y_i} \tag{1.8}$$

On peut montrer que le noyau χ^2 est défini positif :

Démonstration.

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k_{\chi^2}(\boldsymbol{x}_i,\boldsymbol{x}_j) &= \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \sum_{k=1}^{d} 2\frac{x_{ik}x_{jk}}{x_{ik}+x_{jk}} \\
&= \sum_{k=1}^{d}\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j x_{ik} x_{jk} \int_{t=0}^{1} t^{x_{ik}+x_{jk}-1} dt \\
&= \sum_{k=1}^{d} \int_{t=0}^{1} (\sum_{i=1}^{n} x_{ik} t^{x_{ik}-1/2})(\sum_{j=1}^{n} x_{jk} t^{x_{jk}-1/2}) dt \\
&= \sum_{k=1}^{d} \int_{t=0}^{1} (\sum_{i=1}^{n} x_{ik} t^{x_{ik}-1/2})^2 dt \geq 0
\end{aligned}$$

Le noyau k_{χ^2} est donc défini positif. □

Le truc du noyau peut donc être appliqué à tout algorithme où les données n'apparaissent que par l'intermédiaire de produit scalaire et/ou de distance.

En fait, pour tout noyau défini positif $k : \mathcal{X} \times \mathcal{X} \to \mathcal{X}$, il existe une fonction dite, fonction de re-description $\phi : \mathcal{X} \to \mathcal{H}_k$ qui projette les points de $\mathcal{X}$ dans un espace de Hilbert $\mathcal{H}$ tel que :

$$< \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) >_{\mathcal{H}} = k(\boldsymbol{x},\boldsymbol{y}) \tag{1.9}$$

où $< ., . >_{\mathcal{H}_k}$ est le produit scalaire dans $\mathcal{H}$.

Le «truc du noyau» peut donc être vu comme une projection dans un espace de dimension potentiellement infinie et dans lequel on applique l'algorithme considéré. Appliqué à un algorithme de séparation linéaire telle que les machines à vecteurs supports (SVM), cela peut permettre de trouver un hyperplan de séparation dans $\mathcal{H}$ alors que les données ne sont pas linéairement séparables dans $\mathcal{X}$. Autrement dit, cela permet d'appliquer des méthodes linéaires avec des mesures de similarité non-linéaires. L'intérêt du truc du noyau, c'est qu'il n'est pas nécessaire de connaître ϕ explicitement, seul le produit scalaire (et donc k) est nécessaire.

1.2.2.4 Le théorème du représentant

Dans la plupart des problèmes que nous avons à traiter nous disposons d'un ensemble de données d'entraînement $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathcal{X} \times \mathbb{R}$. Les $\boldsymbol{x}_i$ représentent des descripteurs pour les objets étudiés et les y_i des valeurs associées à ces points de données et que nous cherchons à prévoir.

Supposons que nous avons un noyau défini positif $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ et la fonction de re-description associée $\phi : \mathcal{X} \to \mathcal{H}$ telle que $< \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) >_{\mathcal{H}} = k(\boldsymbol{x}, \boldsymbol{y})$.

Considérons maintenant le problème d'optimisation suivant :

$$w^* = \arg\min_{w \in \mathcal{H}} E((\boldsymbol{x}_1, y_1, < w, \phi(\boldsymbol{x}_1) >_{\mathcal{H}}), \ldots, (\boldsymbol{x}_n, y_n, < w, \phi(\boldsymbol{x}_n) >_{\mathcal{H}})) + g(\|w\|) \tag{1.10}$$

où w est un vecteur de projection dans $\mathcal{H}$ que l'on cherche de sorte à minimiser une fonction de coût arbitraire E additionnée à une fonction de régularisation g strictement croissante et à valeur dans $\mathbb{R}$.

Le théorème du représentant nous dit que $\boldsymbol{w}^*$ peut s'écrire sous la forme : $w^* = \sum_{i=1}^{n} \alpha_i \phi(\boldsymbol{x}_i)$

La démonstration est relativement simple :

Démonstration. Décomposons w^* en une partie résidant dans le même sous-espace que l'ensemble des $\boldsymbol{x}_i$ et une partie orthogonale v :

$$w = \sum_i \beta_i \phi(\boldsymbol{x}_i) + v, \quad < v, \phi(\boldsymbol{x}_j) >= 0 \quad \forall j = 1, \ldots, n$$

Le produit scalaire de w avec un point d'entraînement quelconque $\boldsymbol{x}_j$ est :

$$< w, \phi(\boldsymbol{x}_j) >=< \sum_i \beta_i \phi(\boldsymbol{x}_i) + v, \phi(\boldsymbol{x}_j) >= \sum_i < \beta_i \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) >$$

Le premier terme de la formule (1.10) est donc indépendant de v. On a par ailleurs :

$$\begin{aligned} g(\|w\|) &= g\left(\| \sum_i \beta_i \phi(\boldsymbol{x}_i) + v \| \right) \\ &= g\left(\sqrt{\| \sum_i \beta_i \phi(\boldsymbol{x}_i) \|^2 + \|v\|^2} \right) \\ &\geq g\left(\| \sum_i \beta_i \phi(\boldsymbol{x}_i) \| \right) \end{aligned}$$

où l'égalité ne tient que pour $v = 0$. Imposer $v = 0$ n'a aucune répercussion sur le premier terme de (1.10) et décroît le deuxième terme. Pour un minimiseur w^* de (1.10) on doit donc avoir $v = 0$. □

Le théorème se généralise facilement au cas où on cherche plusieurs vecteurs w_i :

$$\begin{aligned} w^* = \arg\min_{w \in \mathcal{H}} E((\boldsymbol{x}_1, y_1, < w_1, \phi(\boldsymbol{x}_1) >_{\mathcal{H}}, \ldots, < w_m, \phi(\boldsymbol{x}_1) >_{\mathcal{H}}), \ldots, \\ (\boldsymbol{x}_n, y_n, < w_1, \phi(\boldsymbol{x}_n) >_{\mathcal{H}}, \ldots, < w_m, \phi(\boldsymbol{x}_n) >_{\mathcal{H}})) + \sum_{j=1}^{m} g(\|w_j\|) \end{aligned} \tag{1.11}$$

ainsi qu'au cas d'une norme $\|.\|$ quelconque pourvu qu'elle vérifie :

$$\|w + v\| \geq \|w\|$$
$$\|w + v\| = \|w\| \Leftrightarrow v = 0$$

1.2.3 Contributions de la thèse

Comme nous l'avons vu, la reconnaissance de visage dans les images est un problème rendu complexe par la grande variabilité de l'apparence d'un même visage en fonction de la pose ou de l'éclairage. Pourtant il existe des régularités, des structures, qu'il est possible d'exploiter pour créer un système capable d'effectuer cette tâche.

Les méthodes basées sur des modèles linéaires ont l'énorme avantage d'être formellement simple, et permettent de manipuler les concepts sous une forme analytique. Toutefois, il est probable qu'un modèle linéaire ne soit pas en mesure de traiter la complexité de structures complexes, telles que nous rencontrons en reconnaissance des visages.

Les méthodes à noyaux représentent une manière commode et puissante de modéliser des structures non-linéaires tout en conservant la simplicité formelle des méthodes linéaires. Le théorème du représentant nous permet de convertir une large gamme d'algorithmes linéaires en méthodes à noyau, et une vaste littérature existe sur ce type de conversion, le livre de Bernhard Schölkopf et Alexander J. Smola «*Learning with Kernels*», est un bon point de départ le lecteur intéressé.

Les travaux présentés dans cette thèse utilisent ce type d'approches et les appliquent au problème de la reconnaissance de visages ou de personnes. Nous abordons trois sujets différents.

Le recalage. Le premier sujet concerne le **recalage de visages**. Le recalage peut être décrit comme la mise en correspondances de parties comparables du visage (les yeux avec les yeux, la bouche avec la bouche, etc), soit physiquement – on déforme les images de sorte à ce qu'un ensemble de points caractéristiques se retrouvent à une position standard – soit conceptuellement (si on peut dire) en décrivant l'apparence locale autour des points caractéristiques, là où ils se trouvent. L'étape de recalage est une étape essentielle de toute procédure de reconnaissance de visage. Dans le chapitre 2, nous présentons une méthode de détection de points caractéristiques. La méthode repose sur deux éléments, la première est un nouveau descripteur local que nous avons baptisé HOLD. Ce descripteur est utilisé pour apprendre un modèle de régression linéaire régularisé dans sa version noyau. Nous présentons également l'ébauche d'une méthode de recalage utilisant un maillage 3D pour tenir compte explicitement des aprioris sur la forme des visages.

Reconnaissance de personnes et apprentissage de distance. Le deuxième sujet que nous traitons est celui de la reconnaissance de visage/personnes vu sous l'angle de l'apprentissage de distance. L'idée que les différentes représentations d'un même visage devraient être plus proches (au sens d'une métrique à définir), que les représentations de visages différents est assez intuitive. Nous formalisons donc cette idée pour construire une méthode où une mesure paramétrique de distance est apprise de sorte à refléter des *aprioris* connus.

Dans le chapitre 3, nous combinons une méthode d'apprentissage de distance à une mesure de distance sur graphe. Dans ce travail préliminaire, nous avons repris et amélioré une méthode d'apprentissage de distance linéaire. Nous introduisons une mesure de distance sur le graphe de voisinage afin de modéliser les relations de proximité de manière non linéaire. La métrique sur le graphe de voisinage correspond à un noyau. Dans le chapitre 4, nous proposons une nouvelle méthode d'apprentissage de distance inspirée de la première. Toutefois, dans notre nouvelle méthode, le truc du noyau est appliqué directement à la méthode d'apprentissage de distance.

Le chapitre 5 montre comment la méthode précédente peut être étendue au cas où les représentations des visages à comparer ne sont pas directement comparables. Cette situation apparaît par exemple lorsque nous désirons comparer un visage représenté dans le spectre visible (une photographie) basse résolution avec un visage représenté dans le spectre infra-rouge en haute définition. Les représentations ne sont pas directement comparables. Il est éventuellement possible d'appliquer une série de filtres pour réduire la différence de représentation mais ceci ne peut se faire sans perte d'information. La méthode que nous proposons, au contraire, utilise les représentations directement

et les projette indépendamment dans un nouvel espace où il est possible de les comparer. Nous appelons la tâche qui consiste à comparer des paires d'objets représentés dans différentes modalités **l'appariement trans-modale**.

Enfin, le chapitre 6 présente des travaux plus théoriques sur la famille des noyaux additifs homogènes. Nous y présentons notamment – pour la première fois à notre connaissance– une forme analytique de la fonction de re-description associée au **noyau de la moyenne puissance**. Bien que nous n'ayons pas eu le temps de mener ces travaux à leur terme, nous pensons que leur intérêt théorique et pratique mérite qu'ils soient présentés ici.

CHAPITRE 2

Le recalage de visages

LE recalage (ou alignement) des visages est une étape cruciale des procédures de reconnaissance de visage. Au sens large, elle consiste à mettre en correspondance les parties homologues du visage (les yeux avec les yeux, la bouche avec la bouche, etc.). La prédiction de la position des points d'intérêt d'un visage à l'intérieur de la boîte englobante – par exemple retournée par un détecteur de visage tel que le détecteur de Viola-Jones [122] – n'est pas très précise et vient pénaliser les algorithmes de reconnaissance qui requièrent souvent un bon recalage des visages à comparer.

Dans ce chapitre, nous présentons deux méthodes que nous avons conçues pour effectuer le recalage. La première s'attache à localiser des points d'intérêts (ou points-clés) dans le visage, alors que la deuxième tente de déterminer la configuration 3D de la tête.

Sommaire

2.1 Recalage et reconnaissance de visages

Le recalage est une étape essentielle dans la reconnaissance de visages et représente souvent un facteur limitant pour les systèmes de reconnaissance.

Pour illustrer ce propos, nous avons mené une expérience permettant d'étudier l'influence de la précision de la détection des points-clés sur la reconnaissance de visages.

Expérience. L'expérience a été menée sur le jeu de test de la vue 1 de la base LFW (voir section 1.1.3.2, page 5) avec les annotations fournies par Dantone *et al.* [39].

Pour cette expérience nous utilisons 4 points-clés du visage qui sont utilisés pour ramener les images de visage dans une position de référence, par application d'une homographie. Les 4 points-clés sont : (1) la moyenne des positions des coins de chaque œil, (2) la moyenne des positions des bords du nez, (3) la moyenne des bords supérieur des lèvres et (4) moyenne des bords inférieur des lèvres.

Une fois les visages ramenés à une position normalisée, ils sont représentés par une signature calculée par une méthode simple mais efficace, inspirée de celle de Hussain *et al.* [116] : des descripteurs LTP sont calculés sur des cellules de 10×10 pixels dans un fenêtre de 80×100 pixels centrée sur le visage. Ces descripteurs sont projetés dans un espace de dimension réduite (150) grâce à une analyse en composantes principales à noyau, avec un noyau χ^2. Nous donnons à chaque composante de l'espace réduit la même importance en la divisant par la racine carrée de la valeur propre associée. Une fois projetés dans l'espace réduit, les descripteurs sont normalisés (norme L_2).

Nous utilisons pour cette expérience une règle de classification très simple basée sur la distance euclidienne entre paires de descripteurs, la valeur médiane des distances entre paires étant choisie comme seuil de classification. Les paires associées à des distances plus petites que ce seuil sont considérées comme positives (même personne) et les autres comme négatives. Cette mesure correspond au taux de bonne classification à taux d'erreur égaux (*Accuracy at equal error rate*).

Pour étudier l'influence de la précision du recalage, nous bruitons les positions des points de référence, en les déplaçant dans une direction aléatoire d'une distance donnée. Cette distance est exprimée comme une fraction r de la distance inter-occulaire. Cette étape est supposée simuler les imprécisions dans la détection des points-clés.

Résultats. La figure 2.1 montre le score de classification obtenu pour différentes valeurs de r (points bleus) ainsi que le résultat de la régression linéaire appliquée à ces résultats (ligne verte).

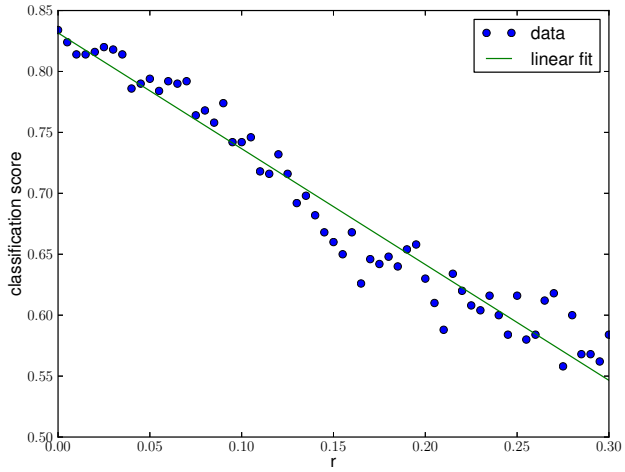


FIGURE *2.1: Évolution du score de classification des paires de visages sur le jeu de test de la vue 1 de la base LFW, en fonction du niveau de bruit appliqué sur la position des points de référence (bruit exprimé en fraction de la distance inter-occulaire).*

Le résultats bruts ne suivent pas une tendance strictement monotone, mais la tendance générale est une forte baisse des taux de reconnaissance en fonction des imprécisions de localisation des points-clés du visage. Nous partons d'un score de environ 83% en utilisant les points non bruités pour un atteindre un score d'environ 55% lorsqu'un bruit de l'ordre de 30% de la distance inter-occulaire est appliqué, soit à peine mieux que le hasard. Avoir une bonne localisation des points-clés est donc une étape importante de la reconnaissance.

2.2 Localisation de points-clés par régression

Le recalage des visages est un problème très étudié [7, 13, 99, 100, 117] car il représente une étape préliminaire nécessaire à la reconnaissance de visage.

Dans cette section nous présentons une technique pour détecter des points caractéristiques des visages basée sur la régression. Le principe est de prédire la position de points-clés du visage en observant l'apparence locale du visage autour de la position moyenne de ces points-clés.Le modèle de régression vient établir un lien entre l'apparence locale et la position des points-clés. Conceptuellement simple, cette approche donne de très bons résultats de localisation, en comparaison aux méthodes de l'état de l'art. La simplicité de cette approche tient principalement au fait qu'elle utilise des modèles d'optimisation bien connus mais utilisés avec des matrices noyaux. Elle offre une alternative efficace à la régression par vecteurs supports [118] utilisée habituellement.

2.2.1 Travaux antérieurs

Les méthodes existantes peuvent être classées en deux catégories selon qu'elles utilisent des caractéristiques globales ou locales.

Les méthodes globales telles que les modèles actifs d'apparence (*Active Appearence Models*) [34, 8], utilisent l'information de texture de toute la zone du visage pour ajuster un modèle linéaire sur les images de test. Ce type d'algorithmes est malheureusement sensible aux changements de conditions d'éclairage et ont tendance à donner des résultats biaisés en direction du visage moyen. De plus, ces méthodes ne fonctionnent pas très bien sur des visages inconnus et ne donnent pas de bons résultats sur les images de basses résolutions [56].

Récemment, des méthodes basée sur des caractéristiques locales se sont progressivement imposées. Des méthodes telles que celle de [124] entraînent des détecteurs indépendants pour 20 points-clés, en se basant sur la réponse de filtres de Gabor. En cas de mauvaise détection ce genre d'approche a tendance à fournir des configurations non cohérentes, en raison de l'absence d'information sur la structure globale du visage. Pour pallier partiellement ces problèmes, les auteurs de [124] restreignent la zone de recherche, ce qui a pour conséquence directe de rendre impossible la détection de configurations s'éloignant trop de celles présentes dans la base d'entraînement.

Plusieurs méthodes issues des modèles actifs de formes de [33], ont été développées pour la détection de points-clés. Les modèles locaux contraints de [36] utilisent l'analyse en composantes principales (ACP) pour modéliser l'apparence des points-clés alors que la méthode *Boosted Regression Active Shape Models* [37] tente de prédire une nouvelle position pour le point-clé à partir de l'apparence observée à la position courante.

Parmi les méthodes qui tentent d'intégrer de manière plus robuste un *apriori* sur la configuration globale, nous notons celle de Everingham *et al*. [45] qui utilise les structures picturales de [47]. Une extension *hiérarchique de cette méthode est utilisée* dans [100].

Valstar *et al*. [117] combinent une régression par vecteurs supports – permettant d'estimer la localisation des points-clés – avec un champ aléatoire de Markov permettant de garder la configuration des points-clés globalement cohérente. Cependant, la méthode est particulièrement coûteuse en temps de calcul. Récemment Amberg et Vetter [7] ont proposé d'utiliser des détecteurs de points-clés (yeux, bouche, etc.) sur l'image complète puis d'utiliser ensuite l'algorithme *séparation et évaluation* (*branch and bound*) pour déterminer la configuration la plus probable des points-clé. Toutefois, leurs résultats ne sont donnés que pour des images de bonne qualité et les temps de calculs pour une image restent longs. Belhumeur *et al*. [13] ont récemment proposé un modèle bayesien combinant les résultats de détecteurs locaux avec un consensus de modèles globaux non paramétriques pour la localisation des parties du visage. Leur méthode est la plus précise de l'état de l'art sur la base *Labeled Face Parts in the Wild* (LFPW) [13], avec des résultats légèrement meilleurs que la méthode Valstar *et al*. [117]. Mais rappelons que [117] donnent des mesures de performance sur une base

qui toutefois contient des images de bien meilleure qualité que celles de *Labeled Faces in the Wild* (LFW). Dernièrement, Dantone *et al*. [39] ont évalué sur la base LFW une méthode de régression par forêts aléatoires conditionnelles qui donne de très bons résultats sur des images de faible qualité, avec des temps compatibles avec des applications en temps réel. Leur méthode extrait des patchs carrés de l'image de manière aléatoire. L'apparence du patch est décrite par les niveaux de gris bruts et normalisés correspondants, ainsi que la réponse d'une série de filtres de Gabor.

La méthode que nous proposons est également une méthode à base de régression. À partir de l'apparence du visage extraite aux positions moyenne des points-clés, nous tentons de prédire leur position réelle. Le modèle de régression utilisé est une adaptation originale du modèle LASSO [111] (c'est-à-dire une régression par moindres carrés avec régularisation sur la norme L_1) appliquée sur des matrices noyaux. Notre méthode représente une variante peu usitée de la régression par vecteurs supports décrite dans [101] et donne de meilleurs résultats que les autres méthodes testées sur la base LFW.

2.2.2 Modèle de régression

Notre modèle est une adaptation de la méthode proposée dans [101]. Nous traitons la régression comme un problème de moindres carrés régularisés avec la norme L_1, également connu sous le nom de Lasso [111]. Formellement, nous disposons d'un ensemble de couples $(\boldsymbol{x}_i, y_i)$, où $\boldsymbol{x}_i \in \mathbb{R}^d$ est un vecteur qui encode un descripteur visuel et où y_i est une valeur réelle que l'on souhaite pouvoir prédire à partir de $\boldsymbol{x}_i$. Si on appelle X la matrice dont la i-ième colonne est formée par le vecteur $\boldsymbol{x}_i$, et $\boldsymbol{y}$ le vecteur dont la i-ième composante est y_i, alors il s'agit de trouver le vecteur w qui minimise :

$$\min_{\boldsymbol{w}} ||X^T\boldsymbol{w} - \boldsymbol{y}||_2^2 + \lambda\|\boldsymbol{w}\|_1 \tag{2.1}$$

Le premier terme correspond à une régression linéaire au sens des moindres carrés. Le deuxième terme est un terme de régularisation qui impose que la norme L_1 du vecteur $\boldsymbol{w}$ ne soit pas trop grande. Ce type de régularisation a la propriété bien connue de conduire à des solutions parcimonieuses, c'est-à-dire des vecteurs $\boldsymbol{w}$ ayant un nombre important des coefficients nuls. Le paramètre $\lambda \geq 0$ permet de fixer l'importance de la régularisation.

En re-paramétrisant par $\boldsymbol{w} = X\boldsymbol{\alpha}$, le problème devient :

$$\min_{\boldsymbol{\alpha}} ||X^TX\boldsymbol{\alpha} - \boldsymbol{y}||_2^2 + \lambda\|X\boldsymbol{\alpha}\|_1$$

Nous remarquons que $K = X^TX$ est la matrice de Gram de nos données. Elle correspond à la matrice noyau issue de l'utilisation du noyau linéaire $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T\boldsymbol{x}'$. Le noyau linéaire peut donc facilement être remplacé par n'importe quel autre noyau défini positif. Le vecteur $K\boldsymbol{\alpha}$ peut donc être vu comme une combinaison linéaire des contributions associées à chaque échantillon de données.

En remplaçant le terme de régularisation $\|X\boldsymbol{\alpha}\|_1$ par $\|\boldsymbol{\alpha}\|_1$, nous forçons la solution à s'exprimer comme une combinaison linéaire d'un nombre limité d'échantillons.

Cette formulation peut être rapprochée de la régression par vecteur support (RVS), quoique la formulation soit assez différente.

En écrivant $K = X^TX$ la matrice de Gram correspondant au noyau linéaire nous obtenons :

$$\min_{\boldsymbol{\alpha}} ||K\boldsymbol{\alpha} - \boldsymbol{y}||_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{2.2}$$

Nous appelons ce modèle de régression LS-SVR (pour *Least Squares Support Vector Regression*). La matrice K peut être remplacée par n'importe quelle matrice noyau semi-définie positive, ce qui permet éventuellement d'utiliser des mesures de similarité plus adaptées que le produit scalaire.

Dans la formulation habituelle de SVR on cherche à résoudre le problème d'optimisation suivant :

$$\min_{\boldsymbol{w}} \frac{\lambda}{2} < \boldsymbol{w}, \boldsymbol{w} > + \sum_i \max(0, | < \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}_i) > - y_i| - \varepsilon) \tag{2.3}$$

Le théorème du représentant nous dit que la solution $\boldsymbol{w}$ peut s'exprimer comme une combinaison linéaire des $\boldsymbol{\phi}(\boldsymbol{x}_i)$, ce qui donne :

$$\min_{\boldsymbol{\alpha}} \sum_i \max(0, |(K\boldsymbol{\alpha})_i - y_i| - \varepsilon) + \frac{\lambda}{2} \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \tag{2.4}$$

SVR et LS-SVR sont donc toutes les deux des méthodes de régressions régularisés mais optimisant des fonctions différentes. La fonction d'attache aux données de SVR est souvent préférée pour sa robustesse aux points aberrants (*outliers*). Dans le problème qui nous intéresse, les visages proches de la pose frontale sont sur-représentés (voir la figure 2.4), il est donc important de donner un poids plus important aux cas les plus rares pour éviter d'avoir une trop forte attraction des résultats vers le cas moyen, ce que permet l'erreur quadratique de LS-SVR.

Le terme de régularisation de LS-SVR a pour effet d'annuler la contribution d'une grande partie des vecteurs d'entraînement. Combiné à la fonction de coût quadratique, notre hypothèse est que la méthode sera mieux à même de généraliser les prédictions pour des valeurs extrêmes.

2.2.3 Algorithme de localisation

Notre algorithme de localisation est construit autour de la méthode de régression que nous venons de voir. Comme nous traitons des images issues du résultat d'un détecteur de visage, nous avons de forts *apriori* concernant la position des points-clés. La figure 2.2 montre la distribution des yeux, du bout du nez et des coins de la bouche dans les images de la base ***Labeled Faces in the Wild*** (LFW). Nous observons que les yeux correspondent aux distributions les plus étroites. Ceci est une propriété connue du détecteur de Viola-Jones [122], qui se base en grande partie sur les yeux pour détecter les visages.

Nous utilisons cet ***apriori*** en calculant des descripteurs aux positions moyennes des points-clés. Notre modèle de régression est ensuite appliqué sur ces descripteurs, ou plutôt, au noyau obtenu à partir de ces descripteurs.

Nous pensons que les positions des points-clés sont corrélées. Par exemple, une rotation sur la droite de la tête déplace tous les points vers la gauche dans l'image. Aussi, lorsque nous cherchons à prédire la position d'un point-clé, nous utilisons l'ensemble des descripteurs de tous les points-clés. Pour cela, les noyaux calculés pour les descripteurs locaux sont additionnés. Pour des noyaux additifs cela revient à concaténer les descripteurs. Pour permettre une pondération de l'influence des descripteurs sur la prédiction d'un point-clé particulier, des poids sont affectés aux différents noyaux en fonction du point-clé considéré. Dans nos expériences la valeur de ces poids est déterminée par recherche sur grille, sur des données de validation.

Pour chaque point-clé i nous appliquons donc notre méthode de régression en utilisant le noyau $\hat{K}_i = \sum_j \alpha_{ij} K_j$, où K_j est le noyau calculé à partir du descripteur calculé au point-clé j, et α_{ij} est le coefficient de pondération. Ces coefficients sont normalisés de sorte que $\sum_j \alpha_{ij} = 1$.

Le modèle appris peut être ensuite utilisé sur de nouveaux visages pour lesquels nous souhaitons avoir une estimation de la position des points-clés. Nous pouvons alors éventuellement recalculer des descripteurs aux nouvelles positions et produire ainsi, de manière itérative, de nouvelles prédictions.

2.2.4 Le descripteur HOLD (histogrammes de différences locales orientées)

Dans nos expériences préliminaires, les descripteurs SIFT [83] ont donné des résultats satisfaisants. Cependant, le calcul de ces descripteurs est relativement long surtout si la procédure d'extraction/prédiction est appliquée plusieurs fois. Pour remédier à ce problème nous avons mis au point un

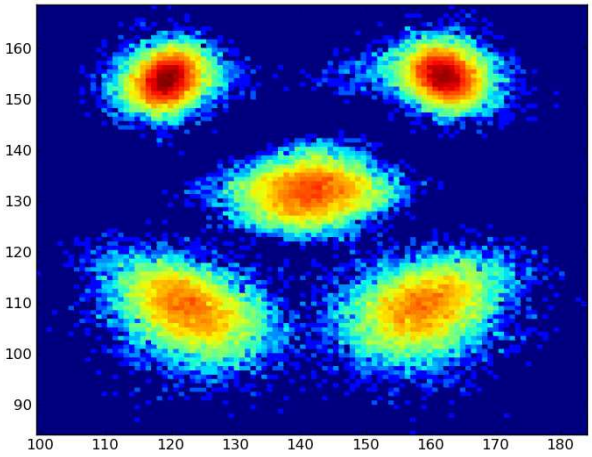


FIGURE *2.2: Distribution de la position des yeux, du bout du nez et des coins de la bouche dans LFW (détecteur de Viola-Jones). Les positions sont données en pixels*

nouveau descripteur inspiré à la fois de SIFT, des histogrammes d'orientation de gradient (HOG) et des motifs binaires locaux (LBP).

Le principe est le suivant. Pour chaque pixel, nous échantillonnons P valeurs d'intensité prises sur un cercle de l'image de rayon R, centré sur ce pixel (même procédé que pour les LBP). Comme pour les LBP, la valeur du pixel central et soustraite des P valeurs échantillonnées. Le résultat obtenu peut donc se présenter sous forme d'un vecteur $\boldsymbol{d}$ à P dimensions tel que :

$$d_i = v_i - v_c$$

où v_i est la i-ème valeur échantillonnée et v_c correspond à la valeur du pixel central. Nous appelons le vecteur $\boldsymbol{d}$ le *vecteur des différences locales*. Chaque composante de $\boldsymbol{d}$ peut être vue comme une estimation du gradient de l'image au pixel central dans une direction particulière. Là où avec SIFT ou HOG, nous aurions une seule valeur correspondant à l'orientation du gradient, nous avons P valeurs pour chaque pixel.

La carte qui à chaque pixel associe son vecteur de différences locales peut être vue comme une image à P canaux. Nous dédoublons ensuite cette carte afin de distinguer les différences à valeurs positives et négatives :

$$d_i^+ = \max(0, d_i), \quad d_i^- = \min(0, d_i)$$

Nous appelons M^+ et M^- les cartes correspondant respectivement aux différences locales positives et négatives pour chaque pixel. M_i^+ désignera le i-ème canal de M^+.

Nous calculons à présent, pour chaque pixel, son histogramme des différences locales en appliquant à chaque canal de M^+ et M^- un filtre moyenneur à deux dimensions. Cela permet de cumuler, pour chaque pixel et chaque canal, les contributions des pixels voisins. Soit k le noyau de convolution correspondant au filtre. Nous avons :

$$\hat{M}^+ = M^+ * k, \quad \hat{M}^- = M^- * k$$

où $*$ correspond au produit de convolution à deux dimensions appliqué canal par canal.

Pour un filtre moyenneur uniforme cela correspond à calculer l'histogramme des différences pour chaque orientation dans la région couverte par le noyau du filtre. En pratique nous utilisons plutôt un noyau Gaussien.

Cependant, les différences positives et négatives ayant été séparées, les cartes résultantes présentent de nombreuses valeurs nulles qui ne devraient pas être prises en compte dans le calcul des histogrammes. Pour remédier à cela, nous appliquons le filtre sur les cartes indicatrices (cartes binaires) correspondant à chaque canal. La carte indicatrice est la carte qui pour chaque pixel et chaque canal indique si la valeur correspondante est positive (I^+) ou négative (I^-). Si M est la carte des vecteurs $\boldsymbol{d}$, on a :

$$M^+ = M \otimes I^+, \quad M^- = M \otimes I^-$$

où $\otimes$ indique le produit élément par élément.

Nous calculons donc les constantes de normalisation par :

$$N^+ = I^+ * k, \quad N^- = I^- * k$$

, et les cartes H^+ et H^- des histogrammes des différences locales sont données par :

$$H^+ = \hat{M}^+ \oslash (N^+ + \varepsilon), \quad H^- = \hat{M}^- \oslash (N^- + \varepsilon)$$

où $\oslash$ indique la division élément par élément, et où ε est une petite constante positive servant à éviter la division par zéro.

A la manière de SIFT, le descripteur correspondant à un pixel est calculé comme la concaténation des histogrammes pris au centre des cellules d'une grille 4×4 centrée sur le pixel, et la contribution de chaque cellule est pondérée en utilisant un profil Gaussien de largeur w centré sur le pixel.

Le descripteur a donc 5 paramètres qui sont répertoriés dans le tableau 2.1. Le tableau indique les valeurs utilisées dans nos expériences (valeurs que nous avons fixées empiriquement, sans chercher à les optimiser particulièrement). Nous appelons ce descripteur HOLD pour *Histograms of Oriented Local Differences* soit en Français *histogrammes des différences locales orientées*.

paramètre	description	valeur
P	nombre d'échantillons	8
R	rayon du cercle d'échantillonage	2.0 pixels
σ	largeur du noyau Gaussien du filtre	4.0 pixels
s	taille des cellules de la grille 4×4	10.0 pixels
w	largeur de la Gaussienne de pondération	3.0 pixels

TABLE *2.1: Les parmètres du descripteur HOLD des différences locales*

2.2.5 Résultats expérimentaux

Base et protocole expérimental. Les expériences ont été réalisées sur la base Labeled Faces in the Wild (LFW) en respectant le protocole expérimental associé à cette base. Les paramètres libres du modèle (paramètres du descripteur, coefficients des noyaux et paramètres de régularisation du LS-SVR), ont été choisis pour donner les meilleurs résultats sur la vue 1.

Les performances ont ensuite été évaluées sur la vue 2, en utilisant deux critères :

1. Le taux de bonne localisation défini comme la proportion de points-clés détectés avec une précision supérieure à 10 % de la distance inter-occulaire.
2. L'erreur de localisation qui correspond simplement à la distance (en pixels) entre le point-clé détecté et la vérité terrain.

Pour cette base nous disposons des annotations fournie par M. Dantone *et al*. [39]. Ces annotations donnent la position de 10 points-clés : les coins internes et externes des yeux (4 points-clés), le bord extérieur des narines (2 points-clés), le milieu du bord supérieur de la lèvre supérieure et du bord inférieur de la lèvre inférieure (2 points-clés) ainsi que les coins de la bouches (2 points-clés). La figure 2.3 montre la position des dix points-clés sur un des visages de la base LFW.

FIGURE *2.3: Position des points-clé annotés pour la base LFW.*

Évaluation. Nos résultats sont comparés à ceux obtenus dans [39] pour la méthode CRF (*Conditional Random Forests*) et la méthode proposée par Everingham *et al*. [45].

Le tableau 2.2 donne les résultats obtenus avec notre méthode pour deux noyaux différents ainsi que les résultats de la méthode CRF de Dantone *et al*. En termes de taux de bonne détection, notre méthode avec le noyau de Bhattacharyya donne des résultats très proches quoique légèrement plus faibles que la méthode CRF. Par contre, avec le noyau χ^2, notre méthode surpasse en moyenne la méthode CRF. Les résultats détaillés montre que notre méthode est meilleure pour la plupart des points-clés à l'exception des coins du nez et du bord de la lèvre supérieure.

face feature	CRF	Ours-Bhat	Ours-χ^2
left eye left	87.7	89.2	**89.9**
left eye right	93.5	94.3	**94.4**
right eye right	92.9	93.4	**94.2**
right eye left	86.2	90.5	**90.8**
mouth left	81.9	81.2	**83.4**
mouth right	80.8	80.5	**84.6**
nose left	**90.4**	87.7	88.8
nose right	**88.2**	85.3	86.8
upper outer lip	**86.7**	83.2	84.8
lower outer lip	71.5	72.4	**74.8**
average	86.0	85.8	**87.3**

TABLE *2.2: Taux de bonne détection en pourcentage.*

Le tableau 2.3 montre les résultats obtenus en terme d'erreur de localisation pour CRF, la méthode d'Everingham et notre méthode avec deux noyaux différents.

La méthode d'Everingham donne des résultats systématiquement moins bons que les deux autres méthodes. En moyenne, c'est encore notre méthode avec le noyau χ^2 qui donne les meilleurs résultats. Notre méthode, avec le noyau de Bhattacharyya, est cette fois plus performante en moyenne que la méthode CRF. En détaillant les performances par point-clé, notre méthode est plus précise quel que soit le noyau, pour tous les points-clés à l'exception du coin gauche du nez et du bord extérieur de la lèvre supérieure.

face feature	CRF	Everingham	Ours-Bhat	Ours-χ^2
left eye left	0.0682	0.1621	0.0601	**0.0581**
left eye right	0.0565	0.1070	0.0507	**0.0494**
right eye right	0.0567	0.0937	0.0525	**0.0496**
right eye left	0.0736	0.1116	0.0602	**0.0580**
mouth left	0.0738	0.1076	0.0729	**0.0690**
mouth right	0.0780	0.1514	0.0743	**0.0684**
nose left	0.0592	0.1085	0.0637	0.0617
nose right	0.0705	0.1208	0.0674	**0.0652**
upper outer lip	0.0640	–	0.0706	0.0682
lower outer lip	0.0953	–	0.0871	**0.0839**
average	0.0696	(0.1222)	0.0660	**0.0631**

TABLE *2.3: Erreur moyenne de localisation.*

La figure 2.7 montre des exemples de résultats obtenus. La rangée (a) montre les dix meilleures détections en terme d'erreur moyenne et la rangée (b) les dix moins bonnes. Il n'est pas très surprenant de voir que les meilleures détections correspondent à des visages vus de face (les rotations dans le plan de l'image ne semblent pas trop influer sur les résultats) alors que les moins bonnes détections correspondent presque toutes à des vues de profil.

Afin d'étudier plus avant ce phénomène, nous définissons une mesure simple permettant d'évaluer le degré de rotation des visages selon l'axe horizontal. Pour cela nous déterminons le décalage horizontal entre le barycentre des coins du nez $\bar{\boldsymbol{x}}_{\text{nez}}$ et le barycentre des coins extérieurs des yeux $\bar{\boldsymbol{x}}_{\text{yeux}}$. Pour rester invariants aux rotations dans le plan de l'image, la direction horizontale est prise comme la direction $\boldsymbol{h}_{\text{yeux}}$ du vecteur joignant par les coins extérieurs des yeux. Le décalage horizontal du nez est donc défini comme :

$$d_{\text{nez}} = \boldsymbol{h}_{\text{yeux}}.(\bar{\boldsymbol{x}}_{\text{nez}} - \bar{\boldsymbol{x}}_{\text{yeux}})/\text{distance inter occulaire}$$

La figure 2.4 représente la distribution de cette grandeur pour LFW. Nous observons que les visages en position frontale sont sur-représentés par rapport aux visages de profil.

Il n'est donc pas étonnant que l'erreur de localisation soit plus grande pour les visages de profil. La figure 2.5 montre la moyenne et la médiane des erreurs correspondant à chaque canal (*bin*) de l'histogramme de la figure 2.4.

