

Simulations numériques sur  
un classifieur récursif  
Rapport de recherche: RR-03-2003

Jean Duong

Laboratoire de Reconnaissance des Formes et Vision (RFV)  
Institut National des Sciences Appliquées (INSA) de Lyon  
20, avenue Albert Einstein, 69621 Villeurbanne Cedex  
France

[duong@rfv.insa-lyon.fr](mailto:duong@rfv.insa-lyon.fr)

19 juillet 2004

# Table des matières

<b>1</b>	<b>Base de données expérimentale</b>	<b>3</b>
1.1	Présentation de la base . . . . .	3
1.2	Composition des données . . . . .	4
1.3	Personnalisation . . . . .	5
<b>2</b>	<b>Simulations</b>	<b>8</b>
2.1	Considérations générales . . . . .	8
2.2	Récapitulatif de performances . . . . .	8
2.3	Expériences . . . . .	11
2.3.1	Réduction de dimensionnalité . . . . .	11
2.3.2	Effet des seuils de tolérance . . . . .	14
<b>3</b>	<b>Considérations algorithmiques</b>	<b>15</b>
3.1	Mécanisme de prédiction . . . . .	15
3.2	Seuillage dynamique . . . . .	19

# Contexte

Nous avons proposé un classifieur en cascade que nous avons mis à l'épreuve sur quelques jeux de données. Les résultats se sont montrés encourageants mais les bases de travail étaient telles que nous ne pouvions tirer de réelles conclusions. Nous procédons ici à quelques nouvelles expériences afin de consolider notre système et en montrer l'efficacité avec plus d'assurance, mais cet aspect ne constitue pas la motivation centrale de notre compte-rendu.

Ce document est rédigé comme la continuation du rapport de recherche [3] dont il reprend les définitions et notations. L'objet de cet opuscule est principalement de mettre en forme un certain nombre d'améliorations (aménagements algorithmiques et introduction d'heuristiques) et développons des voies d'expérimentations.

# Chapitre 1

## Base de données expérimentale

Les expériences commentées précédemment montrent des résultats encourageants. Cependant, il convient de réaliser de nouvelles simulations avant de tirer quelque conclusion quant aux performances de notre classifieur. En effet, les manipulations effectuées jusqu'à présent se sont appuyées sur des jeux de données modestes en cardinalité. Nous présentons ici une série d'expériences impliquant une base plus conséquente et pour laquelle des relevés de performance pour différents autres classifieurs ont été collectés.

### 1.1 Présentation de la base

Le corpus *Modified NIST* ou MNIST a été mis au point par LeCun<sup>1</sup> *et al.* [1]. Il s'agit d'une base de données décrivant une collection de chiffres manuscrits. Chaque chiffre est associé à une image en niveaux de gris sur 256 valeurs et de taille 28 (exemples en figure 1.1). Le vecteur de forme correspondant consigne les valeurs de luminance pour chaque pixel. Ainsi, un chiffre donné est représenté par un vecteur numérique de taille  $28 \times 28 = 784$ .



FIG. 1.1 – Exemples d'images de chiffres manuscrits extraites de MNIST.

---

1. <http://yann.lecun.com/exdb/mnist/index.html>

## 1.2 Composition des données

MNIST est en réalité la réunion de deux sous-ensembles issus des bases **SD-1** et **SD-3** proposées par le consortium **NIST** (*National Institute of Standards and Technology*). Les échantillons manuscrits de SD-1 et SD-3 ont été collectés respectivement parmi une population de lycéens et un groupe d'employés de bureaux du services des recensements. D'où une différence significative dans la qualité des écritures. Les formes de SD-3 sont réputées plus aisées à reconnaître par classification automatique.

SD-1 contient 58527 images de chiffres manuscrits produits par 500 scripteurs différents. Les échantillons sont "mélangés", c'est à dire indexés sans tenir compte des auteurs. A l'opposé, les images de SD-3 sont organisées de par séquences de mains. Un travail de réarrangement a donc été nécessaire lors de la création de MNIST. Les images correspondant aux 250 premiers scripteurs de SD-1 sont réaffectées dans une nouvelle base d'apprentissage, alors que les autres sont consignées dans une nouvelle base de test. La base d'apprentissage est alors complétée par autant d'éléments de SD-3 que nécessaires pour atteindre une cardinalité de 60000 formes. Le choix des individus dans SD-3 est effectué en débutant par le premier et en suivant l'ordre d'indexation. La base de test est augmentée de manière similaire jusqu'à atteindre 60000 vecteurs. L'adjonction d'individus issus de SD-3 débute par celui d'indice 35000 et suit l'ordre d'indexation. La répartition par classe des bases de MNIST est donnée par le tableau 1.1.

