



HAL
open science

Les K Plus Proches Classifieurs

Yann Prudent, Abdel Ennaji

► **To cite this version:**

| Yann Prudent, Abdel Ennaji. Les K Plus Proches Classifieurs. Jun 2004. sic_00001166

HAL Id: sic_00001166

https://archivesic.ccsd.cnrs.fr/sic_00001166v1

Submitted on 6 Dec 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Les K Plus Proches Classifieurs

Yann Prudent – Abdel Ennaji

Laboratoire PSI
Université et INSA de Rouen
F-76821 Mont-Saint-Aignan, France

{Yann.Prudent, Abdel.Ennaji}@univ-rouen.fr

Résumé : *Le travail présenté dans cet article tente d'apporter une contribution dans l'optique de la conception d'un système d'apprentissage incrémental. La démarche adoptée consiste à mettre en place un système de classification multiple où un ensemble de classifieurs de base sont pilotés par une carte neuronale auto-organisatrice. Celle-ci permet de rendre compte de la topologie des données d'apprentissage et d'activer le ou les classifieurs concernés. De plus, le système proposé permet d'établir un critère de confiance s'affranchissant totalement du type de classifieurs utilisés. Ce coefficient permet de régler le compromis Erreur/Rejet La carte utilisée est un Growing Neural Gas (GNG) qui est un modèle non supervisé incrémental en données. Des résultats comparatifs de cette architecture sont donnés sur la base de segmentation d'images de l'UCI et sur des données de reconnaissance de chiffres manuscrits.*

Mots-clés : Classification, Apprentissage supervisée et non supervisé, Réseaux de neurones, Coopération, Reconnaissance de formes.

1 Introduction

En reconnaissance de formes et plus particulièrement de l'écrit, les phases d'apprentissage et de classification constituent l'une des étapes les plus fondamentales qui conditionne en grande partie les performances du système. Ces techniques d'apprentissage artificiel [COR 02] [SHA 90] ont connu ces dernières décennies des avancées fondamentales à travers des modèles bien connus tels que les réseaux de neurones et les *Séparateur à Vaste Marge* (SVM)[VAP 95] et qui ont montré globalement de bonnes performances. Il n'en demeure pas moins que les contraintes de telles applications (grandes masses de données, données pas forcément étiquetées) conduisent souvent à l'échec de ces techniques ou au moins à une complexité d'utilisation qui les rend quasiment inexploitable. De plus, l'exploitation concrète de ces modèles est souvent rendue difficile à cause de leur manque d'intelligibilité, et surtout de la difficulté à contrôler efficacement leur comportement. En d'autres termes, peu de modèles permettent de fournir, en plus de la décision, un coefficient de confiance adapté à la situation et qui ne soit pas simplement une probabilité *a posteriori* comme c'est souvent le cas, difficile à interpréter et qui souvent reflète un comportement interne du classifieur.

Une des techniques admises dans la communauté consiste à adopter une conception modulaire et distribuée du système

de classification. Ces approches ont fait leur preuves depuis plusieurs années comme en témoigne les récentes publications sur le sujet [DUI 02] [KUN 02].

Dans ce contexte, ce papier présente un système d'apprentissage multi-classifieurs dont la conception est pilotée par la topologie des données d'apprentissage. Dans le même sens que [RIB 98] [HéB 99], nous avons adopté le principe d'extraire au préalable la topologie des données grâce à un algorithme de clustering. Notre choix s'est porté naturellement sur le modèle GNG [FRI 95] pour ses performances et ses capacités de construction incrémentale. Comme indiqué précédemment, une des contraintes les plus fondamentales en reconnaissance de formes est la possibilité de régler le compromis Erreur/Rejet. Ce principe rejoint par ailleurs l'hypothèse qu'il est indispensable dans de telles applications de pouvoir élaborer, à travers le classifieur, des frontières de décisions fermées; ceci est démontré par [GOR 98] à propos du *Perceptron Multi-Couches*. Dans ce sens, nous introduisons un critère de confiance local qui est associé à chacun des classifieurs. Ce critère de confiance prend en compte deux facteurs déterminants à notre sens pour ce genre de problématique. Ils concernent, d'une part l'effectif local des données d'apprentissage, et d'autre part la distance du point à classer par rapport aux neurones de la carte auto-organisée qui sont activés par cet élément. Ce facteur de confiance peut, en phase d'exploitation par l'utilisateur, être utilisé pour régler le compromis *Erreur/Rejet* en fonction des problèmes et des situations.

