Classphères : un réseau incrémental pour l'apprentissage non supervisé appliqué à la classification de textes

Juan-Manuel Torres-Moreno, Patricia Veláquez-Morales, Jean-Guy Meunier
Laboratoire d'Analyse Cognitive de l'Information. Université du Québec à Montréal C.P. 8888, Succursale Centre-Ville Montréal H3C 3P8 (Québec), Canada
E-mail : jmtorres@wassim.lanci.uqam.ca, meunier@jean-guy.uqam.ca

Le 7 septembre, 1999

Résumé

La classification des données textuelles pose des problèmes quand les textes sont-ils dynamiques, par exemple, quand on traite des données du cyberspace. Nous proposons *Classphères*, une variante originale des modèles de classifation connexionniste pour résoudre ce problème. Cette approche permet d'appliquer les performances classificatoires dynamiques des réseaux de neurones à des *corpus* textuels et produire donc des regroupements susceptibles d'interprétations sémantiques. Le traitement connexionniste est intégré à des processus linéaires rapides de pre-traitement du texte. La chaîne de traitement ainsi produite allie dynamicité et rapidité pour fournir à l'usager un outil précieux dans les tâches d'extraction des connaissances.

Mots-clés : *Classification, analyse des textes, apprentissage non supervisé, réseaux de neurones increméntaux.*

Abstract

Classification of textual data implies difficulties when texts are dynamic, for example, when working with data from cyberspace. We propose *Classphères*, a new connectionist approach for solving this classification problem. The approach allows the application of neural network dynamic classification performances to *corpus* texts, therefore producing clusterings susceptibles of semantic interpretation. The connectionist treatment is integrated to rapid linear processes of text preprocessing. The treatment machine produced in this way combines dynamism and rapidity, offering a precious tool for knowledge extraction.

Keywords : *Classification, Analyse of texts, Unsupervised Learning, Incremental Neural Networks.*

1. Introduction

De nos jours, un nombre grandissant d'institutions accumulent très rapidement de très grandes quantités de documents qui ne sont souvent catégorisés que d'une façon très sommaire. Rapidement, les tâches de dépistage, d'exploration et de récupération de l'information présentes dans ces textes, c'est-à-dire des *connaissances* sont devenues extrêmement ardues, sinon impossibles[8]. En raison de l'ampleur et de la dynamicité des *corpus*, cette extraction des connaissances devient de moins en moins possible dans des temps raisonnables ou faisables avec des ressources restreintes. Les réseaux de neurones sont actuellement plus utilisés dans plusieurs domaines du traitement de l'information textuelle et plus particulièrment celui de l'indexation[16], de la génération des liens hyper-textes[11] et la catégorisation textuelle[13]. La présente recherche porte sur l'extraction des connaissances sur des *corpus en constante mutation*, et l'approche a été de construire une chaîne de traitement qui inclut au coeur de sa démarche des traitements connexionnistes de type non *supervisé, incrémental et sensible* aux données entièrement nouvelles.

1.1 Les réseaux ART

Carpenter et Grossberg[1-7] ont proposé une théorie pour modéliser l'apprentissage reposant sur l'auto-organisation des connaissances en structures qui tend à résoudre le délicat dilemme stabilité-plasticité, la plasticité spécifiant la capacité du système à appréhender des informations nouvelles, et la stabilité, sa capacité à organiser les informations connues en structures stables. Cette théorie a donné naissance à plusieurs familles de modèles : ART1[1,2], ART2[3,6], *fuzzy* ART[4], ARTmap[5] et *fuzzy* ARTmap[7]. Ces modèles, comme les cartes auto-organisatrices de Kohonen[12], appartiennent aux réseaux de neurones à apprentissage non supervisé, dont les poids des interconnexions codent les prototypes des classes. Le modèle ART1 travaille avec des données binaires, ce qui le rend spécialement utile pour des tâches de classification textuelles (utilisé dans nos tests). Le nombre total de classes obtenu dépend du paramètre de vigilance ρ compris entre 0 et 1, fixé par l'opérateur. Plus ρ est proche de 1, plus les classes seront sélectives (comprendront moins d'éléments) et leur nombre important. Alors que pour des faibles valeurs de ρ le nombre de classes sera faible, chaque classe comportant un grand nombre d'éléments.

