Sélection de prototypes en vue d'une catégorisation de textes avec les K plus proches voisins : étude comparative

Fatiha Barigou*, Baya Naouel Barigou**
Baghdad Atmani***, Bouziane Beldjilali****

Département d'Informatique, Université d'Oran
BP 1524, El M'Naouer, Es Senia, 31 000 Oran, Algérie.
*,**,***,**** Laboratoire d'informatique d'Oran
Équipe de simulation,intégration et fouille de données (SIF)
(fatbarigou, barigounaouel, baghdad.atmani)@gmail.com, bouzianebeldjilali@yahoo.fr

Résumé. La technique des K plus proches voisins (KNN) est une méthode d'apprentissage à base d'instances, elle a été appliquée dans la catégorisation de textes depuis de nombreuses années. En contraste avec ses performances de classification, il est reconnu que cet algorithme est lent pendant la classification d'un nouveau document. Les Techniques de sélection de prototypes sont apparues comme des méthodes très compétitives pour améliorer le KNN grâce à la réduction des données. L'étude contenue dans ce papier a pour objectif d'analyser l'impact de ces méthodes sur la performance de la classification de textes avec l'algorithme KNN.

1 Introduction

En termes de performance de classification de textes, KNN se classe parmi les classifieurs les plus performants, un résultat obtenu d'une multitude de tests de comparaison effectués sur le corpus Reuters Yang (1999). En contraste avec ses performances de classification, il est reconnu que cet algorithme est lent puisqu'il requiert qu'une mesure de similarité soit calculée entre tous les documents d'apprentissage et le nouveau document. Il est caractérisé par un apprentissage très rapide, il est facile à apprendre, il est robuste aux ensembles d'apprentissage bruités et il est efficace si le corpus est grand Bhatia et Vandana (2010). Un inconvénient majeur du KNN reste le temps qu'il met pour classer un nouveau document. Différentes solutions ont été proposées pour réduire la complexité de calcul. Nous nous intéressons, dans ce papier, aux méthodes de sélection de prototypes. Plus précisément, nous étudions l'impact de différentes méthodes de sélection de prototypes sur la performance de la catégorisation de textes avec le classifieur KNN. Essentiellement, voici comment se structure la suite du papier, la section 2 présente une série de méthodes de sélection de prototypes, en décrivant leurs principales caractéristiques. La section 3 présente les différentes expérimentations effectuées sur les différents corpus de textes pour comparer les différentes méthodes de sélection de prototypes. La conclusion générale résume le travail effectué et les résultats obtenus.

2 Sélection de prototypes

Depuis la création de l'algorithme des plus proches voisins en 1967 Hart et Cover (1967), une grande variété de techniques de sélection de prototypes ont fait leur apparition pour remédier aux principaux inconvénients associés à l'algorithme et ses variations José (2002); Olvera-López et al. (2010); Garcia et al. (2012). Leur objectif principal consistait à améliorer le temps de classification du KNN.

2.1 Principe des méthodes de sélection de prototypes

Étant donné un ensemble d'apprentissage DT, l'objectif d'une méthode de sélection de prototypes (notée dans la suite SP) est d'obtenir le sous-ensemble d'instances $DS \subset DT$ tel que DS ne contient pas d'instances inutiles et lorsqu'on classe une nouvelle instance Q par la règle KNN en agissant sur DS au lieu de DT nous avons $P(DS) \simeq P(DT)$ Olvera-López et al. (2010). Selon le type de sélection, ces algorithmes peuvent être classés en trois catégories.

2.2 Algorithme de condensation

Ces algorithmes essayent de trouver une réduction significative de l'ensemble des instances de telle façon les résultats de classification avec KNN sont aussi proches que possible de ceux obtenus en utilisant tous les cas originaux. Ils cherchent les instances qui correspondent à leurs voisins les plus proches. Étant donné que ces instances fournissent les mêmes informations de classification que leurs voisins, elles peuvent être retirées sans dégrader l'exactitude de la classification des autres instances qui les entourent. Nous distinguons dans cette catégorie la méthode la plus ancienne CNN décrite par Hart (1968). La performance de l'algorithme CNN n'est pas bonne, mais elle a inspiré la construction de nouvelles méthodes telles que RNN Gates (1972), SNN Ritter et al. (1975), TCNN Tomek (1976), POP Riquelme et al. (2003) et FCNN Angiulli (2005).

