J Inf Process Syst, Vol.14, No.2, pp.418~434, April 2018
https://doi.org/10.3745/JIPS.02.0080

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Impact of Instance Selection on kNN-Based Text Categorization

Fatiha Barigou*

Abstract

With the increasing use of the Internet and electronic documents, automatic text categorization becomes imperative. Several machine learning algorithms have been proposed for text categorization. The k-nearest neighbor algorithm (kNN) is known to be one of the best state of the art classifiers when used for text categorization. However, kNN suffers from limitations such as high computation when classifying new instances. Instance selection techniques have emerged as highly competitive methods to improve kNN through data reduction. However previous works have evaluated those approaches only on structured datasets. In addition, their performance has not been examined over the text categorization domain where the dimensionality and size of the dataset is very high. Motivated by these observations, this paper investigates and analyzes the impact of instance selection on kNN-based text categorization in terms of various aspects such as classification accuracy, classification efficiency, and data reduction.

Keywords

Classification Accuracy, Classification Efficiency, Data Reduction, Instance Selection, k-Nearest Neighbors, Text Categorization

1. Introduction

Text categorization (or text classification, TC) can be briefly defined as the task of assigning predefined categories to text documents. It is an important component in many information organization and management tasks as well: text retrieval, filtering, sorting and topic identification [1].

A wide variety of machine learning and statistical classification techniques have been applied to text categorization, among them, k-nearest neighbor algorithm (kNN) has shown great potential. The kNN technique is a very simple and powerful instance-based learning algorithm. Despite its simplicity, it can offer very good performance that is why it is one of the most extensively used nonparametric classification algorithms in TC [2].

The kNN algorithm differs from other learning methods because all computation is deferred until classification and no model is induced from the learning examples. The data remains as they are; they are simply stored in memory. To decide the class of a new sample, the algorithm computes similarities (or distances) between this sample and all training samples in order to look for the kNN of the new

Manuscript received August 18, 2014; first revision March 8, 2016; second revision November 18, 2016; accepted November 30, 2016.
Corresponding Author: Fatiha Barigou (fatbarigou@gmail.com)
* Dept. of Computer Science, University of Oran 1 Ahmed Ben Bella, Oran, Algeria (fatbarigou@gmail.com)

sample and predicts the most frequent class of those kNNs. kNN algorithm uses all training data for classification; hence, it requires a high storage memory and a high degree of calculation complexity.

In order to tackle these drawbacks, several improvements are proposed over the years to reduce the number of distance (or similarity) calculation actually performed. They aimed to improve the time performance by accelerating the classification process while keeping the error rate as low as possible. We distinguish between two types of accelerating strategies; instance selection methods [3-5] and computation time reduction methods [6]. The first one aimed at reducing the number of examples by filtering out data that are nonessential from a given training dataset. The second one accelerated the search operation during classification by setting well-organized structures of the training set such as kd-trees, ball trees or hashing functions. Although they often yield impressive speed ups, these methods still store the entire training set and their performance tends to deteriorate with increasing data dimensionality. In this paper, we focus on instance selection methods to reduce kNN-based text categorization computation.

In the most of the previous works, the focus was mainly on the instance selection from structured data sets where the dimensionalities and the size are very low [3]. However, as text categorization has become one of the major techniques for managing large volume of text document, very little research focus on instance selection for TC [7]. Furthermore, the performance of TC systems when performing instance selection for kNN algorithm has not been fully examined. On the other hand, numerous studies are particularly interested with feature weighting [8] and feature selection [9] to improve text categorization.

Therefore, this work investigates and analyzes the impact of using instance selection as reduction methods on the performance of kNN-based text categorization. This is the great question, which we try to answer in this work; does instance selection offers improvement on efficiency of kNN-based text categorization without degrading its classification accuracy?

The rest of paper is organized as follow. Section 2 provides an overview of related literatures, including the concept of instance selection and a set of instance selection approaches, highlighting their main characteristics. To compare the effect of those approaches on kNN-based text categorization, Section 3 presents the different experiments carried on three different text datasets followed by a discussion of the results obtained. Finally, the last section summarizes the work done.

2. Instance Selection

In order to deal with the problems that arise with the use of the nearest neighbor in classification, numerous methods have been developed and proposed in the literature to reduce the number of training data and simultaneously keep the error rate as low as possible.

As Garcia et al. [3] explain in their article, when dealing with the design of training set reduction algorithm, we have to decide or choose between two competing solutions. The first one, called instance selection [3-5,10-12], selects a small representative subset of the initial training set. The second one known as prototype generation, generates a new set of prototypes to replace the initial ones [13].

In this paper we focus only on instance selection methods that we encountered during our literature review. We start this section by defining the concept of instance selection. Then we give an overview of the most representative instance selection algorithms developed to date.

The instance selection is an area of research that has been active more than four decades. Since the creation of the kNN algorithm in 1967 [14], a wide variety of instance selection approaches have emerged to address the main drawbacks associated with the algorithm and its variations. Their main objective was to improve the time classification of kNN by reducing the size of the training dataset using an intelligent selection of instances that must be maintained as learning instances.

2.1 The Principle of Instance Selection

Instance selection aims at obtaining a representative subset of the initial training dataset capable of achieving, at least, the same performance of the whole training set.

Considering the task of text categorization with kNN algorithm, a formal definition of instance selection can be the following: if D_T is the training set consisting of pairs $<x_i, y>$ i = 1..n where x_i defines input vector of features of document d_i and y defines its corresponding class label, then the objective of instance selection (IS) is getting a reduced subset of instances $D_S \subseteq D_T$ such that $|D_S| << |D_T|$ and D_S does not contain useless instances and when classifying a new text by the kNN algorithm using D_S dataset instead of D_T dataset, performance classification is as good as if it has used D_T dataset [5].

