État de l'art: Apprentissage incrémental pour les systèmes biométriques.

Romain Giot

Apprentissage incrémental pour les systèmes biométriques

Romain Giot

9 février 2011

Table des matières

Chapitre 1

État de l'art

1.1 Introduction

Tous les systèmes de reconnaissance de formes produisent des erreurs, même minimes, de reconnaissance. Les méthodes d'authentification et d'identification biométriques reposent principalement sur des méthodes de reconnaissance de formes. Elles sont donc également sujettes à des erreurs de reconnaissance. Celles-ci se traduisent par des faux rejets d'utilisateurs authentiques, et des fausses acceptations d'imposteurs. Les raisons de ces erreurs sont multiples, en voici une liste non exhaustive. Les imprécisions sont inhérentes à la méthode de vérification utilisée, qui n'est pas capable de modéliser parfaitement un utilisateur (et, éventuellement, un imposteur), et, donc, de les différencier avec certitude (il est rarement mathématiquement possible de représenter le *modèle biométrique* d'un utilisateur). La modalité concernée peut également être sujette à une très forte variabilité *intra-classe*. Cette dernière peut être due au mécanisme bien connu de vieillissement du *modèle biométrique* (le *modèle biométrique* capturé à un instant t dérive progressivement du modèle réel à l'instant $t+n$). Drygajlo *et al.* [19] montrent qu'il y a une corrélation non négligeable entre le score de comparaison d'une requête à un modèle et le délai écoulé entre la création de ce modèle et la capture de cette requête (dans le cas de la reconnaissance faciale). Les conditions d'acquisition peuvent également être fortement différentes au cours du temps (luminosité, humidité, bruit, environnement mobile [50], ...) et peuvent affecter les performances. L'utilisation de capteurs différents lors de l'utilisation du système biométrique est également une source de variabilité importante. Ces variabilités peuvent non seulement être dues aux capteurs eux-mêmes, mais également, à la différence d'interaction entre l'utilisateur et chacun d'entre eux. Le point précédent met donc en avant la façon dont l'utilisateur agit avec le capteur. La coopération (ou plus exactement, le manque de coopération) de l'utilisateur est également une grosse source de variabilité *intra-classe*. Les modalités comportementales sont également assujetties à une forte variabilité *intra-classe* dont seul l'utilisateur est responsable. Celles-ci sont dues à l'acquisition d'un réflexe tout au long de l'utilisation du système (plus le temps passe, plus l'utilisateur maitrise le système et l'utilise différemment de sa première fois). On peut donc s'attendre à une dérive de la donnée biométrique au cours du temps. Mais, au contraire, nous pouvons voir la variabilité comme le fait que l'utilisateur effectue des actions qu'il est incapable de répéter exactement de la même façon. Dans ce cas, la variabilité sera toujours relativement importante et non corrélée avec le temps. Les autres conditions de variabilité sont plus ou moins spécifiques aux modalités concernées, nous n'allons donc pas rentrer dans les détails. Il faut noter que les variabilités peuvent être permanentes ou temporaires.

Cette diminution des performances au cours du temps peut devenir fortement gênante pour l'utilisateur. Heureusement, il existe de nombreux mécanismes pour pallier ces problèmes. Une méthode régulièrement employée en *dynamique de frappe au clavier* est la présentation d'une nouvelle capture si la première a été rejetée [6]. Malheureusement, ce type d'astuce n'aide pas à comparer objectivement les algorithmes utilisés, bien qu'elle semble être de plus en plus utilisée dans les papiers récents. La *multimodalité* est également une solution intéressante [24]. Elle permet d'utiliser plusieurs systèmes de

vérification biométrique afin de diminuer le taux d'erreur global. Cette *multimodalité* peut être effectuée à plusieurs niveaux (capteur, données, score, rang, décision). Cette pratique reste donc efficace, mais peut grandement complexifier la configuration des paramètres des différents systèmes, augmenter le cout de l'ensemble du mécanisme d'authentification et devenir moins facile à utiliser. Nous pouvons utiliser différentes caractéristiques pour représenter le *modèle biométrique*. Celles-ci sont dépendantes de la modalité et du capteur utilisé. Certaines d'entre elles peuvent être redondantes, inutiles, à forte variabilité ou à faible variabilité. Rechercher et utiliser uniquement les données stables au cours du temps permet également d'améliorer les performances des systèmes de vérification biométrique [29]. Cette technique n'est pas forcément applicable pour les modalités ne disposant pas de beaucoup de type de données extraites différentes. Les erreurs peuvent être dues à une capture insuffisante de la variabilité *intra-classe* d'un utilisateur (en raison du protocole d'enrôlement qui ne permet pas de capturer toute cette variabilité). Utiliser plusieurs sessions d'enrôlement permet de diminuer cet impact. Cependant, cette technique peut être couteuse en temps et en argent. De plus, nous ne capturons pas les variabilités présentes sur une période plus courte que l'intervalle entre deux sessions, et il est difficile de savoir quand suffisamment de variabilité a été capturée et stopper le processus d'enrôlement. Enfin, une solution qui n'a pas été intensivement testée est l'utilisation de mécanisme de *mise à jour du modèle biométrique* (*template update* ou *online learning* dans la littérature anglophone). Dans ces systèmes, le *modèle biométrique* d'un utilisateur est mis à jour au cours de l'utilisation du système. Cette mise à jour peut être faite de façon *supervisée* (ce qui revient à faire plusieurs enrôlements) ou *semi-supervisée* (c'est à dire de façon automatique), ce qui peut poser différents problèmes en raison de la possibilité d'intégrer des données d'imposteurs dans le *modèle biométrique* (et donc de diminuer les performances de reconnaissance au lieu de les améliorer).

Il existe trois types de systèmes adaptatifs dans la littérature :
- Les systèmes biométriques qui adaptent leurs paramètres en fonction de l'utilisateur (ou de la catégorie d'utilisateurs) [28] ou de la qualité de la capture [48].
- Les systèmes biométriques qui adaptent la frontière de décision au cours du temps [19].
- Les systèmes biométriques qui mettent à jour le modèle de l'utilisateur au cours de l'utilisation du système.

Dans tous les cas, l'objectif est d'obtenir les meilleures performances possible. Dans ce document, nous allons-nous focaliser sur la *mise à jour du modèle biométrique*. Vous n'y trouverez donc pas d'informations précises sur différentes modalités biométriques ou méthodes d'évaluation biométrique. Nous nous efforcerons de présenter clairement les différents points d'un système de reconnaissance biométrique prenant en compte l'évolution de la donnée biométrique au cours du temps.

Rattani *et al.* [53] présentent une taxonomie des systèmes adaptatifs. Nous l'avons reproduit dans la FIGURE 1.1. Nous pensons que ce schéma présente une réalité partielle des systèmes adaptatifs (il convient aux systèmes existants pour le visage et les empreintes, mais pas forcément pour d'autres n'ayant pas encore été étudiés). Le système est trop complexe pour être représenté sous la forme d'une taxonomie. Nous pensons qu'il est préférable de représenter un graphe des variabilités des différents paramètres afin de visualiser la complexité des systèmes adaptatifs. La FIGURE 1.2 est une proposition pour représenter la variation de ces différents éléments que nous allons décrire tout au long de ce chapitre.

Le fascicule est organisé de la façon suivante. La section 1.2 présente ce qu'est un modèle biométrique. Il nous semble important de maitriser ce concept avant de chercher à le mettre à jour. La section 1.3 présente différentes méthodes permettant de prédire ou détecter le vieillissement de la donnée biométrique afin de prendre en compte le vieillissement sans appliquer de mécanisme de mise à jour. La section 1.4 présente le fonctionnement des systèmes de mise à jour en insistant sur différents éléments : le critère de mise à jour, l'étude de la périodicité de la mise à jour, les modes de mise à jour et les stratégies de gestion du modèle. La section 1.5 présente l'évaluation des systèmes de mise à jour. La dernière section conclut cette bibliographie.

1.2 Qu'est-ce qu'un modèle biométrique ?

Avant de s'intéresser à la notion de mise à jour de *modèle biométrique*, il est indispensable de s'attarder sur la notion de *modèle biométrique*. Il est nécessaire de maitriser cette notion pour être capable de

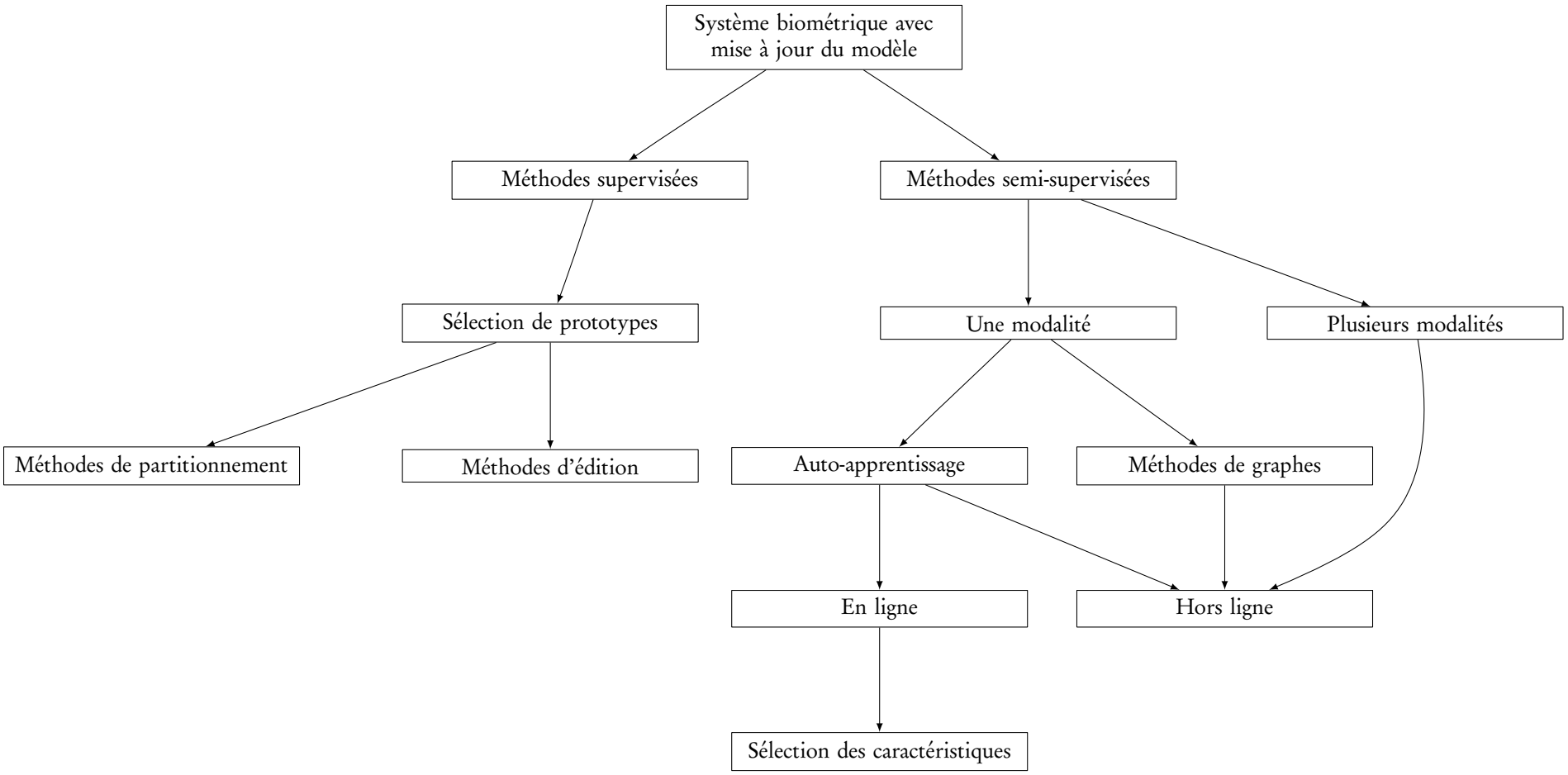


FIGURE 1.1 – Taxonomie des systèmes adaptatifs selon Rattani *et al.* [53]

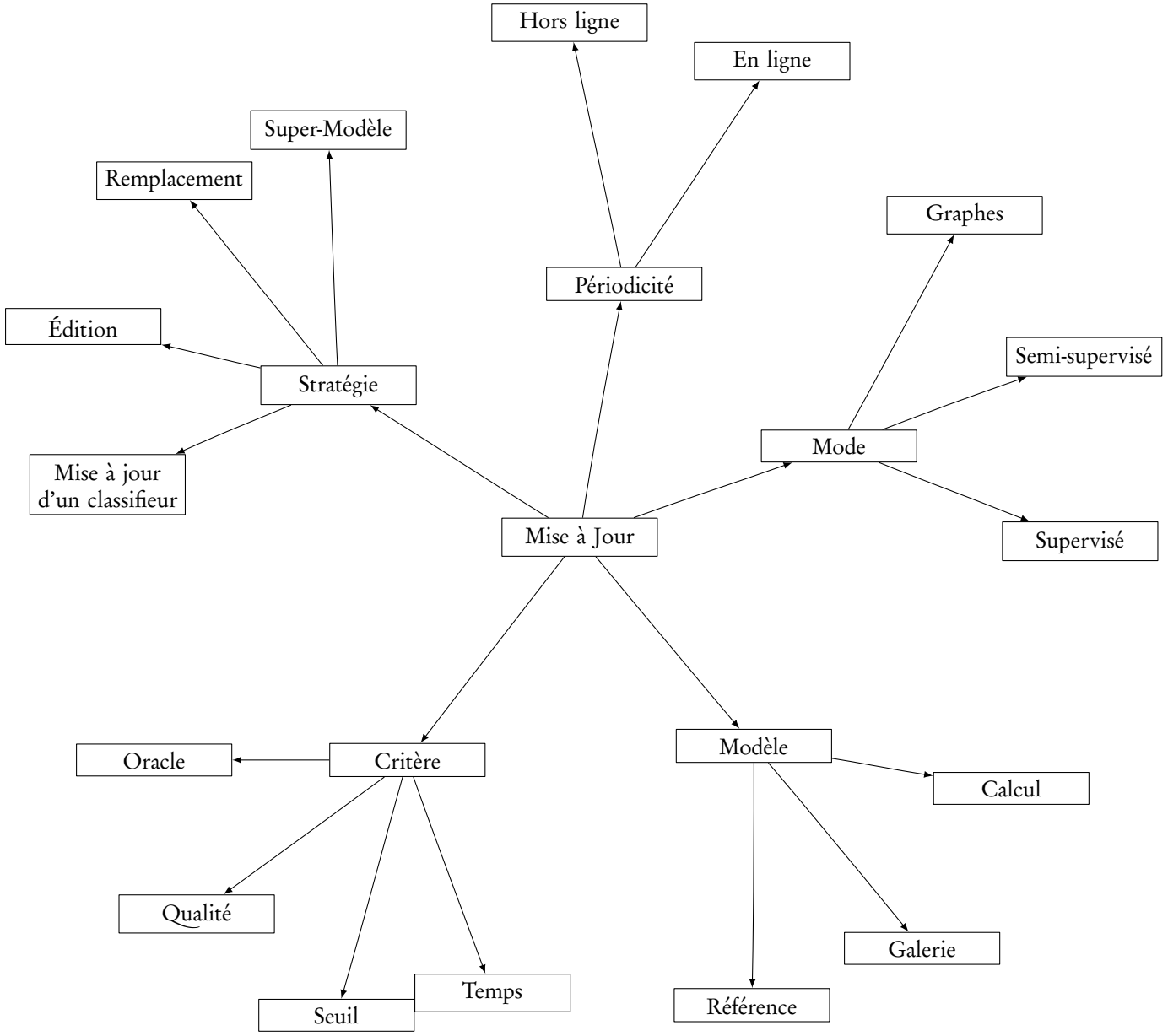


FIGURE 1.2 – Carte heuristique des différentes variations dans les paramètres de systèmes de *mise à jour du modèle biométrique*

développer des mécanismes de *mise à jour* efficaces.

1.2.1 Définition

Qu'est qu'un *modèle biométrique*? Le *modèle biométrique* est l'ensemble des données utilisées pour représenter un utilisateur. Lors d'une vérification, on compare la capture, aussi nommée *requête* (ou *query* dans la littérature anglophone) de l'individu au *modèle biométrique* de l'utilisateur qu'il clame être, afin de déterminer si elle appartient au même utilisateur. Dans la littérature, nous pouvons rencontrer les termes de *référence* en français et *reference*, *model* and *template* (ou *master template*, si *template* désigne aussi une donnée de vérification) en anglais. Pour des raisons évidentes de sécurité, il est fortement recommandé de stocker ce *modèle biométrique* de façon sécurisé, ou d'adopter des mécanismes de révocation pour diminuer les risques liés au vol de cette information. Naturellement, la nature du *modèle biométrique* est complètement dépendante de la nature de la modalité concernée, ainsi que de la méthode de vérification employée. Cependant, nous pouvons dégager plusieurs variantes communes de création de *modèle biométrique*. Nous pouvons distinguer trois catégories majeures de types de *modèle biométrique*:

- L'utilisation d'un *modèle biométrique* à *une seule référence*. Dans ce cas, une seule capture de bonne qualité est nécessaire lors de l'enrôlement. Cette unique capture sert de *référence* à l'utilisateur.
- L'utilisation de *plusieurs références* dans le *modèle biométrique*. Dans ce cas, plusieurs captures de bonne qualité ont été nécessaires lors de l'enrôlement. Nous parlons de *galerie* pour désigner l'ensemble des captures de références stockées dans le *modèle biométrique*.
- L'utilisation de *grappes de références* [42]. Il s'agit d'un cas particulier de type précédent. Les références (ou groupes de références) sont organisées sous forme hiérarchique. Chaque branche de l'arbre correspond à une contrainte particulière. Cette contrainte peut être explicite (profile, face, luminosité, ...) ou implicite (qualité de 0.1, qualité de 0.5, ...).

Cependant, cette distinction n'est pas encore totalement parfaite. Pour la plupart des méthodes d'authentification biométrique, ce ne sont pas les données brutes qui sont stockées dans le *modèle biométrique*, mais le résultat d'un calcul (*cf.* transformée de fourrier, détection de points d'intérêt, information de texture, modèle statistique sur l'ensemble des éléments de la *galerie*...). Dans le cas des systèmes à plusieurs références, le calcul peut être fait sur l'ensemble des références. Dans ce cas, le *modèle biométrique* ne correspond plus à une *galerie*, mais à une *référence unique* calculée grâce aux différents éléments de la galerie (*i.e.*, les modèles en dynamique de frappe au clavier sont majoritairement constitués des vecteurs moyenne et écart type).

1.2.2 Utilisation

Les méthodes de vérification de l'identité d'un individu sont fortement corrélées à la façon de stocker le *modèle biométrique*. En effet, dans le cas où celui-ci est simple, la méthode de vérification calcule un score. Prenons par exemple le cas de la reconnaissance faciale à une référence ; le score est le nombre de correspondances entre les points d'intérêt de l'image requête et l'image de *référence* du *modèle biométrique* [1]. Cependant, lorsque celui-ci est multiple, la méthode de vérification a la possibilité de calculer un score pour chacun de ses éléments. Dans ce cas, il est nécessaire de trouver un mécanisme d'agrégation de ces scores afin de n'en conserver qu'un seul ($minimum(\cdot)$, $moyenne(\cdot)$, ...). Pour l'exemple de la reconnaissance faciale à plusieurs *références* [2] ; il y a un score par référence. Le score maximum est retourné. Il est fort probable que les techniques de *mise à jour du modèle biométrique* soient également fortement dépendantes de la façon dont ce dernier doit être généré.

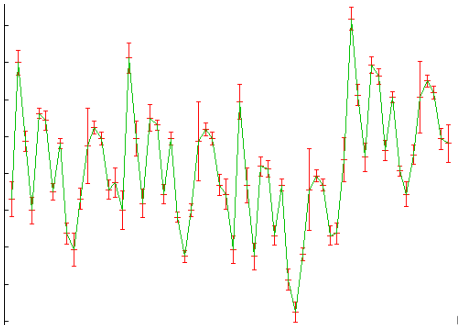
1.2.3 Stockage

Une fois calculé, il est nécessaire de stocker le *modèle biométrique*. Les quatre emplacements principaux de stockage sont : le jeton portable, la base centralisée, la machine individuelle de travail et le

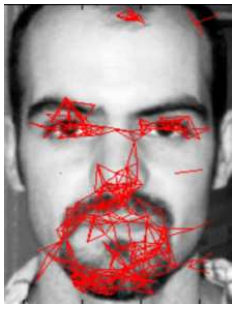
1. le *modèle biométrique* est donc un nuage de points
2. le *modèle biométrique* est une liste de nuages de points

(a) IRIS code

(b) Galerie d'empreintes digitales [75]



(c) Dynamique de frappe au clavier



(d) Graphe d'un visage [64]

FIGURE 1.3 – Exemple de modèle biométrique de différentes modalités.

capteur. Chacun de ces emplacements a ces avantages et inconvénients. Il faut noter que l'utilisation de la base centralisée est proscrite par la CNIL en France.

- Le jeton, ou la carte à puce, sont des équipements avec un microprocesseur, de la mémoire et un système d'exploitation capable d'effectuer des opérations basiques. Le système peut servir de *Template On Card (TOC)* pour héberger le *modèle biométrique* ou de *Match On Card (MOC)* pour héberger le *modèle biométrique* de l'utilisateur et effectuer la vérification dans la carte. Nous pouvons également rencontrer des *System On Card (SOC)* qui fonctionnent comme des MOC, mais intègrent en plus le système d'acquisition de données [51]. Ainsi, aucune donnée biométrique ne transite hors de la carte. Le match on card a été spécifié pour Global Platform [3], et un rapport technique du NIST présente l'intérêt de l'utilisation d'un tel système pour stocker le *modèle biométrique* pour l'administration américaine [12]. L'avantage d'un tel système est que, comme le *modèle biométrique* n'est pas stocké dans une base centralisée, il ne traverse pas le réseau, dans lequel il pourrait être capturé. D'un point de vue de l'utilisabilité, les utilisateurs ont la sensation de contrôler quand et où leur donnée biométrique est utilisée. L'inconvénient d'une telle solution est le cout relativement élevé d'une implémentation en raison de la quantité de matériel supplémentaire. Dans le cas du *TOC*, il est nécessaire de lire la carte avant que l'utilisateur ne soit authentifié, ce qui peut poser d'autres problèmes de sécurité.
- Dans le cas d'une base centralisée, le *modèle biométrique* est stocké dans un environnement nécessitant de le faire transiter sur le réseau. Cette base contient les modèles de tous les utilisateurs. L'intérêt d'un tel système est que les modèles sont centralisés dans un endroit unique, obligeant l'utilisation d'un système de communication, permettant de vérifier l'identité de l'utilisateur dans plusieurs endroits sans qu'il ait à transporter lui même son modèle (*cf.* le cas précédent). L'inconvénient majeur est la possibilité de capturer le *modèle biométrique* pendant son transport dans le réseau.
- Pour l'espace de travail, le *modèle biométrique* est stocké localement sur la machine nécessitant une authentification ou identification biométrique. L'avantage est de proposer plus de respect de

3. `http://www.globalplatform.org`

la vie privée en évitant le vol du modèle lors de son transport. Les inconvénients sont qu'il est nécessaire de créer un modèle par emplacement lorsqu'il est nécessaire que l'utilisateur s'authentifie sur plusieurs stations.

Un point complémentaire au stockage de la donnée biométrique est sa sécurisation (notamment lorsqu'il n'est pas stocké sur une carte à puce).

1.2.4 Sécurisation

Le *modèle biométrique* est un élément sensible d'un système d'authentification biométrique. En effet, lorsque cette donnée n'est pas révocable (ce qui est le cas la plupart du temps), le vol du *modèle biométrique* implique que l'attaquant puisse se faire authentifier à la place du vrai utilisateur. Une technique simple de protection du *modèle biométrique* est d'utiliser un système de « Match On Card (MOC) » [5], qui stocke le *modèle biométrique* dans une carte à puce en possession de l'utilisateur, plutôt que dans une base centralisée. C'est à la charge de la puce de vérifier l'identité de son propriétaire. Néanmoins ce système est relativement couteux (nécessité de gérer les cartes de tous les utilisateurs, de disposer de différents lecteurs de cartes) et n'est pas toujours facilement déployable. En plus de protéger le *modèle biométrique* des vols, nous obtenons un système d'authentification forte qui utilise deux éléments différents :

- *ce que l'on est* : notre donnée biométrique qui est censée nous être propre ;
- *ce que l'on possède* : la carte à puce contenant le *modèle biométrique*.

Dans le cas du MOC, la protection du *modèle biométrique* est due au stockage de celui-ci dans la carte qui est une enceinte sécurisée. Il est également possible de sécuriser un *modèle biométrique* de façon logicielle, sans utiliser de carte à puce. Jain *et al.* [30] présentent un état de l'art des différentes techniques utilisables. Nous ne rentrerons pas dans les détails de cette technique, ce n'est pas le but de ce fascicule. Les mécanismes de protection doivent satisfaire les propriétés suivantes:

1. *La diversité* : le *modèle biométrique* sécurisé de doit pas permettre la vérification croisée entre différentes bases de données, ce qui permet d'assurer le respect de la vie privée de l'utilisateur.
2. *La révocabilité* : il doit être relativement simple de pouvoir révoquer un *modèle biométrique* compromis et d'en émettre un nouveau basé sur la même donnée biométrique.
3. *La sécurité* : obtenir le *modèle biométrique* original à partir du *modèle biométrique* sécurisé doit être calculatoirement difficile. Cette propriété permet de prévenir la création d'une imitation du trait biométrique depuis un *modèle biométrique* volé.
4. *La performance* : le mécanisme de protection du *modèle biométrique* ne doit pas dégrader les performances de reconnaissance du système biométrique.

La protection du *modèle biométrique* est un domaine de recherche d'actualité.

1.2.5 Discussion

Comme nous avons pu le voir, le *modèle biométrique* est un élément essentiel du système d'authentification biométrique. Il est nécessaire que celui-ci soit de bonne qualité, afin d'avoir des performances acceptables. Il est nécessaire de comprendre le fonctionnement d'un *modèle biométrique* avant d'étudier comment il est possible de le mettre à jour. Cette mise à jour devient difficilement générique, car il y a plusieurs façons de représenter le *modèle biométrique*. Il est également possible que, dans le cas mono modal et sans fusion de méthodes utilisant la même donnée, un utilisateur possède plusieurs modèles. Liu *et al.* [40] présente un système de *mise à jour du modèle biométrique* dans le cas de l'identification faciale utilisant deux modèles ; la méthode est nommée *Twin-subspace updating scheme*. Chaque utilisateur dispose de deux modèles, l'initial et celui mis à jour. Lors d'une identification, la requête est projetée dans les espaces propres de chaque utilisateur (modèle initial et modèle mis à jour). La requête appartient à l'utilisateur ayant le résidu minimal dans l'un de ces espaces (différence entre l'image de test et sa projection). Pour décider s'il faut mettre à jour le modèle, le résidu est comparé à un premier seuil, afin de vérifier si le modèle représente suffisamment la donnée. Si c'est le cas, la mise à jour n'est pas faite (le modèle est représentatif et la donnée n'apporte pas de nouvelle information statistique). Une mesure de confiance est calculée en comparant la valeur du résidu le plus faible (l'utilisateur identifié) et celle du second résidu

le plus faible (l'utilisateur le plus proche). Si cette mesure de confiance est supérieure à un seuil, la capture est utilisée pour la mise à jour. Utiliser les deux modèles permet d'avoir un modèle statique qui capture l'apparence globale de l'individu, tandis que le modèle dynamique capture les variations. L'étude montre que les performances sont meilleures qu'avec un unique modèle dynamique. Poh *et al.* [50] suggèrent également l'utilisation de plusieurs modèles biométriques. Chacun de ces modèles étant dirigés par une mesure de qualité. Par exemple, dans un système de reconnaissance faciale, la mesure de qualité pourrait être l'orientation et l'inclinaison du visage, de telle façon à maintenir plusieurs galeries en fonction de ces angles.

1.3 Détection et Prédiction du vieillissement du modèle

1.3.1 Détection

Avant de chercher à trouver des méthodes permettant de pallier le problème de la *mise à jour du modèle biométrique*, il est nécessaire de vérifier si ce problème existe réellement pour la modalité donnée, avec le seuil de configuration donné. Schuckers [69] a mis au point deux méthodes pour vérifier de façon statistique l'existence de problème de vieillissement du *modèle biométrique*.