Classe	Apprentissage	Test	Classe	Apprentissage	Test
0	5923	980	5	5421	892
1	6742	1135	6	5918	958
2	5958	1032	7	6265	1028
3	6131	1010	8	5851	974
4	5842	982	9	5949	1009

TAB. 1.1 – *Distribution des individus par classes dans la collection MNIST.*

Notons encore que les images d'origine dans SD-1 et SD-3 sont binaires. Elles ont été retaillées au format  $20 \times 20$ . Cette transformation est assortie d'un traitement anti-aliasing. D'où des images résultantes en nuances de gris.

Enfin, les formes sont centrées dans un cadre de taille  $28 \times 28$  qui donne les dimensions finales des échantillons de MNIST.

### 1.3 Personnalisation

Nous avons constaté que la représentation des individus de MNIST dans leur format d'origine (vecteurs de luminance à 784 composantes) induit des coûts calculatoires importants. Les expériences effectuées avec de telles données nous ont permis de situer notre classifieur en cascade par rapport à d'autres outils de référence. Mais il est admis que résoudre un problème de reconnaissance de formes ne saurait se ramener à choisir un classifieur "performant" *a priori*. En effet, un tel système n'existe pas: la mesure de performance d'un classifieur est nécessairement tributaire d'un contexte défini par la nature et la taille d'un problème, les données et leur représentation, etc. [2].

Nous préparons donc une mise en situation du classifieur en cascade dans le cadre de la reconnaissance de chiffres manuscrits, usuellement menée sur des formes caractérisées, et non données comme des vecteurs de pixels. Les caractéristiques sélectionnées seront en nombre réduit. Il s'agira principalement de descripteurs scalaires. Afin de les extraire, il est indispensable de disposer d'une version binarisée des images de formes.

La binarisation consiste à séparer les valeurs de luminance en deux classes: les valeurs hautes et basses. Pour ce faire, nous employons la méthode **Iso-Data**. Après une telle opération, nous constatons que certaines formes de la base NMIST (corpus d'apprentissage et de test confondus) souffrent de ruptures de connexités plus ou moins sévères (tableau 1.2). Les individus touchés représentent 2.95% de la collection. Dans la majorité des cas, ces phénomènes de brisure donnent lieu à deux voire trois composantes connexes par chiffre manuscrit et sont vraisemblablement liés au tracé du scripteur (figure 1.2).

Connexité	Apprentissage	Test	NMIST
1	58249	9686	67935
2	1499	229	1728
3	195	53	248
4	46	24	70
5	7	5	12
6	3	1	4
7	1	1	2
8		1	1

TAB. 1.2 – Nombre de composantes connexes par formes dans la base NMIST après binarisation.

Les exemples les plus flagrants sont donnés par certains représentants de la classe du chiffre *cinq*. Il est clair que la manière de calligraphier ce dernier est à l'origine de la coupure fréquemment relevée entre le corps du caractère et la barre supérieure.



FIG. 1.2 – Exemples de chiffres manuscrits au tracé non connexe. La ligne supérieure montre les images en niveaux de gris telles que proposées dans MNIST. La ligne inférieure présente leurs versions binarisées.

Les formes binarisées comptant plus de trois composantes connexes sont plus rares. La mauvaise qualité des images d'origine est à la source de ces "cas pathologiques". Les situations les plus extrêmes rencontrées dans le corpus MNIST sont illustrées par la figure 1.3.



FIG. 1.3 – Exemples d'images de mauvaise qualité (ligne supérieure) et des conséquences sur la binarisation (ligne inférieure). Le nombre de composantes connexes relevé est de 6, 6, 6, 6, 7, 7 et 8 pour chaque image de gauche à droite.



