Accélération de la méthode des K plus proches voisins pour la catégorisation de textes

Fatiha Barigou*, Baghdad Atmani**
Youcef Bouziane***, Naouel Barigou****

Département d'Informatique, Université d'Oran
BP 1524, El M'Naouer, Es Senia, 31 000 Oran, Algérie.
*,** Laboratoire d'informatique d'Oran
(fatbarigou, atmani.baghdad)@gamil.com,
,* (youcefbouzianemi, barigounaouel)@gmail.com

Résumé. Parmi la panoplie de classificateurs utilisés dans la catégorisation de textes, nous nous intéressons à l'algorithme des k-voisins les plus proches. Ces performances le situent parmi les meilleures méthodes de catégorisation de textes. Toutefois, il présente certaines limites: (i) coût mémoire car il faut stocker l'ensemble d'apprentissage en entier et (ii) coût élevé de calcul car il doit explorer l'ensemble d'apprentissage pour classer un nouveau document. Dans ce papier, nous proposons une nouvelle démarche pour réduire ce temps de classification sans dégrader les performances de classification.

1 Introduction

La Catégorisation de textes joue un rôle très important dans la recherche d'information et la fouille de textes. Cette tâche a été couronnée de succès en faisant face à une grande variété d'applications. Ce succès est du principalement à la participation croissante de la communauté d'apprentissage machine. Dans ce travail, nous nous intéressons à l'algorithme des K-plus proches voisins (Cover et Hart, 1967). Ce dernier développé tout d'abord par (Fix et Hodges, 1989) est devenu l'un des algorithmes les plus populaires dans la catégorisation de textes. Il est robuste et placé parmi les meilleurs algorithmes (Sebastiani, 2002). Toutefois, il présente certaines limites, (i) stockage mémoire énorme car il faut stocker l'ensemble complet d'apprentissage et (ii) coût élevé de calcul car il doit explorer l'ensemble d'apprentissage en entier pour pouvoir classer un nouveau document. Une solution intéressante à base d'automate cellulaire appelée CAkNN (Cellular Automaton combined with k-NN) a été proposée dans (Barigou et al., 2012) pour réduire le temps de classification, dans le cadre du filtrage de spam. Les expériences réalisées sur le corpus LingSpam ont montré que la méthode CAkNN permet d'atteindre de meilleures performances de classification comparée à d'autres travaux publiés dans le domaine de filtrage de spam. Dans ce papier, nous allons reprendre cette solution pour la catégorisation de textes et nous allons montrer à travers un ensemble d'expériences que CAkNN permet de réduire le temps de classification par une sélection d'un minimum d'instances d'apprentissage pour la classification d'un nouveau document et ceci sans que la performance prédictive n'en soit affectée. Ce papier est organisé comme suit :la section 2 est dédiée

aux travaux connexes. Dans la section 3 nous décrivons le principe de la méthode KNN. La section 4 décrit notre contribution pour améliorer cette méthode. Les expériences et les résultats sont présentés dans la section 5, et la conclusion est donnée dans la section 6.

2 Travaux connexes

Différentes solutions ont été proposées pour réduire la complexité de calcul. Comme le souligne (Bhatia et SSCS, 2010), nous distinguons les méthodes de sélection d'instances et les méthodes de réduction du temps de calcul. Les premières visent la réduction du nombre d'exemples dans la base d'apprentissage par certaines techniques d'édition en éliminant certains exemples qui sont redondant dans un certain sens (Gates, 1972). Les deuxièmes méthodes accélèrent la procédure de recherche lors de la classification par la mise en structures bien organisées de l'ensemble d'apprentissage (Liu et al., 2006). Cependant, pour des dimensions très importantes, l'espace requis croit d'une manière exponentielle.