The instance selection methods attempt to preserve the character of the original data by deleting data that are nonessential; they are designed to obtain a training set which is representative and with a smaller size than the original one. Their main objective is to reduce the classification time without degrading the performance of classification. Depending on the strategy followed by these methods, they can remove noise, redundancy or both.

2.2 Instance Selection Approaches

The IS problem has been addressed by many authors with different approaches. Garcia et al. [3] gave a complete review of various IS methods and conducted an experimental study comparing 50 related instance selection algorithms using structured datasets from UCI Machine Learning Repository. Instance selection has also been applied to noise detection in gene expression classification data [15], regression [16] and time series prediction [17].

In this paper, we review only some of well-known instance selection algorithms. According to type of selection those algorithms can be divided into four groups: condensation algorithms, edition algorithms, hybrid algorithms, and meta-heuristic algorithms.

Condensation algorithms

The idea of these algorithms is to remove redundant instances, thus reducing the size of the data set and search complexity. They try to extract a consistent subset of the overall training set in such a way the classification results with kNN are as close as possible to those obtained using the whole dataset. Through the nearest neighbours rule they look for instances that match their closest neighbours. Because those instances provide the same classification information than their neighbors, they can be removed without degrading the accuracy of the classification of other instances that surround them. Decisions taken by condensation algorithms are not robust, i.e. they preserve the noise.

In this category, the condensed nearest neighbor (CNN) algorithm is the oldest condensation method described by [18]. An incremental search is used by this algorithm; it begins with an empty subset D_S

and one instance per class is chosen randomly from D_T and inserted in D_S. Then and incrementally adds each instance in D_T to subset D_S if it fulfils the following criteria: any instance misclassified by its nearest neighbours among the prototypes that are already selected is immediately stored. This incremental process is iterated until there are no more misclassified instances. The performance of the CNN algorithm is sensitive to the noise; noisy instances will usually be misclassified by the CNN and thus will be retained. This causes more misclassification than before reduction. CNN has inspired the development of new methods such as reduced nearest neighbor (RNN) [19], selective nearest neighbor (SNN) [20], Tomek condensation nearest neighbor (TCNN) [21], modified CNN (MCNN) [22], pattern by ordered projections (POP) [23] and fast CNN (FCNN) [24].

The RNN algorithm is a modification of CNN introduced by [19]. It starts with $D_S = D_T$ and removes each instance from DS if such removal does not cause any other instances in D_T to be misclassified by the instances remaining in D_S. Since the instance being removed is not guaranteed to be classified correctly, this algorithm is able to remove noisy instances and internal instances while retaining border points. Experiments have shown that this rule yields a slightly smaller subset than the CNN technique, but it is costly.

The SNN algorithm [20] computes the smallest and consistent training set D_S of D_T having the following additional property: each point of D_T is closer to a point of D_S of the same class than to any other point of D_T of a different class. SNN runs in exponential time and, hence, it is not suitable on large training sets.

The POP method [23] is the heuristic approach for finding representative patterns. The main idea of the algorithm is to select only some border instances without calculating distance and eliminate the examples that are not in the boundaries of the regions to which they belong.

Recently, FCNN [24] makes CNN sub-quadratic to train (as opposed to $O(n^3)$ for CNN), with empirically better test generalization accuracy. It works as follows. First, the consistent subset D_S is initialized to the centroids of the classes contained in the training set D_T. Then, during each iteration, for each point p in D_S, a corresponding point q of D_T belonging to the neighbors of point p but having a different class label is selected and added to D_S. The algorithm stops when no further points can be added to D_S, that is, when D_T is correctly classified using D_S.

Edition algorithms

This family follows a strategy which is opposite to condensation; it discards the instances that are harmful to the classification accuracy. This kind of method is mostly used as noise filters and realizes small reductions. The process is decremental; an instance is removed if it is misclassified by a majority vote of its k nearest neighbours. The first and most known edition method was edited nearest neighbor (ENN) algorithm [25]. ENN is based on the following idea: if an instance is misclassified with kNN rule, it must be removed. ENN starts from the initial training data set ($D_S = D_T$), then at each iteration, an instance of D_S is removed if it is not in agreement with the majority of its kNNs. Hence, ENN is an iterative algorithm and the final subset contains only instances that are correctly classified by their kNN. As a result, noisy instances are removed resulting to the improvement of the classification accuracy, but the reduction rate remains low by comparison with other methods [5].

Repeated ENN (RENN) was also proposed by Wilson [25]. The only difference is that the process of ENN is repeated as long as any changes are made in the selected set.

Another variant of the ENN method called ALLKNN is proposed by Tomek [21]; the ENN is

repeated for all k (k = 1, 2, …, l). MENN [26] is a similar algorithm to ENN but in addition it works with a prefixed number of pairs (k, k') where k is employed as the number of neighbors employed to perform the editing process, and k' is employed to validate the edited set DS obtained.

One of the most effective editing techniques is the relative neighborhood graph (RNG) [27] method. The general idea is that after construction of a proximity graph, instances misclassified by their neighbours in this graph are removed.

Few edition algorithms have been proposed in comparison to the other families. The main reason is that the first edition method, ENN, obtains good results in conjunction with kNN and the other edition approaches do not achieve high reduction rates, which is main goal of interest in IS.

Hybrid algorithms

After more than two decades of editing and condensing algorithms, a new trend known as hybrid algorithm appeared as a highly competitive performances combining condensed and edition approaches to remove noisy and redundant instances. They try to find the smallest subset DS which increases the classification performance with a significant reduction rate.

