Un classifieur hybride incrémental pour la reconnaissance de caractères non contraints

A new hybrid adaptive classifier for unconstrained character recognition

L. Prevost L. Oudot A. Moises C. Michel-Sendis M. Milgram

Université Pierre & Marie Curie
Laboratoire des Instruments & Systèmes d'Ile de France
Groupe Perception, Automatique & Réseaux Connexionnistes
4 Pace Jussieu, Paris Cedex 75252, Case 164
lionel.prevost@lis.jussieu.fr

Résumé

La reconnaissance de l'écriture pour les produits "nomades" (assistant personnel électronique, téléphone mobile ...) nécessite des classifieurs robustes et incrémentaux. Il s'agit d'un problème d'une telle complexité qu'il est courant de faire coopérer plusieurs algorithmes de classification. Dans cette communication, nous présentons un classifieur hybride original. Le premier expert est un classifieur par modélisation qui met en compétition des modèles de caractères. Le second expert est un ensemble de classifieurs discriminants qui séparent les paires de classes les plus ambiguës. Cette architecture hybride repose sur le postulat suivant : la « bonne » classe appartient la plupart du temps aux deux classes les plus pertinentes trouvées par le premier classifieur. L'expérience, conduite sur une base de 80000 exemples, montre une amélioration de 30% des performances, pour un problème à 62 classes. De plus, nous montrons expérimentalement que cette architecture est parfaitement adaptée à la classification incrémentale.

Mots Clef

Reconnaissance d'écriture, classifieur par modélisation, classifieur incrémental, réseau de neurones discriminant.

Abstract

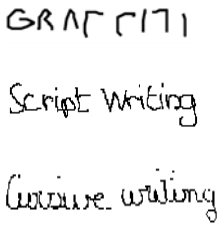
Handwriting recognition for hand-held devices like PDAs requires very accurate and adaptive classifiers. It is such a complex classification problem that it is quite usual now to make co-operate several classification methods. In this paper, we present an original two stages recognizer. The first stage is a model-based classifier which store an exhaustive set of character models. The second stage is a discriminative classifier which separate the most ambiguous pairs of classes. This hybrid architecture is based on the idea that the correct class almost systematically belongs to the two more relevant classes found by the first classifier. Experiments on a 80,000 examples database show a 30% improvement on a 62 classes recognition problem. Moreover, we show experimentally that such an architecture suits perfectly for incremental classification.

Keywords

Handwriting recognition, model-based classifier, adaptive classifier, discriminating neural network.

1 Introduction

L'informatique "nomade" (assistant personnel électronique, téléphone mobile, livre électronique ...) connaît actuellement un développement spectaculaire. Contrairement aux ordinateurs « classiques », ces produits sont de petite taille et dépourvus de clavier comme de souris. Le stylo électronique apparaît comme l'outil de saisie et de pointage idéal ; passant par le développement de systèmes de reconnaissance automatique de l'écriture et d'interfaces homme-machine conviviales. Dans cette communication, nous nous concentrerons sur le premier aspect. Pour une telle application, les performances en reconnaissance doivent être particulièrement élevées au risque de décourager les utilisateurs potentiels. Le principal problème étant la forte variabilité du signal écrit. Il est possible de contourner ce problème en contraignant l'utilisateur à utiliser un style d'écriture unique (l'alphabet *grafiti* des assistants personnels : figure 1.a) ; ou de le résoudre en tentant d'apprendre de façon exhaustive tous les styles d'écriture (naturelle ou scripte : figure 1.b et 1.c) pour développer un système de reconnaissance omni-scripteur. Ce dernier peut être ensuite adapté à un utilisateur donné en apprenant son style d'écriture voire son « alphabet personnel » (abréviations, symboles mathématiques ou chimiques pour les scientifiques …), de façon non-supervisée si possible. Le système devient alors mono-scripteur.



(a) écriture contrainte

(b) écriture scripte

(c) écriture cursive

Figure 1 : Styles d'écriture.