La figure 2.6 montre la superposition de distributions normalisées du décalage horizontal, avec en bleu la distribution pour l'ensemble des données (identique à la figure 2.4), et en vert l'histogramme pondéré correspondant aux vecteurs supports (données d'entraînement de contribution non nulle) du modèle LS-SVR pour la position du bord extérieur de l'œil gauche. Chaque vecteur support i contribue donc à l'histogramme avec un poids w_i donné par :

$$w_i = \frac{\alpha_{xi}^2 + \alpha_{yi}^2}{\sum_j \alpha_{xj}^2 + \alpha_{yj}^2}$$

où $\boldsymbol{\alpha}_x$ et $\boldsymbol{\alpha}_y$ correspondent aux coefficients du modèle LS-SVR pour les coordonnées respectives x et y du coin extérieur de l'œil gauche.

Clairement, la distribution est plus étalée, ce qui signifie qu'une importance plus grande est donnée aux exemples d'entraînement correspondant aux visages de profil pourtant moins représentés dans la base. Ceci confirme notre intuition sur les propriétés du modèle LS-SVR.

La figure 2.7 b) montre les 10 moins bonnes détections. Presque toutes correspondent à des poses proches du profil, ce qui montre qu'un effort reste à faire pour le traitement de ces cas. Toutefois, il

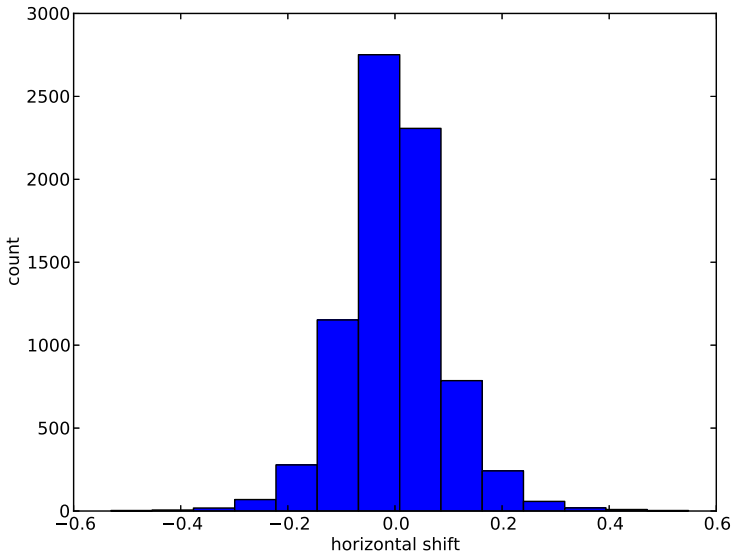


FIGURE *2.4: Distribution du décalage horizontal du nez* d_{nez}.

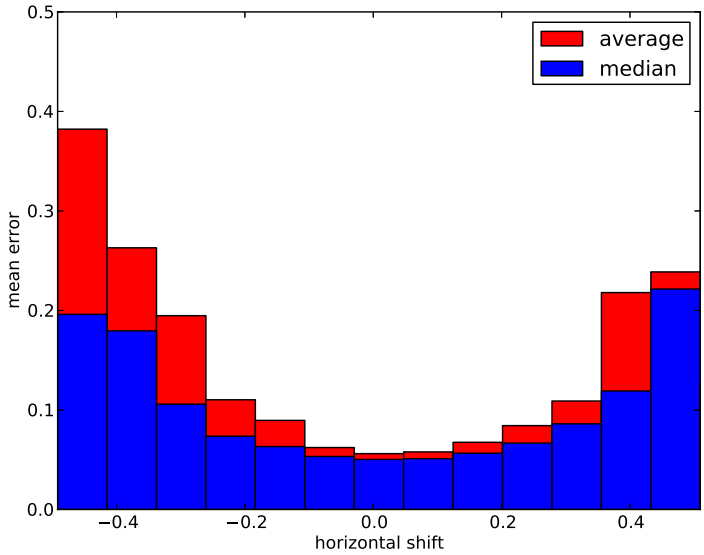


FIGURE *2.5: Moyenne et médiane des erreurs pour chaque canal de l'histogramme de la figure 2.4.*

est intéressant de remarquer que, même pour ces mauvaises détections, la configuration des points-clés reste cohérente. Ce phénomène est d'autant plus intéressant que notre modèle n'incorpore pas d'apriori sur la configuration globale de manière explicite, mais seulement de manière implicite par l'utilisation de tous les descripteurs locaux pour chaque point-clé.

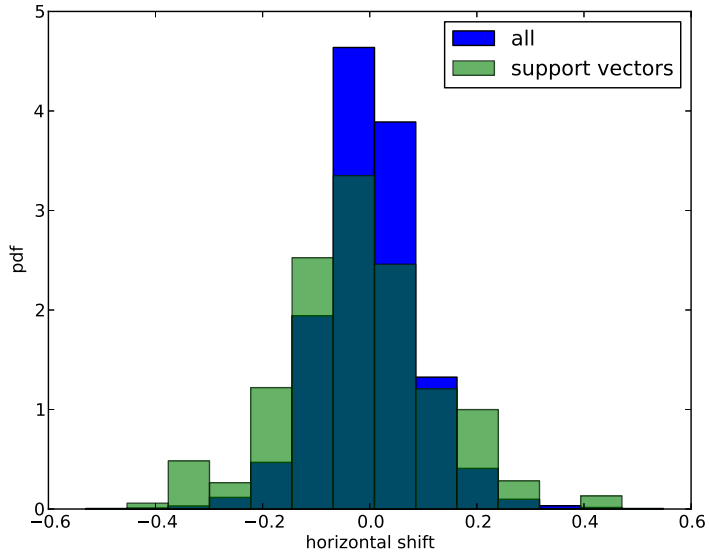


FIGURE *2.6: Distribution du décalage horizontal du nez* d_{nez}.

2.2.6 Améliorations possibles

Ces travaux restent encore à un stade préliminaire et plusieurs pistes sont envisageables pour en améliorer les performances aussi bien en termes de précision que de vitesse d'exécution (problème que nous n'avons pas abordé jusque-là).

Approche itérative. La première piste concerne l'amélioration de la précision de manière itérative. Une fois la première prédiction effectuée, le descripteur HOLD est calculé à l'emplacement de la prédiction et une nouvelle position est prédite. Le processus peut être répété plusieurs fois. Le descripteur HOLD est calculé en deux étapes. La première étape est appliquée à toute l'image et fournit une carte des histogrammes des différences locales en chaque pixel. La deuxième extrait le descripteur local en échantillonnant la carte dans le voisinage du point à décrire. L'étape la plus coûteuse en temps de calcul est le calcul de la carte globale mais celle-ci peut être calculée une fois pour toute pour l'image et seule la deuxième étape d'échantillonnage, bien moins coûteuse, doit être appliquée à chaque itération.

Approximation de noyaux. La deuxième piste concerne la vitesse d'exécution. Une fois le modèle appris, la prédiction des coordonnées $\boldsymbol{y}$ d'un descripteur $\boldsymbol{x}$ est effectuée par :

$$\boldsymbol{y} = \boldsymbol{\alpha}^T \boldsymbol{k}(\boldsymbol{x})$$

où $\boldsymbol{\alpha}$ correspond aux paramètres du modèle LS-SVR et où $\boldsymbol{k}(\boldsymbol{x}) = [k(\boldsymbol{x}, \hat{\boldsymbol{x}}_1)\ k(\boldsymbol{x}, \hat{\boldsymbol{x}}_2)\ \ldots\ k(\boldsymbol{x}, \hat{\boldsymbol{x}}_N)]^T$ est le vecteur dont les éléments correspondent à la valeur du noyau entre $\boldsymbol{x}$ et chacun des descripteurs $\hat{\boldsymbol{x}}_i$ utilisées pour l'apprentissage.

Il est possible d'accélérer le processus si le calcul de la fonction de re-description du noyau utilisé ou une approximation de cette dernière est facilement calculable. Dans ce cas si $\phi(x)$ est la fonction de re-description nous pouvons définir le vecteur $\boldsymbol{w} \in \mathcal{H}^d$ tel que : $w_i = \sum_j \alpha_{ji} \boldsymbol{\phi}(\hat{\boldsymbol{x}}_j)$ et la prédiction se calcule simplement par :

$$\boldsymbol{y} = < \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}) >$$

(a)

(b)

(c)

FIGURE *2.7: Exemples de résultats pour la localisation de points-clés (a) Résultats correspondant à l'erreur moyenne la plus faible. (b) Résultats correspondant à l'erreur moyenne la plus forte. (c) Résultats correspondant à l'erreur la plus faible avec une déviation* $|d_{nez}| > 0.3$ *(voir le texte).*

Dans le cas du noyau de Bhattacharyya, la fonction de re-description est facile à calculer :

$$\boldsymbol{\phi}(\boldsymbol{x}) = [\sqrt{x_1}\ \sqrt{x_2}\ \ldots\ \sqrt{x_D}]^T$$

Pour d'autres noyaux tels que le noyau χ^2 ou le noyau gaussien, il est possible de calculer explicitement une approximation de la fonction de re-description [63, 97]. La fonction de re-description résultante est une fonction $\boldsymbol{\phi} : \mathbb{R}^D \rightarrow \mathbb{R}^{mD}$ où m est un entier d'autant plus grand que l'approximation est précise. Ces méthodes d'approximation font notamment l'objet du chapitre 6.

Utilisation d'un modèle génératif. Certaines mauvaises localisations sont typiquement dues au fait que le visage est mal centré dans la fenêtre de détection par rapport à la majorité des visages. Le phénomène est évident sur la figure 2.8 où les visages sont particulièrement décentrés verticalement pour l'un et horizontalement pour l'autre. Dans l'image de gauche, les coins extérieurs des yeux ont été placés au niveau des coins de la bouche. Alors que dans l'image de droite, les coins d'un œil ont été placés sur les coins de l'autre œil.

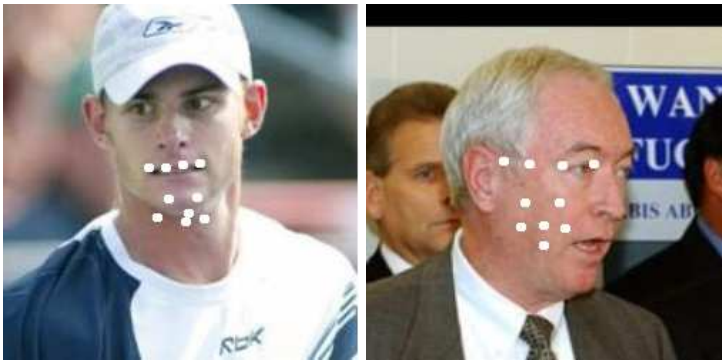


FIGURE *2.8: Mauvaises détections sur des visages décentrés verticalement (à gauche) et horizontalement (à droite)*

Une approche possible pour traiter ce cas consisterait à avoir un modèle de l'apparence pour chacun des points-clés. Dans les cas typiques tels que ceux présentés à la figure 2.8, l'adéquation des descripteurs correspondant aux points prédits avec le modèle d'apparence devrait être plutôt mauvaise. Lorsqu'une telle configuration est détectée, il pourrait être utile de relancer l'algorithme mais en explorant les positions des points moyens autour de la position d'origine. Le résultat final ayant la meilleure adéquation au modèle d'apparence serait alors retenu. Cela revient à détecter le visage avec un détecteur plus performant.

2.3 Alignement de visage 2D-3D

La plupart des méthodes de détection de points-clés sur des images de visages n'incorporent pas d'*apriori* concernant la structure tridimensionnelle propre aux visages humains. Les méthodes existantes sont typiquement basées sur des modèles 3D déformables tels que ceux proposés par Blantz *et al.* [17], où, après une initialisation manuelle grossière, un rendu du modèle 3D est superposé à l'image initiale et les paramètres du modèle sont optimisés de sorte à minimiser la différence entre le résultat du rendu et la photographie d'origine. La construction du modèle 3D nécessite des captures tridimensionnelles de visage avec leur texture et un coûteux travail d'alignement des scans 3D sur un modèle de référence.

Dans cette section, nous présentons des travaux préliminaires ayant pour but d'inclure des *a priori* sur la structure 3D du visage avec un coût réduit en matière de travail humain.

2.3.1 Optimisation auto-cohérente

La méthode proposée s'inspire des méthodes classiques d'Évaluation/Maximisation (EM), fréquemment utilisées en apprentissage statistique.

La méthode proposée utilise un maillage 3D semi-rigide représentant un visage humain. Dans nos expériences nous avons utilisé un maillage issu du logiciel libre MakeHuman™ [1].

Le principe de la méthode repose sur l'alternance entre une phase durant laquelle des modèles d'apparence locaux sont appris pour différents sommets du maillage, et une phase où la position du maillage est ajustée dans le but de maximiser l'adéquation entre les modèles d'apparence appris et les zones de l'image situées au voisinage des sommets correspondants. L'optimisation se fait donc sur un ensemble de visages (et non une unique image).

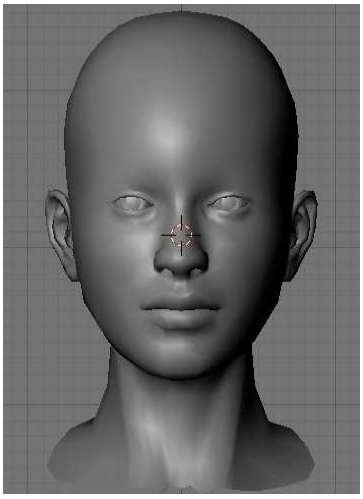


FIGURE *2.9: Modèle 3D de la tête dans MakeHuman*

Pour une image et un sommet du maillage donnés, l'apparence du sommet correspond à une description locale de l'image à la position du sommet. Plusieurs descripteurs sont envisageables. Dans nos expériences, les meilleurs résultats ont été obtenu avec un descripteur SIFT [83] (nous n'avons pas testé ici notre descripteur HOLD), c'est-à-dire un histogramme d'orientations de gradient calculé sur une zone centrée à la position du sommet considéré.

Le modèle d'apparence est en fait un classifieur SVM ayant appris à différencier les descripteurs calculés aux sommets considérés de descripteurs pris au hasard dans l'image. L'adéquation au modèle est mesurée comme la valeur de la fonction de décision du classifieur soit la distance algébrique à l'hyperplan du modèle. Et l'adéquation globale est mesurée comme la somme sur tous les sommets des adéquations locales. L'algorithme du simplex est utilisé pour maximiser l'adéquation en fonction des paramètres de pose.

Les paramètres utilisés sont la translation 2D du barycentre du modèle par rapport au centre de l'image (2 paramètres), les angles d'Euler de la rotation du modèle (3 paramètres) et un facteur d'échelle (1 paramètre) pour déterminer la pose 3D. Un septième paramètre prenant en compte l'élongation verticale du visage est introduit pour rendre le modèle moins rigide.

2.3.2 Résultats expérimentaux

Nous avons appliqué cette méthode à la base *Labeled Faces in the Wild* (LFW) section 1.1.3.2, page 5.

1. http://www.makehuman.org

Il est important d'éviter le sur-apprentissage des modèles d'apparence faute de quoi, la pose optimale serait systématiquement celle à laquelle les modèles ont été appris. Pour éviter ce problème l'ensemble des images utilisé est divisé en deux sous-ensembles et les modèle appris sur l'un sont appliqués à l'autre pour la phase d'optimisation de l'adéquation.

La figure 2.10 montre quelques résultats obtenus. La ligne supérieure montre la position initiale du maillage obtenu en ajustant la position du maillage à partir de 5 points-clés détectés automatiquement. La ligne du bas montre le résultat après la phase d'optimisation. Ces exemples montrent que la pose 3D est bien retrouvée.

FIGURE *2.10: Quelques résultats obtenus avec notre méthode pour récupérer la pose 3D. Ligne du haut : initialisation du modèle 3D. Ligne du bas : position après optimisation.*

Le problème majeur de la méthode tient à l'évaluation de la qualité de l'ajustement. En effet, en l'absence d'annotations il est difficile d'avoir une mesure quantitative des performances obtenues. C'est d'ailleurs une des principales raisons pour lesquelles ces travaux sont restés inachevés.

Une manière indirecte permettant de déterminer la qualité de l'ajustement consisterait à utiliser l'estimation de la pose 3D obtenue pour ramener tous les visages obtenus dans une position standard. Pour ce faire, nous avons projeté l'image sur le maillage ajusté puis reconstruit l'image avec le maillage en position frontale standard.

La figure 2.11 montre le résultat de cette procédure. La colonne de gauche montre l'image originale et la colonne de droite le résultat obtenu. Les pixels noirs dans la colonne de droite correspondent tout simplement à des pixels ne se trouvant pas sur la partie visible du maillage ajusté.

Nous observons que, en ce qui concerne la partie du visage la plus exposée, la compensation de pose fonctionne plutôt bien. Par contre, les parties correspondant à la limite du maillage visible sont très déformées. Ajouté au problème des pixels manquant, cela a pour conséquence que les images obtenues ne sont pas utilisables en l'état pour effectuer une comparaison et un travail important reste à faire pour que la méthode soit utilisable en pratique.

FIGURE *2.11: Compensation de pose 3D.*

CHAPITRE 3

Apprentissage de distance et distance de commutation sur graphe

Ce chapitre porte sur la vérification, tâche dans laquelle l'objectif est de prévoir si deux images de visages représentent la même personne ou non. L'originalité de l'approche proposée ici est de combiner deux mesures de similarités prenant en compte des aspects différents. La première est une mesure de distance apprise en projetant linéairement les représentations des visages dans un espace dans lequel les visages d'une même personne sont plus proches que ceux de personnes différentes, et une méthode de représentation par graphes des visages et de leurs plus proches voisins. Avant de présenter la méthode à proprement parler, nous commencerons par des rappels bibliographiques sur l'apprentissage de distance.

Sommaire

3.1 L'apprentissage de distance

Comme nous l'avons déjà évoqué dans le chapitre précédent, de nombreux problèmes en vision par ordinateur – et plus généralement en analyse de données – reposent sur l'utilisation de fonctions de distance entre paires de points. Parmi ces problèmes, la vérification de visage est une tâche importante.

Comme nous l'avons également expliqué dans le chapitre précédent, la vérification de visage [66, 93] consiste à déterminer si deux images représentent la même personne ou non. Deux ingrédients clés pour aborder ce type de problèmes sont (1) les représentations (descripteurs, signatures) utilisées pour représenter les images et (2) la fonction de distance utilisée pour comparer les signatures. Dans le travail qui suit, nous nous attachons principalement au deuxième aspect du problème et utiliserons des descripteurs puisés dans la littérature du domaine.

Étant donné les nombreuses sources de variation non contrôlées (comme les changements d'illumination, la pose de la personne, les propriétés de la caméra), il est peu probable qu'une mesure de distance standard telle que la distance euclidienne puisse se montrer appropriée, même en présence de signatures d'images particulièrement pertinentes. C'est pourquoi, Guillaumin *et al*. ont, avec succès, abordé le problème de la vérification de visage en apprenant des mesures de distances spécifiques à la tâche à accomplir [58].

L'apprentissage de distance est un sujet déjà bien étudié. En effet de nombreux algorithmes largement utilisés, tels que le partitionnement (*clustering*) non supervisé (par ex. les k-moyennes), les plus proches voisins et les classifieurs à noyaux, nécessitent l'utilisation d'une métrique dans l'espace des données d'entrée. Une telle métrique n'est pas seulement censée refléter les propriétés intrinsèques des données, mais doit de surcroît être adaptée au domaine d'application. Pour cette raison, de nombreuses approches ont tenté d'apprendre la mesure de distance en intégrant des contraintes spécifiques du domaine étudié [132, 84, 104, 53, 9, 51, 126, 74, 112, 58, 139].

Bien que toutes les méthodes d'apprentissage de distance reposent plus ou moins sur l'idée intuitive que des éléments semblables devraient être plus proches que des éléments dissemblables, la plupart ne sont pas adaptées à la tâche qui nous intéresse. En effet, soit elles supposent que les données disponibles pour l'entraînement sont complètement annotées (c'est-à-dire que l'étiquette de classe est donnée) [51, 53, 74, 112, 127], soit elles impliquent des présupposés concernant la structure des données ou des contraintes [9, 27, 139], ou bien encore, elles présentent des problèmes de performances lorsque la dimension des données d'entrée est importante ou lorsque la quantité de données d'entraînement est faible [40, 58].

Dans ce chapitre, nous présentons un nouvel algorithme d'apprentissage de distance applicable lorsque l'on dispose uniquement d'un jeu limité de contraintes données sur des paires de points dans un espace de grande dimension. En d'autres termes, nous nous intéressons aux problèmes où l'information de similarité/dissimilarité n'est connue que pour un nombre restreint de paires de points. Nous construisons pour cela un espace de dimension réduite dans lequel les contraintes sont respectées en minimisant les distances trop grandes pour les paires positives et trop petites pour les paires négatives. La relation entre l'espace des données et l'espace appris est linéaire. Pour capturer les aspects non linéaires des données (variété ou partitionnement naturel), un graphe de voisinage est ensuite construit dans l'espace appris incluant les données de test, ce qui nous autorise finalement à projeter les données dans un espace où la distance euclidienne est une approximation d'une mesure de distance sur le graphe. Notre méthode en deux étapes est ensuite validée sur une base particulièrement ardue, la base Labeled Faces in the Wild [66] (voir la section 1.1.3.2 pour une présentation de la base).

3.2 Les algorithmes pour l'apprentissage de distance

L'apprentissage de distance joue un rôle significatif en reconnaissance de motifs (*pattern recognition*) et a, par conséquent, était particulièrement étudié.

La littérature en apprentissage de distance peut être séparée en deux catégories principales : l'apprentissage de variétés, dont l'idée clé est d'apprendre une variété de basse dimension qui sous-tend les données, et les approches supervisées qui tentent d'apprendre une métrique pour laquelle les points appartenant à la même classe restent proches alors que les points de classes différentes sont éloignées. Notre méthode tente de combiner les deux approches.

3.2.1 La distance de Mahalanobis

Suivant les travaux précurseurs de Xing *et al*. [132], la plupart des approches d'apprentissage de distance apprennent une distance similaire à la distance de Mahalanobis :

$$D_M(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T M (\boldsymbol{x} - \boldsymbol{y})} \tag{3.1}$$

où M est une matrice semi-définie positive permettant de satisfaire les contraintes d'entraînement. Le principal avantage est que l'optimisation par rapport à la matrice M peut être vu comme un problème d'optimisation convexe sous contraintes pour lequel il existe des algorithmes efficaces. Par ailleurs, Kwok *et al*. [113] ont montré comment étendre ces méthodes au cas non linéaire en appliquant l'"astuce du noyau" à ce type de méthodes.

Cependant maintenir la contrainte imposant que M doit être semi-définie positive peut être coûteux en temps de calcul. C'est pourquoi, dans plusieurs travaux, tels que [53, 112, 139], la matrice M est factorisée en $M = L^T L$, ce qui garantit que M est toujours semi-définie positive et définit implicitement une projection dans un espace de dimension réduite où les distances reflètent les contraintes de similarités. Notre travail suit cette ligne de recherche.

À côté de ces méthodes générales, plusieurs approches se sont particulièrement penchées sur le problème de la classification par k-plus proches voisins (kPPV). Dans ces méthodes, des contraintes sur les distances intra- et inter-classes sont introduites soit de manière globale comme dans la méthode NCA (*Neighborhood Component Analysis*) [53] ou la méthode MCC (*Maximally Collapsing Classes*) [51], soit de manière relative en prenant en compte la classe des points dans le k-plus proche voisinage, comme dans la méthode emblématique LMNN (*Large Margin Nearest Neighbours*) [126, 127] ou ses variantes comme LMCA (*Large Margin Component Analysis*) [112], *invariant* LMNN [74] et LMNN-R [41]. Comme nous l'avons signalé plus haut, ces méthodes ont besoin de connaître les étiquettes de classe pour tous les points d'entraînement, et, par conséquent, ne sont pas adaptés aux problèmes pour lesquels seules des contraintes sur les paires sont disponibles.

D'un autre côté, des méthodes comme la méthode OASIS (*On-line Algorithm for Scalable Image Similarity*) de Chechik *et al*. [27] ou la récente méthode PRDC (*Probabilistic Relative Distance Comparison*) de Zheng *et al*. [139] sont spécifiquement conçues pour fonctionner avec des contraintes sur les paires. Cependant, elles reposent toutes les deux sur l'hypothèse que durant l'entraînement sont disponibles pour chaque point de données, un point similaire (ou de même classe) et un point dissimilaire (ou de classe différente). Si ces contraintes sont moins fortes que pour les méthodes nécessitant tous les labels de classes, elles rendent néanmoins ces méthodes en grande partie inapplicables dans un problème tel que la vérification de visage sur la base LFW. Certes, il est éventuellement possible d'étendre un jeu de contraintes portant uniquement sur les paires en détectant les composantes connexes dans les données (les contraintes de similarité jouant le rôle d'arêtes dans un graphe) et en propageant les informations de similarité/dissimilarité à tous les éléments de ces composantes. Toutefois, dans le cas où ces informations sont très restreintes, il est clair que toute l'information ne sera pas exploitable. Ce serait par exemple le cas pour deux points n'intervenant qu'une fois dans une paire négative.

Les méthodes d'apprentissage de distance pouvant s'appliquer directement sur des contraintes données sur les paires ne sont en définitive pas très nombreuses et deux ont particulièrement retenu notre attention. Il s'agit de la méthode ITML (*Information Theoretic Metric Learning*) de Davis *et al*. [40] et de la méthode LDML (*Logistic Discriminant Metric Learning*) de Guillaumin *et al*. [58]. Pourtant, les propriétés de généralisation sont décevantes lorsque la quantité des données d'entraînement est faible [58, 139]. De plus, ITML utilise un critère de régularisation basé sur une divergence de Kullback-Leibler et une méthode itérative de mise-à-jour de matrice M *ad hoc*, pour garantir que M reste semi-définie positive ce qui ajoute un important surcoût en termes de temps de calcul. Les deux méthodes optimisent la matrice de rang complet M ce qui fait que le nombre de paramètres à apprendre croît quadratiquement avec la dimension de l'espace d'entrée, ce qui rend ces deux méthodes difficiles à appliquer sur des données de grande dimension dans réduction de dimension préalable, entraînant nécessairement une perte d'information. Il est enfin intéressant de remarquer que LDML, quoiqu'ignorant les contraintes sur M, semble néanmoins présenter de meilleures performances qu'ITML. Pour cette raison la première étape de notre approche est une amélioration de LDML.

3.2.2 Les approches basées sur les graphes et l'apprentissage de variété

Il est fréquent que la dimension intrinsèque des données soit plus petite que la dimension de l'espace d'entrée. Autrement dit, les données représentées dans une espace à D dimensions sont souvent sous-tendues par une variété de dimension $d < D$ plus petite. Dans le cas où l'on n'a pas d'apriori sur la forme de la variété, les graphes représentent un outil commode pour analyser la structure de voisinage. Le but final de ces méthodes est généralement le même : trouver un plongement du graphe dans lequel la distance euclidienne reflète les distances « naturelles » sur le graphe.

La méthode ISOMAP de Tenenbaum *et al*. [110] est une transcription presque littérale de cette idée. Son but est de plonger les points de données dans un espace de faible dimension dans lequel la distance euclidienne correspond au mieux à la distance géodésique sur le graphe. Un des problèmes de cette méthode est qu'il peut être difficile de plonger un nouveau point dans cet espace.

La méthode LLE (*Locally Linear Embedding*) [102], cherche à préserver la reconstruction locale des points à partir de leurs voisins. D'autres méthodes comme les *Laplacian Eigenmaps* [14], ou LPP (*Locality preserving projections*) [62] tentent plutôt de préserver la structure locale des données.

Yan *et al*. [133] ont proposé un cadre général englobant toutes ces méthodes. Nous résumons ici leur approche.

Soit W une matrice $N \times N$ dont les éléments représentent les similarités deux à deux entre N points. Un plongement à une dimension y_i pour chaque point, tel que les distances entre points reflètent les similarités, est recherché. Ce plongement est obtenu par la minimisation de :

$$\arg\min_{\boldsymbol{y}} \sum_{i=1}^{N} \sum_{j=1}^{N} W_{ij}(y_i - y_j)^2$$

De manière informelle, cette fonction nous dit que la distance entre les points doit être d'autant plus petite que leur similarité est grande. Pour ne pas converger vers des solutions triviales il faut par ailleurs rajouter des contraintes sur le vecteur $\boldsymbol{y}$ qui contient les différentes valeurs y_i. Ces contraintes prennent la forme :

$$\boldsymbol{y}^T B \boldsymbol{y} = 1$$

Où la matrice B est une matrice qui permet de spécifier les propriétés qui doivent être conservées.

Appelons D la matrice diagonale telle que $D_{ii} = \sum_{j=1}^{N} W_{ij}$, la matrice $L = D - W$ est appelée le

Laplacien du graphe et, après quelques manipulations, il est possible d'aboutir au problème suivant :

$$\begin{aligned} \underset{\boldsymbol{y}}{\arg\min} \quad & \boldsymbol{y}^T L \boldsymbol{y} \\ \text{avec} \quad & \boldsymbol{y}^T B \boldsymbol{y} = 1 \end{aligned} \tag{3.2}$$

Les solutions de ce problème sont données par les solutions de l'équation aux vecteurs propres et valeurs propres :

$$L\boldsymbol{y} = \lambda B \boldsymbol{y} \tag{3.3}$$

Les k vecteurs propres associées aux plus petites valeurs propres non nulles, nous donnent donc un plongement en k-dimension.

Cette première étape est appelée le plongement direct (PD). Il peut être nécessaire de pouvoir projeter facilement des nouveaux points dans cet espace. Une approche simple pour réaliser ceci consiste à trouver une approximation linéaire du plongement précédent. Pour ceci on suppose que le plongement $\boldsymbol{y}$ correspond à une simple projection des données sur un vecteur $\boldsymbol{w}$, $\boldsymbol{y} = X^T \boldsymbol{w}$, où X est la matrice dont les colonnes sont les points de données. On obtient alors le problème suivant :

$$\begin{aligned} \underset{\boldsymbol{y}}{\arg\min} \quad & \boldsymbol{w}^T X L X^T \boldsymbol{w} \\ \text{avec} \quad & \boldsymbol{w}^T X B X^T \boldsymbol{w} = 1 \\ \text{ou} \quad & \boldsymbol{w}^T B' \boldsymbol{w} = 1 \end{aligned} \tag{3.4}$$

où l'on introduit la matrice B' afin de pouvoir décrire une plus grande variété de contraintes. Cette formulation correspond à une linéarisation du plongement.

L'étape suivante consiste à appliquer l'astuce du noyau. En écrivant $\boldsymbol{w}$ comme une combinaison linéaire des données $\boldsymbol{w} = X\boldsymbol{\alpha}$. Il vient alors :

$$\begin{aligned} \underset{\boldsymbol{y}}{\arg\min} \quad & \boldsymbol{\alpha}^T K L K \boldsymbol{w} \\ \text{avec} \quad & \boldsymbol{\alpha}^T K B K \boldsymbol{\alpha} = 1 \\ \text{ou} \quad & \boldsymbol{\alpha}^T K \boldsymbol{\alpha} = 1 \\ \text{ou} \quad & \boldsymbol{\alpha}^T B' \boldsymbol{\alpha} = 1 \end{aligned} \tag{3.5}$$

D'autre méthodes bien connues peuvent s'exprimer dans ce cadre comme l'analyse en composante principale (PCA pour *Principal Component Analysis*) et sa version noyau (KPCA [1]) ou l'analyse linéaire discriminante (LDA pour *Linear Discriminant Analysis*) et sa version noyau (KLDA).

Le tableau 3.1, repris de [133], donne pour les différentes méthodes citées, les matrices W et B,

Dans nos travaux, nous utilisons un plongement (*embedding*) basé sur la pseudo-inverse L^+ du laplacien du graphe L issus des travaux de Fouss *et al.* [48]. Si cette approche s'intègre parfaitement à l'intérieur du cadre théorique que nous venons de voir (il suffit de prendre $B = I$ au lieu de $B = D$ dans la méthode LE), elle a de surcroît l'avantage d'être fondée sur le socle théorique des chaînes de Markov et des marches aléatoires sur graphes. Ce cadre nous permet de définir la notion de *distance euclidienne du temps de commutation* (DETC) qui est dans le cadre des marches aléatoires une mesure naturelle de la distance entre les sommets d'un graphe et qui s'avère plus robuste que la distance géodésique utilisée dans la méthode ISOMAP.

La DETC est aussi à mettre en relation avec la notion distance de diffusion développée par Coifman *et al.* [31], et peut être vue comme une variante multi-échelle de cette mesure de distance.

L'utilisation du plongement associé à la DETC constitue la deuxième étape de notre méthode et nous permet de capturer d'éventuelles structures non linéaires dans les données.

Nous présentons à présent les différentes étapes de notre méthode.

1. le « K » signifie *kernel* c'est-à-dire *noyau* en français.

Algorithm	Définition de W et B	Type
PCA/KPCA	$W = 1/N,\ i \neq j;\ B = I$	L/K
LDA/KLDA	$W = \delta_{l_i,l_j}/n_{l_i};\ B = J$	L/K
ISOMAP	$W_{ij} = -JD_GJ/2;\ B = I$	D
LLE	$W = M + M^T - M^TM;\ B = I$	D
LE/LPP	$W_{ij} = v_{ij}e^{-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2/t};\ B = D$	D/L

avec :
- D/L/K : plongement direct (D), linéarisation (L) et version noyau ou *kernelisation* (K) ;
- N : le nombre total d'échantillons ;
- l_i : le label de classe de l'échantillon i ;
- δ_{l_i,l_j} : l'indice de Kroenecker indiquant si i et j on la même classe ;
- n_{l_i} : nombre d'échantillons portant le label l_i ;
- J : matrice centrant les données $J = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$;
- D_G : les matrices dont les éléments sont le carré des distances géodésiques entre les échantillons pris deux à deux ;
- M : la matrice contenant les coefficients permettant de reconstruire chaque échantillon comme une combinaison linéaire de ses voisins ;
- v_{ij} la variable indiquant si i et j sont voisins ;
- t : la largeur du noyau gaussien.

TABLE *3.1: Définitions des matrices W et B pour différents algorithmes connus.*

3.3 Apprentissage de distance par analyse en composantes logistiques discriminantes

3.3.1 Formulation de départ (LDML)

Notre approche reprend et améliore la méthode proposée par Guillaumin *et al.* [58]. La méthode LDML (pour *logistic discriminant metric learning*) définit un critère reposant sur la probabilité que les éléments de la paire $n = (i, j)$ appartiennent à la même classe, autrement dit, que l'étiquette t_n de la paire soit 1. Cette probabilité est calculée à partir d'une distance de Mahalanobis et de la fonction logistique $\sigma(x) = 1/(1 + e^{-x})$:

$$p_n = p(y_i = y_j|\boldsymbol{x}_i, \boldsymbol{x}_j, M) = \sigma(b - D^2_M(\boldsymbol{x}_i, \boldsymbol{x}_j))$$

Le paramètre b agit comme un seuil et est appris en même temps que la matrice M, par maximisation de la vraisemblance de l'étiquetage correct des paires d'apprentissage.

La ***log-vraisemblance*** $\mathcal{L}$ s'écrit :

$$\mathcal{L} = \sum_n t_n \ln p_n + (1 - t_n) \ln(1 - p_n)$$

et son gradient est calculé par :

$$\frac{\partial \mathcal{L}}{\partial M} = -\sum_n (t_n - p_n)C_n$$

où $C_n = (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T$, et :

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_n (t_n - p_n)$$

La fonction $\mathcal{L}$ est lisse et convexe ce qui garantie donc la convergence vers un minimum global. Cependant, cette formulation souffre de plusieurs inconvénients qui la rendent coûteuse en temps de calcul.

3.3.2 Inconvénients de la méthode LDML

Tout d'abord, pour que la mesure $D(\boldsymbol{x}_i, \boldsymbol{x}_j)$ soit une mesure de distance, la matrice M doit être semi-définie positive. Pour satisfaire cette contrainte, une approche courante, dans le cadre d'une méthode itérative, consiste à projeter à chaque itération la matrice M sur le cône des matrices semi-définies positives. Une manière de réaliser cette opération consiste à calculer le spectre des valeurs propres λ_i de la matrice M. Pour toute valeur propre $\lambda_i < 0$, on retire la contribution correspondante. Si $\boldsymbol{u}_i$ est le vecteur propre associé à λ_i, la projection $\hat{M}$ de la matrice M courante sur le cône des matrices semi-définies positives, est obtenue par l'opération :

$$\hat{M} = M - \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^t, \ \forall i \text{ tel que } \lambda_i < 0$$

Lorsque la taille de la matrice M est grande, cette méthode peut toutefois se révéler coûteuse. En pratique, dans [58], Guillaumin *et al*. ne s'intéressent qu'à la classification des paires et ignorent cette contrainte.

Dans d'autres méthodes comme ITML (*Information theoretic metric learning*) [40], le problème est formulé comme la minimisation d'une mesure de la divergence entre la matrice M et une matrice de référence M_0 soumise à des contraintes de distance sur des paires de points. En exploitant les propriétés de la divergence utilisée, le problème est projeté de manière itérative sur chaque contrainte prise l'une après l'autre. La projection utilisée (projection de Bergman) garantit que la matrice M reste semi-définie positive. Les mises à jour de la matrice M sont de la forme :

$$M^{t+1} = M^t + \eta M^t C_n M^t$$

Bien que cette approche ne nécessite pas d'effectuer une décomposition aux valeurs propres et vecteurs propres, elle fait néanmoins intervenir la multiplication de trois matrices potentiellement grandes.

Là encore, lorsque l'espace d'entrée est de (grande) dimension D, la taille de matrice M est $D \times D$. Ceci affecte les temps de calcul mais augmente également les risque de sur-apprentissage. C'est pourquoi une première étape de réduction de dimension est souvent réalisée avant l'application de ces méthodes, par exemple grâce à une analyse en composantes principales (ACP) dans [58].

3.3.3 Notre contribution : *Logistic Discriminant Component Analysis* (LDCA)

La méthode que nous proposons tente de résoudre ces problèmes par l'utilisation d'un factorisation de la matrice M sous la forme $M = L^T L$. L est une matrice rectangulaire de dimension $d \times D$ où $d \ll D$.