Dans la section 2, nous présentons rapidement quelques approches basées sur le principe de distribution de classifieurs. La section 3, présente les principes généraux de l'algorithme d'apprentissage du GNG. La section 4 est consacrée à la conception modulaire du classifieur multiple et la présentation du facteur de confiance qui est affecté à chaque classifieur. Les résultats préliminaires obtenus sur la base de segmentation d'images de l'UCI[MUR 92] et en reconnaissance de chiffres manuscrits sur une portion de la base NIST sont présentés en section 5.

2 Conception distribuée de classifieurs

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. Jacobs et Jordan proposent dans [JAC 91] une approche qui décompose l'ensemble d'apprentissage en plusieurs sous-ensembles puis entraîne plusieurs réseaux sur ces sous-ensembles. Cette approche est basée sur une étude complètement supervisée.

L'approche présentée dans [GOR 98] procède par une phase d'apprentissage non supervisé (clustering) par cartes auto-organisatrices. Celle-ci permet de spécialiser des groupes de neurones d'un PMC en fonction des clusters détectés. Ces deux méthodes présentent comme inconvénient la nécessité de connaître le nombre de clusters *a priori* dans les données, chose non aisée dans des problèmes réels complexes. Un autre inconvénient est le fait de fonder la distribution du problème de classification sur des informations strictement supervisées ou des informations *a priori* fournies par un « professeur », sans tenir compte de la distribution réelle des données dans l'espace de représentations.

Pour réaliser la distribution, la première étape 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 clusters dans l'ensemble d'apprentissage. Les techniques les plus généralement utilisées dans ce domaine sont certainement les cartes Auto-Organisatrices de Kohonen [KOH 82] et les méthodes de regroupement par partitionnement (tels que les k-means [MAC 67]). Dans la pratique, le principal problème avec ces méthodes est la nécessité de fournir le nombre de clusters à l'avance pour obtenir une bonne représentation des données.

Ribert propose une approche [RIB 98] qui se sert d'une étude non supervisée du problème pour le distribuer sur plusieurs PMCs. Il utilise une classification ascendante hiérarchique pour déterminer les clusters sur les données d'apprentissage, et entraîne un PMC sur chaque cluster. Il obtient des résultats équivalents au KPPV [DAS 91] sauf dans le cas où l'erreur doit être nulle. Dans ce cas son approche a de bien meilleurs résultats. Mais la classification hiérarchique ascendante, bien que très efficace sur les problèmes de *clustering*, et n'ayant pas besoin de connaître le nombre de clusters *a priori*, reste très coûteuse en temps et en espace.

Dans [HéB 99] l'auteur tente de palier à 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* [FRI 95]. Cette approche lui permet de reprendre les travaux de [GOR 98] sans avoir à connaître le nombre de clusters *a priori*.

Nous nous servons également d'un GNG pour distribuer le problème initial sur plusieurs classifieurs différents.

3 Growing Neural Gas

Le réseau GNG est une extension des réseaux de Kohonen [KOH 82] et *Growing Cell Structure*(GCS) [FRI 94]. On doit noter que plusieurs des concepts du réseau GNG sont empruntés au réseau *neural gas* proposé par T.M. Martinetz et K.J. Schulten [MAR 94]. Ces réseaux ont pour but de créer une structure qui reflète la topologie des données dans l'espace de représentations. Pour cela on opère une projection de l'espace de description vers une carte topologique de plus pe-

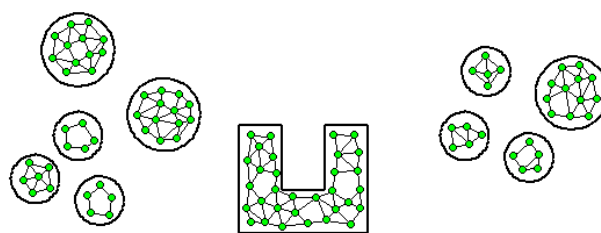


FIG. 1 – le gng sur un problème de clustering

tite dimension qui préserve la relation de voisinage. La projection doit donc respecter deux relations :

- Deux données voisines dans l'espace de description doivent être voisines dans la structure topologique.
- Les régions denses de l'espace de description doivent être assignées à plus de neurones.