En reconnaissance d'écriture dynamique, l'information est constituée de séquences de coordonnées *(x,y)* correspondant au mouvement du stylo lorsqu'il est appuyé sur la tablette [7]. Chaque style d'écriture est caractérisé par des graphèmes ou particuliers incluant, d'une part, des caractères ayant la même image (représentation statique) mais des dynamiques de tracé différentes (nombre de traits composant le caractère et sens de ces traits) ; et d'autre part les différents modèles d'écriture existant pour chaque caractère (script, cursif, mixte …).
L'analyse des erreurs de classification révèle que deux phénomènes sont majoritairement à l'origine de celles-ci :

- Les formes sont très différentes des données de référence. Chaque utilisateur possédant sa propre manière d'écrire, de nombreuses variantes peuvent apparaître (figure 1.a). Ce problème peut être surmonté en classant les représentations dynamique (tracé) et statique (image) des caractères et en fusionnant les résultats de classification [9].
- Les formes peuvent être ambiguës (figure 1.b) et certains couples de classes sont à l'origine de la majorité des erreurs comme (B,D) ou (1,7).

(a) dynamiques inconnues

(b) caractères ambiguës

Figure 2 : Caractères ambiguës.

Fort de cette constatation, nous proposons une architecture hybride originale. Un premier classifieur détermine le couple de solutions (classes) le plus pertinent avec une probabilité élevée. Un second classifieur, en série avec le premier, lève l'indétermination. Nous montrons qu'un classifieur par modélisation s'avère particulièrement robuste lorsque sa tâche se limite à déterminer les deux classes les plus pertinentes. Le classifieur discriminant qui suit n'a qu'à résoudre un problème bi-classe (beaucoup plus simple que le problème à N classes initial !).

Dans la section 2, nous analysons les avantages et inconvénients de différents types de classifieurs. Les sections 3 et 4 sont consacrées respectivement au classifieur initial et au classifieur hybride. Nous montrons dans la section 5 que ce classifieur est incrémental. Finalement, les conclusions et perspectives sont présentées dans la section 6.

2 Génération de modèles ou discrimination ?

En classification, on oppose deux méthodes (figure 3) : celles générant des modèles de chaque classe (classifieurs par modélisation) et celles déterminant des frontière entre les classes (classifieurs discriminants).

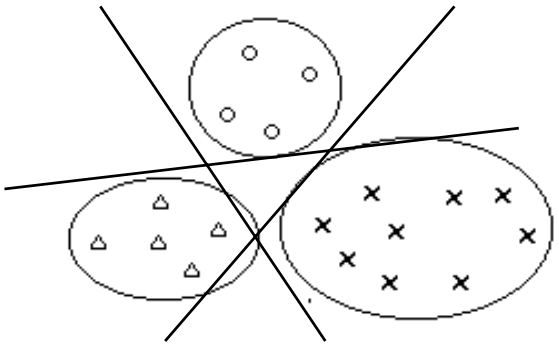


Figure 3 : modélisation et discrimination.

Les classifieurs par modélisation génèrent - durant la phase d'estimation - un ou plusieurs modèle(s) pour chaque classe. Ces modèles peuvent être neuronaux [13], markoviens [3] ou « semi-physiques » (base de prototypes représentatifs [1,14]). En phase d'utilisation, la classification est réalisée en mesurant l'adéquation entre la forme inconnue et les modèles.

Les classifieurs discriminants déterminent - durant la phase d'apprentissage – des frontières entre les classes. La plupart sont des classifieurs neuronaux. Dans [8], pour un problème à N classes, un unique réseau réalise la discrimination. Dans [5], N réseaux sont entraînés, chacun discriminant une classe des (N-1) autres. Dans [11], N(N+1)/2 réseaux de neurones discriminants sont entraînés pour séparer chaque couple de classes. Une telle solution semble prometteuse car elle réduit un problème de classification à N classes à des problèmes de classification binaires. On peut toutefois douter de son efficacité quand N augmente : pour reconnaître des majuscules, par exemple, il faudrait entraîner quelques 325 RND.
La comparaison des deux approches rend pertinente leur hybridation. Un classifieur par modélisation détermine le couple de classes le plus pertinent. Un classifieur discriminant augmente alors localement (autour des frontières entre les paires de classes) la robustesse du classifieur par modélisation.