Cette approche présente plusieurs avantages :

- mise sous cette forme, la matrice M est toujours semi-définie positive ;
- lorsque la valeur de d est petite, le nombre de paramètres à apprendre est également nettement plus petit ce qui rend l'algorithme plus robuste vis-à-vis du sur-apprentissage et réduit les temps de calcul ;
- la transformation $\boldsymbol{x}' = L\boldsymbol{x}$ définit implicitement une projection dans un espace de dimension réduite d où les distances reflètent les contraintes de similarité induites par les étiquettes sur les paires.

La probabilité p_n devient alors :

$$p_n = \sigma(b - \|L(\boldsymbol{x}_i - \boldsymbol{x}_j)\|^2)$$

et le gradient de la fonction $\mathcal{L}$:

$$\frac{\partial \mathcal{L}}{\partial L} = -2L \sum_n (t_n - p_n)\mathbf{C}_n$$

Notons que cette paramétrisation est similaire à celle utilisée par Torresani *et al*. [112] pour l'analyse en composantes à vaste marges LMCA (large margin component analysis) qui traite le problème de la réduction de dimension dans le contexte des k-plus proches voisins à vaste marge proposant ainsi une version factorisée de la méthode LMNN de Weinberger *et al*. (*large margin nearest neighbors*) [126]. L'algorithme de Goldberger *et al*, NCA (*neighborhood component analysis*) [53], utilise également une telle factorisation. C'est pourquoi, par analogie, nous appellerons notre méthode *analyse en composantes logistiques discriminantes* ou ***LDCA*** (*Logistic Discriminant Component Analysis*).

3.3.4 Convexité

Un inconvénient potentiel de la factorisation est que la fonction de coût devient non convexe. Cet argument est d'ailleurs utilisé par Globerson *et al*. [51] pour proposer une version non factorisée de la méthode de NCA de Goldberger *et al*. [53]. Pourtant Goldberger *et al*. [53] aussi bien que Torresani *et al*. [112] ainsi que Weinberger *et al*. dans [127], remarquent – sans l'expliquer – que malgré l'absence de convexité, les solutions trouvées semblent être aux moins de « bons » minima locaux.

L'apprentissage de distance, lorsqu'il est formulé comme la minimisation d'une fonction d'une matrice de Mahalanobis est une instance d'une famille plus grande de problèmes connus sous le nom de ***programmation semi-définie positive*** :

$$\begin{array}{ll} \min\limits_X & f(X) \\ \text{soumis à} & X \succeq 0 \end{array}$$

où f est une fonction convexe définie sur l'espace des matrices symétriques.

Or, il est possible de montrer [23, 70] que, sous certaines conditions (souvent vérifiées en pratique), la solution de ce problème contraint correspond aux points stationnaires du problème non-contraint (mais non-convexe) :

$$\min_Y \quad f(Y^T Y)$$

Journée *et al*. [70] proposent un algorithme général pour résoudre ce problème. Celui-ci est présenté en annexe C. Dans nos travaux, nous avons utilisé une approche plus simple. En effet, la méthode de Journée *et al*. nécessite le calcul fréquent du gradient et de la matrice Hessienne de la fonction de coût. Ces deux informations sont généralement coûteuses à calculer, c'est pourquoi nous avons choisi une méthode basée sur une descente de gradient. En pratique, nous utilisons donc la méthode du gradient conjugué pour l'optimisation.

À l'issue de cette phase, nous disposons d'une mesure de distance qui peut être utilisée directement pour déterminer si deux visages sont identiques ou non, par simple seuillage.

Même si cette méthode donne de bons résultats, comme nous le montrons plus loin, il nous a paru intéressant de la faire suivre d'une phase permettant de tenir compte des non-linéarités dans l'espace de représentation et permettant également de procéder à un entrainement semi-supervisé. Cette phase est présentée dans les sections suivantes.

3.4 Distance sur graphe et apprentissage semi-supervisé

La méthode précédente souffre de deux limitations que nous nous proposons de pallier ici. Tout d'abord la transformation appliquée à l'espace d'origine est une transformation linéaire, ce qui ne permet pas de prendre en compte le fait que les données se trouvent éventuellement sur des variétés

de formes quelconques. Une seconde limitation provient du fait que la méthode est entièrement supervisée, ce qui est un handicap pour ce genre de tâche : s'il est très facile d'obtenir une multitude de paires de visages à partir d'images collectées sur internet, il est très coûteux d'avoir des annotations indiquant quelles sont les paires positives et quelles sont les paires négatives. Une méthode semi-supervisée semble donc particulièrement pertinente dans ce cas.

3.4.1 Apprentissage semi-supervisé

L'apprentissage semi-supervisé repose principalement sur deux hypothèses :

– les données reposent sur un support dont la dimension intrinsèque est plus petite que le nombre de paramètres du problème (hypothèse de la variété),
– les données forment des groupes naturels (hypothèse des agrégats ou *clusters*)

Dans les deux cas, les données correspondent à un échantillonnage d'une distribution spécifique. Le nombre d'échantillons est donc déterminant si on veut pouvoir mettre en évidence cette structure. Les méthodes semi-supervisées utilisent précisément les données non étiquetées pour augmenter le nombre d'échantillons et en faciliter l'apprentissage.

Pour capturer au mieux la structure, il est usuel d'avoir recours à un graphe de voisinage [25]. La figure 3.1 montre l'effet de l'échantillonnage sur une distribution en deux demi-lunes imbriquées : la structure de la distribution sous-jacente se fait plus évidente lorsque le nombre d'échantillons augmente.

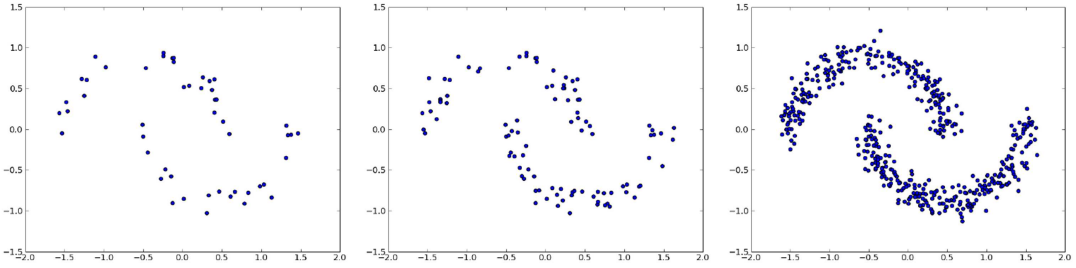


FIGURE *3.1: Effet de l'échantillonnage. Exemple d'une distribution pour un nombre croissant d'échantillons.*

Étant donnée une mesure de distance pertinente pour le problème, chaque point de donnée, considéré comme un sommet du graphe, est relié à ces voisins par une arête, formant ainsi un graphe.

Plusieurs types de voisinages sont possibles :

Le voisinage naturel : les voisins sont définis comme étant les sommets des cellules de Voronoï[2] adjacentes à la cellule centrée sur le sommet considéré. Dans ce cas, le graphe de voisinage formé correspond à la triangulation de Delauney du graphe. Ce voisinage est cependant coûteux à calculer.

Le ε-voisinage : qui correspond à l'ensemble des points situés à moins d'une distance ε les uns des autres. Cependant pour obtenir un graphe connexe, ε doit être au moins égal à la plus petite distance entre deux points du graphe. Si la densité des données dans l'espace est très variable, on peut se retrouver avec des zones trop fortement connectées.

Le k-plus proche voisinage : comme son nom l'indique les k-plus proches voisins d'un point sont les k points dont la distance au point de référence est la plus petite. Contrairement, ce voisinage

2. Soit un ensemble $\mathcal{E}$ dénombrable d'éléments d'un espace de Hilbert $\mathcal{H}$ muni d'une mesure de distance $d(.,.)$, la cellule de Voronoï v_i associée au i-ième élément $\boldsymbol{x}_i$ de $\mathcal{E}$ est l'ensemble des points de $\mathcal{H}$ définis par $v_i = \{\boldsymbol{y} \in \mathcal{H} | d(\boldsymbol{x}_i, \boldsymbol{y}) < d(\boldsymbol{x}_j, \boldsymbol{y}) \quad \forall \boldsymbol{x}_j \in \mathcal{E},\ j \neq i$.

n'est pas symétrique. En effet, si $\mathcal{N}_i^k$ est l'ensemble des k-plus proches voisins de i, $j \in \mathcal{N}_i^k \not\Leftrightarrow i \in \mathcal{N}_j^k$ comme illustré à la figure 3.2c. Pour symétriser la relation on dira que i et j sont k-plus proches voisins si $j \in \mathcal{N}_i^k \vee i \in \mathcal{N}_j^k$

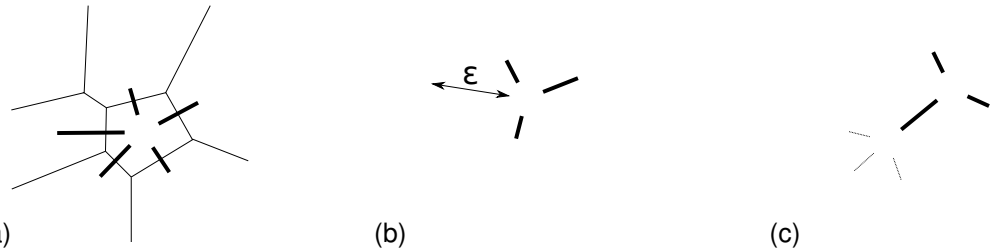


FIGURE *3.2: Différents voisinages. (a) Le voisinage naturel relie les centres des cellules de Voronoï (représentés par les lignes fines) voisines. (b) Le ε-voisinage du point noir sont les points situés à l'intérieur de la boule de rayon ε centrée sur le point. (c) Les k-plus proches voisins ($k = 3$) du point noir sont indiqués, par un petit disque noir alors que les k-plus proches voisins du point blanc sont indiqués par un petit disque blanc.*

La figure 3.3 montre le graphe de k-plus proche voisinage correspondant à la distribution de la figure 3.1. La valeur de k est choisie comme la valeur minimale permettant de connecter le graphe.

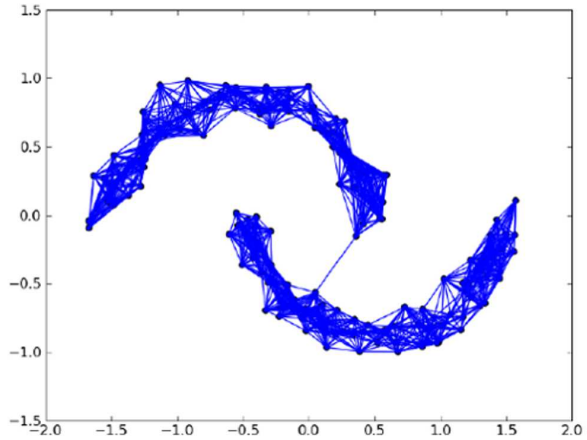


FIGURE *3.3: Le graphe de k-plus proche voisinage correspondant à la distribution de la figure 3.1.*

Une fois le graphe de voisinage construit, il est possible de calculer des distances entre les points des données basées sur la structure du graphe. Alors que la distance associée au plus court chemin est souvent utilisée, nous avons utilisé dans nos travaux, la ***distance euclidienne du temps de commutation*** qui est plus robuste au bruit et reflète mieux le degré de connexion entre les sommets du graphe.

Pour expliquer notre méthode, nous avons besoins de certaines notions de théorie des graphes. Nous commençons donc par quelques définitions et généralités. Ceci nous permettra d'introduire la notion de ***distance euclidienne du temps de commutation*** comme mesure de la distance entre les sommets d'un graphe. Les notations et démonstrations présentées ici, sont principalement empruntées à F. Fouss *et al* [48].

3.4.2 Notions de théories des graphes

Les graphes sont un moyen commode pour représenter des relations entre les éléments constitutifs d'un système. Les éléments de ce système sont représentés comme les sommets du graphe et les relations qui existent entre ces éléments sont représentées comme des arêtes reliant ces sommets, diverses propriétés (direction, poids, ...) pouvant être associées aux arêtes. Les graphes sont par exemple utilisés pour représenter les flux dans les réseaux. Un important arsenal d'outils mathématiques a été développé pour étudier leurs propriétés.

Nous donnons à présent quelques définitions qui nous seront utiles pour la suite de la discussion.

Graphes. Nous définissons un *graphe* $G = (\mathcal{S}, \mathcal{A})$ comme la donnée de deux ensembles : un ensemble $\mathcal{S}$ dénombrable de *sommets* et un ensemble d'*arêtes* $\mathcal{A}$. Une arête notée ij représente un lien entre deux éléments de $\mathcal{S}$. Dans le cas d'un graphe orienté les arêtes possèdent une direction. Ainsi, l'arête ij représente un lien du sommet i vers le sommet j. Les graphes que nous utiliserons ici ne sont pas orientés ce qui signifie que l'arête ij représente un lien dans les deux directions, autrement dit $ij = ji$. Un graphe est un ***graphe pondéré*** si à chaque arête ij on associe un poids w_{ij}. Un graphe non pondéré peut être représenté comme un graphe pondéré tel que $w_{ij} = 1$ si $ij \in \mathcal{S}$.

Graphe complet. Un graphe est complet si pour toute paire (i, j) d'éléments de $\mathcal{S}$ l'arête ij existe dans $\mathcal{A}$.

Chaîne. Dans un graphe non orienté, une chaîne reliant i et j est une suite finie d'arêtes consécutives permettant de passer de i à j.

Graphe connexe. Un graphe non orienté est dit connexe si quels que soient les sommets i et j, il existe une chaîne de i vers j.

Voisinage. Le voisinage $\mathcal{V}_i$ d'un sommet i est défini comme l'ensemble des sommets liés à i par une arête partant de i.

$$\mathcal{V}_i = \{j \in \mathcal{S} \mid ij \in \mathcal{A}\}.$$

Degré d'un sommet. Dans un graphe non orienté G, le degré d_i d'un sommet i est défini comme la somme des poids des arêtes partant de i :

$$d_i = \sum_{j \in \mathcal{V}_i} w_{ij}$$

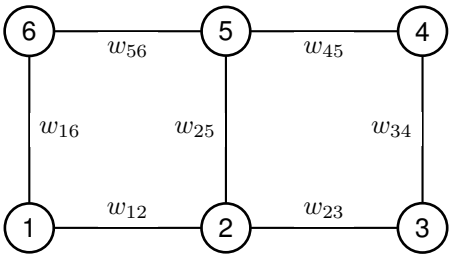
Dans le cas d'un graphe non pondéré, le degré d_i correspond donc simplement au nombre de voisins de i ($d_i = |\mathcal{V}_i|$). Nous définissons la ***matrice des degrés*** D comme la matrice diagonale telle que :

$$[D]_{ii} = d_i$$

Matrice d'adjacence. La matrice d'adjacence est un moyen commode pour représenter la connectivité d'un graphe pondéré. La matrice d'adjacence A d'un graphe G est la matrice telle que :

$$[A]_{ij} = \begin{cases} w_{ij} & \text{si} \quad ij \in \mathcal{A} \\ 0 & \text{sinon} \end{cases}$$

Soit $\mathcal{A}^*$ l'ensemble de toutes les arêtes qu'il est possible de construire à partir des sommets de $\mathcal{S}$. Avec la représentation de la matrice d'adjacence, tout graphe pondéré $G = (\mathcal{S}, \mathcal{A})$ peut être décrit de manière équivalente comme un graphe pondéré complet $G^* = (\mathcal{S}, \mathcal{A}^*)$ tel que $w^*_{ij} = 0$ si $ij \notin \mathcal{A}$ et $w^*_{ij} = w_{ij}$ sinon. Dans le cas d'un graphe non orienté la matrice d'adjacence est symétrique $A = A^T$.



$$A = \begin{pmatrix} 0 & w_{12} & 0 & 0 & 0 & w_{16} \\ w_{12} & 0 & w_{23} & 0 & w_{25} & 0 \\ 0 & w_{23} & 0 & w_{34} & 0 & 0 \\ 0 & 0 & w_{34} & 0 & w_{45} & 0 \\ 0 & w_{25} & 0 & w_{45} & 0 & w_{56} \\ w_{16} & 0 & 0 & 0 & w_{56} & 0 \end{pmatrix}$$

FIGURE *3.4: Un graphe et sa matrice d'adjacence.*

3.4.3 Marche aléatoire sur un graphe

Dès l'origine de la théorie des graphes [44], ceux-ci ont été utilisés pour modéliser les réseaux routiers et naturellement intervient la notion de chemin et de marche (ou promenade), les sommets représentant des sites et les arêtes les routes reliant ces sites entre eux.

Se promener sur un graphe consiste alors à passer d'un sommet à un autre en utilisant les arêtes. Les poids des arêtes peuvent alors être utilisés pour représenter des propriétés telles que la longueur de la route figurée par l'arête.

Une ***marche aléatoire*** sur un graphe consiste à parcourir les sommets du graphe en utilisant ses arêtes. À un sommet donné, on considère donc l'ensemble des arêtes qui en partent et le choix de l'arête s'effectue aléatoirement selon une distribution de probabilité calculée à partir des poids des arêtes. Dans le cadre qui nous intéresse, les poids correspondent à des nombres positifs mesurant la similarité entre les sommets du graphe. Parmi les arêtes qu'il est possible d'emprunter pour quitter un sommet, celle associée à une plus grande similarité doit donc avoir une plus grande probabilité d'être choisie.

En pratique, la probabilité p_{ij} de passer d'un sommet i à un sommet $j \in \mathcal{V}_i$ peut se calculer par :

$$p_{ij} = \frac{w_{ij}}{\sum_{j' \in \mathcal{V}_i} w_{ij'}} = \frac{w_{ij}}{d_i} \tag{3.6}$$

La matrice de transition P permet de représenter l'ensemble des probabilités de passage d'un sommet à un autre $[P]_{ij} = p_{ij}$. Elle s'exprime en fonction des matrices d'adjacence A et des degrés D par :

$$P = D^{-1}A$$

Une marche aléatoire correspond à la chaîne de Markov décrivant la séquence de sommets visités par un marcheur aléatoire. Un variable aléatoire $e(t)$ représente l'état courant de la chaîne de Markov au temps t. Sachant que $e(t) = i$, la probabilité de se trouver sur le sommet j au temps $t+1$ est donnée par :

$$P(e(t+1) = j | e(t) = i) = p_{ij}.$$

Les probabilités de transition ne dépendent que de l'état courant (chaîne de Markov du premier ordre). Soit $\pi_i(t)$ la probabilité d'être sur le sommet i au temps t, l'évolution de la chaîne de Markov est donnée par $\boldsymbol{\pi}(t+1) = P^T\boldsymbol{\pi}(t)$ avec $\boldsymbol{\pi}(0) = \boldsymbol{\pi}^0$. Ceci fournit la distribution de probabilité des états $\boldsymbol{\pi}(t) = [\pi_1(t), \pi_2(t), \ldots, \pi_n(t)]$ au temps t lorsque la distribution initiale $\boldsymbol{\pi}^0$ est connue. Pour une vue plus détaillée sur les chaînes de Markov le lecteur pourra se référer par exemple à [71, 89].

3.4.4 Temps moyen de premier passage et temps moyen de commutation

Le ***temps moyen de premier passage*** $m(k|i)$ est défini comme le nombre moyen de pas qu'il faut à un marcheur aléatoire, partant du sommet $i \neq k$, pour atteindre le sommet k pour la première fois [89]. Si on définit le temps minimum pour atteindre l'état k en partant de i comme

$T_{ik} = \min(t \geq 0 | e(t) = k \wedge e(0) = i)$ pour une réalisation du processus stochastique. Le marcheur aléatoire passera à travers k à plusieurs reprises lors de sa marche, le temps minimum correspond au premier passage. Le temps de premier passage moyen est l'espérance mathématique de cette quantité $m(k|i) = E[T_{ik}|s(0) = i]$.

De manière similaire, le coût moyen de premier passage $o(k|i)$ est le coût supporté par le marcheur aléatoire partant de l'état i pour atteindre le l'état k pour la première fois. Le coût de chaque transition est donné par $c(j|i)$ pour tous les états i et j. On remarquera que $m(k|i)$ est un cas spécial de $o(k|i)$ obtenu lorsque $c(j|i) = 1$ pour tous les i, j.

Les relations de récurrence pour calculer $m(k|i)$ et $o(k|i)$ s'obtiennent facilement en appliquant les définitions [71, 89] :

$$\begin{cases} m(k|k) & = 0 \\ m(k|i) & = 1 + \sum_{j=1}^{n} p_{ij} m(k|j), \text{ for } i \neq k \end{cases} \tag{3.7}$$

$$\begin{cases} o(k|k) & = 0 \\ o(k|i) & = \sum_{j=1}^{n} p_{ij} c(j|i) + \sum_{j=1}^{n} p_{ij} o(k|j), \text{ for } i \neq k \end{cases} \tag{3.8}$$

Le sens de ces formules est évident : pour aller de l'état i à l'état k, il faut d'abord se déplacer vers n'importe quel état j voisin de i et continuer à partir de là. Ces quantités peuvent être calculées en appliquant ces relations de récurrence par l'utilisation d'algorithmes dédiés développés par la communauté des chaînes de Markov [71], ou en utilisant la pseudo-inverse de la matrice Laplacienne du graphe [48], comme nous allons le voir par la suite.

Le ***temps moyen de commutation*** $n(i, j)$ est défini comme le nombre moyen de pas nécessaires à un marcheur aléatoire pour, partant du sommet $i \neq j$, arriver au sommet j et revenir à i. Le temps de commutation moyen est donc relié au temps de premier passage moyen par :

$$n(i, j) = m(i|j) + m(j|i)$$

Nous remarquons que $n(i, j)$ est symétrique par définition alors que $m(j|i)$ ne l'est pas.

[52, 73] montrent que le temps moyen de commutation a les propriétés d'une distance car, quels que soient les sommets i, j et k : (1) $n(i, j) > 0$, (2) $n(i, j) = 0$ si et seulement si $i = j$, (3) $n(i, j) = n(j, i)$, (4) $n(i, j) \leq n(i, k) + n(k, j)$. Pour cette raison, on appellera $n(i, j)$ la ***distance du temps de commutation***.

Il existe une relation étroite entre le modèle des marches aléatoires et la théorie des réseaux électriques [19, 42]. En effet, $n(i, j)$ est proportionnelle à la résistance efficace entre deux nœuds i et j du réseau correspondant où une résistance $w_i j^{-1}$ est attribuée à chaque arête. C'est pourquoi $n(i, j)$ est aussi appelée ***distance de résistance***.

La distance du temps de commutation a la propriété d'augmenter lorsque le nombre de chemins reliant les deux sommets considérés augmente, propriété que n'a pas la distance géodésique (distance associée au plus court chemin) couramment utilisée.

3.4.5 La matrice laplacienne et sa pseudo-inverse

Les quantités définies à la section précédente peuvent s'exprimer en fonction des éléments de la pseudo-inverse de Moore-Penrose de la matrice laplacienne du graphe.

Le laplacien d'un graphe peut être vu comme un opérateur linéaire s'appliquant sur des fonctions définies sur les sommets du graphe. Une telle fonction associe à chaque sommet i du graphe un nombre réel f_i et peut donc se représenter par un vecteur $\boldsymbol{f}$ de taille n où n est le nombre de sommets dans le graphe. La valeur de l'opérateur laplacien au sommet i, notée $\Delta \boldsymbol{f}|_i$ est définie par :

$$\Delta \boldsymbol{f}|_i = \sum_{j \in \mathcal{V}_i} w_{ij} (f_i - f_j)$$

En manipulant cette formule, il apparaît que l'application de l'opérateur laplacien à la fonction $\boldsymbol{f}$ est équivalent à :

$$\Delta \boldsymbol{f} = (D - A)\boldsymbol{f}$$

Ce qui signifie que l'application de l'opérateur laplacien sur $\boldsymbol{f}$ revient à multiplier $\boldsymbol{f}$ à gauche par la matrice :

$$L = D - A \tag{3.9}$$

appelée matrice du laplacien, matrice laplacienne ou par abus de langage laplacien du graphe. Le laplacien ainsi défini est l'équivalent discret de l'opérateur de Laplace-Beltrami en géométrie différentielle et joue un rôle fondamental en théorie des graphes.

Des variantes parfois appelées ***laplaciens normalisés*** existent dans la littérature :

$$\hat{L} = D^{-1}L \qquad = I - D^{-1}A = I - P \tag{3.10}$$

$$\hat{L} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \tag{3.11}$$

Le laplacien normalisé de l'équation (3.10) fait explicitement intervenir la matrice de transition du processus de Markov associé à la marche aléatoire mais brise la symétrie du laplacien, d'où l'utilisation fréquente de la formule (3.11).

La matrice L est donc une matrice symétrique de taille $n \times n$. Elle a toujours au moins une valeur propre nulle. En effet si $\mathbf{1}$ est le vecteur de taille n dont toutes les valeurs sont 1, il est facile de voir que $L\mathbf{1} = 0$. La multiplicité de la valeur propre 0 est en fait un indicateur du nombre de composantes connexes du graphe [29]. Par ailleurs L est une matrice semi-définie positive [29].

La pseudo-inverse de Moore-Penrose généralise la notion d'inverse d'une matrice dans les cas où la matrice n'est pas inversible ou n'est pas carrée. Une description détaillée de la pseudo-inverse peut être trouvée dans [98]. Suivant les notations de [48], nous noterons L^+, la pseudo-inverse de la matrice L, et $l^+_{ij} = [L^+]_{ij}$. Dans le cas du laplacien, L^+ peut être calculée par la formule :

$$L^+ = (L - \mathbf{1}\mathbf{1}^T/n)^{-1} + \mathbf{1}\mathbf{1}^T/n \tag{3.12}$$

La démonstration de cette formule peut être trouvée dans [60].

La distance du temps de commutation peut s'exprimer en fonction des éléments de la matrice L^+. La démonstration de ce résultat basée sur l'équivalence du réseau électrique peut se trouver dans [73]. Pour une démonstration directement basée sur des considérations de marches aléatoires, on se référera à [48] où l'on peut trouver également les formules pour les temps et coût moyens de premier passage. Les formules pour ces deux quantités sont données à l'annexe B en même temps que de nouvelles formules pour d'autres quantités remarquables en lien avec les marches aléatoires.

La distance du temps de commutation en fonction des éléments de la matrice L^+ s'écrit :

$$n(i,j) = V_G(l^+_{ii} + l^+_{jj} - 2l^+_{ij}) \tag{3.13}$$

Comme la matrice L est semi-définie positive, $V_G L^+$ peut être vue comme le noyau générant la distance (au carré) $n(i,j)$. Soit $\boldsymbol{e}_i$ le vecteur de dimension n dont la i-ième composante vaut 1 et toutes les autres 0, l'équation (3.13) peut se réécrire :

$$n(i,j) = V_G(\boldsymbol{e}_i - \boldsymbol{e}_j)^T L^+ (\boldsymbol{e}_i - \boldsymbol{e}_j)$$

De plus, L^+ peut être décomposée en vecteurs propres et valeurs propres $L^+ = U\Lambda U^T$. Nous définissons la transformation :

$$\boldsymbol{x}_i = \sqrt{V_G}\Lambda^{\frac{1}{2}}U^T\boldsymbol{e}_i \tag{3.14}$$

On obtiens alors :

$$\begin{aligned} n(i,j) &= V_G(\boldsymbol{e}_i - \boldsymbol{e}_j)^T U \Lambda U^T (\boldsymbol{e}_i - \boldsymbol{e}_j) \\ &= V_G(\boldsymbol{e}_i - \boldsymbol{e}_j)^T U \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} U^T (\boldsymbol{e}_i - \boldsymbol{e}_j) \\ &= (\sqrt{V_G}\Lambda^{\frac{1}{2}} U^T (\boldsymbol{e}_i - \boldsymbol{e}_j))^T (\sqrt{V_G}\Lambda^{\frac{1}{2}} U^T (\boldsymbol{e}_i - \boldsymbol{e}_j)) \\ &= (\boldsymbol{x}_i - \boldsymbol{x}_j)^T (\boldsymbol{x}_i - \boldsymbol{x}_j) \end{aligned}$$

Clairement, la transformation de l'équation (3.14), projette les sommets du graphe dans un espace où la distance euclidienne est $\sqrt{n(i,j)}$. Pour cette raison, Fouss *et al.* [48] appellent $\sqrt{n(i,j)}$ la distance euclidienne du temps de commutation (DETC). D'autre part, en ne conservant dans Λ et U que les plus grandes valeurs propres et leurs vecteurs propres associés, il est possible d'obtenir un espace de dimension réduite dans lequel la distance euclidienne est approximativement la DETC.

Remarquons enfin que par construction, L et L^+ ont les mêmes vecteurs propres et que leurs valeurs propres respectives λ_i et λ_i^+ sont reliées par la relation :

$$\lambda_i^+ = \begin{cases} \dfrac{1}{\lambda_i} & \text{si } \lambda_i > 0 \\ 0 & \text{sinon.} \end{cases}$$

3.4.6 Classification par k-plus proches voisins sur graphe

Nous venons de voir qu'il est possible de projeter les sommets d'un graphe dans un espace euclidien de faible dimension où les distances correspondent approximativement à la DETC. Il est donc possible, dans cet espace, d'appliquer les méthodes habituelles de classification et notamment la classification par k-plus proches voisins.

3.5 Expériences

Nos expériences ont été réalisées principalement sur la base *Labeled Faces in the Wild* (LFW) présentée à la section 1.1.3.2 du chapitre 1.

3.5.1 Description de la méthode

La méthode LDCA puisqu'elle fonctionne à partir de contraintes sur les paires est particulièrement adaptée pour la base LFW. En effet, le protocole expérimental de LFW consiste à déterminer si des paires de visages représentent la même personne ou non.

Apprentissage d'une distance pertinente. La méthode que nous proposons consiste donc à d'abord apprendre à partir des données d'entraînement une mesure de distance pertinente grâce à la méthode LDCA. La méthode définit une projection dans un espace où les distances respectent les contraintes de similarité. L'ensemble des données d'entraînement comme de test peuvent donc y être projetées.

Apprentissage de la structure des données. Nous transformons ensuite le problème en un problème de classification binaire en construisant pour chaque paire de points $\boldsymbol{x}_i$ et $\boldsymbol{x}_j$, un unique descripteur $\boldsymbol{d}_{ij}$:

$$d_{ij_k} = |x_{i_k} - x_{j_k}|$$

Ces descripteurs sont censés être groupés dans le voisinage du point $\mathbf{0}$ pour les paires positives et éparpillées tout autour de ce point pour les paires négatives. Nous capturons donc cette structure à l'aide d'un graphe de k-plus proche voisinage. Comme vu dans la section 3.4, nous projetons ensuite les données (un point par paire) dans un espace où la distance euclidienne est une approximation de la distance du temps de commutation sur le graphe.

Classification par k-plus proches voisins Dans cet espace, nous effectuons une classification par k-plus proches voisins. L'étiquette d'une paire inconnue est alors définie comme l'étiquette majoritaire parmi ses k-plus proches voisins étiquetés.

3.5.2 Représentation des visages

La représentation des visages joue un rôle essentiel dans leur reconnaissance. Nous nous sommes inspirés de la méthode proposée dans [58], motivés par le fait que c'est cette représentation qui donnait au moment où nous avons réalisé les expériences, les meilleurs résultats sur la base LFW.

Il s'agit de descripteurs SIFT [83] calculés en 9 points caractéristiques, et ce à 3 échelles différentes (voir figure 3.5).

FIGURE *3.5: Les descripteurs de visage sont des descripteurs SIFT calculés sur 9 points-clé du visage à 3 échelles différentes.*

Les 9 points caractéristiques sont détectés par la méthode de Everingham *et al*. [45] et correspondent aux coins des yeux, de la bouche, aux bords des narines et au bout du nez. Cela permet de rendre la représentation invariante pour de petits changements de pose du visage.

Les descripteurs SIFT ont 128 composantes, ce qui conduit pour chaque visage à un ensemble de 3456 ($= 128 \times 3 \times 9$) attributs visuels.

Nous suivons également les conclusions des auteurs de [58] et prenons pour chaque composante sa racine carrée. La distance euclidienne calculée sur ces descripteurs modifiés correspond à la ***distance de Hellinger***, mieux adaptée que la distance euclidienne lorsque les données sont représentées sous forme d'histogrammes.

3.5.3 Évaluation de la méthode LDCA

Nous avons tout d'abord souhaité valider la méthode LDCA seule. Dans ce cas la classification se fait par simple seuillage de la distance entre paires d'images.

Dimensionnalité de l'espace de projection. Dans un premier temps, nous avons cherché à déterminer quelle était la meilleure dimensionnalité pour l'espace de projection. Nous avons pour cela

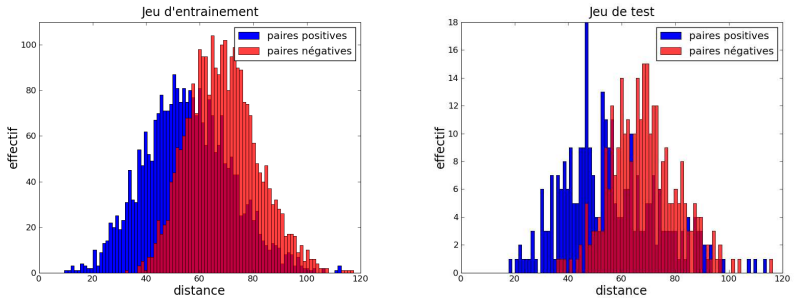


FIGURE *3.6: Distances entre les paires d'images positives et négatives dans l'espace d'entrée.*

conduit des expériences sur la `vue1` (cf. section 1.1.3.2). Nous avons ainsi observé que les performances augmentaient avec la dimensionnalité jusqu'à ce qu'elle atteigne 40. Pour des valeurs supérieures les performances sont les mêmes.

En revanche, les temps de calculs deviennent prohibitifs au-delà de 100 dimensions. Nous avons donc choisi la dimension de sortie la plus petite donnant de bons résultats, à savoir 40.

Notons que l'estimation de $\mathbf{M}$ est faite grâce à la méthode des gradients conjugués du module d'optimisation du paquetage SciPy ; cette estimation nécessite environ 2 minutes de calcul sur un PC standard.

Comparaison avec l'état de l'art. Le tableau 3.2 montre un aperçu des meilleurs résultats publiés pour la base LFW dans l'ordre chronologique, ainsi que les résultats que nous avons obtenus.

Nous constatons que la méthode LDCA donne un taux de bonne classification de $80.00 \pm 0.34\%$. Alors que la méthode LDML, qui donne les meilleurs résultats sur cette base, donne un score de $79.27 \pm 0.60\%$ (valeur donnée dans [58] pour une dimension de sortie de 35).

Visualisation de l'effet de l'apprentissage de distance. Nous pensons qu'il est intéressant de visualiser l'effet de la phase d'apprentissage de distance, et nous nous proposons de le faire au moyen d'histogrammes de distances.

La figure 3.6 montre la répartition des distances dans l'espace d'entrée pour les paires d'images respectivement positives et négatives, à la fois pour les données d'apprentissage et pour les données de test. Nous observons un fort recouvrement, ce qui explique que la distance dans l'espace d'origine ne permette pas une bonne séparation entre les paires ; le taux de paires bien classées n'est que de $68.50 \pm 0.5\%$ (résultat tiré de [58]).

La répartition des distances après entraînement est montrée figure 3.7 : nous constatons que la séparation des distances sur le jeu d'entraînement est parfaite (avec une grande marge) et que sur le jeu de test la séparation des paires positives et négatives est bien plus marquée. Il n'est donc pas surprenant de constater une très forte amélioration des performances.

3.5.4 Évaluation de la méthode par k-PPV sur graphe

3.5.5 Classification par k-PPV sur graphe

Il est donc ainsi possible de calculer le résultat d'une classification par k-plus proches voisins basée sur la distance dans ce domaine spectral réduit. Cette méthode est dénommée `LDCA+KNN-spec` dans les résultats, à distinguer de `LDCA+KNN-spat`, où les k-plus proches voisins sont déterminés dans le domaine spatial.

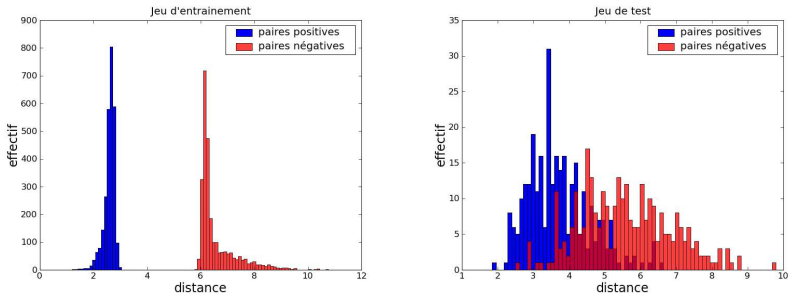


FIGURE *3.7: Distances entre les paires d'images du jeu d'entraînement et du jeu de test après apprentissage*

Nous pouvons finalement combiner les résultats de la classification basée sur la distance de Mahalanobis et ceux de la classification par k-PPV en utilisant un séparateur à vaste marge (SVM) à base radiale. Les données fournies à ce dernier seront pour chaque paire, la valeur p_n donnée par la méthode LDCA et la valeur moyenne $\hat{t}_n$ des étiquettes des k-PPV dans le domaine spectral.

3.5.6 Combinaison des représentations LDCA et k-PPV sur graphe

Notre intuition est que la distance donnée par la méthode LDCA et la topologie du graphe contiennent des informations complémentaires qu'il est intéressant de combiner.

Nous réalisons cette combinaison en entrainant un classfieur SVM-RBF qui reçoit en entrée la probabilité de l'étiquette t_n donné par LDCA et le nombre moyen d'exemples positifs entourant une paire dans le graphe (k-plus proche voisinage). Cette méthode est dénommée `LDCA+KNN-spec+RBF` dans les résultats.

Après avoir présenté différents algorithmes originaux, nous présentons ici leur validation expérimentale sur la base LFW.

Nous remercions les auteurs de [58] de nous avoir donné leurs fichiers de descripteurs, ce qui nous permet de faire une comparaison rigoureuse avec leur approche, sachant que les données de départ sont strictement identiques et que les améliorations de performances ne peuvent être attribuées qu'à la méthode de reconnaissance elle-même.

Dans cette seconde expérimentation, nous évaluons la méthode basée sur une classification par k-PPV sur graphe présentée section 3.4.

Nous supposons ici que les représentations des visages ont été projetées dans l'espace réduit défini section 3.3. Chaque paire est représentée par le vecteur différence entre les deux représentations des visages. Le problème se ramène donc dans ce cas à un problème de classification binaire.