# Chapitre 2

## Simulations

### 2.1 Considérations générales

Dans la plupart des expériences menées sur la base MNIST, la totalité des vecteurs d'apprentissage est exploitée, mais seuls 10000 individus (provenant à part égale de SD-1 et SD-3) sont testés. Afin de comparer les résultats obtenus par notre classifieur avec ceux constatés par ailleurs, nous nous conformons à ce choix. Pour nos besoins, nous avons scindé la base d'apprentissage en deux parties de cardinalités égales. Les 30000 premières formes sont destinées faire office de support pour l'élaboration des classifieurs élémentaires. Les 30000 représentants non encore utilisés servent à leur validation.

### 2.2 Récapitulatif de performances

Les tableaux 2.1, 2.2 et 2.3 consignent les performances relevées à l'issue d'expériences sur la base MNIST. Pour certaines de ces simulations, la base d'apprentissage a été artificiellement augmentée par adjonction d'individus supplémentaires. Ces derniers résultent de distorsions appliquées sur des formes choisies aléatoirement dans la base d'apprentissage. Les opérations de déformation sont principalement la translation, la rotation (inclinaison) et la dilatation.

Classifieur	Reconnaissance (%)
Réseau neuronal (linéaire) à une couche	88.0
Réseau neuronal (linéaire) à une couche avec correction d'inclinaison	91.6
Classifieurs linéaires combinaisons deux à deux	92.4
Plus proches voisins (distance euclidienne)	95.0
Plus proches voisins (distance euclidienne) avec correction d'inclinaison	97.6
Analyse en composantes principales (40 compo- santes principales) et classifieur quadratique	96.7
Plus proches voisins et distance tangente (16)	98.9
SVM à noyau polynômial de degré 4	98.9
SVM à noyau polynômial de degré 5 (ensemble réduit)	99.0
SVM à noyau polynômial de degré 9 avec base augmentée	99.2

TAB. 2.1 – Performances réalisées sur la base MNIST par divers classifieurs (linéaires et quadratiques) et machines à vecteurs de support.

Classifieur	Reconnaissance (%)
Réseau neuronal à deux couches (300 unités cachées)	95.3
Réseau neuronal à deux couches (300 unités cachées) avec base augmentée	96.4
Réseau neuronal à deux couches (300 unités cachées) avec correction d'inclinaison	98.4
Réseau neuronal à deux couches (1000 unités cachées)	95.5
Réseau neuronal à deux couches (1000 unités cachées) avec base augmentée	96.2
Réseau neuronal à trois couches (300 et 100 unités cachées)	96.95
Réseau neuronal à trois couches (300 et 100 unités cachées) avec base augmentée	97.5
Réseau neuronal à trois couches (500 et 150 unités cachées)	97.05
Réseau neuronal à trois couches (500 et 150 unités cachées) avec base augmentée	97.55

TAB. 2.2 – Performances réalisées sur la base MNIST par classifieurs neuronaux. Le tableau montre les taux de confusion constatés sur la base de test MNIST à 10000 individus.

Classifieur	Reconnaissance (%)
LeNet-1 (entrées de taille 16)	98.3
LeNet-4	98.9
LeNet-4 avec plus proches voisins en remplacement de la dernière couche neurale	98.9
LeNet-4 avec apprentissage local en remplacement de la dernière couche neurale	98.9
LeNet-5	99.05
LeNet-5 avec base augmentée (distorsions importantes)	99.25
LeNet-5 avec base augmentée (distorsions modérées)	99.2
LeNet-4 avec boosting avec base augmentée	99.3
Plus proches voisins et appariement contextuel	99.33

TAB. 2.3 – Performances réalisées sur la base MNIST par le classifieur LeNet.

## 2.3 Expériences

### 2.3.1 Réduction de dimensionnalité

Afin de limiter les coûts calculatoires engendrés par les opérations de prédiction (lors des étapes de validation intermédiaires et de test sur les exemples non appris), il est possible de procéder à une **réduction de dimensionnalité** des formes. Il suffit pour cela de ne conserver qu'un nombre réduit  $d'$  de composantes principales ( $d' < d$ ) à chaque récursion du classifieur. Par effet de changement de base, les vecteurs prototypes présenteront donc  $d'$  composantes au lieu des  $d$  initiales. Il en ira de même pour les individus du corpus de validation.