3 L'algorithme des K-plus proches voisins

L'algorithme des k-plus proches voisins (KNN) est une méthode d'apprentissage à base d'instances. Il ne comporte pas de phase d'apprentissage en tant que telle. Les documents faisant partie de l'ensemble d'apprentissage sont seulement enregistrés. Lorsqu'un nouveau document à classer arrive, il est comparé aux documents d'apprentissage à l'aide d'une mesure de similarité. Ses k plus proches voisins sont alors considérés : on observe leur catégorie et celle qui revient le plus parmi les voisins est affectée au document à classer. La méthode utilise donc deux paramètres : le nombre k et la fonction de similarité. Une mesure de similarité très utilisée et que nous avons adoptée dans ce papier est la similarité cosinus (équation 1), qui consiste à quantifier la similarité entre deux documents en calculant le cosinus de l'angle entre leurs vecteurs.
Soit Q le nouveau document et soit $D = \{(d_1, c_1), (d_2, c_2), ...(d_n, c_N)\}$ l'ensemble d'apprentissage déjà étiquetés et soit $C = \{c_1, c_2, ...c_k\}$ l'ensemble des classes. $d_i \in D$ et Q sont représentés dans le vocabulaire $V = \{t_1, t_2, ...t_M\}$.
Nous définissons f comme étant la fonction KNN qui attribue une classe à une nouvelle instance Q. Dans notre cas, Cette fonction, utilise le vote majoritaire pondéré donné en équation 2.

$$sim(Q, d_i) = \frac{\sum_{t \in V} p_t(Q) * p_t(d_i)}{\sqrt{\sum_{t \in V} p_t(Q)^2 * p_t(d_i)^2}} \quad (1)$$

avec $p_t(d_i)$ le poids du terme t dans d_i et $p_t(Q)$ son poids dans Q.

$$f(Q) = \max_{c_k \in C}(\sum_{d_i \in kNN} sim(Q, d_i) * y(c_k, d_i)); \; y(c_k, d_i) = 1 \text{ si } d_i \text{ de classe } c_k; \; 0 \text{ sinon} \quad (2)$$

4 Méthode proposée

Dans cette section nous reprenons la méthode étudiée dans (Barigou et al., 2012) pour cette fois-ci la catégorisation de textes. Nous proposons une solution originale permettant de

surmonter l'un des inconvénients majeurs de la méthode KNN qui est le coût de classification dans une tâche comme la catégorisation de textes où nous manipulons des milliers de documents voire des milliers de milliers. Le principe de cette méthode est comme suit : au lieu de faire participer toutes les instances d'apprentissage pour la recherche des k-voisins ce qui va augmenter le temps de calcul, une sélection d'un sous ensemble réduit d'instances est tout d'abord réalisée. Cette opération de sélection a comme conséquence une réduction significative du temps de classification. L'approche proposée utilise la machine cellulaire CASI (Atmani et Beldjilali, 2007) pour représenter les instances d'apprentissage, d'une part, et extraire les documents pertinents, d'autre part.

4.1 Représentation des instances d'apprentissage

Nous proposons une nouvelle stratégie de représentation des documents d'apprentissage ; ces derniers vont être encodés dans une structure cellulaire. L'ensemble d'apprentissage est tout d'abord pré traité pour construire l'index. Nous distinguons trois étapes :

1. établir une liste initiale de termes en effectuant une segmentation de texte en mots ;
2. éliminer les mots inutiles en utilisant une liste prédéfini de mots vides et enfin ;
3. utiliser une variante de l'algorithme de Porter pour effectuer la racinisation des différents mots retenus.

Puisque trop de termes sont généralement extraits, certains d'entre eux devraient être sélectionnés comme des caractéristiques représentatives. Dans ce travail, les meilleurs termes sont sélectionnés par le gain informationnel. Trois couches d'automates d'états finis sont définies pour représenter la base d'apprentissage :

1. **La couche CelTerm** : composée de M cellules, représente le vocabulaire V. Les états des cellules se composent de trois parties : ET, IT, et ST étant l'entrée, l'état interne et la sortie. $ET, IT, ST \in \{0, 1\}$;
2. **La couche CelDoc** : composée de N cellules représente l'ensemble des documents d'apprentissage D. Les états des cellules se composent de trois parties : ED, ID et SD étant l'entrée, l'état interne et la sortie. $ED, ID, SD \in \{0, 1\}$;
3. **La couche CelRule** : composée de M règles, joue le rôle d'un index. Pour chaque terme appartenant à V, nous associons une règle ; elle nous indique dans quel (s) document (s) le terme se trouve. Les états des cellules se composent de trois parties ; ER, IR et SR étant l'entrée, l'état interne et la sortie. $ER, IR, SR \in \{0, 1\}$

Les termes sont liés à leurs documents par deux matrices de voisinage IM et OM :

- $\forall t \in \{t_j, t_j \in CelTerm; i = (1, M)\}; \forall r \in \{R_j; R_j \in CelRule; j = (1, M)\}$
  Si (t est une prémisse de r) Alors IM(t,r)=1 Sinon IM(t,r)=0 ;
- $\forall d \in \{d_i; d_i \in CELDOC; i = (1, N)\}; \forall r \in \{R_j; R_j \in CELRULE; j = (1, M)\}$
  Si (d est une conclusion de r) Alors OM(d, r)=1 Sinon OM(d,r)=0. .

