

Clustering incrémental pour un apprentissage distribué et robuste: vers un système évolutif...

Yann Prudent, Abdel Ennaji

PSI - FRE CNRS 2645 - Université de Rouen
{yann.prudent, abdel.ennaji}@univ-rouen.fr

Résumé. Ce papier présente un système d'apprentissage multi-classifieurs dont la conception est pilotée par la topologie des données d'apprentissage. Il présente un nouvel algorithme d'apprentissage de carte neuronale auto-organisée incrémentale en données. Cette carte est utilisée pour distribuer les tâches de classification sur un ensemble de classifieurs. Seuls certains classifieurs sont activés en fonction de l'activité des neurones de la carte pour une donnée. De plus, le système proposé permet d'établir un critère de confiance s'affranchissant totalement du type de classifieurs utilisés. Ce coefficient permet de contrôler efficacement le compromis *Erreur/Rejet*. Des résultats comparatifs de cette architecture sont donnés sur la base de segmentation d'images de l'UCI et en reconnaissance de chiffres manuscrits.

1 Introduction

En reconnaissance de formes, les phases d'apprentissage et de classification constituent des étapes fondamentales qui conditionnent en grande partie les performances. Les techniques d'apprentissage artificiel ont connu ces dernières décennies des avancées fondamentales à travers des modèles tels que les réseaux de neurones et les SVM (*Support Vector Machines* ou *Séparateurs à Vaste Marge*) [Vapnik, 1995], qui montrent globalement de bonnes performances. Ces méthodes, basées sur différentes théories et méthodologies, ont été considérées comme autant de solutions possibles à un même problème. Cependant, leur développement n'a pas permis de mettre en évidence la supériorité incontestable d'une méthode sur une autre face aux contraintes et à la complexité des applications pratiques. De plus, peu de modèles permettent de fournir une estimation de la fiabilité de la décision ou un coefficient de confiance qui ne soit pas simplement une probabilité *a posteriori* comme c'est souvent le cas. Une solution admise dans la communauté pour résoudre ce type de problèmes consiste à adopter une conception modulaire et distribuée du système de classification [Duin, 2002] [Kuncheva, 2002] [Gori et Scarselli, 1998].

Dans ce contexte, ce papier présente un système d'apprentissage multi-classifieurs dont la conception est pilotée par la topologie des données d'apprentissage. Plusieurs contributions sont à noter dans le domaine de la décomposition de tâches de classification. Nous pouvons distinguer les approches purement supervisées des approches hybrides combinant à la fois des algorithmes supervisés et non supervisés.

Dans [Jacobs *et al.*, 1991], les auteurs proposent une approche qui décompose l'ensemble d'apprentissage en plusieurs sous-ensembles puis entraîne plusieurs réseaux sur

ces sous-ensembles. L'approche présentée dans [Gori et Scarselli, 1998] procède par une phase d'apprentissage non supervisé (clustering) par cartes auto-organisatrices qui suppose de connaître *a priori* le nombre de clusters. Cette carte permet de spécialiser des groupes de neurones d'un Perceptron Multi-couches (PMC) en fonction des clusters détectés. Ces deux méthodes fondent la distribution du problème de classification soit sur des informations strictement supervisées soit des informations *a priori* difficiles à estimer pour un problème réel. Elles ne prennent pas en compte la distribution réelle des données dans l'espace de représentation.

Pour effectuer la distribution, certaines méthodes hybrides réalisent une première étape qui consiste à capturer la topologie des données dans l'espace de description. Ceci revient à un problème de classification automatique et implique donc de déterminer le nombre et la constitution des groupes (clusters) dans l'ensemble d'apprentissage. Les techniques les plus généralement utilisées dans ce domaine sont certainement les cartes Auto-Organisatrices de Kohonen [Kohonen, 1982] et les méthodes de regroupement par partitionnement. Dans la pratique, le principal inconvénient de ces méthodes est la nécessité de fournir le nombre de groupes à l'avance pour obtenir une bonne représentation des données. [Ribert, 1998] se sert d'une étude non supervisée par Classification Ascendante Hiérarchique (CAH) du problème pour le distribuer sur plusieurs PMCs (un PMC sur par cluster). Il obtient des résultats équivalents à un classifieur global (KPPV) pour les taux d'erreur maximaux, mais de bien meilleures performances quand il s'agit de minimiser le taux d'erreur. Toutefois la CAH, bien que très efficace sur les problèmes de *clustering* et ne nécessitant pas de connaître le nombre de clusters *a priori*, reste très coûteuse en temps et en espace. Dans [Hébert *et al.*, 1999], l'auteur reprend les travaux de [Gori et Scarselli, 1998] et tente de palier ces inconvénients en se servant d'un algorithme d'apprentissage non supervisé qui n'a besoin d'aucune connaissance préalable du problème: le réseau *Growing Neural Gas* (GNG) [Fritzke, 1995].