A titre d'illustration, si nous entreprenons un apprentissage sur 30000 vecteurs de construction et 30000 vecteurs de validation, il suffit de retenir les 298 premières composantes pour restituer 90% de la variance dans le problème basique initial à dix classes. C'est à dire que nous pouvons réduire de plus de moitié la dimensionnalité des individus tout en conservant l'essentiel de

l'information.

Dans la pratique, nous fixons un seuil de variance expliquée minimal  $\nu \in ]0; 1]$ . Pour chaque sous-problème rencontré, nous arrangeons les vecteurs propres de la matrice de covariance par ordre décroissant des valeurs propres. Notons  $v_k$  (resp.  $V_k$ ) la  $k$ -ième valeur (resp. le  $k$ -ième vecteur) propre ainsi trié. Nous conservons les  $n_\nu$  premiers vecteurs propres pour constituer une matrice de changement d'espace en définissant

$$n_\nu = \min \left\{ n \in \llbracket 1..d \rrbracket \left| \frac{\sum_{k=1}^n v_k}{d} \geq \rho \right. \right\}$$

Autrement dit,  $n_\rho$  est le nombre minimal de composantes principales à conserver afin de restituer  $100\nu\%$  de la variance expliquée sur les données de construction. Il est propre à chaque niveau de récursion et à chaque jeu de données.

Le tableau 2.4 et la figure 2.3.1 consignent les résultats de quelques expériences menées sur la base MNIST. Nous avons fixé un seuil de reconnaissance minimal pour les classifieurs basiques à  $T_R = 0.95$ , et un seuil de confusion tolérable à  $T_C = 0.01$ . Nous examinons l'effet de la réduction de dimensionnalité pour différentes valeurs de  $\nu$ .

Construction	Validation	Taux de variance expliquée ( $\nu$ )	Reconnaissance globale (%)
30000	30000	0.80	95.78
30000	30000	0.85	95.89
30000	30000	0.90	96.06
30000	30000	0.95	96.03
30000	30000	1.00	96.09

TAB. 2.4 – Performances réalisées sur les formes "brutes" de la base MNIST. Les prédictions sont calculées sur 10000 formes de test. Les taux de reconnaissance basique requis et de confusion admissible sont fixés respectivement à 0.95 et 0.01.

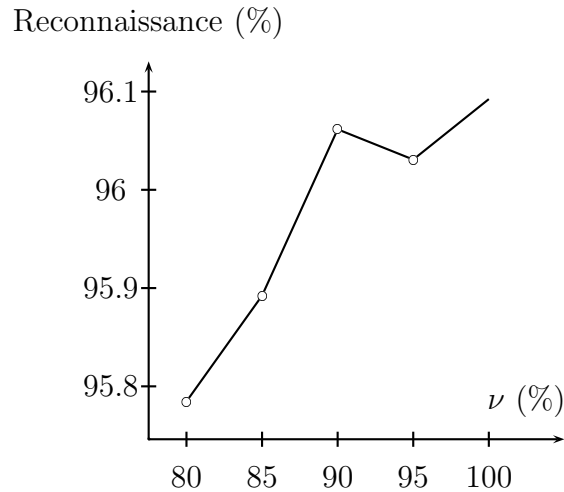


FIG. 2.1 – *Progression des performances en test par rapport au taux de variance expliquée.*

Lors de la première simulation, nous avons scindé arbitrairement la base d'apprentissage en deux ensembles de 30000 individus chacun dédiés à la construction et la validation. Nous conservons une part de variance expliquée de  $\nu = 0.8$  afin de réduire les temps de calcul. Le taux de reconnaissance obtenu est de 95.78% et dépasse largement celui des classifieurs de type purement linéaire (tableau 2.1). Nous entendons par là les réseaux neuronaux à une couche cachée (avec et sans correction des inclinaisons) et les combinaisons par paires sont concernés. Nous gagnons encore en performance face à la reconnaissance par plus proches voisins fondée sur la distance euclidienne et sans correction d'inclinaison. En revanche nous restons en dessous des classifieurs de type machine à vecteur de support.

L'inspection de la matrice de confusion globale pour les formes de la base de test nous permet de dégager quelques tendances. Nous notons un très bon comportement du classifieur sur les échantillons des classes de chiffres *zero* et *un*. Les taux de confusion sont tous très faibles (en dessous de 0.5%). A l'opposé, les classes *quatre* et *neuf* sont très adhérentes entre elles. De même pour les classes *trois* et *cinq*. L'apparition de tels agglomérats s'explique par une certaine similitude morphologique des chiffres manuscrits.