4.2 Sélection des instances

Le processus de sélection permet de déterminer la contribution de chaque document d'apprentissage. Les documents ayant un plus grand nombre de termes communs avec l'instance

à classer seront sélectionnés pour participer dans sa classification. Nous allons tout d'abord, définir les concepts suivants :

1. *NTT* :le Nombre Total des Termes du vocabulaire V trouvés dans Q. ;
2. T(η) : le seuil défini par :

$$T(\eta) = \lceil \frac{NIT}{\eta} \rceil, \, avec \, \eta \geq 2 \tag{3}$$

3.

$$NTC(d_i) = \{t_j \in V; t_j \in d_i \cap Q\} \tag{4}$$

4.

$$d_i \, est \, pertinent \, si \, : NTC(d_i) > T(\eta) \tag{5}$$

Le processus de sélection des instances passe par trois étapes :
– Initialisation de la couche CelTerm ;
– Recherche de l'ensemble des documents $A = \{d_i\}$ tel que $d_i \cap Q \neq \emptyset$;
– Recherche du sous ensemble $E \subset A$ vérifiant la condition donnée dans (l'inéquation 5).

La recherche des instances est réalisée par l'application de la fonction booléenne globale $\delta fact \bullet \delta rule$ qui va récupérer les documents partageant au moins un terme avec Q (sous ensemble A). Les deux fonctions booléennes sont définies comme suit :

1. $\delta fact : (ET, \, IT, \, ST, \, ER, \, IR, \, SR) \, \overrightarrow{\delta fact} \, (ET, \, IT, \, ET, ER + (IM^T * ET), IR, SR)$;
2. $\delta rule : (ET, IT, ST, ER, IR, SR) \, \overrightarrow{\delta rule} (ED + (OM * ER), ID, SD, ER, IR, \neg(ER))$.

chaque document d_i de l'ensemble *A* se voit attribuer une valeur $NTC(d_i)$. Cette valeur correspond au nombre total de cellules actives obtenues par le produit booléen du vecteur *ET* de la couche CelTerm avec le vecteur $OM^T(d_i)$ Le seuil T(η) est ensuite appliqué afin de réduire davantage les données d'apprentissage et obtenir l'ensemble E qui sera utilisé par la méthode KNN. $E = \{d_i \in A \, avec \, NTC(d_i) \geq T(\eta)\}$; $|E| \ll |D|$

5 Étude expérimentale et Résultats

Dans ce qui suit nous évaluons la solution proposée et la comparons avec la méthode KNN.

5.1 Corpus et mesures de performances

Nous utilisons, dans ce papier, le corpus 20 NewsGroups. Ce corpus traite 20 catégories, chaque catégorie représente 5% du corpus, il contient au total 18828 documents. Nous avons utilisé 80% du corpus pour l'apprentissage et 20% de ce corpus pour le test. Pour évaluer les performances des deux méthodes KNN et CAkNN, nous calculons pour chaque catégorie la mesure $F_1(ci)$ (Sebastiani, 2002), donnée par la formule suivante : $F1(ci) = \frac{2*\pi*\rho}{\pi+\rho}$ avec π la précision et ρ le rappel. La mesure F1 globale, sur toutes les classes, est calculée à travers une moyenne des résultats obtenus pour chaque catégorie.
Le temps nécessaire pour classifier une nouvelle instance est calculé comme suit : dans le cas de la méthode KNN, ce temps considère le parcours de l'ensemble d'apprentissage *D* en entier. Par contre, dans le cas de CAkNN, le temps de classification est calculé en sommant le temps de sélection du sous ensemble *E* avec le temps de classification en considérant seulement le sous ensemble *E*.