Generalized Linear Model Le but de ce test est de vérifier si les taux d'erreurs ont changé linéairement au cours du temps. Notons Y_i la décision pour le i^e exemple, et t_i la différence de temps, en jours, entre l'image d'enrôlement et le moment de son utilisation, avec $i = 1, \ldots, N$. Ainsi :

$$Y_i = \begin{cases} 1 & \text{si la } i^e \text{ décison est erronée} \\ 0 & \text{si la } i^e \text{ décison est correct} \end{cases} \tag{1.1}$$

Notons, π le taux d'erreur (soit Taux de Fausse Acceptation (TFA), soit Taux de Fause Non Acceptation (TFNA)) devant être estimé pour un seuil donné. π est donc estimé par $\hat{\pi} = N^{-1} \sum Y_i$. Le modèle linéaire généralisé est :

$$g(\mu_{Y_i|t_i}) = g(\pi) = \beta_0 + \beta_1 t_i \tag{1.2}$$

avec $g(z)$ le lien canonique défini de la façon suivante :

$$g(z) = logit(z) = log\left(\frac{z}{1-z}\right) \tag{1.3}$$

En raison d'une possible surdispersion, une approche de type quasibinomial est utilisée. Le calcul de la variance se fait de la façon suivante :

$$V[Y_i] = \pi(1-\pi)\phi \tag{1.4}$$

avec ϕ représentant le degré de surdispersion des données (celui-ci est donc calculé depuis les données). Avec cette méthodologie, le but est de vérifier si β_1 est statistiquement différent de zéro. Cela revient à conclure qu'il y a une modification dans les *log-odds* de l'erreur de décision au cours du temps.

Test de ratio de vraisemblance . Le but de ce test est de vérifier si le taux d'erreur change brusquement à un certain moment au cours du temps. Les tests de ratio de vraisemblance nécessitent de spécifier une fonction de densité pour les données. Schuckers [69] utilise une distribution beta-binomiale. Ainsi, pour une variable aléatoire, X :

$$\begin{cases} E[X|n,\pi,\rho] = n\pi \\ V[X|n,\pi,\rho] = n\pi(1-\pi)(1+(n-1)\rho) \end{cases} \tag{1.5}$$

avec n le nombre de décisions, π le taux de réussite et ρ la corrélation *intra-classe*. Un *succès* est défini comme une erreur (quel que soit son type), donc π est le taux d'erreur d'une décision. La probabilité d'avoir x erreurs parmi n décision est alors :

$$\begin{aligned} P(X = x|n,\pi,\rho) \quad = \quad & C_x^n \frac{\Gamma((1-\rho)\rho^{-1})}{\Gamma(\pi(1-\rho)\rho^{-1})\Gamma((1-\pi)(1-\rho)\rho^{-1})} \\ & \times \frac{\Gamma(\pi(1-\rho)\rho^{-1}+x)\Gamma((1-\pi)(1-\rho)\rho^{-1}+n-x}{\Gamma((1-\rho)\rho^{-1}+n)} \end{aligned} \tag{1.6}$$

Pour tester un changement dans le taux d'erreur, on observe si le taux d'erreur π est constant ou change sur une période de T. Soit X_j le nombre d'erreurs au j^e instant avec n_j décisions, $j = 1, \ldots, T$. Pour chaque instant j, le taux d'erreurs est π_j. Pour un $j^\star$ prédéfini, on vérifie si il y a une modification du taux d'erreurs entre le moment π_j et le moment π_{j+1}, en partant de principe que le taux d'erreur est constant avant, et après. On veut donc tester l'hypothèse nulle :

$$H_0 : \pi_1 = \pi_2 = \ldots = \pi_T \tag{1.7}$$

contre une hypothèse alternative :

$$H_a : \pi_1 = \pi_2 = \ldots = \pi_{j^\star}, \pi_{j^\star+1} = \ldots = \pi_T \tag{1.8}$$

Sous l'hypothèse nulle, H_0, la probabilité est alors :

$$\begin{array}{rcl} L_0 & = & \prod_{j=1}^{T} C_{x_j}^{n_j} \frac{\Gamma((1-\rho)\rho^{-1})}{\Gamma(\pi(1-\rho)\rho^{-1})\Gamma((1-\pi)(1-\rho)\rho^{-1})} \\ & \times & \frac{\Gamma(\pi(1-\rho)\rho^{-1}+x_j)\Gamma((1-\pi)(1-\rho)\rho^{-1}+n_j-x_j}{\Gamma((1-\rho)\rho^{-1}+n_j)} \end{array} \tag{1.9}$$

Tandis que sous l'hypothèse alternative :

$$\begin{array}{rcl} L_0 & = & \prod_{j=1}^{j^\star} C_{x_j}^{n_j} \frac{\Gamma((1-\rho_1)\rho_1^{-1})}{\Gamma(\pi_1(1-\rho_1)\rho_1^{-1})\Gamma((1-\pi_1)(1-\rho_1)\rho_1^{-1})} \\ & \times & \frac{\Gamma(\pi_1(1-\rho_1)\rho_1^{-1}+x_j)\Gamma((1-\pi_1)(1-\rho_1)\rho_1^{-1}+n_j-x_j}{\Gamma((1-\rho_1)\rho_1^{-1}+n_j)} \\ & \times & \prod_{j=j^\star+1}^{T} C_{x_j}^{n_j} \frac{\Gamma((1-\rho_T)\rho_T^{-1})}{\Gamma(\pi_T(1-\rho_T)\rho_T^{-1})\Gamma((1-\pi_T)(1-\rho_T)\rho_T^{-1})} \\ & \times & \frac{\Gamma(\pi_T(1-\rho_T)\rho_T^{-1}+x_j)\Gamma((1-\pi_T)(1-\rho_T)\rho_T^{-1}+n_j-x_j}{\Gamma((1-\rho_T)\rho_T^{-1}+n_j)} \end{array} \tag{1.10}$$

Schuckers adopte comme principe que les ρ_j sont les miroirs de π_j. L_0 et L_a sont donc évalués à leur valeur maximum en fonction de leurs paramètres respectifs. En suivant la théorie des tests de ratio de vraisemblance, on doit rejeter l'hypothèse nulle d'un taux d'erreur constant en faveur de l'alternative générale si la p-value ($= P(\chi^2 > \lambda)$ est faible (sous un signe significatif $\alpha = 0,05$, avec χ^2 une variable aléatoire suivant la loi du khi carré avec $4 - 2 = 2$ degrés de liberté et $\lambda = -2 * (\ln(L_a) - ln(L_0))$.

Discussion Les deux méthodes ont été validées sur la base de scores du NIST [4] sur différentes implémentations de reconnaissance faciale et d'empreintes digitales en utilisant différents seuils de décision (avec comme seuil de signification statistique, $\alpha = 0,05$). Les résultats montrent que les deux méthodes ne détectent pas forcément des problèmes de vieillissement au même moment (ce qui n'est pas surprenant étant donné qu'ils ne testent pas les mêmes propriétés). Les résultats ne sont pas constants en fonction des paramètres : autrement dit, la configuration du seuil joue sur les effets du vieillissement. Les systèmes de reconnaissance faciales sont plus sujet au vieillissement que les systèmes de reconnaissance d'empreintes digitales. Il ne semble pas que d'autres travaux similaires existent dans la littérature. Ramanathan et Chellappa [52] présentent un framework de reconnaissance faciale qui permet, à la fois, de distinguer un imposteur d'un individu authentique, et de donner l'intervalle de temps (en nombre d'années) entre la collecte de l'image de référence et de l'image de test. Le cas d'utilisation évoqué est l'utilisation d'un système de reconnaissance lié au passeport. Cependant, il n'y a pas de système lié à l'apprentissage incrémental dans cette revue.

1.3.2 Anticipation du vieillissement

L'utilisation d'un mécanisme de *mise à jour du modèle biométrique* n'est pas le seul moyen de prendre en compte le phénomène de vieillissement du modèle. Plutôt que de mettre à jour le *modèle biométrique* au cours du temps, il est également possible d'avoir une frontière de décision (lors de la comparaison d'une requête à un modèle) qui est variable au cours du temps.

4. http://www.itl.nist.gov/iad/894.03/biometricscores/

1.3.2.1 Évolution de la frontière de décision

Li *et al.* [38] ont récemment étudié un tel système dans le cas d'un système de reconnaissance faciale à une référence. Leur étude est validée sur une base de 42 utilisateurs (21 pour l'apprentissage, 21 pour le test) ayant fournis plusieurs photos pendant deux ans. Deux classifieurs sont empilés l'un à la suite de l'autre [78] :

- le premier classifieur (qui prend en entrée la requête), retourne un score de comparaison entre la requête et le modèle ;
- le second classifieur (qui prend en entrée, le score du classifieur précédent et la différence de temps entre les captures de la requête et du modèle, après normalisation), retourne la décision d'acceptation. Son rôle est de corriger les erreurs potentielles du premier classifieur.

De cette façon, la décision est dépendante de l'intervalle de temps entre les deux captures. Le système développé propose un taux d'erreur moyen inférieur au système de base. Le taux de fausses acceptations est nettement inférieur, mais le taux de faux rejets tend à être supérieur. Ainsi, leur méthode permet de prendre en compte l'évolution des scores au cours du temps, sans avoir à utiliser de mécanisme de mise à jour du *modèle biométrique*. Leur étude montre que le score des clients a tendance a augmenter au cours du temps. Dans [19], les mêmes auteurs ont montré qu'il y a une corrélation entre le score de de comparaison et l'intervalle de temps entre la capture de la requête et la création du modèle. La FIGURE 1.4 présente une généralisation de ce système. Plus l'écart entre la date d'acquisition du modèle et de la requête est important, plus le système montre son intérêt face à un mécanisme de prise en compte de ce vieillissement.

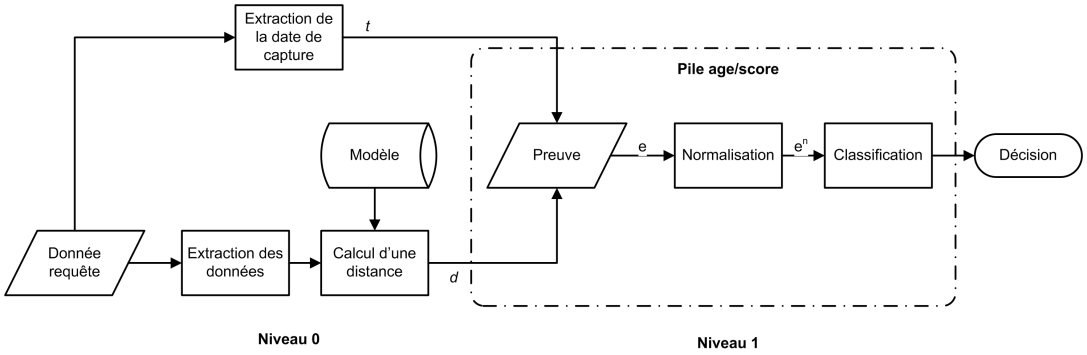


FIGURE 1.4 – Architecture du mécanisme de classification par empilement proposé dans [38].

1.3.2.2 Vieillissement artificiel du modèle

De nombreuses méthodes existent pour faire vieillir artificiellement un visage. Sethuram *et al.* [70] montrent qu'ajouter des données artificiellement vieillies dans l'ensemble d'apprentissage augmente les performances. Les auteurs ont utilisé les bases MORPH [59] et FG-NET [1] pour leur expérience. L'ensemble d'apprentissage est constitué de visages des utilisateurs entre 18 et 30 ans. Trois ensembles de test sont constitués, avec des images dans un intervalle d'age différent : 18-30, 31-40 et 41-50. Les résultats sont nettement moins bons pour les deux derniers intervalles que pour le premier. En effet, les individus ayant vieilli de plusieurs (dizaines d') années, il est plus difficile pour les systèmes d'identification de les reconnaitre. Les auteurs ont pu augmenter les performances du système d'identification (performance au rang 1 de 31.25% au lieu de 18.75%) en ajoutant des données vieillies artificiellement à la galerie.

1.3.3 Discussion

Nous avons vu dans cette section qu'il est possible de prédire le vieillissement du *modèle biométrique*. Cette connaissance du vieillissement peut être utilisée afin de connaitre le moment où il est nécessaire de mettre à jour le *modèle biométrique*. Mais, elle peut également être utilisée dans un autre cadre, afin

d'améliorer les performances sans avoir à calculer un nouveau modèle. Il est possible de changer la frontière de décision au cours du temps, afin de diminuer le taux de faux rejets des utilisateurs (avec un impact minimum sur le taux de fausse acceptations des imposteurs). Il est également possible de vieillir artificiellement le modèle, afin de comparer la requête à ce modèle artificiel, plutôt qu'au modèle réel qui est considéré comme obsolète. Il faut noter que, à notre connaissance, ces procédures ont uniquement été testées dans le cas de systèmes de reconnaissance faciale.

1.4 Mise à jour du modèle biométrique

Cette partie présente les différents mécanismes de *mise à jour du modèle biométrique*. Nous allons voir que de nombreux paramètres sont différents en fonction des études et des algorithmes employés. Ces paramètres sont les suivants :

1. le choix du *critère de mise à jour du modèle biométrique* ;
2. la *périodicité* (en ligne ou hors ligne) de *mise à jour du modèle biométrique* qui est dépendante du point précédent ;
3. le *mode* (supervisé ou semi-supervisé) de fonctionnement du mécanisme de *mise à jour du modèle biométrique*;
4. la *stratégie* (l'algorithme utilisé) de *mise à jour du modèle biométrique*;
5. la *technique* (la façon d'intégrer les modifications dans le *modèle biométrique*) de *mise à jour du modèle biométrique*.

1.4.1 Critères de mise à jour

Pour qu'un système puisse mettre à jour un *modèle biométrique*, il est nécessaire qu'il puisse prendre la décision d'appliquer, ou pas, cette mise à jour. Différentes techniques de décision de mise à jour existent dans la littérature.

1.4.1.1 Acceptation de la requête

Se baser uniquement sur l'acceptation de la capture *requête* par le système n'est pas une solution satisfaisante : une donnée mal étiquetée (autrement dit, une capture d'imposteur) serait trop facilement ajoutée au modèle de l'utilisateur, et, il en résulterait une déviance non attendue du nouveau *modèle biométrique* en comparaison de ce qu'il devrait être (l'utilisateur sera moins facilement accepté, tandis que les imposteurs pourraient l'être plus facilement.). C'est probablement pour cette raison que nous ne rencontrons pas ce critère dans la littérature.

1.4.1.2 Double seuillage

Une technique, couramment utilisée dans la littérature, est d'utiliser un système à double seuillage :
- le *premier seuil* consiste à déterminer si une requête est de type client ou imposteur ;
- tandis que *le deuxième* consiste à décider si le motif correspond suffisamment à un client pour pouvoir être ajouté dans son modèle.

Ce double seuillage est souvent noté de la façon suivante dans la littérature : la requête a une *forte probabilité* d'être une donnée cliente. Cependant, la majorité du temps, il n'y a guère plus d'information sur la configuration du seuil. On peut s'attendre à ce que celui-ci soit configuré de telle façon à obtenir un taux de fausse acceptation (ou faux rejet) d'une certaine valeur depuis une base de test [55]. Il semble que ce soit cette sélection qui est la plus couramment utilisée dans la littérature. Dans le cas de la configuration d'un seuil automatique (*cf.* de telle façon à avoir un TFA de 1%, par exemple), il est nécessaire que le système biométrique dispose de données d'imposteur. Il est trivial d'en obtenir dans le cas d'une modalité morphologique, mais c'est loin d'être le cas pour les biométries demandant aux utilisateurs d'effectuer une action particulière (*cf.* saisie d'un mot de passe, pour la dynamique de frappe au clavier).

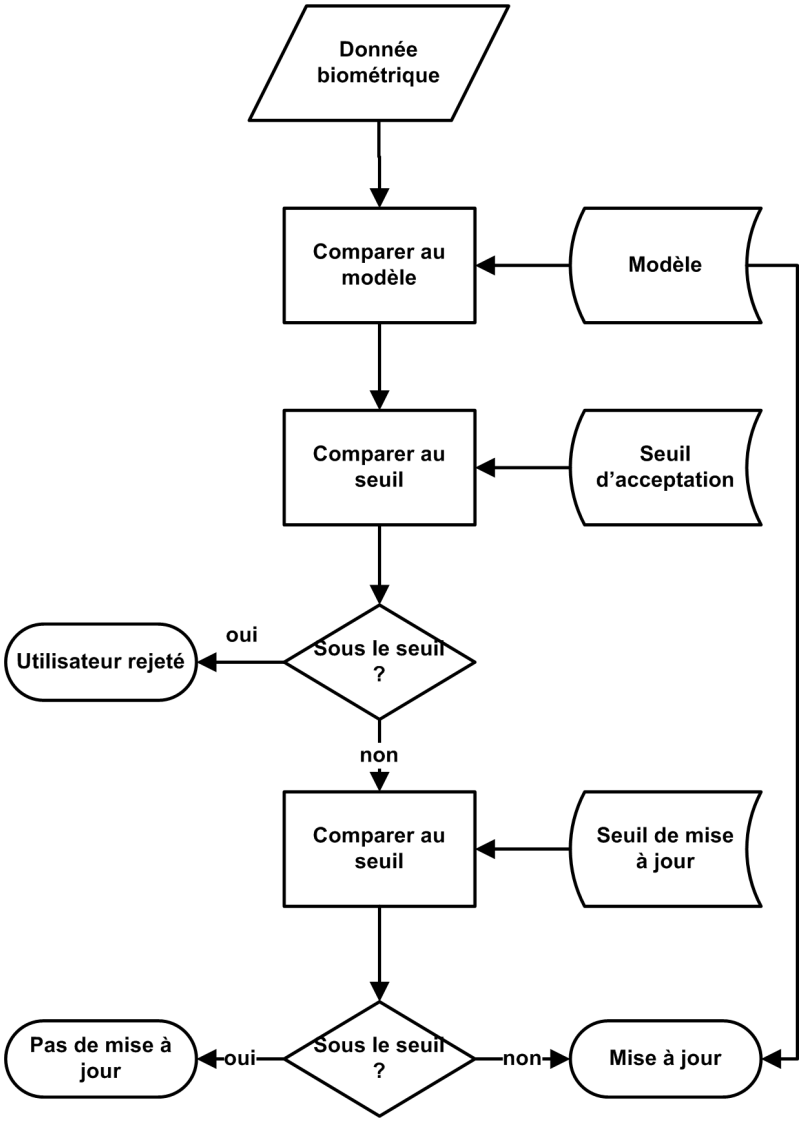


FIGURE 1.5 – Décision d'acceptation avec un double seuil

1.4.1.3 Indice de qualité

Une variante de la technique du double seuillage est d'utiliser un indice de qualité de la nouvelle capture. Si cette dernière est de meilleure qualité que celle du *modèle biométrique*, elle le remplace. Noval et López [45] présentent l'utilisation d'un tel indice de qualité : si la capture est suffisamment proche de celle de l'utilisateur, et, suffisamment éloignée de celles des imposteurs, alors elle est utilisée dans le nouveau *modèle biométrique*. L'intérêt d'un tel mécanisme est de pouvoir remplacer les captures de faible qualité obtenues durant l'enrôlement de l'utilisateur. Cependant, leur vérification n'est pas faite en ligne : les captures nouvellement acceptées sont stockées dans une seconde base. Ensuite, de manière périodique, les captures de meilleure qualité que celles du modèle y sont incluses. La FIGURE 1.6 présente un système équivalent en mode en ligne.

L'information qualité peut également être utilisée de façon totalement différente et être intégrée dans le mécanisme de reconnaissance. Poh *et al.* [48] utilisent la notion d'indice de qualité pour améliorer les performances du système. Dans ce cas, la qualité représente plus une qualité de conditions d'acquisition qui est due aux différentes modifications de l'environnement, qu'à une qualité de la donnée biométrique

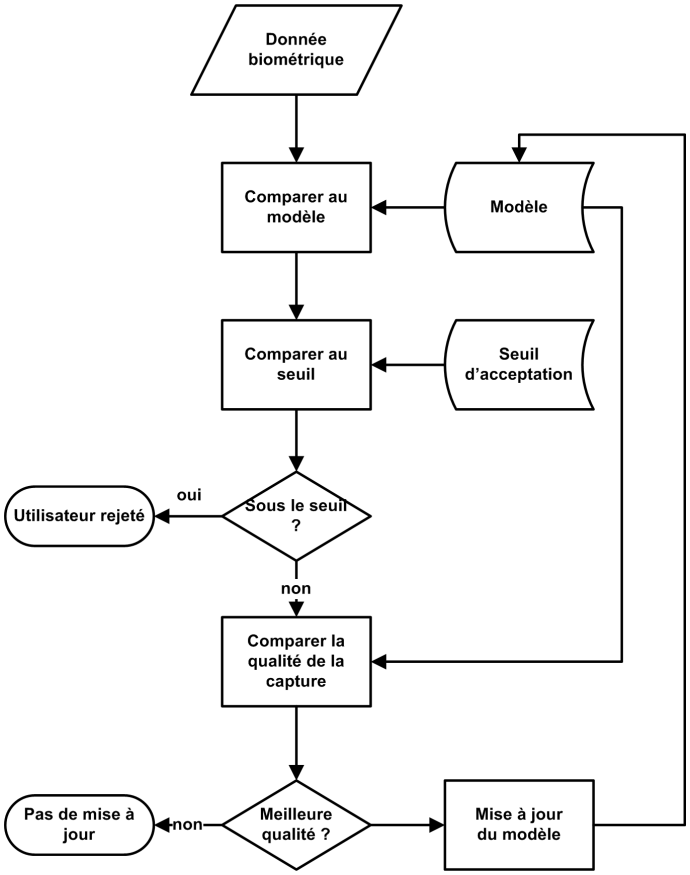


FIGURE 1.6 – Décision de mise à jour en utilisant une information de qualité

qui serait mauvaise car les conditions ne sont pas idéales. Dans leur papier, les auteurs ont trois conditions différentes d'acquisition, ce qui implique qu'ils ont trois types de qualité dans leur donnée. Cette mesure permet d'adapter les paramètres à l'environnement d'acquisition (et non pas à l'évolution de l'utilisateur comme dans toutes les autres études de *mise à jour du modèle biométrique*).

1.4.1.4 La prédiction

En analysant les scores (de comparaisons authentiques) produits au fil de l'utilisation du système biométrique, il est possible de détecter un vieillissement du modèle biométrique. Carls [10] présente un framework appelé CTARP qui analyse les scores de vérifications des données clientes et leur déviation au cours du temps (en fonction de la prédiction du score). À l'aide de ces informations (scores précédents, score prédit, et score mesuré), le système est capable de détecter le moment où il est utile de ré-enrôler la personne [11]. Un gain de performance de 20% a été observé sur des bases de reconnaissance faciale sur une longue période. Cependant, il semble que l'étude n'ait pris en compte que l'analyse de faux rejets, et pas de fausses acceptations. L'étude ne donne aucune indication sur l'évolution du taux de fausse acceptation en fonction de la mise à jour (qui fonctionne de manière supervisée), ni le comportement du framework dans un environnement non supervisé. Pour chaque utilisateur, une matrice nommée *Perfect Match Scores Matrix (PMSM)* est générée. Cette matrice est créée en comparant toutes les captures de la galerie entre elles. Ces captures sont triées par ordre chronologique, et, la diagonale vaut 1 (score de comparaison entre les deux mêmes images). Cette matrice peut ne pas être symétrique. On voir clairement une limite de cette méthode [5] : elle ne fonctionne qu'avec des systèmes où une seule instance est nécessaire dans le modèle. Une seconde matrice est générée à partir de la *PMSM* : la *Error Score Matrix*

5. comme une majorité des méthodes présentées dans ce fascicule

(ESM). Il s'agit tout simplement de $ESM = 1 - PMSM$. La FIGURE 1.7 présente un exemple de matrice partielle pour un utilisateur. Une partie de la matrice est utilisée pour l'apprentissage (les réseaux

Image	118	122	128	134	140	146	152
118	0	0	0.06	0.13	0.07	0	0.12
122	0	0	0	0	0	0.10	0.16
128	0.04	0	0	0	0	0.08	0.18
134	0.13	0	0	0	0	0.16	0.21
140	0.07	0	0	0	0	0.12	0.23
146	0.01	0.10	0.07	0.16	0.11	0	0.04
152	0.14	0.17	0.18	0.19	0.24	0.04	0

FIGURE 1.7 – Exemple d'une matrice de scores d'erreur, tiré de [10].

de neurones ont donné de meilleurs résultats dans [10]), tandis que l'autre est utilisée pour la prédiction. L'algorithme consiste à calculer la pente entre deux points : $m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} + y_2$, avec y_2 le nouveau score, x_2 la date de la nouvelle capture, y_1 le score précédent et x_1 la date de la capture précédente. De cette façon, il est possible de prédire la valeur du nouveau score de comparaison (de la future requête). L'erreur de détérioration (decay error (DE)) permet de savoir si la prédiction correspond au nouveau score. Son estimation est utilisée pour déterminer si la quantité d'erreurs entre la donnée d'enrôlement et la requête actuelle est suffisante pour appliquer la mise à jour du modèle. Elle est calculée de la façon suivante : $DE = \frac{\Delta_{score}}{\text{moyenne}\Delta_{scores}}$, avec Δ_{score} la pente entre l'erreur du dernier score et l'erreur actuelle, et moyenneΔ_{scores} la moyenne de toutes les pentes (incluant la pente actuelle). Si l'estimation du DE est supérieure à un seuil, alors le modèle devra être mis à jour lors de la prochaine itération (*i.e.*, la prochaine fois qu'une requête est acceptée) si le score actuel approxime le score prédit (*i.e.*, la requête précédente n'était pas un outlier).

1.4.1.5 L'appel à un oracle

La sélection peut également être faite manuellement par un opérateur. Les critères de sélection de l'opérateur peuvent être implicites ou explicites. Nous ne nous attarderons pas non plus sur ce point, car il ne correspond pas à nos attentes (automatisation totale du processus de *mise à jour du modèle biométrique*).

1.4.1.6 L'erreur de vérification

Vandana [76] propose un algorithme itératif pour mettre à jour l'espace propre de chaque utilisateur lorsque qu'une capture est rejetée de façon anormale. Cependant, contrairement aux autres papiers de la littérature, ce n'est pas l'administrateur qui est chargé d'initier cette mise à jour, mais les utilisateurs eux-mêmes en fonction de leur retour. Ce choix n'est pas justifié par l'auteur, mais on pourrait penser qu'il est parti du principe qu'une capture acceptée signifie qu'elle est suffisamment proche du modèle, et, que cela n'apporterait pas de variabilité supplémentaire de l'y intégrer. Tandis qu'une capture rejetée est forcément sujette à apporter plus de variabilité dans le modèle. Cependant, cette variabilité peut être la cause d'une erreur de capture, et son inclusion d'office dans le modèle peut être problématique.

1.4.1.7 Le système de mise à jour

Cette sélection de la donnée biométrique à intégrer dans le nouveau *modèle biométrique* peut être automatiquement faite par le système de *mise à jour du modèle biométrique*. Nous partons du principe que toutes les données capturées sont sujettes à être utilisées dans la mise à jour. Le filtrage est fait automatiquement par le système de *mise à jour du modèle biométrique*. Un tel exemple est présenté dans la section 1.4.3.3 page 23.

1.4.1.8 Le temps

Afin de ne pas capturer une variation *intra-classe* locale (d'un point de vue temporel, et donc qui disparaitra au bout d'un temps relativement court), il est possible de se baser sur l'utilisation d'un intervalle. Ainsi, la *mise à jour du modèle biométrique* peut être effectuée à intervalle régulier de façon automatique. La durée de l'intervalle est spécifique à la modalité, et, plus cette modalité est sujette à variation, plus cet intervalle doit être court. Kekre et Bharadi [34] indiquent qu'un intervalle de un mois pour un système de reconnaissance faciale semble être couramment adopté.

1.4.1.9 Un système hybride

On peut tout à fait imaginer que le critère de décision de mise à jour soit dépendant de plusieurs des critères présentés précédemment. Le système utilise une modalité *indépendante*, n'ayant pas de variation *intra-classe* au cours du temps, et, une modalité *indépendante* ayant une variation *intra-classe* non négligeable au cours du temps. La modalité indépendante peut être vue comme un oracle. La *mise à jour du modèle biométrique* n'est effectuée qui si l'utilisateur est rejeté par le système multimodal alors que l'oracle l'accepte et que:

- soit le nombre de rejets par la modalité dépendante (tandis que la modalité indépendante accepte l'utilisateur) atteint un seuil;
- soit la durée d'utilisation du système sans utiliser la mise à jour a atteint son délai.

Dans tous les cas, la *mise à jour du modèle biométrique* ne peut être appliquée que si la modalité dépendante a fait une erreur de reconnaissance.

1.4.1.10 Discussion

Les méthodes présentées précédemment ont toutes leurs avantages et inconvénients. Pour l'utilisation du double seuillage, seules les données biométriques à forte probabilité d'appartenance à l'utilisateur sont utilisées. Cela implique fortement que peu de variabilité soit capturée (étant donné que l'on ne sélectionne que des données que l'on sait reconnaitre). Nous évitons donc au maximum d'intégrer des données d'imposteurs, mais nous limitons également l'amélioration des performances en restant dans un minimum local. Utiliser un indice de qualité pour faire la *mise à jour du modèle biométrique* est intéressant lorsque l'on contrôle au maximum les conditions d'acquisition de la donnée biométrique. Cependant, dans un cas où les conditions d'acquisition sont toujours dégradées pour des raisons particulières (hygrométrie importante à cause de la localisation géographique du capteur) il est nécessaire que le modèle soit capable d'intégrer ces données de mauvaise qualité. Or, cette mise à jour sera rarement effectuée en raison du filtrage sur la qualité des données. L'utilisation d'un oracle implique un cout supplémentaire (faire appel à une personne) et n'est pas nécessairement automatisable. L'utilisation d'un mécanisme d'acceptation interne au processus de mise à jour du système semble être une façon intéressante, mais ne peut pas être utilisable dans tous les cas.