Afin d'améliorer les résultats précédents, nous devons effectuer des ana-

lyses multivariées plus précises. Nous pouvons moduler la part de variance explicative retenue lors de chaque analyse en composantes principales. Mais il existe d'autres paramètres sur lesquels agir.

### 2.3.2 Effet des seuils de tolérance

Deux autres facteurs susceptibles d'influer sur la capacité de généralisation de notre classifieur en cascade sont les seuils de référence ( $T_R$  en reconnaissance et  $T_C$  en confusion) utilisés lors de l'examen des matrices de confusion intermédiaires pour établir les récursions. Afin de faciliter l'étude, nous choisissons de nous intéresser à  $T_R$  uniquement.  $T_C$  est déterminé automatiquement par la formule heuristique

$$T_C = \frac{1 - T_R}{k - 1}$$

où  $k$  est le nombre de classes du problème.

Nous effectuons plusieurs simulations mettant en jeu différentes valeurs de  $T_R$ . Nous remarquons que si nous conservons toute la variance expliquée lors de l'analyse en composantes principales, les variations de  $T_R$  sont sans effet sur les performances constatées. Nous projetons donc de fixer différents taux de variance conservée  $\nu$  et de réaliser autant de jeux d'expériences. Cette série expérimentale sera développée dans la suite de nos travaux.

# Chapitre 3

## Considérations algorithmiques

### 3.1 Mécanisme de prédiction

La procédure de classification telle que proposée précédemment [3] ne souffre pas d'alternative lorsque le corpus de test est réduit à une seule forme. Nous examinons à présent la situation dans laquelle la prédiction doit être menée sur un groupe de plusieurs individus.

L'**approche naïve** consiste à examiner isolément chaque vecteur de test en utilisant la procédure dédiée à ce cas de figure. Cette démarche peut s'avérer extrêmement coûteuse en temps de calcul. En effet, pour chaque individu dont nous cherchons à inférer la classe d'appartenance, nous chargeons en mémoire un classifieur basique (c'est à dire la totalité des prototypes, leurs étiquettes, l'individu de centrage, la matrice de changement de base) et cela autant de fois que l'exige le mécanisme de récursion pour produire un label de prédiction définitif. Ainsi, il est probable que certains classifieurs soient invoqués, stockés et détruits de nombreuses fois.

Afin d'éviter ce genre d'inflations de la complexité algorithmique, nous apportons quelques adaptations à nos procédures calculatoires.

Lors de l'apprentissage, nous indexons les classifieurs basiques construits à chaque récursion, par exemple par ordre d'apparition. La prédiction sur l'ensemble de test appelle des aménagements plus profonds. Contrairement à l'approche naïve, nous n'effectuons pas une classification complète pour



chaque individu successivement. Nous procédons par **classification partielle**.

A chaque vecteur de test nous affectons un label temporaire et une référence vers un classifieur basique. Nous sélectionnons  $\chi$ , le classifieur pointé de rang maximal. Lorsque plusieurs candidats sont recevables, nous optons pour le plus fréquemment référencé d'entre eux. Et si un choix est encore possible, nous le porterons vers le classifieur d'indice minimum. Nous chargeons les éléments constitutifs de  $\chi$  et effectuons une prédiction sur les formes dont la référence désigne  $\chi$ . Nous parlerons des **formes appelantes** et noterons  $\mathcal{T}_\chi$  leur ensemble. Pour chaque individu  $X$  de  $\mathcal{T}_\chi$ , le label temporaire est mis à jour. Il reçoit pour nouvelle valeur  $j$ , l'indice de la classe inférée via  $\chi$  sur  $X$ . De même, les références doivent être actualisées. La liste des classes adhérentes à  $j$  (pour le classifieur  $\chi$ ) permet de désigner un nouveau classifieur vers lequel pointer.