Contrairement aux modèles de Kohonen et GCS, qui engendrent des structures topologiques dont la dimension est fixe (rectangle ou hypertétraèdre), le réseau GNG génère une structure dont la dimension peut varier d'une région de l'espace à l'autre.

L'idée principale de cette méthode est d'ajouter successivement de nouveaux neurones à un réseau qui est initialement très petit, grâce à une analyse locale de l'erreur quadratique engendrée par les données précédentes. 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 du réseau GNG ne possède qu'un vecteur de référence qui correspond à ses coordonnées dans l'espace de description. A chaque étape le réseau essaye de minimiser l'erreur quadratique engendrée par le nouvel exemple en déplaçant les neurones les plus proches de cette nouvelle donnée.

Les connexions qui relient les neurones n'ont pas de poids, elles n'ont pour but que de définir la structure topologique des données. Ces connexions ont un âge, et chaque connexion ayant un âge trop élevé est supprimée. Ainsi chaque donnée présentée au réseau cherche à activer 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 et son âge initial vaut 0. Lorsqu'une connexion n'est jamais activée, elle peut vieillir et disparaître.

Des neurones sont rajoutés au réseau si celui-ci n'arrive pas à apprendre assez bien. Ils sont rajoutés après un nombre constant d'étapes. L'erreur quadratique accumulée par les différents neurones permet de définir le vecteur de référence et les connexions du nouveau neurone. La figure 1 illustre le résultat d'un apprentissage du GNG sur un problème de *clustering* à deux dimensions. Un des principaux avantages de ce réseau réside dans sa structure qui est évolutive et n'est pas fixée *a priori*.

De part sa nature constructive, le GNG nous semble être le réseau le mieux approprié pour l'extraction de la topologie du problème initial.

De plus l'apprentissage d'un GNG se prête à un apprentissage incrémental en données.

4 Création de classifieurs spécialisés

La structure topologique représentée par le réseau *Growing Neural Gas* permet de distribuer le problème sur un ensemble de classifieurs. En effet grâce à ce réseau notre système génère plusieurs sous-ensembles de l'ensemble d'apprentissage. Il utilise chacun de ces sous-ensembles pour entraîner un ou plusieurs classifieurs. Chacune des régions définies par le GNG est ainsi traitée, de manière indépendante par un ou plusieurs classifieurs supervisés. Ces classifieurs n'ont à considérer que le sous-ensemble d'apprentissage filtré par chaque neurone du GNG. Pour chaque sous-ensemble, notre système a la possibilité d'entraîner plusieurs classifieurs, de différents types (PMC, SVM, KPPV, ...), ou aux paramètres d'apprentissage différents (noyaux, nombre de neurones, ...). Notre système sélectionne ensuite le ou les classifieurs adéquats au problème par les méthodes de sélection de classifieurs proposées dans la littérature des systèmes multi-classifieurs [GIA 99][GIA 00][ROL 02]. Il reste donc à trouver une méthode qui permet à partir du GNG de générer un ensemble de sous-ensembles de données d'apprentissage.

4.1 Partition en régions de Voronoi

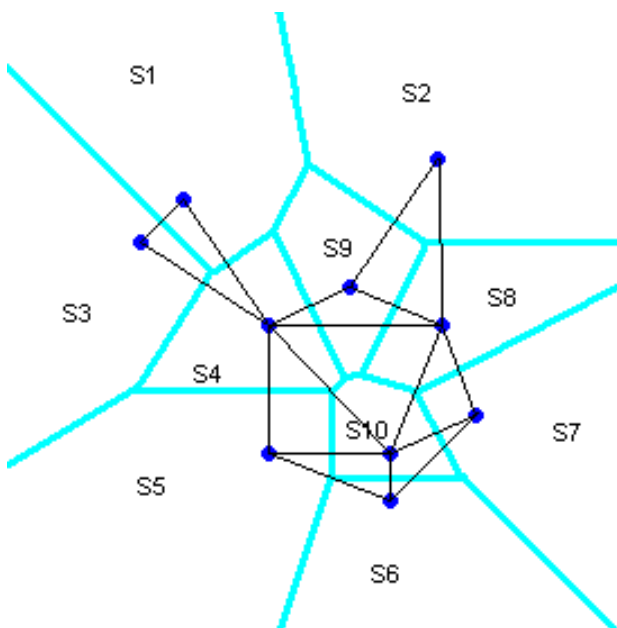


FIG. 2 – Partition en régions de Voronoi

La décomposition la plus « naturelle » consiste à créer un partitionnement $\pi = \{\pi_1, \dots, \pi_n\}$ tel que :