Aha et al. [28] proposed a series of algorithms including Instance Based-3 (IB3) that is the most complete version. IB3 was the first hybrid method which combines an edition stage with a condensation one. IB3 is an incremental algorithm that uses a classification score to determine which instances to preserve.

Randall Wilson and Martinez [29] presented a series of subtractive algorithms called Decremental Reduction Optimization Procedure (DROP1–5). DROP1 is the basic reduction model, while DROP2–5 are expansions that enhance the performance of the algorithm via noise filters and other extensions. But, the most efficient of the algorithms is DROP3, which best addresses the problem of noisy instances. Compared to DROP2 a filter is added as a pre-processing step to remove samples that are misclassified by their k nearest neighbours. Brighton and Mellish [30] conducted some comparative experiments. They found that DROP3 makes the kNN classifier providing better performances over other instance selection methods.

Evolutionary algorithms

Considering instance selection as a search problem, genetic algorithms have been widely used for this task in the last two decades. Among them, evolutionary algorithms (EA) stand out [3]. A complete survey of them can be found in [31]. An evolutionary algorithm begins with a set of randomly generated solutions called a population, then, new solutions are obtained by the combination of two existing solutions; crossover operator and mutation operator. All the individuals are then evaluated assigning to each one a value called fitness, which measures its ability to solve the problem. After this process, the best individuals, in terms of higher fitness, are selected and an evolution cycle is completed. This cycle is termed a generation.

The main weakness of those approaches is the high computational cost that puts them at a disadvantage compared to other approaches when they come to practical application [32]. However, when compared to non-EAs, which have a short execution time, EA-based algorithms offer more reduction without over fitting.

Cano et al. [33] performed an experimental study of different evolutionary algorithms. Based on their results, the adaptive search algorithm CHC got the best performance in accuracy and reduction with

less time classification.

Another evolutionary method for instance selection called steady-state memetic algorithm (SSMA) has been presented in [34] to cover a disadvantage of evolutionary methods; their lack of convergence facing big applications. The algorithm integrates global and local searches, as it uses adaptation concepts to produce a training set, and later employs a mimetic optimization to obtain the final population of instances. This method, contrasted to the CHC algorithm [33], produces better accuracy results.

Discussion

The IS problem has been addressed by many authors with different approaches. However; any definite conclusion can be given on the best method. Experimental analysis on selection techniques has shown that no ideal method exists. Especially, Garcia et al. [3] realize that the choice then depends on the problem at hand. Nevertheless, the results of different experiments obtained by several researchers could always help us to move towards some methods which they consider interesting.

Indeed, this literature review allowed us to discover several methods that are interesting point of view performance and efficiency. As shown in Table 1 we have the following findings:

- Among the condensation methods, authors in [24] consider FCNN algorithm as a powerful technique and one of the fastest approaches. But its accuracy is sensitive to noise. On the other hand the best reduction is achieved by RNN but its reduction is time consuming.
- According to [32], in the editing methods family, the performance of ENN and its low computational costs make it the preferred edition method for most authors.
- The hybrid methods combine noise filters with condensation to overcome problems of the editing and condensing strategies. According to [29] experimental results showed that DROP3 had a higher average accuracy than IB3, and had the best mix of storage reduction and generalization accuracy
- As representatives of the meta-heuristic family, we noted that the SSMA and CHC methods offer an excellent compromise between the reduction rate and classification performance but their runtimes execution is very high.

It can be noticed that the most interesting methods in terms of effectiveness are EAs. In fact, according to Derrac et al. [31], EAs can improve the performance of data mining algorithms. In particular, Cano et al. [33] have shown, trough an experimental study, that evolutionary algorithm can obtain better results than many non evolutionary instance selection methods in terms of better instance selection rates and higher classification accuracy. The main limitation of those methods is their computational complexity, due to the evolution process involved [33].

However, currently, data sets sizes have grown considerably which means that they are not suitable for large quantities of data. Based on these limitation, several solutions have been proposed to deal with massive data challenge.

Cano et al. [35] proposed stratification for large problems. The original data set is divided into smaller subsets of instances then a CHC evolutionary algorithm is applied to each subset. According to [36], the algorithm shows good performance, but it is still too computationally expensive for huge datasets. Another interesting work that face the challenge of applying EAs to large data sets concerns the parallelization of the task of instance selection. This idea is recently proposed by Triguero et al. [37] where they developed a map reduce approach for IS algorithms.

Table 1. Instance selection approaches

Family	Algorithm	Strength	Weakness
Condensation	CNN [18]	Interesting reduction rate, memory cost and time classification reduced.	Will not find a minimal consistent subset. Very fragile in respect to noise and depends on the order of arrival of instances.
	RNN [19]	Significantly reduces the size of the training set by removing redundant instances.	Cost of reduction remains high and does not guarantee a minimal output set.
	SNN [20]	Computes the smallest and consistent training set.	Runs in exponential time; it is not suitable on large training sets.
	FCNN [24]	It is order independent, requires few iterations to converge scales well on large-sized multi-dimensional data sets. Discards redundant and harmful instances; the size of the training set and time of classification is then reduced.	Sensitive to the selection of points that are very close to the decision border.
	POP [23]	A considerable reduction of training data. No need for distance computation.	Works independently within each dimension.
Edition	ENN [25]	Low computational costs Good performance Edits out noisy instances	The rate of reduction is not very significant.
	RNG [27]	The relative neighbourhood graph can be computed in linear time.	Decision-boundary changes are often drastic, and not guaranteed to be training set consistent.
	AllKNN [21]	Serves more as noise filter. Better reduction and high accuracy than ENN.	Can leave internal instances intact, thus limiting the amount of reduction.
Hybrid	DROP3 [29]	Best mix of storage reduction and generalization accuracy.	It can in rare cases remove too many instances in the noise reduction-pass which lead to decrease performance.
	IB3 [28]	Offers noise tolerance and high reduction rates.	Does not work well with big data sets and several irrelevant attributes.
Evolutionary	CHC [33]	Better accuracy and high reduction rates.	Lot slower
	SSMA [34]	Improved convergence for large problems.	Very slow during reduction