Nous proposons dans ce papier une extension du réseau GNG, le réseau *Long Life Growing Neural Gas* (LLGNG), qui a la propriété d'être incrémental en données. La notion d'apprentissage incrémental évoquée ici se définit comme la propriété de pouvoir adapter un réseau déjà entraîné pour apprendre de nouvelles données sans nécessiter de disposer des données déjà apprises. Autrement dit, cette notion traduit les capacités d'un modèle à répondre au dilemme stabilité/plasticité. L'algorithme mis au point est ensuite utilisé pour distribuer un problème de classification sur plusieurs classifieurs différents. La section suivante est consacrée à la présentation de notre algorithme de clustering, le réseau LLGNG. La démarche globale de conception modulaire de notre système et le facteur de confiance associé aux différents modules sont présentés dans la section 3. Des résultats préliminaires sont présentés dans la section 4 avant une conclusion et des perspectives de ce travail en dernière section.

2 Long Life Growing Neural Gas

Un des objectifs possibles de l'apprentissage non supervisé est d'effectuer un apprentissage qui peut être qualifié d' "apprentissage topologique". Dans ce but, les cartes de Kohonen [Kohonen, 1982] et le *Growing Cell Structure* (GCS) [Fritzke, 1994] effectuent une projection non linéaire des données d'apprentissage dans un sous-

espace discret de dimension choisie *a priori*. Pour la carte de Kohonen, la taille du réseau doit être fixée au préalable, à l'inverse du réseau GCS qui ajoute des neurones et des connexions au fur et à mesure de son apprentissage. Dans [Martinetz et Schulten, 1994] les auteurs proposent d'allier deux méthodes, l'algorithme *Neural Gas*(NG) [Martinetz et Schulten, 1991] et le *Competitive Hebbian Learning*(CHL) [Martinetz, 1993]. L'algorithme NG permet de déplacer les centres des classes en suivant la distribution des données. Le CHL permet d'établir des relations topologiques en induisant un sous-graphe de la triangulation de Delaunay (Triangulation de Delaunay induite). Dans ce type de réseau (NG/CHL), la dimension n'est pas fixe (Rectangle pour Kohonen, Hypertétraèdre pour GCS), mais varie d'une région de l'espace à l'autre. De plus, celle-ci n'est pas fixée *a priori*. Malgré ces avantages, la taille de ce réseau reste constante et fixée à l'avance. Dans [Fritzke, 1995], l'auteur propose le réseau *Growing Neural Gas* qui combine les avantages des réseaux GCS et NG/CHL. Il n'y a ni taille ni dimension à fixer *a priori*, et cette approche est présentée comme étant incrémentale en données.

Nous introduisons dans la suite et discuterons le principe de l'algorithme GNG. Un exemple synthétique nous permettra d'évaluer les capacités réellement incrémentales du GNG. Nous proposerons ensuite un nouvel algorithme s'inspirant du GNG mais qui se caractérise par de véritables capacités d'apprentissage incrémental vérifiant le dilemme plasticité/stabilité.

2.1 Growing Neural Gas

L'idée principale de cette méthode est d'ajouter progressivement de nouveaux neurones à un réseau composé initialement de seulement deux neurones connectés, et ceci grâce à une analyse locale de l'erreur quadratique engendrée par les données présentées. Les connexions qui relient les neurones n'ont pas de poids, elles n'ont pour but que de définir la structure topologique des données (voisinage). Ce réseau se caractérise par une gestion dynamique du nombre de neurones basée sur un principe compétitif pour ajouter ou supprimer des neurones et des connexions.

Chaque neurone n_i du réseau GNG possède un vecteur de référence w_i qui correspond à ses coordonnées dans l'espace de description. A chaque étape le réseau tente de minimiser l'erreur quadratique engendrée par le nouvel exemple en déplaçant les neurones les plus proches de cette nouvelle donnée de la même manière que dans l'algorithme de notre réseau (alg. 1). Dans cet algorithme, le coefficient ϵ_b (ϵ_n) correspond au pas d'adaptation du neurone gagnant (de ces voisins). Ces connexions ont un âge, et chaque connexion ayant un âge trop élevé est supprimée. Ainsi, chaque donnée présentée au réseau active la connexion qui relie les deux neurones qui sont situés le plus près d'elle dans l'espace de description. Si la connexion existe, son âge est réinitialisé à 0, sinon, elle est créée avec un âge initial nul. Des neurones sont rajoutés au réseau après un nombre constant d'étapes (λ). L'erreur quadratique accumulée par les différents neurones permet de définir le vecteur de référence et les connexions du nouveau neurone. Ceci est fait de la même manière que dans l'algorithme 1. Cette erreur est diminuée à chaque étape en la multipliant par un coefficient d .