$$\pi_i = \{x \in S / d(x, w_i) = \min_{j=1, n} (d(x, w_j))\}$$

w_i étant la position du neurone c_i dans l'espace de description et $d(x, y)$ la distance euclidienne entre x et y . Ceci revient à créer une partition représentant les régions de Voronoi. Chaque partition π_i est un sous-ensemble s_i de l'ensemble d'apprentissage S qui est associé au neurone c_i , la figure 2 illustre ce partitionnement. Celle ci, bien qu'étant

la plus intuitive, a quelques défauts non négligeables. En effet le ou les classifieurs associés à chaque neurone ne considèrent que les données affectées à sa (leurs) région(s), ce qui peut être à l'origine de confusions dues à la nature même des frontières établies. C'est pour cette raison que cette méthode ne nous semble pas appropriée.

4.2 Sous-ensembles RBF

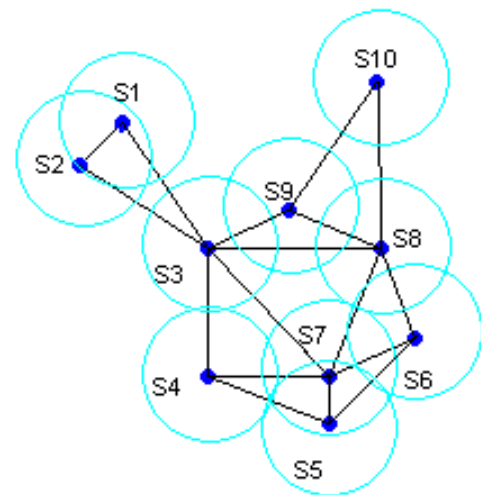


FIG. 3 – Sous-ensembles RBF

Cette méthode, contrairement à la première ne crée pas une partition mais plusieurs sous-ensembles qui ne représentent pas forcément tout l'ensemble d'apprentissage. Elle associe à chaque neurone c_i 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 GNG. La figure 3 illustre ces sous-ensembles, chaque neurone est associé à une zone d'influence hypersphérique de rayon σ_i . Cette méthode a pour avantage de fermer les frontières de décision engendrées par les classifieurs. 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. De ce fait, ces classifieurs, reconnus comme performants, ne peuvent rejeter de manière efficace les données qui n'appartiennent à aucune classe. La figure 4 illustre ce problème. Notre choix s'est donc orienté vers cette méthode pour organiser la base d'apprentissage.

4.3 Les K Plus Proches Classifieurs (KPPC)

Une fois la base d'apprentissage décomposée en plusieurs sous-ensembles, un ou plusieurs classifieurs supervisés sont associés à chaque sous-ensemble d'apprentissage défini pour chaque neurone du GNG. Chaque neurone du GNG est donc associé à la fois à un sous-ensemble d'apprentissage et à un ou plusieurs classifieurs entraînés sur ce sous-ensemble. Ces