3. Experimental Study

In this experimental study we focus on a particular problem, we evaluate the impact of instance selection techniques on effectiveness and efficiency of kNN-based text categorization. We will compare the performance of kNN algorithm using the whole training data set and its performance when using a selected subset of training data set obtained with instance selection methods. All instance selection methods used in this study are collected from KEEL software (http://www.keel.es) and summarized in Table 2.

All of our experiments were performed on an Intell Pentium Dual CPU T2330 @1.6 GHZ with 2.00 GO memory.

Table 2. Overview of the algorithms evaluated in the experimental study

	Description
CNN [18]	Condensed method
MCNN [22]	Condensed method
FCNN [24]	Condensed method
POP [23]	Condensed method
ENN [25]	Noise filter
MENN [26]	Noise filter
AllkNN [21]	Noise filter
RNG [27]	Noise filter
DROP3 [29]	Hybrid method (noise filter and condensation)
IB3 [28]	Hybrid method (noise filter and condensation)
CHC [33]	Evolutionary based wrapper method
SSMA [34]	Evolutionary based wrapper method

3.1 Data Collection

We conduct an experimental study involving three documents data sets to evaluate kNN classification performance with instance selection.

Table 3. The description of text datasets

	Number of categories	Number of documents
WebKB	4	4,199
Reuters-52	52	9,100
20-Newsgroups	20	18,828

As shown in Table 3, we used as benchmark dataset three widely-used corpora obtained from the web site http://web.ist.utl.pt/acardoso/datasets/.

The first data set is the well-known WebKB corpus; it consists of 4,199 documents belonging to four categories. Reuters-52 version is the second data set; it consists of 9,100 documents belonging to 52 categories. Finally, the 20-Newsgroups is the third corpus used in our experiments. It contains about 18,828 documents uniformly divided in 20 categories.

One noticeable issue of the Reuters and WebKB corpora is the skewed category distribution problem. The most common category in the Reuters corpus is the earn category; it account for 43% of the whole set. Similarly, the most common category in WebKB corpus is student category, it accounts for 39.1% of the whole set.

We used the KEEL tool to randomly divide the corpus into 80% for training dataset and 20% for testing dataset.

3.2 Performance Measures

To assess the impact of instance selection on kNN-based text categorization system, we measure performance in terms of accuracy, reduction rate, classification time, and reduction time. We have recorded reduction time with reduction rate achieved on the training set and accuracy rate with time of

kNN classification achieved on the test data set.

We used the following measures:

- Accuracy: it counts the number of correct classifications regarding the total number of instances classified.
- Reduction rate ($|D_S|$) : it measures (in percent) the reduction rate achieved by a IS algorithm:
- Time of reduction: We calculate the total time spent by IS to generate the D_S subset from the D_T dataset
- Time of classification: we calculate the time needed to classify all the instances of test dataset regarding the reduced training set D_S.

3.3 Preprocessing

The first step in the process of constructing a classifier is to produce from training documents a format appropriate for the classification algorithms, Vector Space Model. We establish an initial list of terms by performing a segmentation of text into words, eliminate stop words using a pre-defined stop list and use the Porter algorithm [38] to perform stemming of the different retained words. Each document "d_i" is represented by the characteristic vector (w_{i1}, w_{i2}, ...w_{iM}) where "w_{ij}" is the weight of term "t_j" in the document "d_i" and "M" is the number of unique terms obtained after feature selection using information gain method [39]. With this measure, we have selected 400 terms.

Binary weighting is also used; the method checks if a particular term appears in the document. The values in the characteristic vector of document "d_i" can be 0 or 1. If the term appears in the document, then the weight value "w_{ij}" is set to 1, otherwise is set to 0.

To predict the class of a new document "q", the algorithm searches for its kNN by calculating the Euclidean distance (eq.1) with all training documents $D=\{d_i; i=1,N\}$ and then by majority vote (Eq. (2)) predicts the most common response of those kNNs.

$$D_e(d_i,q)=\sqrt{\sum_{k=1}^{M}\left(w_k(d_i)-w_k(q)\right)^2} \tag{1}$$

where $w_k(d_i)$ is the weight of the term t_k in document d_i, $w_k(q)$ is the weight of the term t_k in new document "q".

$$class(q)=\underset{c_j\in C}{ArgMax}\left(\sum_{d_i\in kNN} D_e(d_i,q)\times y(d_i,c_j)\right)$$
$$where\ y(d_i,c_j)=\begin{cases}1 & d_i \text{ is of class } c_j\\ 0 & \text{otherwise}\end{cases} \quad \text{and} \quad C=\{c_1,c_2,\cdots c_{|C|}\} \text{ is the set of categories} \tag{2}$$

3.4 Experimental Results

Figs. 1–6 show respectively information about accuracy and time of classification on the unseen documents (20%) according to the compression provided by the instance selection algorithms. The figures correspond to kNN classification (k = 3) combined with several instance selection algorithms.