Fritzke décrit ce réseau comme un réseau incrémental en données puisqu'il traite les données une par une et qu'il a une architecture constructive. Toutefois aucun résultat

sur un apprentissage incrémental tel que défini précédemment n'est présenté. Un réseau GNG peut-il apprendre réellement de nouvelles données sans utiliser celles déjà apprises et sans détériorer le réseau déjà entraîné? En d'autres termes, le réseau GNG est-il apte à répondre correctement au dilemme stabilité/plasticité.

La figure 1-a illustre un réseau GNG qui a été entraîné sur une distribution de données de forme cubique. Dans la figure 1-b, la distribution des données a évolué et seules les nouvelles données situées à droite ont servi à poursuivre l'apprentissage du réseau. Il en résulte, comme le montre la fig. 1-c, une déformation du motif initial traduisant un manque de stabilité du réseau malgré de bonnes propriétés en plasticité. Ce problème synthétique montre bien l'échec du modèle GNG en apprentissage réellement incrémental.

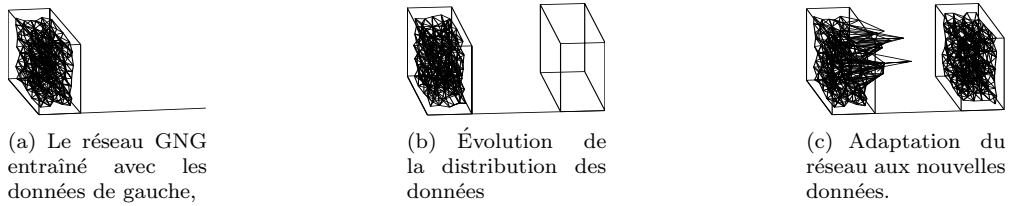


FIG. 1 – Apprentissage incrémental du réseau GNG

2.2 Notre contribution: Long Life Growing Neural Gas (LLGNG)

L'algorithme du LLGNG proposé reprend les principes généraux du réseau GNG mais introduit de nouvelles propriétés qui lui sont propres pour répondre au problème de l'apprentissage incrémental. Le réseau LLGNG possède deux types de neurones:

- des neurones *embryons*,
- des neurones *matures*.

Le réseau n'est constitué en réalité que de neurones *matures*, les neurones *embryons* ne servant que pour l'apprentissage (détaillé dans l'algorithme 1). Une donnée d'apprentissage \mathbf{x} active un neurone n_i si $\|\mathbf{x} - \mathbf{w}_i\|^2 < \sigma^2$. σ définit une région d'influence autour de chaque neurone de la carte. Ce paramètre est encore fixé *a priori*, mais devrait faire l'objet d'une estimation automatique durant l'apprentissage.

Quand une nouvelle donnée est présentée au réseau, l'adaptation du réseau se fait en distinguant trois cas de figure:

- Si la donnée n'active aucun neurone, un nouveau neurone *embryon* est créé avec un vecteur de référence $\mathbf{w} = \mathbf{x}$.
- Si la donnée active un seul neurone, un nouveau neurone *embryon* est également créé avec un vecteur de référence $\mathbf{w} = \mathbf{x}$, mais l'apprentissage continue de la même manière que pour le réseau GNG.
- Enfin si la donnée active plusieurs neurones, l'apprentissage de cette donnée est identique à celui du réseau GNG.

Chaque neurone possède en plus du vecteur de référence, un âge qui vaut initialement zéro pour un neurone *embryon*, et *majorite* pour un neurone *mature* (*majorite* étant un paramètre fixé *a priori*). Lorsqu'un neurone est le plus proche de la nouvelle donnée, son âge est incrémenté ainsi que celui de tous ses voisins directs. Si durant cette phase, un neurone *embryon* atteint un âge supérieur à *majorite*, il passe alors à l'état *mature*.

La distribution des figures 2-a et 2-b a été proposée dans [Martinetz et Schulten, 1991] pour tester le réseau NG, puis réutilisée dans [Fritzke, 1995] pour le GNG. Cette distribution possède des dimensions différentes dans différentes régions de l'espace de description. Ces deux figures donnent les résultats obtenus avec le GNG et le LLGNG, et montrent clairement les mêmes performances des 2 modèles dans le cas d'un apprentissage passif.