2.3 Algorithme d'édition

Les algorithmes d'édition tentent de découvrir et de supprimer les instances bruitées. Les instances bruitées peuvent provoquer des erreurs de classification. Par conséquent, leur suppression devrait aider à augmenter l'exactitude de la classification. Le procédé est décrémental et une instance est éliminée si elle est mal classifiée par un vote à la majorité sur ses K plus proches voisins. C'est l'algorithme ENN de Wilson (1972). ENN permet de résoudre le problème d'instances bruitées avec une bonne performance, mais le taux de réduction reste toujours faible en le comparant à d'autres méthodes Olvera-López et al. (2010). Une autre variante de la méthode ENN est ALLKNN Tomek (1976).

2.4 Algorithmes hybrides

Ces algorithmes permettent à la fois une élimination des instances bruitées et inutiles. Aha et al. (1991) ont proposé une série d'algorithmes dont IB3 est la version la plus aboutie. Cano et al. (2003) ont présenté une étude expérimentale de différents algorithmes évolutionnaires, en fonction de leurs résultats, l'approche génétique CHC a obtenu les meilleures performances en

précision et en réduction. En outre, CHC est la méthode qui exigeait moins de temps d'exécution. La méthode Drop3 Wilson et Martinez (2002), utilise le filtrage de bruit avant de trier les instances de DS. Les objets restants sont classés par mesure de distance avec l'objet de classe différente qui est le plus proche restant dans DS, et donc les objets loin de la frontière de décision réelle sont supprimés en premier. la méthode SSMA Garcia et al. (2012) a été proposée pour couvrir un inconvénient majeur des méthodes évolutionnaires classiques : leur manque de convergence face à de grands problèmes.

3 Étude expérimentale

Notre étude se concentre sur un problème particulier, il s'agit de voir si l'utilisation de l'une des techniques de sélection de prototypes aidera à améliorer la catégorisation de textes avec les K plus proches voisins point de vue efficacité et efficience.

3.1 Corpus utilisés et mesures d'évaluation

Nous menons une étude expérimentale impliquant différentes tailles d'ensembles de documents pour mesurer la performance des méthodes de sélection de prototypes en termes de précision, de capacités de réduction et d'exécution dans le cadre de la catégorisation de textes. Les textes des différents corpus subissent un ensemble de traitements pour récupérer une représentation numérique exploitable par l'algorithme d'apprentissage. Cette représentation est appelée représentation vectorielle. Pour prédire la classe d'un nouveau document, l'algorithme cherche les k plus proches voisins de ce nouveau document en calculant la distance euclidienne et ensuite par vote majoritaire prédit la réponse la plus fréquente de ces k plus proches voisins. Nous avons calculé, dans chaque expérience, le taux de réduction (Red), l'exactitude (A), la F-mesure micro (F^{μ}) la F-mesure macro (F^{m}), le temps de réduction (TR) en secondes et le temps de classification (TC) en secondes.

3.2 Résultats et discussion

Dans le tableau 1 sont donnés les meilleurs résultats des différentes expériences réalisées avec le corpus Webk. Pour ce corpus, les quatre méthodes IB3, RNG, SSMA et ENN combinées avec KNN donnent les meilleurs résultats en termes d'Exactitude. Par contre aucune méthode de condensation n'a amélioré les résultats de KNN. En examinant le taux de réduction, nous constatons que les trois méthodes MCNN, CHC et SSMA donnent les taux de réduction les plus élevés. Comme on peut le voir sur le tableau 1, ALLKNN et FCNN donnent un taux de Fmesure (micro ou macro) le plus élevé par rapport à l'ensemble des méthodes, mais elles sont moins précises que KNN. Il est particulièrement remarquable que plus le taux de réduction augmente plus le temps de classification diminue. CHC et SSMA produisent un taux de réduction le plus élevé mais on voit bien qu'elles nécessitent un temps de réduction très élevé. En termes de rapidité de classification ce sont les approches de condensation MCNN, POP et FCNN qui sont les meilleures. Nous remarquons à partir de ces expériences et en tenant compte à la fois de l'exactitude, et du taux de réduction que la méthode SSMA est meilleur que KNN dans la plupart des cas. En d'autres termes, elle permet un meilleur compromis entre exactitude et taux de réduction mais elle reste toujours gourmande en temps de réduction, par