Notons qu'une classification par k-plus proches voisins dans cet espace donne des résultats médiocres : 74.31% (voir table 3.2, ligne `LDCA+KNN-Spat` pour comparaison).

Évaluation quantitative. En revanche, la classification par k-plus proches voisins dans le domaine spectral (méthode `LDCA+KNN-Spec`) donne un taux de paires bien classées de 79.22 ± 0.29, ce qui montre que travailler dans un plongement spectral du graphe de voisinage permet bien de tirer parti d'une approche semi-supervisée. Toutefois, le score est légèrement inférieur à la méthode LDCA, probablement en raison du classifieur utilisé.

Visualisation des paires dans le domaine spectral. Une fois le graphe de voisinage construit, il est intéressant de regarder comment sont répartis les $\boldsymbol{d}_n$ dans le domaine spectral. La figure 3.8 représente la projection des données sur les deux premiers axes de la matrice L^+. Nous observons

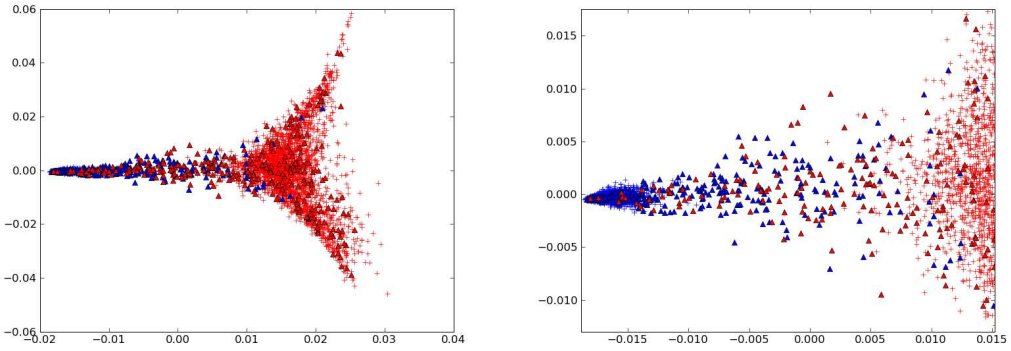


FIGURE *3.8: Répartition des descripteurs différence $\mathbf{d}_n$ dans le domaine spectral. Projection sur les deux premiers vecteurs propres de la matrice L^+. Les croix représentent les données d'entrainement et les triangles les données de test pour la première série de la $\mathtt{vue2}$. En bleu les paires positives, en rouge les paires négatives.*

notamment que la séparation entre les paires positives et négatives est préservée pour les données d'entrainement. La figure met également en évidence le *cluster* formé par les paires positives.

Méthode	score (%)
Nowak [90]	73.9 ± 0.5
MERL+Nowak [67]	76.2 ± 0.6
Hybrid descriptor-based [129]	78.5 ± 0.5
ITML [58]	$\mathbf{76.2 \pm 0.5}$
LDML [58]	$\mathbf{77.5 \pm 0.5}$
LDML-multi [58]	79.3 ± 0.6
LDCA	$\mathbf{80.0 \pm 0.3}$
LDCA+kNN-Spat	74.3 ± 1.8
LDCA+kNN-Spec	79.2 ± 0.3
LDCA+kNN-Spec+SVM-RBF	$\mathbf{80.4 \pm 0.4}$

TABLE *3.2: Aperçu des meilleurs résultats publiés et comparaison avec nos résultats.*

3.5.7 Évaluation de la méthode complète

La méthode complète consiste à classifier les paires au moyen d'un classifieur SVM-RBF prenant en entrée pour chaque paire la probabilité de l'étiquette $t_n = +1$ donnée par la méthode LDCA et la moyenne de ces étiquettes pour les plus proches voisins dans le graphe.

La combinaison de ces résultats donne au final, un score de $80.40 \pm 0.39\%$, significativement au-dessus de l'état de l'art (voir table 3.2) et des résultats précédents au moment où nous avons réalisé ces travaux.

3.6 Conclusions et perspectives

Dans cet article, nous avons proposé une méthode pour la classification de visages dans le contexte de vérification de visages. L'approche proposée combine les avantages d'un apprentissage de distance par analyse en composantes logistiques discriminantes et l'apprentissage semi-supervisé au moyen d'un graphe.

Au moment où ces travaux ont été réalisés, ils ont permis d'obtenir des résultats supérieurs à l'état de l'art, sur la très difficile base ***Labeled Faces in the Wild***.

Un article très récent [108] fait état de résultats encore meilleurs, par un découplage astucieux de la pose du visage et de la similarité. Une des pistes que nous allons suivre dans nos travaux futurs est justement d'intégrer cette idée dans notre approche.

CHAPITRE 4

Apprentissage de distance contrainte sur les paires

FORTS de l'expérience acquise avec LDCA, nous avons conçu une nouvelle méthode d'apprentissage conservant les points fort de LDCA tout en corrigeant ses faiblesses. Dans ce chapitre, nous présentons cette nouvelle méthode et montrons sa supériorité en matière de performance sur LDCA pour la vérification de visage. Afin de tester son caractère généraliste, nous appliquons également notre algorithme au problème de la ré-identification où la formulation des contraintes de similarité par paires n'est pas *a priori* la plus naturelle. Dans ce qui suit nous commençons par expliciter la démarche complète que nous avons suivie pour la conception de notre algorithme avant de décrire les expériences réalisées.

Sommaire

4.1 Pairwise Constrained Component Analysis (PCCA) [1]

4.1.1 Contraintes de similarité sur les paires

Nous nous plaçons dans le cas où l'on dispose d'un ensemble de points de données décrivant des objets. Nous disposons en plus d'informations concernant la similarité entre certaines paires d'objets.

1. Analyse en composantes contraintes par paires

Cela signifie que pour ces paires, nous savons si les objets qui les constituent sont similaires ou différents. Une bonne mesure de la distance entre ces objets devrait indiquer une faible distance entre des objets similaires et, comparativement, une grande distance entre objets différents.

4.1.2 Formulation mathématique

Soit $D_{\mathcal{X}}$ un ensemble de points $\boldsymbol{x}_i$ de $\mathcal{X}$ numérotés de 1 à $|D_{\mathcal{X}}|$, $\mathcal{P}$ un ensemble de paires (i, j) d'éléments distincts de $[1 \dots |D_{\mathcal{X}}|]$. et l_{ij} une étiquette prenant la valeur $l_{ij} = 1$ si les éléments (x_i, x_j) d'une paire sont similaires et $l_{ij} = -1$ s'ils sont différents.

Nous cherchons une application $\mathcal{A} : \mathcal{X} \mapsto \mathbb{R}^k$ qui « projette » les éléments de $\mathcal{X}$ dans un espace euclidien de dimension k (idéalement petite) tel que la distance euclidienne dans $\mathbb{R}^k$ reflète les contraintes sur les paires.

Nous définissons donc une mesure de distance entre les éléments de $\mathcal{X}$ paramétrée par l'application $\mathcal{A}$:

$$D^2_{\mathcal{A}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\mathcal{A}(\boldsymbol{x}_i) - \mathcal{A}(\boldsymbol{x}_j)\|^2_2 \tag{4.1}$$

Pour que cette distance reflète les contraintes sur les paires, nous devons choisir l'application $\mathcal{A}$ adéquate. Nous donnerons la forme explicite de $\mathcal{A}$ plus tard. Pour l'instant nous allons construire une fonction de coût adaptée à notre problème.

Nous voulons donc une mesure de distance telle que la distance entre les éléments d'une paire positive (c'est-à-dire une paire d'éléments similaires) soit petite comparativement à la distance entre les éléments d'une paire négative (une paire d'éléments différents). Pour cela nous voulons une fonction de coût c qui attribue un coût élevé à une paire si elle est positive et que la distance entre ses éléments est plus grande qu'un seuil s ou, respectivement, si la paire est négative et que la distance entre ses éléments est inférieure à ce même seuil. Notre fonction de coût globale E pourrait donc être de la forme :

$$E(\mathcal{A}) = \sum_{(i,j)\in\mathcal{P}} c\left(l_{ij}(D^2_{\mathcal{A}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - s)\right) \tag{4.2}$$

où la fonction $c(x)$ prend de faibles valeurs lorsque $x < 0$ et des valeurs élevées lorsque $x > 0$.

Un choix évident pour cette fonction serait la fonction rampe :

$$r(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Cependant cette fonction n'est pas dérivable en 0. On lui préférera donc la fonction de coût logistique généralisée :

$$\ell_\beta(x) = \frac{1}{\beta}\log(1 + e^{\beta x})$$

qui a l'avantage, pour des valeurs positives de β, d'être continue et dérivable partout. Par ailleurs, $\ell_\beta(x)$ a la propriété de tendre vers $r(x)$ pour les grandes valeurs de β :

$$\lim_{\beta\to\infty} \ell_\beta(x) = r(x)$$

La fonction $\ell_\beta(x)$ peut donc être vue comme une approximation lisse de la fonction rampe (voir figure 4.1).

La fonction de coût globale sera donc :

$$E(\mathcal{A}) = \sum_{(i,j)\in\mathcal{P}} \ell_\beta\left(l_{ij}(D^2_{\mathcal{A}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - s)\right) \tag{4.3}$$

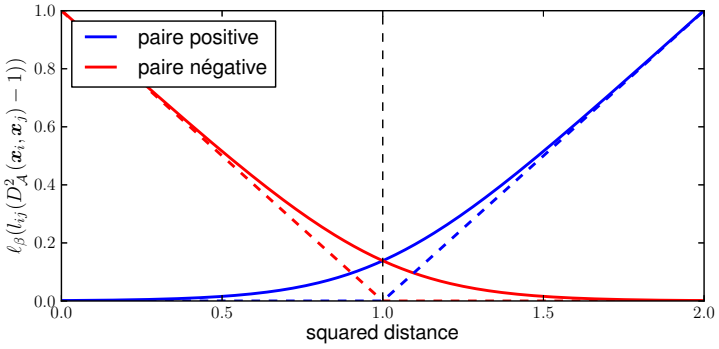


FIGURE *4.1: Fonction de coût ($\beta = 5$) en fonction de la distance pour une paire positive (ligne bleue) et une paire négative (ligne rouge). La fonction de coût analogue correspondant à la fonction rampe est représentée à titre de comparaison (ligne brisée).*

Échelle et seuil. Nous pouvons remarquer que, dans l'équation (4.3), le paramètre de seuil s, et le paramètre de lissage β jouent des rôles similaires. En effet, remplaçant $c(x)$ dans l'équation (4.2) par la fonction de coût logistique ℓ_β, il vient :

$$\begin{aligned}\ell_\beta(l_{ij}(D^2_{\mathcal{A}}(\boldsymbol{x}_i,\boldsymbol{x}_j)-s) &= \tfrac{1}{\beta}\log\left(1+e^{\beta l_{ij}(D^2_{\mathcal{A}}(\boldsymbol{x}_i,\boldsymbol{x}_j)-s)}\right)\\ &= \tfrac{1}{\beta}\log\left(1+e^{l_{ij}(D^2_{\frac{\mathcal{A}}{\sqrt{\beta}}}(\boldsymbol{x}_i,\boldsymbol{x}_j)-\beta s)}\right)\end{aligned}$$

La partie d dépendant de la distance dans l'exponentielle est donc :

$$d = D^2_{\frac{\mathcal{A}}{\sqrt{\beta}}}(\boldsymbol{x}_i,\boldsymbol{x}_j) - \beta s$$

Or comme $\mathcal{A}$ est une variable d'optimisation sa multiplication par $\frac{1}{\sqrt{\beta}}$ n'a pas d'effet sur la fonction distance finale car il est possible d'effectuer un simple changement d'échelle en posant $\mathcal{A}'(\boldsymbol{x}) = \frac{1}{\beta}\mathcal{A}(\boldsymbol{x})$. De plus, en posant $t = \beta s$, d se réécrit simplement :

$$d = D^2_{\mathcal{A}'}(\boldsymbol{x}_i,\boldsymbol{x}_j) - t$$

Ceci montre bien que β et s ont des rôles équivalents et il est donc inutile de conserver les deux paramètres. Par la suite nous fixons arbitrairement $s = 1$. Finalement notre fonction de coût globale est donc :

$$E(\mathcal{A};\beta) = \sum_{(i,j)\in\mathcal{P}} \ell_\beta\left(l_{ij}(D^2_{\mathcal{A}}(\boldsymbol{x}_i,\boldsymbol{x}_j)-1)\right) \tag{4.4}$$

Choix optimal du paramètre d'échelle β. En pratique, nous obtenons de meilleurs résultats en déterminant la valeur du paramètre β par validation croisée plutôt qu'en l'incluant comme variable d'optimisation. Pour comprendre pourquoi, observons la figure 4.2 Nous y voyons la fonction de coût pour une paire positive en fonction de la distance pour différentes valeurs du paramètre β. Lorsque le paramètre β augmente, la fonction ℓ_β tend vers la fonction rampe $r(x)$. Même si les trois courbes ont le même comportement asymptotique, elles sont relativement différentes autour du seuil. D'autre part, pour une distance donnée, le coût diminue lorsque le paramètre β augmente. Cela signifie que pour une application $\mathcal{A}$ fixe, augmenter la valeur de β fera toujours baisser la valeur de la fonction de coût global, tout en diminuant la contribution des paires correspondant à des distances proches

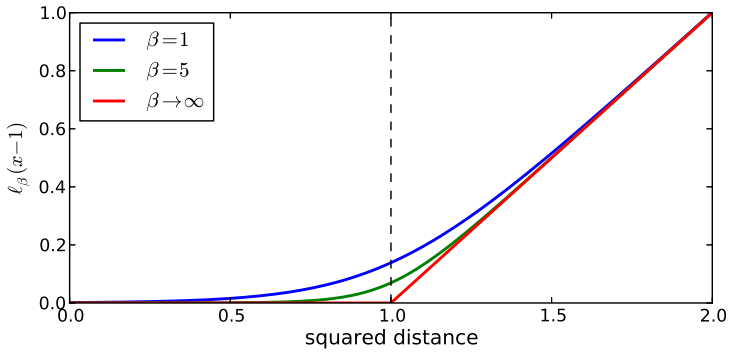


FIGURE *4.2: Influence du paramètre β autour du seuil.*

du seuil. Comme la classification *a posteriori* d'une nouvelle paire se fait justement en comparant la distance correspondante au seuil, les paires d'apprentissage dans cette zone sont particulièrement importantes. Il est donc impératif de choisir le paramètre β de façon optimale pour éviter le sur-apprentissage d'une part (le paramètre β est trop petit est la distance apprise est trop adaptée aux paires d'apprentissage) et pour maximiser le pouvoir discriminant de la métrique d'autre part. C'est pourquoi, contrairement à LDML [58], le paramètre contrôlant l'échelle β doit être déterminé par validation croisée et non optimisé en même temps que la métrique. En pratique dans toutes les expériences que nous avons réalisées une valeur de $\beta = 3$ semblait être toujours quasi-optimale.

4.1.3 Choix de l'application $\mathcal{A}$

Il nous reste à déterminer le choix de l'application $\mathcal{A}$. Dans la plupart des cas, l'espace des données $\mathcal{X}$ peut être vu comme un espace euclidien $\mathbb{R}^d$.

Cas linéaire. Un choix évident pour $\mathcal{A}$ est donc une transformation linéaire paramétrée par une matrice L. Celle-ci définit alors une projection dans un espace $\mathbb{R}^k$ par $\boldsymbol{x}' = L\boldsymbol{x}$. La mesure de distance devient alors :

$$D_L^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|L(\boldsymbol{x}_i - \boldsymbol{x}_j)\|_2^2 \tag{4.5}$$

Il s'agit ici du cas linéaire. Il est possible d'utiliser certaines applications non linéaires en appliquant le « truc du noyau ».

Cas non linéaire : le « truc du noyau ». Reparamétrisons L en $L = AX^T$ où X est la matrice de dimension $d \times |\mathcal{D}_\mathcal{X}|$ dont les colonnes sont les éléments de l'ensemble des données $\mathcal{D}_\mathcal{X}$.

Il vient :

$$\begin{aligned} D_L^2(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \|AX^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\|_2^2 \\ &= \|A(X^T\boldsymbol{x}_i - X^T\boldsymbol{x}_j)\|_2^2 \end{aligned}$$

Le terme $X^T\boldsymbol{x}_i = (\boldsymbol{x}_1^T\boldsymbol{x}_i, \ldots, \boldsymbol{x}_{|\mathcal{D}_\mathcal{X}|}^T\boldsymbol{x}_i)^T$ correspond donc au vecteur dont les éléments sont les produits scalaires de $\boldsymbol{x}_i$ avec chacun des éléments de $\mathcal{D}_\mathcal{X}$. Soit K la matrice $(|\mathcal{D}_\mathcal{X}| \times |\mathcal{D}_\mathcal{X}|)$ telle que $K_{ij} = \boldsymbol{x}_i^T\boldsymbol{x}_j$ et appelons $\boldsymbol{k}_i$ la i-ème colonne de K, on a $\boldsymbol{k}_i = X^T\boldsymbol{x}_i$. Notre mesure de distance s'écrit donc :

$$\begin{aligned} D_L^2(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \|A(\boldsymbol{k}_i - \boldsymbol{k}_j)\|_2^2 \\ &= D_A^2(\boldsymbol{k}_i, \boldsymbol{k}_j) \end{aligned} \tag{4.6}$$

Nous remarquons que, d'une part, la forme analytique de la distance en fonction des vecteurs $\boldsymbol{x}_i$ ou en fonction des vecteurs $\boldsymbol{k}_i$ reste inchangée. D'autre part, la matrice K a pour élément les produits scalaires de toutes les paires d'éléments de $\mathcal{D}_{\mathcal{X}}$, il s'agit donc par définition de la matrice de Gram des éléments de $\mathcal{D}_{\mathcal{X}}$. Nous pouvons donc appliquer le « truc du noyau » en remplaçant les éléments $K_{ij} = \boldsymbol{x}_i^T \boldsymbol{x}_j$ par $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ où $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ est un noyau défini positif.

En pratique la version à noyau du problème ne diffère donc de la version linéaire que par l'utilisation des colonnes $\boldsymbol{k}_i$ de la matrice noyau K à la place des colonnes $\boldsymbol{x}_i$ de la matrice des données X.

4.1.4 Optimisation

Nous considérons par la suite le cas linéaire de notre problème. C'est-à-dire que $\mathcal{X} = \mathbb{R}^d$ et $\mathcal{A} : \boldsymbol{x} \mapsto L\boldsymbol{x}$ où L est une matrice $(k \times d)$.

En injectant la fonction distance de l'équation (4.5) dans la fonction de coût globale (4.4) nous obtenons le problème de minimisation suivant :

$$\min_L E(L;\beta) = \sum_{(i,j)\in\mathcal{P}} \ell_\beta \left(l_{ij}(\|L(\boldsymbol{x}_i - \boldsymbol{x}_j)\|_2^2 - 1) \right) \tag{4.7}$$

Nous rappelons que β n'est pas une variable d'optimisation mais un paramètre dont la valeur optimale est à déterminer, par exemple, par validation croisée. Par ailleurs, la dimension k de l'espace de projection reste aussi à déterminer.

4.1.4.1 Convexité

Bien que le problème donné à l'équation (4.7) ne soit pas convexe, nous allons voir qu'il est pourtant possible d'optimiser efficacement ce problème avec des garanties théoriques quant à la convergence de l'algorithme.

Posons $M = L^T L$, la distance $D_L(\boldsymbol{x}_i, \boldsymbol{x}_j)$ devient :

$$D_L^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T M (\boldsymbol{x}_i - \boldsymbol{x}_j)$$

ce qui correspond à la distance de Mahalanobis paramétrée par M. Notre problème est donc équivalent au problème suivant :

$$\begin{array}{ll} \min\limits_M & E'(M;\beta) = \sum_{(i,j)\in\mathcal{P}} \ell_\beta \left(l_{ij}((\boldsymbol{x}_i - \boldsymbol{x}_j)^T M (\boldsymbol{x}_i - \boldsymbol{x}_j) - 1) \right) \\ \text{soumis à} & M \succeq 0 \\ & \operatorname{rang}(M) \leq k \end{array} \tag{4.8}$$

Sous cette forme, l'énergie $E'(M;\beta)$ est une fonction convexe de M. La matrice M doit être semi-définie positive ($M \succeq 0$) et de rang inférieur ou égale à k pour être factorisable en $M = L^T L$ où L est de dimension $k \times d$.

La contrainte $M \succeq 0$ est convexe mais pas la contrainte de rang. En relaxant cette contrainte nous obtenons le problème d'optimisation suivant :

$$\begin{array}{ll} \min\limits_M & E'(M;\beta) = \sum_{(i,j)\in\mathcal{P}} \ell_\beta \left(l_{ij}((\boldsymbol{x}_i - \boldsymbol{x}_j)^T M (\boldsymbol{x}_i - \boldsymbol{x}_j) - 1) \right) \\ \text{soumis à} & M \succeq 0 \end{array} \tag{4.9}$$

qui lui est convexe. Il s'agit donc d'un problème d'optimisation convexe dans le cône des matrices semi-définies positives. Or il est possible de montrer que résoudre le problème de l'équation (4.9) par rapport à M est équivalent à résoudre le problème de l'équation (4.7) à condition que le rang de l'optimum M^* soit inférieur ou égal à k. Une démonstration de résultat fondamental peut être trouvée dans [70].

4.1.4.2 Algorithme d'optimisation utilisé

Dans [70], en même temps que la démonstration du caractère optimal des solutions factorisées, Journée *et al.* proposent un algorithme général pour résoudre les problèmes d'optimisation sur le cône des matrices semi-définies positives, dont fait partie notre algorithme. Cette approche, quoique théoriquement fondée et possédant des propriétés de convergence quadratique, s'avère n'être pas forcément la plus efficace en pratique. En effet, leur algorithme est une méthode du deuxième ordre et nécessite, entre autre par conséquent le calcul répété de la matrice hessienne du problème, ce qui, pour des matrices de grande taille, peut se révéler particulièrement coûteux en temps de calcul. Nous présentons succinctement cet algorithme à l'annexe C, où l'on trouvera également une discussion plus détaillée sur les difficultés de la mise en pratique de cette approche avec PCCA.

Notre algorithme de prédilection est, en définitive, une simple descente de gradient avec à chaque itération une recherche linéaire basée sur la méthode de Brent (voir par exemple [95] pour une description détaillée). Celle-ci combine la méthode de bissection, la méthode du sécant et l'interpolation quadratique pour trouver le minimum local d'une fonction à une dimension. Par rapport à la méthode du gradient conjugué utilisée pour LDCA au chapitre précédent, cette méthode de calcul a l'avantage d'effectuer moins d'estimations du gradient. Celui-ci étant plutôt lourd à calculer, ceci nous permet une convergence plus rapide malgré des propriétés théoriques moins avantageuses.

Le gradient de la fonction d'énergie E de l'équation (4.7) par rapport à la matrice de projection L est donné par :

$$\nabla_L E(L;\beta) = L \sum_{(i,j)\in\mathcal{P}} \mathcal{L}'_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \tag{4.10}$$

où, par souci de concision, on a posé $\mathcal{L}'_{ij} = 2l_{ij}\ell'_\beta\left(l_{ij}(\|L(\boldsymbol{x}_i - \boldsymbol{x}_j)\|^2 - 1)\right)$.

Par ailleurs, le choix de la dimension de l'espace de projection k peut être déterminé classiquement par validation croisée. Le surplus de calcul engendré par les multiples essais peut être compensé par la parallélisation de ces derniers. Une autre approche possible consiste à effectuer une Analyse en Composante Principale (ACP) sur les données et à choisir le paramètre k optimal. Les vecteurs de projection ainsi trouvés peuvent par ailleurs fournir un bon point de départ pour notre algorithme, accélérant ainsi la convergence.

4.1.4.3 Cas des noyaux et préconditionnement

Lorsque les échantillons de données $\boldsymbol{x}_i$ sont remplacés par les colonnes correspondantes $\boldsymbol{k}_i$ de la matrice noyau K, le problème d'optimisation de l'équation (4.7) reste formellement inchangé puisque si l'on pose $L = AX^T$ et $K = X^TX$, on a :

$$D^2_L(\boldsymbol{x}_i, \boldsymbol{x}_j) = D^2_A(\boldsymbol{k}_i, \boldsymbol{k}_j)$$

Cependant, l'application naïve de l'algorithme présenté plus haut conduit à des vitesses de convergence particulièrement basses. Ce phénomène a déjà été observé dans le cas des séparateurs à vaste marge (SVM). Dans [26], Olivier Chapelle attribue ce problème de convergence lente à un mauvais conditionnement de la matrice hessienne. Alors que souvent, ce problème est évité en résolvant le problème dual, Olivier Chappelle propose de simplement pré-conditionner le gradient par l'inverse de la matrice noyau. Dans notre cas, la même solution est applicable. Nous allons par ailleurs montrer qu'utiliser le gradient ainsi pré-conditionné correspond dans le cas linéaire à choisir comme direction de descente la direction donnée par le gradient par rapport à L plutôt que par rapport à A.

Pour voir ceci, nous écrivons l'équation de la descente de gradient dans le cas linéaire :

$$L^{t+1} \leftarrow L^t - \eta\nabla_L E(L^t; X) \tag{4.11}$$

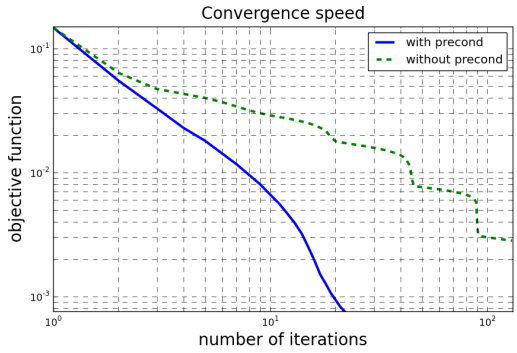


FIGURE *4.3: Vitesse de convergence de l'algorithme PCCA en mode noyau avec (ligne continue) et sans (ligne pointillée) préconditionnement sur la base VIPeR (voir la section expérimentale 4.2). Remarquez l'échelle log-log.*

où L^t représente la valeur de la matrice de projection L à l'itération t, $\eta > 0$ est le pas d'adaptation et où la notation $E(.;X)$ met en évidence que X est un paramètre fixe de la fonction de coût. Nous obtenons donc :

$$L^{t+1} \leftarrow L^t - \eta L^t \sum_{(i,j)\in\mathcal{P}} \mathcal{L}'_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T$$

en substituant $L^t = A^t X^T$ nous trouvons :

$$A^{t+1}X^T \leftarrow A^t X^T - \eta A X^T \sum_{(i,j)\in\mathcal{P}} \mathcal{L}'_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T$$

supposons que K est inversible, en multipliant tous les termes à droite par XK^{-1}, il vient :

$$\begin{aligned}
A^{t+1}X^TXK^{-1} &\leftarrow A^tX^TXK^{-1} &&- \eta A^t \sum_{(i,j)\in\mathcal{P}} \mathcal{L}'_{ij} X^T(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T XK^{-1} \\
A^{t+1}KK^{-1} &\leftarrow A^tKK^{-1} &&- \eta A^t \sum_{(i,j)\in\mathcal{P}} \mathcal{L}'_{ij} (\boldsymbol{k}_i - \boldsymbol{k}_j)(\boldsymbol{k}_i - \boldsymbol{k}_j)^T K^{-1} \\
A^{t+1} &\leftarrow A^t &&- \eta A^t \sum_{(i,j)\in\mathcal{P}} \mathcal{L}'_{ij} (\boldsymbol{k}_i - \boldsymbol{k}_j)(\boldsymbol{k}_i - \boldsymbol{k}_j)^T K^{-1}
\end{aligned}$$

d'où

$$A^{t+1} \leftarrow A^t - \eta \nabla_A E(A^t;K)K^{-1} \tag{4.12}$$

Formellement, les équations (4.11) et (4.12) ne diffèrent que par l'ajout du K^{-1} au gradient, ce qui peut être vu, comme indiqué plus haut, comme un pré-conditionnement du gradient.

Il est intéressant de remarquer, que contrairement à ce que l'on pourrait penser le pré-conditionnement par K^{-1} n'entraîne pas de surcoût de calcul, au contraire. En effet, nous pouvons remarquer que $\boldsymbol{k}_i K^{-1} = \boldsymbol{e}_i$ où $\boldsymbol{e}_i$ est le vecteur de la base canonique dont la i-ème composante vaut 1 et toutes les autres 0. Par conséquent l'équation (4.12) peut se réécrire :

$$A^{t+1} \leftarrow A^t - \eta \sum_{(i,j)\in\mathcal{P}} (\mathcal{L}'_{ij} A^t(\boldsymbol{k}_i - \boldsymbol{k}_j))(\boldsymbol{e}_i - \boldsymbol{e}_j)^T$$

Cela signifie que pour chaque paire (i,j), nous ne devons mettre à jour que deux colonnes de la matrice A au lieu de la matrice complète dans le cas classique. Ceci a pour effet bénéfique d'alléger d'autant la charge de calcul.

4.2 Expériences

Nous avons validé notre approche sur deux tâches de vision par ordinateur : la vérification de visage et la ré-identification de personnes. Pour ces deux méthodes, avoir une bonne métrique est crucial. Les expériences sont menées sur deux bases publiques : la base Labeled Faces in the Wild (LFW) (voir la section 1.1.3.2, page 5) pour la vérification de visages et la base View-point Invariant Person Re-identification (VIPeR) (voir la section 1.1.3.4, page 8) pour la ré-identification de personnes. Cette deuxième base nous permet de comparer les performances de PCCA à celles d'algorithmes nécessitant plus d'information pour leur apprentissage.

4.2.1 Vérification de visages

Pour la vérification de visage sur LFW, les scores reportés représentent le taux de bonne classification des paires. Nous avons testé l'évolution des performances de l'algorithme en fonction de deux critères :

- la quantité de données d'entraînement utilisée ;
- le noyau utilisé.

Notre but est en effet de montrer que notre méthode 1) généralise mieux et 2) bénéficie de l'utilisation des noyaux non linéaires.

4.2.1.1 Variation du nombre de paires d'entraînement

Pour étudier l'évolution des performances de notre algorithme en fonction de la quantité de données d'entraînement (nombre de paires), nous avons dû recourir au *paradigme non restreint* de LFW. En effet, la `vue2` de LFW comprend 10 sous-ensembles avec 600 paires de visages chacun. L'utilisation sans modification de ces 10 sous-ensembles avec le protocole standard de LFW consistant à calculer les scores comme le score moyen de la validation croisée à 10 plis est appelée par les auteurs de la base le *paradigme restreint* car il n'utilise pas l'information d'identité des personnes représentées dans la base.

La validation croisée utilise 9 sous-ensembles pour l'entraînement et 1 sous-ensemble pour le test, chaque sous-ensemble étant utilisé comme ensemble de test à tour de rôle. Dans le paradigme non restreint, seuls les sous-ensembles d'entraînement sont modifiés, le sous-ensemble de test restant lui inchangé. Par ailleurs, les nouvelles paires sont créées uniquement à partir des personnes présentes dans chacun des sous-ensembles individuellement. Ceci garantit la conservation de la propriété qu'une même personne ne peut être représentée dans deux-sous ensembles.

Enfin, pour évaluer notre algorithme avec un nombre plus petit de paires d'entraînement, nous avons simplement sélectionné un sous-ensemble des paires existantes dans chacun des sous-ensembles.

4.2.1.2 Les noyaux utilisés

Nous avons testé trois noyaux différents.

Le plus simple des noyaux utilisés est le noyau de Bhattacharyya :

$$k_{\mathrm{Bhat}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} \sqrt{x_i y_i}$$

L'utilisation de ce noyau, plus adapté aux histogrammes que la distance euclidienne, est simple puisqu'il suffit de prendre pour chaque composante sa racine carrée. Le noyau de Bhattacharyya appliqué sur les histogrammes non transformés est alors équivalent au noyau linéaire sur les histogrammes

transformés. C'est pourquoi, dans les résultats qui suivent, les méthodes linéaires appliquées sur les histogrammes transformés sont indiquées par le suffixe « -sqrt ». On notera que la transformation qui consiste à prendre la racine carré de chaque composante d'un histogramme peut être comprise comme la fonction de «re-description» permettant de projeter l'histogramme dans l'espace de Hilbert associé au noyau de Bhattacharyya (ou espace de re-description). Dans cet espace la distance euclidienne est simplement la distance de Hellinger :

$$D^2_{\mathrm{Hell}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} (\sqrt{x_i} - \sqrt{y_i})^2$$

La version à noyau de notre algorithme a, quant à elle, été appliquée avec le noyau du χ^2 :

$$k_{\chi^2}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} \frac{x_i y_i}{x_i + y_i}$$

ce qui correspond à la distance du χ^2, très utilisée pour mesurée la distance entre histogrammes :

$$D^2_{\chi^2}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} \frac{(x_i - y_i)^2}{x_i + y_i}$$

Nous avons également utilisé le noyau que nous noterons RBF-χ^2 [2] :

$$k_{\text{RBF-}\chi^2}(\boldsymbol{x}, \boldsymbol{y}) = e^{-\alpha D^2_{\chi^2}(\boldsymbol{x}, \boldsymbol{y})}$$

ce qui peut être vu comme une généralisation du noyau gaussien pour une mesure de distance différente de la distance euclidienne. Notons que ce noyau introduit un nouveau paramètre α qui contrôle la dispersion du noyau.

4.2.1.3 Choix des paramètres

Les paramètres libres de l'algorithme sont choisis pour être optimaux sur la `vue1`. On trouve donc :

- la dimension de sortie $= 40$
- $\beta = 3$
- $\alpha = 1.0$ (pour le noyau RBF-χ^2)

Enfin, les résultats reportés sont la moyenne des scores obtenues par validation croisée sur les dix sous-ensembles de la `vue2` (voir la section 1.1.3.2 page 5 pour une description détaillée).

Nous comparons notre méthode avec ITML, LDML et LDCA avec les mêmes descripteurs SIFT qu'au chapitre précédent.

Le tableau 4.1 présente les résultats que nous avons obtenus et les compare à ceux obtenus avec ITML, LDML et LDCA [3] pour les mêmes descripteurs SIFT que ceux utilisés au chapitre précédent (voir page 48).

La première colonne indique les résultats obtenu avec 600 paires d'entraînement par sous-ensemble (soit 5400 paires d'entraînement au total). Dans les mêmes conditions que pour les trois autres méthodes, PCCA atteint un score de 82.2% de bonne classification ce qui est significativement au-dessus des scores que nous avions obtenus avec LDCA qui déjà surpassait ITML et LDML. En mode noyau, les résultats s'améliorent encore pour atteindre un score maximal de 83.8% avec le noyau RBF-χ^2, ce qui montre l'avantage d'une modélisation non-linéaire des données.

Avec 10000 paires par sous-ensemble, les scores de toutes les méthodes s'en trouve nettement améliorés et LDML et PCCA obtiennent des scores presque équivalents soit respectivement 83.2% pour LDML et 83.3% pour PCCA. La méthode ITML reste moins performante que les deux autres.

Une fois encore, l'utilisation des noyaux permet d'améliorer les scores avec un taux maximum de 85.0% de bonne classification pour le noyau RBF-χ^2.

2. RBF signifie *Radial Basis Function* c'est-à-dire fonctions à base radiale
3. Dans le cas de LDCA, nous n'avons pas effectué les expériences avec un nombre plus élevé de paires d'entraînement.

Metric	Number of training pairs 600	10,000
ITML-sqrt [58]	76.2 $\pm$ 0.5	80.5 $\pm$ 0.5
LDML-sqrt [58]	77.5 $\pm$ 0.5	83.2 $\pm$ 0.4
LDCA-sqrt [85]	80.0 $\pm$ 0.3	–
PCCA-sqrt	82.2 $\pm$ 0.4	83.3$\pm$ 0.5
PCCA-χ^2	83.1 $\pm$ 0.5	84.3 $\pm$ 0.5
PCCA-χ^2_{RBF}	**83.8** $\pm$ **0.4**	**85.0** $\pm$ **0.4**

TABLE *4.1: Taux de bonne classification sur LFW avec 600 and 10000 paires d'entraînement par sous-ensemble. Voir le texte pour les détails.*

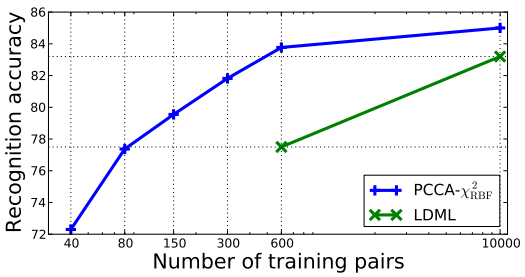


FIGURE *4.4: Précision de la reconnaissance avec les méthodes PCCA-χ^2_{RBF}, en fonction du nombre de paires d'entraînement utilisées.Les résultats obtenus avec LDML [58] avec 600 et 10000 pairs sont également rapportés dans le graphique.*

La figure 4.4 montre plus précisément l'évolution du score de PCCA avec le noyau RBF-χ^2. À titre de comparaison, les résultats obtenus avec LDML pour 600 et 10000 paires d'entraînement par sous-ensemble ont été ajoutées sur la figures. On y voit l'effet bénéfique de l'augmentation du nombre de paires d'entraînement sur le score avec un effet de saturation pour un nombre élevé de paires. Il est aussi intéressant de constater PCCA peut égaler le score obtenu avec LDML pour 600 paires avec seulement 80 paires et le score obtenu par LDML pour 10000 paires avec seulement 500 paires environ.

L'ensemble de ces résultats confirme que le pouvoir de généralisation de PCCA est bien supérieur à celui de LDML et que l'utilisation de noyaux non-linéaires peut grandement améliorer les résultats. Il faut de plus ajouter que PCCA, comme LDCA, peut traiter les données directement dans leur espace de représentation alors qu'une réduction de dimensions par PCA est nécessaires pour appliquer LDML et ITML.

Par ailleurs, nos résultats sont tout à fait compétitifs avec d'autres méthodes de l'état de l'art sur LFW, sachant que nous utilisons seulement 9 descripteurs SIFT (à trois échelles) pour représenter les visages. En effet, le descripteur LE de [24], bien plus élaboré et spécifiquement conçu pour les visages donne seulement $83.4 \pm 0.6\%$ en mode restreint. Dans [87], un score de $85.6 \pm 0.5\%$ est rapporté en utilisant des LBP (*Local Binary Patterns*) finement échantillonnés et combinés à une mesure de similarité en cosinus apprise.

4.2.2 Ré-identification

La ré-identification consiste à mettre en correspondance des personnes observées par des caméras sous des angles différents. Pour cette tâche nous utilisons la base VIPeR, présentée à la section

1.1.3.4 page 8.