1.4.2 Périodicité

Nous pouvons distinguer deux façons majeures d'appliquer les systèmes de mise à jour dans la littérature : la mise à jour *en ligne*, et, la mise à jour *hors ligne*.

1.4.2.1 La mise à jour hors ligne.

Le mécanisme de *mise à jour du modèle biométrique* est lancé par lot (ou mode "batch") lorsque nous disposons de suffisamment de nouvelles données (*i.e.*, les données capturées lors de l'utilisation du système biométrique). L'ordre d'apparition des captures n'importe pas nécessairement, car, les données sont traitées dans leur ensemble sans nécessairement prendre en compte leur chronologie. Cette *mise à jour du modèle biométrique* peut être *semi-supervisée* (*i.e.*, utiliser les étiquettes calculées par le classifieur), ou, *supervisée* si les étiquettes des captures sont fournies par un oracle (l'administrateur du système, par

exemple). Un problème ouvert est la fréquence de la *mise à jour du modèle biométrique*. Quelle est la meilleure stratégie à adopter ? Attendre d'avoir collecté suffisamment de données ? Ou attendre l'expiration d'un certain délai ? Dans la plupart des études, l'appel n'est fait qu'une seule fois quand suffisamment de données sont collectées : les études partagent leur base de données en trois ensembles : un pour l'apprentissage initial, un pour l'application de la mise à jour, et un pour valider les nouvelles performances (après mise à jour). La FIGURE 1.8 présente le schéma d'un tel système.

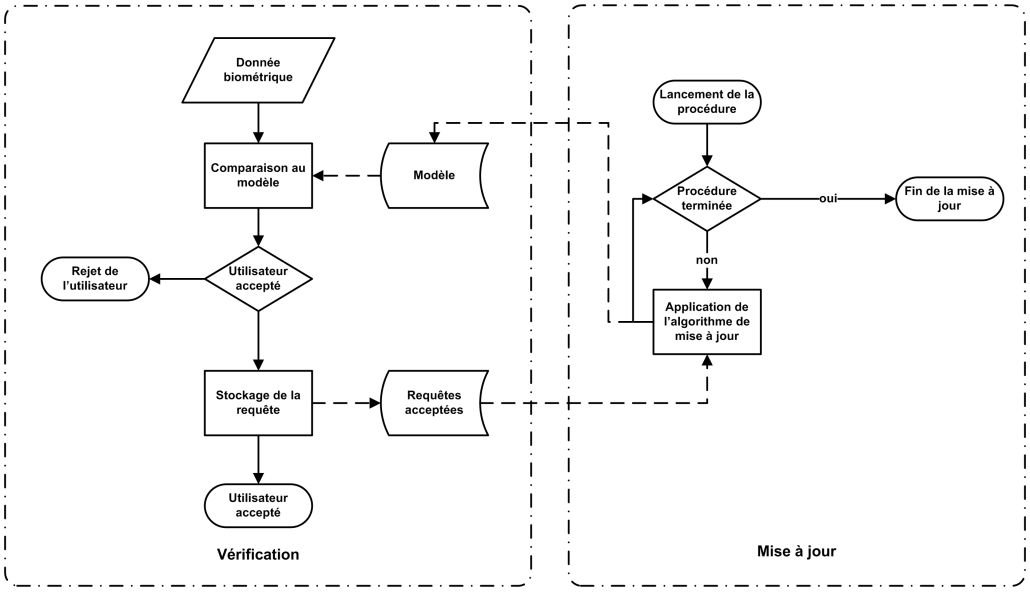


FIGURE 1.8 – Schéma de la mise à jour hors ligne du *modèle biométrique*.

1.4.2.2 La mise à jour en ligne

Le mécanisme de *mise à jour du modèle biométrique* est lancé dès la fin de la vérification, si la capture *requête* est considérée comme étant celle de l'utilisateur attendu et que le système désire la prendre en compte pour la mise à jour (voir section 1.4.1 page 13). Il s'agit donc d'une mise à jour en temps réel, qui se déroule façon itérative, *requête* par *requête*. Il est connu qu'avec ce type de mécanisme, l'ordre d'apparition des captures influe sur la qualité de la mise à jour du modèle [55]. Cette *mise à jour du modèle biométrique* est donc nécessairement *semi-supervisée* : le système se sert de l'étiquette (client ou imposteur) calculée par le système de vérification pour la capture donnée. Un tel mécanisme est particulièrement adapté aux environnement ayant de faibles capacités de stockage ou de calculs, tels les appareils mobiles [50]. La FIGURE 1.9 présente le schéma d'un tel fonctionnement.

1.4.2.3 Discussion

Les façons des gérer les mises à jour sont donc différentes en fonction de la périodicité choisie. La mise à jour en ligne ne nécessite pas de stocker, progressivement, l'ensemble des requêtes acceptées, étant donné qu'elle effectue la mise à jour dès son acceptation. Le cout d'utilisation mémoire est donc faible. En contrepartie, il est nécessaire d'effectuer le calcul dès l'acceptation de la requête, ce qui peut être relativement couteux en temps de calcul. La mise à jour hors ligne nécessite le stockage des requêtes acceptées au fil de l'utilisation du système. Elle est donc plus couteuse en consommation mémoire. Cependant, la vérification est moins couteuse en temps de calcul, car elle ne nécessite pas d'effectuer la mise à jour instantanément. Les papiers de Fabio Roli (en reconnaissance d'empreintes digitales ou reconnaissance de visage) insistent sur le fait que le mode en ligne est sensible à l'ordre de présentation des données. Pour cette raison, les calculs sont effectués plusieurs fois en utilisant un ordre de présentation aléatoire. Il est

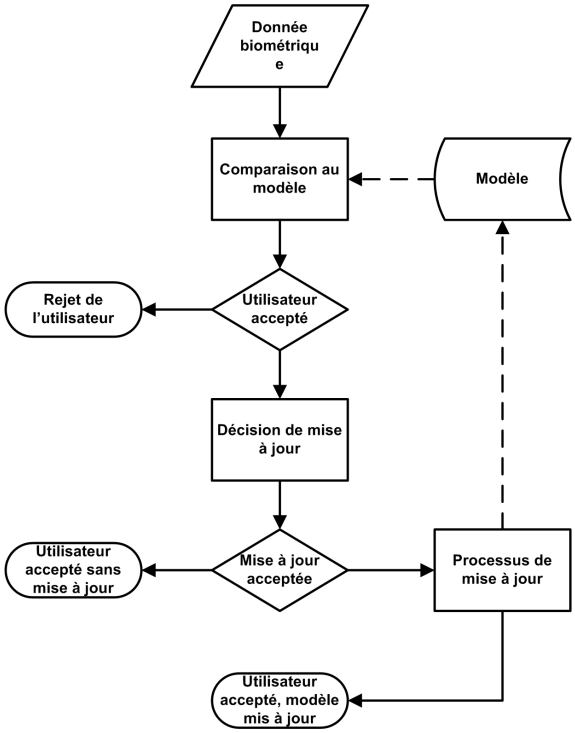


FIGURE 1.9 – Schéma de la mise à jour en ligne du *modèle biométrique*.

fort probable que cette assertion soit vraie pour de telles biométries (bien que les papiers de [19] semblent montrer le contraire), cependant, nous pensons que dans le cas des biométries comportementales (notamment la dynamique de frappe au clavier), cet ordre doit être impérativement conservé afin de prendre en compte l'aspect apprentissage cognitif de l'utilisateur. Il nous semble utile d'explorer ce point.

1.4.3 Modes de mise à jour

Comme nous l'avons vu précédemment, il existe deux familles principales de mécanisme de *mise à jour du modèle biométrique*: la *mise à jour supervisée* qui nécessite l'aide d'un opérateur humain, et la *mise à jour semi-supervisée* qui est totalement automatique, et, se sert des informations de ses classifieurs. Nous n'avons pas trouvé dans la littérature de méthode *non supervisée*, ou la mise à jour du modèle de l'utilisateur se ferait sans aucune indication sur l'étiquette des nouveaux exemples.

1.4.3.1 Mise à jour supervisée

Nous n'allons pas nous attarder sur la gestion *supervisée* de la *mise à jour du modèle biométrique*, car celle-ci a suffisamment été étudiée dans la littérature, et, ne présente pas particulièrement de défi technique. Comme cela a été dit précédemment, l'étiquette des captures est fournie par un *administrateur*. Celui-ci a pu se les procurer de deux façons distinctes :

- en gérant *plusieurs sessions d'enrôlement*. Cette étape peut être longue, fastidieuse et couteuse. Elle implique de demander aux différents utilisateurs de participer à plusieurs sessions d'enrôlement. L'administrateur supervise ces enrôlement pour empêcher les erreurs (*i.e.*, enrôlement d'un individu β au lieu d'un individu λ) ;
- en *étiquetant manuellement* les données ayant été capturées tout au long de l'utilisation de l'application. Dans ce cas, l'administrateur doit être un expert du domaine, et cette méthodologie n'est peut-être pas applicable pour toutes les biométries. Cette pratique est également couteuse en temps. Sukthankar et Stockton [72] proposent à un administrateur de modifier, au cours de l'utilisation

d'un système de portier électronique, manuellement les étiquettes des visages reconnus en cas d'erreur du système de reconnaissance faciale. Ces nouvelles données sont utilisées dans le système de reconnaissance faciale afin d'en augmenter les performances. La FIGURE 1.10 présente le fonctionnement d'un système de *mise à jour du modèle biométrique* supervisé.

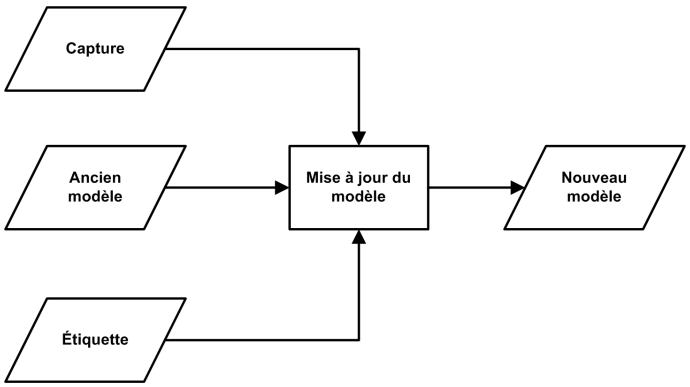


FIGURE 1.10 – Principe d'une mise à jour supervisée

1.4.3.2 Mise à jour semi-supervisée

Plusieurs techniques semi-supervisées sont référencées dans la littérature. Le principe de base est de comprendre que l'étiquette d'une capture est définie par l'algorithme de reconnaissance. Si l'étiquette correspond à celle de l'utilisateur, le système peut l'ajouter à son modèle. Il n'y a pas d'étiquetage manuel. Les méthodes utilisant l'apprentissage semi-supervisé vont utiliser ces données non étiquetées pour compléter l'apprentissage supervisé. La FIGURE 1.10 présente le fonctionnement d'un système de *mise à jour du modèle biométrique* semi-supervisé.

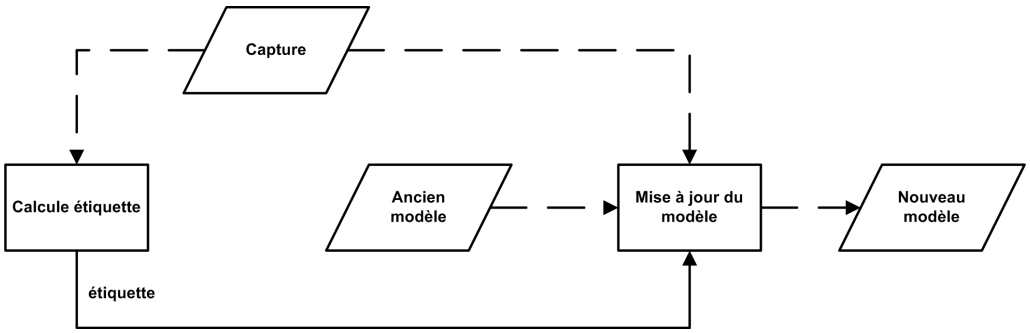


FIGURE 1.11 – Principe d'une mise à jour semi-supervisée

Plusieurs méthodes d'apprentissage semi-supervisé existent. Nous allons présenter l'auto apprentissage, le co apprentissage, et des méthodes basées sur la propagation d'étiquettes dans un graphe.

Auto apprentissage les techniques d'*auto apprentissage* (ou *self-training* dans la littérature anglophone sont des méthodes ayant été étudiées dans la littérature [43, 55, 62, 63]. Le terme *auto apprentissage* est utilisé car la méthode se met à jour incrémentalement elle-même en utilisant sa propre connaissance. Dans le cas de la *mise à jour du modèle biométrique*, le fonctionnement est le suivant:

1. Lors de l'enrôlement, le *modèle biométrique* est généré en entrainant le classifieur avec les données étiquetées (D_L) récupérées de façon supervisée.

2. Un ensemble de données non étiquetées (D_U) est collecté tout au long de l'utilisation du système biométrique. Lorsque suffisamment de données sont collectées, nous pouvons passer à l'étape suivante. La difficulté réside à savoir ce qu'est "suffisamment".
3. Le classifieur est utilisé pour étiqueter les données incomplètes (D_U).
4. Les données étiquetées avec un fort degré de confiance sont ajoutées aux données d'apprentissage (D_L). La notion "fort degré de confiance" est également subjective.
5. Le classifieur est ré entrainé sur l'ensemble des données étiquetées (D_L) et la procédure est répétée à l'étape 3 jusqu'à satisfaire un critère d'arrêt. Ce critère d'arrêt peut être un nombre de répétitions de l'algorithme.

L'auto apprentissage a tout de même quelques limites. Il est toujours possible d'ajouter des données d'imposteur si la comparaison de celles-ci au *modèle biométrique* fait penser qu'il s'agit d'une donnée cliente à fort degré de confiance. Augmenter ce degré réduit donc la probabilité d'inclure des données erronées. Cependant, le travers de l'utilisation de ce degré est que seules les données clientes fortement probables sont ajoutées au *modèle biométrique*. Le co-apprentissage (*cf.* section 1.4.3.2) est une solution à ce travers. Ce sont donc des données fortement ressemblantes à celles déjà présentes. Elles n'apportent pas forcément beaucoup d'information de variabilité supplémentaire. Ainsi, la configuration de ce seuil dépend grandement de la performance du système. Cette technique a pu être appliquée avec succès à des systèmes de reconnaissance faciale utilisant une Analyse en Composantes Principales (ACP) et de reconnaissance d'empreintes digitales utilisant la méthode de mesure *strings*. El Gayar *et al.* [20] proposent des évolutions de l'auto apprentissage adaptées à la multimodalité. Le premier d'entre eux est nommée *Single Classifier Self-training*.

Single Classifier Self-training Étant donné qu'il s'agit d'un système multimodal, nous disposons de K classifieurs (dans [20], il s'agit de trois algorithmes de reconnaissance faciale travaillant sur le même jeu de données). Le fonctionnement du système est le suivant :

1. Lors de l'enrôlement, le *modèle biométrique* est généré en entrainant chaque classifieur avec les données étiquetées ($D_{L_i} = D_L$, pour chaque classifieur CL_i avec $i \in [1; K]$) récupérées de façon supervisée.
2. Un ensemble de données non étiquetées (D_U) est collecté tout au long de l'utilisation du système biométrique. Lorsque suffisamment de données sont collectées, nous pouvons passer à l'étape suivante.
3. Une partition $D_U\prime$ est sélectionnée aléatoirement depuis D_U.
4. Pour chaque classifieur CL_i:
   (a) Le classifieur CL_i est utilisé, de façon indépendante aux autres classifieurs, pour étiqueter les données incomplètes. ($D_U\prime$).
   (b) Les n exemples classés avec le plus de confiance de chaque classe sont sélectionnés.
   (c) Ces exemples sont ajoutés à l'ensemble de données étiquetées du classifieur (D_{L_i}) qui est ré entrainé.
   (d) Sélectionner aléatoirement des données de D_U pour les intégrer dans D_{U_i} afin de remplacer les nouvelles données étiquetées.
5. Le processus est répété un certain nombre de fois depuis l'étape 4

De cette façon, les classifieurs sont mis à jour indépendamment les uns des autres (la connaissance d'un classifieur n'influe pas la mise à jour d'un autre classifieur).

Co-apprentissage Le *co-apprentisage* (ou *co-training* dans la littérature anglophone) est une version de l'autoapprentissage adaptée à l'utilisation de deux (ou éventuellement plusieurs, même si la littérature ne dispose pas de tel exemple) classifieurs qui vont s'entraider pour s'améliorer mutuellement. Nous avons précédemment présenté une limite de l'auto apprentissage : la possibilité d'inclure uniquement des motifs relativement ressemblants, et, donc, encodant une faible variabilité. Leur efficacité est donc potentiellement limitée. Le co-apprentissage permet de faire totalement abstraction de ce problème. L'utilisation

d'un second classifieur va justement permettre l'inclusion de données à forte variabilité qui n'auraient pas été retenues par le premier classifieur [17, 55, 57, 58, 61, 63] Plus les modalités utilisées sont indépendantes les unes des autres, plus les performances du co-apprentissage sont censées être meilleures. Lorsque l'environnement est fortement contrôlé, les performances sont semblables aux techniques d'autoapprentissage, mais les performances sont nettement meilleures lorsque l'environnement n'est pas contrôlé [58]. Les données sont donc des couples de données : la donnée de la modalité 1, ainsi que la donnée de la modalité 2 (naturellement, le système peut être utilisé avec plus de modalités). Dans le cas de la *mise à jour du modèle biométrique*, le fonctionnement est le suivant :

1. Lors de l'enrôlement, le *modèle biométrique* est généré pour les deux modalités à l'aide de l'ensemble D_L.
2. Un ensemble de données (sous forme de couple) non étiquetées (D_U) est collecté tout au long de l'utilisation du système biométrique. Lorsque suffisamment de données sont collectées, nous pouvons passer à l'étape suivante. La difficulté réside à savoir ce qu'est « suffisamment ».
3. Chacun des deux classifieurs est utilisé pour étiqueter les données incomplètes (D_U)
4. Les données étiquetées avec un fort degré de confiance (par un des classifieurs) sont ajoutées aux données d'apprentissage (D_L). La notion « fort degré de confiance » est également subjective ; en général, il s'agit d'un second seuil.
5. Les classifieurs sont ré entrainés sur l'ensemble des données étiquetées (D_L) et la procédure est répétée à l'étape 3 jusqu'à satisfaire un critère d'arrêt. Ce critère d'arrêt peut être un nombre de répétition de l'algorithme.

Rattani *et al.* [58] utilise une légère variante lors de l'ajout des nouvelles données dans le système (dans le cas d'une reconnaissance d'empreintes digitales et de visage). Le système de co apprentissage est exécuté deux fois, avec un classifieur primaire et un classifieur secondaire (les deux classifieurs intervertissent donc leur rôle au deuxième lancement). Si la requête est acceptée par les deux classifieurs, elle est considérée comme encodant peu de variabilité : elle est donc fusionnée avec le modèle le plus proche (*cf.* techniques de super modèle, section 1.4.4.1 page 28) du classifieur secondaire. Si la requête n'est acceptée que par le classifieur primaire, elle est considérée comme encodant une forte variabilité, et, est ajoutée à la galerie du classifieur secondaire. Comme le co apprentissage permet d'intégrer des données avec plus de variabilités, les nouveaux modèles encodent donc plus de variabilité *intra-classe*. Rattani *et al.* [57] montrent que les performances sont meilleures avec le co-apprentissage qu'avec l'auto apprentissage, notamment car plus de données non labellisées sont intégrées dans le *modèle biométrique*.

Ensemble Driven Training El Gayar *et al.* [20] proposent un autre mécanisme utilisant plusieurs classifieurs. Dans ce cas, l'auto apprentissage de chaque classifieur n'est plus indépendant. Les classifieurs sont mis à jour lorsqu'ils sont majoritairement d'accord sur l'étiquette à affecter aux exemples non étiquettés.

1. Lors de l'enrôlement, le *modèle biométrique* est généré en entrainant chaque classifieur avec les données étiquetées ($D_{L_i} = D_L$, pour chaque classifieur CL_i avec $i \in [1; K]$) récupérées de façon supervisée.
2. Un ensemble de données non étiquetées (D_U) est collecté tout au long de l'utilisation du système biométrique. Lorsque suffisamment de données sont collectées, nous pouvons passer à l'étape suivante.
3. Une partition $D_U\prime$ est sélectionné aléatoirement depuis D_U.
4. Chaque classifieur CL_i étiquette tous les exemples de $D_U\prime$
5. Les K classifieurs sont combinés sur chaque exemple de $D_U\prime$ en utilisant un vote majoritaire.
6. Les données de $D_U\prime$ pour lesquels les K classifieurs sont majoritairement d'accord sont intégrées à l'ensemble de données étiquetées (D_L).
7. Les classifieurs sont ré entrainés avec le nouvel ensemble étiqueté (D_L).
8. Sélectionner des données de D_U pour les intégrer dans $D_U\prime$ afin de remplacer les nouvelles données étiquetées.
9. Le processus est répété un certain nombre de fois depuis l'étape 4

Contrairement à la méthode *single classifier self-training*, cette approche permet aux classifieurs de se mettre à jour en utilisant la connaissance des autres classifieurs. Mais, contrairement à la méthode de co-apprentissage précédente, on peut s'attendre à capturer moins de variabilité en raison du consensus (le vote majoritaire) utilisé.

1.4.3.3 Les méthodes à base de graphes

Comme nous l'avons dit précédemment (*cf.* section 1.4.3.2 page 21), dans le cas d'un système mono modal, l'utilisation d'un système d'auto apprentissage peut amener à améliorer la fonction utilisateur (son modèle) à un minimal local (*i.e.*, seules les données fortement ressemblantes sont utilisées pour la mise à jour). Le co apprentissage est une solution dans le cas d'un système multimodal uniquement (*i.e.*, les connaissances d'un classifieur plus fort permettent d'améliorer un classifieur plus faible). L'utilisation de méthodes à base de graphes peut être une solution au manque de capture de variabilité dans le cas mono modal [54, 55]. Cette technique nécessite de disposer d'un certain nombre de données non étiquetées. La plupart des études travaillent hors ligne, on verra à la fin de cette section, que récemment une méthode en ligne d'un des algorithmes a été proposé. Contrairement aux méthodes semi supervisées classiques, les techniques à base de graphes sont capable de capturer énormément de variabilité *intra-classe*. L'utilisation de ces nouvelles données va donc permettre d'améliorer la mise à jour et les modèles, en encodant plus de variabilité. Le principe des modèles semi supervisés à base de graphes est de propager les étiquètes dans un graphe depuis les nœuds étiquetés vers les nœuds non étiquetés. Les différents mécanismes prennent en compte la structure de l'ensemble des données, dans le graphe généré à l'aide des données biométriques reliées en fonction de leur similarité. Chaque nœud représente donc une donnée biométrique, et, chaque arête relie deux nœuds ayant une certaine similarité entre eux. Celle-ci peut être pondérée en fonction de la valeur de la similarité.

La technique du flot max/coupe min Blum et Chawla [9] présentent un système de classification des données non étiquetées en utilisant l'algorithme *flot-max/coupe-min* (voir l'annexe A page 51). Le but est de partitionner le graphe en deux zones : une zone contenant les données authentiques, et, une zone contenant les données imposteur. La partition appartenant à la classe authentique est utilisée comme nouveaux exemples de l'utilisateur. Les *nœuds* du graphe sont constitués des données biométriques (qu'elles soient étiquetées ou non). Les *arcs* du graphe sont pondérés par le score obtenu en comparant deux nœuds (la méthode fonctionne donc avec les systèmes capables de comparer deux captures entre elles, ce qui n'est pas le cas de la dynamique de frappe au clavier). Le problème de *mise à jour du modèle biométrique* devient un problème d'optimisation de l'étiquetage. Utiliser l'algorithme de *flot-max/coupe-min* convient bien à la résolution de ce problème, car l'approche est [9] :

- basée sur un binarisation des données en deux classes : les données *authentiques*, et, les données *imposteurs* ;
- l'optimisation est faite en temps polynomial ;
- la méthode est revendiquée comme étant robuste au bruit, et, qu'elle donne de bonnes performances, même avec peu de données étiquetées (ce qui est réaliste dans le cas de la *mise à jour du modèle biométrique*).

La méthode est la suivante : soit L l'ensemble étiqueté de données biométriques (dans certains cas, nous disposons de données authentiques, mais pas de données imposteurs. Celles-ci sont représentées par les données authentiques ayant les scores de comparaison les plus faibles[6] avec les autres [55]), et d'un ensemble U de données non étiquetées (les données biométriques capturées tout au long de l'utilisation du système). Comme cela a été précisé précédemment, nous sommes dans un cas de classification binaire, nous notons L_+ l'ensemble de données positives (les données authentiques), et L_-, l'ensemble des données négatives (les données imposteurs).

1. Nous construisons un graphe pondéré $G = (V, E)$, avec l'ensemble des nœuds $V = L \cup U \cup \{v_+, v_-\}$, et l'ensemble des arêtes $E \subseteq V \times V$. Un poids $w(e)$ est associé à chaque arête $e \in E$. Les nœuds v_+ et v_- sont des *nœuds de classification*, tandis que tous les autres nœuds sont des *nœuds d'exemple* (étiquetés ou pas).

6. plus le score est faible, moins les données sont proches. Plus le score est élevé, plus les données sont proches.

2. Les nœuds de classification sont connectés par des arcs de poids infini aux exemples ayant la même étiquette que eux. En particulier :

$$\begin{cases} w(v, v_+) = \infty, & \text{pour tout } v \in L_+ \\ w(v, v_-) = \infty, & \text{pour tout } v \in L_- \end{cases} \tag{1.11}$$

3. Les arcs entre les nœuds d'exemple sont assignés d'un poids basé sur une relation de similarité entre eux. Cette notion de similarité dépend donc de la nature des données d'exemples (en gros du type de donnée biométrique utilisée). La fonction d'assignation des poids est désignée comme étant la *fonction de pondération des arcs* w.
4. Il faut déterminer une coupure minimum (v_+, v_-) pour le graphe. Cela signifie, trouver l'ensemble d'arêtes ayant le poids total minimum lorsque leur retrait déconnecte v_+ de v_-. Cette solution est obtenue en utilisant l'algorithme de flot max/coupe min pour lequel v_+ est la source, v_- est le puits, et les poids des arêtes sont considérés comme étant des capacités (*cf.* annexe A). Supprimer les arêtes au niveau de la coupe partitionne le graphe en deux ensembles de nœuds que nous appelons V_+ et V_-, avec $v_+ \in V_+$ et $v_- \in V_-$. Si il existe plusieurs coupures minimums, nous pouvons configurer l'algorithme pour qu'il choisisse le V_+ le plus petit.
5. Une étiquette positive est assignée à tous les exemples non étiquetés de V_+ (ce sont donc des données biométriques que nous considérons authentiques, et que nous pouvons utiliser dans le nouveau modèle). Un label négatif est assigné aux exemples de l'ensemble V_- (cependant, ces données ne nous sont d'aucun intérêt, si nous utilisons uniquement des algorithmes de détection d'anomalies).

L'algorithme part du principe que si les liens entre les nœuds similaires ont un poids important, alors, deux exemples similaires ont une forte probabilité d'être placés dans le même sous-ensemble de nœuds obtenu par la coupure minimum.

Blum et Chawla [9] ont prouvé qu'étiqueter le graphe avec cette approche a une erreur bornée qui serait l'erreur de validation croisée en utilisant un *leave one out* pour un algorithme des k plus proches voisins. Rattani [55] a testé ce mécanisme pour la mise à jour de modèles 2D faciaux. Différents nombres de nœuds à lier à chaque nœud ont été utilisés avec les k plus proches voisins avec $k \in \{3, 5, 10\}$. Le pourcentage de données d'imposteur intégré au modèle mis à jour est moins important qu'en utilisant une méthode d'auto apprentissage classique. La variabilité des données intégrée est également plus importante. Comme il s'agit d'un problème binaire, un graphe par utilisateur est créé. Comme la galerie initiale ne dispose pas de données d'imposteurs, les données ayant les plus petits scores sont considérées comme imposteurs. La FIGURE 1.12 illustre le fonctionnement d'un tel système. Dans le cas d'un mécanisme d'auto apprentissage classique, les données d'imposteur auraient été incluses dans le modèle. Ici, même si ils sont les plus proches voisins de données de l'individu, ils sont filtrés par la coupure (représentée par l'arc).