Notons que des classifieurs hybrides du même type ont récemment été proposés pour la reconnaissance de l'écrit. Dans [2] : un perceptron multicouche discriminant détermine les deux classes les plus pertinentes. Il est suivi par un ensemble de SVM discriminants bi-classe activés par rejet d'ambiguïté. Dans [12], un classifieur par modèles est suivi d'un discriminant un contre N.

3 Le classifieur par modélisation

3.1 Classifieur à base de prototypes

Le classifieur mesure la similarité entre le caractère inconnu (séquence des coordonnées (x,y) composant le caractère) et un ensemble exhaustif de modèles de caractères stockés sous forme de prototypes. La base de prototypes de traits (B_{proto}) a été constituée en utilisant l'algorithme de clustering proposé dans [10]. Cet algorithme se décompose en deux phases : une phase d'*agglomération* des références autour des prototypes suivant un indice de proximité fourni par l'analyse de la matrice des distances inter-références et une phase d'*adaptation* qui optimise la position des prototypes et améliore ainsi la modélisation.

L'algorithme du plus-proche-voisin est utilisé pour la classification. La distance entre le caractère inconnu (décomposé en traits) et chaque prototype est calculée par programmation dynamique. La sortie du classifieur est le vecteur $\mathbf{D}=(D_1,D_2, \ldots, D_N)$ des distances du caractère inconnu à chaque classe.. En postulant que ces distances sont distribuées normalement, nous pouvons calculer un vecteur de probabilités a posteriori $\mathbf{p}=(p_1,p_2 ,\ldots, p_N)$ et déterminer la classe la plus pertinente (de probabilité maximale : C_1=argmax1(**p**)) et la seconde classe (C_2=argmax2(**p**)).

L'avantage de la modélisation sur la discrimination apparaît clairement (figure 4) : au lieu d'une « boîte noire », on dispose de modèles explicites pour chaque classe. L'adjonction de nouvelles classes se fait donc aisément, en estimant leur(s) modèle(s) et sans passer par un ré-apprentissage complet du système.

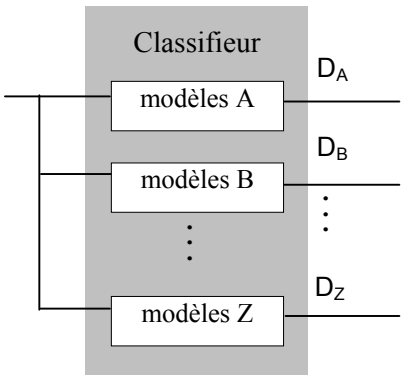


Figure 4 : Classifieur à base de prototypes.

3.2 Base de données, codages et résultats expérimentaux

Les expérimentations ont été conduites sur la base Unipen Train R01-V07 [4] qui a été divisée en trois corpus : d'estimation B_{ES} (utilisée pour le clustering), de validation croisée B_{VC} et de test B_{TE} (tableau 1). Les pré-traitements son réduits : la séquence des coordonnées (x,y) formant le caractère est ré-échantillonnée à raison de 20 points par traits, centrée et normalisée en conservant le ratio hauteur/largeur.

	Base de données			
	B_{ES}	B_{TE}	B_{VC}	B_{proto}
Chiffres	8000	4000	2000	476
Majuscules	12826	6355	3188	1158
Minuscules	23922	11443	5974	1114

Tableau 1 : Bases de données.

Les résultats de classification sur la base de test B_{TE} (tableau 2 : Top1 et Top2) montrent les bonnes performances du classifieur et la justesse de notre démarche : trouver une méthode permettant de « lever » les confusions apparaissant entre les deux classes les plus pertinentes (C_1 et C_2) devrait augmenter significativement la robustesse du système.

	Taux de reconnaissance	
	Top 1	**Top 2**
Chiffres	98.9 %	**99.8 %**
Majuscules	96.7 %	**99.0 %**
Minuscules	96.3 %	**98.8 %**

Tableau 2 : Taux de reconnaissance du classifieur à base de prototypes (base de test : B_{TE})

4 Le classifieur hybride

4.1 Apprentissage des réseaux de neurones discriminants

Le classifieur par modélisation génère un vecteur de probabilités. Les deux probabilités les plus élevées (Top1 et Top2) correspondent aux deux classes les plus pertinentes C_1 et C_2. Un réseau de neurones discriminant (RND) peut être entraîné à discriminer ces deux classes et à trouver la plus pertinente.