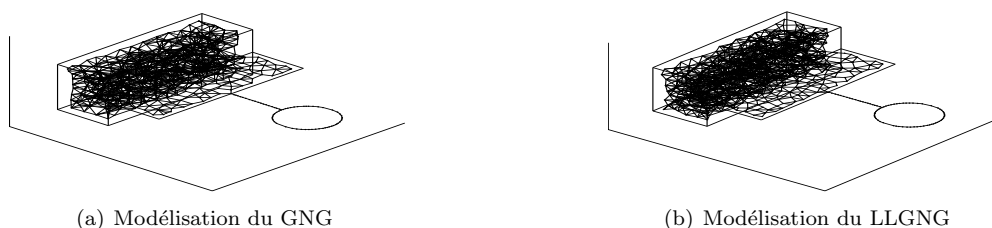


FIG. 2 – Apprentissage de la Distribution de Martinetz dans le cas d'un apprentissage passif

Les figures 3(a) et 3(b) présentent le réseau LLGNG entraîné sur les mêmes données et dans les mêmes conditions que pour la figure 1. Et contrairement au GNG, notre modèle est capable d'apprendre les nouvelles données tout en préservant les connaissances déjà apprises.



FIG. 3 – Apprentissage incrémental du réseau LLGNG

Dans toutes les validations proposées dans cet article, nous avons fixé $\sigma = 1$ et *majorite* = 50. Ces deux paramètres sont fixés *a priori* mais cette tâche ne demande que peu de connaissances du problème à traiter. Notons que si l'on définit $\sigma = \infty$ et *majorite* = 0, le réseau est identique au réseau GNG. Le réseau LL-

```

début
  tant que Le critère d'arrêt n'est pas satisfait faire
    Choisir une donnée  $\mathbf{x} \in S$ ;
    si  $n_1$  le neurone le plus près de l'entrée  $\mathbf{x}$  n'est pas activé alors
      créer un nouveau neurone embryon de coordonnées  $\mathbf{x}$ ;
      retourner;
    si  $n_2$  le deuxième neurone le plus près de l'entrée n'est pas activé alors
      créer un nouveau neurone embryon de coordonnées  $\mathbf{x}$ ;
     $e(n_1) += \|\mathbf{x} - w_{n_1}\|^2$ ; ( $e(n_1)$  : erreur du neurone  $n_1$ )
     $w_{n_1} += \epsilon_b(\mathbf{x} - w_{n_1})$ ;  $w_n += \epsilon_n(\mathbf{x} - w_n)$ ;  $// (n$  : voisins directs de  $n_1)$ 
    si  $n_1$  et  $n_2$  sont connectés alors
       $age_{n_1 \rightarrow n_2} = 0$ ;
    sinon
      créer une connexion entre  $n_1$  et  $n_2$ ;
    Supprimer les connexions avec un âge supérieur à  $a_{max}$ ;
    si un neurone se trouve isolé alors
      supprimer ce neurone;
    si le nombre d'itérations  $k$  est un multiple de  $\lambda$  alors
      Trouver le neurone  $q$  qui a la plus importante erreur accumulée;
      Trouver son voisin  $p$  qui a le plus d'erreur accumulée;
      Insérer le nouveau neurone mature  $r$  entre  $q$  et  $p$  ( $w_r = 0.5(w_q + w_p)$ );
      Créer les connexions  $q \rightarrow r$  et  $r \rightarrow p$ ;
      Supprimer la connexion  $q \rightarrow p$ ;
       $e(q) = \alpha \times e(q)$ ;  $e(p) = \alpha \times e(p)$ ;  $e(r) = e(q)$ ;
    pour Chaque neurone  $n$  faire
       $e(n) = d \times e(n)$ ;
    Incrémenter l'âge des neurones émanants de  $n_1$ ;
    pour Chaque neurone embryon  $n$  faire
      si  $age(n) \geq majorite$  alors
         $n$  devient mature
  fin