The horizontal axis of Figs. 1–6 shows the compression of the training set in percent (100 = the whole training set is used). The vertical axis corresponds to accuracy changes (Figs. 1–3) and Time of classification changes (Figs. 4–6) on the test data set of WebKB, Reuters, and 20-Newsgroups corpuses,

respectively for a given instance selection algorithm.

Figs. 7–9 give information about time elapsed in seconds to complete a run of an instance selection method when used to compress the training WebKB, Reuters, and 20-Newsgroups corpuses, respectively.

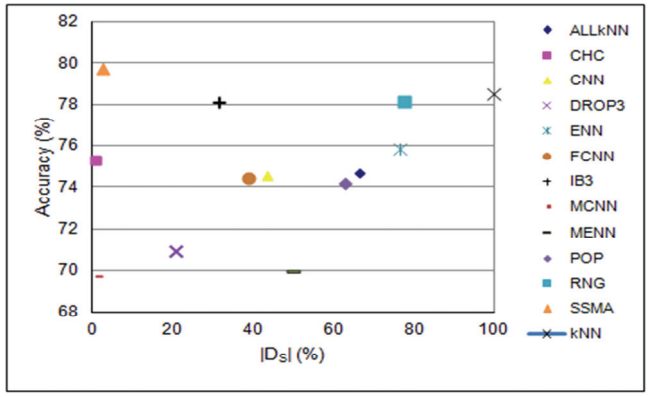

Fig. 1. Variation of accuracy according to the rate reduction obtained by the instance selection algorithms over WebKB data set.

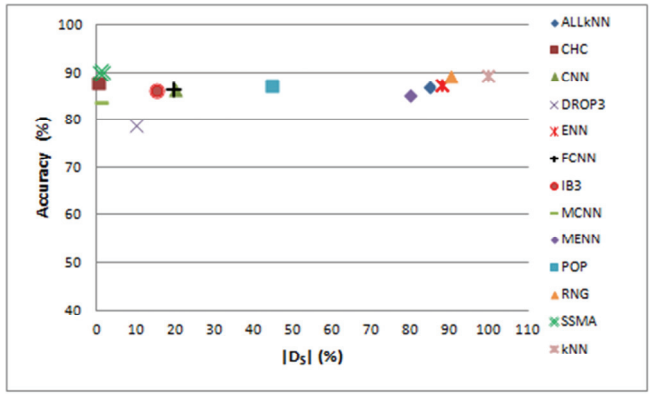

Fig. 2. Variation of accuracy according to the rate reduction obtained by the instance selection algorithms over Reuters data set.

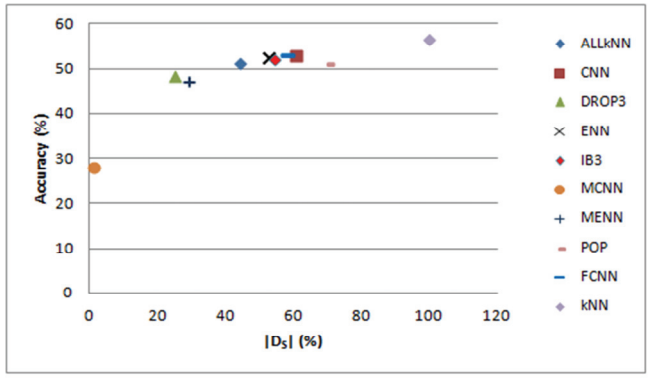

Fig. 3. Variation of accuracy according to the rate reduction obtained by the instance selection algorithms over 20-Newsgroups data set.

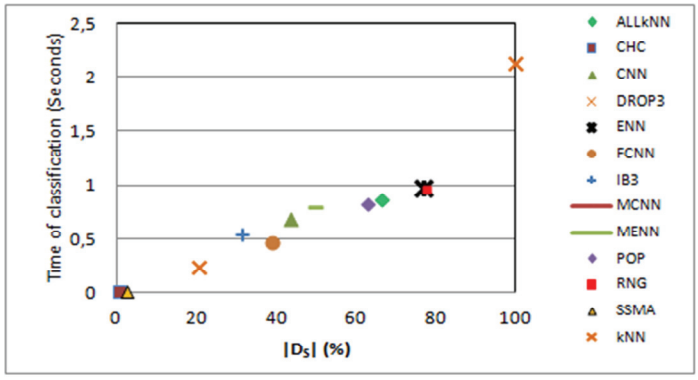

Fig. 4. Variation of time classification according to the reduction rate obtained by the instance selection algorithms over the WebKB corpus.

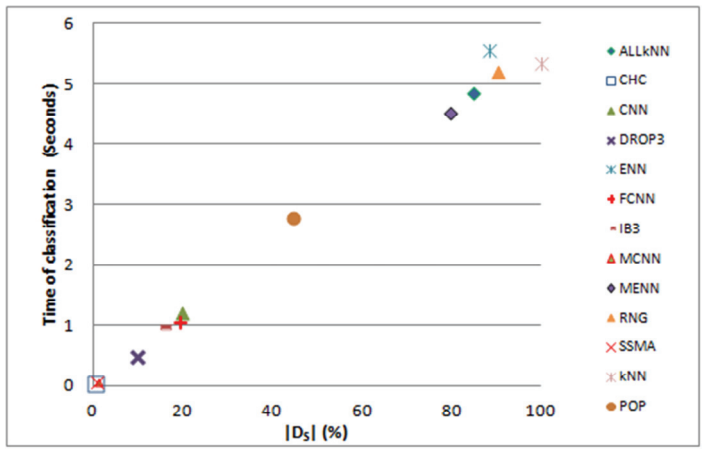

Fig. 5. Variation of time classification according to the reduction rate obtained by the instance selection algorithms over the Reuters corpus.