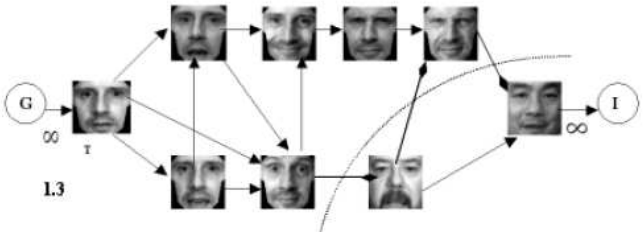


FIGURE 1.12 – Illustration du flot maximum/coupe minimum sur une mise à jour de modèle facial (source Rattani *et al.* [54])

Champs gaussien et fonction harmonique Dans [79, 80], Zhu montre qu'il est possible d'étiqueter les nœuds d'un graphe, de façon semi-automatique, dont seuls quelques nœuds sont étiquetés. Le problème est formulé sous la forme d'un champ aléatoire gaussien sur le graphe avec la moyenne du champ étant

caractérisée en terme de fonctions harmoniques qui sont obtenues à l'aide de méthodes matricielles ou de propagation de croyance. Soit l le nombre d'exemples étiquetés $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, u le nombre d'exemples non étiquetés $\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}$, et $n = l + u$ le nombre total de points ($l \ll u$). Les étiquettes sont binaires: $y \in \{0, 1\}$. Considérons un graphe connecté $G = (V, E)$ avec les nœuds V correspondant aux n exemples, avec les nœuds $L = \{1, .., l\}$ correspondant aux exemples étiquetés avec les étiquettes $y_1, \ldots, y_l$, et les nœuds $U = \{l + 1, \ldots, l + u\}$ correspondant aux exemples non étiquetés. Le but est donc d'assigner les étiquettes aux nœuds U. Le graphe est représenté par une matrice symétrique $n \times n$ d'adjacence pondérée $\mathbf{W}$. Le poids des arcs peut être calculé de la façon suivante, avec $\mathbf{x} \in \mathbb{R}$:

$$w_{ij} = exp\left(-\sum_{d=1}^{m} \frac{(x_{id} - x_{jd})^2}{\sigma_d^2}\right) \tag{1.12}$$

avec x_{id} la d^{e} composante de $\mathbf{x}_i$ représentée par un vecteur $\mathbf{x}_i \in \mathbb{R}^m$, et $\sigma_1, \ldots, \sigma_m$ des hyper-paramètres de normalisation. Dans le cas de la biométrie, le poids peut tout simplement être le score de similarité entre deux exemples. De cette façon, les points les plus proches ont un poids important. Il est maintenant nécessaire de calculer une fonction réelle $f : V \to \mathbb{R}$ sur G possédant certaines propriétés particulières, et, d'assigner des étiquettes en fonction de f. f doit donc être contraint à prendre les valeurs $f(i) = f_l(i) \equiv y_i$ sur les données étiquetées $i = 1, \ldots, l$. De façon intuitive, il est nécessaire que les points non étiquetés qui sont proches dans le graphe doivent avoir une étiquette similaire. C'est ce qui motive l'utilisation d'une fonction d'énergie quadratique :

$$E(f) = \frac{1}{2} \sum_{i,j} w_{ij}(f(i) - f(i))^2 \tag{1.13}$$

Afin d'assigner une distribution de probabilité sur les fonctions f, il faut créer le champ Gaussien $p_\beta(f) = \frac{e^{-\beta E(f)}}{Z_\beta}$, où β est un paramètre de « température inverse » et Z_β est la fonction de partition $Z_\beta = \int_{f|L=f_l} exp(-\beta E(f))df$, qui normalise sur toutes les fonctions contraintes aux données étiquetées. La fonction d'énergie minimum $f = \arg\min_{f|L=f_l} E(f)$ est *harmonique* ; cela signifie qu'elle satisfait $\mathbf{\Delta} f = 0$ sur les points non étiquetés U, et qu'elle vaut f_l sur les points étiquetés. Ici, $\mathbf{\Delta}$ est le *Laplacien combinatoire* que nous définissons de la façon suivante :

$$\mathbf{\Delta} = \mathbf{D} - \mathbf{W} \tag{1.14}$$

$\mathbf{D}$ est la matrice diagonale de degrés, telle que :

$$d_{i,j} = \begin{cases} \deg(v_i) & \text{si } i = j \\ 0 & \text{autrement} \end{cases} \tag{1.15}$$

et $\mathbf{W} = [w_{ij}]$ est la matrice d'adjacence. La propriété harmonique de f signifie que chacune de ses valeurs aux exemples non étiquetés est la moyenne de f sur les exemples du voisinage :

$$f(j) = \frac{1}{d_j} \sum_{i \sim j} w_{ij} f(i) \quad \text{, pour} \quad j = l + 1, \ldots, l + u \tag{1.16}$$

On peut exprimer légèrement différemment avec $f = \mathbf{P} f$, où $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$. Pour calculer la fonction harmonique en terme d'opérations matricielles, il faut séparer la matrice d'adjacence $\mathbf{W}$ (ainsi que $\mathbf{D}$ et $\mathbf{P}$) en quatre matrices après la l^{e} ligne et colonne.

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{ll} & \mathbf{W}_{lu} \\ \mathbf{W}_{ul} & \mathbf{W}_{uu} \end{bmatrix} \tag{1.17}$$

En posant $\mathbf{f} = \begin{bmatrix} \mathbf{f}_l \\ \mathbf{f}_u \end{bmatrix}$ avec $\mathbf{f}_u$ les valeurs sur les points non étiquetés, la solution harmonique $\mathbf{\Delta f} = 0$ avec $f|_L = f_l$ est donnée par :

$$\mathbf{f}_u = (\mathbf{D}_{uu} - \mathbf{W}_{uu})^{-1}\mathbf{W}_{ul}\mathbf{f}_l = (\mathbf{I} - \mathbf{P}_{uu})^{-1}\mathbf{P}_{ul}\mathbf{f}_l \tag{1.18}$$

Lorsque $f(i) > \frac{1}{2}$, on assigne l'étiquette 1, autrement, on assigne l'étique 0. Les performances peuvent être améliorées en incorporant de la connaissance sur la proportion des classes (grâce aux exemples étiquetés). La proportion q_c de données avec l'étiquette c peut être estimée depuis l'ensemble d'apprentissage. L'heuristique de *class mass normalization (CMN)* échelonne les probabilités afin de correspondre aux proportions. Il faut donc trouver un ensemble de coefficients $\mathbf{a} = a_1, \ldots, a_C$ tels que :

$$a_1 \sum_{i \in U} Y_u(i,1) : \cdots : a_C \sum_{i \in U} (i,C) = q1 : \cdots : qc \tag{1.19}$$

La classification d'un exemple i non étiqueté est faite en trouvant $argmax_c a_c Y_u(i,c)$. L'implémentation originale est disponible ici : `http://pages.cs.wisc.edu/~jerryzhu/pub/harmonic_function.m`. Le principe a été validé dans [7], sur un système d'identification dans un contexte de vidéo surveillance (seules les premières données sont étiquetées, les images sont de faibles qualités et rares). Les données sont constituées de plusieurs flux vidéos (0,5 images par seconde) de 10 personnes sur 3 mois. Seules les images avec une seule personne présente dans la scène sont conservées. Pour chaque image, le fond est soustrait avec un algorithme relativement simple, afin de récupérer la zone de l'image contenant l'individu (après avoir au préalable appliqué différents opérateurs morphologiques). Ainsi, chaque image est caractérisée par trois descripteurs différents :

- le moment d'acquisition (date et heure) ;
- un histogramme du premier plan ;
- le visage (face ou profile) extrait dans le premier plan (35% des images n'ont pas détecté de visage).

À l'aide de ces informations, il est possible de générer trois différents types d'arc dans le graphe :

le temps : comme il y a une seule personne par image, deux images espacées de quelques secondes correspondent probablement à la même personne. Les images ayant une différence de temps inférieure à un seuil t_1 (de quelques secondes) sont reliées.

la couleur : l'histogramme de couleur est fortement dépendant des habits de l'utilisateur. Les gens changent d'habits régulièrement, donc cette propriété est vraie sur un temps court. Pour chaque image i, les images ayant une différence de temps entre t_1 et t_2 (un autre seuil d'une demi-journée) sont sélectionnées. L'image i est ensuite reliée à ses k_c ($k_c = 3$) plus proches voisins (en terme de dissimilarité d'histogramme, avec la distance cosinus).

le visage : le visage des individus ne varie pas énormément au cours du temps. Pour chaque image i (pour laquelle un visage a été détecté), on recherche les images (avec un visage) dans un delta de temps supérieur à t_2. Les k_f ($k_f = 1$) plus proches visages (en terme de distance euclidienne) sont reliés entre eux.

Le graphe final est donc constitué de l'ensemble des images (les nœuds), avec les trois types d'arcs (qui ne sont pas différenciés, ni pondérés contrairement à l'algorithme précédent, section 1.4.3.3 page 23). Les expériences ont montré que la méthode semi-supervisée donne de meilleurs résultats que l'algorithme de base : un classifieur SVM linéaire (le noyau est une combinaison linéaire de noyaux linéaires sur les 3 types de descripteurs – les meilleurs poids de combinaisons sont sélectionnés avec une validation croisée). Utiliser CMC donne également de meilleures performances que la fonction harmonique seule. Le système est donc fait pour reconnaitre les personnes après avoir collecté un ensemble conséquent de données, il ne fonctionne donc pas en temps réel. Bien que le nombre d'images utilisé pour l'expérience est relativement grand ($>$ 5000), le nombre d'utilisateurs concerné est relativement faible (10). Il est probable que la technique ne soit pas autant performante avec plus de données.

Kveton *et al.* [37] proposent une amélioration de l'equation (1.18) page précédente en contrôlant la quantité d'extrapolation pour les donnés non étiquetées. Ils régularisent le Laplacien $\mathbf{\Delta}$ avec $\mathbf{\Delta} + \gamma_g \mathbf{I}$ avec γ_g un entier positif et $\mathbf{I}$ la matrice identité. Ainsi, les étiquettes sont obtenues de la façon suivante :

$$\mathbf{Y}_l = (\mathbf{\Delta}_{uu} + \gamma_g \mathbf{I})^{-1} \mathbf{W}_{ul} \mathbf{Y}_l \tag{1.20}$$

γ_g permet de contrôler de combien la confiance apportée aux exemples étiquetés décroît avec le nombre de sauts pour atteindre un exemple étiqueté. En voit clairement que l'equation (1.18) et l'equation (1.20) fonctionnent en mode hors ligne. Kveton *et al.* [37] proposent une évolution pour l'utiliser en ligne. Une façon simple de résoudre le problème est de maintenir le graphe d'adjacence complet à chaque étape

y et de l'utiliser pour déduire l'étiquette du nouvel exemple $\mathbf{x}_t$. Cette solution n'est pas pratique car sa complexité croit avec le temps t et est $O(t^3)$. Pour contourner ce problème, les auteurs proposent de quantifier les données afin de maintenir une version compacte de la matrice d'adjacence. Étant donné que la quantité de données non étiquetées est relativement plus importante que la quantité de données étiquetées, ce sont essentiellement les données non étiquetées qui sont quantifiées. Ainsi, la matrice d'adjacence utilisée devient :

$$\hat{\mathbf{W}} = \mathbf{VWV} \tag{1.21}$$

$\mathbf{W}$ est la matrice d'adjacence du graphe dérivé du graphe $\tilde{\mathbf{W}}$ en supprimant toutes les instances, sauf une, des nœuds identiques. La méthode *doubling algorithm* [13] est utilisée pour mettre à jour, de façon incrémentale, le graphe $\tilde{\mathbf{W}}$. La solution devient donc ($\hat{\mathbf{\Delta}}$ est le Laplacien de $\hat{\mathbf{W}}$):

$$\hat{\mathbf{Y}}_u = (\hat{\mathbf{\Delta}}_{uu} + \gamma_g \mathbf{V})^{-1} \hat{\mathbf{W}}_{ul} \mathbf{Y}_l \tag{1.22}$$

Kveton *et al.* [37] ont également validé leur algorithme dans un système de reconnaissance faciale.

1.4.3.4 Discussion

Les techniques basées sur l'*auto-apprentissage* semblent avoir été les plus couramment utilisées dans la littérature, même si le terme n'a pas été systématiquement employé. Nous parlons de méthodes *semi-supervisées* plutôt que de méthodes *non-supervisées* car nous nous servons de la pseudo étiquette calculée par la méthode de vérification biométrique. Il est souvent reporté dans la littérature que les méthodes semi-supervisées monomodales sont sujettes à capturer une faible variabilité, et, par conséquent, oublier d'intégrer une quantité non négligeable de données. Les méthodes à base de graphes pourraient être une solution partielle à ce problème. On peut également noter dans [74] l'utilisation d'un mécanisme d'autoapprentissage associé à un mécanisme d'*apprentissage actif* [16]. Il s'agit donc d'une utilisation partiellement supervisée de la mise à jour du modèle : les données relativement proches du modèle initial sont intégrées au modèle à l'aide du mécanisme d'auto apprentissage, et, les données difficiles à classifier sont étiquetées par un opérateur. Le cout est donc plus faible que d'utiliser un système entièrement supervisé, et plus de variabilité est capturée comparée à l'utilisation d'un système uniquement semi-supervisé. Le papier traite d'un problème de compréhension de phrase, et pas de biométrie, mais le concept peut facilement être adapté à un système d'identification faciale. Les méthodes à base de graphes ont été utilisées (du moins dans le cas des applications biométriques) pour mettre à jour la galerie des données biométrique. Même si cette piste ne semble pas avoir été explorée à notre connaissance, nous pensons que cette technique pourrait également être utilisée pour filtrer les données et extraire les outliers.

1.4.4 Stratégies de gestion des modèles

Nous avons vu dans la section 1.2 page 4 qu'il existe plusieurs façons de représenter un *modèle biométrique*. La gestion de la mise à jour de ces modèles est donc également spécifique. Nous allons donc présenter les différentes stratégies de gestion de *mise à jour du modèle biométrique* en fonction de leur représentation.

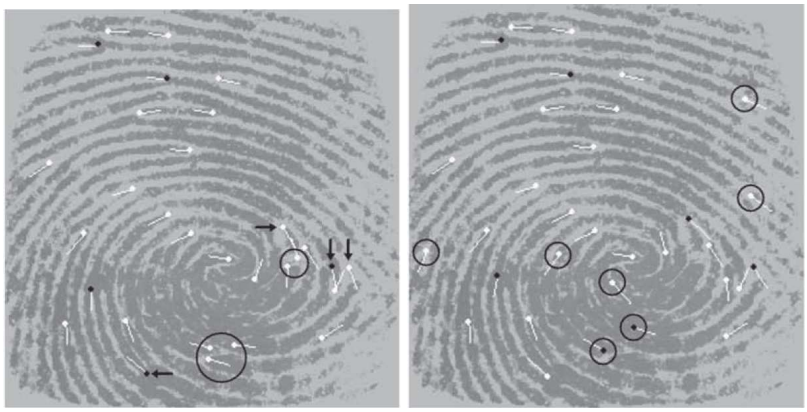
1.4.4.1 Référence

Lorsque le *modèle biométrique* est représenté par une unique référence, trois méthodes principales peuvent être utilisées : le remplacement de cette référence, l'ajout de la nouvelle référence ou la fusion de la nouvelle référence avec l'ancienne.

Le remplacement de l'ancienne référence par la nouvelle Dans ce cas, nous partons du principe que l'ancienne référence est obsolète, et, le modèle sera plus juste en utilisant uniquement la nouvelle. Cette méthode à uniquement un intérêt lorsque la place disponible pour stocker le *modèle biométrique* est relativement limitée (et que l'on ne peut stocker qu'une seule référence), et qu'il n'est pas possible d'appliquer un traitement plus sophistiqué et couteux en temps. Cette méthode ne semble pas avoir été expérimentée dans la littérature.

L'ajout de la nouvelle capture Dans ce cas, nous nous retrouvons dans le cas de la gestion des modèles par galerie. Une fois la nouvelle capture ajoutée à l'ensemble (qui était donc initialement constitué d'un seul et unique élément), les méthodes spécifiques aux galeries sont employées. Cette méthode est régulièrement employée dans la littérature.

La génération d'un super modèle Dans ce cas, il y a toujours une seule référence dans le *modèle biométrique*. La technique est nommée *super template generation* dans la littérature anglophone. Les super modèles sont des modèles générés à partir de plusieurs captures. Ces modèles générés peuvent être considérés comme une nouvelle capture. De tels systèmes peuvent être utilisés pendant l'enrôlement où plusieurs captures sont effectuées : un modèle est généré pour chacune de ces captures (*i.e.*, la liste des minuties de la capture, pour un système d'empreintes digitales), puis un super modèle est généré à partir des autres modèles (*i.e.*, la liste des minuties jugées pertinentes pour un utilisateur, pour un système d'empreintes digitales. Ce mécanisme peut aussi être employé dans le cas d'un système adaptatif, afin de modifier le super modèle précédent en fonction des variations nouvellement capturées. Une requête acceptée correspond donc à une capture de l'utilisateur. Cette capture peut encoder des variations par rapport au *modèle biométrique* actuel (variabilité naturelle de la donnée biométrique, meilleures conditions d'acquisitions ou erreurs d'acquisitions). Il est intéressant d'intégrer ces variations, qui peuvent encoder une variabilité réelle qui n'a pas pu être capturée précédemment. Cette méthode a été essentiellement utilisée dans le cas de l'utilisation de systèmes d'empreintes digitales [31, 65, 67] : de nouvelles minuties (qui n'ont pas été détectées lors de l'enrôlement) sont ajoutées à la référence, tandis que d'autres (considérées comme étant des erreurs) sont supprimées de la référence. Différentes familles de méthodes existent : fusion des images, fusion des minuties ou fusion des deux. Certaines méthodes utilisent un historique des captures, tandis que d'autres n'en utilisent que deux : le modèle et la requête. Les super modèles peuvent être utilisés uniquement à l'enrôlement, sans faire de mise à jour au cours de l'utilisation Ryu *et al.* [65]. Dans ce papier, le super modèle est généré à partir de plusieurs empreintes d'enrôlement. Les super mo-



(a) modèle initial (b) super modèle

FIGURE 1.13 – Illustration de la génération d'un super modèle. Les minuties encerclée sont celles qui disparaissent ou apparaissent. Les flèches indiquent un changement de type de minuties. (source : [31])

dèles ont également été étudiés dans le cas de la reconnaissance faciale. Rattani *et al.* [56] présentent une méthode pour générer un super modèle, pour la reconnaissance faciale, en utilisant une image frontale, une image de profil droit, et une image de profile gauche. De cette, façon, il n'est pas nécessaire d'avoir un modèle pour les trois vues du visage. Cette technique a été utilisée pour générer le super modèle depuis les données d'enrôlement, mais, on peut facilement imaginer son utilisation dans un système adaptatif. Des descripteurs SIFT [41] sont utilisés pour faire correspondre leur localisation dans les trois images. Les points d'intérêt correspondant à plusieurs images sont moyennés (à la fois pour leur localisation, description, et la valeur du descripteur). Le super modèle consiste en l'ensemble des points d'intérêts satisfaisant

la relation de l'equation (1.23) :

$$concat = \mathbf{F}_s \cup \mathbf{L}_s \cup \mathbf{R}_s - \sum(\mathbf{F}_s \cap \mathbf{L}_s \cap \mathbf{R}_s) \tag{1.23}$$

avec $\mathbf{F}_s$, $\mathbf{L}_s$, $\mathbf{R}_s$ respectivement les points d'intérêt de l'image de face, le profil gauche et le profil droit. Rattani *et al.* [56] montrent que les performances sont meilleures en utilisant ce super modèle plutôt qu'une unique image frontale.

1.4.4.2 Galeries

Plusieurs approches différentes existent pour la *mise à jour du modèle biométrique* basée sur l'utilisation de galerie. Comme les problèmes de sélection de *modèle biométrique* sont intimement liés au problème de *mise à jour du modèle biométrique*, nous présentons les deux types de méthodes. Le premier cas consiste à ne sélectionner que les exemples représentatifs (selon des critères spécifiques à ces méthodes) d'une galerie, tandis que le second consiste à mettre à jour le contenu de la galerie. Cette mise à jour de la galerie est une nécessité, car les individus collectés lors de l'enrôlement ont tendance à devenir de moins en moins représentatifs avec le temps.

Les méthodes de partitionnement ou de sélection Les méthodes de partitionnement ont initialement été développées pour permettre d'avoir plusieurs modèles par utilisateurs dans les méthodes utilisant une seule référence comme modèle [75]. Bien qu'il ne s'agisse pas à proprement parler de stratégies de mise à jour, elles permettent de disposer de plusieurs modèles pour un même utilisateur. Il s'agit donc d'une technique permettant d'encoder une forte variabilité intra-classe. La technique devient encore plus intéressante lorsqu'on lui rajoute un système de *mise à jour du modèle biométrique*. Cette méthode n'a été utilisée que dans le cas de la reconnaissance d'empreinte digitale, où une unique capture est présente dans le *modèle biométrique* (la vérification se fait en comparant la capture requête à la capture modèle). L'intérêt de partitionner est de pouvoir récupérer K captures parmi N (avec $K < N$) de telle façon à ce que ces K captures encodent un maximum de variabilité. Les auteurs présentent deux façons majeures de partitionner les données de la galerie :

- *DEND* [75]. Les captures sont agglomérées entre elles en utilisant un classifieur hiérarchique à l'aide d'une mesure de similarité. Cette agglomération est faite à l'aide d'un dendogramme. Celui-ci est sectionné de telle façon à obtenir K partions. La FIGURE 1.14 présente un tel dendograme dans le cas d'un système de reconnaissance d'empreintes digitales. Un représentant de chaque partition est utilisé comme prototype (c'est lui qui constituera le modèle de la partition). Le prototype sélectionné est la capture ayant la distance moyenne minimale avec les autres membres de la partition. Dans le cas où une partition n'a que deux éléments, le prototype est choisi de façon aléatoire. Étant donné que cette méthode cherche à capturer le plus de variabilité possible, elle est également sujette à prendre facilement des intrus.
- *MDIST* [75]. Cette version trie les captures en fonction de leur distance moyenne aux autres. Les K captures donnant les K plus petites valeurs sont sélectionnées comme prototype. Les prototypes choisis sont donc ceux ayant la plus forte similarité avec l'ensemble des données. Pour chaque utilisateur, la procédure à suivre est :
  1. Calculer les distances entre les N exemples.
  2. Pour le j^e exemple, calculer sa distance moyenne, d_j, en le comparant aux autres $(N-1)$ exemples.
  3. Choisir les K exemples ayant la plus petite distance moyenne. Ces exemples constituent la galerie T.

Naturellement, le choix de K dépend donc de l'application et reste un problème ouvert. Les deux méthodes ont été validées en générant des partitions aléatoires et en comparant les performances avec ces deux types de partitionnement. Les résultats ont montré qu'un choix aléatoire donne de moins bons résultats. Ces deux techniques de partitionnement du *modèle biométrique* ont été utilisées avec des méthodes de *mise à jour du modèle biométrique*, en les exécutant plusieurs fois au cours de l'utilisation du

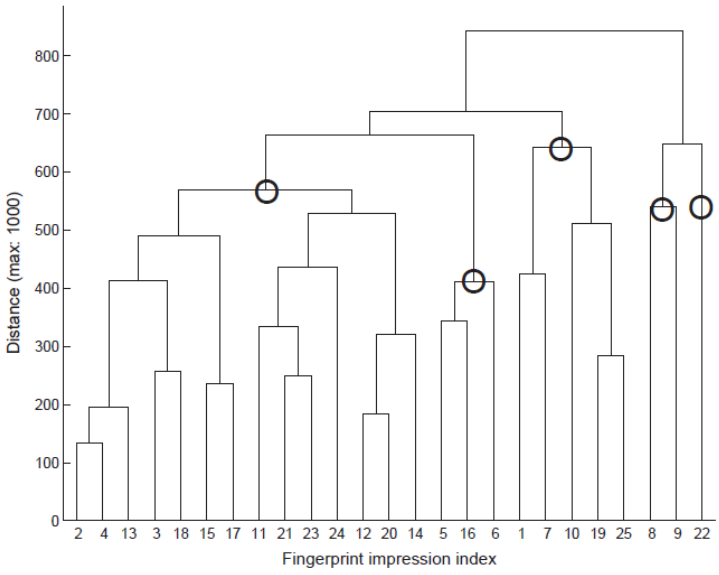


FIGURE 1.14 – Illustration du dendograme calculé pour un utilisateur avec 25 empreintes digitales dans sa galerie. Les cercles indiquent la coupure pour obtenir 5 partitions. (source : Uludag *et al.* [75])

système en intégrant de nouvelles captures. Lors d'une vérification, la requête est comparée aux K prototypes. Le score final est le score moyen des K scores. Ces méthodes ont été utilisées dans le cas supervisé. Li *et al.* [39] proposent également une méthode de sélection de K prototypes parmi N. La méthode a également été validée dans un contexte de reconnaissance d'empreintes digitales dans le cas supervisé. Une matrice de confusion $S_{N\times N}$ est créée en comparant tous les exemples entre eux. Elle est constituée d'un ensemble de scores (entre 0 et 1) $s_{m,n}$ qui représentent le score de comparaison entre le prototype t_m et la requête i_n. La distance entre l'ensemble A des données d'apprentissage et l'ensemble B des données de test est définie de la façon suivante (avec $N(A)$, $N(B)$ et $s(A,B)$ respectivement le nombre d'éléments de A, de B et le score de comparaison entre A et B) :

$$d(A,B) = N(A) * N(B) - s(A,B) \tag{1.24}$$

Ainsi, la distance des K prototypes sélectionnés aux exemples restants est représentée par :

$$d(T_K, I_{N-K}) = K * (N-K) - s(T_K, I_{N-K}) \tag{1.25}$$

$K * (N-K)$ étant une constante, trouver les K prototypes parmi N revient à trouver la partition qui maximise $s(T_K, I_{N-K})$. Li *et al.* [39] ont proposé deux méthodes pour obtenir la partition idéale :

- *Maximum Match Scores (MMS).* Le but de cette méthode est de tester tous les cas possibles, et, de ne sélectionner que le meilleur d'entre eux. La méthode est donc calculatoirement complexe et exponentielle en fonction de N.

Initialiser N, K, $S_{N\times N}$, $selection[K]$;
Lister tous les T_K^i et T_{N-K}^i $(1 \leq i \leq C_N^K)$;
pour i *de* 1 *à* C_N^K **faire**
  Calculer $s_i \leftarrow s(T_K^i, T_{N-K}^1)$ avec $S_{N\times N}$
Trouver i^* ($i^* \in \{1, ..., C_N^K\}$) tel que $s_{i^*} \geq s_i$ ($i \in \{1, ..., C_N^K\}$) $selection[K] \leftarrow T_K^{i^*}$

- *Greedy Maximum Match Scores (GMMS).* Étant donné que l'algorithme précédent est trop gourmand en temps de calcul, une version plus légère a été développée.

Initialiser N, K, $S_{N\times N}$, $selection[K]$;
Lister tous les T_K^i et T_{N-K}^i $(1 \leq i \leq C_N^K)$;
pour *i de 1 à K* **faire**
  Calculer $sum(j) = \sum_{m=1,m\neq j}^{N} s_{j,m} \mid j = 1, \ldots N$
Trouver i^* ($i^* \in \{1, ..., C_N^K\}$) tel que $s_{i^*} \geq s_i$ ($i \in \{1, ..., C_N^K\}$) $selection[K] \leftarrow T_K^{i^*}$

Les méthodes d'édition Le but des méthodes d'édition, comme les méthodes de partitionnement, est de réduire le nombre d'individus représentatifs de la galerie. C'est donc une technique qui n'est pas nécessairement liée à l'utilisation de mécanisme de *mise à jour du modèle biométrique*. Freni [22] a présenté dans sa thèse de doctorat, de nombreuses techniques d'édition dans le cas de systèmes de reconnaissance faciale et d'empreinte digitale. Il s'agit, une fois de plus, d'une technique intéressante pour les systèmes utilisant des classifieurs de type plus proches voisins. Le principe est : pour un ensemble d'apprentissage T, il faut trouver un sous-ensemble E qui permet d'avoir le même taux de classification que T sur lui-même. L'intérêt de la technique est d'avoir une taille de galerie dépendante de la difficulté de reconnaissance du client (et donc de réduire au maximum sa taille). Les méthodes d'édition permettent de ne conserver que les instantes représentantes d'un jeu de données lors de l'utilisation des k plus proches voisins (et donc de diminuer la combinatoire lors de la vérification, en diminuant le nombre de tests à effectuer). Deux familles de méthodes d'édition existent :

- Les méthodes *incrémentales* qui partent d'un ensemble E vide, et, le peuplent progressivement jusqu'à atteindre un critère de satisfaction.
- Les méthodes *décrémentales* qui partent d'une ensemble E complet (tous les éléments de T), et, suppriment les instances progressivement jusqu'à satisfaire un critère de décision.