```

Algorithme 1: Algorithme d'apprentissage du réseau LLGNG

GNG est donc capable de rendre compte des relations topologiques d'une distribution quelconque de données. Ce modèle est capable en outre de reprendre un apprentissage sans tenir compte des données précédemment apprises. Il répond donc de manière satisfaisante au dilemme stabilité/plasticité contrairement aux autres techniques de clustering. Cette notion d'apprentissage incrémental, validée par ailleurs dans [Prudent et Ennaji, 2004], est fondamentale à notre sens dans la mesure où elle permet de répondre aux besoins grandissants en fouille de données et découverte de connaissances dans les problèmes actuels. Il devient donc possible de traiter de manière progressive et continue des problèmes complexes avec des bases d'apprentissage trop volumineuses et/ou en évolution continue. Cette limitation liée également à la complexité des algorithmes de clustering a été démontrée dans [Ribert, 1998] pour la CAH. De plus le réseau LLGNG possède tous les avantages requis pour traiter de tels problèmes. En effet, il n'est pas nécessaire de fixer *a priori* des paramètres tels que le nombre de clusters, la dimension de la carte ou la nature de la distribution des données. En outre, les paramètres utilisés sont constants dans le temps contrairement aux réseaux NG et aux cartes de Kohonen. Enfin le réseau LLGNG converge plus vite que le GNG vers une bonne représentation de la topologie des données. Ceci a été confirmé sur plusieurs problèmes et en particulier un problème de clustering à densité de points très variable proposé dans [Ribert, 1998] et qui met en échec la plupart des méthodes de clustering classiques. Tout ceci fait du réseau LLGNG une alternative aux approches classiques.

3 Conception du système de décision distribué

La structure topologique modélisée par le réseau *Long Life Growing Neural Gas* est exploitée pour la conception d'un système de décision distribué multi-classifieurs. En effet, ce réseau est utilisé dans un premier temps pour partitionner l'ensemble d'apprentissage initial en plusieurs sous-ensembles. Ce partitionnement est réalisé simplement en identifiant tous les éléments d'apprentissage ayant activé un neurone donné de la carte comme un sous-ensemble particulier. Chaque sous-ensemble est ensuite associé à un ou plusieurs classifieurs qui seront chargés de l'identifier. Ainsi, chacune des régions définies par le LLGNG est traitée de manière indépendante par un ou plusieurs classifieurs supervisés. Cette conception pilotée par la carte topologique nous offre la possibilité d'entraîner plusieurs classifieurs de différents types (PMC, SVM, KPPV, ...), ou aux paramètres d'apprentissage différents (noyaux, nombre de neurones, ...) pour chaque sous-ensemble. Il est ensuite possible de sélectionner le ou les classifieurs adéquats au problème par les méthodes de sélection de classifieurs proposées dans la littérature [Roli et Giacinto, 2002] [Kuncheva, 2002]. La diminution de la complexité du problème engendrée par ce partitionnement permet aussi d'envisager plus aisément de mettre en place des règles de combinaison des classifieurs pour améliorer la robustesse du processus de décision.

Ainsi, à chaque neurone n_i du LLGNG sont associées les données d'apprentissage telles que:

$$s_i = \{x \in S / d(x, w_i) \leq \sigma_i\}$$

σ_i est fixé *a priori* ou peut être calculé pendant la phase d'apprentissage du réseau

LLGNG. Comme nous le montre la figure 4, chaque neurone est donc associé à une zone d'influence hypersphérique de rayon σ_i .

Une fois la base d'apprentissage décomposée, un ou plusieurs classifieurs supervisés sont associés à chaque sous-ensemble d'apprentissage défini pour chaque neurone du LLGNG. Nous appelons par la suite cette association (neurone, classifieurs, sous-ensemble) un GNeurone. La décision prise par un GNeurone en phase de test concerne uniquement les éléments qui ont activé son neurone de la carte LLGNG et sera celle du ou des classifieurs qui lui sont associés.

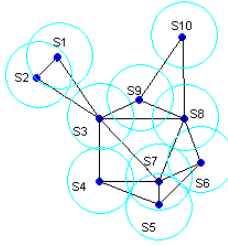


FIG. 4 – *Distribution de l'ensemble d'apprentissage S en plusieurs sous-ensembles s_i*

Cette technique de conception distribuée de classifieurs a pour autre avantage d'engendrer au niveau des classifieurs des frontières de décision fermées (fig.5).