4.2.2.1 Protocole expérimental

Contrairement à la base LFW, il n'y a pas de protocole standard fourni avec la base VIPeR. Nos expériences, cependant, utilisent le même protocole expérimental que dans [55] et [139].

La base contient les photographies de 632 personnes capturées par deux caméras différentes. On utilise p personnes choisies aléatoirement pour constituer le jeu de test et les personnes restantes pour l'entraînement. Le score est calculé à partir des *courbes de correspondances cumulées* (notées CMC pour *Cumulative Matching Curves*) [139].

Lors de la phase de test, un ensemble d'images appelé *galerie* est constitué en utilisant certaines des images disponibles pour chaque personne. Les images restantes sont utilisées comme des sondes (*probes* en anglais). Pour chaque image sonde, les images de la galerie sont triées en fonction de leur degré de similarité à la sonde. Les correspondances cumulées au rang r, donnent le taux moyen de présence de l'image correspondante de la galerie pour l'ensemble des images sondes. Le tracé de la courbe représentant ce taux pour différentes valeurs de r donne la courbe dite CMC [139]. Toute l'opération est répétée 10 fois et les scores pour chaque valeur de r sont rapportés comme la moyenne sur les 10 essais.

Le score au rang $r = 1$ caractérise donc la capacité du système à retrouver le bon correspondant. Cependant, les scores pour des rangs plus élevés sont également intéressants car, en pratique, il est possible de laisser à un opérateur humain la décision finale de mise en correspondance, et dans ce cas un petit nombre d'images peut lui être présenté. Un score élevé pour des faibles valeurs de r indique que le système est capable de présenter à l'utilisateur un nombre limité d'images parmi lesquelles la bonne correspondance a de fortes probabilité de se trouver.

4.2.2.2 Variations des paramètres

Les noyaux. Comme dans l'expérience précédente trois noyaux sont testés : le noyau de Bhattacharyya (sous les mêmes modalités), le noyau χ^2 et le noyau RBF-χ^2.

L'évolution des performances par rapport à la quantité de données d'entraînement disponible est cette fois testée selon deux critères : le nombre de personnes représentées dans le jeu d'entraînement, et le nombre de contraintes utilisées.

Le nombre de personnes représentées. Les algorithmes de reconnaissance que nous testons s'appliquent dans un cadre où les personnes utilisées pour le test ne sont pas connues au moment de l'entraînement. Pour obtenir une méthode capable de généraliser correctement, il est préférable qu'un nombre important de sujets aient été utilisés pour l'entraînement afin d'éviter des problèmes de sur-spécialisation de l'algorithme. Dans nos expériences, le paramètre p indique le nombre de personnes présentes dans la base de test. Le nombre de personnes dans la base d'entraînement est donc $(632 - p)$.

Le nombre de contraintes. La base est conçue de sorte qu'il existe deux occurrences de chaque personne présente. Cela signifie, du point de vue des contraintes possibles sur les paires d'images, qu'il n'y a qu'une contrainte positive possible par personne. Par contre une contrainte négative peut être créée pour chaque paire d'images représentant des personnes différentes. Dans les résultats qui suivent le paramètre n^- indique le nombre de contraintes négatives utilisées pour chaque contrainte positive.

4.2.2.3 Les autres méthodes

Les résultats présentés ici, se basent sur les travaux de Zheng *et al*. qui, dans [139] proposent une méthode appelée PRDC (pour *Probabilistic Relative Distance Comparison*) et proposent une comparaison assez complète avec les principaux algorithmes d'apprentissage de distance. Parmi les algorithmes testés, nous retiendrons les scores rapportés pour 3 d'entre eux : PRDC, MCC (*Maximally Collapsing Classes*) [51], et ITML (*Information Theoretic Metric Learning*) [40]. Ces trois méthodes ont déjà été évoquées dans les chapitres précédents, mais nous en donnons ou redonnons une description plus technique.

PRDC. La méthode PRDC met en jeu une formulation assez semblable à celle de LDML et basée sur la fonction logistique $\sigma(x) = (1 + e^{-x})^{-1}$ pour modéliser un critère de probabilité. Elle utilise un ensemble de points pour lesquels un point de la même classe ainsi qu'un point d'une classe différente sont connus. La distance entre les deux points similaires doit être plus petite que la distance entre le point de référence et le point appartenant à une autre classe. La distance est paramétrisée de la même façon que dans PCCA. La différence entre ces deux distances est injectée dans la fonction logistique ce qui permet de modéliser la probabilité que les deux points soient correctement ordonnés par rapport au point de référence. Si cette méthode présente des similarités avec PCCA, LDCA ou LDML, une différence majeure réside dans le fait que les contraintes sont données pour des triplets de points ou autrement dit sous forme de paires de contraintes : une contrainte positive étant toujours associée à une contrainte négative. Ceci signifie, par exemple, que PRDC n'est pas applicable pour la vérification de visages telle que proposée dans LFW. Une autre différence, pouvant être à l'avantage de PRDC, réside dans le fait que les contraintes sur les distances ne sont prises en compte que de manière relative et non absolue (avec un seuil sur les distances) comme dans le cas de PCCA. Dans les expériences de Zheng *et al*, c'est la méthode qui donne les meilleurs résultats.

MCC. La méthode MCC a été proposée à l'origine pour améliorer les résultats de classification par k-plus proches voisins. Cette méthode nécessite de connaître les labels de classe pour tous les points d'entraînement et a pour but de trouver une mesure de distance (modélisée par une distance de Mahalanobis) pour laquelle les éléments d'une classe se regroupent sous forme de partitions (*clusters* en anglais) dans l'espace. Pour cela, la probabilité que deux points appartiennent à la même classe est modélisée par une exponentielle décroissante de la distance entre les points. La méthode tente ensuite de minimiser la divergence de Kullback-Leibler entre la distribution effective et la distribution idéale. Nous avons décidé de rapporter les résultats obtenus par Zheng *et al* avec cette méthode car il s'agit de la méthode utilisant les informations de classes qui donne les meilleurs résultats.

ITML. Dans la méthode ITML, la matrice de Mahalanobis associée à la mesure de distance est considérée comme la paramétrisation d'une distribution normale multivariée. La méthode effectue la minimisation d'une mesure de la divergence entre cette distribution et une distribution de référence soumise à un jeu de contraintes sur les distances entre des paires de points correspondant à des paires positives et négatives. La formulation des contraintes est donc similaires à celle de PCCA et nous avons rapporté les résultats avec cette méthode pour faire le lien avec les résultats rapportés dans l'expérience précédente.

4.2.3 Résultats expérimentaux

Le tableau 4.2 présente les résultats obtenus avec PRDC, MCC et ITML tels qu'ils apparaissent dans [139] ainsi que les résultats obtenus avec notre méthode pour les différents noyaux et différentes valeurs du nombre de contraintes négatives n^{-1} par contrainte positive ($n^- = 1$ et $n^- = 10$). Les scores sont donnés à différents rangs r. Le tableau du haut indique les scores pour un nombre de

personnes dans le jeu de test $p = 316$ (donc 316 personnes dans le jeu d'entraînement) alors que le tableau du bas regroupe les mêmes scores pour un nombre de personnes dans le jeu de test $p = 532$ (100 personnes dans le jeu d'entraînement).

$p = 316$	$r = 1$	$r = 5$	$r = 10$	$r = 20$
PRDC $n^- = 1$ [139]	15.66	38.42	53.86	70.09
MCC $n^- = 631$ [139]	15.19	41.77	57.59	73.39
ITML $n^- = 631$ [139]	11.61	31.39	45.76	63.86
PCCA-sqrt $n^- = 1$	13.48	34.84	49.43	67.18
PCCA-χ^2 $n^- = 1$	13.67	35.22	49.93	68.20
PCCA-χ^2_{RBF} $n^- = 1$	17.02	43.26	58.67	76.36
PCCA-sqrt $n^- = 10$	17.28	42.41	56.68	74.53
PCCA-χ^2 $n^- = 10$	17.28	43.64	59.68	76.04
PCCA-χ^2_{RBF} $n^- = 10$	**19.27**	**48.89**	**64.91**	**80.28**

$p = 532$	$r = 1$	$r = 5$	$r = 10$	$r = 20$
PRDC $n^- = 1$ [139]	9.12	24.19	34.40	48.55
MCC $n^- = 199$ [139]	5.00	16.32	25.92	39.64
ITML $n^- = 199$ [139]	4.19	11.11	17.22	24.59
PCCA-sqrt $n^- = 1$	7.34	21.02	31.30	45.37
PCCA-χ^2 $n^- = 1$	7.03	20.32	30.86	45.71
PCCA-χ^2_{RBF} $n^- = 1$	7.61	22.42	33.40	48.42
PCCA-sqrt $n^- = 10$	8.44	24.34	35.62	50.07
PCCA-χ^2 $n^- = 10$	7.95	24.23	35.73	50.45
PCCA-χ^2_{RBF} $n^- = 10$	**9.27**	**24.89**	**37.43**	**52.89**

TABLE *4.2: Courbes Cumulative Match Characteristic sur la base VIPeR avec $p = 316$ et $p = 532$ personnes dans le jeu de test, pour r images retournées.Voir le texte pour les détails.*

Comme dans l'expérience précédente, les performances ont tendances à s'améliorer avec la complexité du noyau. Les meilleurs scores étant obtenus pour le noyau RBF-χ^2. L'augmentation du nombre de contraintes négatives influe également positivement sur les performances jusqu'à un certain point.

La figure 4.5 montre l'évolution du score pour $r = 20$ en fonction du paramètre n^-, clairement jusqu'à environ $n^- = 10$, les performances augmentent. Au-delà elles ont tendance à stagner, voir à régresser. Ceci tend à montrer que seul un nombre limité de contraintes sont utiles. Lorsque trop de contraintes sont utilisées, l'algorithme a tendance à sur-apprendre la distribution des données d'entraînement, au détriment du pouvoir de généralisation.

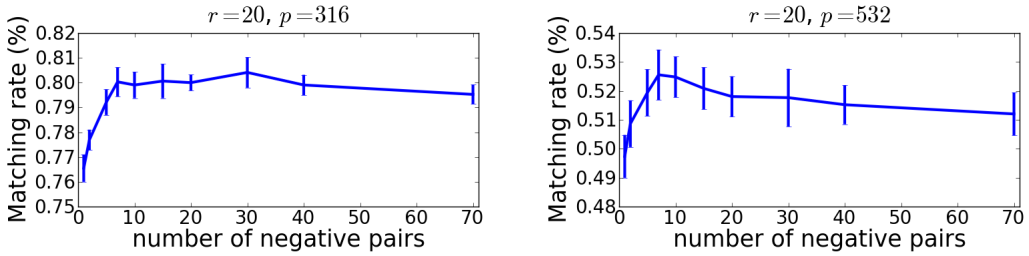


FIGURE *4.5: Courbe CMC de PPCA en fonction du nombre de paires négatives utilisées pour l'entraînement, base VIPer au rang $r = 20$, pour $p = 316$ et $p = 532$.*

Comme dans l'expérience précédente ITML donne systématiquement des résultats médiocres comparés aux autres méthodes. PCCA surpasse MCC dans toutes les configurations pour $p = 532$, ce qui tend à montrer un meilleur pouvoir de généralisation de la part de PCCA par rapport à MCC. Dans les cas les plus défavorables, PRDC semble donner de meilleurs résultats et n'est surpassé par PCCA que lorsque $n^- = 10$ avec le noyau RBF-χ^2

En revanche, dans le cas où $p = 316$, dès $n^- = 1$ PCCA donne de meilleurs résultat avec le noyau RBF-χ^2 que MCC et PRDC et avec $n^- = 10$, ses résultats sont supérieurs pour tous les noyaux (sauf avec le noyau Bhattacharyya pour $r = 10$ où PRDC surpasse légèrement PCCA).

La figure 4.6 présente les courbes CMC pour PRDC, CMC et PCCA avec le noyau RBF-χ^2 et pour $n^- = 10$. Dans le cas $p = 316$, nous avons par ailleurs ajouté la comparaison avec la méthode récente de Dikmen *et al* [41], LMNN-R. Il s'agit d'une variante de *Large Margin Nearest Neighbors* [127] avec un mécanisme additionnel de rejet. Les résultats publiés pour cette méthode n'étant pas disponibles sous forme numérique, la comparaison est difficile. Sur la figure 4.6, nous voyons que PCCA et LMNN-R donnent des résultats très similaires.

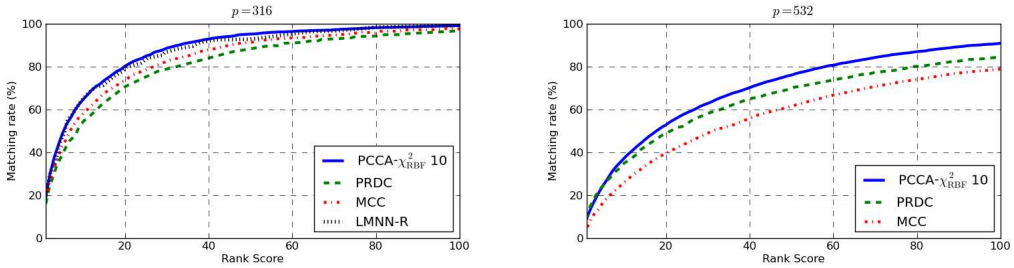


FIGURE *4.6: Base VIPeR : courbe CMC en fonction de r, pour PCCA et deux autres méthodes de l'état de l'art.*

4.3 Conclusion

La méthode PCCA a été spécialement conçue pour dépasser les limites de la méthode LDCA présentée au chapitre précédent. Les expériences que nous avons réalisées sur LFW montrent clairement que ce but est atteint. PCCA offre une méthode robuste lorsque les données d'entraînement sont en quantité limitée et capable de traiter des données de grandes dimensions. Nous avons également fourni une solution algorithmique pour appliquer efficacement la méthode en mode noyau ce qui améliore significativement les performances de l'algorithme.

Les expériences en ré-identification de personnes sur la base VIPeR, nous ont permis de confronter PCCA à des algorithmes moins généraux mais potentiellement plus adaptés à la tâche. Ainsi, alors que PCCA se comporte très bien face à des algorithmes utilisant toutes l'information disponible tels que MCC, la méthode PRDC s'avère plus efficace dans les cas les plus défavorables. Cela est sans doute dû au critère optimisé par PRDC qui, basé sur des distances relatives, est sans doute mieux adapté à la mesure de performance utilisée.

Une amélioration possible de PCCA pourrait consister à combiner les contraintes sur les paires avec des contraintes sur les triplets comme pour PRDC lorsque celles-ci sont disponibles.

Dans le chapitre précédent, nous avions utilisé des distances calculées sur les graphes de voisinage pour prendre en compte, de manière semi-supervisée d'éventuelles structures (variété ou agrégats) des données. Comme nous l'avons vu, l'utilisation de noyaux permet de modéliser avantageusement les non-linéarités. Un autre avantage potentiel de l'utilisation des matrices noyaux tient au fait qu'il est possible d'inclure facilement de l'information concernant les données non étiquetées. En effet, dans nos expériences, les vecteurs $\boldsymbol{k}_i$ contenaient les valeurs de noyau entre le point i et tous

les autres points de la base d'entraînement. Il est tout à fait envisageable d'inclure dans les vecteurs $\boldsymbol{k}_i$ les valeurs du noyau entre le point i et tous les points de la base de données, points non étiquetés compris. L'implémentation reste inchangée, il suffit de passer en entrée la matrice noyau complète plutôt que seulement la matrice noyau correspondant aux données d'entraînement.

Une autre amélioration possible de cet algorithme consisterait à étendre l'apprentissage de distance aux cas où les objets à comparer proviendraient de domaines d'observation différents ou, autrement dit, de modalités différentes. Ce type d'apprentissage de distance trans-modal est le sujet du chapitre suivant.

CHAPITRE 5

Apprentissage de distance trans-modale

Ce chapitre s'intéresse à une forme particulière de vérification de visages, la vérification trans-modale. Il s'agit d'apparier des données (nous appliquerons cela aux visages) provenant de différentes modalités, lorsque des paires de données similaires/dissimilaires sont disponibles pour l'entraînement. Nous proposons dans ce chapitre une nouvelle approche pour l'apprentissage de distance trans-modale et montrons sa pertinence pour la reconnaissance de visages.

Dans cette situation, les approches standards telles que les ***moindres carrés partiels*** (PLS pour *Partial Least Squares*) [128] ou l'***analyse en corrélations canoniques*** (CCA pour *Canonical Correlation Analysis*) [64], projettent les données dans un espace latent commun dans lequel la covariance est maximisée en utilisant l'information portée par les paires positives (similaires) seulement. Notre contribution est un nouvel algorithme d'apprentissage de distance qui lève cette limitation en considérant à la fois les contraintes positives et négatives. L'information portée par les paires négatives nous permet de construire de manière efficace un espace latent possédant de meilleures propriétés de discrimination.

L'algorithme proposé est une généralisation au cas trans-modal de l'algorithme PCCA présenté au chapitre précédent. Nous validons notre approche sur différentes bases de données où notre algorithme surpasse PLS, CCA et d'autres approches plus récentes.

Sommaire

5.1 L'appariement trans-modal

Dans le chapitre précédent, nous avons mis au point un algorithme capable d'apprendre une projection vers un espace de dimension potentiellement faible dans lequel la distance euclidienne reflète un jeu de contraintes de similarité portant sur des paires d'échantillons d'apprentissage.

Dans ce chapitre, nous nous intéresserons au problème que nous appellerons *l'appariement trans-modal* (ATM), qui est la tâche consistant à prévoir si une paire de points de données provenant de modalités différentes représente le même objet ou non. De nombreuses tâches en vision par ordinateur sont liées à l'ATM. On peut citer par exemple, pour la reconnaissance de personne, la

comparaison entre des photographies et des dessins, entre des photos en haute et basse résolution, entre des visages vus de profil ou de face, etc. La ré-identification de personnes entre des vues prises par plusieurs caméras peut aussi être vue comme une tâche d'ATM. En dehors de la vision par ordinateur, l'ATM est une tâche importante dans beaucoup d'autres domaines puisque, lorsque des observations sont réalisées avec des appareils différents ou lorsque les données sont représentées de manière différentes (par exemple une description textuelle associée à une image) un lien doit pouvoir être établi si l'on veut pouvoir les comparer.

Il existe *grosso modo*, comme nous le verrons dans la section suivante, deux manières de traiter le problème de l'ATM. La première consiste à transférer (ou synthétiser) les données représentées dans une modalité vers l'autre modalité afin que toutes les données soient comparables. La deuxième consiste à apprendre un espace latent de représentation commun, différent des deux espaces initiaux de représentation et ensuite projeter les données depuis les deux espaces distincts de représentation d'origine vers ce nouvel espace dans lequel elles peuvent être comparées directement. Le travail présenté dans ce chapitre suit cette seconde approche.

Plus spécifiquement, nous nous intéressons à des tâches pour lesquelles sont disponibles des données d'entraînement pouvant être utilisées pour mettre en relation les deux modalités de manière optimale. Nous considérons le cas où les données d'entraînement sont données sous la forme de deux jeux d'échantillons, dans deux modalités différentes. Les deux jeux de données sont liés à travers un ensemble de contraintes portant sur des paires de points et indiquant si ces points représentent des objets similaires (devant être appariés) ou différents (ne devant pas être appariés). Par exemple, pour la vérification de visages avec des photographies et des dessins, les données d'entraînement seront composées d'un ensemble de paires photographie-dessin représentant la même personne et de paires photographie-dessin représentant des personnes différentes. La figure 5.1 représente un tel dispositif. Par abus de langage, nous nous référerons aux paires d'échantillons reliés par une contrainte de similarité (ou contrainte positive) comme des paires positives et aux paires d'échantillons reliés par une contrainte de dissimilarité (ou contrainte négative) comme des paires négatives.

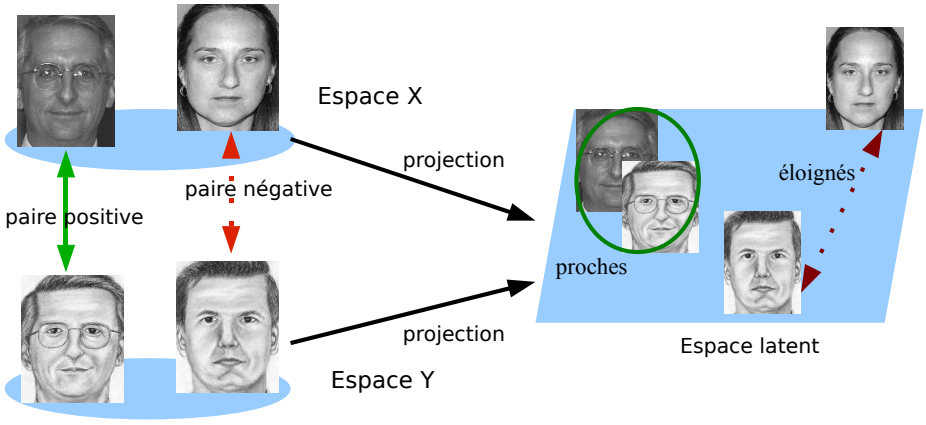


FIGURE *5.1: Appariement trans-modal : deux jeux de points de données dans deux espaces de représentation différents X (photographie) et Y (dessins) sont mis en relation à travers un ensemble de contraintes de similarité/dissimilarité portant sur des paires de points. Nous apprenons un espace de représentation latent commun dans lequel les distances reflètent ces contraintes.*

Comme nous l'avons déjà signalé au chapitre précédent, la formulation des contraintes sous forme de paires est très générale puisqu'elle permet de décrire, entre autre, les problèmes dans lesquels les labels de classes sont connus. En effet, dans le cas où les labels sont connus, les relations de similarité/dissimilarité sont connues pour toutes les paires.

La découverte d'un espace latent commun pour l'ATM est un problème important et a par conséquent été très étudié. Plusieurs approches, largement utilisées, ont été proposées telles que les ***moindres carrés partiels*** (PLS pour *Partial Least Squares*) ou l'***analyse en corrélations canoniques*** (CCA pour *Canonical Correlation Analysis*). Ces méthodes ont pour but de trouver deux matrices de projection conduisant à maximiser la covariance dans l'espace de projection, en se basant sur les contraintes positives disponibles dans les données d'entraînement.

Un inconvénient majeur de ces approches est qu'elles ignorent les contraintes négatives, c'est-à-dire les contraintes définies par des paires de points représentant des objets différents. Cependant, pour de nombreuses tâches, ces contraintes négatives sont disponibles et les mettre à contribution peut aider à améliorer les performances. Plusieurs approches récentes tentent d'utiliser les deux types de contraintes [81, 80], mais elles souffrent d'un certain nombre de limitations que nous évoquerons dans la section suivante. La levée de ces limitations est précisément le but de notre contribution.

Dans la suite de ce chapitre, nous proposons une approche du problème de l'ATM par l'adaptation de notre méthode PCCA (*Pairwise Constrained Component Analysis*), qui met à profit les contraintes positives et négatives. Nous reprenons donc l'idée intuitive que, dans l'espace latent commun aux deux modalités, les objets similaires devraient être proches alors que les objets différents devraient être éloignés. Notre approche est validée sur trois bases de données pour différentes tâches de reconnaissance de visages : la base CUFSF [125] pour la reconnaissance entre dessins et photographies, la base Multi PIE [57] pour la reconnaissance multi-pose, et la base LFW [66] pour la vérification avec des descripteurs différents. Nos expériences montrent qu'incorporer explicitement les contraintes négatives améliore les performances de l'ATM et que notre algorithme surpasse généralement les autres algorithmes apprenant des espaces latents tels que PLS et CCA.

5.2 Les algorithmes existants pour l'appariement trans-modal

Puisqu'il s'agit d'un problème important, l'appariement trans-modal (ATM) a été abondamment étudié dans la littérature. Comme nous l'avons indiqué précédemment, les travaux publiés peuvent être organisés selon deux catégories : d'un côté les algorithmes qui transfèrent les données d'une des modalités vers l'autre, et, de l'autre côté, celles qui construisent un espace latent commun dans lequel les données appartenant aux deux modalités peuvent être projetées et comparées.

Lorsque les données représentées dans une modalité sont transférées dans l'espace correspondant à la deuxième modalité, les données ainsi synthétisées peuvent alors être directement comparées dans ce second espace. Dans [28], par exemple, Chen *et al.* transfèrent des images en infrarouge vers des images du spectre visible. Grâce au fait que les images sont alors directement comparable dans le spectre visible, cette transformation facilite la tâche à un opérateur devant analyser les deux images. De la même manière, Lui *et al.* [82] et Wang *et al.* [125] synthétisent des dessins de visages à partir de photographies pour ensuite effectuer une comparaison directe entre les dessins.

Par ailleurs, lorsque les deux modalités sont suffisamment proches, l'application de filtres spécifiques peut grandement réduire les différences entre les deux représentations et même éventuellement les rendre directement comparables [54, 72]. Cette stratégie ne peut toutefois être utilisée que dans des cas très particuliers.

Malgré les bons résultats obtenus par les approches basées sur la synthèse, celles-ci sont souvent très spécifiques au domaine étudié dans le sens où le processus de synthèse doit être redéfini pour chaque nouveau problème, lorsque cela est possible.

La représentation latente repose sur l'hypothèse que les deux modalités peuvent s'expliquer par un jeu restreint de variables latentes. Sa mise en œuvre repose souvent sur l'utilisation d'outils statistiques très répandus tels que l'***analyse en corrélations canoniques*** (CCA pour *Canonical Correlation Analysis*) [64] ou les ***moindres carrés partiels*** (PLS pour *Partial Least Squares*) [128], dont le but est de trouver une représentation latente en maximisant la covariance dans l'espace latent.

La principale différence entre ces deux méthodes réside dans les contraintes utilisées pour éviter les solutions triviales (tous les points sont projetés au même endroit) et qui contrôlent la qualité de la reconstruction dans l'espace latent [20]. PLS a été récemment utilisé dans [106] pour différentes tâche d'ATM (reconnaissance photographie/dessins, comparaison en haute et basse résolution, et reconnaissance multi-pose) pour lesquels Abhishek *et al*. montrent que la méthode est compétitive avec d'autres méthodes spécialisées pour la tâche [82, 72]. Li *et al*. utilisent CCA dans [78] pour reconnaître des visages soumis à des occlusions partielles en comparant des parties différentes du visage (par exemple les yeux et le nez). Zhang *et al*. [137] encodent des photographies et des dessins de sorte que les deux représentations soient les mêmes pour une même personne. Pour arriver à ce résultat, ils apprennent une structure arborescente dans laquelle les nœuds projettent un sous-ensemble de la représentation des photographies et des dessins dans un espace commun en utilisant CCA. Les résultats rapportés pour cette méthode battent l'état de l'art sur la base CUFSF (voir section 1.1.3.3 page 8).

Par construction, ces méthodes ignorent les contraintes négatives, c'est-à-dire les contraintes données par les paires de points non-concordants. Pour remédier à ce problème, Yi *et al*. [134] suggèrent l'emploi de l'analyse linéaire discriminante (LDA pour *Linear Discriminant Analysis*) dans chacun des espaces d'entrée, puis apprennent l'espace latent commun avec CCA pour comparer des visages capturés dans les spectres infra-rouge et visible. Lei *et al*. [77] adoptent une démarche similaire mais introduise un terme de couplage pour effectuer les deux LDA de manière simultanée dans leur méthode dénommée CSR (pour *Coupled Spectral Regression*). Pour ces deux méthodes, l'information discriminante est utilisée sans l'espace d'entrée alors qu'il semble plus judicieux de l'intégrer directement dans la construction de l'espace latent.

C'est précisément ce que proposent Zhou *et al*. [140], en dérivant un équivalent de LDA directement dans l'espace latent, de sorte que la projection et l'analyse discriminante soient réalisées simultanément. Le point commun entre ces méthodes basée sur LDA est qu'elles nécessitent la connaissance des labels de classe pour toutes les données d'entraînement. Cette information n'est pas disponible lorsque seule une information de similarité/dissimilarité donnée pour des paires de points est connue.

De manière alternative, Lin *et al*. ont proposées deux méthodes proches, CDFE (pour *Common Discriminant Feature Extraction*) [81] et la méthode DMSL (pour *Discriminant Mutual Subspace Learning*) [80], pour prendre en compte un ensemble arbitraire de contraintes sur les paires, directement dans l'espace latent. Toutes les deux ont pour but de maximiser les distances pour les paires négatives tout en minimisant la distance pour les paires positives. CDFE incorpore également un terme de régularisation pour favoriser la cohérence entre les distances dans les espaces d'entrée et l'espace latent. Une des limitations de ce genre d'approche tient, de notre point de vue, dans la fonction de coût utilisée pour apprendre l'espace latent. Bien qu'intuitif, maximiser les distances des paires négatives tout en minimisant les distances des paires positives peut ne pas être pertinent. Si deux points d'une paire négative sont déjà plus éloignés que les points de n'importe quelle paire positive, agrandir encore l'écart entre ces points – ce qui a toutes les probabilités d'arriver puisque les distances pour les paires positives sont minorées par zéro – n'améliore pas la discrimination et peut même conduire à du sur-ajustement. Nous pensons que les contributions des paires à la fonction de coût ne devraient être importantes que lorsque les contraintes ne sont pas satisfaites. Autrement dit, nous estimons que l'information importante réside dans la « marge », c'est-à-dire, l'intervalle de distances pour lesquelles les distances des paires positives et négatives sont proches. Par ailleurs, dans leurs expériences sur CSR, Lei *et al*. [77] montrent que les performances de CDFE diminuent considérablement lorsque les personnes à reconnaître ne sont pas présentes dans la base d'apprentissage. Si un tel système est envisageable pour une application de sécurité (il faut décider si la personne qui se présente fait partie des personnes autorisées à entrer dans une zone sécurisée par exemple), dans le cas général, les visages à traiter ne sont pas connus à l'avance.

La méthode que nous proposons ressemble à CDFE et DMSL de par le fait que nous traitons

également des contraintes sur les paires directement pour apprendre un espace latent discriminant. Cependant notre méthode d'***apprentissage de métrique trans-modale*** (CMML pour *Cross-Modal Metric Learning*) aborde le problème du point de vue opposé dans le sens où nous pénalisons les grandes distances pour les paires positives et les petites distances pour les paires négatives. La fonction de coût de CMML est une adaptation de la fonction de coût de PCCA présentée au chapitre précédent dans le cas trans-modal. Pour cette raison, on peut estimer que la fonction de coût utilisée est du même type que celle utilisée dans les Séparateurs à Vaste Marge (SVM) ou la régression logistique.

5.3 CMML : Apprentissage de distance trans-modale

Nous reprenons les notations du chapitre précédent. Dans le cas mono-modale, nous avions utilisé une application $\mathcal{A} : \mathcal{X} \mapsto \mathbb{R}^k$, pour projeter les points de $\mathcal{X}$ dans un espace euclidien $\mathbb{R}^k$ de dimension k. La mesure de distance correspondait alors à la distance euclidienne dans $\mathbb{R}^k$ que l'on écrivait :

$$D^2_{\mathcal{A}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\mathcal{A}(\boldsymbol{x}_i) - \mathcal{A}(\boldsymbol{x}_j)\|_2^2 \tag{5.1}$$

Nous considérons à présent le cas où nous disposons de deux ensembles d'échantillons $\mathcal{D}_{\mathcal{X}}$ et $\mathcal{D}_{\mathcal{Y}}$ dans deux espaces de Hilbert distincts respectivement notés $\mathcal{X}$ et $\mathcal{Y}$. Nous cherchons deux applications distinctes $\mathcal{A} : \mathcal{X} \mapsto \mathbb{R}^k$ et $\mathcal{B} : \mathcal{Y} \mapsto \mathbb{R}^k$ projetant respectivement les points de $\mathcal{A}$ et $\mathcal{B}$ dans le même espace euclidien $\mathbb{R}^k$. La distance entre un point $\boldsymbol{x} \in \mathcal{X}$ et un point $\boldsymbol{y} \in \mathcal{Y}$ est alors mesurée comme la distance euclidienne dans $\mathbb{R}^k$:

$$D^2_{\mathcal{A},\mathcal{B}}(\boldsymbol{x}_i, \boldsymbol{y}_j) = \|\mathcal{A}(\boldsymbol{x}_i) - \mathcal{B}(\boldsymbol{y}_j)\|_2^2 \tag{5.2}$$

Notons respectivement $n_{\mathcal{X}} = |\mathcal{D}_{\mathcal{X}}|$ et $n_{\mathcal{Y}} = |\mathcal{D}_{\mathcal{Y}}|$ les cardinaux de $\mathcal{D}_{\mathcal{X}}$ et $\mathcal{D}_{\mathcal{Y}}$, les contraintes sont alors données comme un ensemble $\mathcal{P}$ de paires d'indices $(i, j) \in \{1 \dots n_{\mathcal{X}}\} \times \{1 \dots n_{\mathcal{Y}}\}$ et une fonction label $l_{ij} : \mathcal{P} \to \{1, -1\}$ qui, à chaque élément de $\mathcal{P}$, associe un label indiquant si les éléments $\boldsymbol{x}_i$ et $\boldsymbol{y}_j$ indexés par la paire (i, j) sont similaires ($l_{ij} = 1$) ou non ($l_{ij} = -1$).

Le but de la méthode est donc de trouver les deux applications $\mathcal{A}$ et $\mathcal{B}$ qui projettent les points de $\mathcal{X}$ et $\mathcal{Y}$ respectivement dans le même espace $\mathbb{R}^k$ de telle façon que les distances euclidiennes dans cet espace reflètent au mieux les contraintes.

La fonction de coût que nous utilisons est l'adaptation de la fonction de coût du chapitre précédent :

$$E(\mathcal{A}, \mathcal{B}) = \sum_{(i,j)\in\mathcal{P}} \ell_\beta \left(l_{ij}(D^2_{\mathcal{A},\mathcal{B}}(\boldsymbol{x}_i, \boldsymbol{y}_j) - 1) \right) \tag{5.3}$$

$$\ell_\beta(z) = \frac{1}{\beta} \log(1 + e^{\beta z})$$

5.4 Applications $\mathcal{A}$ et $\mathcal{B}$ et optimisation

Cas linéaire. Lorsque $\mathcal{X}$ et $\mathcal{Y}$ sont des espaces euclidiens, la forme la plus simple pour les applications $\mathcal{A}$ et $\mathcal{B}$ est l'application linéaire paramétrée respectivement par des matrices A et B. La mesure de distance est alors :

$$D^2_{A,B}(\boldsymbol{x}_i, \boldsymbol{y}_j) = \|A\boldsymbol{x}_i - B\boldsymbol{y}_j\|_2^2 \tag{5.4}$$

Convexité. Nous montrons à présent qu'avec ce choix pour les applications $\mathcal{A}$ et $\mathcal{B}$, la minimisation de la fonction d'énergie de l'équation (5.3) est, comme dans le chapitre précédent, la forme factorisée d'une fonction convexe définie sur l'ensemble des matrices semi-définies positives.

Construisons le vecteur $\boldsymbol{z}_n = (\boldsymbol{x}_{i_n}^T, -\boldsymbol{y}_{j_n}^T)^T$ comme la concaténation des vecteurs $\boldsymbol{x}_{i_n}$ et $-\boldsymbol{y}_{j_n}$ correspondant à la n-ième paire de $\mathcal{P}$. Soit $d_\mathcal{X}$ et $d_\mathcal{Y}$ les dimensions respectives des espaces euclidiens $\mathcal{X}$ et $\mathcal{Y}$. Nous construisons également la matrice $C = [A\ B]$ de dimension $k \times (d_\mathcal{X} + d_\mathcal{Y})$ comme la matrice dont les lignes sont formées par la concaténation des lignes correspondantes des matrices A et B.

La mesure de distance se réécrit alors :

$$D^2_{A,B}(\boldsymbol{x}_i, \boldsymbol{y}_j) = \|Cz_n\|_2^2 = z_n^T C^T C z_n = z_n^T M z_n \tag{5.5}$$

où $M = C^T C$ est une matrice semi-définie positive par construction. La mesure de distance est donc une fonction convexe de M et toute la discussion du chapitre précédent concernant l'optimalité des solutions trouvées en optimisant par rapport à C ainsi que les algorithmes qui en découlent restent valides.

Cas non linéaire. Nous pouvons étendre la méthode au cas non linéaire en utilisant le « truc » du noyau. Reparamétrisons les matrices $\hat{A}$ et $\hat{B}$ par :

$$\begin{aligned} A &= \hat{A} X^T \\ B &= \hat{B} Y^T \end{aligned} \tag{5.6}$$

où X et Y sont les matrices dont chaque colonne représente respectivement un élément de $\mathcal{D}_\mathcal{X}$ et $\mathcal{D}_\mathcal{Y}$. Cela signifie que chacune des lignes de A (resp. B) peut être vue comme une combinaison linéaire des éléments de $\mathcal{D}_\mathcal{X}$ (resp. $\mathcal{D}_\mathcal{Y}$).

Nous définissons les matrices noyaux (ou matrices de Gram) suivantes $K_\mathcal{X} = XX^T$ et $K_\mathcal{Y} = YY^T$. On note alors $\boldsymbol{k}_{\mathcal{X}i}$ et $\boldsymbol{k}_{\mathcal{Y}j}$ respectivement la i-ème colonne de $K_\mathcal{X}$ et la j-ième colonne de $K_\mathcal{Y}$.

La mesure de distance prend alors la forme :

$$\begin{aligned} D^2_{A,B}(\boldsymbol{x}_i, \boldsymbol{y}_j) &= \|\hat{A} X^T \boldsymbol{x}_i - \hat{B} Y^T \boldsymbol{y}_j\|_2^2 \\ &= \|\hat{A} \boldsymbol{k}_{\mathcal{X}i} - \hat{B} \boldsymbol{k}_{\mathcal{Y}j}\|_2^2 \\ &= D^2_{\hat{A},\hat{B}}(\boldsymbol{k}_{\mathcal{X}i}, \boldsymbol{k}_{\mathcal{Y}j}) \end{aligned} \tag{5.7}$$

Comme dans le cas mono-modal, la distance garde la même forme analytique lorsqu'elle est exprimée en fonction des colonnes de X et Y ou en fonction des colonnes de $K_\mathcal{X}$ et $K_\mathcal{Y}$.

Les mêmes problèmes de pré-conditionnement que ceux évoqués dans le cas mono-modale du chapitre précédent apparaissent.