On suppose d'une part que le comportement du classifieur est décrit par sa matrice de confusion ; d'autre part que son comportement a priori (sur la base d'estimation) est représentatif de son comportement a

posteriori (sur la base de test). Dès lors, les couples de classes confusives (i,j) peuvent être trouvés en analysant la matrice de confusion. Cette dernière (figure 5) est déterminée sur l'ensemble d'estimation B_{ES}.
Comme nous nous intéressons aux confusions, la diagonale (qui indique les classifications correctes) est mise à zéro et le nombre $N_{i/j}$ de confusions pour chaque paires de classes est déterminé en sommant le nombre de confusions pour chaque paire (i,j) et (j,i). Trois conclusions s'imposent :

- De nombreux couples ne produisent aucune confusions : (4,2), (5,2) ... il est donc inutile d'entraîner un RND pour ces couples.
- Certains couples produisent quelques confusions, il peut être utile d'entraîner un RND.
- Quelques couples sont à l'origine de la majorité des confusions : (1,2), (1,7) …Ces couples doivent impérativement être discriminés.

On peut donc choisir de ne traiter que les couples dont la probabilité de confusion dépasse un certain seuil et régler ainsi la robustesse du classifieur hybride et le nombre de RND crées.

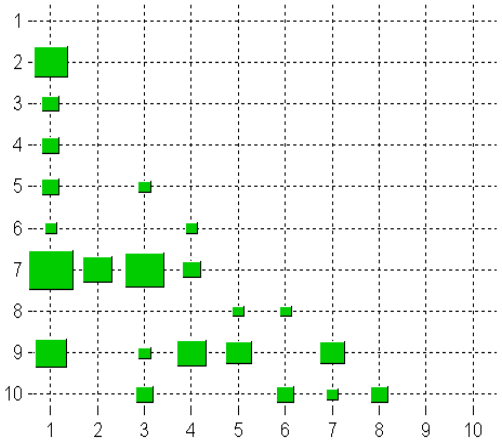


Figure 5 : Matrice de confusion (chiffres : B_{ES}).

Soit M_{conf} la matrice de confusion (déterminée sur l'ensemble d'estimation B_{ES}) et $N_{i/j}$ le nombre de confusions pour chaque paire (i,j), nous pouvons calculer le nombre total de confusions N_{tot} et la probabilité $p_{i,j}=p(C_1=i,C_2=j)$ de confusion pour chaque paire (i,j) :

$$N_{tot} = \sum_i \sum_j N_{i/j} \qquad p(i,j) = \frac{N_{i/j}}{N_{tot}}$$

Nous pouvons décider de prendre en compte les confusions les plus fréquentes dont la probabilité de confusions dépasse un certain seuil :

$$p(i,j) > \frac{p_{max}}{\delta_{conf}} \text{ avec } p_{max} = \max\{p(i,j)\}$$

Le tableau 3 précise l'ensemble des paires ambiguës détectées sur les chiffres pour un seuil de confusion δ_{conf} donné.

Cet ensemble est noté $S_{\delta conf} = \left\{ (i,j) \middle| p(i,j) > \frac{p_{max}}{\delta_{conf}} \right\}$

Pour chaque paire ambiguë, un RND est entraîné. Il s'agit dès lors d'un problème de classification binaire : la classe i est associée au label +1 et la classe j au label –1.

S_2	S_4	S_6	S_8	S_{10}
1,2	1,2	1,2	1,2	1,2
1,7	1,7	1,3	1,3	1,3
1,9	1,9	1,4	1,4	1,4
3,7	2,7	1,5	1,5	1,5
	3,7	1,7	1,7	1,7
	4,9	1,9	1,9	1,9
	5,9	2,7	2,7	2,7
	7,9	3,7	3,7	3,7
		3,0	3,0	3,0
		4,7	4,7	4,7
		4,9	4,9	4,9
		5,9	5,9	5,9
		6,0	6,0	6,0
		7,9	7,9	7,9
		8,0	8,0	8,0

Tableau 3 : Ensemble des paires de chiffres ambiguës en fonction du seuil de confusion .