exemple elle nécessite environ 259 secondes pour réduire un corpus de 4199 documents indexé avec 300 termes. Dans le cas du corpus 20NewsGroups, les expériences sont effectuées avec 80 % pour l'apprentissage et les 20 % restants pour le test. En examinant le tableau 2, les résultats montrent que les deux méthodes MCNN et ENRBF qui donnent le meilleur taux de réduction par rapport aux autres, donnent aussi un meilleur compromis entre le temps de classification et le taux de réduction. Par contre les méthodes POP, IB3 et FCNN nécessitent un temps de classification plus élevé par apport à KNN. Une dernière remarque, qui est peut-être très importante, est que CNN et DROP3 sont assez lentes en temps de réduction, également SSMA et CHC qui n'ont pas participé pour ce corpus à cause de cette contrainte de temps.
Les expériences avec le corpus Reuters sont effectuées avec 80 % du corpus, l'équivalent de 9100 documents, pour l'apprentissage et 20 % pour le test. D'après les résultats présentés dans le tableau 3, POP et SSMA offrent les meilleurs résultats de classification en termes de F mesure. Aucune méthode de condensation n'a pu améliorer le 1NN. Lorsque K=10, les méthodes de condensation POP, FCNN dépassent KNN en termes de Fmesure bien que les taux de réduction restent faibles, en particulier celui de la méthode POP. On peut remarquer aussi que les méthodes les plus performantes sont les approches incrémentales de condensation. En termes d'exactitude, de réduction et du temps de classification RNG , SSMA et CHC offrent un bon taux, mais elles sont lentes pendant la réduction.

4 Conclusion

De nombreuses méthodes de SP ont été étudiées Garcia et al. (2012), mais une conclusion précise ne peut être donnée sur la meilleure méthode. Nous réalisons que le choix dépend alors du problème à résoudre, mais les résultats des différentes expériences obtenus par plusieurs chercheurs pourraient toujours nous aider à s'orienter vers certaines méthodes qu'ils considèrent comme intéressantes. En effet cette étude bibliographique et expérimentale nous a permis de découvrir plusieurs méthodes qui sont intéressantes sur le plan performance et sur le plan efficacité. En général, les meilleures méthodes de SP en termes de performance sont RNG et SSMA, mais elles ont pour principal défaut le temps de réduction qui reste élevé. Les meilleures méthodes pour la réduction sont MCNN, CHC et SSMA. Nous avons constaté que les méthodes hybrides permettent des taux de réduction élevés, tout en préservant la performance mais elles sont les plus lentes. Des méthodes plus rapides permettant d'atteindre des taux de réduction élevés sont les approches de condensation comme MCNN, mais nous constatons que cette dernière n'est pas en mesure d'améliorer le KNN en termes de précision. Certaines méthodes présentent des différences claires lorsqu'il s'agit d'un grand corpus (voir tableau 2), c'est le cas de POP, FCNN et le cas également de ENRBF qui ont amélioré le KNN en temps de classification avec un taux de réduction intéressant. Les autres méthodes comme AllKNN, IB3, ENN ont pu améliorer KNN qu'avec le petit corpus WebK.

Méthode	K	$\|T\|$	A(%)	Red (%)	F^{μ} (%)	F^{m}(%)	TC	TR
ALLKNN	5	300	74.668	33.635	74.548	73.246	0.86	26.177
CHC	1	300	75.271	99.096	58.543	72.866	0	1099.887
CNN	21	300	74.5476	56.359	61.987	72.673	0.69	99.441
DROP3	1	300	70.929	79.234	70.929	69.346	0.234	48.141
ENN	1	300	75.874	23,297	59.861	74.532	0.98	14.24
ENRBF	1	900	39.423	59.182	52.607	37.988	1.326	35.444
FCNN	5	300	74.427	61.061	74.427	72.318	0.47	8.549
IB3	21	400	78.072	68.383	62.868	75.746	0.53	8.7
MCNN	1	300	69.722	99.156	58.351	69.070	0.015	4.883
MENN	1	400	70	50.075	60.008	69.057	0.8	18.762
POP	10	300	74.186	36.949	56.141	74.203	0.827	7.987
RNG	5	300	78.166	22.513	58.502	75.584	0.957	348.678
SSMA	1	300	79.735	97.498	58.824	77.807	0.016	259.196
KNN	1	300	75.151	0	75.151	72.634	2.01	0
KNN	5	300	78.528	0	76.839	76.645	2.118	0
KNN	21	300	75.151	0	76.447	74.408	2.266	0
KNN	1	900	71.755	0	74.988	69.648	5.266	0
KNN	21	400	74.217	0	75.814	72.700	2.694	0
KNN	1	400	73.976	0	75.953	71.386	2.668	0
KNN	10	300	76.960	0	76.879	74.969	1.973	0