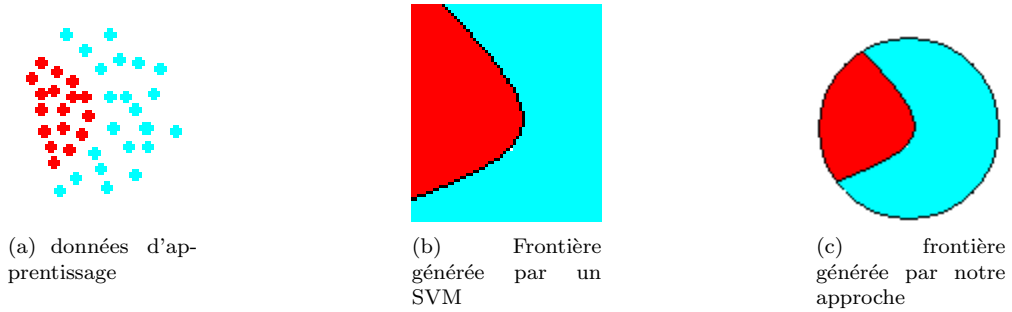


FIG. 5 – *Fermeture des frontières de décisions*

En effet, la génération de frontières de décisions ouvertes est un des principaux défauts de beaucoup de classifieurs tels que les PMCs ou les SVMs [Gori et Scarselli, 1998]. Ainsi, bien que reconnus comme performants, ils ne peuvent rejeter de manière efficace les données n'appartenant à aucune classe.

Par ailleurs, l'union de tous les sous-ensembles s_i générés par le LLGNG n'est pas forcément équivalente à l'ensemble d'apprentissage total S . Les données n'appartenant pas à cette union ne sont donc pas apprises et seront rejetées par notre système. Ce

deuxième pouvoir de rejet, en plus de celui pouvant être pris localement par la fonction de décision de chaque GNeurone, est un autre avantage indéniable de cette approche vis-à-vis de la fiabilité du processus de décision dans de grandes masses de données.

Lorsqu'une nouvelle donnée x à classifier est présentée au système, tous les GNeurones répondant à la condition $d(\mathbf{x} - w_i) \leq \sigma_i$ sont sélectionnés pour la décision. Nous choisissons alors au plus les K plus proches GNeurones de la nouvelle donnée (K est fixé *a priori*). A chaque GNeurone sélectionné on associe un coefficient de confiance v donné par:

$$v = \frac{|s_i|}{d(x, w_i)^\beta}$$

Ce coefficient tient compte de deux facteurs importants pour la fiabilité d'une décision:

- le nombre de données d'apprentissage du GNeurone ($|s_i|$),
- la distance de celui-ci avec la donnée à classifier.

β est une constante permettant de définir un compromis ***distance/taille de la base d'apprentissage***. Ce coefficient de confiance permet de pondérer le vote de chaque GNeurone, et par la même occasion, de rejeter des données lorsque la décision prise par l'ensemble des K GNeurones est jugée peu fiable. Il est à noter que le coefficient de confiance de chaque GNeurone s'affranchit totalement du ou des classifieurs associés à celui-ci. Ce critère permet à l'utilisateur de contrôler le compromis *Erreur/Rejet*. Il introduit donc un troisième niveau possible de rejet qui permet en particulier de prendre en compte des informations liées à la nature et les contraintes du problème traité, la qualité des données d'apprentissage et leur distribution dans l'espace de représentation.

4 Résultats

La validation de notre approche a été effectuée à travers plusieurs expérimentations pour mettre en évidence les performances en classification indépendamment de la pertinence des caractéristiques. Nous n'avons utilisé dans cette étude préliminaire que des SVM comme classifieurs pour les GNeurones. Notre système appelé KPPG (K plus proches GNeurones) a été comparé à des classifieurs bien connus, le KPPV (k plus proches voisins) et les SVM. Le noyau utilisé pour les SVM a été déterminé par un procédé *d'essai/erreur*. La première expérimentation a été effectuée avec la base de segmentation d'images de l'UCI [Murphy et Aha, 1992] appelée *Image*. Les données en dimension 19 de ce problème à 7 classes se répartissent en 210 exemples pour l'apprentissage et 2100 exemples en test. Le tableau 1 présente les résultats obtenus sur la base *Image* de l'UCI.

Nous avons ensuite validé notre approche sur le problème de reconnaissance de chiffres manuscrits de la base NIST. Le vecteur de caractéristiques considéré est constitué par les 85 caractéristiques (niveaux de gris) issues d'une pyramide de résolution à 4 niveaux ($1+4+16+64$) [Ballard et Brown, 1982]. La figure 6 donne un exemple de chiffres de la base NIST et la représentation retenue sur l'exemple d'un chiffre 2. Les bases d'apprentissage et de test contiennent respectivement 2626 et 2621 exemples.