La procédure de prédiction sur un corpus de plusieurs individus peut être résumée par l'encadré suivant. Précisons que  $n$  est le nombre de classifieurs basiques construits et indexés par ordre d'apparition lors de la phase d'apprentissage. Ainsi, les fréquences de référencement peuvent être consignées dans un tableau de taille  $n$ . De même pour les rangs.  $adh(c)$  est la liste d'indices correspondant aux classes adhérentes à la classe  $c$ . Notons que par construction, une famille de classes adhérentes correspond de manière bi-univoque à un classifieur. Enfin,  $ppv(X)$  désigne le plus proche voisin de l'individu  $X$  parmi les prototypes connus.

```

Pour tout individu  $X \in \mathcal{T}$ 
     $label(X) \leftarrow -1$ 
     $référence(X) \leftarrow 0$ 
Finpour
 $fréquence(0) = |\mathcal{T}|$ 
Pour  $k \in \llbracket 1..n - 1 \rrbracket$ 
     $fréquence(k) \leftarrow 0$ 
Finpour
 $Faire \leftarrow \mathbb{T}$ 
Tant que  $Faire$ 
     $Faire \leftarrow \mathbb{F}$ 
     $R_m \leftarrow Argmax(\{rang(k), k \in \llbracket 0..n - 1 \rrbracket\})$ 
     $F_m \leftarrow Argmax(\{fréquence(k), k \in R_m\})$ 
     $\chi \leftarrow min(F_m)$ 
     $fréquence(\chi) \leftarrow 0$ 
    Charger le vecteur de centrage du classifieur  $\chi$ 
    Charger la matrice de changement de base du classifieur  $\chi$ 
    Charger les prototype du classifieur  $\chi$ 
    Charger les labels du classifieur  $\chi$ 
    Pour tout individu  $X \in \mathcal{T}$ 
        Si  $référence(X) = \chi$ 
             $label(X) \leftarrow label(ppv(X))$ 
             $N \leftarrow adh(label(X))$ 
             $référence(X) \leftarrow indice(N)$ 
             $fréquence(N) \leftarrow fréquence(N) + 1$ 
        Finsi
    Finpour
     $Cumul \leftarrow 0$ 
    Pour  $k \in \llbracket 0..n - 1 \rrbracket$ 
         $Cumul \leftarrow Cumul + fréquence(k)$ 
    Finpour
     $Faire \leftarrow (Cumul > 0)$ 
Fintant

```

Notons que la stratégie de sélection de  $\chi$  nous garantit de ne jamais invoquer plus d'une fois un même classifieur basique. La démonstration de cette propriété ne pose pas de difficulté majeure.

Considérons un problème à  $k$  classes, indexées par  $\llbracket 0..k-1 \rrbracket$  sans perte de généralité. Soit encore trois bases de données  $\mathcal{X}_L$ ,  $\mathcal{X}_V$  et  $\mathcal{X}_T$  quelconques mais fixées. Ces collections sont dédiées respectivement à l'apprentissage, la validation et au test. Par définition, tout classifieur opère nécessairement sur au moins deux classes. Donc le rang d'un classifieur quelconque est supérieur ou égal à 2 et il est alors possible de former au plus  $2^k - k - 1$  classifieurs basiques durant l'apprentissage.

Nous examinons la famille des classifieurs effectivement construits lors de la phase d'apprentissage que nous notons  $\Xi$ . Si  $\chi$  et  $\chi'$  sont deux éléments de  $\Xi$ , nous dirons que  $\chi'$  est un **raffinement** de  $\chi$ , et nous noterons  $\chi' \triangleleft \chi$  si  $\chi' = \chi$  ou s'il existe une suite  $\chi_0, \dots, \chi_p$  telle que

- Pour tout  $n \in \llbracket 0..p \rrbracket$ ,  $\chi_n$  est un classifieur basique effectivement construit.
- $\chi_0 = \chi$
- $\chi_p = \chi'$
- Pour tout  $n \in \llbracket 1..p \rrbracket$ , le classifieur  $\chi_n$  opère sur une famille de classes adhérentes issues de la construction de  $\chi_{n-1}$ .

Il est clair que la relation de raffinement décrit un **ordre** sur  $\Xi$ . L'ensemble ordonné  $(\Xi, \triangleleft)$  admet un élément **maximum**. Il s'agit du classifieur basique initial en phase d'apprentissage, opérant sur les  $k$  classes du problème. Le mécanisme de sélection des classifieurs lors de la prédiction sur le corpus de test est équivalent à une forme altérée de **parcours en largeur** dans le diagramme de Hasse associé 'à  $\Xi$  en ce sens qu'il se déroule par "épuisement" des maximaux.