TAB. 1 – *Résultats obtenus avec le corpus WebK*

Méthode	A(%)	Red (%)	F^{μ}(%)	F^{m}(%)	TC	TR
ALLKNN	51.026	55.585	48.272	53.429	14.844	314.838
CNN	52.918	39.153	47.057	53.469	20.793	6360.01
DROP3	48.388	74.908	48.388	49.474	14.922	1089.61
IB3	52.092	45.487	37.211	52.487	32.554	167.722
MCNN	27.791	98.774	41.154	30.398	0.32	85.991
MENN	46.789	70.652	37.451	50.508	15.534	320.454
POP	50.999	29.767	35.577	52.326	33.695	61.319
FCNN	52.864	41.464	52.864	53.393	33.957	307.184
ENRBF	13.002	90.521	32.934	21.309	5.314	361.38
ENN	52.144	47.239	43.183	54.186	24.532	236.005
KNN	56.515	0	69.866	56.986	31.84	0

TAB. 2 – *Résultats obtenus avec le corpus 20NewsGroups avec un vocabulaire égal à 400 termes*

Méthode	K	A(%)	Red (%)	F^{μ}(%)	F^{m}(%)	TC	TR
ALLKNN	1	86.823	14.966	54.136	63.963	4.847	109.415
CHC	1	87.736	99.364	70.277	64.436	0.032	9141.28
CNN	1	86.301	79.997	68.825	67.345	1.207	220.141
DROP3	1	78.865	90.006	71.286	61.290	0.468	293.717
ENN	1	87.149	11.689	70.200	63.783	5.56	46.981
ENRBP	1	51.402	48.614	71.179	18.138	2.75	45.122
FCNN	1	86.171	80.437	68.972	69.565	1.03	19.994
IB3	1	86.040	84.708	71.002	67.096	0.974	13.808
MCNN	1	83.822	98.712	71.080	64.629	0.05	20.912
MENN	1	85.258	20.134	68.483	59.284	4.51	65.504
POP	10	87.215	55.298	70.073	71.235	2.782	14.666
RNG	1	89.237	97.163	70.759	66.889	5.2	3903.89
SSMA	1	90.085	78.794	71.776	74.889	0.04	647.473
KNN	1	89.171	0	69.349	71.713	5.34	0
KNN	10	87.867	0	69.606	65.451	5.71	0

TAB. 3 – *Résultats obtenus avec le corpus Reuters avec un vocabulaire égal à 400*

Références

Aha, D. W., D. Kibler, et M. K. Albert (1991). Instance-based learning algorithms. *Machine Learning*, 37–66.

Angiulli, F. (2005). Fast condensed nearest neighbor rule. In *22d International Conference on Machine Learning*, Bonn, Germany, pp. 25–32.

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

Cano, J., F. Herrera, et M. Lozano. (2003). Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study. *IEEE Transactions on Evolutionary Computation 7*(6), 561–575.

Garcia, S., J. Derrac, et J. Cano (2012). Prototype selection for nearest neighbor. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*(3), 417–435.

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

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

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

José, V. (2002). *Ship Noise Classification*. Ph. D. thesis, University of Lisbon, College of Science and Technology.

Olvera-López, J., J. Carrasco-Ochoa, J. Martínez-Trinidad, et J. Kittler (2010). A review of instance selection methods. *Artificial Intelligence Review 34*(issue 2), 133–143.

Riquelme, J., J. Aguilar-Ruiz, et M. Toro (2003). Finding representative patterns with ordered projections. *Pattern Recognition 36*(4), 93–102.

Ritter, G. L., H. B. Woodruff, S. R. Lowry, et T. L. Isenhour (1975). An algorithm for a selective nearest neighbor decision rule. *IEEE Transactions on Information Theory 21*(6), 665–669.

Tomek, I. (1976). Two modifications of cnn. *IEEE Transactions on systems, Man, and Cybernetics 6*(6), 769–772.

Wilson, D. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics 2*, 408–421.

Wilson, D. et T. Martinez (2002). Reduction techniques for instance-based learning algorithms. *Machine Learning 38*, 257–286.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information retrieval 1-2, 1*, 67–88.

Summary

KNN is a learning method based on instances; it has been applied to text categorization for many years. In contrast to its classification performance, it is recognized that this algorithm is running slow for the classification of a new document. Prototype selection methods have emerged as highly competitive methods for improving KNN with data reduction. The study contained in this paper aims at analyzing the impact of these methods on performances of text classification with KNN algorithm.