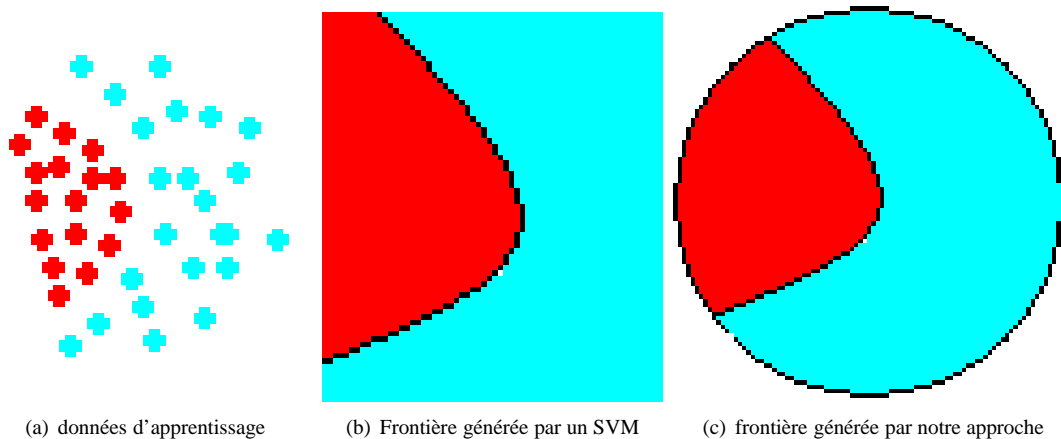


FIG. 4 – Fermeture des frontières de décisions

classifieurs vont se prononcer en phase de test pour tous les points qui activent ce neurone.

Dans la suite du papier nous parlerons pour simplifier d'un GNeurone comme étant l'association d'un neurone, d'un sous-ensemble et de classifieurs. Nous parlerons aussi de la décision d'un GNeurone comme étant la décision prise localement par la fonction de combinaison des classifieurs associés à un neurone du GNG. Lorsqu'une nouvelle donnée x à classifier est présentée au système, tous les Gneurones répondant à la condition $d(x - w_i) \leq \sigma_i$ sont sélectionnés pour la décision. Parmi tous les GNeurones sélectionnés, nous choisissons les K plus proches de la nouvelle donnée dans l'espace de description (K est fixé *a priori*).

A chaque GNeurone sélectionné on associe un coefficient de confiance v défini par :

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

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

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

β est une constante permettant de définir un compromis entre l'importance de la distance entre la nouvelle donnée et le GNeurone et l'importance de la taille du sous-ensemble d'apprentissage de celui-ci.

Ce coefficient de confiance permet de pondérer le vote de chaque classifieur et par la même occasion de rejeter des données lorsque la décision prise par les GNeurones est jugée peu fiable. Ce critère permet par ailleurs à l'utilisateur d'introduire un seuil qui permet de contrôler un compromis *Erreur/Rejet*.

5 Résultats

Pour tester la pertinence de notre système, nous l'avons appliqué à plusieurs bases. Ces expérimentations avaient pour but de mettre en évidence la performance de classifieurs et non de tester la pertinence des caractéristiques. On a utilisé comme classifieurs de base pour le KPPC uniquement des SVMs. Notre système appelé KPPC (K plus proches

Classifieurs) a été comparé à des classifieurs bien connus et éprouvés, le KPPV (k plus proches voisins) et le SVM.

Le noyau utilisé pour le SVM a été déterminé par un procédé *d'essai/erreur* en utilisant les noyaux suivants :

- noyau linéaire : $K(x, y) = x * y$,
- noyau polynomial : $K(x, y) = (\gamma xy + c_0)^{degree}$,
- noyau RBF : $K(x, y) = \exp(-\gamma|x - y|^2)$,
- noyau sigmoïd : $K(x, y) = \tanh(\gamma xy + c_0)$,

en faisant varier γ entre 0 et 1 par pas de 0.1, et *degree* entre 2 et 5 par pas de 1, nous avons pris $c_0 = 0$.

La première était la base de segmentation d'images de l'UCI [MUR 92]. Ce problème était composé d'une base d'apprentissage de 210 exemples et d'une base de test de 2100 exemples. L'espace de description était de dimension 20 et le problème composé de 7 classes (BRICKFACE, SKY, FOLIAGE, CEMENT, WINDOW, PATH, GRASS). Les résultats des classifieurs sur la base image sont ceux donnés par le tableau 1.

Le noyau SVM utilisé pour ce problème est un noyau linéaire.

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

TAB. 1 – Reconnaissance et erreur des trois classifieurs sur la base image

Nous avons aussi appliqué cette approche au problème de la reconnaissance de chiffres manuscrits sur la base de données NIST. Le vecteur de caractéristiques considéré était constitué par les 85 (1+4+16+64) niveaux de gris d'une pyramide de résolution à 4 niveaux [BAL 82]. La figure 5 donne un exemple des chiffres de la base de données NIST et la figure 6 illustre la représentation retenue sur l'exemple du chiffre 2. La base d'apprentissage était composée de 2626 exemples et la base de test de 2621.

Le noyau SVM utilisé pour ce problème est un noyau RBF avec $\gamma = 0.2$.