Le gradient de la fonction $E(A, B)$ s'écrit :

$$\begin{aligned} \nabla_A E(A,B) &= \sum_{(i,j)\in\mathcal{P}} \mathcal{L}'_{ij}(A\boldsymbol{x}_i - B\boldsymbol{y}_j)\boldsymbol{x}_i^T \\ \nabla_B E(A,B) &= -\sum_{(i,j)\in\mathcal{P}} \mathcal{L}'_{ij}(A\boldsymbol{x}_i - B\boldsymbol{y}_j)\boldsymbol{y}_j^T \end{aligned} \tag{5.8}$$

avec $\mathcal{L}'_{ij} = 2 l_{ij} \ell'_\beta \left(l_{ij}(\|A\boldsymbol{x}_i - B\boldsymbol{y}_j\|^2 - 1)\right)$. La règle d'adaptation lors de l'utilisation des noyaux est donc :

$$\begin{aligned} \hat{A}^{t+1} &\leftarrow \hat{A}^t - \eta \textstyle\sum_{(i,j)\in\mathcal{P}} (\mathcal{L}'_{ij}(\hat{A}^t \boldsymbol{k}_{\mathcal{X}i} - \hat{B}^t \boldsymbol{k}_{\mathcal{Y}j}))\boldsymbol{k}_{\mathcal{X}i} K_\mathcal{X}^{-1} \\ \hat{B}^{t+1} &\leftarrow \hat{B}^t - \eta \textstyle\sum_{(i,j)\in\mathcal{P}} (\mathcal{L}'_{ij}(\hat{A}^t \boldsymbol{k}_{\mathcal{X}i} - \hat{B}^t \boldsymbol{k}_{\mathcal{Y}j}))\boldsymbol{k}_{\mathcal{Y}j} K_\mathcal{Y}^{-1} \end{aligned} \tag{5.9}$$

5.5 Résultats expérimentaux

Dans cette section, nous validons notre contribution pour l'apprentissage de distance trans-modal CMML à travers des expériences sur trois bases de données différentes : Multi-PIE, CUFSF et Labeled

Faces in the Wild. Nous rapportons les performances de notre algorithme sur ces bases de données et fournissons des comparaisons avec trois autres méthodes pouvant utiliser des contraintes sur les paires : l'analyse en corrélations canoniques (CCA) [64], les moindres carrés partiels (PLS) [128] et l'extraction de caractéristiques discriminantes communes (CDFE) [81, 80].

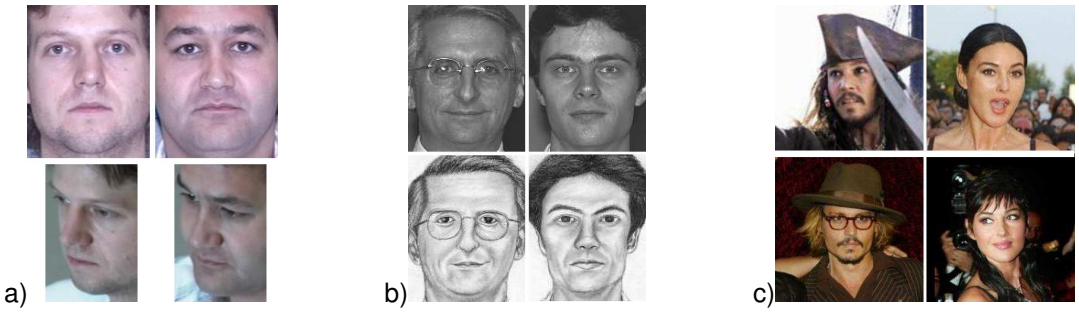


FIGURE *5.2: Exemples d'images issues des bases Multi PIE, CUFSF et LFW datasets. a) Multi PIE : images provenant des vues "05_1" (rangée du haut) et "08_1" (rangée du bas) (voir Fig. 1.1 pour plus de précision que la position des caméras). b) CUFSF : deux images (ligne du haut) et les dessins correspondant (ligne du bas). c) LFW : deux paires d'images positives.*

Nous décrivons d'abord succinctement les base de données utilisées (une description plus détaillée peut être trouvée au chapitre 1) ainsi que les protocoles expérimentaux associés.

5.5.1 Reconnaissance multi-pose : résultats sur Multi-PIE

Dans nos expériences, nous utilisons la première session de Multi-PIE (voir 1.1.3.1 page 4 pour les détails) car c'est celle dans laquelle le plus de personnes sont représentées. Nous n'utilisons que les images correspondant à une illumination frontale et une expression neutre. Nous disposons donc de 249 personnes, représentées sous 15 poses différentes chacune.

Les images ont été alignées en utilisant des annotations manuelles pour les yeux le nez et les coins de la bouche. Un recadrage serré a également été appliqué pour retirer l'essentiel du fond. Ensuite, nous avons extrait des descripteurs LTP (*Local Ternary Patterns*) [109] uniforme sur une grille. Comme la taille des images recadrées est différente pour chaque pose (figure 5.2), la taille des descripteurs correspondants est également différente et varie de 5664 à 9440 composantes.

Le jeu d'entraînement est obtenu en choisissant 149 personnes parmi les 249 disponibles, les 100 restantes sont assignées au jeu de test. Les contraintes positives sont créées en produisant des paires contenant la même personne, alors que les contraintes négatives sont obtenues avec des images de personnes différentes. Les meilleurs résultats ont été obtenus en utilisant le même nombre de paires positives et négatives pendant l'entraînement, toutefois, comme les contraintes négatives peuvent être générées en grande quantité, nous avons également étudié l'apport de contraintes négatives supplémentaires pour les performances.

Les différents algorithmes ont été entraînés pour toutes les combinaisons possibles de poses, ce qui signifie qu'un espace latent est créé pour chaque paire de position de caméra.

Les performances sont évaluées avec deux mesures différentes. La première estime à quel degré l'image d'une pose donnée (la sonde), est appariée avec l'image la plus proche correspondant à l'autre pose. L'ensemble des images dans l'autre pose constituant la galerie. Le nombre moyen d'appariements corrects (le plus proche voisin correspond à la même personne) pour le jeu de test est rapporté sous l'appellation *nearest neighbor accuracy* ou taux de bonne classification du plus proche voisin.

La deuxième mesure considère un ensemble de test constitué d'autant de paires positives que de paires négatives. Chaque personne participe à une paire positive et une paire négative. Après

entraînement, la distance est calculée pour chaque paire et la valeur médiane de ces distances est utilisée comme seuil pour classifier les paires comme positives (la distance est plus petite que le seuil) ou négatives (la distance est plus grande que le seuil). Le taux de paires bien classifiées correspond au taux de bonne classification (*accuracy*), lorsque les taux d'erreurs de classification sont égaux pour les paires positives et négatives (*Equal Error Rate* ou EER). Ce score est donc rapporté sous l'appellation *Accuracy at EER*. L'ensemble du processus est répété 10 fois avec des répartitions et paires négatives aléatoires différentes et nous rapportons la moyenne des résultats obtenus.

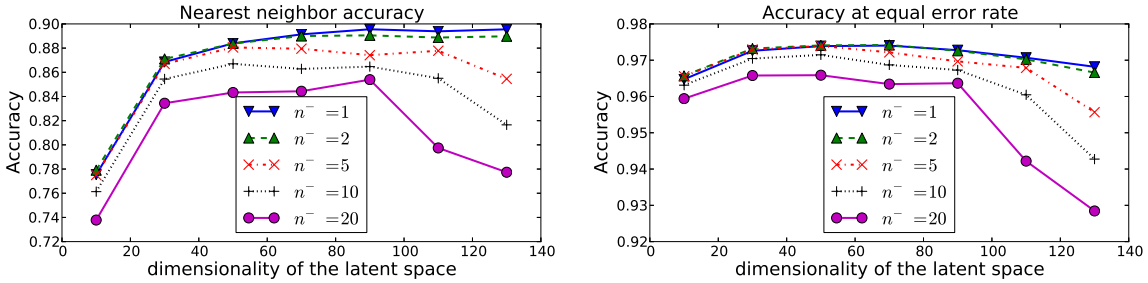


FIGURE *5.3: Taux moyens de bonne classification (accuracy) du premier voisin (nearest neighbor) et à taux d'erreur égaux (equal error rate) pour CMML sur Multi PIE pour différentes valeurs du rapport n^- des paires négatives/positives en fonction de la dimension de l'espace latent.*

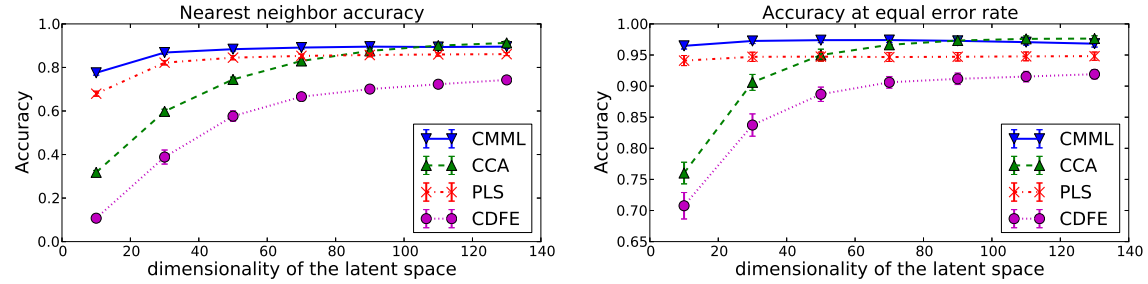


FIGURE *5.4: Multi PIE : Taux moyens de bonne classification pour CMML, CCA, PLS and CDFE, avec $n^- = 1$.*

Method \ Gallery	11_0	12_0	09_0	08_0	08_1	13_0	14_0	05_0	04_1	19_1	19_0	20_0	01_0	24_0
CMML	0.75	0.82	0.89	0.95	0.85	0.94	1.00	1.00	0.99	0.82	0.92	0.89	0.76	0.64
CCA	0.28	0.31	0.30	0.52	0.26	0.43	0.53	0.95	0.94	0.46	0.47	0.38	0.73	0.28
PLS	0.50	0.72	0.80	0.93	0.69	0.93	1.00	1.00	0.99	0.60	0.88	0.76	0.67	0.50
CDFE	0.14	0.28	0.26	0.35	0.15	0.43	0.58	0.51	0.31	0.14	0.33	0.21	0.15	0.18

TABLE *5.1: Taux moyens de bonne classification du premier voisin (nearest neighbor accuracy) pour les paires constituées d'une image de la caméra `05_1` (vue frontale) et des images venant de différentes vues (une caméra par colonne) pour CMML, CCA et PLS. La dimension de l'espace de sortie est 30 et $n^- = 1$. Voir la figure 1.1 page 5 pour les labels de caméra.*

Nous commençons par évaluer les performances de notre méthode (CMML) pour différentes valeurs du rapport n^- du nombre de paires négatives par rapport au nombre de paires positives en fonction de la dimensionalité de l'espace latent. Comme nous l'avons expliqué plus haut, nous utilisons deux mesures de performances : (a) Le taux de bonne classification du premier voisin (*Nearest*

neighbor accuracy) et (b) le taux de bonne classification à taux d'erreur égaux (*accuracy at Equal Error Rate* ou EER). La figure 5.3 montre les performances de CMML moyennées sur toutes les paires de poses, pour différentes tailles de l'espace latent et différentes valeurs du rapport n^- ($n^- = k$ signifie qu'on a utilisé k paires négatives pour 1 paire positive). Contrairement à ce que nous avions observé dans le cas mono-modale au chapitre précédent, les meilleures performances sont obtenues avec un taux $n^- = 1$. Les performances chutent lorsque le rapport augmentent et la chute semble même plus prononcée pour de grandes dimensions de l'espace latent. Nous avons identifié deux facteurs qui peuvent expliquer cela : 1) lorsque n^- augmente, trop de poids est donné aux paires négatives et 2) le sur-apprentissage.

Nous comparons ensuite les résultats de CMML avec ceux de ses compétiteurs (CCA, PLS et CDFE). Les performances correspondent aux résultats moyens pour toutes les paires de poses en fonction de la dimension de l'espace latent. Pour des espaces latents de faible dimension, CMML et PLS surpassent clairement CCA et CDFE avec un avantage significatif pour CMML, dont les performances restent systématiquement supérieures à celle de PLS. Les performances de PLS et CMML sont très stables par rapport à la dimension de l'espace latent alors que celles de CCA et CDFE augmentent avec cette dimension. Alors que les performances de CDFE restent toujours plus faibles que celles des autres méthodes, celles de CCA dépassent les performances de CMML lorsque la dimension de l'espace est supérieure à 100 environ.

Le tableau 5.1 rapporte le taux de classification du plus proche voisin pour des paires constituées d'une image en prise de vue frontale (caméra `5_01`) et d'une autre image venant des autres caméras. Chaque colonne correspond à une vue différente. Par exemple, la première colonne donne les performances pour une image sonde provenant de la caméra *05_1* et une galerie correspondant à la caméra `11_0`. Les vues sont classées de gauche à droite en accord avec l'organisation spatiale des caméras telle qu'illustrée à la figure 1.1. En conséquence, les vues de profil (caméras `11_0` et `24_0`) se trouvent aux extrémités. La dimension de l'espace latent est 30. Comme attendu, l'appariement est plus facile pour les vues proches de la vue frontale (caméras `14_0` et `5_0`) pour lesquelles CMML et PLS donnent des résultats parfaits. Avec des différences de vues plus marquées, CMML surpasse toutes les autres méthodes pour une dimension de l'espace latent très réduite.

5.5.2 Reconnaissance de visages photographie/dessin : résultats sur CUFSF

Nous avons mené des expériences similaires sur la base CUFSF (voir la description section 1.1.3.3 page 8).

Comme dans les expériences sur Multi PIE, nous avons extrait des descripteurs LTP sur les images alignées et recadrées en utilisant les annotations fournies avec la base de données. Les descripteurs forment des vecteurs de 9440 dimensions pour les deux modalités.

Les algorithmes sont entraînés avec 500 personnes sur les 860 disponibles choisies aléatoirement. Les 360 restantes sont utilisées pour le test.

Les protocoles expérimentaux sont exactement les mêmes que pour les expériences sur Multi PIE ainsi que les mesures de performances.

La figure 5.5 montre que CMML surpasse toutes les autres méthodes avec une large marge. Comme pour Multi PIE, les performances sont plutôt stables par rapport à la dimension de l'espace d'entrée. Alors que le taux de bonne classification du premier voisin augmente pour saturer autour de 70 dimensions, le taux de bonne classification à taux d'erreur égaux reste plus ou moins identique sur l'intervalle de dimensions étudié. PLS suit une évolution parallèle à CMML mais avec un écart important en faveur de CMML. Contrairement aux résultats obtenus sur Multi PIE, les performances de CCA sur CUFSF, après une légère augmentation jusqu'à 30 dimensions, baissent pour les plus grandes dimensions. CDFE montre les mêmes tendances que sur Multi PIE, ces performances augmentent pour saturer aux plus grandes dimensions atteignant des performances similaires à celles de CCA.

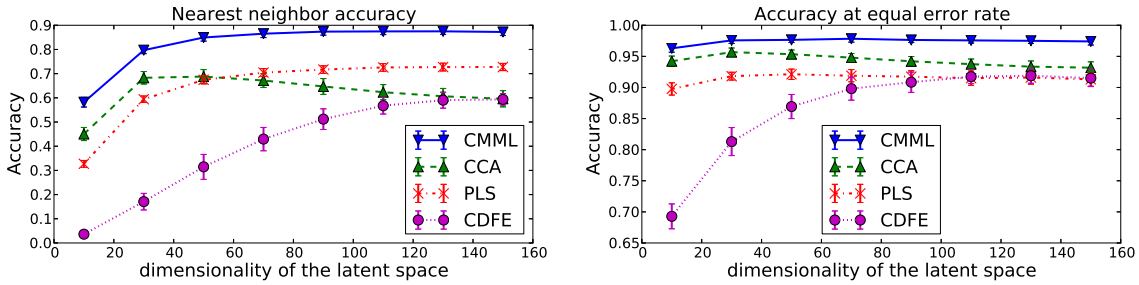


FIGURE *5.5: CUFSF : Taux moyens de bonne classification pour le premier voisin (nearest neighbour accuracy, à gauche, et à taux d'erreur égaux (EER), à droite, pour CMML ($n^- = 1$), CCA, PLS et CDFE en fonction de la dimension de l'espace latent.*

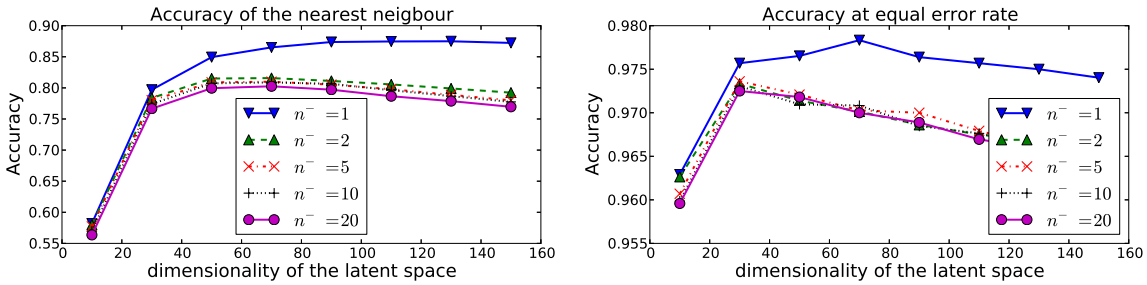


FIGURE *5.6: CUFSF : Taux moyens de bonne classification pour le premier voisin (nearest neighbour accuracy, à gauche, et à taux d'erreur égaux (EER), à droite, avec CMML pour différentes valeurs du rapport $n^- = 1$ en fonction de la dimension de l'espace latent.*

Le comportement de CMML sur CUFSF (figure 5.6) en fonction du nombre de contraintes négatives est similaires à celui observé avec Multi PIE. Les meilleurs résultats sont obtenus avec 1 paire négative par paire positive, et la baisse de performance observée lorsque le rapport n^- augmente s'accentue avec le nombre de dimension de l'espace latent.

5.5.3 Vérification de visage trans-modale : résultats sur LFW

Comme nous l'avons vu dans les deux chapitres précédents, la tâche testée par LFW n'est pas à l'origine une tâche d'ATM. Cependant, nous avons introduit de la trans-modalité en représentant chaque visage d'une paire d'entraînement avec des descripteurs différents. Pour le premier élément de chaque paire, nous utilisons le même descripteur SIFT que dans les deux précédents chapitres (ce sont des descripteurs fournis par Guillaumin *et al.* [58] et téléchargeables depuis la page Web des auteurs). Pour le deuxième élément de la paire, nous utilisons, comme dans les expériences précédentes, des descripteurs LTP calculés sur la version alignée de la base fournie par Huang *et al.* [67]. La taille des descripteurs est respectivement de 3456 et 14160 dimensions. Nos expériences sur LFW utilisent donc ces deux descripteurs de natures très différentes ce qui est équivalent à avoir deux modalités différentes ne pouvant être comparées directement.

Comme nous utilisons le protocole restreint de LFW (voir section 1.1.3.2, page 5), nous ne disposons d'annotations que sur la qualité de la paire (positive ou négative) et pas sur les personnes représentées. C'est pourquoi le taux de classification du plus proche voisin ne peut être calculé et nous ne reportons que le taux de bonne classification à taux d'erreur égaux.

Dans Multi PIE et CUFSF, les visages étaient capturés dans des conditions de pose, d'éclairement

et d'expression strictement contrôlées. À l'inverse, dans LFW, les photographies ont été prises sur le vif, et les conditions de prise de vue présentent donc une grande variabilité. Sur cette base, on voit (figure 5.7) que PLS, CCA et CDFE présentent de très mauvaises performances. Les performances de CDFE restent à peu près stables avec la dimension de l'espace de sortie, mais le score est à peine meilleur que le hasard (entre 55 et 60%). Le score de PLS est autour de 63% pour un espace latent de 20 à 40 dimensions. Les performances de CCA augmente de manière continue avec le nombre de dimension mais à 150 dimensions, le score reste en dessous de 70%, alors que le score de CMML pour 70 dimensions reste très stable autour de 80% ce qui, quoique plus faible que les 85% rapportés dans le cas mono-modal au chapitre précédent, reste très encourageant étant donnée l'incompatibilité structurelle des descripteurs utilisés.

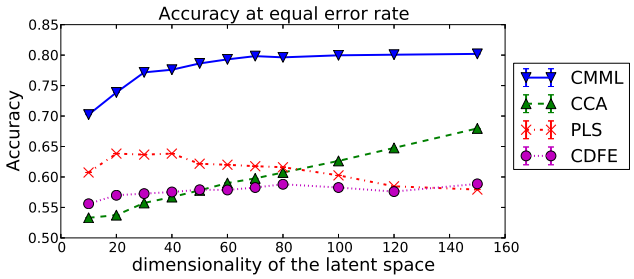


FIGURE *5.7: Taux de bonne classification sur LFW à taux d'erreurs égaux (accuracy at equal error rate) pour CMML ($n^- = 1$), CCA, PLS et CDFE.*

5.6 Conclusion

Dans ce chapitre nous avons apporté une contribution sur la question de l'appariement trans-modal. Pour traiter ce problème, nous avons introduit une nouvelle méthode d'apprentissage de distance, CMML, capable d'apprendre un espace latent de basse dimension de manière discriminative en présence de données représentées dans deux modalités différentes ne pouvant être comparées directement. Des expériences sur trois bases de données différentes dédiées à la reconnaissance de visage ont montré que CMML est capable de trouver des structures pertinentes de basse dimension tout en restant robuste vis-à-vis du sur-apprentissage lorsque la dimension de l'espace latent augmente. De plus CMML présente de meilleures performances que d'autres méthodes couramment utilisées pour ce genre d'analyse telles que CCA, PLS et CDFE.

CMML est l'adaptation au cas trans-modal de notre algorithme PCCA. Nous avons l'intuition, que cette approche d'apprentissage de distance pourrait encore être généralisée aux cas où plus de deux modes sont en jeux.

En effet, dans les expériences que nous avons menées sur la reconnaissance multi-pose, nous n'avons considéré que des paires de poses. Or l'algorithme pourrait tout à fait être adapté pour prendre en compte l'ensemble des poses en même temps. Ceci conduirait à la construction d'un seul espace commun pour toutes les poses avec une matrice de projection par pose. Appliqué correctement, cela pourrait conduire à un algorithme de reconnaissance de visage robuste aux variations de poses.

L'application d'une telle approche suppose cependant que l'on soit capable de reconnaître correctement la pose pour savoir quelle projection utiliser. Une autre piste pourrait être de projeter les deux visages en utilisant toutes les matrices de projection disponibles (sans essayer de déterminer la pose exacte) et considérer la distance minimum entre toutes les paires de poses.

D'autres applications sont envisageables. Une adaptation de la méthode pour l'apprentissage multi-instance à l'image de ce qu'ont fait Guillaumin *et al.* avec LDML [59] est un exemple parmi d'autres.

CHAPITRE 6

Approximation et apprentissage de noyaux additifs homogènes

Les méthodes à noyaux permettent d'approcher n'importe quelle fonction ou frontière de décision, au moyen des données d'entraînement. Cependant, cette dépendance aux données d'entraînement implique souvent que le calcul de la fonction noyau soit faite pour toutes les paires de points d'entraînement, avec des fonctions parfois longues à calculer. Ceci peut conduire à des coûts importants, aussi bien en termes de consommation de mémoire (pour de grandes matrices noyau) que de ressources de calcul.

Pour limiter ce coût, des techniques d'approximation ont été développées. Alors que, chronologiquement parlant, les premières méthodes [49, 3, 18] tendaient à approcher la matrice noyau, plus récemment sont apparues des techniques portant sur l'approximation de la fonction de re-description [97, 79, 121].

Dans ce chapitre, nous nous intéressons spécifiquement à la fonction de re-description des noyaux additifs homogènes. Cette fonction de re-description ne diffère, entre deux noyaux additifs homogènes, que par la donnée d'une fonction de densité $\kappa(\lambda)$. La contribution de ce chapitre est double. Nous donnons tout d'abord la forme analytique de la fonction de densité du ***noyau de la moyenne puissance*** qui généralise plusieurs noyaux d'usage commun. Nous proposons ensuite une approche pour la construction de noyaux additifs homogènes basée sur l'apprentissage explicite de la fonction de densité $\kappa(\lambda)$.

Sommaire

6.1 Travaux antérieurs

Avec l'introduction des méthodes à noyaux en apprentissage statistique [119, 118], la nécessité de calculer des approximations rapides pour les problèmes à grande échelle s'est rapidement fait sentir. En 1998, Frieze *et al.* [49] proposent une méthode pour la décomposition rapide et approchée de matrices. Dans [3], plusieurs techniques d'échantillonnage et d'approximation sont présentées pour

accélérer l'analyse en composante principale par noyaux (*Kernel PCA*). Dans [18], la matrice noyau est approchée à partir de projections aléatoires linéaires.

Toutes les méthodes que nous venons de citer cherchent à approcher la matrice noyau. Cela a pour conséquence, entre autre, que le traitement de nouveaux points de données passe par l'utilisation de méthodes qui, même approchées, sont potentiellement lourdes telles que la méthode de Nyström [43].

Plus récemment, des méthodes utilisant des résultats théoriques sur la structure des fonctions de re-description ont été proposées. Dans [97], Rahimi *et al*. utilisent le théorème de Bochner [103] pour approcher la fonction de re-description de noyaux invariants par translation (en particulier le noyau Gaussien), à partir d'un échantillonnage aléatoire dans le domaine de Fourier. Dans [79], Li *et al*. généralisent cette méthode pour des noyaux multiplicatifs adaptés aux mesures de similarité entre histogrammes.

De manière similaire, Vedaldi *et al*. [121] utilisent une expression analytique de la fonction de re-description des noyaux additifs homogènes proposée par Hein *et al*. [63] en corollaire d'un théorème de Fuglede [50] sur les distances adaptées aux mesures de probabilité.

Dans [107], les auteurs définissent les noyaux à base radiale généralisés :

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp -\alpha D^2(\boldsymbol{x}, \boldsymbol{y})$$

où $D^2(\boldsymbol{x}, \boldsymbol{y})$ est une mesure de distance associée à un noyau additif homogène (le noyau χ^2 par exemple). Pour construire l'approximation de la fonction de re-description, les techniques d'approximation de [97] et [121] sont simplement appliquées l'une à la suite de l'autre.

La technique d'approximation de [121] nécessite la connaissance de la transformée de Fourier de la signature du noyau considéré. Dans la suite, nous donnons l'expression analytique de cette transformée pour le noyau de moyenne puissance qui généralise plusieurs noyaux d'usage courant.

6.2 Noyaux additifs homogènes et fonction de re-description

Les noyaux additifs sont d'usage courant lorsqu'il s'agit de traiter des vecteurs dans $\mathbb{R}^d$. Par exemple, si $\boldsymbol{x}$ et $\boldsymbol{y}$ sont deux vecteurs de $\mathbb{R}^d$, nous pouvons appliquer un noyau défini sur $\mathbb{R} \times \mathbb{R}$ pour chaque composante x_i et y_i indépendamment les unes des autres, puis sommer les résultats.

6.2.1 Définitions

Formellement nous pouvons donner la définition suivante :

Définition 6.1. *Un noyau $K : \mathcal{X}^d \times \mathcal{X}^d \to \mathbb{R}$ est un noyau additif s'il existe une fonction $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ symétrique telle que :*

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} k(x_i, y_i)$$

où $x_i \in \mathcal{X}$ représente la i-ème composante de $\boldsymbol{x} \in \mathcal{X}^d$.

Par abus de notation, nous confondrons par la suite le noyau additif K avec le noyau à partir duquel on le construit k lorsque qu'il n'y a pas d'ambiguïté. Par exemple k désignera le noyau additif, lorsqu'il sera utilisé sur des vecteurs $k(\boldsymbol{x}, \boldsymbol{y})$ (au lieu de $K(\boldsymbol{x}, \boldsymbol{y})$) et le noyau servant à le générer lorsqu'il sera utilisé sur des scalaires $k(x, y)$. Avec ces conventions nous écrirons donc :

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} k(x_i, y_i)$$

Le produit scalaire dans $\mathbb{R}^d$ et le noyau χ^2 sont deux exemples de noyaux additifs. Nous remarquons par ailleurs que si k est un noyau défini positif alors K est aussi un noyau défini positif.

L'homogénéité d'un noyau décrit la manière dont la valeur du noyau est modifiée par un changement d'échelle de ses variables :

Définition 6.2. *Un noyau k est α-homogène si et seulement si :*

$$k(c\boldsymbol{x}, c\boldsymbol{y}) = c^{\alpha} k(\boldsymbol{x}, \boldsymbol{y})$$

pour tout réel $c > 0$.

Nous remarquons que le noyau linéaire est un noyau 2-homogène :

$$k_{\text{lin}}(cx, cy) = (cx)(cy) = c^2 xy = c^2 k_{\text{lin}}(x, y)$$

alors que le noyau χ^2 est un noyau 1-homogène (ou simplement homogène) :

$$k_{\chi^2}(cx, cy) = 2\frac{(cx)(cy)}{cx + cy} = 2\frac{c^2 xy}{c(x+y)} = c \times 2\frac{xy}{x+y} = c k_{\chi^2}(x, y)$$

6.2.2 Fonction de re-description

Nous reproduisons à présent un résultat essentiel donné par M. Hein et O. Bousquet dans [63] :

Proposition 6.1. *Une fonction symétrique $k : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ avec $k(x, x) = 0 \Leftrightarrow x = 0$ est un noyau défini positif 2α-homogène sur $\mathbb{R}_+$ si et seulement s'il existe une mesure $\kappa \geq 0$ symétrique (nécessairement unique) bornée non-identiquement nulle sur $\mathbb{R}$ telle que k est donnée par :*

$$k(x, y) = \int_{\mathbb{R}} x^{(\alpha + i\lambda)} y^{(\alpha - i\lambda)} d\kappa(\lambda) \tag{6.1}$$

Suivant la démarche de A. Vedaldi et A. Zisserman dans [121], nous considérons à présent les mesures de la forme $d\kappa(\lambda) = \kappa(\lambda) d\lambda$, où $\kappa(\lambda)$ peut être vue comme une densité de probabilité. L'équation (6.1) peut se réécrire :

$$k(x, y) = (xy)^{\alpha} \int_{\mathbb{R}} e^{-i\lambda \log(\frac{y}{x})} \kappa(\lambda) d\lambda \tag{6.2}$$

Sous cette forme les dépendances en x et en y peuvent être factorisées et nous obtenons une forme explicite de la fonction de re-description.

Proposition 6.2. *La fonction de re-description d'un noyau 2α-homogène peut s'écrire :*

$$\phi(x)(\lambda) = \sqrt{\kappa(\lambda)} x^{\alpha} e^{-i\lambda \log x} \tag{6.3}$$

et le noyau peut s'écrire comme le produit scalaire suivant :

$$k(x, y) = < \phi(x), \phi(y) > = \int_{\mathbb{R}} \phi(x)(\lambda)^* \phi(y)(\lambda) d\lambda \tag{6.4}$$

où * représente la conjugaison complexe.

Nous remarquons qu'ici, l'espace $\mathcal{H}$ dans lequel on projette les données est un espace de fonction.

Démonstration. La démonstration est évidente. Partant de l'équation (6.2), remarquons que

$$e^{-i\lambda \log(\frac{y}{x})} = e^{-i\lambda(\log y - \log x)} = e^{+i\lambda \log x} e^{-i\lambda \log y}$$

La factorisation en utilisant l'équation (6.3) suit naturellement. □

Noyau	$k(x,y)$	$\mathcal{K}(\omega)$	$\kappa(\lambda)$
Battacharrya	$\sqrt{xy}$	1	$\delta(\lambda)$
χ^2	$\frac{2xy}{x+y}$	$\operatorname{sech}(\frac{\omega}{2})$	$\operatorname{sech}(\pi\lambda)$
intersection	$\min(x,y)$	$e^{-\frac{\lvert\omega\rvert}{2}}$	$\frac{1}{2}\frac{1}{1+4\lambda^2}$

TABLE *6.1: Différents noyaux et leurs signatures*

Nous disposons à présent d'une formule analytique pour la fonction de re-description. Cependant cette formule repose sur l'existence d'une fonction $\kappa(\lambda)$ qui reste à déterminer. Pour ce faire, nous avons utilisé la propriété d'homogénéité du noyau et utilisons le résultat suivant :

Proposition 6.3. *Soit $k : \mathbb{R}_+ \times \mathbb{R}_+ \mapsto \mathbb{R}_+$ un noyau défini positif 2α-homogène. k peut toujours s'écrire sous la forme :*

$$k(x,y) = (xy)^\alpha \mathcal{K}\left(\log \frac{y}{x}\right) \tag{6.5}$$

où $\mathcal{K}(\omega) = k(e^{+\frac{\omega}{2}}, e^{-\frac{\omega}{2}})$.

Vedaldi et Zisserman [121] appellent la fonction $\mathcal{K}$ la signature du noyau.

Démonstration. Le noyau k est 2α-homogène donc $k(cx,cy) = c^{2\alpha}k(x,y)$. Posons $c = \sqrt{xy}$, il vient :

$$\begin{aligned} k(x,y) &= k(\sqrt{xy}\sqrt{\tfrac{x}{y}}, \sqrt{xy}\sqrt{\tfrac{y}{x}}) \\ &= (\sqrt{xy})^{2\alpha} k\left(\sqrt{\frac{x}{y}}, \sqrt{\frac{y}{x}}\right) \\ &= (xy)^\alpha k\left(e^{\frac{1}{2}\log\frac{y}{x}}, e^{-\frac{1}{2}\log\frac{y}{x}}\right) \end{aligned}$$

or $k\left(e^{\frac{1}{2}\log\frac{y}{x}}, e^{-\frac{1}{2}\log\frac{y}{x}}\right) = \mathcal{K}(\log\frac{y}{x})$ d'où :

$$k(x,y) = (xy)^\alpha \mathcal{K}\left(\log \frac{y}{x}\right)$$

□

En identifiant les équations (6.5) et (6.2) et en posant $\omega = \log\frac{y}{x}$, nous trouvons :

$$\mathcal{K}(\omega) = \int_{\mathbb{R}} e^{-i\lambda\omega}\kappa(\lambda)d\lambda \tag{6.6}$$

Ce qui signifie que la signature $\mathcal{K}$ est la transformée de Fourier de la densité de probabilité $\kappa(\lambda)$. Réciproquement nous avons donc :

Proposition 6.4. *La densité de probabilité $\kappa(\lambda)$ est la transformée de Fourier inverse de la signature $\mathcal{K}(\omega)$ du noyau 2α-homogène k.*

$$\kappa(\lambda) = \frac{1}{2\pi}\int_{\mathbb{R}} e^{i\lambda\omega}\mathcal{K}(\omega)d\omega \tag{6.7}$$

Connaissant un noyau et sa signature, il est donc facile de calculer la densité κ correspondante soit analytiquement soit numériquement. Pour un certain nombre de noyaux couramment utilisés, la transformée de Fourier de leur signature est bien connue. Le tableau 6.1 donne la signature et la densité de probabilité κ correspondante pour le noyau de Battacharrya (Hellinger), le noyau χ^2 et le noyau intersection ($\min$).

6.2.3 Fonction de re-description approchée

L'application caractéristique projette un élément de $\mathbb{R}_+$ dans un espace de fonctions. Si cette représentation n'est pas exploitable directement, nous pouvons en revanche construire une approximation de cette application en échantillonnant la fonction $\phi(x)(\lambda)$ selon λ.

Plus précisément, Vedaldi *et al.* [121], montrent qu'on peut calculer une approximation $\hat{\mathcal{K}}(\omega)$ de la signature du noyau par :

$$\hat{\mathcal{K}}\left(\log\frac{y}{x}\right) = \sum_{j=-n}^{n} L\kappa(jL)e^{-ijL\log\frac{y}{x}}$$

comme la transformée de Fourier de la fonction $\kappa(\lambda)$ échantillonnée avec un pas d'échantillonnage L et tronquée.

Comme $\kappa(\lambda)$ est une fonction paire :

$$\begin{aligned}
\hat{\mathcal{K}}\left(\log\frac{y}{x}\right) &= L\kappa(0) + 2\sum_{j=1}^{n} L\kappa(jL)\cos(jL\log\frac{y}{x}) \\
&= L\kappa(0) + 2\sum_{j=1}^{n} L\kappa(jL)\cos(jL(\log y - \log x)) \\
&= L\kappa(0) + 2\sum_{j=1}^{n} L\kappa(jL)\left(\cos(jL\log x)\cos(jL\log y) + \sin(jL\log x)\sin(jL\log y)\right)
\end{aligned}$$

donc en posant :

$$\Phi(x)_j = \begin{cases} \sqrt{x^\alpha \hat{\kappa}_0} & j = 0 \\ \sqrt{2x^\alpha \hat{\kappa}_{\frac{j+1}{2}}}\cos\left(\frac{j+1}{2}L\log x\right) & j > 0 \text{ impair} \\ \sqrt{2x^\alpha \hat{\kappa}_{\frac{j}{2}}}\sin\left(\frac{j}{2}L\log x\right) & j > 0 \text{ pair} \end{cases} \tag{6.8}$$

pour $j = 0 \ldots 2n$, avec $\hat{\kappa}_j = L\kappa(jL)$. Il est facile de vérifier que :

$$< \Phi(x), \Phi(y) >= (xy)^\alpha \hat{\mathcal{K}}\left(\log\frac{x}{y}\right)$$

représente une approximation de la fonction noyau.

6.3 Le noyau de la moyenne puissance

Dans cette section, nous introduisons le noyau de la moyenne puissance [131] et montrons comment toute une classe de noyaux usuels peut être décrite par ce noyau. De plus nous montrons qu'il est possible de trouver la forme analytique de la fonction de densité $\kappa(\lambda)$ du noyau.

6.3.1 Moyenne généralisée

Nous nous intéressons ici à une famille de moyenne qui généralise plusieurs moyennes courantes.

Il existe dans la littérature plusieurs types de moyennes. Les plus connues sont la moyenne arithmétique, la moyenne géométrique et la moyenne harmonique. De manière générale, la moyenne $M(x, y)$ de deux nombres x et y peut être définie à partir d'une fonction bijective $f : \mathbb{R} \to \mathbb{R}$ par :

$$M(x, y) = f^{-1}\left(\frac{f(x) + f(y)}{2}\right) \tag{6.9}$$

Lorsque x et y sont positifs, un choix intéressant pour f est la fonction puissance $f(x) = x^\delta$ car suivant les valeurs de δ elle recouvre plusieurs moyennes connues (voir tableau 6.2). La moyenne ainsi définie s'écrit donc :

$$M_\delta(x, y) = \left(\frac{x^\delta + y^\delta}{2}\right)^{\frac{1}{\delta}} \tag{6.10}$$

On notera que la formule (6.10) n'est pas définie en $(0, 0)$. On utilise un prolongement par continuité pour définir :

$$M_\delta(0, 0) = \lim_{y \to 0^+} M_\delta(0, y) = 0$$

La fonction est ainsi définie sur tout $\mathbb{R}_+ \times \mathbb{R}_+$.