Il semble que ces techniques d'édition aient été appelées sur l'ensemble des galeries en une fois (*i.e.*, le processus est effectué de manière globale, et non pas individuelle pour chaque utilisateur). Ce n'est pas le cas de MDIST et DEND qui travaillent uniquement avec une galerie à la fois. Les auteurs ont travaillé dans le cas du plus proche voisin avec la distance euclidienne comme mesure de similarité. Plusieurs méthodes d'édition existent dans la littérature et ont été testées pour la biométrie :

Condensed NN (CNN) [27]. Il s'agit d'un algorithme incrémental. Le but de cet algorithme est de trouver un ensemble édité E inclus dans T tel que l'instance $y \in T$ la plus proche de $x \in E$ possède la même étiquette que x. De cette façon, nous obtenons une performance de classification maximale sur T. Les données initiales sont ordonnées de n'importe quelle façon ; nous disposons de deux ensembles nommés E et Y et procédons comme ci-dessous:

1. Le premier élément est placé dans E.
2. Le second élément est classifié par la règle des plus proches voisins en utilisant comme ensemble de référence le contenu actuel de E. Si ce second élément est classé correctement, il est placé dans Y ; autrement il est placé dans E.
3. En procédant itérativement, l'élément i est classifié grâce au contenu actuel de E. Si il est classifié correctement, il est placé dans Y ; autrement, il est placé dans E.
4. Après une passe sur l'ensemble de données initiales, la procédure continue de boucler à travers Y, jusqu'à la terminaison qui peut arriver de deux façons différentes:
   - Y est vide, tous ces membres sont maintenant dans E (dans ce cas, l'ensemble est identique à l'ensemble original), ou
   - Une passe complète est faite sur Y avec aucun transfert sur E (les boucles suivantes amèneront au même résultat).
5. Le contenu final de E est utilisé comme galerie.

Freni [22] initialise E avec un exemple de chaque classe.

Selective NN (SNN) [60] Cette méthode est une évolution de la précédente essayant de générer un ensemble E le plus petit possible. Les prototypes sélectionnés sont plus proches de la frontière de décision qu'avec la méthode précédente. Freni [22] initialise E avec un exemple de chaque classe.

Reduced NN (RNN) [23] Il s'agit d'une méthode décrémentale. Les instances sont progressivement supprimées de E, tant que leur suppression n'implique pas une mauvaise classification sur T. L'algorithme s'arrête donc lorsqu'il n'est plus possible de supprimer d'instances à E. Cependant, la méthode n'est pas appliquée à partir de l'ensemble initial, mais à partir du résultat de CNN qui n'est pas considéré comme étant minimal.

1. Copie des données résultats de CNN appliqué à T dans E.
2. Suppression du premier élément de E.
3. Utilisation de E pour classifier tous les éléments de T:
   - si tous les éléments sont classifiés correctement, aller à 4,
   - si un élément est classifié de façon incorrecte, remettre l'élément retiré dans E, puis, aller à 4.
4. Si tous les éléments de E ont été supprimé une fois (en ayant éventuellement été ré-insérés), alors la procédure s'arrête. Sinon, il faut retirer l'élément suivant et réitérer à 3.

Edited NN (ENN) [77] Cette méthode est également décrémentale. Les instances sont progressivement supprimées de E si elles ne correspondent pas avec la majorité de ces k plus proches voisins (avec k = 3, les méthodes précédemment présentées utilisent k = 1). L'étude montre que le risque de cette méthode approche le risque bayésien.

1. Copie de T dans E.
2. Selection de x, le premier élément de E.
3. Récupération de Y l'ensemble des éléments de T étant les k plus proches voisins de x.
4. m est la classe majoritaire de Y.
5. Si l'étiquette de x est différente de m, alors supprimer x de E.
6. Si tous les éléments de E ont été séléctionnés, alors la procédure s'arrête. Sinon, il faut sélectionner l'élément suivant et réitérer à 2.

Il semble également que ces méthodes n'aient été utilisées que dans le cas supervisé. Les auteurs montrent que les méthodes d'édition sont compétitives vis-à-vis de celles de l'état de l'art (*MDIST* et *DEND*), et, qu'elles permettent de fortement diminuer la taille de la galerie (7 représentants à la place de 50 pour la méthode la plus agressive, CNN), d'autant plus que l'administrateur n'a pas à se préoccuper de la taille des galeries. Cependant, les performances sont meilleures car les méthodes peuvent avoir un nombre conséquent d'éléments dans les galeries. Les méthodes ne peuvent être utilisées que lorsqu'un nombre suffisant de données est accessible. Ces méthodes ont été vérifiées sur la base de visage EQUINOXE en sélectionnant 50 clients avec 100 images. La moitié des données a été utilisée pour la sélection des prototypes, tandis que l'autre a été utilisée pour le test.

Ces méthodes ne sont donc pas réellement des méthodes de *mise à jour du modèle biométrique*, mais elles permettent de diminuer la taille de la galerie qui est susceptible de grossir tout au long du système de *mise à jour du modèle biométrique*, rendant la vérification de plus en plus lente. Les méthodes basées sur l'édition nécessitent d'appliquer le processus sur l'ensemble des données d'enrôlement. Il y a donc des données de tous les utilisateurs. Cette méthode est donc applicable aux modalités proposant le même type de données biométriques quel que soit l'utilisateur (i.e., la reconnaissance faciale, où tout le monde peut donner une photo de son visage), mais pas aux modalités proposant des des données biométriques n'ayant aucune relation en fonction des utilisateurs (i.e., la reconnaissance de la dynamique de frappe, lorsque chaque utilisateur a un mot de passe différent).

Les méthodes de remplacement Le but des méthodes de remplacement est de ne pas faire grossir la galerie au cours des différentes mises à jour. Cependant, il s'agit bien de méthodes spécifiques à la *mise à jour du modèle biométrique*, car elles utilisent les nouvelles données capturées tout au long de l'utilisation du système biométrique (certaines sont spécifiques au mode en ligne, d'autres au mode hors ligne, et d'autres disposent de variantes dans les deux cas). Différentes méthodes de remplacement existent. Nous présentons dans cette partie la plupart d'entre elles. Scheidat *et al.* [68] présentent plusieurs techniques intéressantes de mise à jour de *modèle biométrique*. Cependant, leurs méthodes sont adaptées aux techniques utilisant une seule référence pour la comparaison (et donc des mécanismes de plus proches voisins

pour obtenir la distance de comparaison entre une requête et un modèle (l'ensemble des éléments de la galerie)). Les méthodes de cet article ont juste été présentées, mais pas testées.

Remplacement aléatoire Freni *et al.* [21] ont testé différentes méthodes de remplacement. L'une d'entre elle est le remplacement d'un exemple, de l'ensemble d'apprentissage, sélectionné aléatoirement, par la nouvelle requête. Naturellement, il ne s'agit pas de la méthode la plus performante, cependant, Freni *et al.* ont montré que plus la taille de la galerie est importante [7], plus les autres méthodes (plus complexes, et censées être plus efficaces) se rapprochent de la méthode de remplacement aléatoire (nommé RANDOM, dans leur papier). Des méthodes plus complexes n'ont un sens que si la taille de la galerie est relativement faible.

Fenêtre glissante Une des principales techniques de remplacement est basée sur le remplacement des captures les plus anciennes. En mode *hors ligne*, l'ensemble des données d'apprentissage est remplacé par l'ensemble des nouvelles données. Cette technique est appelée *BATCH-UPDATE* dans [75] et c'est révélée efficace dans le cas de la *mise à jour du modèle biométrique* d'un système basé sur les empreintes digitales (TEE de 7.69% au lieu de 10.32%). En mode *en ligne*, la donnée la plus ancienne est remplacée par la nouvelle donnée fraichement acceptée. Cette technique est appelée *moving window* dans [33] et *First In First Out (FIFO)* dans [21, 34, 68].

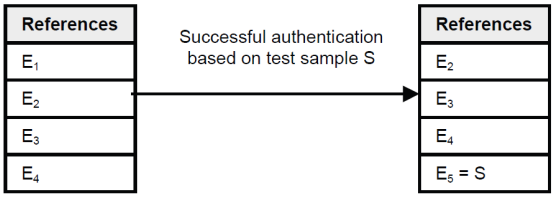


FIGURE 1.15 – Mécanisme de remplacement en utilisant une fenêtre glissante. (source : Scheidat *et al.* [68])

Moins fréquemment utilisé Cette technique est présentée dans [21, 68] sous le nom de *Least Frequently Used (LFU)*. Le principe est de remplacer l'exemple le moins souvent utilisé pour la vérification de l'utilisateur. Il est donc nécessaire de maintenir le nombre d'utilisations de chaque exemple de la galerie en tant que prototype ayant authentifié l'utilisateur. Le problème majeur de cette technique est le fait que les captures présentes depuis le plus longtemps dans le *modèle biométrique* ont nécessairement été les captures les plus proches le plus souvent. Elles ont donc un poids plus important que des captures plus récentes potentiellement de meilleures qualités.

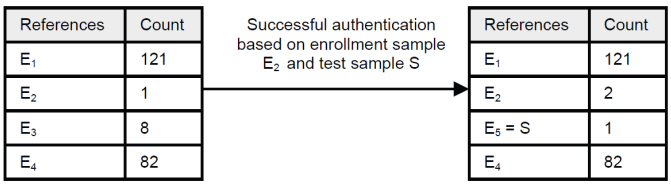
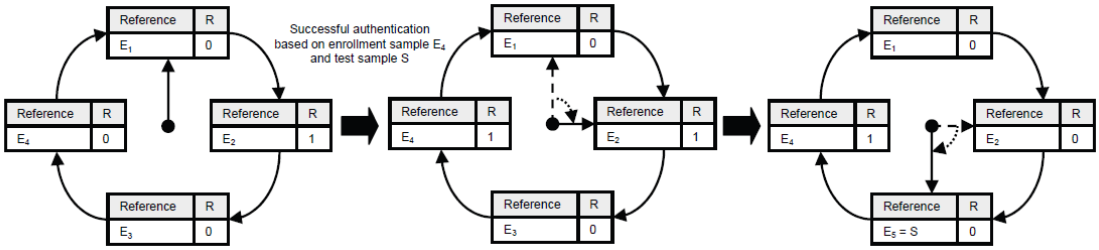


FIGURE 1.16 – Mécanisme de remplacement en remplaçant l'exemple le moins souvent utilisé. (source : Scheidat *et al.* [68])

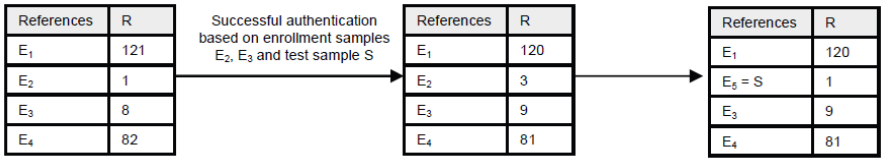
Moins récemment utilisé Cette technique est également présentée dans [68] sous le nom de *Least Recently Used (LRU)*. La méthode considère qu'il est préférable que la nouvelle donnée remplace celle qui a été utilisée le moins récemment lors des précédentes vérifications. Une première méthode est d'utiliser un timestamp avec chaque exemple de la galerie, ce qui peut être trop couteux [68]. Les auteurs proposent deux algorithmes pour contourner ce problème : la méthode de la seconde chance (*Second Chance (clock) algorithm* dans la littérature anglophone, et, une méthode étendue.

7. Il faut noter, que pour cette étude, relativement importante n'a pas forcément de sens, étant donné que les utilisateurs ont 8 captures

– seconde chance. Cette méthode permet de remplacer un vieil exemple, et non pas l'exemple le plus vieux. Les exemples sont stockés dans une liste circulaire, chaque exemple étant étiqueté par une valeur de référence valant 0 ou 1. Les exemples sont tous initialisés avec la valeur 0. Lorsqu'un exemple est responsable de l'acceptation d'un utilisateur, il est étiqueté à 1. Lorsqu'il est nécessaire de mettre à jour la galerie (*i.e.*, remplacer un élément par la nouvelle capture), le système parcourt la liste afin de déterminer quel est l'exemple à remplacer. A chaque fois qu'il rencontre un exemple dont la valeur de référence vaut 1, il réinitialise cette valeur à 0. Dès qu'il rencontre un élément dont la valeur de référence vaut 0, il s'agit d'un exemple qui n'a pas été utilisé depuis le dernier parcours de la liste. Il s'agit donc de la référence la moins récemment utilisée. L'exemple est donc remplacé par le nouveau.
– méthode étendue. L'approche de la seconde chance n'est pas spécifique aux systèmes biométriques, elle est principalement utilisée dans d'autres contextes. La méthode étendue tire parti des informations inhérentes à la biométrie, afin d'améliorer les performances de l'algorithme de remplacement. Ainsi, un attribut R est ajouté à chaque exemple de la galerie, afin de quantifier sa pertinence. Les attributs sont initialisés à 0. Supposons qu'après la classification, il y a i meilleurs candidats avec différents scores de reconnaissance. Si ces exemples sont suffisamment proches de la requête, et que le système décide de l'accepter, cela signifie que le système peut en remplacer un par la requête. Les i meilleurs exemples sont donc triés par ordre descendant de leur attribut de pertinence R. L'attribut de pertinence du premier est incrémenté de i, celui du second est incrémenté de $i-1$ et ainsi de suite. L'exemple ayant l'attribut de pertinence R le plus faible est remplacé par la requête. Les autres exemples de la galerie n'ayant pas pris part au processus, voient leur valeur de R décrémentée de 1. De cette façon, une hiérarchie linéaire est créée pour chaque utilisateur à chaque moment, ainsi la pertinence de chaque exemple de la hiérarchie est définie par l'attribut R. La FIGURE 1.17(b) illustre le mécanisme lorsque les exemples E2 et E3 sont sélectionnés pour l'authentification, et, que leur valeur de pertinence est respectivement incrémentée de 2 et 1. L'exemple E2 ayant la plus petite valeur de pertinence est remplacé par la requête S.



(a) algorithme d'horloge



(b) algorithme étendu

FIGURE 1.17 – Mécanisme de remplacement en utilisant le principe de moins récemment utilisé. (source : Scheidat *et al.* [68])

Les techniques de remplacement permettent d'obtenir une galerie dont la taille ne croit pas au cours du temps. Elles peuvent donc être utilisées dans les systèmes ne disposant pas de beaucoup d'espace pour stocker le *modèle biométrique* d'un utilisateur.

Les méthodes d'ajout Le principe d'utiliser des méthodes d'ajout, est de grossir progressivement la taille de la galerie du *modèle biométrique*. Cette technique permet de commencer à utiliser le système biométrique après une période d'enrôlement relativement courte (n'encodant pas forcément une grande variabilité) et d'augmenter la taille de celle-ci au cours du fonctionnement du système biométrique (et potentiellement encoder une plus grande variabilité de l'utilisateur). Uludag *et al.* [75] présente une méthode nommée *AUGMENT-UPDATE* qui ajoute en mode hors ligne l'ensemble des nouvelles données dans la galerie de l'utilisateur. La taille de la galerie croit donc au fil du temps. Kang et Cho [32] utilisent également cette technique dans un système de dynamique de frappe au clavier, avec un système de mise à jour semi-supervisé : les données ajoutées au modèle sont celles étant reconnues par le classifieur (et pas celle ayant la bonne étiquette). D'autres variantes ont été proposées dans la littérature: les nouvelles données sont progressivement ajoutées au *modèle biométrique* jusqu'à obtenir un nombre maximal de données dans la galerie. Une fois ce nombre maximal atteint, la galerie est modifiée en utilisant des systèmes de remplacement. Grabham et White [26] ont implémenté avec succès cette méthode dans un système d'authentification par dynamique de frappe au clavier sur un clavier avec des capteurs de pression.

1.4.4.3 Classifieur

Dans le cas de l'utilisation d'un classifieur, le modèle n'est plus constitué d'une galerie ou d'une capture unique, mais des paramètres de ce classifieur. Au lieu de mettre à jour le classifieur, en effectuant à nouveau l'apprentissage en mode batch avec l'ensemble des données d'apprentissage augmenté de la nouvelle capture, le classifieur doit être entrainé en ligne afin d'en adapter les paramètres. L'intérêt d'utiliser une telle technique, plutôt qu'une méthode à base de galerie qui recalcule les paramètres du classifieur, après chaque ajout de nouvelles données, est de grandement diminuer les temps de calcul et de pouvoir être utilisé avec d'énormes bases de données.

Les séparateurs à vaste marge Singh *et al.* [71] présentent un système de mise à jour en ligne supervisé pour la reconnaissance faciale 2D proche infrarouge. Le classifieur est un « 2ν-Online Granular Soft Support Vector Machine ». Le calcul granulaire permet de s'adapter aux variations globales et locales dans la distribution des données, et, les étiquettes souples permettent d'être résistantes au bruit. Le fonctionne d'un SVM classique est présenté en annexe B. Les paramètres optimaux (le C et les paramètres de la fonction noyau) sont obtenus en testant manuellement en testant plusieurs jeux de paramètres, jusqu'à obtenir un taux d'erreur optimal. Le double ν-SVM (2ν-SVM), proposé dans [14] est une variante du SVM calculatoirement plus efficace. Il est plus flexible pendant l'apprentissage et surmonte les problèmes lorsque les classes n'ont pas les mêmes quantités d'apprentissage. Des paramètres additionnels (ρ, v et C_i) sont introduits, dans l'equation (B.8) page 57 et l'equation (B.7) page 57. La formulation devient donc :

$$\begin{cases} \min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} C_i(v\rho - \xi_i) \\ \forall i, t_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq \rho - \xi_i, \rho, \xi_i >= 0 \end{cases} \tag{1.26}$$

ρ est la position de la marge, et v est le paramètre d'erreur qui peut être calculé en utilisant v_+ et v_- qui sont les paramètres d'erreur en apprenant les classes positives et négatives.

$$v = \frac{2v_+v_-}{v_+ + v_-} \tag{1.27}$$

$C_i(v\rho - \xi_i)$ est le cout de l'erreur et C_i est la pénalité d'erreur pour chaque classe qui est calculée comme :

$$C_i = \begin{cases} C_+, & \text{si } y_i = +1 \\ C_-, & \text{si } y_i = -1 \end{cases} \tag{1.28}$$

avec

$$\begin{aligned} C_+ &= \frac{v}{2n_+v_+} \\ C_- &= \frac{v}{2n_-v_-} \end{aligned} \tag{1.29}$$

avec n_+ et n_- respectivement le nombre d'exemples positifs et négatifs. Ainsi, la fonction objective du 2ν-SVM peut être réécrite (formulation duale de Wolfe) :

$$L = \sum_i \alpha_i - \left\{ \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \right\} \tag{1.30}$$

avec $i, j \in 1, \ldots, N$, α_i, α_j les multiplicateurs de Lagrange tels que $0 <= \alpha_i <= C_i$, $\sum_i \alpha_i y_i = 0$ et $\sum_i \alpha_i >= v$. Comme pendant l'apprentissage, il est possible que certains exemples soient bruités ou mal étiquetés, le 2ν-SVM est susceptible d'effectuer des erreurs de classification. Pour contourner ce problème, il est possible d'utiliser des étiquettes souples [73]. L'utilisation d'un tel mécanisme peut diminuer les erreurs de classification, ainsi que le nombre de vecteurs supports. Notons z_i l'étiquette souple du i^{e} exemple d'apprentissage $\mathbf{x}_i$. Le 2ν-SVM peut être transformé en 2ν-Soft SVM (2ν-SSVM) comme ceci :

$$\begin{cases} \min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} C_i(v\rho - \xi_i) \\ \forall i, z_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq z_i^2(\rho - \xi_i) \end{cases} \tag{1.31}$$

Même avec un 2ν-SSVM, l'apprentissage d'une grosse base de données est consommateur de temps. Le calcul granulaire [8] est basé sur un principe de *diviser pour régner*, et, permet de réduire le temps de calcul. Singh *et al.* [71] proposent donc le 2ν-Granular SSVM en divisant l'espace en c sous espaces avec un 2ν-SSVM pour chaque sous espace. Cette division des sous-espaces est appliquée en utilisant un algorithme de partitionnement du type C-moyenne flou, qui, génère des partitions d'exemples semblables. Notons 2ν-SSVM$_i$ le i^{e} 2ν-SSVM et 2ν-SSVM$_i :\rightarrow L_i$ représente le 2ν-SSVM opérant sur le sous espace i ($i = 1, 2, \ldots, c$). La largeur W, issue de la composition des marges des différents classifieurs, est calculée comme ceci :

$$W = \left| \sum_{i=1}^{c} \frac{t_i}{t} (2\nu - SSVM_i :\rightarrow L_i) - L0 \right| \tag{1.32}$$

avec

$$t = \sum_{i=1}^{c} t_i \tag{1.33}$$

où t_i est le nombre d'exemples d'apprentissage dans le i^{e} sous espace. L'apprentissage du 2ν-SSVM génère L_i au niveau local et L_0 est obtenu en apprenant un autre 2ν-SSVM sur l'espace complet des attributs au niveau global. Le 2ν-Granular SSVM a été entrainé en mode batch lors de l'enrôlement, puis en ligne au cours de l'utilisation du système biométrique. La mise à jour consiste à ajouter de nouveaux vecteurs supports (qui sont linéairement indépendants), et à supprimer les anciens vecteurs supports qui n'augmentent pas les performances du classifieur. De cette façon, le nombre de vecteurs supports n'augmente pas énormément au fil de l'utilisation du système de mise à jour. Il s'agit donc à la fois d'un algorithme incrémental et décrémental. L'algorithme du 2ν-Online GSSVM est le suivant :

1. Le 2ν-GSSVM est entrainé en utilisant l'ensemble initial d'apprentissage et un hyperplan de décision est obtenu avec les m vecteurs supports.
2. Pour chaque nouvel exemple $\bar{\mathbf{x}}_i$,
   (a) $\bar{\mathbf{x}}_i$ est classifié en utilisant le 2ν-GSSVM.
   (b) Le résultat de classification est comparé avec l'étiquette $\bar{\mathbf{z}}_i$; Si la classification est correcte, alors il n'est pas nécessaire d'effectuer le réapprentissage.
   (c) Sinon,
      i. L'hyperplan de décision est recalculé en utilisant les m vecteurs supports et $\{\bar{\mathbf{x}}_i, \bar{\mathbf{z}}_i\}$ à l'aide de l'apprentissage en mode batch. Cette fois-ci l'apprentissage est plus rapide, il y a moins de données d'apprentissage.
      ii. Après avoir recalculé l'hyperplan, le nombre de vecteurs supports s'accroit. Si le nombre de vecteurs supports est supérieur à $m + \lambda$ (où λ est un seuil qui contrôle le nombre de vecteurs supports), alors un vecteur support le plus loin de la frontière de décision est sélectionné.

iii. Le vecteur support sélectionné est supprimé de la liste des vecteurs supports et stocké dans la liste l. Le classifieur avec $m + \lambda - 1$ vecteurs supports est utilisé pour la validation et le test.

3. Les vecteurs supports dans la liste l sont utilisés pour tester le nouveau classifieur. Si il y a des erreurs de classification, l'étape 2(c) est répétée pour minimiser l'erreur de classification.
4. Les vecteurs supports les moins récemment inclus sont supprimés de la liste, l, dans le classifieur final.

Le système a été utilisé dans un système à deux classes (+1 pour les clients, -1 pour le imposteurs), les attributs des données étant constitués des scores de différents systèmes de reconnaissance faciale. L'apprentissage en ligne c'est révélé calculatoirement plus efficace que l'apprentissage hors ligne, aussi bien pour l'apprentissage que le test.

Les cartes auto-adaptives Dozono *et al.* [18] utilisent une carte auto-adaptive [36] pour faire de la classification d'utilisateurs dans un système de reconnaissance de dynamique de frappe (temps de pression, intervalle entre deux touches et niveau sonore maximum lors de la pression de la touche). Le système est donc constitué d'un seul et unique *modèle biométrique* pour l'ensemble des utilisateurs. Celui-ci est appris en mode batch. Une fois les vecteurs représentatifs de chaque neurone correctement configurés, un label leur est associé. Ce label est le label (numéro de l'utilisateur) de la donnée la plus proche. La vérification se fait en récupérant le label du neurone pour lequel le vecteur représentatif est le plus proche du vecteur requête. L'étude présente une méthode de mise à jour supervisée et une méthode semi-supervisée. Dans le premier cas, le vecteur représentatif du neurone sélectionné est légèrement modifié, que la réponse soit correcte (ajout d'une portion de la différence) ou incorrecte (soustraction d'une portion de la différence entre le neurone et l'exemple). Dans le cas semi-supervisé, la mise à jour ne peut être faite que lorsque la donnée est acceptée. Dozono *et al.* [18] ont nommé leur carte auto-adaptive : Multi-Winner SOM (MW-SOM). L'apprentissage de la carte se fait de la façon suivante :

1. Initialisation de la carte en générant des vecteurs aléatoires pour chaque neurone.
2. Séléction, aléatoirement, d'un exemple $\mathbf{x}$ des données d'apprentissage.
3. Recherche du nœud U_w associé au vecteur $\mathbf{w}_w$ le proche de x, en minimisant l'erreur $|\mathbf{x} - \mathbf{m}|$.
4. Le nœud sélectionné U_w et ses nœuds voisins $U\prime_w$ sont mis à jour en utilisant l'équation suivante :

$$\mathbf{m}_i = \mathbf{m}_i + fn(d) * \eta * (\mathbf{x} - \mathbf{m}_i) \tag{1.34}$$

   avec $\mathbf{m}_i$ représentant le vecteur à mettre à jour, $fn(d)$ la fonction de voisinage décroissante en fonction de la distance entre U_w et $U\prime_w$ et η le taux d'apprentissage.
5. Répéter depuis l'étape 2, en diminuant la fonction de voisinage $fn(d)$ et le taux d'apprentissage η jusqu'à ce que l'erreur de classification converge ou au bout d'un certain nombre d'itérations.

La classification, à l'aide de l'algorithme MW-SOM, se fait de la façon suivante :

1. Après avoir appris la carte, chaque unité est étiquetée avec l'identifiant de la classe associé à l'exemple le plus proche du vecteur associé au nœud.
2. Chercher les n plus proches unités de la carte par rapport à la requête (même processus que pour l'apprentissage à l'étape 2).
3. Si plus de k unités parmi les n sont étiquetées avec la même classe, l'exemple appartient à la classe. Autrement, la classe est indéterminée.

Le système peut donc être paramétré en utilisant le seuil k. Dozono *et al.* [18] proposent une méthode incrémentale de mise à jour de la carte. Celle-ci est appelée toutes les trois authentifications, car des mises à jour trop nombreuses ont des effets négatifs sur la carte. L'équation suivante est utilisée si l'authentification est bonne :

$$\mathbf{m}\prime_{ij} = \mathbf{m}\prime_{ij} + \eta\prime(\mathbf{x}\prime - \mathbf{m}\prime_{ij}) \tag{1.35}$$

sinon, l'équation suivante est utilisée en cas d'erreur lors de l'authentification [8] :

$$\mathbf{m}\prime_{ij} = \mathbf{m}\prime_{ij} - \eta\prime(\mathbf{x}\prime - \mathbf{m}\prime_{ij}) \tag{1.36}$$

8. Les auteurs utilisent le terme authentification, mis il doit plutôt s'agir d'identification

$\eta\prime$ est le taux d'apprentissage LVQ1, $\mathbf{x}\prime$ est l'exemple requête, et $\mathbf{m}_{ij}$ est le vecteur associé aux unités gagnantes (nous avons vu que pour MW-SOM, il y a n unités gagnantes). L'étude montre que l'apprentissage semi-supervisé est relativement sensible au bruit (plus les données sont bruitées, plus les performances se dégradent), tandis que l'apprentissage supervisé est plus stable. Il faut noter que l'étude ne travaille qu'avec 10 utilisateurs et 10 captures par utilisateur (les données supplémentaires sont générées artificiellement en modifiant aléatoirement les données originales) ...

Les analyses en composantes El Gayar *et al.* [20] utilisent une Analyse en Composante Principale pour réduire la dimension des données manipulées dans le cadre d'un système de reconnaissance faciale. Le système utilise des mécanismes d'auto apprentissage et de co apprentissage sur un jeu de données non labellisé. Ce qui nous importe dans cette section, n'est pas la façon de collecter ces données, mais la façon dont le modèle est mis à jour. $2/3$ des données de test sont utilisées pour la mise à jour, et $1/3$ est utilisé pour la validation. La mise à jour consiste simplement à recalculer l'ACP sur le nouveau jeu de données (qui n'intègre que les cinq meilleures nouveaux exemples pour chaque classe).