2. Un algorithme de classification non supervisé : Classphères

Bien que puisant pour des applications textuelles[9,10,15], ART1 présente les handicaps de ne pas pouvoir faire appartenir un même segment à plusieurs classes, et de dépendre du choix de ρ par l'utilisateur. Ce pour ceci que nous avons développé l'algorithme incrémental *Classphères*, basé sur des hypersphères. Les perceptrons hypersphériques avaient déjà été proposés pour l'apprentissage supervisé par Torres-Moreno[14,17], ce qui permet d'obtenir des classes avec des recouvrements. Dans le cadre du projet *CONTERM* du *LANCI* de l'*UQAM* a été développé une chaîne de traitement d'information textuelle. Elle comporte des processus de segmentation, de filtrage du lexique (enlever les mots fonctionnels, les mots à haute fréquence d'apparition, etc.) et de lemmatisation au besoin de l'utilisateur. Au contraire de l'analyse symbolique classique, ce premier processus est performant et peut être appliqué à des gros *corpus*, car il est rapide. Le texte peut contenir des mots ou des phrases mal écrites (ce qui est souvent le cas), mais *CONTERM* tolère une certaine quantité d'erreurs aux entrées. La segmentation peut être faite en utilisant soit des mots soit des *N*-grammes. Dans ce travail nous avons utilisé que des mots. La segmentation transforme un texte initial dans un ensemble de *P* vecteurs à composantes binaires. Chaque segment est alors représenté par un vecteur $\vec{\xi} = \{0,1\}^N$; où *N*=taille du lexique (determinée par les processus de filtrage et de lemmatisation) qui corresponde à la dimension *N* de l'espace des entrées Ainsi, chaque composant du vecteur montre la présence $\xi_i = 1$ ou l'absence $\xi_i = 0$ du mot *i* dans un segment. L'ensemble d'apprentissage est alors représenté par une matrice creuse binaire (car chaque segment ne contient qu'une petite quantité du lexique). La classification s'effectue alors sur l'ensemble

d'apprentissage $\vec{\xi}^{\mu}=\{0,1\}^{N}$;μ=1,2,...,*P* segments. Le classifieur est un réseau de neurones à apprentissage non supervisé qui fait un regroupement des segments qui se ressemblent en fonction d'une mesure de distance adéquate.

2.1 Distances de Minkowsky

Les vecteurs, pouvant s'interpréter comme des points dans un espace, on utilise souvent des mesures de distance entre eux. De manière générale, une distance est une fonction qui associe aux vecteurs $\vec{\xi}^{\mu}, \vec{\xi}^{\nu}$ et $\vec{\xi}^{\omega}$ de la même dimension *N*, un nombre $d \in \Re$; $d \geq 0$ tel que :

i) $d(\vec{\xi}^{\mu}, \vec{\xi}^{\mu}) = 0$ ii) $d(\vec{\xi}^{\mu}, \vec{\xi}^{\nu}) = d(\vec{\xi}^{\nu}, \vec{\xi}^{\mu})$ iii) $d(\vec{\xi}^{\mu}, \vec{\xi}^{\nu}) \leq d(\vec{\xi}^{\mu}, \vec{\xi}^{\omega}) + d(\vec{\xi}^{\omega}, \vec{\xi}^{\nu})$

La distance de Minkowsky de degré *p* entre deux vecteurs est définie comme :

$$d_p(\vec{\xi}^{\mu}, \vec{\xi}^{\nu}) \equiv \left\| \vec{\xi}^{\mu} - \vec{\xi}^{\nu} \right\|_p = \sqrt[p]{\sum_i^N \left\| \xi_i^{\mu} - \xi_i^{\nu} \right\|^p} \quad (1)$$

Pour la classification, nous avons utilisé les distances de degré p={1, 2}. La distance du degré p=2 est connue aussi comme distance euclidienne. Celle du degré p=1 s'appelle aussi distance d'Hamming quand les vecteurs sont à composantes binaires.