Les expériences (tableau 4), conduites pour $\delta_{conf} \in \{2,4,6,8,10\}$, montrent logiquement l'augmentation du nombre de RND crées lorsque la probabilité de prise en compte de la confusion croît. Pour $\delta_{conf} = 10$, seuls 112 RND ont été entraînés. Si nous avions tenté de discriminer toutes les paires de classes en utilisant uniquement l'étage discriminant, 695 RND auraient été nécessaires (en considérant séparément les chiffres, les majuscules et les minuscules).

δ_{conf}	2	4	6	8	10
Chiffres	4	8	15	15	15
Majuscules	4	14	18	24	48
Minuscules	9	18	27	39	49
Total	**17**	**40**	**50**	**78**	**112**

Tableau 4 : Nombre de réseaux discriminants crées en fonction du seuil de confusion.

4.2 Utilisation des réseaux discriminants

Pour une forme inconnue de l'ensemble de test B_{TE} , le classifieur par modélisation fournit le couple de classes le plus pertinent.

- Si le couple (C_1,C_2) n'appartient pas à l'ensemble des classes couramment confondues $S_{\delta conf}$, on conserve la réponse du classifieur par modélisation (C_1).
- Sinon, une mesure de l'ambiguïté de la réponse du classifieur par modélisation permet d'activer le RND.

Soient p_{C1} et p_{C2} les probabilité a posteriori des deux classes les plus pertinentes, on considère qu'il y a ambiguïté si :

$\Delta p = p_{C1} - p_{C2} < \delta_{amb}$

Le RND correspondant à la confusion (C_1,C_2) est alors activé. Les deux situations extrêmes sont donc :

- $\delta_{amb} = 0$: l'étage discriminant est inhibé et le classifieur par modélisation fonctionne seul.
- $\delta_{amb} = 1$: l'étage discriminant est constamment actif quelque soit l'ambiguïté entre les deux classes.

On constate logiquement (figure 6) l'augmentation du taux d'activation des RND quand augmente le seuil d'ambiguïté.

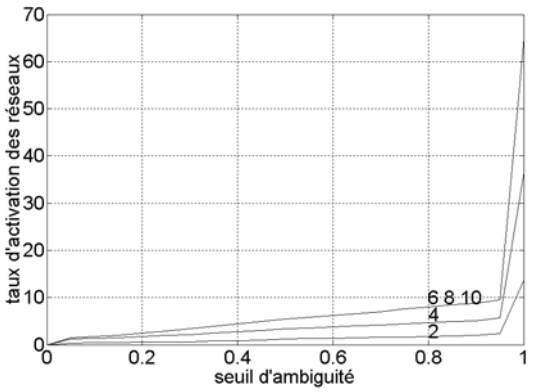


Figure 6 : Taux d'activation des RND en fonction des seuils de confusion et d'ambiguïté (chiffres : base de test B_{TE}).

4.3 Résultats expérimentaux

Les bases d'apprentissage des classes i et j n'ont généralement pas la même taille. Cette dernière est directement proportionnelle à la fréquence d'apparition des lettres de classes i et j dans le langage écrit. De plus, le nombre d'exemples confondus est généralement très faible. La frontière entre les deux classes n'est pas bien définie et le RND correspondant généralise mal. Pour surmonter cet inconvénient, les exemples confondus de la base d'apprentissage ont été détectés et leur nombre augmenté artificiellement par transformation. Chaque exemple confondu génère ainsi un ensemble de 30 nouveaux exemples par rotation (de quelques degrés autour du centre de gravité), dilatation et contraction (suivant l'une ou l'autre des coordonnées).

L'apprentissage du RND sur la base d'apprentissage modifiée B'_{TE} permet d'améliorer les performances du réseau en phase de test (tableau 5).

Les caractères confondus par le classifieur par modélisation ont toujours le même nombre de trait(s). Afin de disposer de vecteurs d'entrée de taille constante, les caractères sont ré-échantillonnés à 20 points. L'architecture des RND est la suivante : 40 cellules d'entrée correspondant aux 20 coordonnées (x,y), 10 cellules cachées et une cellule de sortie. Les RND sont entraînés sur la base d'estimation B_{ES} et l'apprentissage est stoppé par validation croisée sur B_{VC}. Plusieurs apprentissages ont été réalisés pour optimiser le nombre cellules de la couche cachée.