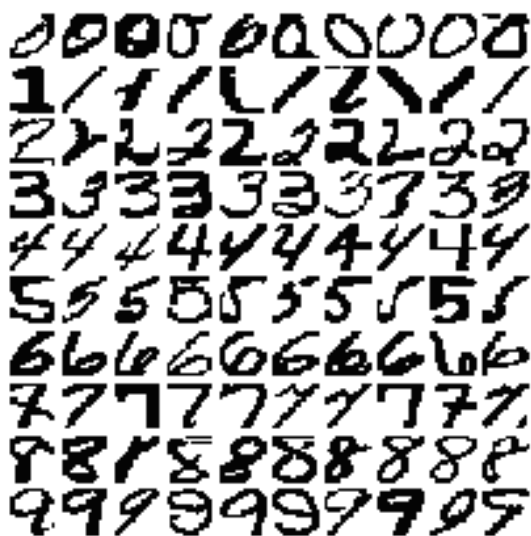


FIG. 5 – Exemple de chiffres de la base NIST

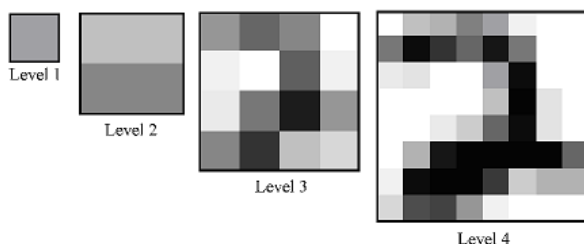


FIG. 6 – Pyramide de résolution à 4 niveaux (chiffres 2)

classifieur	reco(%)	erreur(%)
KPPV	96.83	3.17
SVM	97.90	2.10
KPPC	97.90	2.10

TAB. 2 – Reconnaissance et erreur des trois classifieurs sur la base NIST(85)

Les résultats du tableau 5 et les courbes de la figure 7 représentent l'évolution du taux de reconnaissance en fonction du taux d'erreur exigé. Les résultats du tableau 5 ont été obtenus pour chaque classifieur considéré de la manière suivante :

- Les résultats du KPPV ont été obtenus par un vote unanime des k plus proches voisins en augmentant k jusqu'à ce que le taux d'erreur demandé soit respecté (le taux maximum d'identification est obtenu quand $k=1$).
- De même, les résultats du KPPC ont été obtenus par un vote unanime des k plus proches GNeurones en augmen-

tant k jusqu'à ce que le taux d'erreur demandé soit respecté (le taux maximum d'identification est obtenu quand $k=1$). Dans cette figure le coefficient de confiance n'a pas été pris en compte pour le rejet, un simple vote à l'unanimité a été utilisé pour obtenir ces courbes.

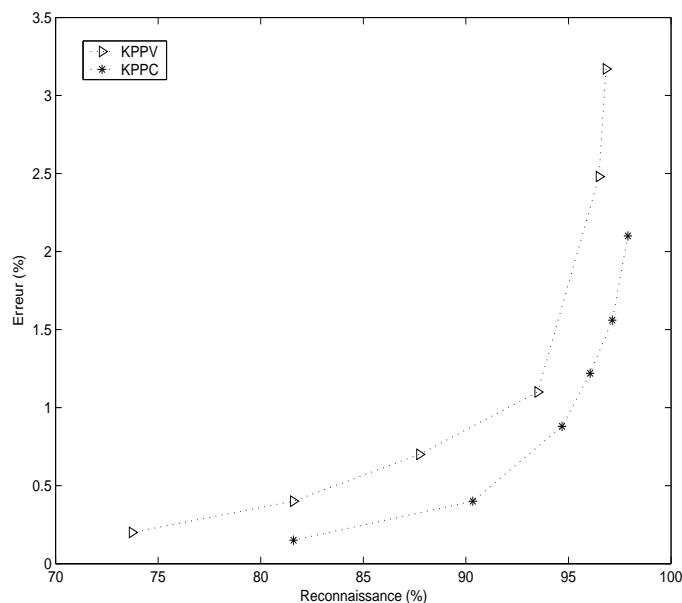


FIG. 7 – Erreur/Reconnaissance pour le KPPV et le KPPC sur la base NIST(85)

6 Conclusion

Les premiers résultats expérimentaux sur un problème de classification réel sont prometteurs, puisqu'ils prouvent que cette distribution conduit à définir des frontières de décision efficaces, particulièrement quand des taux d'erreur faibles sont exigés. 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 KPPC obtient de bien meilleurs résultats que le KPPV. Quand aux SVMs, ils ne sont pas, à notre connaissance, capable de rejeter efficacement des données. Notre méthode permet donc d'associer les performances en reconnaissance de certains classifieurs comme les SVMs et un rejet efficace. Ce rejet peut être obtenu grâce au coefficient de confiance de chaque GNeurone qui s'affranchit totalement du ou des classifieurs associés à celui-ci. Le choix du réseau GNG, 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.