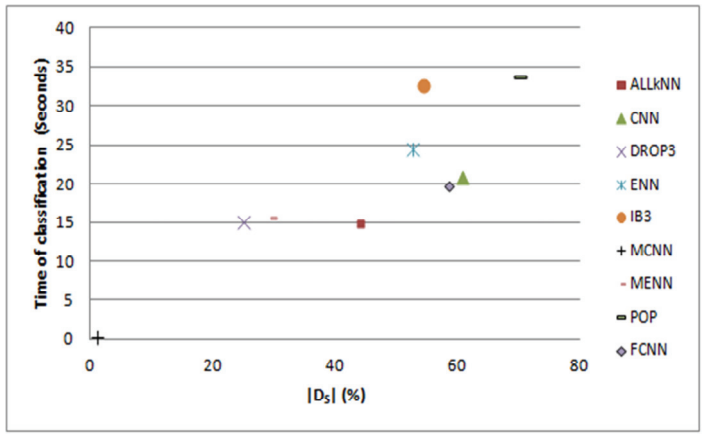

Fig. 6. Variation of time classification according to the reduction rate obtained by the instance selection algorithms over the 20-Newsgroups corpus.

Through these figures we have interesting results; reducing instances yields the best trade-off between accuracy and time of classification. Nevertheless, we notice a small loss in accuracy when using 20-Newsgroups because the latter presents class overlapping.

The first significant result of this study (see Figs. 1 and 2) is that SSMA and RNG combined with kNN give the best accuracy. When using WebKB corpus, the accuracy of kNN with SSMA is 79.7% and with RNG is 78.17%. The base accuracy of kNN is 78.5%. When using Reuters corpus the accuracy of kNN with SSMA is 90.09% and with RNG is 89.24%. The base accuracy of kNN is 89.17%. It can be noticed that although the dataset is not balanced, the results improved.

Looking at Figs. 1 and 2, it can be noticed that the MCNN, the CHC and the SSMA algorithms give the highest reduction rate. The training set of WebKB left with CHC was 0.90%, with MCNN was 0.85% and with SSMA was 2.5%. The training set of Reuters left with CHC was 0.64%, with SSMA was 1.21% and with MCNN was 1.29%.

Looking at Fig. 3, we noted that the property of data corpus has a great impact on instance selection. Unlike WebKB and Reuters-52 corpora, small performance degradation is noted with 20-Newsgroups. In the absence of SSMA and CHC methods in this experiment, we find that the performance of condensation methods like CNN and FCNN is superior to that of other instance selection methods. But, if we look also at time of classification in Fig. 6 and time of reduction in Fig. 9, we see that IB3 yields the best trade-off between performance (accuracy and time of classification) and time of reduction.

It is particularly remarkable in Figs. 4–6 that the more the rate of reduction increases more time classification decreases. CHC, SSMA, and MCNN allow the highest reduction rate. They are followed by IB3, DROP3, and FCNN. On the other hand, as can be seen in Figs. 7 and 8 there is a great difference between the time costs of the meta-heuristic approaches and the other approaches. SSMA and CHC offer the worst time of reduction. In terms of speed, these are the condensation approaches MCNN, POP, FCNN and the hybrid approach IB3 which are the best.

In Fig. 9, it can be observed that the time of reduction increases with the 20-Newsgroups corpus. Instance selection algorithms are affected when the size of data set increases. For example, the CNN algorithm represents a greater cost of time reduction.

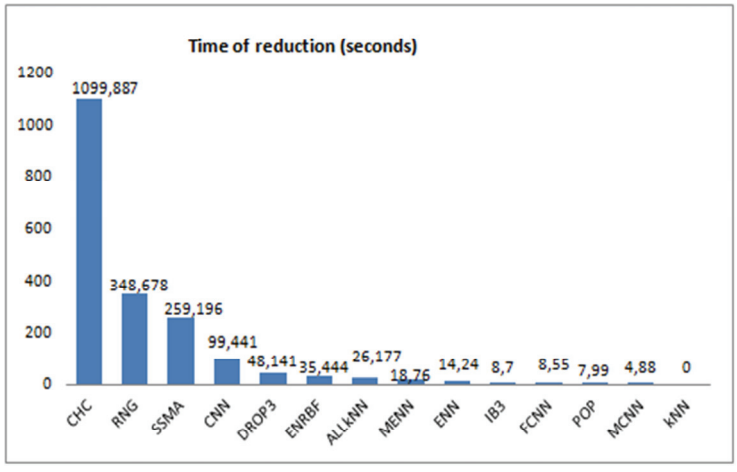

Fig. 7. Time of reduction obtained over the WebKB corpus.

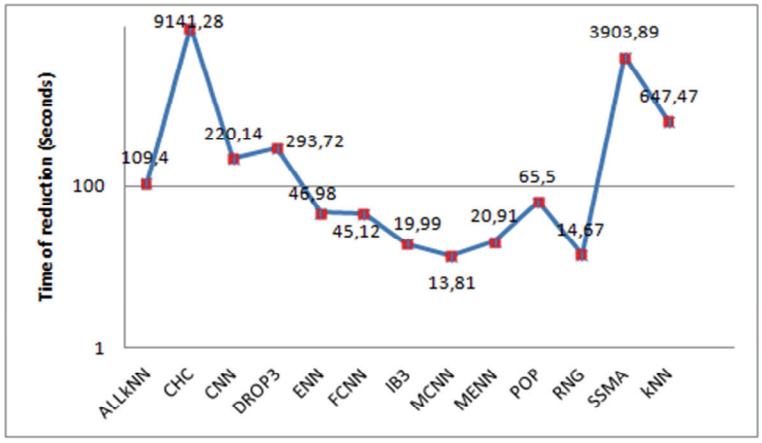

Fig. 8. Time of reduction obtained over the Reuters corpus.