	Base initiale	**Base modifiée**
Chiffres	99.7 %	**99.6 %**
Majuscules	99.1 %	**99.3 %**
Minuscules	98.8 %	**99.5 %**

Tableau 5 : Taux de reconnaissance moyen des RND (base de test B_{TE}).

Le classifieur hybride a été évalué sur la base de test B_{TE} . Les figures 7, 8 et 9 précisent l'évolution du taux de reconnaissance en fonction du seuil d'ambiguïté et du seuil de confusion sur les chiffres, les majuscules et les minuscules. On peut constater que plus le seuil de confusion est élevé, plus le taux de reconnaissance augmente (sauf pour les chiffres, mais le nombre d'exemples mis en jeu dans ce cas est trop faible pour conclure). Le taux de reconnaissance croît aussi avec le seuil d'ambiguïté ; il est toujours supérieur au taux de reconnaissance du classifieur par modélisation. Les performances du classifieur hybride (pour δ_{conf}=10 et δ_{amb}=1) sont bien meilleures que celles du classifieur par modélisation initial (tableau 6).

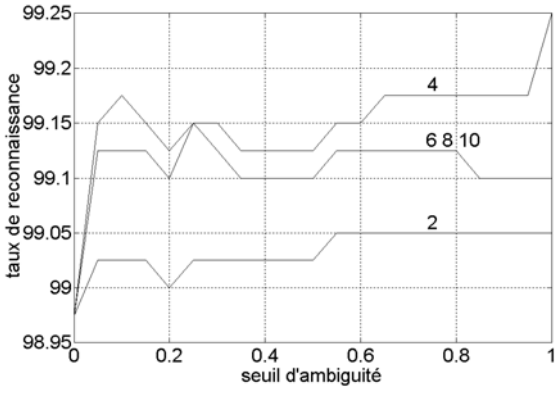


Figure 7 : Taux de reconnaissance du classifieur hybride en fonction des seuils de confusion et d'ambiguïté (chiffres : base de test B_{TE}).

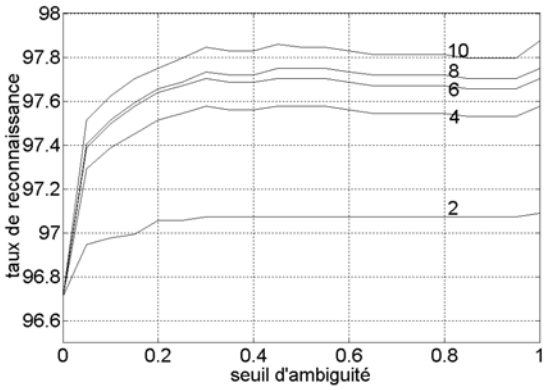


Figure 8 : Taux de reconnaissance du classifieur hybride en fonction des seuils de confusion et d'ambiguïté (majuscules : base de test B_{TE}).

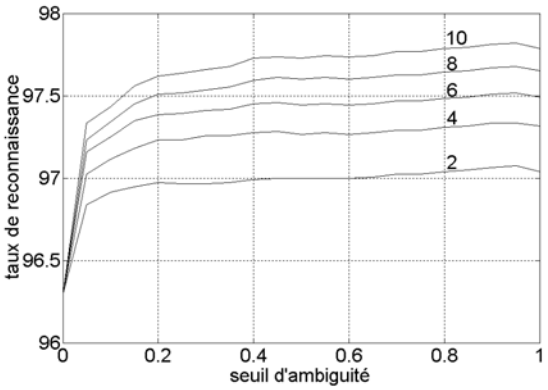


Figure 9 : Taux de reconnaissance du classifieur hybride en fonction des seuils de confusion et d'ambiguïté (minuscules : base de test B_{TE}).

	Classifieur par modélisation	**Classifieur hybride**
Chiffres	98.9 %	**99.1 %**
Majuscules	96.7 %	**97.9 %**
Minuscules	96.3 %	**97.8 %**

Tableau 6 : Taux de reconnaissance du classifieur hybride (δ_{conf}=10, δ_{amb}=1, base de test B_{TE}).