Vandana [76] propose la mise à jour itérative de l'espace propre de chaque utilisateur. Le modèle d'un utilisateur intègre les différentes informations nécessaires à effectuer le changement de repère de la capture requête afin de la projeter dans cet espace propre (calculé à l'aide des données d'enrôlement de l'utilisateur (classe 0), et, des données d'enrôlement des autres utilisateurs (classe 1)). La vérification utilise un algorithme de *kppv*. La méthode employée en mode batch pour calculer l'espace propre initial est baptisée : Incremental Biased Discriminant Analysis (IBDA). Le modèle d'un utilisateur est constitué des données suivantes :

$$\mathbf{\Omega} = (S_x; S_y; \mathbf{m}_x; \mathbf{m}_y; N_x; N_y) \tag{1.37}$$

Nous ne présentons que la modification itérative du modèle, et pas son calcul initial. S_x, S_y, $\mathbf{m}_x$, $\mathbf{m}_y$, N_x, N_y représentent respectivement la dispersion intraclasse, la dispersion inter-classe, la moyenne des exemples positifs, la moyenne des exemples négatifs, le nombre d'exemples positifs, et, le nombre d'exemples négatifs. Notons X, l'ensemble des données clientes, et Y l'ensemble des données imposteurs. $\mathbf{x}_{nouv}$ est la nouvelle donnée à intégrer au modèle. Le but de la mise à jour itérative, est de n'utiliser que $\mathbf{\Omega}$ et $\mathbf{x}_{nouv}$ afin d'obtenir :

$$\mathbf{\Omega}' = (S'_x; S'_y; \mathbf{m}'_x; \mathbf{m}'_y; N'_x; N'_y) \tag{1.38}$$

Le nouveau modèle obtenu ($\mathbf{\Omega}'$) sera utilisé pour les vérifications futures. Cette mise à jour s'effectue en deux étapes :

1. Mise à jour du modèle pour la classe positive, *i.e.,* $x_{nouv} \in X$. Dans ce cas, l'ensemble des exemples négatifs ne change pas, il n'est pas nécessaire de recalculer les paramètres du modèle qui en dépendent. Il faut incrémenter le nombre d'exemples de la classe positive :

$$N'_x = N_x + 1 \tag{1.39}$$

puis mettre à jour la moyenne des exemples positifs :

$$\mathbf{m}'_x = \frac{N_x \mathbf{m}_x + \mathbf{x}_{nouv}}{N_x + 1} \tag{1.40}$$

la dispersion intraclasse des exemples positifs devient :

$$S'_x = S_x + \frac{(\mathbf{x}_{nouv} - \mathbf{m}_x)(\mathbf{x}_{nouv} - \mathbf{m}_x)^T}{N_x + 1} \tag{1.41}$$

la dispersion interclasse en fonction de la moyenne des exemples positifs devient :

$$\begin{aligned} S'_y &= S_y \\ &\quad - \frac{Ny}{N_x + 1}[(\mathbf{m}_y - \mathbf{m}_x)(\mathbf{x}_{nouv} - \mathbf{m}_x)^T + (\mathbf{x}_{nouv} - \mathbf{m}_x)(\mathbf{m}_y - \mathbf{m}_x)^T] \\ &\quad + \frac{N_y}{(N_x + 1)^2}(\mathbf{x}_{nouv} - \mathbf{m}_x)(\mathbf{x}_{nouv} - \mathbf{m}_x)^T \end{aligned} \tag{1.42}$$

2. Mise à jour du modèle pour la classe négative, *i.e.,* $\mathbf{x}_{nouv} \in \mathbf{Y}$. Dans ce cas, l'ensemble des exemples positifs ne change pas, il n'est pas nécessaire de recalculer les paramètres du modèle qui en dépendent. Il faut incrémenter le nombre d'exemples de la classe négative :

$$N'_y = N_y + 1 \tag{1.43}$$

puis mettre à jour la moyenne des exemples négatifs :

$$\mathbf{m}'_y = \frac{N_y \mathbf{m}_y + \mathbf{x}_{nouv}}{N_y + 1} \tag{1.44}$$

la dispersion des exemples négatifs devient :

$$S'_y = S_y + \frac{(\mathbf{x}_{nouv} - \mathbf{m}_y)(\mathbf{x}_{nouv} - \mathbf{m}_y)^T}{N_y + 1} \tag{1.45}$$

L'intérêt d'utiliser cette façon itérative est multiple :
- gain de place : il n'est pas nécessaire de stocker l'ensemble des exemples ;
- gain de temps : les calculs sont nettement moins couteux qu'avec l'ensemble de données.

Les auteurs ont montré une amélioration significative des performances de leur système, en utilisant le mécanisme de mise à jour du modèle comparé au même système sans mise à jour.

Liu *et al.* [40] ont également utilisé un mécanisme de mise à jour de vecteurs propres de façon itérative. Leur étude porte sur la reconnaissance faciale, ou l'Analyse en Composante Principale (ACP)[15] est effectuée pour chaque utilisateur. Contrairement au cas précédent, ils partent du principe que plus les données sont anciennes, plus elles sont obsolètes, et moins il est nécessaire de les prendre en compte dans le modèle. Ainsi, la moyenne et la matrice de covariance sont mis à jour en attribuant des poids aux exemples. Plus les exemples sont anciens, plus le poids est faible. Afin de profiter de l'intérêt de cette mise à jour en ligne, les deux calculs sont effectués de façon récursive. Ainsi la moyenne $\hat{\mathbf{m}}_n$ à l'instant n est calculée de la façon suivante :

$$\hat{\mathbf{m}}_n = \alpha_m \hat{\mathbf{m}}_{n-1} + (1 - \alpha_m)\mathbf{x}_n \tag{1.46}$$

avec $\mathbf{x}_n$ le nouvel exemple à intégrer dans le modèle et α_m le facteur de pondération indiquant la vitesse d'oubli des anciens exemples. La valeur de α_m dépend principalement de la connaissance de l'évolution du processus aléatoire (le visage dans ce cas). La matrice de covariance $\hat{\mathbf{C}}$ est mise à jour de la façon suivante :

$$\hat{\mathbf{C}}_n = \alpha_v \hat{\mathbf{C}}_{n-1} + (1 - \alpha_v)(\mathbf{x}_n - \hat{\mathbf{m}}_n)(\mathbf{x}_n - \hat{\mathbf{m}}_n)^T \tag{1.47}$$

avec α_v le facteur de pondération a choisir en fonction de la vitesse de modification de la covariance du processus aléatoire. Une fois $\hat{\mathbf{C}}_n$ obtenu à l'instant n, l'ACP peut être effectuée en récupérant les vecteurs propres correspondants. Les N vecteurs propres correspondant aux N plus grosses valeurs propres sont conservés. Ainsi, il est nécessaire de ne stocker que le vecteur moyen $\hat{\mathbf{m}}_n$ et la matrice de covariance $\hat{\mathbf{C}}_n$ dans ce système de mise à jour récursif. Les exemples ayant servit au calcul n'ont pas besoin d'être conservés. Cependant, la quantité de données nécessaire au stockage reste relativement importante : si les images ont une résolution de 32×32, la matrice de covariance est en 1024×1024. Liu *et al.* [40] proposent donc une méthode de mise à jour basée sur le produit scalaire de deux matrices en utilisant une approximation de $\hat{\mathbf{C}}_{n-1}$ en ne conservant que les Q premiers vecteurs propres (ceux ayant les plus grandes valeurs propres). Ainsi, $\hat{\mathbf{C}}_{n-1}$ est approximé de la façon suivante :

$$\hat{\mathbf{C}}_{n-1} \approx \lambda^{(1)}_{n-1}\varphi^{(1)}_{n-1}\varphi^{(1)T}_{n-1} + \lambda^{(2)}_{n-1}\varphi^{(2)}_{n-1}\varphi^{(2)T}_{n-1} + \cdots + \lambda^{(Q)}_{n-1}\varphi^{(Q)}_{n-1}\varphi^{(Q)T}_{n-1} \tag{1.48}$$

avec $\varphi^{(i)}_{n-1}$ les vecteurs propres et $\lambda^{(i)}_{n-1}$ les valeurs propres telles qu'elles soient triées par ordre descendant, avec l'exposant (i) indiquant l'ordre des valeurs propres. Les auteurs présentent différentes façon de

choisir Q qui dépendent de l'application concernée (et ont une complexité calculatoire différente). Ainsi, en remplaçant $\hat{\mathbf{C}}_{n-1}$ dans l'equation (1.47) et l'equation (1.48), nous obtenons :

$$\begin{aligned}\hat{\mathbf{C}}_n &\approx \alpha_v \lambda_{n-1}^{(1)} \varphi_{n-1}^{(1)} \varphi_{n-1}^{(1)T} \\ &+ \alpha_v \lambda_{n-1}^{(2)} \varphi_{n-1}^{(2)} \varphi_{n-1}^{(2)T} \\ &+ \cdots \\ &+ \alpha_v \lambda_{n-1}^{(Q)} \varphi_{n-1}^{(Q)} \varphi_{n-1}^{(Q)T} \\ &+ (1-\alpha_v)(\mathbf{x}_n - \hat{\mathbf{m}}_n)(\mathbf{x}_n - \hat{\mathbf{m}}_n)^T\end{aligned} \tag{1.49}$$

qui possède une forme équivalente comme ceci :

$$\hat{\mathbf{C}}_n \approx \mathbf{B}_n \mathbf{B}_n^T \tag{1.50}$$

avec

$$\mathbf{B}_n = \left[\sqrt{\alpha_v \lambda_{n-1}^{(1)}} \varphi_{n-1}^{(1)} \quad \sqrt{\alpha_v \lambda_{n-1}^{(2)}} \varphi_{n-1}^{(2} \cdots \quad \sqrt{\alpha_v \lambda_{n-1}^{(Q)}} \varphi_{n-1}^{(Q} \quad \sqrt{1-\alpha_v}(\mathbf{x}_n - \hat{\mathbf{m}}_n) \right] \tag{1.51}$$

En utilisant la matrice $\mathbf{B}_n$, un produit scalaire peut être défini de cette façon :

$$\mathbf{A}_n = \mathbf{B}_n^T \mathbf{B}_n \tag{1.52}$$

Ainsi, $\mathbf{A}_n$ peut être décrit à l'aide des équations suivantes :

$$(A_n)_{i,j} = \alpha_v \sqrt{\lambda_{n-1}^{(i)} \lambda_{n-1}^{(j)}} \delta_{ij}, \quad i,j = 1,2,\ldots,Q \tag{1.53}$$

$$(A_n)_{i,Q+1} = (A_n)_{Q+1,i} = \sqrt{\alpha_v(1-\alpha_v)\lambda_{n-1}^{(i)}}(\mathbf{x}_n - \hat{\mathbf{m}}_n) \quad i,j = 1,2,\ldots,Q \tag{1.54}$$

$$(A_n)_{Q+1,Q+1} = (1-\alpha_v)(\mathbf{x}_n - \hat{\mathbf{m}}_n)^T(\mathbf{x}_n - \hat{\mathbf{m}}_n) \tag{1.55}$$

Comme $\mathbf{A}_n$ est une petite matrice de dimension $Q+1 \times Q+1$, ses vecteurs propres ψ_n peuvent être déterminés directement, ce qui donne :

$$\mathbf{A}_n \psi_n^{(i)} = \mathbf{B}_n^T \mathbf{B}_n \psi_n^{(i)} = \lambda_n^{(i)} \psi_n^{(i)} \quad i,j = 1,2,\ldots,Q+1 \tag{1.56}$$

En pré-mulitpliant l'equation (1.56) par $\mathbf{B}_n$, nous pouvons obtenir les vecteurs propres de la matrice de covariance $\hat{\mathbf{C}}_n$ de la façon suivante :

$$\varphi_n^{(i)} = \lambda^{(i)} - \frac{1}{2_n} \mathbf{B}_n \psi_n^{(i)} \quad i,j = 1,2,\ldots,Q+1 \tag{1.57}$$

avec le terme $\lambda^{(i)} - \frac{1}{2_n}$ permettant au vecteur propre d'être un vecteur unitaire. Afin de généraliser les exemples précédent, nous allons présenter des travaux plus récents [50]. Dans le cas d'un modèle statistique, Poh *et al.* [50] présentent une interprétation bayésienne de la mise à jour du *modèle biométrique*. Ils énumèrent trois opérations basiques essentielles dans la vie du *modèle biométrique*:

- La création du modèle :

$$Créer : données \rightarrow nouveau\ modèle \tag{1.58}$$

- L'adaptation du modèle :

$$Adapter : modèle, données \rightarrow modèle\ adapté \tag{1.59}$$

- La suppression du modèle:

$$Supprimer : modèle \rightarrow \emptyset \tag{1.60}$$

Ajouter un *modèle biométrique* et le supprimer est relativement facile, ce qui est loin d'être le cas pour la mise à jour qui nécessite la combinaison de plusieurs exemples. Comme nous avons pu le voir, les classifieurs basés sur des modèles statistiques peuvent souvent être implantés dans des formes en ligne (*i.e.*, les paramètres de l'ancien modèle peuvent être mis à jour avec les nouveaux exemples obtenus après sa création). Le classifieur est de la forme :

$$p(\boldsymbol{\theta}|\mathbf{x}) \propto p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{1.61}$$

avec $p(\mathbf{x}|\boldsymbol{\theta})$, la probabilité de l'exemple (suivant le modèle paramétré par $\boldsymbol{\theta}$) et $p(\boldsymbol{\theta})$ est la probabilité à priori du paramètre $\boldsymbol{\theta}$. Maximiser le côté droit de l'équation permet de trouver la valeur de $\boldsymbol{\theta}$ la plus probable : $\boldsymbol{\theta}^\star$. C'est cette valeur, estimée lors de la création du modèle, qui peut être mise à jour lorsqu'un nouvel exemple est disponible. Si nous disposons de T exemples $\mathbf{x}_1, \ldots, \mathbf{x}_T$, pour trouver la valeur de $\boldsymbol{\theta}$ (avec le maximum à priori) qui maximises $p(\boldsymbol{\theta}|\mathbf{x}_1 : \mathbf{x}_T)$, il faut suivre (en ignorant le facteur de normalisation à chaque étape étant donné que nous sommes uniquement intéressé à maximiser la fonction en fonction $\boldsymbol{\theta}$) :

$$\begin{array}{rcl} p(\boldsymbol{\theta}|\mathbf{x}_1 : \mathbf{x}_T) & \propto & \prod_{i=1}^{T} p(\mathbf{x}_i|\boldsymbol{\theta})p(\boldsymbol{\theta}) \\ & \propto & \prod_{i=2}^{T} p(\mathbf{x}_i|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{x}_1) \\ & \propto & \prod_{i=3}^{T} p(\mathbf{x}_i|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{x}_1, \mathbf{x}_2) \\ & \propto & \vdots \\ & \propto & p(\mathbf{x}_T|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{x}_1 : \mathbf{x}_{T-1}) \end{array} \tag{1.62}$$

avec $p(\boldsymbol{\theta}|\mathbf{x}_1) \propto p(\mathbf{x}_1|\boldsymbol{\theta})p(\boldsymbol{\theta})$. La forme récursive de l'equation (1.62) implique que pour calculer la valeur optimale de $\boldsymbol{\theta}$ grâce aux T précédents exemples, il est uniquement nécessaire d'utiliser les paramètres calculés à l'itération $T-1$. De cette façon, il n'est pas nécessaire de stocker l'ensemble des exemples précédents (ce qui est un gain de mémoire non négligeable). Poh *et al.* [50] ont aussi modélisé la mise à jour pour des modèles non statistiques. Ils présentent les techniques de super-modèles (voir la section 1.4.4.1 page 28) comme un modèle statistique simplifié sous la forme d'une distribution gaussienne multivariée avec une covariance isotrope. Soit x_m^t une minutie à l'emplacement m de la t^e empreinte digitale. Les emplacements des minuties ont tous été alignés avant le traitement. Une minutie peut contenir son emplacement et des informations de direction. En partant du principe que les minuties sont indépendantes, un modèle peut être représenté par $p(\mathbf{X}|\boldsymbol{\theta}) = \prod_m p(x_m|\boldsymbol{\theta})$ avec $p(x_m|\theta)$ qui suit une loi normale et $\boldsymbol{\theta} = \epsilon$ est commun à toutes les observations t et emplacements m :

$$\begin{array}{rcl} p(\mathbf{X}|\boldsymbol{\theta}, t) & = & \prod_{m=1}^{M} p(x_m|\boldsymbol{\theta}, t) = \prod_{m=1}^{M} \mathcal{N}(x_m|\mu_m^t, \epsilon) \\ & = & \frac{1}{(2\pi\epsilon)^{M/2}} exp\left\{-\frac{1}{2\epsilon}\sum_{m=1}^{M} ||x_m - \mu_m^t||^2\right\} \end{array} \tag{1.63}$$

μ_m^t est l'emplacement (et orientation) d'une minutie du t^e modèle. ϵ correspond à la variance de x_m. La probabilité à postériori devient :

$$\begin{array}{rcl} p(\boldsymbol{\theta}|\mathbf{X}_1 : \mathbf{X}_T) & \propto & p(\mathbf{X}_1 : \mathbf{X}_T|\boldsymbol{\theta})p(\boldsymbol{\theta}) \\ & = & \prod_{t=1}^{T} p(\mathbf{X}_t|\boldsymbol{\theta}) \\ & \propto & exp\left\{-\frac{1}{2\epsilon}\sum_{t=1}^{T}\sum_{m=1}^{M} ||x_m - \mu_m^t||^2\right\} \end{array} \tag{1.64}$$

avec une égalité seulement en utilisant le facteur de normalisation $\frac{1}{(2M\pi\epsilon)^{MT/2}}$. Ici, $p(\boldsymbol{\theta})$ est un distribution uniforme sur l'espace de $\boldsymbol{\theta}$. Quand $\epsilon \to 0$, alors la minutie x_m^t qui est proche de μ_m^t aura une valeur importante, sinon proche de zéro. Avec un grand nombre d'exemples T, les minuties peu nombreuses ont un poids faible, et, jouent un rôle moins important dans la reconnaissance. Dans le cas des mises à jour d'espaces propres, $p(\mathbf{x}|\boldsymbol{\theta})$ est une gaussienne multivariée avec comme paramètres $\boldsymbol{\theta}$, la moyenne et la matrice de covariance.

1.4.4.4 Discussion

Nous avons vu que de nombreuses techniques de gestion d'ajout d'une capture dans le *modèle biométrique* existent. Chacune d'entre elles est plus ou moins adaptée à telle ou telle modalité. Cependant,

nous pouvons remarquer que la plupart des méthodes sont vouées à des systèmes biométriques dont la vérification est basée sur le calcul d'une distance à une autre capture ou de l'utilisation des k plus proches voisins. Il est fort probable qu'une majorité de ces techniques ne soient pas utilisables dans le cas de la dynamique de frappe au clavier par exemple. Rattani *et al.* [57] ne se contentent pas d'utiliser une seule des méthodes dans leur système basé le co-apprentissage. Ils utilisent deux techniques différentes en fonction de leur cas :

- Si la modalité A accepte avec forte probabilité la requête mais pas la modalité B, alors la requête de la modalité (il s'agit d'une requête facile) A est fusionnée (création d'un super modèle) à l'élément de la galerie l'ayant vérifié, et, la requête de la modalité B (il s'agit d'une requête difficile) est ajoutée à la galerie.
- Si les deux modalités acceptent la requête avec une forte probabilité, alors une fusion est opérée pour ces deux modalités.

Poh *et al.* [50] insistent sur le fait qu'il est nécessaire de prendre en compte des informations de qualité avant d'appliquer la mise à jour du *modèle biométrique*. L'utilisation de cet indice de qualité peut permettre d'avoir plusieurs modèles (un par indice de qualité), et, évite de diminuer fortement les performances. Cette assertion est due au fait que les auteurs considèrent que la variabilité *intra-classe* peut être beaucoup plus importante que la variabilité *inter-classe* (*cf.* deux images du visage du même utilisateur sous des conditions d'illumination totalement différentes et deux images de deux utilisateurs différents sous des conditions d'illumination identiques). Incorporer ces deux informations (*i.e.*, images du même utilisateur avec des conditions d'illuminations différentes) sans prendre en compte un indice de qualité (*i.e.*, la condition d'illumination), réduira nécessairement son pouvoir de discrimination, et les performances décroitront.

1.5 Évaluation des systèmes de mise à jour

De nombreuses mesures existent dans la littérature pour évaluer de façon statistique les performances des systèmes biométriques. Cependant, dans leur version de base, ces métriques ne prennent pas en compte l'aspect temporel (qui nous semble être un point relativement important du cas de la *mise à jour du modèle biométrique*). Il est donc nécessaire de les utiliser dans un cadre d'utilisation spécifique, en suivant un protocole propre à l'évaluation des systèmes de *mise à jour du modèle biométrique*. A notre avis, donner le Taux d'Erreur Égal d'un système de *mise à jour du modèle biométrique* n'a aucun sens. Nous n'avons pas d'information sur son évolution au cours du temps. De plus, nous ne savons pas si il est plus judicieux de tester la mise à jour de façon hors ligne ou en temps réel. Quelques travaux de la littérature ont essayé d'apporter des solutions à ces différents problèmes. Nous présentons donc les métriques qui doivent être utilisées pour l'évaluation.

1.5.1 Calcul des performances

1.5.1.1 Métriques utilisables

Métriques génériques Il est possible d'utiliser différentes métriques pour l'évaluation de la *mise à jour du modèle biométrique*. Les trois métriques principales sont

- Le *Taux de Fausse Reconnaissance (TFR)* (*False Match Rate (FMR)* dans la littérature anglophone) qui va consister à calculer le ratio d'*imposteurs* considérés comme étant des *utilisateurs légitimes* en utilisant des paramètres prédéfinis (seuil d'acceptation, seuil de mise à jour, ...).
- Le *Taux de Fausse Non Reconnaissance (TFNR)* (*False Non Match Rate (FNMR)* dans la littérature anglophone) qui va consister à calculer le ratio d'*utilisateurs légitimes* considérés comme étant des *imposteurs* en utilisant des paramètres prédéfinis (seuil d'acceptation, seuil de mise à jour, ...). Cette information va de paire avec le TFR.
- Le *Taux d'Erreur Egale (TEE)* (*Error Equal Rate (EER)* dans la littérature anglophone) qui est le taux d'erreur lorsque le système est configuré de telle façon à obtenir un TFR égal au TFNR. Il est donc nécessaire de faire varier un ou plusieurs paramètre pour l'obtenir, et, ne semble pas forcément un indicateur réaliste dans le cas de mise à jour.

Ryu *et al.* [67] présentent le *TEE* et le *TFR* lorsque le *TFNR* est nul, ainsi que le *TFNR* lorsque le *TFR* est nul. L'article ne précise pas comment les seuils sont calculés, mais cela suppose de disposer d'un jeu de données suffisamment grand (ce qui n'est pas forcément problématique) et de données d'imposteurs pour le cas du *oTFR* (ce qui n'est pas forcément applicable dans la dynamique de frappe au clavier). En règle générale, les autres études présentent les résultats avec le *EER* et le *1TFR*. Ces métriques donnent les performances de reconnaissance du système. Plus les valeurs sont faibles, meilleur est le système. Cependant, elles ne sont pas spécifiques au problème de *mise à jour du modèle biométrique*, ce sont les métriques habituellement utilisées dans n'importe étude biométrique. Elles ne sont pas suffisantes dans notre cas.

Métriques spécifiques à la mise à jour Une information pertinente pour la *mise à jour du modèle biométrique* est le ratio d'imposteurs ajoutés dans le *modèle biométrique* [43]. Cette information semble être rarement présentée dans la littérature. Nous nous attendons donc à avoir le moins d'imposteurs possible pour les meilleurs algorithmes de mise à jour. Nous allons nommer cette métrique *Taux d'Imposteurs Ajoutés aux Modèle (TIAM)*, elle se calcule de la façon suivante :

$$TIAM = \frac{\sum \mathbb{1}\{étiquette(galerie) \in imposteurs\}}{|galerie|} \tag{1.65}$$

avec $galerie$ l'ensemble des données collectées lors de la *mise à jour du modèle biométrique*. L'intérêt d'utiliser un système de *mise à jour du modèle biométrique* est d'augmenter les performances du système de reconnaissance (ou du moins, de faire en sorte qu'elles ne chutent pas au cours du temps). Il est donc nécessaire de présenter également le pourcentage d'amélioration du système utilisant la *mise à jour du modèle biométrique* comparé au système n'employant pas de tel mécanisme.

1.5.1.2 Fréquence de l'utilisation des métriques employées

Il nous semble intéressant de nous demander à quelle fréquence nous devons employer les métriques de calcul d'erreur. Nous avons vu que l'aspect temporel est relativement important dans la gestion de la *mise à jour du modèle biométrique*. Il nous semble également important de le prendre en compte dans la gestion du calcul des erreurs. La première façon de prendre ce point est compte est de toujours présenter les données à tester par ordre chronologique afin de suivre l'évolution temporelle de la *donnée biométrique*. La plupart des études présentant les données de façon aléatoire ne semblent pas avoir respecté ce principe. Il est possible que ça n'ait pas d'implication particulière sur l'évolution du modèle des modalités biométriques à faible variabilité (*cf.* empreintes digitales), mais nous pensons que dans le cas des modalités à forte variabilité temporelle (*cf.* dynamique de frappe au clavier), il est plus que nécessaire de garder ce point à l'esprit. Toujours pour cette raison, il nous semble que ne fournir qu'un seul taux d'erreur global ne soit pas la meilleure solution (ce qui est fait dans la quasi-totalité des études car elles ne comprennent pas forcément de données sur une longue période). Pourquoi ne pas calculer ces métriques à chaque authentification ? À chaque journée ? Ou à chaque session de capture de la base de donnée utilisée ? Pour le moment, dans la majorité des études, l'ensemble des données E est séparé en trois ensembles disjoints: $M \in E$, $A \in E$, $T \in E$, respectivement pour la création du modèle initial, l'utilisation de techniques de *mise à jour du modèle biométrique* et l'étude des performances ; avec $A \cap M = A \cap T = T \cap M = \emptyset$. Choisir la fréquence de calcul des métriques reste donc encore une question ouverte.

1.5.1.3 Calcul des performances en-ligne ou hors-ligne ?

Quelle que soit la métrique d'évaluation utilisée, il existe deux façons principales d'effectuer cette évaluation :

- *Hors ligne* : le *modèle biométrique* est mis à jour un certain nombre de fois à l'aide d'un ensemble de données de test. Au bout d'un certain temps, qui dépend de la fréquence de calcul définie plus haut, un autre jeu de test est utilisé afin d'être vérifié sur le *modèle biométrique* mise à jour. La métrique d'évaluation est ensuite calculée sur ces scores calculés hors ligne (sans faire de mise à jour du modèle). Poh *et al.* [50] nomment ce mécanisme : *separate adapt-and-test strategy*.

- *En ligne* : le *modèle biométrique* est évalué à la volée tout au long de sa mise à jour. Chaque donnée sert à la fois à l'évaluation (en produisant un score de comparaison entre la donnée *requête* et le *modèle biométrique* qui sera stocké avec les scores *intra-classe* ou *inter-classe*) et à la *mise à jour du modèle biométrique* (en étant intégré, le cas échéant dans le nouveau modèle). Au bout d'un certain temps, qui dépend de la fréquence de calcul définie plus haut, la métrique d'évaluation est calculée sur l'ensemble des scores précédemment produits. Poh *et al.* [50] nomment cette méthode : *joint adapt-and-test strategy*.

Sauf erreur de notre part, nous n'avons pas trouvé de papier de calcul en ligne dans la littérature (excepté dans [50]), alors qu'il semble plus réaliste (*i.e.*, dans un système réel, les données utilisées pendant la vérification sont celles qui servent pour la mise à jour) et permet de disposer de plus de données pour les calculs (*i.e.*, la même requête set à la fois pour la vérification, et, peut potentiellement être utilisée pour la *mise à jour du modèle biométrique*). Comme expliqué précédemment, toutes les évaluations sont faites hors ligne avec un sous-ensemble de test.

1.5.2 Respect d'un protocole

Il est nécessaire de respecter un protocole particulier afin de pouvoir comparer les études entre elles. L'idéal étant d'avoir un protocole commun entre les études [55]. Cependant, respecter un tel protocole n'est pas toujours possible ou suffisant. Dans tous les cas, il faut préciser toutes les informations nécessaires afin que l'étude puisse être reproductible facilement. Suffisamment d'informations doivent être données concernant la base de données utilisée pour l'étude. Les points les plus importants sont:

- Le nombre total d'utilisateurs concernés.
- Le nombre de captures fournies par chaque utilisateur.
- L'intervalle de temps pris pour la création de la base de données, voire même la date de chacune des captures.
- Le nombre de sessions, le cas échéant, constituant la base de données.
- Le partitionnement de la base de données :
  - la partition de données utilisées pour l'enrôlement ;
  - la partition de données utilisée pour la mise à jour (données non étiquetées) ;
  - le cas échéant, la partition de données utilisée pour la validation hors ligne (données de test).

D'après la Table 1 de Rattani *et al.* [53] (synthétisant 14 études), la quantité de données utilisée pour la *mise à jour du modèle biométrique* (27 en moyenne) est nettement plus faible que la quantité de données utilisée pour la validation (62 en moyenne). Il n'existe pas encore de consensus sur le ratio à garder (dans le cas d'une évaluation hors ligne), mais il parait évident que ce ratio influe sur les résultats.

Pour chaque évaluation des performances, il peut être utile de connaitre le nombre scores *intra-classe* et *inter-classe* impliqué. Ces informations permettent de donner une indication sur le ratio de données d'imposteurs utilisées dans le calcul des performances. Le protocole doit également présenter l'ordre de présentation des données de test aux modèles biométriques à faire évoluer. Il est notamment intéressant de savoir si:

- Toutes les données clientes sont présentées en premier.
- Toutes les données d'imposteurs sont présentées en premier.
- Les données sont présentées aléatoirement [43].