5.2 Résultats expérimentaux

Les figures 1 et 2 regroupent les résultats obtenus pour le corpus 20Newsgroups dans le cas de $\eta = 5$. A partir de ces figures nous observons la contribution de la méthode CAkNN, qui consomme moins de temps, tout en restant plus performante que la méthode KNN.

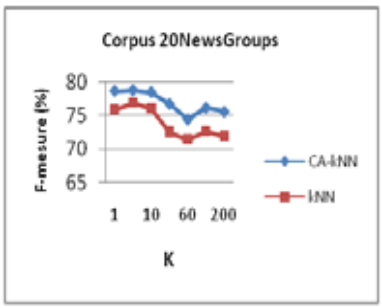

FIG. 1 – *Performance de classification du corpus 20NG en fonction du seuil K.*

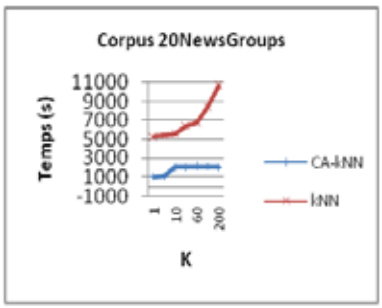

FIG. 2 – *Temps de classification de 20 % du corpus 20NG en fonction du seuil K.*

Nous en dégageons deux résultats intéressants, le premier concerne l'efficacité de l'approche. La qualité de prédiction de CAkNN est meilleure que celle du classifieur KNN. Le deuxième concerne la réduction du temps de classification obtenue grâce à la réduction drastique des instances d'apprentissage. Les résultats indiquent que l'écart entre les résultats avant et après application de la méthode CAkNN sont suffisamment significatifs.

6 Conclusion

Dans ce papier, nous avons proposé une nouvelle solution pour améliorer le temps de classification de la méthode KNN. Nous n'avons pas besoin de tout l'ensemble d'apprentissage pour classifier une nouvelle instance. Dans ce travail, nous proposons de sélectionner un sous ensemble réduit de documents pour cette tâche de catégorisation. Ce problème de sélection est traduit en un problème de manipulation d'opérations booléennes par l'utilisation de l'automate cellulaire de la machine CASI. Cet automate est premièrement censé filtrer les instances pouvant produire du bruit et deuxièmement, assurer la convergence de l'algorithme KNN en un temps de calcul intéressant. Comme perspective, nous prévoyons une étude comparative entre la méthode CAkNN et les autres solutions proposées dans (Bhatia et SSCS, 2010) (Gates, 1972), (Hart, 1968) et (Wilson et Martinez, 1989).

Références

Atmani, B. et B. Beldjilali (2007). Knowledge discovery in database : Induction graph and cellular automaton. *Computing and Informatics Journal 26,2*, 171–197.

Barigou, F., B. Beldjilali, et B. Atmani (2012). Improving knn spam based filter. *The Mediterranean Journal of Computers and Networks 8,1*, 21–29.

Bhatia, N. et V. SSCS (2010). Survey of nearest neighbor techniques. *International Journal of computer science and information security 8,2*, 302–305.

Cover, T. et P. Hart (1967). Nearest neighbor pattern classification. *IEEE Transactions Information Theory 13*, 21–27.

Fix, E. et J. Hodges (1989). Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review 57,3*, 238–247.

Gates, W. (1972). Reduced nearest neighbor rule. *IEEE Transactions on Information Theory 18,3*, 431–433.

Hart, P. E. (1968). The condensed nearest neighbour rule. *IEEE Transactions on Information Theory 18,5*, 515–516.

Liu, T., A. W. Moore, et A.Gray (2006). New algorithms for efficient high dimensional nonparametric classification. *Journal of Machine Learning Research 7*, 1135–1158.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys 34,1*, 1–47.

Wilson, R. et R. Martinez (1989). Reduction techniques for instance-based learning algorithms. *Machine Learning 38,3*, 257–286.

Summary

Among the panoply of classifiers used in text categorization, we are concerned with the k-nearest neighbors algorithm. Its performances allow it to be among the best text categorization methods. However, it has some limitations: (i) memory cost because it must store the entire training set for classifying a new document, and (ii) the high cost of computing because it must explore all the training set to classify a new document. In this paper, we propose a new approach to reduce the classification time without degrading the performance of classification.