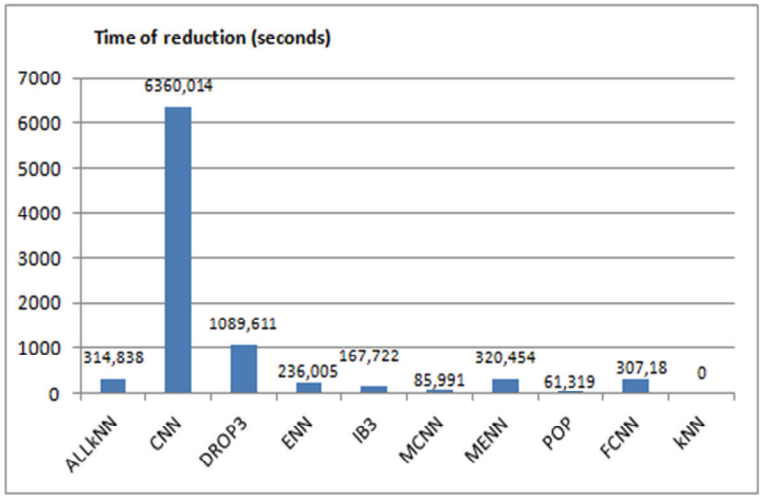

Fig. 9. Time of reduction obtained over the 20-Newsgroups corpus.

3.5 Discussion

We note from these experiences and taking into account the accuracy, time of classification and the rate of reduction that instance selection applied to training documents provides a compromise between accuracy and time of classification. This result hints at the fact that kNN-based text categorization can benefits from instance selection.

The results show also some effects related to the property of data corpus. The latter has a great impact on instance selection. We see a better performance of classification when using instance selection with WebKB and Reuters-50 data corpora however we notice a small loss in accuracy with 20-Newsgroups because this latter represents class overlapping.

The SSMA EA is the best method; it allows a better compromise between accuracy and reduction rate but it is still time consuming during reduction, for example, it takes about 259 seconds (more than 4 minutes) to reduce a corpus composed of 3,359 documents (80% of the WebKB corpus) with a

vocabulary of 400 words.

We conclude that instance selection methods, which enable high reduction, while maintaining performance are generally slower during the selection stage. Those methods belong to the category of meta-heuristic. Faster methods that achieve a good reduction ratio are condensation approaches such as POP, FCNN and MCNN and the hybrid method IB3.

Therefore, to summarize this study, we can point out the best performing instance selection methods:

- The best condensation method in terms of efficacy is FCNN and.
- Within the edition family, RNG achieves the best accuracy but it is slower.
- Considering the hybrid family, IB3 is better than DROP3 in terms of efficacy but the last one allows a better reduction rate.
- The global best methods in terms of accuracy are SSMA, RNG, CHC, IB3, and FCNN.
- Considering trade of accuracy, time of classification and reduction rate, the global best methods are SSMA and CHC.
- Considering time of reduction, the global best methods are IB3, POP, FCNN, MCNN, and ENN.

4. Conclusion

This paper addressed the analysis of the instance selection algorithms and their use in data reduction in text categorization with kNN classifier.

kNN is a simple and widely used machine learning method in the field of text categorization. Behind its simplicity, however, kNN algorithm is time consuming. There have been efforts to minimize the running time. Here we are interested by instance selection approaches to reduce the running time.

Instance selection in large data sets are important algorithms needed by a variety of high performance computing applications, for example, text categorization for information retrieval, spam filtering, and text mining of large corpuses.

In this paper, we review a set of methods for selecting instances and study their impact on kNN-based text categorization.

Experiments show that improvements can be made by using instance selection; the results show that with higher dimensional data, in most cases instance selection methods combined with kNN perform better than directly using kNN.

An experimental study was carried out to compare the results of various instance selection algorithms over three text data sets. The main conclusions reached are as follows.

- Evolutionary algorithms outperform the classical algorithms (condensation, edition and hybrid), simultaneously offering two main advantages: better data reduction percentages and higher classification accuracy.
- The CHC and SSMA are the most appropriate EA, according to the algorithms that we have compared. They offer good performance in both accuracy and reduction; however, their main disadvantage is that they spent during reduction more runtime than the others approaches. The problem then is runtimes grows when very large datasets are processed like in text categorization.

Therefore, as a final concluding remark, we consider EAs to be a good mechanism for data reduction to improve kNN-based text categorization efficacy and efficiency and in particular the CHC and SSMA algorithms. They can select the most representative instances satisfying both the objectives of high

accuracy and reduction rates. As mentioned earlier, their main limitation is their long reduction time, which makes it difficult to apply these algorithms to very large data sets.

To cover as much as possible the problem of time cost, future work will consider how to speed up the evolutionary algorithms like CHC and SSMA. We would say that future research could be directed toward the study of using graphic processing unit (GPU) to parallelize the evolutionary algorithms. For future work, more experiments on more large text datasets can be also performed.

References

[1] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no 1, pp. 1-47, 2002.

[2] T. Songbo, "An effective refinement strategy for KNN text classifier," *Expert Systems with Applications*, vol. 30, no. 2, pp. 290-298, 2006.

[3] S. Garcia, J. Derrac, R. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 317-435, 2012.

[4] J. Bien and R. Tibshirani, "Prototype selection for interpretable classification," *Annals of Applied Statistics Journal*, vol. 5, no. 4, pp. 2403-2424, 2011.