5 Classifieur incrémental

Dans cette section, nous montrons que le classifieur hybride précédemment décrit est parfaitement adapté à l'apprentissage incrémental c'est à dire que l'adjonction de nouvelles classes peut se faire de façon aisée. Cette étude préliminaire est juste une simulation : nous considérons que le classifieur initial ne reconnaît que les 26 majuscules et tentons de lui adjoindre 10 nouvelles classe correspondant aux chiffres.

5.1. Adaptation du classifieur à base de prototypes

De par sa structure modulaire, le classifieur à base de prototypes peut aisément être adapté afin de traiter de nouvelles classes. Contrairement aux classifieurs discriminants qui nécessitent un ré-apprentissage complet, les classifieurs par modélisation sont incrémentaux par nature ; le modèle de chaque classe étant estimé sur des données appartenant à cette seule classe. De plus, une fois estimés, les nouveaux modèles sont directement inclus dans le classifieur et la décision est prise en considérant l'ensemble des modèles, initiaux et nouveaux (figure 10).

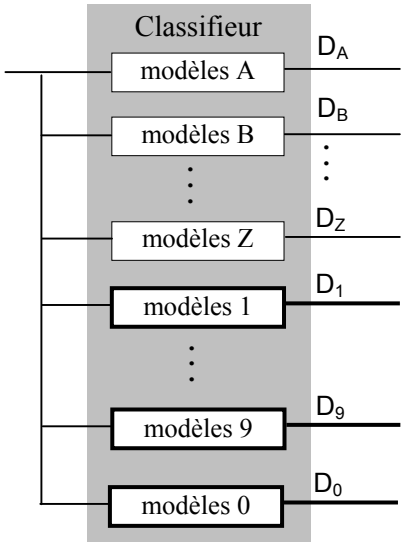


Figure 10 : Classifieur adapté.

5.2. Adaptation du classifieur hybride

L'adaptation est faîte comme suit :

(1) la matrice de confusion étendue est déterminée sur l'ensemble d'estimation en considérant toutes les classes (ici les 26 majuscules initiales et les 10 « nouveaux » chiffres soient 36 classes : figure 11) ;

(2) les confusions croisées entre classes initiales et nouvelles classes sont détectées. La procédure de détection est identique à celle présentée au §4.1 et utilisant le seuil de confusion optimal δ_{conf}=10 ;

(3) les RND correspondant aux confusions croisées sont entraînés (§4.1) et activés (§4.2).

Le tableau 7 précise les taux de reconnaissance du classifieur hybride dans les cas suivants :

- Classifieur par modélisation seul : le taux de reconnaissance chute drastiquement en raison des nombreuses confusions croisées entre lettres majuscules et chiffres comme (O,0), (I,1), (S,5) et (Z,2) par exemple.

- Classifieur hybride initial : les confusions entre majuscules (M,M) d'une part et chiffres (C,C) d'autre part sont traitées par l'étage discriminant mais les confusions croisées sont ignorées.
- Classifieur hybride adapté : toutes les confusions sont prises en compte, confusions croisées (M,C) inclues. Le résultat est très concluant : le taux d'erreur a été divisé par deux (comparé à celui du classifieur par modélisation).

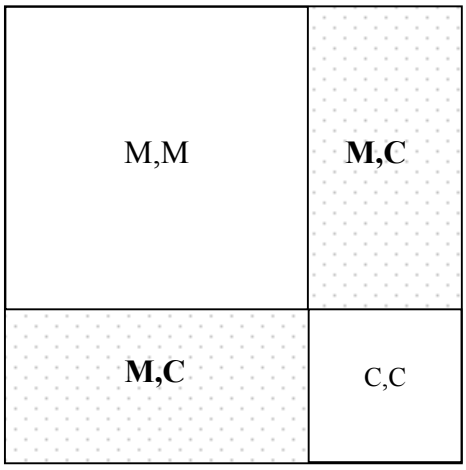


Figure 11 : Matrice de confusion étendue et confusions croisées.

	Classifieur par modélisation	**Classifieur hybride**	
		M,M C,C	**M,M C,C M,C**
Taux de reco	90.3 %	**91.5 %**	**95.4 %**
RND	-	**63**	**79**

Tableau 7 : Classifieur hybride incrémental : Taux de reconnaissance (base de test B_{TE}).