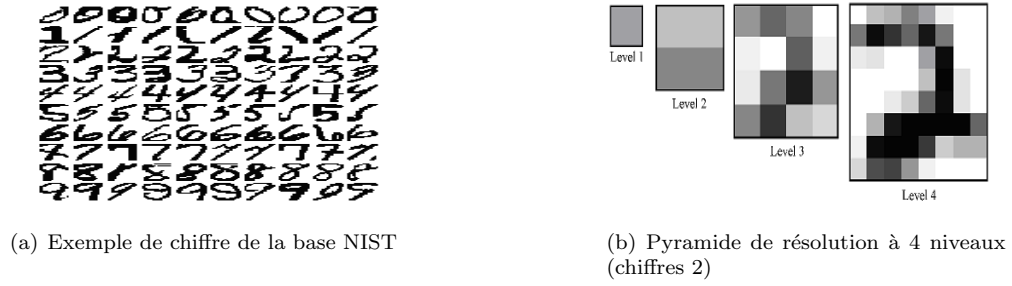


FIG. 6 – exemples et représentation de chiffres de de la base NIST

classifieur	reco(%)	erreur(%)
KPPV	87.57	12.43
SVM	93.95	6,05
KPPG	94.34	5.66

TAB. 1 – Résultats sur Image

	KPPV	SVM	KPPG
Max	96.83	97.90%	97.90%
$\leq 1.2\%$	93.5%	-	96.06%
$\leq 0.4\%$	81.59%	-	90,34%
$\leq 0.2\%$	73.73%	-	81.6%

TAB. 2 – Résultats sur NIST

Les deux courbes de la figure 7 donnent le taux de classification (taux d'erreur en fonction du taux de reconnaissance) pour chaque classifieur considéré à l'exception des SVM pour lesquels le taux d'erreur ne peut être paramétré. La courbe du KPPV a été obtenue en diminuant k, tout en vérifiant que tous les voisins soient de la même classe (le taux maximum d'identification est obtenu quand k=1). La courbe du KPPG a été obtenue de la même manière.

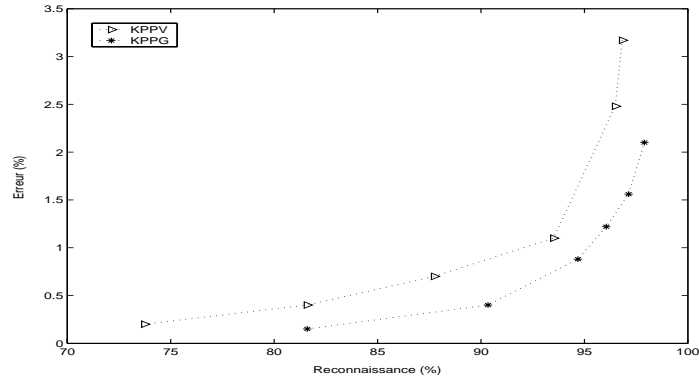


FIG. 7 – Erreur/Reconnaissance pour le KPPV et le KPPG sur la base NIST(85)

Les résultats préliminaires obtenus avec notre approche sont prometteurs. Le choix des 2 problèmes considérés ici a été motivé par la volonté d'évaluer les capacités de

généralisation des modèles à partir de bases d'apprentissage réduites. Le problème de la reconnaissance de chiffres manuscrits confirme ces capacités en évitant le risque de sur-apprentissage susceptible de se produire pour les bases bien connues de l'UCI à cause du déséquilibre des bases d'apprentissage et de test. Les performances sont équivalentes aux autres classifieurs pour un taux d'erreur maximal, mais se révèlent supérieures quand des taux d'erreurs faibles sont recherchés. En effet, les résultats obtenus jusqu'à présent montrent que dans le cas où un taux de reconnaissance maximal est nécessaire, le système proposé est au moins aussi performant qu'un SVM. De plus, dans le cas où le taux d'erreur doit être faible, le KPPG obtient de bien meilleurs résultats que le KPPV. Quand aux SVMs, ils ne sont pas, à notre connaissance, capables de rejeter efficacement des données (sauf pour des données se projetant dans la marge). Notre méthode permet donc d'allier les performances en reconnaissance de certains classifieurs comme les SVMs et un rejet efficace.

Le choix du réseau LLGNG, et la liberté totale du choix de classifieurs nous permettent de voir ce système comme un nouvel outil pour la conception d'un système d'apprentissage incrémental. Enfin, seul un classifieur de type SVM a été associé à chaque neurone du réseau. Le fait de multiplier les classifieurs à ce niveau permettra probablement d'améliorer les résultats obtenus.