Références

- [BAL 82] BALLARD D., BROWN C. M., *Computer Vision*, Prentice Hall, 1982.
- [COR 02] CORNUÉJOLS A., MICLET L., *Apprentissage Artificiel : concept et algorithmes*, Eyrolles, 2002.
- [DAS 91] DASARATHY B. V., *NN concepts and techniques, Nearest Neighbor (NN) Norms : NN Pattern Classification*

- Techniques, *IEE Computer Society Press*, , 1991, pp. 1-30, IEE Computer Society Press.
- [DUI 02] DUIN R., The Combining Classifier : to Train or Not to Train, SOCIETY I. C., Ed., *ICPR*, Quebec, 2002, pp. 765-771.
- [FRI 94] FRITZKE B., Growing Cell Structures — A Self-Organizing Network for Unsupervised and Supervised Learning, *Neural Networks*, vol. 7, n° 9, 1994, pp. 1441–1460.
- [FRI 95] FRITZKE B., A growing neural gas network learns topologies, TESAURO G., TOURETZKY D. S., , LEEN T. K., Eds., *Advances in Neural Information Processing Systems 7*, pp. 625–632, MIT Press, Cambridge MA, 1995.
- [GIA 99] GIACINTO G., ROLI F., Automatic Design of Multiple Classifier Systems by Unsupervised Learning, SPRINGER, Ed., *Proc. of the first Int. Workshop on Machine Learning and Data Mining in Pattern Recognition*, sept 1999, pp. 131-143.
- [GIA 00] GIACINTO G., ROLI F., Dynamic Classifier Selection, SPRINGER, Ed., *Proc. of the First Int. Workshop on Multiple Classifier Systems*, jun 2000, pp. 177-189.
- [GOR 98] GORI M., F. S., A Multilayer Perceptron adequate for pattern recognition and verification, *IEE Transactions on pattern analysis and machine intelligence*, vol. 20, 1998, pp. 1121-1132.
- [HÉB 99] HÉBERT J.-F., PARIZEAU M., , GHAZZALI N., An Hybrid Architecture for Active and Incremental Learning : The Self-Organizing Perceptron (SOP) Network, *IJCNN99*, Washington, DC, USA, July 1999.
- [JAC 91] JACOBS R., JORDAN M., NOWLAN S., , HINTON G., Adaptive mixtures of local experts, *Neural Computation*, vol. 3, n° 1, 1991, pp. 79-87.
- [KOH 82] KOHONEN T., Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, vol. 43, 1982, pp. 59–69.
- [KUN 02] KUNCHEVA L. I., A theoretical study on six classifier fusion strategies, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, Feb 2002.
- [MAC 67] MACQUEEN J., Some methods for classification and analysis of multivariate observations, *Proceedings of the 5th Symposium on Mathematics Statistics and Probability*, 1967, pp. 281-297.
- [MAR 94] MARTINETZ T., SCHULTEN K., Topology Representing Networks, *Neural Networks*, vol. 7, n° 2, 1994.
- [MUR 92] MURPHY P. M., AHA D. W., UCI repository of machine learning databases, Machine-readable data repository, 1992, University of California, Department of Information and Computer Science, Irvine, CA.
- [RIB 98] RIBERT A., Structuration Evolutive de données : Application à la construction de classifieurs distribués, PhD thesis, Université de Rouen, 1998.
- [ROL 02] ROLI F., GIACINTO G., design of multiple classifier systems, *Hybrid Methods in Pattern Recognition*, , 2002, pp. 199-226.
- [SHA 90] SHAVLIK J., DIETTERICH V., *Reading in Machine Learning*, Morgan Kauffman, 1990.
- [VAP 95] VAPNIK V., *The nature of statistical theory*, Springer, 1995.