2.2 *Classphères* avec des distances d'Hamming

Classphères travaille sur un ensemble d'apprentissage $\vec{\xi}^{\mu}$ en créant des hypersphères où à l'intérieur on trouve les segments qui se ressemblent dans l'espace des mots. Chaque segment apporte de l'information nécessaire pour placer l'hypersphère. Si l'on utilise la distance euclidienne, on fixe un rayon ρ et on fait la présentation des segments séquentielle. L'algorithme trouve donc, les prototypes qui mieux codent les classes. Si l'on utilise la distance d'Hamming, alors on cherche les voisins les plus proches de chaque segment, en les regroupant par rapport à une distance minimale. L'architecture est montrée sur la fig. 1. Nous avons préféré surtout la version d'Hamming, car elle est mieux adaptée et performante dans un espace binaire. L'algorithme commence par créer la matrice diagonale d'Hamming $\{H(\mu,\nu)\}$ $1 \leq \mu \leq P-1; \quad \mu+1 \leq \nu \leq P$; entre les segments μ et ν; puis, il construit le vecteur $\vec{V}$ qui correspond à la distance minimale trouvée dans chaque colonne de *H*, c'est-à-dire, dans chaque segment. Une classe est ainsi formée en cherchant pour chaque ligne μ (segment) de *H*, les segments ν voisins (dans chaque colonne) qui correspondent au minimum de $V(\mu)$. Cet algorithme construit des hypersphères de rayon variable $\vec{V}$ où à l'intérieur on trouve les segments voisins.

3. Expériences et résultats

Nous avons effectué des plusieurs tests sur de textes de petite taille (<2740 mots) : «*Puces*», «*Souris*», «*Souris biologiques et informatiques*» (extraits de la presse française sur l'Internet), et «*Les rêveries du promeneur solitaire* (J.J. Rousseau)» ; et de taille moyenne (<42906 mots) : «*Discours de la méthode* (Descartes)», «*Discours sur l'origine et les fondements de l'inégalité parmi les hommes*» et «*Du contrat social ou principes du droit politique* (J.J. Rousseau)». Sur la fig. 2 nous montrons le nombre de classes obtenues avec *Classphères* à différentes tailles de segmentation pour chacun des textes. On voit que plus la segmentation est petite, on obtient plus de classes pour le même texte. Sur la fig. 3 nous montrons le nombre moyen de classes des différentes segmentations sur les mêmes textes obtenus avec ART et avec *Classphères*. On constate ici qu'ART obtient un nombre plus important de classes avec des écarts type plus grands. L'écart type augment en fonction de la taille du texte, pour les deux classifieurs.

Analyse du texte «*Puces (filtré)*». Ce texte est un mélange de deux textes appartenant à des auteurs différents : le premier parle des puces électroniques d'ordinateur, et l'autre de la retrouvée des puces et poux dans une compagnie militaire. Le texte a été filtré des mots fonctionnels. Compte tenu de sa petite taille, nous avons fait une segmentation de 50 termes. Les segments 1 à 5 traitent le sujet des puces informatiques, et les segments 5, 6 et 7 de poux et des puces. Il faut noter que le segment 5 parle des deux sujets. Nous avons comparé les résultats des classifieurs ART et *Classphères*, en constatant d'abord qu' ART trouve plus de classes: 5 ou 6 (selon $\rho = 0.001$ ou 0.2) contre 3 de *Classphères*; et en suite que celles de *Classphères* sont plus pertinentes, car les segments peuvent appartenir à des classes différents ; fait qui est interdit dans ART. Par exemple, dans le contexte, le mot «*puces*» est polysémique, donc il doit appartenir à des différentes classes, tel qu'elles ont été trouvées par notre classifieur : dans la classe I (segments 1 et 5), la classe II (segment 5) et finalement, la classe III (segments 5 et 6) ; qui correspondent respectivement à la classe purement «*informatique*», celle «*informatique et biologique*» et à la classe purement «*biologique*». Par contre, ART classe le mot «*puces*» dans la classe I (segment 1) et V (segments 5 et 6), en considérant à tort que le segment 5 traite *seulement* du sujet «*puces biologiques*». Nous montrons dans la fig. 4 les classes trouvées par les deux classifieurs.