5 Conclusion

Une nouvelle approche de conception de système de décision distribué et modulaire est présentée dans ce papier. Le principe de base est fondé sur l'exploitation de la topologie des données dans l'espace de représentation. Une telle conception est rendue possible grâce à l'utilisation combinée d'outils de clustering et d'apprentissage supervisé. Deux objectifs majeurs ont été recherchés dans cette étude. D'une part, la nécessité de disposer d'un système évolutif/incrémental a conduit à la mise au point d'un algorithme de clustering incrémental en données, performant et ne nécessitant que peu de paramétrages a priori contrairement aux approches classiques. La capacité d'apprentissage réellement incrémental en données ne pourra être atteinte qui si l'on dispose de classifieurs supervisés ayant cette propriété. Ce dernier point est l'une des principales perspectives de ce travail. Le deuxième objectif poursuivi concerne la fiabilité et la robustesse dans le processus de décision. Cette caractéristique est devenue actuellement un enjeu majeur dans les nouvelles applications de fouille et d'exploration de grandes masses de données et d'aide à la décision. Nous avons ainsi introduit dans notre système plusieurs niveaux de rejet. Le premier est relatif à la carte topologique et peut être qualifié de rejet par distance. Il permet en outre de fermer les frontières de décision engendrées par les classifieurs pilotés par la carte. Le deuxième niveau de rejet est associé aux différents classifieurs supervisés utilisés dans le système. La multiplication de ces classifieurs et l'introduction de règles de combinaison adéquates permettront de mieux exploiter cet aspect. Enfin, le dernier niveau de rejet est introduit pour mieux prendre en compte des éléments du contexte du problème indépendamment du comportement intrinsèque des outils d'apprentissage et de clustering. Le facteur de rejet proposé à cette fin considère ainsi des paramètres tels que l'effectif de la base d'apprentissage et permet par ailleurs à l'utilisateur de mieux intervenir sur le contrôle

du processus de décision.

Références

- [Ballard et Brown, 1982] D.H. Ballard et C. M. Brown. *Computer Vision*. Prentice Hall, 1982.
- [Duin, 2002] R.P.W. Duin. The combining classifier: to train or not to train. In IEEE Computer Society, editor, *ICPR*, pages 765–771, Quebec, 2002.
- [Fritzke, 1994] B. Fritzke. Growing cell structures — A self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7(9):1441–1460, 1994.
- [Fritzke, 1995] B. Fritzke. A growing neural gas network learns topologies. In *NIPS*, pages 625–632. MIT Press, 1995.
- [Gori et Scarselli, 1998] M. Gori et F. Scarselli. A multilayer perceptron adequate for pattern recognition and verification. *IEE TPAMI*, 20:1121–1132, 1998.
- [Hébert et al., 1999] Jean-François Hébert, Marc Parizeau, et Nadia Ghazzali. An hybrid architecture for active and incremental learning: The self-organizing perceptron (sop) network. In *IJCNN99*, Washington, DC, USA, July 1999.
- [Jacobs et al., 1991] R.A. Jacobs, M.J. Jordan, S.J. Nowlan, et G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.
- [Kohonen, 1982] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.
- [Kuncheva, 2002] L. I. Kuncheva. A theoretical study on six classifier fusion strategies. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 24, Feb 2002.
- [Martinetz et Schulten, 1991] T. Martinetz et K. Schulten. A "Neural-Gas" network learns topologies. In T. Kohonen, K. Mäkisara, O. Simula, et J. Kangas, editors, *Proc. International Conference on Artificial Neural Networks* (Espoo, Finland), volume I, pages 397–402, Amsterdam, Netherlands, 1991. North-Holland.
- [Martinetz et Schulten, 1994] T. Martinetz et K. Schulten. Topology representing networks. *Neural Networks*, 7(2), 1994.
- [Martinetz, 1993] T. Martinetz. Competitive Hebbian learning rule forms perfectly topology preserving maps. In Stan Gielen et Bert Kappen, editors, *Proc. ICANN'93*, pages 427–434, London, UK, 1993. Springer.
- [Murphy et Aha, 1992] P. M. Murphy et D. W. Aha. UCI repository of machine learning databases. Machine-readable data repository, University of California, Department of Information and Computer Science, Irvine, CA, 1992.
- [Prudent et Ennaji, 2004] Y. Prudent et A. Ennaji. Extraction incrémentale de la topologie des données. A paraître dans les Actes de SFC'04, 2004.
- [Ribert, 1998] A. Ribert. *Structuration Evolutive de données: Application à la construction de classifieurs distribués*. PhD thesis, Université de Rouen, 1998.
- [Roli et Giacinto, 2002] Fabio Roli et Gorgio Giacinto. design of multiple classifier systems. *Hybrid Methods in Pattern Recognition*, pages 199–226, April 2002.
- [Vapnik, 1995] V.N Vapnik. *The nature of statistical theory*. Springer, 1995.