6 Conclusions

Nous avons montré au cours de cette étude l'intérêt d'un couplage modélisation-discrimination. Une première compétition permet de déterminer un ensemble de classes pertinentes (deux par exemple). A ce niveau, l'indétermination ne peut être levée par simple mise en compétition entre les modèles mais nécessite de la discrimination. En effet, l'ambiguïté restante se concentre dans des caractéristiques locales, discriminantes (penser à la différence entre les majuscules B et D). Cette discrimination est apportée à l'aide d'un ensemble de perceptrons multicouches, entraînés à discriminer les paires de classes les plus ambiguës.

De plus, l'architecture proposée est parfaitement adaptée à la classification incrémental : contrairement aux classifieurs discriminants, la combinaison modélisation-discrimination peut supportée l'adjonction de nouvelles classes sans ré-apprentisssage. Elle nécessite seulement l'estimation des nouveaux modèles, la détection des confusions croisées et l'apprentissage des réseaux discriminants correspondants.
Nous avons récemment développé un système complet de reconnaissance de textes [6] et le classifieur présenté dans cette communication pourrait aisément être intégré à ce système afin d'en optimiser la robustesse. L'évaluation de cette architecture sur un problème comportant un très grand nombre de classes - la reconnaissance de visages – est en cours.
Néanmoins, une question demeure : actuellement, l'étape de discrimination (apprentissage des réseaux de neurones) se construit autour d'une base de données de grande taille (système résident) ; quelles seront les performances des apprentissage lorsque le nombre d'exemples sera réduit (dans le cas typique d'un système embarqué) ?

7 Références

[1] Anquetil E. & Lorette G., On-line Handwriting Recognition system Based on Hierarchical Qualitative Fuzzy Modeling, *IWFHR'96, pp 47-52, 1996.*

[2] Bellili A., Gilloux M. & Gallinari P., An hybrid MLP-SVM handwritten digit recognizer, *ICDAR*'01, http://icdar.djvuzone.org/jss/index.html

[3] Connell S. & A.K. Jain, Writer adaptation of on-line handwriting models, *ICDAR'99, pp 434-437, 1999.*

[4] Guyon I., Schomaker L., Plamondon R., Liberman M. & Janet S., UNIPEN project of on-line data exchange and recognizer benchmarks, *ICPR'94, pp. 29-33, 1994.*

[5] Oh I.S. & Suen C.Y., A class-modular feed-forward neural network for handwriting recognition, *Pattern Recognition, 35, pp 229-244, 2002.*

[6] Oudot L. , Prevost L. & Milgram M., Système de lecture multi-contextuel pour la reconnaissance de textes manuscrits dynamiques, *CFD'2002, pp 335-344, 2002.*

[7] Plamondon R & Srihari S.N., On-line and off-line recognition : a comprehensive survey, *IEEE PAMI, 22(1), 2000.*

[8] Poisson E., Viard-Gaudin C. & Lallican P.M., Combinaison de réseaux de neurones à convolution pour la reconnaissance de caractères manuscrits en-ligne, *CFD'02, pp 315-324, 2002.*

[9] Prevost L. & Milgram M., Static and dynamic classifier fusion for character recognition, *ICDAR'97, (2), pp499-506, 1997.*

[10] Prevost L. & Milgram M., Modelizing character allographs in omni-scriptor frame : a new non-supervised algorithm, *Pattern Recognition Letters, 21(4), pp 295-302, 2000.*

[11] Price D., Knerr S., Personnaz L. & Dreyfus G., Pairwise neural network classifiers with probabilistic outputs, *NIPS, 7, 1994.*

[12] Ragot N. & Anquetil E., Combinaison hiérarchique de systèmes d'inférence floue, *CFD'02, pp 305-314, 2002.*

[13] Schwenk H. & Milgram M., Constraint tangent distance for on-line character recognition, *ICPR'96, (D), pp 520-524, 1996.*

[14] Vuori V., Laaksonen J. & Kangas,J., Influence of erroneous learning samples on adaptation in on-line handwriting recognition*, Pattern Recognition, 35(4), pp 915-925, 2002.*