4. Conclusions

Pour des tâches de classification textuelles, l'algorithme *Classphères* semble être un classifieur plus performant qu'ART. Bien qu'ART, en diminuant le paramètre ρ (choix parfois très difficile) obtient moins de classes, elles ne semblent guère plus pertinentes, car on a toujours l'interdiction d'avoir des segments appartenant à des différentes classes. Par contre, *Classphères* n'a pas besoin d'aucun paramètre, le nombre de classes est plus stable par rapport à la taille du segment, et les classes peuvent avoir des recouvrements, particulièrement dans les cas des textes ambigus. Il est en plus, au moins dix fois plus rapide qu'ART sur la même tâche classificatoire. Toutes ces qualités font de *Classphères* un algorithme qui peut être spécialement utile pour la classification de textes dynamiques, par exemple sur le web.

5. Bibliographie

[1] Carpenter, G., & Grossberg, S. An Adaptive resonnance Algorithm for Rapid Category Learning and Recognition. Neural Networks (4):493-504, 1991.

[2] Grossberg, S. The Adaptive Brain, volume I and II. Elsevier/North Holland, Amsterdam, 1987.

[3] Grossberg, S. & Carpenter, G. Art2: self-organizing of stable category recognition codes of analog input patterns. Applied Optics (26):4919-4930, 1987.

[4] Rosen, D. B., Carpenter, G. & Grossberg, S. Fuzzy art: fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, 759-771, 1991.

[5] Reynolds, J.H., Carpenter, G. & Grossberg, S. Artmap: a self-organizing neural metwork architecture for fast supervised learining and pattern recognition. IJCNN, pages 863--868, 1991.

[6] Grossberg, S. & Carpenter, G. Art2: an adaptive resonance algorithme for rapid category learining recognition. Neural Networks, 493-504, 1991.

[7] Markuzon, R., Carpenter, G. & Grossberg, S. Fuzzy artmap: a neural metwork architecture for incremental supervised learining of analog multidimensional maps. IEEE transactions on Neural Networks, 1992.

[8] Buckley, C., Salton, G. & Allan, J. Automatic stucturing and retrieval of large text file. ACM 37(2):97-107, 1994.

[9] Nault, G. & Meunier, J.G. A neural approach to terminological knowledge extraction in full text. Technical report, LANCI-UQAM, 1995.

[10] Seffah, A. & Meunier, J. G. Aladin: Un atelier de génie logiciel oriente objets pour l' analyse cognitive de textes. Technical report, LANCI-UQAM, 1996.

[11] Harie, S. & Vernnis, J. Utilisation de grands réseaux de neurones comme modèles de représentation sémantiques. NeuroNimes, 1990.

[12] Kohonen, T. Clustering, taxonomy and topological maps of patterns. IEEE Sixth International Conference on Pattern Recognition, (8):114-122, 1982.

[13] Balpe, J. P., Lelu, A., Papy, F., & I, S. (1996). Techniques avancées pour l' hypertexte. Paris.: Hermes.

[14] Torres-Moreno, J.M. Apprentissage et généralisation par des réseaux de neurones : Étude des nouveaux algorithmes constructifs. Thèse doctorat INPG, Grenoble, France 1997.

[15] Gabi, K., Extraction Dynamique de Connaissances à partir de Textes par Réseaux Neuronaux. Rapport de DEA en Sciences Cognitives, INPG, Grenoble, France 1997.
[16] Deerwester, S., Dumais, S., Furnas, T., Landauer, G. & Harshman, T.K. (1990) "Indexing by latent semantic analysis", Journal of the Amer.Soc for Infor science, 391-407.
[17] Reilly, D.E., Cooper, L.N., Elbaum, C. A neural model for category learning. Biological cybernetics. (45):35-41, 1982.

