

Coopération modélisation – discrimination pour la reconnaissance d’écriture manuscrite

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

Université Pierre & Marie Curie LISIF/PARC
4 Place Jussieu, Paris Cedex 75252, Case 164
lionel.prevost@lis.jussieu.fr

Résumé – Reconnaître l’écriture manuscrite est un problème d’une telle complexité qu’il est devenu courant de faire coopérer plusieurs algorithmes de classification. Dans cet article, nous présentons un classifieur hybride original. Un premier expert de modélisation détermine les deux classes les plus pertinentes en comparant le symbole inconnu à un ensemble exhaustif de symboles. Le second, discriminant, permet de lever les ambiguïtés. Cette architecture hybride exploite le fait que la “bonne” classe appartient le plus souvent aux deux classes les plus pertinentes trouvées par le premier classifieur. Les expérimentations, conduites sur une base de test de 20000 formes (62 classes), montrent que l’apport relatif de la coopération s’élève à 30%.

Abstract – Handwriting recognition is such a complex classification problem that it is quite usual now to make co-operate several methods at the pre-processing stage or at the classification stage. In this paper, we present an original two stages recognizer. The first stage is a model-based classifier that stores an exhaustive set of character models. The second stage is a discriminative classifier that separates 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 Unipen database show a 30% improvement on a 62 classes recognition problem.

1. Introduction

Nous étudions la reconnaissance de l’écriture dynamique depuis 1995. Nous avons récemment développé un système complet de reconnaissance de textes manuscrits [5] qui pourrait être implémenté sur un assistant personnel électronique ou tout autre produit « nomade ». Pour une telle application, les performances en reconnaissance doivent être particulièrement élevées au risque de décourager les utilisateurs potentiels. Cet objectif peut être atteint en rendant plus robuste l’expert de classification.

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 [7].
- 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).

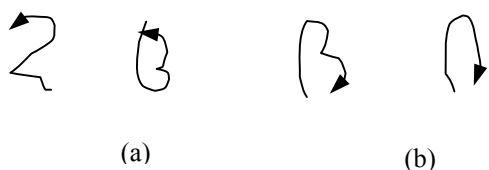


FIG. 1 : caractères ambiguës.

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. Les conclusions et perspectives sont présentées dans la section 5.

2. Génération de modèles ou discrimination ?

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

Les classifieurs par modélisation génèrent - durant la phase d’estimation - un ou plusieurs modèle(s) pour chaque classe. Ces modèles peuvent être neuronaux [10], markoviens [2] ou « semi-physiques » (liste de prototypes représentatifs [1,7]). 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 [6], pour un problème à N classes, un unique réseau réalise la discrimination. Dans [4], N réseaux sont entraînés, chacun discriminant une classe des $(N-1)$ autres. Dans [9], $N(N+1)/2$ réseaux de neurones discriminants (les RND) 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.

3. Le classifieur par modélisation

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 [8]. 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 $D=(D_1, D_2, \dots, D_N)$ 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 $p=(p_1, p_2, \dots, p_N)$ et déterminer la classe la plus pertinente (de probabilité maximale : $C_1 = \text{argmax}_1(p)$) et la seconde classe ($C_2 = \text{argmax}_2(p)$).

L'avantage de la modélisation « semi-physique » sur les modélisations neuronales ou markoviennes apparaît clairement dès que l'on considère sa capacité d'adaptation : un simple caractère suffit pour « apprendre » un nouveau style d'écriture, complètement différent de ceux déjà connus.

TAB 1 : Bases de données.

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

Les expérimentations ont été conduites sur la base Unipen Train R01-V07 [3] qui a été divisée en trois bases : d'estimation B_{ES} (utilisée pour le clustering), de validation

croisée B_{CV} et de test B_{TE} . Les pré-traitements sont 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.

Les résultats de classification sur la base de test B_{TE} (tableau 3 : 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.

4. Le classifieur hybride

4.1 Apprentissage des réseaux 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.

Les couples de classes confusives (i,j) peuvent être trouvés en analysant la matrice de confusion. Cette dernière (figure 2) est déterminée sur l'ensemble d'estimation B_{ES} en additionnant le nombre des confusions pour chaque paire (i,j) et (j,i) . Trois conclusions s'imposent :

- de nombreux couples ne produisent aucune confusion : $(4,2)$, $(5,2)$... 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.

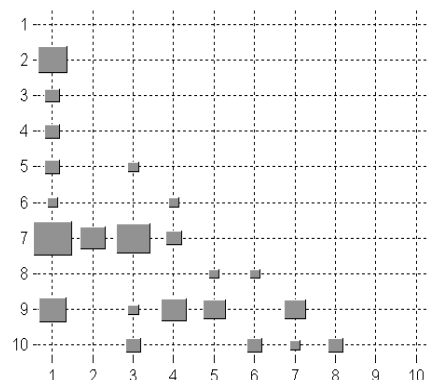


FIG. 2 : Matrice de confusion (chiffres : S_{ES}).

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

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 maximum de confusions N_{max} et la probabilité $p_{i,j} = p(C_1=i, C_2=j)$ de confusion pour chaque paire (i,j) :

$$N_{\text{max}} = \sum_{i,j} N_{i,j} \quad p_{i,j} = \frac{N_{i,j}}{N_{\text{max}}}$$

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

$$p_{i,j} > \frac{p_{\text{max}}}{\delta_{\text{conf}}} \quad \text{où } p_{\text{max}} = \max_{i,j}(p_{i,j})$$

Les expérimentations (tableau 2), conduites pour $\delta_{\text{conf}} \in \{2,4,6,8,10\}$, montrent logiquement l'augmentation du nombre de RND entraînés lorsque la probabilité de prise en compte de la confusion croît. Considérant toujours les majuscules, seuls 48 RND sont entraînés au lieu des 325 nécessaires pour lever toutes les confusions possibles.