La progression débute avec l'élément maximum  $M$ , c'est à dire le (seul) classifieur de rang  $k$ . Toutes les formes de la base de test sont examinées. Nous effectuons donc exactement  $|\mathcal{T}|$  prédictions. Une fois  $M$  examiné, nous formons  $\mathcal{M}_1$  l'ensemble des éléments de rang maximal de  $\Xi \setminus \{M\}$ . Par construction,  $\mathcal{M}_1$  est une antichaîne de maximaux dans  $(\Xi \setminus \{M\}, \triangleleft)$ . Il est donc possible d'invoquer successivement les classifieurs de  $\mathcal{M}_1$  puisqu'ils sont deux à deux incomparables pour la relation de raffinement. Et par argument de maximalité, nous sommes assurés qu'ils ne seront plus sollicités par la suite. Ce qui nous permet de réitérer le raisonnement avec  $\mathcal{M}_2$ , l'ensemble des éléments

de rang maximal dans  $\Xi \setminus (\{M\} \cup \mathcal{M}_1)$ .

En définitive, chaque classifieur sera bien sollicité une fois au plus, ses éléments constitutifs n'étant chargés en mémoire que si une forme de test l'appelle.

## 3.2 Seuillage dynamique

La nature-même du classifieur en cascade que nous étudions pose la question de la signification de  $T_R$  pour chaque niveau de récursion. En effet, une valeur donnée  $T_{R_0}$  ne représentera pas le même degré de contrainte dans l'analyse des confusions selon le nombre de classes impliquées dans le sous-problème considéré. Nous ajoutons donc un mécanisme de réévaluation de  $T_R$  basé sur le rang du classifieur basique  $\chi$  en cours de validation. La règle de correction sur  $T_R$  sera donnée par

$$T_R = \frac{T_{R_0} - 1}{k - 2} \tilde{k} - 2 \frac{T_{R_0} - 1}{k - 2} + 1$$

avec  $\tilde{k}$  (resp.  $k$ ) le rang de  $\chi$  (resp. du classifieur maximal).

Nous envisageons de reprendre la série d'expériences précédente en introduisant le mécanisme de réestimation de  $T_R$  et  $T_C$  par niveau de récursion. Le taux de reconnaissance basique initial sera fixé à 0.95 et le taux de confusion admissible  $T_C$  sera calculé en fonction de  $T_R$ . Les résultats feront l'objet d'un rapport ultérieur.

# Perspectives

Nous avons proposé une architecture de classifieurs en cascade entièrement conditionnée par les matrices de confusion intermédiaires. Cette construction s'est révélée féconde au niveau applicatif, tant sur des bases de données conventionnelles (par exemple Iris de Fisher) que des compilations plus proches de la vie courante (illustrées par MNIST) en termes d'enjeux et de difficulté technique. Mais elle est encore loin d'être épuisée en matière de possibilités de développement.

En premier lieu, des améliorations dans l'implémentation sont probablement envisageables, tant au niveau des structures de données que de l'algorithmie. Quoiqu'il s'agisse d'un aspect technique primordial de toute réalisation logicielle, l'amélioration de code n'est pas une préoccupation majeure de cet exposé. Nous préférons la laisser en suspens et y consacrer ultérieurement des développements spécifiques.

Une autre ouverture peut être faite quant au choix des classifieurs fondamentaux. Nous avons opté pour la règle du plus proche voisin assortie de l'analyse en composante. La motivation sous-jacente était multiple. Il s'agissait d'adopter un paradigme simple afin d'éviter des obstacles conceptuels et des difficultés de mise en œuvre exagérées.

Enfin, les paramètres tels les taux de reconnaissance minimal et de confusion maximal acceptables peuvent être déterminés par rapport à des heuristiques restant à définir. Ces règles de décision pourraient tenir compte de la taille des échantillons de données, du nombre de classes à discriminer, de la profondeur de récursion par rapport au problème principal, etc.

# Bibliographie

- [1] Yann LeCun Léon Bottou Yoshua Bengio Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.
- [2] Richard O. Duda Peter E. Hart David G. Stork. *Pattern Classification*. Wiley Interscience, 2001.
- [3] Jean Duong. Proposition d'un classifieur récursif. Technical Report RR01, Laboratoire de Reconnaissance de Formes et Vision (RFV), July 2003.