Ryu et Kim [66], Ryu *et al.* [67] présentent les résultats pour les trois types de présentation en indiquant que cela permet de mieux évaluer les performances des systèmes commerciaux. Ces papiers traitent de la reconnaissance d'empreintes digitales, et, les performances sont sensiblement identiques dans les trois cas. Nous ne nous attendons pas à obtenir le même résultat avec des modalités biométriques plus faibles. Les trois façons ont été utilisées dans la littérature. Il est également important de voir si les données sont également présentées de façon chronologique (*i.e.*, $date(presentation_t) < date(presentation_{t+i}), \quad \forall i$). Ce point semble ne jamais être quasiment pris en compte dans les études. À notre connaissance, ce point n'est respecté que dans [40] (et encore, leur base de données est constituée de séquences vidéos sur une courte période, avec donc une variabilité quasiment nulle sur la forme du visage ou sa texture, et ils ont présenté les données aléatoirement pour le test), ou l'espace propre est calculé récursivement tout au long du temps, en donnant moins de poids aux plus anciens exemples. C'est pourtant surprenant car le but de ces études est de capturer la variabilité au cours du temps, ce qui implique donc de respecter un

minimum de chronologie. Il est fort probable que ce non-respect de chronologie ne soit pas réellement problématique dans le cas des modalités morphologiques, mais, ce point n'est pas à négliger dans le cas des modalités comportementales, où l'utilisateur acquière un réflexe, au cours de l'utilisation du système, qui implique une évolution chronologique. Le ratio de données d'imposteur comparé aux données de test est également un point à prendre en compte. En effet, plus il y a d'imposteurs dans le jeu de données, plus la probabilité d'ajouter un imposteur dans le *modèle biométrique* du client est grande. Le problème est donc plus difficile dans les études ou la quantité de données d'imposteur est grande. Cette information n'est pas toujours présentée dans la littérature. Marcialis *et al.* [43] utilisent le même nombre de données d'imposteurs et de clients pour la mise à jour, et le même nombre de données d'imposteurs et de clients pour le test. Les derniers points importants à vérifier sont ceux ayant été présentés précédemment : les évaluations sont-elles faites en ligne ou hors-ligne ? À quelle fréquence ? Avec quelle mesure ?

1.5.3 Bases de données

Pour pouvoir travailler sur des problèmes de *mise à jour du modèle biométrique*, il est nécessaire de disposer d'une base contenant des données capturées sur une période relativement grande afin de capturer suffisamment de variabilité, et d'avoir suffisamment d'utilisateurs et de données afin que les résultats soient statistiquement valides. Nous allons présenter les bases publiques qui ont été utilisées dans les études de l'état de l'art.

1.5.3.1 Reconnaissance faciale 2D

Equinox Face Dataset La base Equinox est régulièrement utilisée dans la littérature [2], mais elle ne semble pas librement disponible. Freni [22] utilise 50 individus choisis aléatoirement avec 100 images. Rattani *et al.* [58] utilisent 40 individus avec 20 images frontales de leur visage avec d'importantes variations d'illumination et d'expression.

MORPH La base MORPH [59] a été utilisée dans plusieurs études. 14 utilisateurs ayant plus de 20 images (par individus) ne présentant pas de variations trop importantes pour la pose et l'expression sont utilisés dans Drygajlo *et al.* [19]. Li *et al.* [38] utilisent un sous ensemble de la base ayant peu de variabilité en terme d'illumination, de pose, et d'expression du visage. Cela permet de ne pas biaiser les résultats en ayant uniquement une variabilité temporelle. Ainsi, seulement 42 utilisateurs de la base avec au minimum 5 images par utilisateurs.

UMIST Face database La base UMIST contient 564 images de 20 utilisateurs. Plusieurs poses sont enregistrées pour chaque utilisateur. El Gayar *et al.* [20] utilisent 20 utilisateurs ayant entre 25 et 55 images.

AR La base AR [44] est constituée des visages 2d couleur de 126 individus (bien que la majorité des études travaillent avec des versions en noir et blanc). L'intérêt de la base est que les conditions d'illumination sont différentes d'une image à l'autre, que les visages ont plusieurs expressions, et qu'il y des occlusions. Il y a donc une forte variabilité. En contrepartie, le nombre d'images par individus et relativement faible, ainsi que la chronologie, car il n'y a eu que deux sessions.

Youtube Drygajlo *et al.* [19] ont utilisé une vidéo présente sur youtube qui contient les visages de quelques individus sur plusieurs années (trois ans ou plus), ce qui est nettement supérieur à la majorité des bases de visage. Cette base contient donc une grande variabilité sur le temps, mais elle ne contient pas suffisamment d'utilisateurs. Seulement quatre utilisateurs sont utilisés dans [19]. Les auteurs ne proposent pas le lien, nous ne pouvons donc pas utiliser cette base de données...

1.5.3.2 Reconnaissance faciale 3D

Face Recognition Grand Challenge (FRGC) Experiment 3 [47] Il s'agit d'une base de visages 3D associés à leur couleur. Différents points ont été annotés sur le coin de yeux, le bout du nez et le bout du menton. L'ensemble d'apprentissage consiste en 943 captures du visage et de leur couleur sur 270 individus et la base de test est constituée de 410 individus ayant fourni en 1 et 22 captures. Poh *et al.* [49] ont étudié l'évolution des scores (clients et imposteurs) au cours du temps sur une période de 250 jours sur un sous ensemble de 285 utilisateurs ayant utilisé le système plus de six fois.

1.5.3.3 Reconnaissance d'empreinte digitale

FVC2002 La base [3] est issue de la compétition « Fingerprint Verification Competition ». Quatre différentes bases de données sont disponibles, chacune d'entre elles contenant 110 doigts avec 8 captures par doigt (10 pour configurer les algorithmes, et 100 pour les tester). Les empreintes appartiennent à différents types de population (variation en genre, age, ...) et sont capturées avec différents capteurs (optique, capacitif, et généré artificiellement). Les conditions d'acquisition sont proches de l'utilisation réelle (pas de nettoyage des capteurs après utilisation, pas de période d'apprentissage pour les utilisateurs). Freni *et al.* [21] font remarquer que cette base n'est pas appropriée, même s'ils l'ont utilisé dans leur étude, du fait qu'elle ne contient que huit exemples par utilisateurs. Cependant, la variabilité intra-classe est relativement importante. Pour chaque sous-ensemble, ils utilisent 50 utilisateurs authentiques et 50 utilisateurs jouant le rôle d'imposteurs.

Il faut également noter que plusieurs bases privées ont été utilisées dans la littérature. Cependant, les spécificités sont relativement proches des bases publiques. Nous pouvons également noter l'existence d'autres bases de données potentiellement utilisable pour les modalités dynamiques de frappe au clavier et signature manuscrite.

1.5.4 Dynamique de frappe au clavier

GREYC keystroke Nous avons développé la base GREYC Keystroke [25] dans le cadre d'une étude sur la dynamique de frappe au clavier. Elle est tout de même intéressante pour les études de mise à jour du modèle biométrique car elle a été réalisée sur plusieurs sessions étalées sur plus de deux mois. Ainsi 100 utilisateurs ont participé à 5 sessions, espacées en général d'au moins une semaine, en effectuant 12 captures du mot de passe par session.

DSN2009 Killourhy et Maxion [35] proposent également une base de données de dynamique de frappe au clavier. Celle-ci est composée de 551 utilisateurs ayant participé à 8 sessions (sans préciser l'espacement de ces sessions, le délai minimum étant d'une journée) en fournissant 50 captures par session.

1.5.5 Signature manuscrite

MCYT-100 La base MCYT-100 [46] est une base de données multimodale empreinte digitale et signature manuscrite en ligne. Un jeu de 100 utilisateurs est disponible. Cette base a été utilisée pour vérifier la pertinence de données extraites au cours du temps [29].

1.5.5.1 Discussion

Comme nous pouvons le voir, il existe de nombreuses bases de données publiques utilisées dans la littérature. Cependant, la totalité de ces bases concerne la reconnaissance faciale et la reconnaissance d'empreintes digitales. Nous n'avons pas recensé d'études sur des bases de modalités comportementales, alors qu'il est connu que la variabilité *intra-classe* est plus importante sur de telles modalités. Ensuite, il est indéniable que les bases utilisées encodent une certaine variabilité *intra-classe*, mais celle-ci est plus souvent due aux conditions d'acquisition qu'au vieillissement de la donnée biométrique.

1.6 Conclusion

Rattani *et al.* [53] présentent également un tableau résumant les différentes études de *mise à jour du modèle biométrique* avant 2010. La TABLE 1.1 présente une version étendue de ce résumé. Nous avons ajouté des études supplémentaires, et, des colonnes qui nous semblent pertinentes pour la comparaison des différentes études.

Nous pouvons voir sur cette table que la majorité des études pertinentes concerne la reconnaissance faciale ou d'empreintes digitales, et, que la taille des bases utilisées est majoritairement faible.

Les méthodes d'apprentissage semi-supervisées (auto apprentissage et co apprentissage) ont montré leur intérêt et efficacité dans le problème de la *mise à jour du modèle biométrique*. Il pourrait être également intéressant de s'intéresser aux algorithmes d'apprentissage actif. De cette façon, un administrateur pourrait intervenir de façon ponctuelle, et déterminée par le système, afin que celui-ci l'aide pour l'étiquetage des données complexes.

La majeure partie des techniques présentées dans l'état de l'art fonctionnent avec des modalités ne nécessitant qu'une seule capture pour faire la vérification. Ce paradigme n'est pas vrai dans le cas de l'utilisation de la dynamique de frappe au clavier (comme de n'importe quelle modalité nécessitant d'avoir un modèle généré calculatoirement à partir de plusieurs captures) et peut empêcher l'utilisation de telles techniques. Nous pouvons supposer que ces études ont majoritairement analysé la variabilité *intra-classe* des utilisateurs due aux conditions d'acquisition, mais pas au temps, car les bases sont relativement petites.

Les études sur la *mise à jour du modèle biométrique* semblent être relativement récentes. Même si des articles sont proposés dès 2002 [31], la majeure partie des papiers sont postérieurs à 2006, et la majorité d'entre eux proviennent de l'équipe de Fabio Roli. Certaines études ont présenté des méthodes de *mise à jour du modèle biométrique* sans les expérimenter et les tester sur des jeux de données [34, 68]. Ces méthodes expérimentales peuvent donc ne pas être efficaces dans une expérimentation réelle.

Un certain nombre d'études porte sur l'utilisation d'un système multimodal, afin de capturer un maximum de variabilité (même si ça n'est pas indiqué clairement dans les différents papiers). Il est donc possible que l'utilisation d'un système multimodal soit le passage obligé pour améliorer les performances d'un système comportemental avec une mise à jour du *modèle biométrique*. Dans le cas des systèmes unimodaux, la majorité des études utilisent comme modalité la reconnaissance d'empreintes digitales ou la reconnaissance faciale. Toutes les modalités n'ont pas été explorées, alors que les conclusions à prendre, dans les différentes études, peuvent être différentes d'une modalité à l'autre.

Enfin, la majeure partie des études est faite sur des modalités non comportementales. Même si celles-ci sont sujettes à une variation intra-classe, il est indéniable que cette variation est largement inférieure à celle de modalités comportementales. Nous nous attendons donc, dans le cas des modalités comportementales, à obtenir des résultats moins intéressants. Rattani *et al.* [53] fait également remarquer que les études de l'état de l'art portent sur la reconnaissance d'empreintes digitales et de visage seulement. Il est fort probable que les méthodes à adopter aux systèmes comportementaux soient relativement différentes.

Le sujet de la *mise à jour du modèle biométrique* est suffisamment important pour avoir été inclus dans les spécifications de la BioAPI [4] (BioAPI_TEMPLATEUPDATE, BioAPI_ADAPTATION). Ainsi, la fonction BioAPI_CreateTemplate est capable de mettre à jour un modèle existant s'il lui est fourni en paramètre. Ce mécanisme est par contre optionnel. De plus, la norme fait la distinction entre le fait de mettre à jour le modèle (CreateTemplate, Enroll) et le fait de décider de mettre à jour le modèle (BioSPI_Verify, BioSPI_VerifyMatch).

Un point qu'il serait intéressant d'explorer est de savoir si la *mise à jour du modèle biométrique* est compatible avec l'utilisation de mécanismes de génération de données qui ont pour vocation de ne pas stocker les données dans leur format original ? Si ce n'est pas le cas, quelles sont les mesures à apporter pour résoudre le problème.

La mise à jour du modèle n'est pas non plus le seul point à prendre en compte afin d'améliorer les performances au cours du temps (ou du moins, de ne pas les faire décroitre). Prendre en compte l'évolution des scores au cours du temps est un point qui semble important [49], tout comme améliorer la qualité de l'enrôlement.

TABLE 1.1 – Résumé des plus importantes études.

Étude	Auto-apprentissage	Co-apprentissage	Supervisé	En-ligne	Batch	Modalité	Base	Taille	# apprentissage	# mise à jour	# évaluation	Gain
Singh *et al.* [71]			✓	✓		2D visage proche infrarouge	CBSR + EQUINOX + WVU	328x17,7	4	13,7	les mêmes	5,77%
Uludag *et al.* [75]			✓		✓	empreintes	personnelle	50x125	25	25	75	9,99% 31,39%
Freni *et al.* [21]	✓			✓		empreintes	FVC2002	100x8	1	6	1	53,00%
Jiang et Ser [31]			✓	✓		empreintes	FCV2000 Privée	100x8 12x200	2 1	6 199	les mêmes les mêmes	31,39% 50,00%
Ryu *et al.* [67]	✓			✓		empreintes	privée	41x?	1	?	=	32,81%
Liu *et al.* [40]	✓			✓		Visage	Privée	20x220 30x75 7x24	10 5 3	210 70 21	les mêmes les mêmes les memes	75,00% 80,00% 60,00%
Roli et Marcialis [62]	✓				✓	visage	AR	100x14	1	6	7	74,00%
Roli *et al.* [61]		✓			✓	Visage + empreintes	AR + FVC2002	100x8	1	7	les mêmes	46,00%
Rattani *et al.* [58]		✓			✓	visage empreintes	Equinox and DIEE,	42x20	1	9	10	59%
Rattani *et al.* [54]	✓				✓	visage	Equinox	57x129	1	50	78	40%-27%
Drygajlo *et al.* [19]	ne correspond pas					visage	Youtube MORPH	4x300 14x20	100 5	0 0	200 15	67,6% (au bout de 2.5 ans) 79,37%
Araujo *et al.* [6]	✓			✓		dynamique de frappe au clavier	Privée	30x15	10	5	les memes	62,30%

On peut voir que les méthodes sont plus ou moins complexes en fonction des études. Cependant, dans le cas de la dynamique de frappe au clavier, ce sont systématiquement des méthodes simplistes de gestion de galerie qui sont utilisées. Utiliser des méthodes plus compliquées implique probablement une meilleure augmentation des performances.

Voici les points qu'il serait intéressant d'explorer, dans le cas de modalités biométriques telles que la dynamique de frappe au clavier :

- analyser l'évolution des scores de reconnaissance au cours du temps, afin de prendre en compte cet aspect dans la décision (*cf. Stacked Generalization* [78])
- modifier des méthodes de reconnaissance existantes, de telle façon qu'elles soient capables d'intégrer les nouvelles données, tout en donnant progressivement moins de poids aux données plus anciennes, sans toutefois les oublier trop rapidement :
  - en s'inspirant des méthodes de mise à jour des espaces propres [40] dans le cas des méthodes de dynamique de frappe utilisant la moyenne et l'écart type (pour les problèmes à une classe) ;
  - en utilisant des mécanismes de pondération des exemples pour méthodes de classification à deux ou plusieurs classes (*cf.* notre algorithme basé sur les séparateurs à vastes marges pour les secrets partagés) ;
- intégrer les deux mondes : mise à jour du modèle biométrique au cours du temps, ainsi que de la frontière de décision ;
- analyser le comportement des techniques à base de graphes pour filtrer les outliers lors de la mise à jour.

Annexes

Annexe A

Flot Max/Coupe Min

Les illustrations sont inspirées de `http://www-igm.univ-mlv.fr/~desar/Cours/imac-algo/ch7.pdf`. Dans le problème du flot max/coupe min, nous travaillons avec un graphe orienté et valué. Le graphe est représenté par ses nœuds et ses arêtes : $G = (V, E)$ avec une valuation $c : E \to \mathbb{N}$. Le graphe contient deux nœuds spéciaux : la source s et le puits t tels que $deg_{entrant}(s) = 0$ et $deg_{sortant}(t) = 0$.

A.1 Les réseaux de transport

Le graphe est donc vu comme un réseau de transport quelconque. Chaque arête est capable de transporter une certaine quantité de données (qui dépend du problème concerné, mais l'unité de mesure n'est pas corrélée avec l'algorithme – dans le cas de la biométrie, il s'agit du score de comparaison de deux captures), et en transporte une certaine quantité (voir FIGURE A.1).

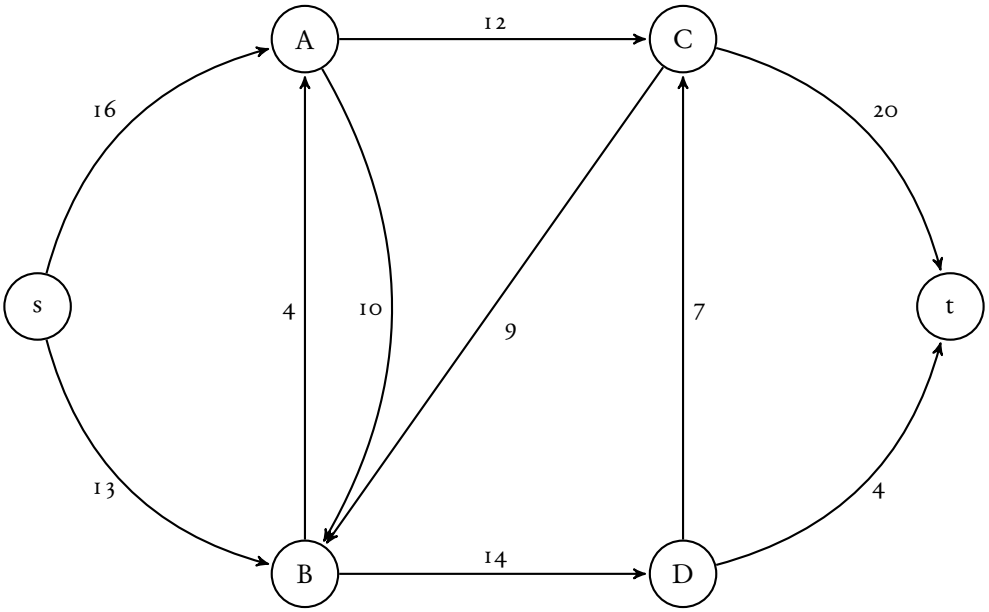


FIGURE A.1 – Exemple de réseau de transport.

Aucune arête n'arrive sur la source s, et aucune arête ne sort du puits t. Un *flot* (voir FIGURE A.2) est une application ϕ de E dans $\mathbb{N}$ telle que :

- le flot en chaque arête e est inférieur à sa capacité : $\phi(e) <= c(e)$. Il s'agit de la *contrainte de capacité* ;
- pour tout sommet de $G \setminus \{s, t\}$, la quantité de flots entrant dans ce sommet vaut la quantité de flots sortants (il n'y a aucune perte). Il s'agit de la *conservation de flot*.

La valeur $|\phi|$ du flot ϕ est le flot sortant de s (ainsi que le flot entrant de t). Un flot ϕ est dit saturé, si sur tout chemin de s à t, il existe une arête e tel que $\phi(e) = c(e)$.

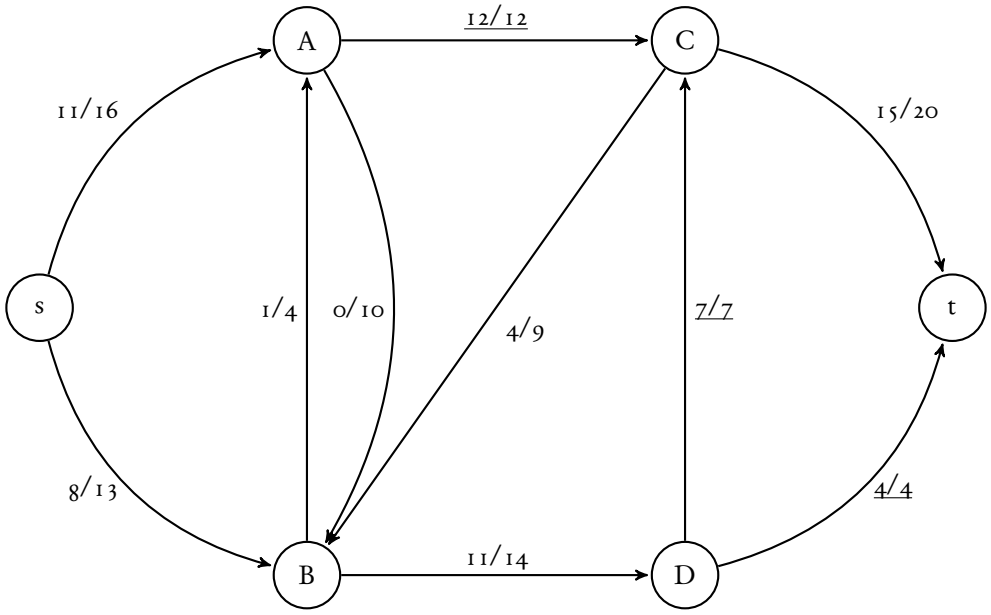


FIGURE A.2 – Un flot sur le réseau de transport présenté en FIGURE A.1. Celui-ci est saturé (*cf.* arêtes AC, DC et Dt)

Une *coupe* est une partition de l'ensemble des sommets de G en deux parties disjointes. L'une des parties, Y, contient la source s, tandis que l'autre, Z, contient le puits t. Les propriétés suivantes sont vérifiées :

$$\begin{cases} Y \cup Z = G \\ Y \cap Z = \emptyset \\ s \in Y \\ t \in Z \end{cases} \tag{A.1}$$

La somme des valeurs du flot sur les arêtes de Y vers Z moins la somme des valeurs du flot sur les arêtes de Z vers Y vaut aussi $|\phi|$. Cette différence est appelée flot net. La *capacité* d'une coupe est la somme des capacités des arêtes de Y à Z.

A.2 Flot maximum et coupe minimum

L'idée est de chercher le flot maximum en améliorant progressivement le chemin en déterminant le réseau résiduel (voir la FIGURE A.3). Pour chaque arête $e = ij$, $\phi(e) <= c(e)$. On peut donc augmenter le flot de $c(e) - \phi(e)$, et, le diminuer de $\phi(e)$, donc le faire passer sur l'arête $-e = ji$. Si l'arête n'existe pas, il faut la créer, sinon, on ajoute donc $\phi(e)$ à $c(-e)$.

Il faut ensuite chercher un meilleur chemin de s à t dans le réseau résiduel. Il correspond à une possibilité d'amélioration du flot, en modifiant la valeur du minimum des capacités résiduelles sur le chemin (voir la FIGURE A.4). FIGURE A.5 présente le flot amélioré.

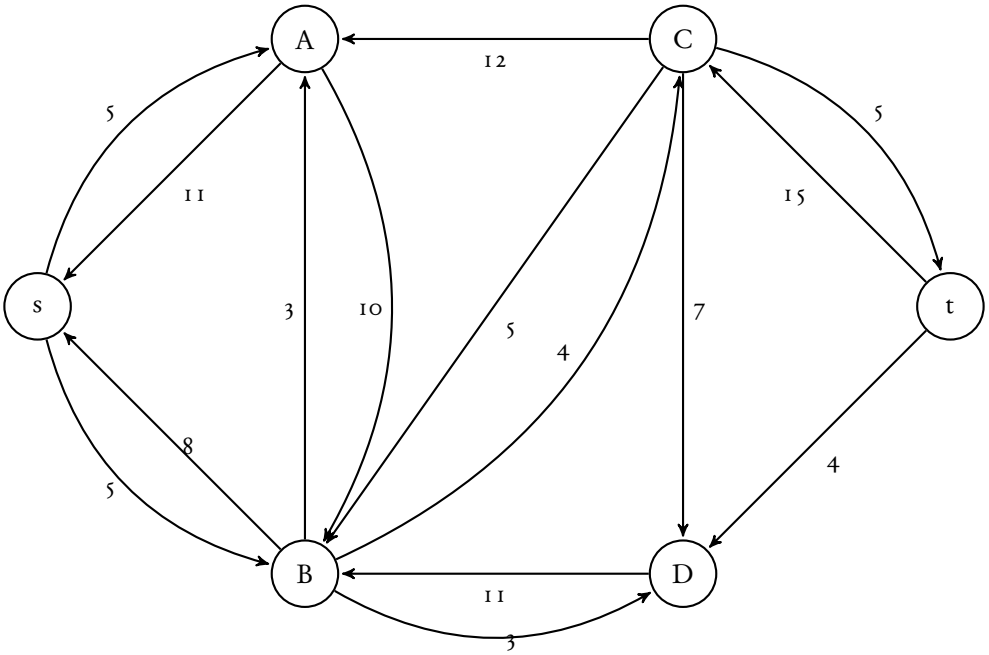


FIGURE A.3 – Le réseau résiduel sur le réseau de transport présenté en FIGURE A.2. Notez la présence de nouveaux arcs (les arcs $-e$) et la modification de la valeur pour les arcs existants (les arcs e).

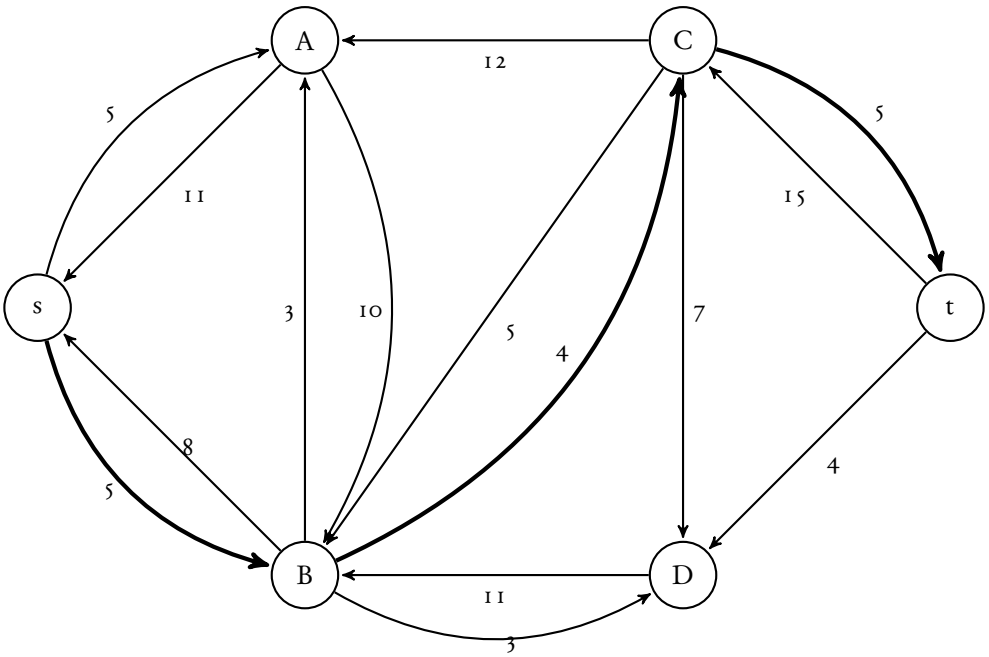


FIGURE A.4 – Un chemin améliorant (arêtes en gras) depuis le graphe de la FIGURE A.3.

L'algorithme est effectué itérativement jusqu'à ce qu'il n'y ai plus de chemin entre s et t, voir la FIGURE A.6 (donc pas de chemin améliorant).

Théorème. *Voici donc le théorème du flot maximum et de la coupe minimum : Si ϕ est un flot dans un réseau de transport, les trois conditions suivantes sont équivalentes :*

– *ϕ est un flot maximum ;*

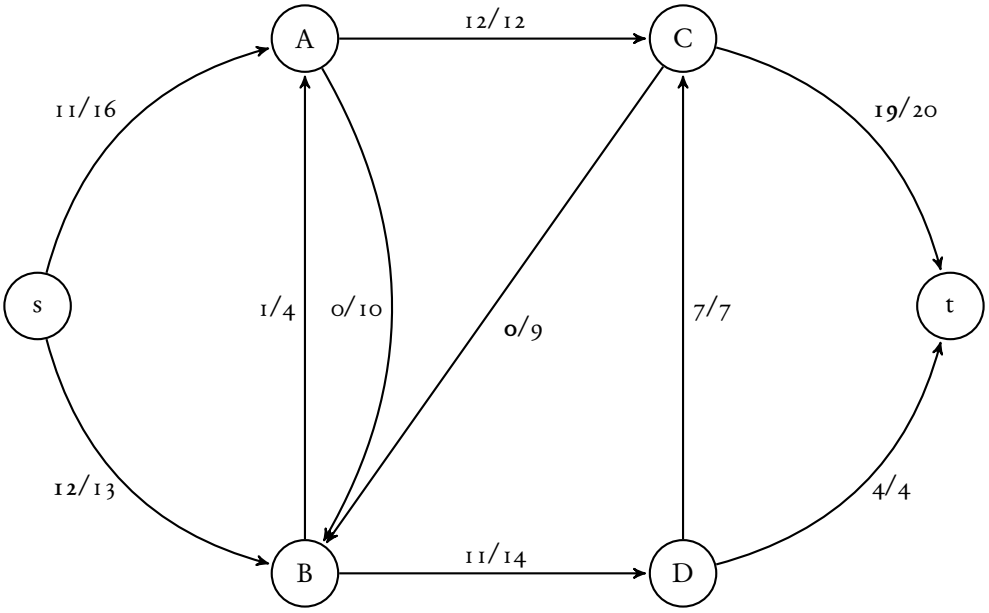


FIGURE A.5 – Flot après amélioration en utilisant le chemin améliorant de la FIGURE A.4.