TAB 2 : Nombre de réseaux discriminants créés en fonction du seuil de confusion.

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

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 n'appartient pas à l'ensemble des classes couramment confondues, 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_{C_1} et p_{C_2} les probabilités a posteriori des deux classes les plus pertinentes, on considère qu'il y a ambiguïté si $\Delta p = p_{C_1} - p_{C_2} < \delta_{\text{amb}}$, le RND correspondant à la confusion (C_1, C_2) est alors activé. On constate logiquement (figure 3) l'augmentation du taux d'activation des RND quand augmente le seuil d'ambiguïté

4.3 Résultats expérimentaux

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 de cellules de la couche cachée.

Le classifieur hybride a été évalué sur la base de test B_{TE} . La figure 4 précise l'évolution du taux de reconnaissance en fonction du seuil d'ambiguïté et du seuil de confusion. On peut constater que plus le seuil de confusion est élevé, plus les RND sont sollicités et plus le taux de reconnaissance augmente. Ce dernier croît aussi avec le seuil d'ambiguïté. Les performances du classifieur hybride (pour $\delta_{\text{conf}}=10$ et $\delta_{\text{amb}}=1$) sont bien meilleures que celles du classifieur par modélisation initial (tableau 3).

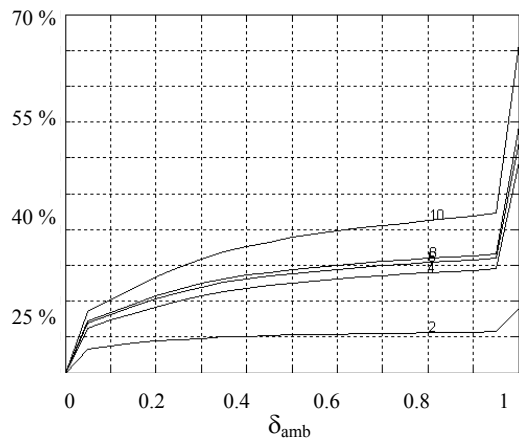


FIG. 3 : Taux d'activation du classifieur hybride en fonction des seuils δ_{conf} et δ_{amb} (majuscules B_{TE}).

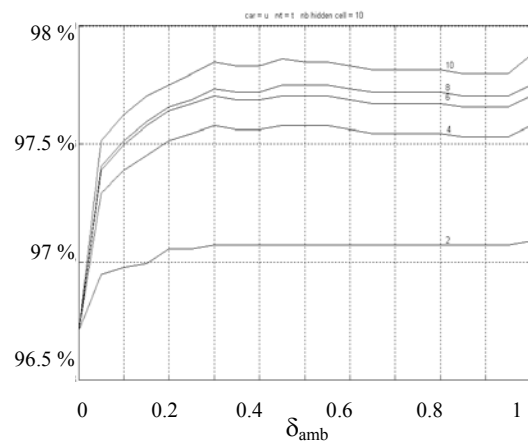


FIG. 4 : Taux de reconnaissance du classifieur hybride en fonction des seuils δ_{conf} et δ_{amb} (majuscules B_{TE}).

5. 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). L'indétermination ne peut être levée par simple compétition des 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). Il serait intéressant de montrer l'efficacité d'une telle architecture pour des problèmes comportant un très grand nombre de classes comme la reconnaissance de visages.

TAB 3 : Performances des deux classifieurs.

	Taux de reconnaissance		
	Top1	Top2	Classifieur hybride
Chiffres	98.8 %	99.8 %	99.3 %
Majuscules	96.6 %	99.0 %	97.8 %
Minuscules	96.3 %	98.8 %	97.8 %

6. Références

- [1] Anquetil E. & Lorette G., On-line Handwriting Recognition system Based on Hierarchical Qualitative Fuzzy Modeling, *IWFHR'96*, pp 47-52, 1996
- [2] Connell S. & A.K. Jain, Writer adaptation of on-line handwriting models, *ICDAR'99*, pp 434-437, 1999.
- [3] 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.
- [4] Oh I.S. & Suen C.Y., A class-modular feed-forward neural network for handwriting recognition, *Pattern Recognition*, 35, pp 229-244, 2002.
- [5] Oudot L. , Prevost L. & Milgram M., Système de lecture multi-contextuel pour la reconnaissance de textes manuscrits dynamiques, *CFD'2002*, pp 335-344, 2002.
- [6] 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.
- [7] Prevost L. & Milgram M., Static and dynamic classifier fusion for character recognition, *ICDAR'97*, (2), pp499-506, 1997.
- [8] Prevost L. & Milgram M., Modeling character allographs in omni-scriptor frame : a new non-supervised algorithm, *Pattern Recognition Letters*, 21(4), pp 295-302, 2000.
- [9] Price D., Knerr S., Personnaz L. & Dreyfus G., Pairwise neural network classifiers with probabilistic outputs, *NIPS*, 7, 1994.
- [10] Schwenk H. & Milgram M., Constraint tangent distance for on-line character recognition, *ICPR'96*, (D), pp 520-524, 1996.