Cette moyenne généralisée définie à partir de la fonction puissance, sera appelée par la suite la *moyenne puissance*.

Nom	Formule	Exposant δ
Max	$\max(x, y)$	$\delta \to \infty$
Arithmétique	$\frac{x+y}{2}$	$\delta = 1$
Géométrique	$\sqrt{xy}$	$\delta \to 0$
Harmonique	$\frac{2xy}{x+y}$	$\delta = -1$
Min	$\min(x, y)$	$\delta \to -\infty$

TABLE *6.2: Différentes moyennes réalisables grâce à la fonction puissance*

6.3.2 Le noyau de la moyenne puissance

Pour $\delta < 0$, la moyenne puissance correspond à un noyau défini positif (voir l'annexe A pour une démonstration). Le noyau de la moyenne puissance peut se définir à partir de la définition de la moyenne puissance :

Définition 6.3. *Le noyau de la moyenne puissance* $k_\gamma : \mathbb{R}_+ \times \mathbb{R}_+ \mapsto \mathbb{R}$ *est défini par*

$$k_\gamma(x, x') = \left(\frac{x^{-\gamma} + x'^{-\gamma}}{2}\right)^{-\frac{1}{\gamma}}, \; k_\gamma(0, 0) = 0 \tag{6.11}$$

Le noyau additif correspondant est donc :

$$k_\gamma(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{d} \left(\frac{x_i^{-\gamma} + x_i'^{-\gamma}}{2}\right)^{-\frac{1}{\gamma}}$$

La *moyenne puissance* (avec $\delta = -\gamma$) est appliquée composante par composante et on prend la somme du résultat pour toutes les composantes. Il est intéressant de remarquer que cette famille de noyaux paramétrée par $\gamma \geq 0$, correspond à des noyaux couramment utilisés pour comparer des histogrammes. Le noyau résultant étant la somme de noyaux défini positif, il est lui-même défini positif.

Par ailleurs le noyau de la moyenne puissance est *homogène*. Cela signifie qu'il a la propriété suivante :

$$k_\gamma(c\boldsymbol{x}, c\boldsymbol{x}') = ck_\gamma(\boldsymbol{x}, \boldsymbol{x}')$$

pour tout $c > 0$

Démonstration.

$$\begin{aligned} k_\gamma(c\boldsymbol{x}, c\boldsymbol{x}') &= \sum_{i=1}^{d} \left(\frac{(cx_i)^{-\gamma} + (cx_i')^{-\gamma}}{2} \right)^{-\frac{1}{\gamma}} \\ &= \sum_{i=1}^{d} \left(\frac{c^{-\gamma}(x_i^{-\gamma} + x_i'^{-\gamma})}{2} \right)^{-\frac{1}{\gamma}} \\ &= \sum_{i=1}^{d} c \left(\frac{x_i^{-\gamma} + x_i'^{-\gamma}}{2} \right)^{-\frac{1}{\gamma}} \\ &= c \sum_{i=1}^{d} \left(\frac{x_i^{-\gamma} + x_i'^{-\gamma}}{2} \right)^{-\frac{1}{\gamma}} \\ &= c k_\gamma(\boldsymbol{x}, \boldsymbol{x}') \end{aligned}$$

Donc k_γ est défini positif. □

Pour certaines valeurs de γ, ce noyau coïncide avec des noyaux d'usage courant. Notamment, les trois noyaux donnés au tableau 6.1, correspondent à des cas particuliers du noyau de la moyenne puissance. La correspondance est donnée dans le tableau 6.3 avec le nom de la moyenne correspondante.

Noyau	$k(x, y)$	γ	Moyenne
Battacharrya	$\sqrt{xy}$	$\gamma \to 0$	Géométrique
χ^2	$\frac{2xy}{x+y}$	$\gamma = 1$	Harmonique
intersection	$\min(x, y)$	$\gamma \to \infty$	Min

TABLE *6.3: Différents noyaux et leurs signatures*

6.3.3 Fonction de re-description du noyau de la moyenne puissance

Comme nous l'avons vu dans la section précédente, la densité de fonction κ correspondant à certains noyaux très utilisés est connue. En particulier, les noyaux présentés dans le tableau 6.1 sont tous des cas particuliers de la moyenne puissance. Dans cette section nous donnons la forme analytique de la densité de probabilité correspondant au noyau de la moyenne puissance. À notre connaissance, ce résultat est nouveau.

Proposition 6.5. *Le noyau de la moyenne puissance* $k_\gamma(x, y) = \left(\frac{x^{-\gamma} + y^{-\gamma}}{2} \right)^{-\frac{1}{\gamma}}$ *a pour signature*

$$\mathcal{K}_\gamma(\omega) = \operatorname{sech}^{\frac{1}{\gamma}} \left(\frac{\gamma\omega}{2} \right) \tag{6.12}$$

où $\operatorname{sech}(x) = \cosh(x)^{-1}$*, et la densité de probabilité* κ_γ *correspondante est*

$$\kappa_\gamma(\lambda) = \frac{2^{\frac{1}{\gamma}-1}}{\gamma\pi} \mathrm{B}\left(\frac{1}{\gamma}\left(\frac{1}{2} + i\lambda \right), \frac{1}{\gamma}\left(\frac{1}{2} - i\lambda \right) \right) \tag{6.13}$$

où B *est la fonction spéciale Bêta définie par* $\mathrm{B}(x, y) = \int_0^\infty \frac{t^{x-1}}{(1+t)^{x+y}} dt,\ \Re(x) > 0, \Re(y) > 0$

Démonstration. Par la proposition (6.3), nous savons que :

$$\mathcal{K}_\gamma(\omega) = k_\gamma(e^{+\frac{\omega}{2}}, e^{-\frac{\omega}{2}})$$

d'où

$$\begin{aligned}\mathcal{K}_\gamma(\omega) &= \left(\frac{e^{-\gamma\omega/2}+e^{\gamma\omega/2}}{2}\right)^{-\frac{1}{\gamma}} \\ &= \cosh^{-\frac{1}{\gamma}}(\frac{\gamma\omega}{2}) \\ &= \operatorname{sech}^{\frac{1}{\gamma}}(\frac{\gamma\omega}{2})\end{aligned}$$

Ce qui démontre l'équation (6.12).

Comme $\mathcal{K}_\gamma(\omega)$ est une fonction paire, sa transformée de Fourier inverse peut se réécrire :

$$\begin{aligned}\kappa_\gamma(\lambda) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\cos(\lambda\omega)\mathcal{K}_\gamma(\omega)d\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{\cos(\lambda\omega)}{\cosh^{\frac{1}{\gamma}}(\gamma\omega/2)}d\omega \\ &= \frac{2^{1/\gamma}}{2}\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{e^{i\lambda\omega}+e^{-i\lambda\omega}}{(e^{\gamma\omega/2}+e^{-\gamma\omega/2})^{1/\gamma}}d\omega \\ &= 2^{1/\gamma-1}\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{e^{i\lambda\omega}+e^{-i\lambda\omega}}{(e^{\gamma\omega}+1)^{1/\gamma}e^{-\omega/2}}d\omega \\ &= 2^{1/\gamma-1}\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{e^{i\lambda\omega}e^{\omega/2}+e^{-i\lambda\omega}e^{\omega/2}}{(e^{\gamma\omega}+1)^{1/\gamma}}d\omega \\ &= 2^{1/\gamma-1}\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{e^{\omega(1/2+i\lambda)}+e^{\omega(1/2-i\lambda)}}{(e^{\gamma\omega}+1)^{1/\gamma}}d\omega\end{aligned}$$

On effectue le changement de variable $t = e^{\gamma\omega}$, $dt = \gamma e^{\gamma\omega}d\omega = \gamma t d\omega$

$$\begin{aligned}\kappa_\gamma(\lambda) &= 2^{1/\gamma-1}\frac{1}{2\pi}\int_0^{\infty}\frac{t^{(1/2+i\lambda)/\gamma}+t^{(1/2-i\lambda)/\gamma}}{(t+1)^{1/\gamma}}\frac{1}{\gamma t}dt \\ &= \frac{2^{1/\gamma-1}}{\gamma}\frac{1}{2\pi}\int_0^{\infty}\frac{t^{(1/2+i\lambda)/\gamma-1}+t^{(1/2-i\lambda)/\gamma-1}}{(t+1)^{1/\gamma}}dt \\ &= \frac{2^{1/\gamma-1}}{\gamma}\frac{1}{2\pi}\left(\int_0^{\infty}\frac{t^{(1/2+i\lambda)/\gamma-1}}{(t+1)^{1/\gamma}}dt+\int_0^{\infty}\frac{t^{(1/2-i\lambda)/\gamma-1}}{(t+1)^{1/\gamma}}dt\right)\end{aligned}$$

Nous utilisons ensuite la définition de la fonction Bêta $\mathrm{B}(x,y) = \int_0^{\infty}\frac{t^{x-1}}{(1+t)^{x+y}}dx$ et la propriété $\mathrm{B}(x,y) = \mathrm{B}(y,x)$ pour écrire :

$$\kappa_\gamma(\lambda) = \frac{2^{1/\gamma-1}}{\gamma}\frac{1}{2\pi}\left(\mathrm{B}\left(\frac{1}{\gamma}(\frac{1}{2}+i\lambda),\frac{1}{\gamma}(\frac{1}{2}-i\lambda)\right)+\mathrm{B}\left(\frac{1}{\gamma}(\frac{1}{2}-i\lambda),\frac{1}{\gamma}(\frac{1}{2}+i\lambda)\right)\right)$$

et donc :

$$\kappa_\gamma(\lambda) = \frac{2^{\frac{1}{\gamma}-1}}{\gamma\pi}\mathrm{B}\left(\frac{1}{\gamma}\left(\frac{1}{2}+i\lambda\right),\frac{1}{\gamma}\left(\frac{1}{2}-i\lambda\right)\right)$$

Ce qui nous donne la forme analytique de la densité de probabilité associée au noyau de la moyenne puissance. □

La figure 6.1 montre le profil de la fonction $\kappa_\gamma(\lambda)$ du noyau de la moyenne puissance pour différentes valeurs du paramètre γ. Nous observons que l'augmentation de la valeur du paramètre γ à

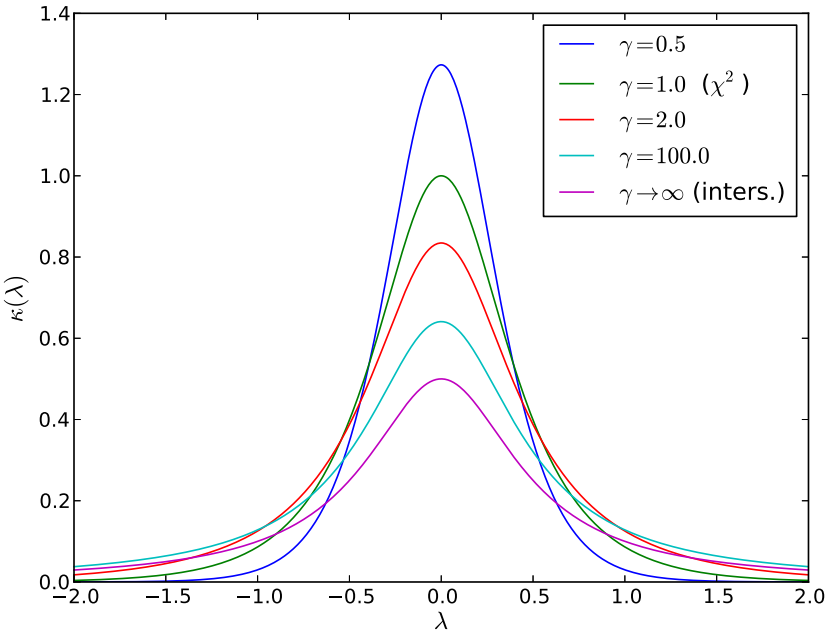


FIGURE *6.1: La fonction de densité $\kappa_\gamma(\lambda)$ du noyau de la moyenne puissance pour différentes valeurs du paramètre γ.*

pour effet un étalement de la fonction de densité dans les hautes fréquences λ. La limite haute est atteinte par la fonction de densité correspondant au noyau $\min$ (voir tableau 6.1) alors que la limite basse est donnée par la fonction Dirac correspondant au noyau de Bhattacharyya.

6.3.4 Utilisation du noyau de la moyenne puissance

Dans les expériences que nous avons menées, notamment sur la reconnaissance de visage, nous n'avons pas été en mesure de montrer une amélioration des performances liée à l'utilisation du noyau de la moyenne puissance ce qui nous a poussé à laisser de côté les investigations sur cette famille de noyaux. Récemment, Wu *et al.* [131] ont montré que l'emploi du noyau de la moyenne puissance pouvait améliorer les résultats de classifieurs SVM dans des tâches de classification à grande échelle. Leurs expériences montrent l'intérêt pratique de ce noyau. La possibilité d'en calculer une approximation est donc un avantage potentiel.

6.4 Conclusion

Les travaux développés dans ce chapitre peuvent être compris comme la tentative de contrôler la forme de la fonction de densité $\kappa(\lambda)$ du noyau afin d'obtenir des propriétés optimales pour une tâche donnée. Avec le noyau de la moyenne puissance, nous disposons d'une famille de fonctions déterminée par un paramètre unique.

Malheureusement, nos expériences n'ont pas démontré un avantage décisif à l'utilisation du noyau de la moyenne puissance. Cependant, comme le montrent les résultats obtenus par Wu *et al*. [131], le noyau de la moyenne puissance peut améliorer les performances dans certaines applications.

Malgré le manque de validation expérimentale, l'obtention de la forme analytique de la fonction de re-description du noyau de la moyenne puissance, de par son utilisation potentielle dans le cadre de l'approximation de noyaux nous parait assez intéressante pour être présentée ici.

CHAPITRE 7

Conclusion

LA reconnaissance de personnes dans les images est un problème particulièrement intéressant pour au moins deux raisons.

La première provient de son intérêt pratique. Comme nous l'avons déjà indiqué, les besoins en termes de biométrie ou d'interactions homme-machine ou encore en termes de gestion de documents multi-média sont de plus en plus grandissants et la reconnaissance de personnes – et même plus particulièrement de visages – est un enjeu crucial pour le développement de ces domaines.

La seconde réside le champ des domaines de recherche impliqués. En tant que problème de vision par ordinateur, il fait appel à des techniques de traitement des images (filtrage, recalage, extraction d'information) aussi bien qu'a des techniques d'apprentissage statistique (classification, apprentissage de distances).

Durant cette thèse, nous avons abordé ces différents aspects pour lesquels avons apporté un certain nombre de contributions originales.

Sommaire

7.1 Recalage et extraction d'informations

L'algorithme de recalage. Le recalage est la première étape des algorithmes de reconnaissance de visages [1]. Le recalage permet de limiter les effets des variations de la pose 3D des visages et de compenser les imprécisions de la détection. La méthode que nous avons mise au point, basée sur un simple algorithme de régression s'est révélée extrêmement performante et surpasse toutes les autres méthodes existantes déjà proposées.

Le descripteur HOLD. Un ingrédient clé de cette méthode réside dans le descripteur HOLD. Ce descripteur emprunte des ingrédients à différents descripteurs connus pour leur efficacité, comme les descripteurs SIFT, HOG ou LBP. Alors que notre utilisation de ce descripteur s'est pour l'instant limitée au recalage [2], les résultats encourageants obtenus dans ce contexte nous permettent de penser qu'il pourrait être utilisé avantageusement dans la phase de reconnaissance proprement dite.

1. Durant nos travaux, nous n'avons pas traité de la question de la détection des visges, considérée – à tort ou à raison – comme réglée.

2. bien que le descripteur soit présenté au début du mémoire, nous ne l'avons introduit que récemment et n'avons pas encore pu mesurer sa pertinence pour d'autres tâches de vision

Perspectives Quoi qu'offrant déjà de bons résultats, l'algorithme de recalage reste très perfectible. Comme la plupart des algorithmes concurrents, ses performances chutent lorsque les visages sont présentés dans des poses proches d'une vue de profil. Deux paramètres peuvent expliquer ce phénomène.

Le premier tient à la structure même du visage. En position frontale, les éléments les plus caractéristiques du visage sont parfaitement visibles alors que lorsque l'on se rapproche des poses de profil, certains d'entre eux sont masqués. Ce phénomène d'auto-occlusion pourraient être traité explicitement par une méthode générique de recalage.

La deuxième explication des baisses de performances pour les visages de profil tient dans la composition des bases de données utilisées pour l'entraînement. Dans la base *Labeled Faces in the Wild* (LFW), les visages de profil sont sous-représentés par rapport aux visages en position frontale ou proche de la position frontale. La conséquence est que les modèles de régressions appris ne sont pas capables d'atteindre ces configurations relativement rares. Une solution possible serait évidemment d'élargir la base en incorporant plus de visages de profil, mais à la vue des résultats obtenus, un traitement heuristique des cas de mauvaise détection semble possible. En effet, nous avons observé que dans les cas de mauvaise détection, certains des points-clés détectés se trouvaient en fait à la position d'autres points-clés ressemblant mais dont la position était atypique. En prenant en compte cette régularité dans les erreurs de détection, nous pensons qu'il pourrait être possible de compenser ce problème.

Une autre approche possible consiste à diviser les données d'entraînement en fonction de la pose et à entraîner des modèles spécialement pour chaque sous-partie. C'est le type d'approche utilisée par exemple dans [39]. Cela dit, ce type de méthode nécessite l'entraînement préalable de modèles capables de reconnaitre la pose et par conséquent, requiert les annotations correspondantes. La reconnaissane de la pose est d'autre part un problème encore ouvert.

7.2 L'apprentissage de distance

Les techniques d'apprentissage de distances appliquées à la reconnaissance de visages sont séduisantes car elles permettent un cadre formel qui correspond bien à l'intuition selon laquelle deux représentations d'un même visage devraient être plus «proches» que deux représentations de visages différents et ce quelle que soit la pose selon laquelle les visages sont observés.

Apprentissage mono-modal. Les techniques d'apprentissage mono-modales développées aux chapitres 3 et 4 donnent de relativement bons résultats sur LFW, sans doute parce que d'une part, comme nous l'avons déjà signalé, les visages en position presque frontale sont majoritaires et d'autre part parce que les descripteurs SIFT utilisés sont calculés localement sur des points-clés détectés automatiquement, ce qui permet d'ajouter une certaine robustesse à la pose. Encore une fois, les variations de pose et d'expression restent un défi majeur pour ce type de reconnaissance. Dans le cas concret, d'une comparaison entre un visage de profil et un visage de face, au mieux, les descripteurs sont difficilement comparables, au pire, la détection des points-clés en vue de profil ayant échoué, les descripteurs n'ont pas de sens.

Apprentissage trans-modal. La technique d'apprentissage de distance trans-modale développée au chapitre 5 permet justement de comparer des visages décrits dans des modalités différentes. Les résultats obtenus sur Multi-PIE sont encourageants. Cependant, la mise en œuvre pratique de cette approche nécessite une détection correcte de la pose et ses points-clés (quelle que soit la pose).

7.3 Les noyaux

Nous avons appliqué le «truc du noyau» à tous les algorithmes que nous avons développés, aussi bien pour le recalage que pour l'apprentissage de distance. Alors que pour le cas du recalage ou celui de l'apprentissage de distance mono-modale, l'utilisation de noyaux peut être vue comme un simple moyen d'améliorer les performances en introduisant des mesures de similarité non-linéaires, il est intéressant de constater que dans le cas trans-modal, ou dans le cas du modèle de régression utilisé pour le recalage, la version à noyau de l'algorithme est en général la seule ayant un sens. En effet, en reformulant le problème linéaire en un problème ne faisant intervenir que les comparaisons des points de données (au travers du produit scalaire), la formulation devient indépendante de la représentation choisie.

Toutefois, cette abstraction ne doit pas faire oublier que les résultats restent très dépendants du noyau choisi (et donc de la représentation). C'est pourquoi nous avons mené une réflexion sur les propriétés des noyaux couramment utilisés (6). Si nos résultats expérimentaux sont décevants, ceux obtenus par d'autres équipes [131] laissent à penser qu'il peut être utile de pousser plus loin les investigations.

L'utilisation des noyaux, de par le théorème du représentant, nous permet également de comprendre à quel point les données d'entraînement sont essentielles. Plus que leur nombre c'est leur représentativité qui conditionne les performances des modèles appris. Puisque la solution des problèmes d'optimisation rencontrés réside dans le sous-espace décrit par les données, il est essentiel que ce sous espace recouvre le mieux possible les données que nous pouvons rencontrer. Les baisses de performances rencontrées sur LFW pour les visages de profil sont, de ce point de vue, très représentatifs de ce problème.

7.4 Vers une méthode de reconnaissance robuste

Les progrès récents en matière de reconnaissance de visages sont impressionnants. Pour en juger il suffit de consulter la page *web* du site de LFW compilant les résultats expérimentaux obtenus sur cette base [3].

Les évolutions récentes. Les premiers résultats publiés pour la base LFW ont été réalisés au moment de la publication avec des performances maximales autour de 74% de bonne classification avec la méthode de Nowak et Jurie [90]. Cette méthode étant une méthode générique non spécialisée pour les visages.

L'introduction de méthodes plus spécialisées utilisant notamment une étape de détection de points-clés ont rapidement permis un gain de performance important. Parmi les méthodes rapportant les meilleures performances, les méthodes basées utilisant des techniques d'apprentissage de distances sont bien représentées [87, 136], notre méthode PCCA étant elle-même en bonne position. Les meilleures performances pour ce type de méthodes sont de 88.0% de bonne classification pour l'algorithme CSML de [87]. Une méthode différente utilisant un grand nombre de descripteurs différents et basés sur un algorithme efficace de sélection de ces descripteurs rapporte des résultats légèrement supérieurs avec 88.1%.

Finalement, les meilleurs résultats correspondent à des méthodes reposant sur l'utilisation intensive de données supplémentaires. Ces données supplémentaires pouvant être des annotations supplémentaires [75] et/ou des bases de visages annexes [135, 16]. La méthode de Berg et Belhumeur [16] rapportant les meilleurs résultats à ce jour avec 93.3%.

Ces résultats quoique très bons, restent pourtant très en dessous des 99.2% obtenus par des humains [75].

3. http://vis-www.cs.umass.edu/lfw/results.html

Information non utilisée. Dans [75], des expériences supplémentaires ont été réalisées avec les êtres humains. La première consiste à recadrer l'image pour ne conserver que le visage. Il est intéressant de voir que les performances des êtres humains chutent à 97.6%. Encore plus étonnant, en inversant le masque utilisé pour isoler le visage, et donc en retirant la partie correspondant au visage, les opérateurs humains obtiennent néanmoins un score de 94.3%, c'est-à-dire au-dessus du meilleur score obtenu par une méthode automatique. Cette expérience montre, qu'en se concentrant uniquement sur les traits du visage, une masse très importante d'information utile à la reconnaissance est négligée.

Pose, modèles et données d'entraînement. Dans [135], les auteurs avancent l'hypothèse qu'une des raisons pour lesquelles les êtres humains sont si performants est qu'ils ont pu observer au cours de leur vie, un nombre important de visages divers dans diverses conditions de pose, éclairage, etc. Ceci nous permettrait en quelque sorte de construire un modèle de visage nous permettant d'inférer l'apparence d'une personne dans une pose donnée alors qu'elle est observée dans une autre pose.

Ceci nous conduit à deux approches possibles pour la reconnaissance de visages robuste aux variations de pose :

1. La constitution de modèles permettant de générer l'apparence d'un visage vu sous une pose dans une autre pose. Les modèles 3D déformables [17] serait une méthode évidente de cette approche, pourtant leur difficulté de mise en œuvre les rend difficiles à appliquer en pratique dans le cadre d'une méthode complètement automatique. Notre tentative de recalage 2D-3D du chapitre 2 tente également d'utiliser l'apriori tridimensionnel pour compenser la pose. L'approche de Berg et Kumar [16] semble plus réaliste. À partir de points-clés détectés automatiquement, ils tentent de reconstruire une vue de face en utilisant des heuristiques *ad hoc* pour compenser les zones cachées.
2. La deuxième approche consiste à utiliser la connaissance de l'aspect du visage de nombreuses personnes sous différents points de vue pour retrouver l'aspect probable d'un nouveau visage dans une pose différente de celle où il est observé. C'est l'approche retenue par [135]. Fondamentalement, c'est aussi ce que fait notre méthode d'apprentissage trans-modal CMML. La différence fondamentale avec [135] réside dans le fait que nous n'avons pas utilisé de base extérieure dans nos expériences.

Passage à l'échelle des algorithmes. Pour appliquer ce dernier type d'approche il est donc nécessaire de disposer de bases de données variées du point de vue de la physionomie des visages aussi bien que de la pose. Des bases comme multi-PIE (voir section 1.1.3.1, page 4). tentent d'apporter ce genre de matériel mais le nombre de personnes représentées reste limité et peu varié. La nécessité de traiter un nombre important de données pose la question du passage à l'échelle des algorithmes existants et dans le cadre qui nous intéresse, des algorithmes présentés.

Nos algorithmes d'apprentissage peuvent subir plusieurs types d'optimisation pour faciliter leur passage à l'échelle. Une première approche consiste à utiliser les méthodes d'approximation de noyaux vues au chapitre 6 pour ne pas avoir à stocker des matrices noyaux de grande taille ou à calculer des fonctions noyaux complexes.

Une deuxième piste pour optimiser nos algorithmes d'apprentissage de distance consiste à les modifier pour en obtenir des versions «en ligne». Autrement dit à appliquer la méthode d'optimisation du gradient stochastique.

Comme nous l'avons montré, les pistes de recherches sont nombreuses et il reste encore beaucoup à accomplir pour obtenir un algorithme de reconnaissance de personnes robuste et universel.

ANNEXE A

Moyenne puissance : noyau défini positif

NOUS donnons ici la démonstration que la moyenne engendrée par la fonction puissance est un noyau défini positif pour les valeurs négatives de l'exposant. Nous reprenons la démonstration donnée dans [21] pour le noyau intersection.

Tout, d'abord rappelons la définition d'un noyau conditionnellement défini positif :s

Définition A.1. *Soit $\mathcal{X}$ un ensemble non vide. Un noyau K est dit conditionnellement défini positif (CDP) si et seulement s'il est symétrique et si :*

$$\sum_{j,k=1}^{n} c_j c_k K(\boldsymbol{x}_j, \boldsymbol{x}_k) \geq 0$$

pour tout $n \in \mathbb{N}^*$, $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \mathcal{X}$, $\{c_1, \ldots, c_n\} \subseteq \mathbb{R}$ avec $\sum_{j=1}^{n} c_j = 0$.

La seule différence avec un noyau défini positif consiste en une contrainte supplémentaire sur les c_j.

Nous présentons maintenant un exemple simple de noyau CDP, qui sera utile pour le reste de la démonstration.

Proposition A.1.

$$K_f(\boldsymbol{x}, \boldsymbol{x}') = f(\boldsymbol{x}) + f(\boldsymbol{x}'),\ (\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X} \times \mathcal{X}$$

est un noyau CDP pour n'importe quelle fonction f.

Démonstration. Il est évident que K_f est symétrique. Considérons $\boldsymbol{x}_1, ...\boldsymbol{x}_n \in \mathcal{X}$, et $c_1, \ldots, c_n \in \mathbb{R}$ tels que $\sum_{j=1}^{n} c_i = 0$, alors

$$\sum_{j,k=}^{n} c_j c_k (f(\boldsymbol{x}_j) + f(\boldsymbol{x}_k)) = \overbrace{\sum_{k=1}^{n} c_k}^{0} \sum_{j=1}^{n} c_j f(\boldsymbol{x}_j) + \underbrace{\sum_{j=1}^{n} c_j}_{0} \sum_{k=1}^{n} c_k f(\boldsymbol{x}_k) \ = 0$$

Donc, K_f est CDP. □

Nous rappelons à présent un résultat issu de [15], page 74, un théorème reliant les noyaux PD et CPD :

Théorème A.1. *Soit $\mathcal{X}$ un ensemble non vide et soit $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ un noyau symétrique. Alors K est CDP si et seulement si $\exp(uK)$ est DP pour tout $u > 0$.*

Nous utilisons à présent ce théorème pour montrer que l'on peut déduire une autre famille de noyaux DP qui dérivent de noyaux CDP.

Proposition A.2. *Si K est un noyau CDP à valeurs négatives alors $\frac{1}{(-K)^\delta}$ est DP pout tout $\delta \geq 0$.*

Démonstration. Il est facile de montrer le résultat suivant :

$$\frac{1}{(-s)^\delta} = \frac{1}{\Gamma(\delta)} \int_0^\infty u^{\delta-1} e^{su} du, \ s \leq 0, \ \delta > 0 \tag{A.1}$$

Du théorème 1, on en déduit que si K est un noyau CDP alors e^{Ku} est DP. Comme K est à valeurs négatives, l'intégrale dans (A.1) est finie lorsque l'on remplace s par K. Donc $\frac{1}{(-K)^\delta}$ est DP en tant qu'une somme de noyaux DP, pour $\delta > 0$. Pour $\delta = 0$, le résultat est évident. □

Nous avons maintenant à notre disposition tous les outils nécessaires pour montrer que la « moyenne puissance » est un noyau DP :

Proposition A.3. *Le noyau de la moyenne puissance*

$$K_\gamma(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{d} \left(\frac{x_i^{-\gamma} + x_i'^{-\gamma}}{2} \right)^{-\frac{1}{\gamma}}, \ (\boldsymbol{x}, \boldsymbol{x}') \in \mathbb{R}_+^d \times \mathbb{R}_+^d$$

est DP pour tout $\gamma > 0$:

Démonstration. Soit la fonction $f(x) = -\frac{1}{2} x^{-\gamma}, \ x \in \mathbb{R}_+$. D'après la proposition A.1, le noyau $K_f(x, x') = -(x^{-\gamma} + x'^{-\gamma})$ est CDP. Il est, de plus à valeurs négatives, donc, d'après la proposition A.2, le noyau $K_{\gamma,\delta} = (x^{-\gamma} + x'^{-\gamma})^{-\delta}$ est DP. Posons $\delta = \frac{1}{\gamma}$ on a :

$$K_\gamma(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{d} K_{\gamma, \frac{1}{\gamma}}(x_i, x_i')$$

Donc le noyau de la moyenne puissance K_γ est DP, puisqu'il est la somme de noyaux DP. □

ANNEXE B

Quantités remarquables et pseudo-inverse du laplacien d'un graphe

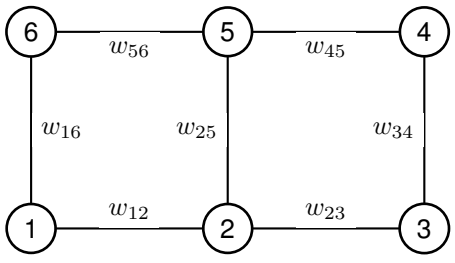
LE laplacien d'un graphe contient toute l'information sur sa connectivité. Ses propriétés en font un outil formel puissant pour l'analyse des graphes et de leurs propriétés. Nous présentons ici quelques résultats concernant la pseudo-inverse du laplacien d'un graphe et son lien avec quelques quantités remarquables des graphes et de leurs analogues physiques : les réseaux électriques résistifs.

Sommaire

B.1 Rappels et définitions

Rappelons tout d'abord la définition des quantités définies au chapitre 3. Nous considérons le cas d'un graphe non orienté $G = (\mathcal{S}, \mathcal{A})$. À toute arête $ij \in \mathcal{A}$ on associe un poids w_{ij} et on définit sa matrice d'adjacence A par :

$$[A]_{ij} = \begin{cases} w_{ij} & \text{si } ij \in \mathcal{A} \\ 0 & \text{sinon} \end{cases}$$



$$A = \begin{pmatrix} 0 & w_{12} & 0 & 0 & 0 & w_{16} \\ w_{12} & 0 & w_{23} & 0 & w_{25} & 0 \\ 0 & w_{23} & 0 & w_{34} & 0 & 0 \\ 0 & 0 & w_{34} & 0 & w_{45} & 0 \\ 0 & w_{25} & 0 & w_{45} & 0 & w_{56} \\ w_{16} & 0 & 0 & 0 & w_{56} & 0 \end{pmatrix}$$

FIGURE *B.1: Un graphe et sa matrice d'adjacence.*

Le voisinage du sommet i est $\mathcal{V}_i = \{j \in \mathcal{S} | ij \in \mathcal{A}\}$. Le degré du i-ème sommet de la matrice est défini par $d_i = \sum_{j \in \mathcal{V}_i} w_{ij}$ et la matrice des degrés est la matrice diagonale D telle que $[D]_{ii} = d_i$. Le laplacien du graphe est défini par $L = D - A$ et sa pseudo-inverse est notée L^+.

On définit la marche aléatoire sur le graphe G comme la chaîne de Markov dont les états correspondent aux sommets du graphe et où la probabilité de transition entre les états i et j est définie par :

$$p_{ij} = \begin{cases} \frac{w_{ij}}{d_i} & \text{si } j \in \mathcal{V}_i \\ 0 & \text{sinon} \end{cases}$$

On définit la matrice de transition P telle que $[P]_{ij} = p_{ij}$. On a donc $P = D^{-1}A$.

Temps moyen de premier passage. Le temps moyen de premier passage $m(j|i)$ est le nombre de sauts nécessaires en moyenne à un marcheur aléatoire pour, partant du sommet i, atteindre pour la première fois le sommet j. La quantité $m(j|i)$ peut se définir par récurrence :

$$\begin{array}{rl} m(k|k) = & 0 \\ m(k|i) = & 1 + \sum_{j \in \mathcal{V}_i} p_{ij} m(k|j) \end{array} \tag{B.1}$$

Coût moyen de premier passage. Le coût moyen de premier passage se définit comme le coût moyen qu'il faut pour, partant de i, arriver pour la première fois à j connaissant le coût $c(j|i)$ associé à la transition entre deux sommets quelconques i et j. Cette quantité peut se définir également par récurrence :

$$\begin{array}{rl} o(k|k) = & 0 \\ o(k|i) = & \sum_{j \in \mathcal{V}_i} p_{ij} c(j|i) + \sum_{j \in \mathcal{V}_i} p_{ij} o(k|j) \end{array} \tag{B.2}$$

Notons que le $m(k|i)$ s'obtient à partir de $o(k|i)$ lorsque $c(i|j) = 1$ quels que soient i et j.

Temps moyen de commutation. Le temps moyen de commutation $n(i, j)$ entre les sommets i et j est le nombre moyen de sauts nécessaire pour, partant de i, aller jusqu'à j et revenir à j. Il s'exprime en fonction du temps de premier passage par :

$$n(i, j) = m(i|j) + m(j|i) \tag{B.3}$$

Nous introduisons à présent la nouvelle quantité remarquable suivante :

Temps moyen de séjour. Le temps moyen de séjour au sommet x, dans la marche aléatoire partant de i et s'arrêtant en j est le nombre moyen de passages par le sommet x. On le note s_x^{ij} et il peut s'exprimer en fonction des temps de séjours de ses voisins :

$$s_x^{ij} = \sum_{y \in \mathcal{V}_x} p_{yx} s_y^{ij} \tag{B.4}$$

En effet, pour un voisin y de x avec un temps de séjour s_y^{ij}, $p_{yx} s_y^{ij}$ représente le nombre de fois où l'on traverse l'arête ji. En remarquant que $d_x p_{xy} = d_y p_{yx} = w_{ij}$, on trouve :

$$\frac{s_x^{ij}}{d_x} = \sum_{y \in \mathcal{V}_x} p_{xy} \frac{s_y^{ij}}{d_y} \tag{B.5}$$

Pour les points de départ et d'arrivée de la marche, le temps moyen de séjour s'exprime différemment. En effet, on quitte toujours le point de départ i une fois de plus qu'on y arrive. Par ailleurs, comme la marche s'arrête au moment où l'on atteint le point d'arrivée j, on a :

$$s_i^{ij} = 1 + \sum_{y \in \mathcal{V}_i} p_{yi} s_y^{ij}, \quad s_j^{ij} = 0 \tag{B.6}$$

On peut encore remarquer que :

$$m(j|i) = \sum_{x \in \mathcal{S}} s_x^{ij} \tag{B.7}$$

puisque le temps mis pour aller de i à j correspond bien à la somme des temps passés sur chacun des sommets du graphe.

Le temps moyen de séjour est une quantité importante car elle permet, dans un réseau, de modéliser l'encombrement d'un nœud.

B.2 Calcul en fonction des éléments de L^+

Dans [48], Fouss *et al.* montrent que le coût moyen de premier passage peut s'écrire :

$$o(k|i) = \sum_{j=1}^{n} (l_{ij}^+ - l_{ik}^+ - l_{kj}^+ + l_{kk}^+) b_j \tag{B.8}$$

avec $b_i = \sum_{j=1}^n a_{ij} c(j|i)$ et $a_{ij} = [A]_{ij}$. Pour le temps moyen de premier passage $c(i|j) = 1$ quels que soient i et j donc $b_i = \sum_{j=1}^n a_{ij} = \sum_{j \in \mathcal{V}_i} w_{ij} = d_i$ d'où :

$$m(k|i) = \sum_{j=1}^{n} (l_{ij}^+ - l_{ik}^+ - l_{kj}^+ + l_{kk}^+) d_j \tag{B.9}$$

Le temps moyen de commutation est :

$$\begin{aligned} n(i,j) &= m(i|j) + m(j|i) \qquad \text{(B.10)} \\ &= \sum_{k=1}^{n} (l_{ik}^+ - l_{ij}^+ - l_{jk}^+ + l_{jj}^+) d_k + \sum_{k=1}^{n} (l_{jk}^+ - l_{ji}^+ - l_{ik}^+ + l_{ii}^+) d_k \\ &= \sum_{k=1}^{n} (l_{jj}^+ - l_{ij}^+ - l_{ji}^+ + l_{ii}^+) d_k \\ &= (l_{jj}^+ - l_{ij}^+ - l_{ji}^+ + l_{ii}^+) \sum_{k=1}^{n} d_k \\ n(i,j) &= (l_{jj}^+ - 2l_{ij}^+ + l_{ii}^+) V_G \qquad \text{(B.11)} \end{aligned}$$

où $V_G = \sum_{i=1}^n d_i$ est le volume du graphe G.