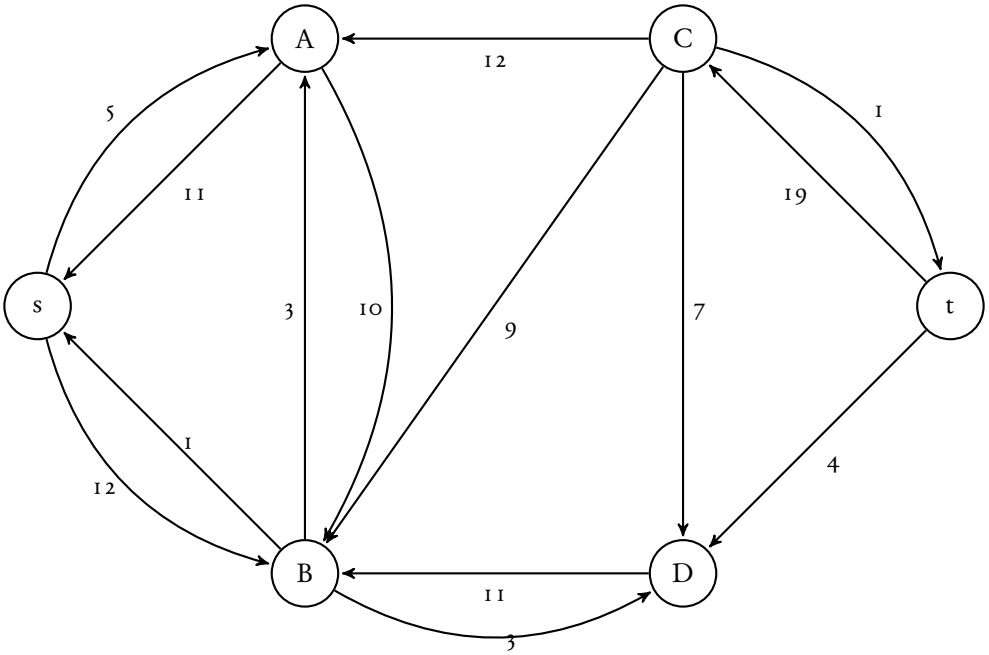


FIGURE A.6 – Le nouveau réseau résiduel sur le réseau de transport présenté en FIGURE A.4.

- *Le réseau résiduel de ϕ ne contient aucun chemin améliorant ;*
- *Il existe une coupe Y/Z dont la capacité vaut $|\phi|$.*

La condition 3 implique que ϕ est la valeur minimum des capacités des coupes du réseau, puisqu'on sait déjà que ϕ est inférieur à la capacité de n'importe quelle coupe.

Démonstration.
- Pour la propriété 2, si on trouve un chemin améliorant, on peut augmenter ϕ. L'ancien flot n'était donc pas maximum.

– Pour la propriété 3, si il n'y a pas de chemin améliorant, on pose Y la composante fortement connexe de s dans le graphe résiduel. Le complémentaire $Z = G \setminus Y$ contient t. Toutes les arêtes entre Y et Z dans le graphe résiduel vont de Y vers Z. Donc $|\phi|$ vaut la capacité de la coupe $Y|Z$.
– Pour la propriété 1, si un flot a comme valeur la capacité d'une coupe, il est nécessairement maximum, puisque tout les flots sont inférieurs à la capacité de n'importe quelle coupe.

□

Annexe B

Machines à vecteurs support

B.1 Introduction

Le but des machines à vecteurs support (ou Séparateurs à Vaste Marge (SVM)) et d'effectuer une classification binaire de la donnée requête. L'étiquette de l'exemple ($\{-1, 1\}$) dépend du signe de la fonction apprise. L'apprentissage consiste à trouver $y : \mathcal{X} \rightarrow \mathbb{R}$ à l'aide de l'ensemble d'apprentissage. La détermination de l'étiquette de l'exemple $\mathbf{x}$ est faite de la façon suivante : $l = sgn(y(\mathbf{x}))$. Cette fonction est faite de telle façon à effectuer une séparation linéaire des données d'apprentissage en maximisant la marge. Une séparation non linéaire peut être obtenue en transformant les données d'entrée dans un autre espace des attributs $\mathcal{F}$ ($\Phi : \mathcal{X} \rightarrow \mathcal{F}$). Une fonction noyau peut être utilisée à la place du produit cartésien de deux vecteurs permet d'effectuer ce changement de repère de façon transparente.

B.2 Séparation (non) linéaire

La séparation (linéaire si $\phi(x) = x$) de deux ensembles s'obtient grâce à la fonction suivante :

$$y(\mathbf{x}) = \mathbf{w}^T \cdot \phi(\mathbf{x}) + b \tag{B.1}$$

Avec cette notation, si $y(\mathbf{x}) > 0$, la classe $+1$ est assignée à $\mathbf{x}$, autrement, la classe -1 lui est assignée. L'opération $\phi(\mathbf{x})$ correspond à la transformation de l'espace de représentation des attributs, qui permet de projeter $\mathbf{x}$ dans un espace de plus grande dimension (pouvant même être infini, dans le cas d'un noyau gaussien).

Avec N exemples d'apprentissage, $\mathbf{x}_1, \ldots, \mathbf{x}_N$ étiquetés $t_1, \ldots, t_N$, l'apprentissage consiste donc à trouver $\mathbf{w}$ et b tels que :

$$\forall n, t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 \tag{B.2}$$

B.3 Maximisation de la marge

$\mathbf{w}$ et b sont choisis de telle façon que la distance entre la frontière de décision ($\mathbf{w}^T \phi(\mathbf{x}) + b = 0$) et les exemples les plus proches soit maximisée. Cette distance est calculée de la façon suivante :

$$d(\mathbf{x}) = \frac{|y(\mathbf{x})|}{||\mathbf{w}||} \tag{B.3}$$

La distance entre la frontière de décision et les exemples les plus proches est appelée la *marge*. Géométriquement, cette marge vaut $2/||\mathbf{w}||^2$, et le problème de maximisation de la marge peut être écrit de façon

équivalent comme un problème de minimisation :

$$argmin_{\mathbf{w},b}\frac{1}{2}||\mathbf{w}||^2 \tag{B.4}$$

sous la contrainte de l'equation (B.2). Pour que les calculs soient plus pratiques, en général, b est intégré à $\mathbf{w}$ et une dimension supplémentaire, toujours égale à 1, est ajoutée aux exemples.

B.4 Représentation duale

Étant donné qu'il s'agit d'un problème d'optimisation contraint, il est possible d'obtenir un problème dual en utilisant les multiplicateurs $a_n \geq 0$ de Lagrange [1] (il y a un multiplicateur par exemple d'apprentissage). Le problème dual est le suivant :

$$\begin{cases} \max \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \\ \forall i, a_n >= 0 \\ \sum_{n=1}^{N} a_n t_n = 0 \end{cases} \tag{B.5}$$

Il s'agit d'un problème de programmation quadratique [2] de dimension N (le nombre d'exemples). La fonction $k(\mathbf{x}_n, \mathbf{x}_m)$ est appelée la fonction noyau et est équivalente à $\phi(\mathbf{x}_n)^T\phi(\mathbf{x}_m)$. C'est grâce à cette fonction noyau qu'il est possible d'obtenir un séparateur non linéaire. Les multiplicateurs de Lagrange a_n sont obtenus en utilisant un résolveur de problèmes quadratiques.

$y(\mathbf{x})$ peut également être exprimée avec seulement les multiplicateurs de Lagrange :

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) \tag{B.6}$$

L'utilisation des multiplicateurs de Lagrange a transformé le problème d'une somme sur M dimensions (lors du produit vectoriel de (B.1) en une somme sur N points. La quantité N est souvent très largement supérieure à la quantité M, mais cette technique permet de bénéficier du *kernel trick* en ne calculant pas explicitement $\phi(\mathbf{x})$. De plus, les N points ne sont pas utilisés, car seulement quelques multiplicateurs de Lagrange a_n ne sont pas nuls. Il est donc utile de ne stocker que ceux-ci dans le modèle généré. Les exemples $\mathbf{x}_n$ correspondants aux $a_n > 0$ sont appelés *vecteurs supports*.

B.5 Marge souple (ou poreuse)

Les explications précédentes ne sont vraies que si le jeu de données d'apprentissage est séparable, ce qui n'est pas toujours vrai avec des jeux de données réels. Par conséquent, il est nécessaire de relaxer les contraintes pour avoir un mécanisme qui fonctionne également lorsque les données d'apprentissage ne sont pas séparables. La contrainte (B.2) est donc réécrite en utilisant des variables ressort ξ_n afin d'assouplir les contraintes :

$$\forall n, t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) \geq 1 - \xi_n \tag{B.7}$$

Si $\xi = 0$, l'exemple d'apprentissage correspondant est classifié correctement. Si $0 < \xi_n \leq 1$, l'exemple d'apprentissage est dans la marge, du bon côté de la fonction de décision. Si $\xi_n > 1$, l'exemple d'apprentissage est mal classé. L'equation (B.4) devient donc :

$$argmin_{\mathbf{w},b}\frac{1}{2}||\mathbf{w}||^2 + C\sum_{n=1}^{N}\xi_n \tag{B.8}$$

1. http://en.wikipedia.org/wiki/Lagrange_multipliers
2. http://en.wikipedia.org/wiki/Quadratic_programming

$C > 0$ est le paramètre qui contrôle le lien entre la pénalité des variables ressorts et la marge. Une fois de plus, nous pouvons introduire les multiplicateurs de Lagrange, dériver la fonction Lagrangienne en fonction de $\mathbf{w}$, b et ξ_n, et injecter la solution dans la fonction Lagrangienne. On obtient le système suivant à résoudre :

$$\begin{cases} \max \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \\ \forall i, C >= a_n >= 0 \\ \sum_{n=1}^{N} a_n t_n = 0 \end{cases} \tag{B.9}$$

Les variables ressorts ont disparu, et, la seule différence avec une marge dure (le cas précédent) est que les a_n ont une borne maximum valant C.

B.6 Fonctions noyaux

Plusieurs fonctions noyaux sont communément utilisées, voici les principales.

Polynôme

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d \tag{B.10}$$

Gaussien

$$k(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2) \tag{B.11}$$

avec $\gamma > 0$.

Tangente hyperbolique

$$k(\mathbf{x}_i, \mathbf{x}_j) = tanh(k\mathbf{x}_i \cdot \mathbf{x}_j + c) \tag{B.12}$$

Bibliographie

[1] The fg-net aging database:. URL http://www.fgnet.rsunit.com/. [cité p. 12]

[2] http://www.equinoxsensors.com/products/hid.html. [cité p. 45]

[3] Fvc2002, 2002. URL http://bias.csr.unibo.it/fvc2002/. [cité p. 46]

[4] Information - technology - biometric application programming interface - part 1: Bioapi specification, 2006. [cité p. 47]

[5] Vincent ALIMI, Rima BELGUECH et Christophe ROSENBERGER : Secure and Privacy Preserving Management of Biometric Templates. *In NISK Conference*, pages 1–12, 2010. [cité p. 9]

[6] L.C.F. ARAUJO, Jr. SUCUPIRA, L.H.R., M.G. LIZARRAGA, L.L. LING et J.B.T. YABU-UTI : User authentication through typing biometrics features. *IEEE Transactions on Signal Processing*, 53(2 Part 2):851–855, 2005. [cité p. 3, 48]

[7] M.F. BALCAN, A. BLUM, P.P. CHOI, J. LAFFERTY, B. PANTANO, M.R. RWEBANGIRA et X. ZHU : Person identification in webcam images: An application of semi-supervised learning. *In ICML 2005 Workshop on Learning with Partially Classified Training Data*, volume 2, page 6. Citeseer, 2005. [cité p. 26]

[8] A. BARGIELA et W. PEDRYCZ : *Granular computing: an introduction*. Springer, 2003. ISBN 1402072732. [cité p. 36]

[9] A. BLUM et S CHAWLA : Learning from labeled and unlabeled data using graph mincuts. *In MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 19–26, 2001. [cité p. 23, 24]

[10] John W. CARLS : *A FRAMEWORK FOR ANALYZING BIOMETRIC TEMPLATE AGING AND RENEWAL PREDICTION*. Thèse de doctorat, AIR FORCE INSTITUTE OF TECHNOLOGY, 2009. [cité p. 15, 16]

[11] John W. CARLS, Richard RAINES, Michael GRIMAILA et Steven ROGERS : Biometric enhancements: Template aging error score analysis. *In 8th IEEE conference series on Automatic Face and Gesture Recognition (FG2008)*, 2008. [cité p. 15]

[12] R. CHANDRAMOULI, D. BAILEY, N. GHADIALI, D. BRANSTAD et C.M. GUTIERREZ : NIST Special Publication 800-79-1 Guidelines for the Accreditation of Personal Identity Verification Card Issuers. 2008. [cité p. 8]

[13] M. CHARIKAR, C. CHEKURI, T. FEDER et R. MOTWANI : Incremental clustering and dynamic information retrieval. *In Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 626–635. ACM, 1997. ISBN 0897918886. [cité p. 27]

[14] Hong-Gunn CHEW, Cheng-Chew LIM et Robert E. BOGNER : An implementation of training dual-nu support vector machines. *In* Panos M. PARDALOS, Donald W. HEARN, Liqun QI, Koklay TEO et Xiaoqi YANG, éditeurs : *Optimization and Control with Applications*, volume 96 de *Applied Optimization*, pages 157–182. Springer US, 2005. ISBN 978-0-387-24255-2. URL http://dx.doi.org/10.1007/0-387-24255-4_7. [cité p. 35]

[15] Y. CHIEN et King-Sun FU : On the generalized karhunen-loève expansion (corresp.). *Information Theory, IEEE Transactions on*, 13(3):518 – 520, juillet 1967. ISSN 0018-9448. [cité p. 39]

[16] David COHN, Les ATLAS et Richard LADNER : Improving generalization with active learning. *Machine Learning*, 15:201–221, 1994. ISSN 0885-6125. URL `http://dx.doi.org/10.1023/A:1022673506211`. 10.1023/A:1022673506211. [cité p. 27]

[17] L. DIDACI, G. MARCIALIS et F ROLI : Modelling frr of biometric verification systems using the template co-update algorithm. *In Lecture Notes in Computer Sciences, ICB 2009*, 2009. [cité p. 22]

[18] Hiroshi DOZONO, Shinsuke ITOU et Masanori NAKAKUNI : Comparison of the adaptive authentication systems for behavior biometrics using the variations of self organizing maps. *INTERNATIONAL JOURNAL OF COMPUTERS AND COMMUNICATIONS*, 1(4):108–116, 2007. [cité p. 37]

[19] A. DRYGAJLO, W. LI et K. ZHU : Q-stack aging model for face verification. *In Proc. 17th European Signal Processing Conference (EUSIPCO 2009)*, 2009. [cité p. 3, 4, 12, 19, 45, 48]

[20] N. EL GAYAR, S.A. SHABAN et S. HAMDY : Face recognition with semi-supervised learning and multiple classifiers. *In Proceedings of the 5th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, pages 296–301. World Scientific and Engineering Academy and Society (WSEAS), 2006. ISBN 9608457564. [cité p. 21, 22, 38, 45]

[21] B. FRENI, G. MARCIALIS et F. ROLI : Replacement algorithms for fingerprint template update. *In International Conference on Image Analysis and Recognition (ICIAR 2008)*, pages 884–893. Springer, 2008. [cité p. 33, 46, 48]

[22] Biagio FRENI : *Template Editing and Replacement: novel methods for biometric Template Selection and Update*. Thèse de doctorat, University of Cagliari, 2010. [cité p. 31, 45]

[23] G. GATES : The reduced nearest neighbor rule (Corresp.). *Information Theory, IEEE Transactions on*, 18(3):431–433, 1971. ISSN 0018-9448. [cité p. 32]

[24] R. GIOT, B. HEMERY et C. ROSENBERGER : Low cost and usable multimodal biometric system based on keystroke dynamicsand 2d face recognition. *In IAPR International Conference on Pattern Recognition (ICPR)*, pages 1128–1131, Istanbul, Turkey, août 2010. IAPR. Acecptance rate: 54/100. [cité p. 3]

[25] romain GIOT, Mohamad EL-ABED et Christophe ROSENBERGER : Greyc keystroke: a benchmark for keystroke dynamics biometric systems. *In IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2009)*, pages 1–6, Washington, District of Columbia, USA, septembre 2009. IEEE Computer Society. Acceptance rate : 56/100. [cité p. 46]

[26] N.J. GRABHAM et N.M WHITE : Use of a novel keypad biometric for enhanced user identity verification. *In Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, pages 12–16, 2008. [cité p. 35]

[27] P. HART : The condensed nearest neighbor rule (corresp.). *Information Theory, IEEE Transactions on*, 14(3):515 – 516, mai 1968. ISSN 0018-9448. [cité p. 31]

[28] Sylvain HOCQUET : *Authentification biométrique adaptative. Application à la dynamique de frappe et à la signature manuscrite*. Thèse de doctorat, Université de Tours, 2007. [cité p. 4]

[29] N. HOUMANI, S. GARCIA-SALICETTI et B. DORIZZI : On assessing the robustness of pen coordinates, pen pressure and pen inclination to time variability with personal entropy. *In Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, 2009. [cité p. 4, 46]

[30] Anil K. JAIN, Karthik NANDAKUMAR et Abhishek NAGAR : Biometric template security. *EURASIP J. Adv. Signal Process*, 2008:1–17, 2008. ISSN 1110-8657. [cité p. 9]

[31] Xudong JIANG et Wee SER : Online fingerprint template improvement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1121–1126, 2002. [cité p. 28, 47, 48]

[32] Pilsung KANG et Sungzoon CHO : A hybrid novelty score and its use in keystroke dynamics-based user authentication. *Pattern Recognition*, 42(11):3115–3127, 2009. [cité p. 35]

[33] Pilsung KANG, Seong-seob HWANG et Sungzoon CHO : Continual retraining of keystroke dynamics based authenticator. *In* Seong-Whan LEE et Stan LI, éditeurs : *Proceedings of ICB 2007*, volume 4642 de *Lecture Notes in Computer Science*, pages 1203–1211. Springer Berlin / Heidelberg, 2007. URL `http://dx.doi.org/10.1007/978-3-540-74549-5_125`. [cité p. 33]

[34] HB KEKRE et VA BHARADI : Adaptive feature set updating algorithm for multimodal biometrics. *In Proceedings of the International Conference on Advances in Computing, Communication and Control*, pages 277–282. ACM, 2009. [cité p. 17, 33, 47]

[35] K.S. KILLOURHY et R.A. MAXION : Comparing anomaly-detection algorithms for keystroke dynamics. *In IEEE/IFIP International Conference on Dependable Systems & Networks, 2009. DSN'09*, pages 125–134, 2009. [cité p. 46]

[36] T. KOHONEN : *Self-Organizing Maps*. Numéro 30 *in* Information Sciences. Springer, Heidelberg., second édition, 1997. [cité p. 37]

[37] B. KVETON, M. PHILIPOSE, M. VALKO et L. HUANG : Online semi-supervised perception: Real-time learning without explicit feedback. *In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 15–21. IEEE, 2010. [cité p. 26, 27]

[38] W. LI, A. DRYGAJLO et H. QIU : Aging face verification in score-age space using single reference image template. *In Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–7. IEEE, 2010. [cité p. 12, 45]

[39] Y. LI, J. YIN, E. ZHU, C. HU et H. CHEN : Score based biometric template selection and update. *In Future Generation Communication and Networking, 2009. FGCN'08. Second International Conference on*, volume 3, pages 35–40. IEEE, 2008. [cité p. 30]

[40] X. LIU, T. CHEN et S.M. THORNTON : Eigenspace updating for non-stationary process and its application to face recognition. *Pattern Recognition*, 36(9):1945–1959, 2003. ISSN 0031-3203. [cité p. 9, 39, 44, 48, 49]

[41] D.G. LOWE : Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. ISSN 0920-5691. [cité p. 28]

[42] Alessandra LUMINI et Loris NANNI : A clustering method for automatic biometric template selection. *Pattern Recognition*, 39(3):495–497, 2006. ISSN 0031-3203. [cité p. 7]

[43] G. MARCIALIS, A. RATTANI et F. ROLI : Biometric template update: an experimental investigation on the relationship between update errors and performance degradation in face verification. *In SSPR&SPR*, pages 684–693. Springer, 2008. [cité p. 20, 43, 44, 45]

[44] A.M. MARTINEZ et R. BENAVENTE : The AR face database. Rapport technique, CVC Technical report, 1998. [cité p. 45]

[45] Ricardo García NOVAL et Francisco Perales LÓPEZ : Adaptative templates in biometric authentication. *In The 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2008*, 2008. [cité p. 14]

[46] J. ORTEGA-GARCIA, J. FIERREZ-AGUILAR, D. SIMON, J. GONZALEZ, M. FAUNDEZ-ZANUY, V. ESPINOSA, A. SATUE, I. HERNAEZ, J.J. IGARZA, C. VIVARACHO *et al.* : MCYT baseline corpus: a bimodal biometric database. *In Vision, Image and Signal Processing, IEE Proceedings-*, volume 150, pages 395–401. IET, 2004. [cité p. 46]

[47] P.J. PHILLIPS, P.J. FLYNN, T. SCRUGGS, K.W. BOWYER, J. CHANG, K. HOFFMAN, J. MARQUES, J. MIN et W. WOREK : Overview of the face recognition grand challenge. *In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 947–954. IEEE, 2005. ISBN 0769523722. [cité p. 46]

[48] N. POH, J. KITTLER, S. MARCEL, D. MATROUF et J.F. BONASTRE : Model and score adaptation for biometric systems: Coping with device interoperability and changing acquisition conditions. *In Proceedings of Internal Conference on Pattern Recogntion (ICPR 2010)*, 2010. [cité p. 4, 14]

[49] N. POH, J. KITTLER, R. SMITH et J. TENA : A method for estimating authentication performance over time, with applications to face biometrics. *In 12th Iberoamerican Congress on Pattern Recognition CIARP*, 2007. [cité p. 46, 49]

[50] Norman POH, R. WRONG, J. KITTLER et F. ROLI : Challenges and research directions for adaptive biometric recognition systems. *In Advances in Biometrics*, pages 753–764, 2009. [cité p. 3, 10, 18, 40, 41, 42, 43, 44]

[51] Z. POZGAJ et I. DURINEK : Smart card in biometric authentication. *In Proc. of. Information and Intelligent Systems*, 2007. [cité p. 8]

[52] N. RAMANATHAN et R. CHELLAPPA : Face verification across age progression. *Image Processing, IEEE Transactions on*, 15(11):3349–3361, 2006. ISSN 1057-7149. [cité p. 11]

[53] A. RATTANI, B. FRENI, G. MARCIALIS et F. ROLI : Template update methods in adaptive biometric systems: A critical review. *In Internal Conference on Biometrics 2009 (ICB 2009)*, 2009. [cité p. 4, 5, 44, 47]

[54] A. RATTANI, G.L. MARCIALIS et F. ROLI : Biometric template update using the graph mincut algorithm : A case study in face verification. *In Biometrics Symposium, 2008. BSYM '08*, pages 23 –28, 2008. [cité p. 23, 24, 48]

[55] Ajita RATTANI : *Adaptive Biometric System based on Template Update Procedures*. Thèse de doctorat, Dept. of Electrical and Electronic Engineering University of Cagliari, 2010. [cité p. 13, 18, 20, 22, 23, 24, 44]

[56] Ajita RATTANI, D. R. KISKU, Andrea LAGORIO et Massimo TISTARELLI : Facial template synthesis based on sift features. 2007. [cité p. 28, 29]

[57] Ajita RATTANI, Gian Luca MARCIALIS et Fabio ROLI : Boosting gallery representativeness by co-updating face and fingerprint verification systems. 5th Summer School for Advanced Studies on Biometrics for Secure Authentication, 2008. [cité p. 22, 42]

[58] Ajita RATTANI, Gian Luca MARCIALIS et Fabio ROLI : Capturing large intra-class variations of biometric data by template co-updating. *In Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, 2008. [cité p. 22, 45, 48]

[59] K. RICANEK et T. TESAFAYE : Morph: A longitudinal image database of normal adult age-progression. *In Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 341–345. IEEE, 2006. ISBN 0769525032. [cité p. 12, 45]

[60] G. RITTER, H. WOODRUFF, S. LOWRY et T. ISENHOUR : An algorithm for a selective nearest neighbor decision rule (Corresp.). *Information Theory, IEEE Transactions on*, 21(6):665–669, 1975. ISSN 0018-9448. [cité p. 31]

[61] F. ROLI, L. DIDACI et G. MARCIALIS : Template co-update in multimodal biometric systems. *In International Conference on Biometrics (ICB 2007)*, pages 1194–1202. Springer, 2007. [cité p. 22, 48]

[62] F. ROLI et G. MARCIALIS : Semi-supervised pca-based face recognition using self-training. *In Structural, Syntactic, and Statistical Pattern Recognition*, pages 560–568. Springer, 2006. [cité p. 20, 48]

[63] Fabio ROLI, Luca DIDACI et Gian Luca MARCIALIS : *Advances in Biometrics*, chapitre Adaptive Biometric Systems That Can Improve with Use, pages 447–471. SpringerLink, 2008. [cité p. 20, 22]

[64] Chritophe ROSENBERGER et Luc BRUN : Similarity-based matching for face authentication. *In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2009. [cité p. 8]

[65] C. RYU, Y. HAN et H. KIM : Super-template generation using successive bayesian estimation for fingerprint enrollment. *In Audio-and Video-based Biometric Person Authentication (AVBPA 2005)*, pages 710–719. Springer, 2005. [cité p. 28]

[66] C. RYU et H. KIM : Fingerprint verification testing scenarios for multi-impression enrollment and template adaptation. *In Proc. of Biometric Symposium*, pages 39–40, 2005. [cité p. 44]

[67] Choonwoo RYU, Hakil KIM et Anil K. JAIN : Template adaptation based fingerprint verification. *In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006. [cité p. 28, 43, 44, 48]

[68] T. SCHEIDAT, A. MAKRUSHIN et C. VIELHAUER : Automatic template update strategies for biometrics. Rapport technique, 2007. [cité p. 32, 33, 34, 47]

[69] M.E. SCHUCKERS : Statistical inference for template aging. *In Proceedings of SPIE*, volume 6202, page 62020M, 2006. [cité p. 10, 11]

[70] A. SETHURAM, E. PATTERSON, K. RICANEK et A RAWLS : Improvements and performance evaluation concerning synthetic age progression and face recognition affected by adult aging. *In ICB 2009*, LNCS Advances in Biometrics, pages 62–71. Springer, 2009. [cité p. 12]

[71] R. SINGH, M. VATSA, A. ROSS et A. NOORE : Biometric classifier update using online learning: A case study in near infrared face verification. *Image and Vision Computing*, 28(7):1098–1105, 2010. ISSN 0262-8856. [cité p. 35, 36, 48]

[72] R. SUKTHANKAR et R. STOCKTON : Argus: the digital doorman. *Intelligent Systems, IEEE*, 16 (2):14–19, 2005. ISSN 1541-1672. [cité p. 19]

[73] Q. TAO, G.W. WU, F.Y. WANG et J. WANG : Posterior probability support vector machines for unbalanced data. *Neural Networks, IEEE Transactions on*, 16(6):1561–1573, 2005. ISSN 1045-9227. [cité p. 36]

[74] G. TUR, D. HAKKANI-TüR et R.E. SCHAPIRE : Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186, 2005. ISSN 0167-6393. [cité p. 27]

[75] U. ULUDAG, A. ROSS et A. JAIN : Biometric template selection and update: a case study in fingerprints. *Pattern Recognition*, 37(7):1533–1542, 2004. [cité p. 8, 29, 30, 33, 35, 48]

[76] Kumari VANDANA : Enhancing weak biometric authentication by adaptation and improved user-discrimination. Mémoire de D.E.A., International Institute of Information Technology Hyderabad, INDIA, 2007. [cité p. 16, 38]

[77] Dennis L. WILSON : Asymptotic properties of nearest neighbor rules using edited data. *Systems, Man and Cybernetics, IEEE Transactions on*, 2(3):408 –421, 1972. ISSN 0018-9472. [cité p. 32]

[78] D.H. WOLPERT : Stacked generalization. *Neural networks*, 5(2):241–259, 1992. ISSN 0893-6080. [cité p. 12, 49]

[79] Xiaojin ZHU : *Semi-Supervised Learning with Graphs*. Thèse de doctorat, Language Technologies Institue School of Computer Science Carnegie Mellon University, 2005. [cité p. 24]

[80] Xiaojin ZHU, Zoubin GHAHRAMANI et John LAFFERTY : Semi-supervised learning using gaussian fields and harmonic functions. *In Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, pages 1–9, Washington DC, 2003. [cité p. 24]