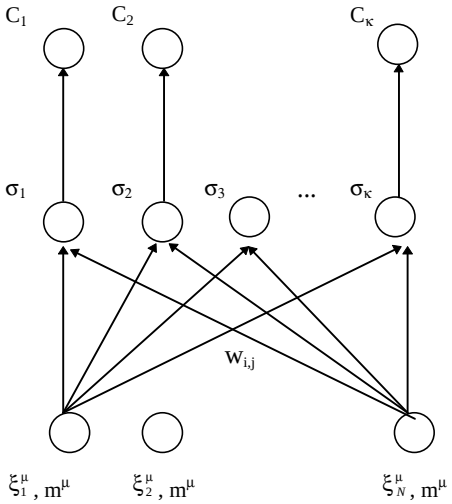

Fig. 1. Réseau engendré par *Classphères* avec les N entrées binaires correspondant au segment μ et sa masse m^{μ}. Nous montrons aussi les unités cachées σ reliés aux entrées par de poids $w_{i,j}$ qui codent les prototypes de C_k classes.

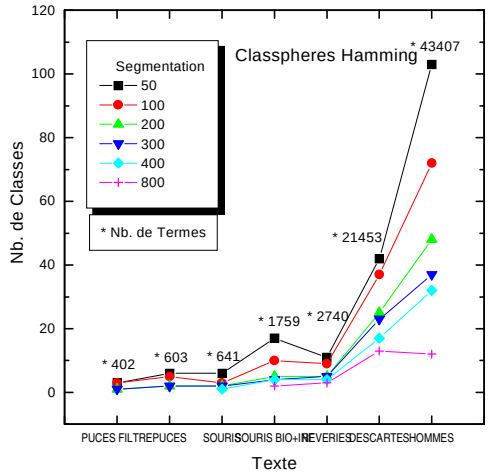

Fig. 2. Classes trouvées par *Classphères* en fonction de la taille du segment.

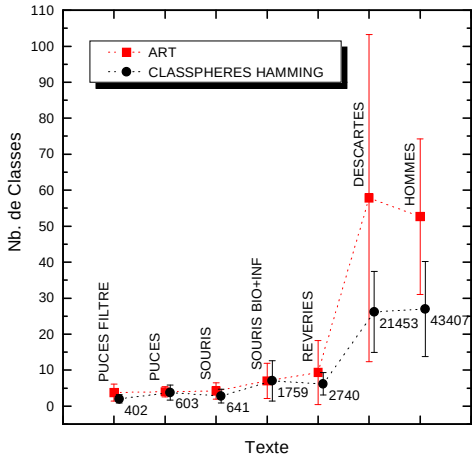

Fig. 3. Nombre moyen de classes trouvées par *Classphères* et ART pour les textes étudiés.

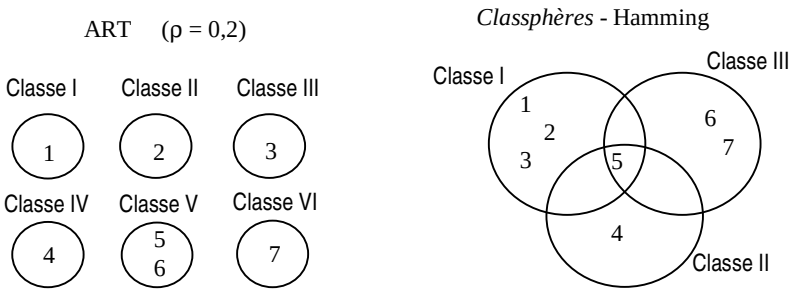

Fig. 4. Classes pour le texte « *Puces (filtré)* ». Chaque cercle représente une classe et à l'intérieur les segments qu'elle comporte.