Nous montrons à présent comment exprimer le temps de séjour en fonction des éléments de L^+ [1]. Le temps moyen de séjour en x, lors de la marche aléatoire partant de i et s'arrêtant en j s'obtient à partir du coût moyen de premier passage de l'équation (B.8) en attribuant un coût $c(y,x) = 1$ au passage de x vers n'importe lequel de ses voisins y et un coût nul dans tous les autres cas. Pour tous sommets u et v on a donc :

$$c(v|u) = \begin{cases} 1 & \text{si } u = x \wedge v \in \mathcal{V}_x \\ 0 & \text{sinon} \end{cases} \tag{B.12}$$

En combinant les équations (B.8) et (B.12) on obtient :

1. À notre connaissance c'est la première fois que cette formule est publiée.

$$
\begin{aligned}
s_x^{ij} &= o(j|i) \\
&= \sum_{k=1}^{n} (l_{ik}^+ - l_{ij}^+ - l_{kj}^+ + l_{jj}^+) \sum_{y=1}^{n} a_{yk} c(y|k) \\
&= (l_{ix}^+ - l_{ij}^+ - l_{xj}^+ + l_{jj}^+) \sum_{y \in \mathcal{V}_x}^{n} w_{xy} \\
s_x^{ij} &= (l_{ix}^+ - l_{ij}^+ - l_{xj}^+ + l_{jj}^+)\, d_x
\end{aligned}
\tag{B.13}
$$

B.3 Réseaux électriques

Il existe une relation très étroite entre les marches aléatoires sur graphes et la théorie des réseaux électriques. On considère ici des réseaux électriques résistifs, c'est-à-dire, des réseaux composés uniquement de ***résistances***. Ces réseaux sont communément représentés par un graphe (figure B.2). Si r_{ij} est la résistance entre les nœuds i et j, son inverse, la ***conductance***, est notée $c_{ij} = \frac{1}{r_{ij}}$. On définit par ailleurs pour tout nœud i la quantité C_i par :

$$C_i = \sum_{j \in \mathcal{N}_i} c_{ij}$$

Appliquons une différence de potentiel v_0 entre les nœuds i et j. Appelons alors v_x^{ij}, le potentiel au nœud x et imposons par convention que $v_j^{ij} = 0$. On a alors $v_i^{ij} = v_0$. On peut montrer [19, 42], par les lois de Kirschhoff dans les circuits électriques, que le potentiel en x est donné par :

$$C_x v_x^{ij} = \sum_{y \in \mathcal{N}_x} c_{xy} v_y^{ij}$$

soit

$$v_x^{ij} = \sum_{y \in \mathcal{N}_x} \frac{c_{xy}}{C_x} v_y^{ij} \tag{B.14}$$

En posant les équivalences suivantes :

$$
\begin{aligned}
c_{xy} &\equiv w_{xy} \\
C_x &\equiv d_x \\
v_x^{ij} &\equiv \frac{s_x^{ij}}{d_x}
\end{aligned}
\tag{B.15}
$$

L'équation (B.14) devient :

$$\frac{s_x^{ij}}{d_x} = \sum_{y \in \mathcal{N}_i} \frac{w_{xy}}{d_x} \frac{s_y^{ij}}{d_y} = \sum_{y \in \mathcal{N}_i} p_{xy} \frac{s_y^{ij}}{d_y}$$

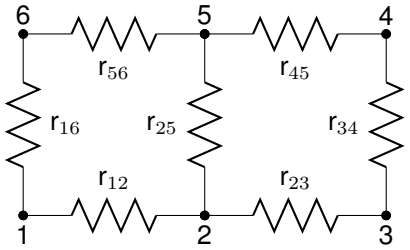
Ce qui n'est autre que l'équation (B.5).

Ceci met en évidence la relation qui existe entre les marches aléatoires et les réseaux résistifs. Le circuit équivalent au graphe de la figure B.1 est présenté à la figure B.2.

Il est intéressant de regarder à quoi correspondent d'autres grandeurs caractéristiques des réseaux résistifs en termes de marche aléatoire.

Soit I_{xy}^{ij} l'intensité qui circule dans l'arête xy lorsque la différence de potentiel v_0 est appliquée entre les nœuds i et j. La loi d'Ohms nous dit que :

$$v_x^{ij} - v_y^{ij} = r_{xy} I_{xy}^{ij}$$



$$A = \begin{pmatrix} 0 & c_{12} & 0 & 0 & 0 & c_{16} \\ c_{12} & 0 & c_{23} & 0 & c_{25} & 0 \\ 0 & c_{23} & 0 & c_{34} & 0 & 0 \\ 0 & 0 & c_{34} & 0 & c_{45} & 0 \\ 0 & c_{25} & 0 & c_{45} & 0 & c_{56} \\ c_{16} & 0 & 0 & 0 & c_{56} & 0 \end{pmatrix}$$

FIGURE *B.2: Réseau électrique équivalent à celui de la figure B.1 et la matrice d'adjacence correspondante. On a ici* $c_{ij} = 1/r_{ij}$.

soit :

$$\begin{aligned} I_{xy}^{ij} &= \frac{1}{r_{xy}}(v_x^{ij} - v_y^{ij}) && \text{(B.16)} \\ &= c_{xy}(v_x^{ij} - v_y^{ij}) && \text{(B.17)} \\ &= \frac{w_{xy}}{d_x}s_x^{ij} - \frac{w_{xy}}{d_y}s_y^{ij} && \text{(B.18)} \\ I_{xy}^{ij} &= p_{xy}s_x^{ij} - p_{yx}s_y^{ij} && \text{(B.19)} \end{aligned}$$

La quantité $p_{xy}s_x^{ij}$ correspond au nombre de fois où l'arête xy est traversée de x vers y. L'intensité indique donc la balance entre les passages dans l'arête dans un sens et dans l'autre.

Regardons le bilan des intensités partant du point de départ i :

$$\sum_{y\in\mathcal{N}_i} I_i^{ij} = \sum_{y\in\mathcal{N}_i} p_{iy}s_i^{ij} - \sum_{y\in\mathcal{N}_i} p_{yi}s_y^{ij}$$

En utilisant (B.6) on obtient :

$$\begin{aligned} \sum_{y\in\mathcal{N}_i} I_i^{ij} &= \sum_{y\in\mathcal{N}_i} p_{iy}s_i^{ij} - s_i^{ij} + 1 \\ &= s_i^{ij}\sum_{y\in\mathcal{N}_i} p_{iy} - s_i^{ij} + 1 \\ &= s_i^{ij} - s_i^{ij} + 1 \\ \sum_{y\in\mathcal{N}_i} I_i^{ij} &= 1 && \text{(B.20)} \end{aligned}$$

Cela signifie qu'avec les équivalences choisies, le courant qui circule dans le circuit est le courant unité.

La résistance efficace vue entre i et j est donc simplement :

$$\begin{aligned} R_{\text{eff}}^{ij} &= v_i^{ij} - v_j^{ij} \\ &= v_i^{ij} \\ R_{\text{eff}}^{ij} &= \frac{s_x^{ij}}{d_x} && \text{(B.21)} \end{aligned}$$

Autrement dit la différence de potentiel v_0 appliquée entre i et j vaut R_{eff}^{ij}. Si on note v_x^{ij} le potentiel en x lorsque $v_i^{ij} = R_{\text{eff}}^{ij}$ et $v_j^{ij} = 0$ et v_x^{ji} le potentiel en x lorsqu'on inverse les rôles de i et j, le principe

de superposition dans les réseaux électriques nous permet d'affirmer que $v_x^{ij} + v_x^{ji} = v_0 = R_{\text{eff}}^{ij}$. On en déduit que :

$$\frac{s_x^{ij}}{d_x} + \frac{s_x^{ji}}{d_x} = R_{\text{eff}}^{ij}$$

soit :

$$s_x^{ij} + s_x^{ji} = d_x \; R_{\text{eff}}^{ij} \tag{B.22}$$

En combinant les équations (B.10) et (B.7) on obtient :

$$n(i,j) = \sum_{x \in V} s_x^{ij} + s_x^{ji}$$

d'où

$$\begin{aligned} n(i,j) &= \sum_{x \in V} d_x \; R_{\text{eff}}^{ij} \\ &= (\sum_{x \in V} d_x) \; R_{\text{eff}}^{ij} \\ n(i,j) &= V_G \; R_{\text{eff}}^{ij} \end{aligned} \tag{B.23}$$

Ce qui montre l'équivalence entre le temps moyen de commutation et la résistance efficace (à un facteur constant près).

En réinjectant ce résultat dans (B.22) on obtient :

$$s_x^{ij} + s_x^{ji} = d_x \; \frac{n(i,j)}{V_G}$$

soit :

$$\frac{s_x^{ij} + s_x^{ji}}{n(i,j)} = \frac{d_x}{V_G} = \pi_x \tag{B.24}$$

Cela signifie que le taux moyen d'occupation du nœud x pendant la marche aléatoire de i vers j correspond simplement au taux d'occupation π_x de x de la distribution stationnaire $\boldsymbol{\pi}$ de la chaîne de Markov associée au graphe G.

Ayant obtenu le temps de séjour en fonction des éléments de L^+ grâce à la formule (B.13), les quantités vues à la section précédente peuvent être exprimées à leur tour en fonction de ces éléments, notamment :

$$v_x^{ij} = (l_{ix}^+ - l_{ij}^+ - l_{jx}^+ + l_{jj}^+) \tag{B.25}$$

$$I_{xy}^{ij} = (l_{ix}^+ - l_{jx}^+ - l_{iy}^+ + l_{jy}^+) \; w_{xy} \tag{B.26}$$

$$R_{\text{eff}}^{ij} = (l_{ii}^+ + l_{jj}^+ - 2l_{ij}^+) \tag{B.27}$$

La formule (B.27) est connue depuis 1993, et peut être démontrée par les méthodes classiques des réseaux électriques. La première démonstration est due à Klein et Randić [73]. Il faut attendre 2007 et la démonstration de Fouss et Pirotte pour une démonstration de (B.27) basée sur les marches aléatoires.

Il semble que la démonstration de la formule (B.25) a été trouvée indépendamment en 2009 par nous-même et par Cinkir [30]. La démonstration de Cinkir utilise des considérations classiques sur les réseaux électriques alors que notre démonstration est basée sur les marches aléatoires par l'intermédiaire de l'expression du temps de séjour.

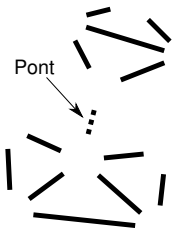


FIGURE *B.3: Un pont relie deux composantes connexes d'un graphe.*

B.4 Caractérisation des ponts d'un graphe

Dans un graphe G non orienté connexe, une arête ij est un pont si, lorsqu'on la supprime, on obtient deux composantes connexes. Une illustration est donnée à la figure B.3.

Les deux composantes sont représentées par l'ensemble des points rouges et des points hachurés respectivement. Supposons que ce graphe représente un circuit électrique et qu'un courant unité soit injecté de part et d'autre du pont. Il est évident que tout le courant doit passer par le pont. L'intensité dans celui-ci vaut donc 1. En termes de marche aléatoire cela signifie que pour aller d'une composante à une autre, il faut franchir le pont au moins une fois de plus dans un sens que dans l'autre.

La formule (B.26) nous donne donc un moyen commode pour vérifier si une arête est un pont ou non.

ANNEXE C

Optimisation sur la variété des matrices semi-définies positives

DANS [70], M. Journée P.-A. Absil et R. Sepulchre proposent une méthode générale pour résoudre les problèmes d'optimisation sur le cône des matrices semi-définies positives.

Cette approche est justement basée sur la factorisation de M en $L^T L$. La méthode proposée résout le problème d'optimisation suivant :

$$\begin{array}{ll} \min\limits_{M \in \mathbb{S}^d} & f(M) \\ \text{soumis à} & \mathrm{Tr}(A_i M) = b_i,\ A_i \in \mathbb{S}^d,\ b_i \in \mathbb{R}^d,\ i \in [1 \dots m] \\ & M \succeq 0 \end{array} \tag{C.1}$$

où f est une fonction à valeurs réelles convexe et $\mathbb{S}^d = \{M \in \mathbb{R}^{d \times d} | M = M^T\}$ est l'ensemble des matrices symétriques de $\mathbb{R}^{d \times d}$.

M. Journée *et. al* propose donc une méthode efficace en termes de coût de calcul pour résoudre le problème de l'équation C.1 lorsque les deux suppositions suivantes sont respectées.

Supposition 1. *Le problème de l'équation (C.1) à une solution M^* de rang faible, c'est-à-dire :*

$$\mathrm{rang}(M^*) = r \ll d$$

Supposition 2. *Soit le nombre m des contraintes d'égalité est 1, soit les matrices symétriques A_i satisfont :*

$$A_i A_j = 0$$

La méthode proposée repose sur la factorisation de M en $M = L^T L$ et propose de résoudre à la place du problème de l'équation (C.1) le problème suivant :

$$\begin{array}{ll} \min\limits_{L \in \mathbb{R}^{p \times d}} & f(L^T L) \\ \text{soumis à} & \mathrm{Tr}(L A_i L^T) = b_i,\ A_i \in \mathbb{S}^d,\ b_i \in \mathbb{R}^d,\ i \in [1 \dots m] \end{array} \tag{C.2}$$

Elle applique une généralisation des méthodes d'optimisation à *région de confiance*[1] sur les variétés riemaniennes, originalement proposée par P.-A. Absil, C.G. Baker et K.A. Gallivan dans [1]. L'application spécifique de ces méthodes aux variétés des matrices est par ailleurs présentée dans [2] par P.-A. Absil et R. Sepulchre.

L'idée générale derrière ces méthodes et que, s'il est difficile d'optimiser directement sur les variétés, on peut effectuer une partie des opérations dans l'espace tangent à la variété. Ce sont des méthodes itératives et à chaque itération on effectue : i) une séquence d'optimisation dans l'espace tangent à la variété au point courant, ii) une *rétraction* sur la variété, une rétraction étant une opération qui associe à tout point de l'espace un point sur la variété. La projection est un exemple de rétraction.

L'optimisation de $f(L^T L)$ présente une difficulté particulière pour les méthodes du deuxième ordre (c'est-à-dire les méthodes qui utilisent l'information de la matrice hessienne du problème) telles que les méthodes à région de confiance car ses solutions ne sont pas isolées. En effet, le problème

1. En anglais *trust region*.

est invariant par multiplication à gauche de L par une matrice orthogonale Q. Pour tenir compte de la symétrie du problème, la méthode considère conceptuellement un espace de recherche dont les points font partie de la classe d'équivalence $\{QL|Q \in \mathbb{R}^{p\times p}, Q^TQ = I\}$. En effet les solutions du problème de l'équation (C.2) peuvent être isolées dans cet *espace quotient* qui a la structure d'une variété. L'optimisation peut alors être réalisée sur cette variété.

La méthode proposée dans [70], résout le problème factorisé et augmente itérativement la valeur de k. Une fois le problème résolu pour k la matrice, on rajoute une nouvelle ligne remplie de zéro à la matrice de projection L. La matrice ainsi obtenue se trouve alors sur un point selle et on utilise le vecteur propre associé à la plus petite valeur propre de la matrice hessienne du problème non factorisé pour s'en échapper. La plus petite valeur propre donne par ailleurs un critère quant à la convergence globale de l'algorithme. En effet, à convergence cette valeur propre doit être (presque) positive.

Si la méthode de M. Journée *et al*. reste la méthode de référence, une approche plus simple peut se révéler mieux adaptée en pratique.

Tout d'abord, dans le problème qui nous intéresse à l'équation 4.7, on remarquera l'absence de contraintes de type $\mathrm{Tr}(LA_iL^T) = b_i$. En effet, ces contraintes sont généralement utilisées pour éviter des solutions triviales ainsi que pour fixer l'échelle du problème dans notre cas, l'équilibre entre les contributions des paires positives et négatives suffit à éviter les solutions triviales alors le paramètre d'échelle β fixe l'échelle (voir la discussion de la section précédente 55).

D'autre part, bien que l'approche précédente soit bien fondée et présente des garanties quant à la convergence, les temps de calculs associés restent dominés par les calculs du gradient, de la matrice hessienne et de sa plus petite valeur propre. En pratique, sur des problèmes de grande taille, une approche plus simple basée sur l'information de premier ordre uniquement se montre souvent plus rapide. Le calcul du gradient étant lui-même coûteux on privilégiera des méthodes où le gradient n'est calculé que sporadiquement.

L'utilisation d'une méthode du premier ordre uniquement a pour conséquence bénéfique de rendre l'algorithme insensible aux problèmes d'isolation des minima. Ajouté à cela l'absence des contraintes en $\mathrm{Tr}(LA_iL^T) = b_i$, l'approche basée sur les variétés n'est plus utile et on est ramené à un problème d'optimisation plus classique.

Publications

Les travaux présentés dans cette thèse ont donné lieu à plusieurs publications :

- Alexis Mignon et Frédéric Jurie. *Reconnaissance de visages, une méthode originale combinant analyse discriminante logistique et distance sur graphe*. Reconnaissance des formes et intelligence artificielle (RFIA), 2010.

- Alexis Mignon et Frédéric Jurie. *PCCA : A New Approach for Distance Learning from Sparse Pairwise Constraints*. In Computer Vision and Pattern Recognition (CVPR), 2012.

- Alexis Mignon et Frédéric Jurie. *CMML : A New Metric Learning Approach for Cross Modal Matching*. In Asian Conference on Computer Vision (ACCV), 2012.

Bibliographie

[1] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Found. Comput. Math.*, 7(3) :303–330, July 2007.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[3] Dimitris Achlioptas, Frank McSherry, and Bernhard Schölkopf. Sampling techniques for kernel methods. In *conference on annual advances in neural information processing systems*, pages 335–342. MIT Press, 2001.

[4] Yael Adini, Yael Moses, and Shimon Ullman. Face recognition : the problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 :721–732, 1997.

[5] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face Recognition with Local Binary Patterns Computer Vision - ECCV 2004. volume 3021 of *Lecture Notes in Computer Science*, chapter 36, pages 469–481. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2004.

[6] Timo Ahonen, Student Member, Abdenour Hadid, Matti Pietikäinen, and Senior Member. Face description with local binary patterns : Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 :2037–2041, 2006.

[7] Brian Amberg and Thomas Vetter. Optimal landmark detection using shape models and branch and bound. In *ICCV'11*, pages 455–462, 2011.

[8] Simon Baker and Iain Matthews. Lucas-kanade 20 years on : A unifying framework. *International Journal of Computer Vision*, 56(3) :221–255, March 2004.

[9] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning a mahalanobis metric from equivalence constraints. *J. Mach. Learn. Res.*, 6 :937–965, 2005.

[10] Ronen Basri and David Jacobs. Lambertian reflectance and linear subspaces. 25 :383–390, 2000.

[11] Peter Belhumeur and David Kriegman. What is the set of images of an object under all possible lighting conditions ? *IJCV*, 28 :270–277, 1996.

[12] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces : Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :711–720, august 1997.

[13] Peter N. Belhumeur, David W. Jacobs, David J. Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011.

[14] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, 2001.

[15] C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups : Theory of Positive Definite and Related Functions*. Springer-Verlag, 1984.

[16] Thomas Berg and Peter N. Belhumeur. Tom-vs-pete classifiers and identity-preserving alignment for face verification. In *British Machine Vision Conference*, 2012.

[17] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9) :1063–1074, 2003.

[18] Avrim Blum. Random projection, margins, kernels, and feature-selection. In Craig Saunders, Marko Grobelnik, Steve R. Gunn, and John Shawe-Taylor, editors, *Subspace, Latent Structure and Feature Selection, Statistical and Optimization, Perspectives Workshop, SLSFS 2005, Bohinj, Slovenia, February 23-25, 2005, Revised Selected Papers*, volume 3940 of *Lecture Notes in Computer Science*, pages 52–68. Springer, 2005.

[19] Béla Bollobás. *Modern Graph Theory*, volume 184 of *Graduate Texts in Mathematics*. Springer, 1998.

[20] Magnus Borga, Tomas Landelius, and Hans Knutsson. A unified approach to pca, pls, mlr and cca, 1992.

[21] S. Boughorbel, J.-P. Tarel, and N. Boujemaa. Generalized histogram intersection kernel for image recognition. In *Proceedings of IEEE International Conference on Image Processing (ICIP'05)*, volume III, pages 161–164, Genova, Italy, 2005. http ://perso.lcpc.fr/tarel.jean-philippe/publis/icip05.html.

[22] Kevin W. Bowyer, Kyong Chang, and Patrick Flynn. A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition. *Comput. Vis. Image Underst.*, 101(1) :1–15, 2006.

[23] Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (series B*, 95 :2003, 2001.

[24] Zhimin Cao, Qi Yin, Xiaoou Tang, and Jian Sun. Face recognition with learning-based descriptor. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 2707–2714, 2010.

[25] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[26] Olivier Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19 :1155–1178, 2007.

[27] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *The Journal of Machine Learning Research*, 11 :1109–1135, March 2010.

[28] Jie Chen, Dong Yi, Jimei Yang, Guoying Zhao, Stan Z. Li, and Matti Pietikäinen. Learning mappings for face synthesis from near infrared to visual light images. In *CVPR*, pages 156–163, 2009.

[29] F. R. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[30] Zubeyir Cinkir. Generalized Foster's identities. Available at : http://arxiv.org/abs/0907.3770v1, 2009.

[31] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21 :5–30, 2006.

[32] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Proceedings of the European Conference on Computer Vision*, 2 :484–498, 1998.

[33] T. F. Cootes and C. J. Taylor. Cj.taylor, "active shape models - "smart snakes. In *in Proceedings of the British Machine Vision Conference*, 1992.

[34] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 :681–685, 2001.

[35] David Cox and Nicolas Pinto. Beyond simple features : A large-scale feature search approach to unconstrained face recognition. In *Automatic Face and Gesture Recognition*, 2011.

[36] David Cristinacce and Timothy F. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, pages 929–938, 2006.

[37] David Cristinacce and Timothy F. Cootes. Boosted regression active shape models. In *BMVC*, 2007.

[38] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

[39] M. Dantone, J. Gall, G. Fanelli, and L. van Gool. Real-time facial feature detection using conditional regression forests. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2578–2585, 2012.

[40] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML '07 : Proceedings of the 24th international conference on Machine learning*, pages 209–216, New York, NY, USA, 2007. ACM.

[41] Mert Dikmen, Emre Akbas, Thomas S. Huang, and Narendra Ahuja. Pedestrian recognition with a learned metric. In *Asian Conference in Computer Vision (ACCV)*, 2010.

[42] Peter G. Doyle and J. Laurie Snell. Random walks and electric networks, 2000. arXiv.org :math/0001057.

[43] Petros Drineas and Michael W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *JOURNAL OF MACHINE LEARNING RESEARCH*, 6, 2005.

[44] Leonard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, 8 :128–140, 1741.

[45] M. Everingham, J. Sivic, and A. Zisserman. Hello ! my name is... buffy – automatic naming of characters in tv video. In *Proceedings of the British Machine Vision Conference*, 2006.

[46] Hans G. Feichtinger and Thomas Strohmer. *Gabor analysis and algorithms : theory and applications*. Birkhäuser, Boston, 1998.

[47] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vision*, 61(1) :55–79, January 2005.

[48] François Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19(3) :355–369, March 2007.

[49] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. In *In Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 370–378, 1998.

[50] B. Fuglede. Spirals in Hilbert space : With an application in information theory. *Expositiones Mathematicae*, 23(1) :23–45, April 2005.

[51] A. Globerson and S. Roweis. Metric learning by collapsing classes. *Advances in Neural Information Processing Systems*, 18 :451–458, 2006.

[52] F. Gobel and A. Jagers. Random walks on graphs. *Stochastic Processes and their Applications*, 2 :311–336, 1974.

[53] Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, 2004.

[54] Debaditya Goswami, Chi-Ho Chan, David Windridge, and Josef Kittler. Evaluation of face recognition system in heterogeneous environments (visible vs nir). In *ICCV Workshops*, pages 2160–2167, 2011.

[55] Doug Gray, Shane Brennan, and Hai Tao. Evaluating appearence models for recognition, reacquisition and tracking. In *Performance Evaluation of Tracking and Surveillance (PETS)*, IEEE International Workshop, 2007.

[56] Ralph Gross, Iain Matthews, and Simon Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11) :1080–1093, November 2005.

[57] Ralph Gross, Iain Matthews, Jeffrey F. Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image Vision Comput.*, 28(5) :807–813, 2010.

[58] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Is that you ? metric learning approaches for face identification. In *International Conference on Computer Vision*, sep 2009.

[59] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *European Conference on Computer Vision*, pages 634–647, sep 2010.

[60] Ivan Gutman and W. Xiao. Generalized inverse of the laplacian matrix and some applications. In *Bulletin : Classe des sciences mathématiques et naturelles – Siences mathématiques*, volume 129, 2004.

[61] Hu Han, Shiguang Shan, Xilin Chen, and Wen Gao. Maximizing intra-individual correlations for illumination-insensitive face recognition. In *ICIP*, pages 3833–3836, 2010.

[62] Xiaofei He and Partha Niyogi. Locality preserving projections. In *In Advances in Neural Information Processing Systems 16*. MIT Press, 2003.

[63] M. Hein and O. Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In Z. R. Cowell Ghahramani, editor, *AISTATS*, pages 136–143, 01 2005.

[64] Harold Hotelling. Relations Between Two Sets of Variates. *Biometrika*, 28(3/4) :321–377, 1936.

[65] Chang Huang, Sheng Zhu, and Kai Yu. Large scale strongly supervized ensemble metric learning with applications to face verification and retrieval. Technical Report TR115, NEC Laboratories America, 2011.

[66] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild : A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, October 2007.

[67] G.B. Huang, M.J. Jones, and E. Learned Miller. LFW results using a combined Nowak plus MERL recognizer. In *Faces in Real-Life Images Workshop in European Conference on Computer Vision (ECCV)*, 2008.

[68] Michael J. Jones and Tomaso Poggio. Multidimensional morphable models. In *ICCV*, pages 683–688, 1998.

[69] Micheal J. Jones. Face recognition : Where we are and where we go from here. *IEEJ Transactions on Electronic, Information and Systems*, 129 :770–777, 2009.

[70] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5) :2327–2351, 2010.

[71] J.G. Kemeny and J.L. Snell. *Finite Markov Chains*. Undergraduate Texts in Mathematics. Springer-Verlag, 1960.

[72] Brendan Klare, Zhifeng Li, and Anil K. Jain. Matching forensic sketches to mug shot photos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(3) :639–646, 2011.

[73] J. D. Klein and M. Randić. Resistance distance. *Journal of Mathematical Chemistry*, 12(1) :81–95, December 1993.

[74] M. P. Kumar, P. H. S. Torr, and A. Zisserman. An invariant large margin nearest neighbour classifier. In *ICCV*, 2007.

[75] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, Shree K. Nayar, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, pages 365–372, 2009.

[76] Jinho Lee, Baback Moghaddam, Hanspeter Pfister, and Raghu Machiraju. A bilinear illumination model for robust face recognition. In *ICCV '05 : Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1177–1184, Washington, DC, USA, 2005. IEEE Computer Society.

[77] Zhen Lei and Stan Z. Li. Coupled spectral regression for matching heterogeneous faces. In *CVPR*, pages 1123–1128, 2009.

[78] Annan Li, Shiguang Shan, Xilin Chen, and Wen Gao. Face recognition based on non-corresponding region matching. In *ICCV*, pages 1060–1067, 2011.

[79] Fuxin Li, Catalin Ionescu, and Cristian Sminchisescu. Random Fourier approximations for skewed multiplicative histogram kernels. In *Lecture Notes for Computer Science (DAGM)*, September 2010. DAGM paper prize.

[80] Zhifeng Li, Dahua Lin, Helen M. Meng, and Xiaoou Tang. Discriminant mutual subspace learning for indoor and outdoor face recognition. In *CVPR*, 2007.

[81] Dahua Lin and Xiaoou Tang. Inter-modality face recognition. In *ECCV (4)*, pages 13–26, 2006.

[82] Qingshan Liu, Xiaoou Tang, Hongliang Jin, Hanqing Lu, and Songde Ma. A nonlinear approach for face sketch synthesis and recognition. In *CVPR (1)*, pages 1005–1010, 2005.

[83] David G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, pages 1150–1157, September 1999.

[84] S. Mahamud and M. Hebert. The optimal distance measure for object detection. In *CVPR*, 2003.

[85] Alexis Mignon and Frédéric Jurie. Reconnaissance de visages : une méthode originale combinant analyse discriminante logistique et distance sur graphe. In *Actes de la conférence Reconnaissance de Formes et Intelligence Artificielle (RFIA)*, jan. 2010.

[86] Alexis Mignon and Frédéric Jurie. PCCA : A new approach for learning distances from sparse pairwise constraints. In *CVPR*, 2012.

[87] Hieu V. Nguyen and Li Bai. Cosine similarity metric learning for face verification. In *Proceedings of the 10th Asian conference on Computer vision - Volume Part II*, ACCV'10, pages 709–720, Berlin, Heidelberg, 2011. Springer-Verlag.

[88] Hieu V. Nguyen, Li Bai, and LinLin Shen. Local gabor binary pattern whitened pca : A novel approach for face recognition from single image per person. In *ICB*, pages 269–278, 2009.

[89] J.R. Norris. *Markov Chains*. Cambridge Series on Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.

[90] Eric Nowak and Frederic Jurie. Learning visual similarity measures for comparing never seen objects. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[91] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 24(7) :971–987, 2002.

[92] Alex Pentland, Baback Moghaddam, and Thad Starner. View-based and modular eigenspaces for face recognition.

[93] Jonathon P. Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The FERET Evaluation Methodology for Face-Recognition Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10) :1090–1104, 2000.

[94] Nicolas Pinto, James J. DiCarlo, and David D. Cox. How far can you get with a modern face recognition test set using only simple features ? In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR '09.*, 2009.

[95] William H. Press, Saul A. Teukolsky, Willial T. Vetterling, and Brian P. Flannery. *Numerical Recipes*. Cambridge University Press, 2007.

[96] Simon J.D. Prince, James H. Elder, Jonathan Warrell, and Fatima M. Felisberti. Tied factor analysis for face recognition across large pose differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30 :970–984, 2008.

[97] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. MIT Press, Cambridge, MA, 2008.

[98] C.R. Rao and S.K. Mitra. *Generalized inverse of matrices and its applications*. Wiley series in probability and mathematical statistics : Applied probability and statistics. Wiley, 1971.

[99] Vincent Rapp, Thibaud Senechal, Kevin Bailly, and Lionel Prevost. Multiple kernel learning svm and statistical validation for facial landmark detection. In *FG*, pages 265–271. IEEE, 2011.

[100] Gemma Roig, Xavier Boix Bosch, Fernando De la Torre, Joan Serrat Gual, and Carles Vilella. Hierarchical crf with product label spaces for parts-based models. In *FG'11*, pages 657–664, 2011.

[101] Volker Roth. Sparse kernel regressors. In *ICANN*, pages 339–346, 2001.

[102] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290 :2323–2326, 2000.

[103] W. Rudin. *Fourier Analysis on Groups*. John Wiley & Sons, 1990.

[104] Shai Shalev-shwartz, Yoram Singer, and Andrew Y. Ng. Online and batch learning of pseudo-metrics. In *In International Conference on Machine Learning (ICML*, pages 743–750. ACM Press, 2004.

[105] Shiguang Shan, Wen Gao, Bo Cao, and Debin Zhao. Illumination normalization for robust face recognition against varying lighting conditions. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, AMFG '03, pages 157–, Washington, DC, USA, 2003. IEEE Computer Society.

[106] Abhishek Sharma and David Jacobs. Bypassing Synthesis : PLS for Face Recognition with Pose, Low-Resolution and Sketch. In *CVPR*, pages 593–600, 2011.

[107] V. Sreekanth, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Generalized RBF feature maps for efficient detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2010.

[108] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *The British Machine Vision Conference (BMVC)*, Sept. 2009.

[109] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19(6) :1635–1650, 2010.

[110] Joshua B. Tenenbaum, Vin Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500) :2319–2323, 2000.

[111] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58 :267–288, 1994.

[112] Lorenzo Torresani and Kuang C. Lee. Large margin component analysis. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1385–1392. MIT Press, Cambridge, MA, 2007.

[113] Ivor W. Tsang, James T. Kwok, and Clear Water Bay. Distance metric learning with kernels. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 126–129, 2003.

[114] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pages 586–591, 1991.

[115] Sibt ul Hussain, Thibault Napoléon, and Frédéric Jurie. Face recognition using local quantized patterns. In *British Machine Vision Conference*, 2012.

[116] Sibt ul Hussain and Bill Triggs. Visual recognition using local quantized patterns. In *Europen Conference on Computer Vision*, 2012.

[117] M. F. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *Proceedings of IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR'10)*, pages 2729–2736, San Francisco, USA, June 2010.

[118] Vladimir Vapnik, Steven E. Golowich, and Alex Smola. Support vector method for function approximation, regression estimation, and signal processing. In *Advances in Neural Information Processing Systems 9*, pages 281–287. MIT Press, 1996.

[119] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[120] Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles : Tensorfaces. In *In Proceedings of the European Conference on Computer Vision*, volume 1, pages 447–460, 2002.

[121] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[122] Paul Viola and Michael Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57 :137–154, 2004.

[123] Ngoc-Son Vu and Alice Caplier. Enhanced patterns of oriented edge magnitudes for face recognition and image matching. *IEEE Transactions on Image Processing*, 21(3) :1352–1365, 2012.

[124] Danijela Vukadinovic and Maja Pantic. Fully automatic facial feature point detection using gabor feature based boosted classifiers. In *In SMC'05*, pages 1692–1698, 2005.

[125] Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(11) :1955–1967, 2009.

[126] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*. MIT Press, 2006.

[127] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10 :207–244, June 2009.

[128] Herman Wold. *Path models with latent variables : the NIPALS approach*, pages 307–357. Quantitative Sociology : International Perspectives on Mathematical and Statistical Modeling. Academic Press, London, academic press edition, 1975.

[129] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Real-Life Images workshop at the European Conference on Computer Vision (ECCV)*, October 2008.

[130] Lior Wolf, Tal Hassner, and Yaniv Taigman. Similarity scores based on background samples. In *Asian Conference on Computer Vision*, 2009.

[131] Jianxin Wu. Power mean svm for large scale visual classification. In *CVPR*, pages 2344–2351, 2012.

[132] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, volume 15, pages 505–512, 2002.

[133] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions : A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 :40–51, 2007.

[134] Dong Yi, Rong Liu, Rufeng Chu, Zhen Lei, and Stan Z. Li. Face matching between near infrared and visible light images. In *ICB*, pages 523–530, 2007.

[135] Qi Yin, Xiaoou Tang, and Jian Sun. An associate-predict model for face recognition. In *CVPR*, pages 497–504, 2011.

[136] Yiming Ying, Peng Li, and Peng Li. Distance metric learning with eigenvalue optimization. pages 1–26, 2012.

[137] Wei Zhang, Xiaogang Wang, and Xiaoou Tang. Coupled Information-Theoretic Encoding for Face Photo-Sketch Recognition. In *CVPR*, 2011.

[138] Wenchao Zhang, Shiguang Shan, Xilin Chen, and Wen Gao. Local gabor binary patterns based on mutual information for face recognition. *Int. J. Image Graphics*, 7(4) :777–793, 2007.

[139] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*, pages 649–656, 2011.

[140] Changtao Zhou, Zhiwei Zhang, Dong Yi, Zhen Lei, and Stan Z. Li. Low-resolution face recognition via simultaneous discriminant analysis. *Biometrics, International Joint Conference on*, 0 :1–6, 2011.

Apprentissage de métriques et méthodes à noyaux appliqués à la reconnaissance des personnes dans les images

Résumé : Nos travaux portent sur la reconnaissance des personnes dans des images vidéo en se basant principalement sur les visages. Nous nous intéressons aux étapes d'alignement et de reconnaissance, en supposant que les positions des visages dans les images sont connues.
L'alignement vise à compenser les variations de position et d'orientation des visages, les rendant plus facilement comparables. Nous présentons une méthode de détection de points-clés basée sur une régression parcimonieuse. Elle permet de prédire le décalage entre les positions moyennes et réelles d'un point-clé à partir de l'apparence de l'image autour des positions moyennes.
Nos contributions à la reconnaissance de visages reposent sur l'idée que deux représentations différentes d'une même personne devraient être plus proches, au sens d'une certaine mesure de distance, que celles de deux personnes distinctes. Nous proposons une méthode d'apprentissage de métriques vérifiant ces propriétés. L'approche est par ailleurs assez générale pour être en mesure d'apprendre une distance entre des modalités différentes.
Les modèles utilisés dans nos approches sont linéaires. Pour pallier cette limitation, ces modèles sont étendus au cas non-linéaire grâce au «truc» du noyau.
Une partie de cette thèse porte justement sur l'étude des propriétés des noyaux additifs homogènes, adaptés aux comparaisons d'histogrammes. Nous apportons notamment des résultats théoriques originaux sur la fonction de re-description du noyau de la moyenne puissance.

Mots-clés : Vision par ordinateur, Apprentissage automatique, Noyaux (analyse fonctionnelle), Perception des visages.

Metric learning and kernel methods for person recognition in images

Abstract : Our work is devoted to person recognition in video images and focuses mainly on faces. We are interested in the registration and recognition steps, assuming that the locations of faces in the images are known.
The registration step aims at compensating the location and pose variations of the faces, making them easier to compare. We present a method to predict the location of key-points based on sparse regression. It predicts the offset between average and real positions of a key-point from the appearence of the image around the average positions.
Our contributions to face recognition rely on the idea that two different representations of faces of the same person should be closer, with respect to a given distance measure, than those of two different persons. We propose a metric learning method that verifies these properties. Besides, the approach is general enough to be able to learn a distance between different modalities.
The models we use in our approaches are linear. To alleviate this limitation, they are extended to the non-linear case through the use of the kernel trick.
A part of this thesis precisely deals with the properties of additive homogeneous kernels, well adapted for histogram comparisons. We especially present some oringal theoretical results on the feature map of the power mean kernel.

Keywords : Computer Vision, Machine Learning, Kernel Functions, Face Perception.

Discipline : Informatique et applications.

Laboratoire : GREYC (UMR 6072)
Campus Côte de Nacre – Boulevard du Maréchal Juin – BP 5186 – 14032 Caen CEDEX