[5] J. A. Olvera-Lopez, J. A. Carrasco-Ochoa, J. F. Martinez-Trinidad, and J. Kittler, "A review of instance selection methods," *Artificial Intelligence Review*, vol. 34, no. 2, pp. 133-143, 2010.

[6] T. Liu, A. W. Moore, and A. Gray, "New algorithms for efficient high dimensional nonparametric classification," *Journal of Machine Learning Research*, vol. 7, pp. 1135-1158, 2006.

[7] C. F. Tsai, Z. Y. Chen, and S. W. Ke, "Evolutionary instance selection for text classification," *Journal of Systems and Software* vol. 90, pp. 104-113, 2014.

[8] F. Barigou, "A new term weighting scheme for text categorization," *International Journal of Intelligent Systems Technologies and Applications*, vol. 14, no. 3/4, pp. 256–272, 2015.

[9] H. Zhou, J. Guo, and Y. Wang, "A feature selection approach based on term distributions," *SpringerPlus*, vol. 5, article no. 249, 2016.

[10] M. Grochowski and N. Jankowski, "Comparison of instance selection algorithms II. Results and comments," in *Proceedings of the 7th International Conference on Artificial Intelligence and Soft Computing*, Zakopane, Poland, 2004, pp. 580-585.

[11] N. Jankowski and M. Grochowski, "Comparison of Instance Selection Algorithms I. Algorithms survey," in *Proceedings of the 7th International Conference on Artificial Intelligence and Soft Computing*, Zakopane, Poland, 2004, pp. 598-603.

[12] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," *Data Mining and Knowledge Discovery*, vol. 6, no. 2, pp. 153-172, 2002.

[13] I. Triguero, J. Derrac, S. Garcia, and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest neighbor classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 86-100, 2012.

[14] T. Cover amd P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, 1967.

[15] G. L. Libralon, A. C. P. D. L. Carvalho, and A. C. Lorena, "Pre-processing for noise detection in gene expression classification data," *Journal of the Brazilian Computer Society*, vol. 15, no. 1, pp. 3-11, 2009.

[16] A. Arnaiz-Gonzalez, J. F. Diez-Pastor, J. J. Rodriguez, and C. Garcia-Osorio, "Instance selection for regression: adapting DROP," *Neurocomputing,* vol. 201, pp. 66-81, 2016.

[17] M. B. Stojanovic, M. M. Bozic, M. M. Stankovic, & Z. P. Stajic, "A methodology for training set instance selection using mutual information in time series prediction," *Neurocomputing*, vol. 141, pp. 236-245, 2014.

[18] P. Hart, "The condensed nearest neighbor rule," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515-516, 1968.

[19] G. Gates, "The reduced nearest neighbor rule," *IEEE Transactions on Information Theory*, vol. 18, no. 3, pp. 431-433, 1972.

[20] G. Ritter, H. Woodruff, S. Lowry, and T. Isenhour, "An algorithm for a selective nearest neighbor decision rule," *IEEE Transactions on Information Theory*, vol. 21, no. 6, pp. 665-669, 1975.

[21] I. Tomek, "Two modifications of CNN," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, no. 6, pp. 769-772, 1976.

[22] V. Devi and M. Murty, "An incremental prototype set building technique," *Pattern Recognition*, vol. 35, no. 2, pp. 505-513, 2002.

[23] J. Riquelme, J. Aguilar-Ruiz, and M. Toro, "Finding representative patterns with ordered projections," *Pattern Recognition*, vol. 36, no. 4, pp. 1009-1018, 2003.

[24] F. Angiulli, "Fast condensed nearest neighbor rule," in *Proceedings of the 22d International Conference on Machine Learning*, Bonn, Germany, 2005, pp. 25-325.

[25] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 2, no. 3, pp. 408-421, 1972.

[26] K. Hattori and M. Takahashi, "A new edited k-nearest neighbor rule in the pattern classification problem," *Pattern Recognition*, vol. 33, no. 3, pp. 521-528, 2000.

[27] J. S. Sanchez, F. Pla, and F. J. Ferri," Prototype selection for the nearest neighbor rule through proximity graphs," *Pattern Recognition Letters*, vol. 18, no. 6, pp. 507-513, 1997.

[28] D. Aha, D. Kibler, and M. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37-66, 1991.

[29] D. Randall Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 257-286, 2000.

[30] H. Brighton and C. Mellish. "Advances in instance selection for instance-based learning algorithms," *Data Mining and Knowledge Discovery*, vol. 6, no. 2, pp. 153-172, 2002.

[31] J. Derrac, S. Garcia, and F. Herrera, "A survey on evolutionary instance selection and generation," *International Journal of Applied Metaheuristic Computing*, vol. 1, no. 1, pp. 60-92, 2010.

[32] E. Leyva, A. Gonzalez, and R. Perez, "Knowledge-based instance selection: a compromise between efficiency and versatility," *Knowledge-Based Systems*, vol. 47, pp. 65-76, 2013.

[33] J. R. Cano, F. Herrera, and M. Lozano, "Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 6, pp. 561-575, 2003.

[34] S. Garcia, J. R. Cano, and F. Herrera, "A memetic algorithm for evolutionary prototype selection: a scaling up approach," *Pattern Recognition*, vol. 8, no. 41, pp. 2693-2709, 2008.

[35] J. R. Cano, F. Herrera, and M. Lozano, "Stratification for scaling up evolutionary prototype selection," *Pattern Recognition Letters*, vol. 26, no. 7, pp. 953-963, 2005.

[36] C. Garcia-Osorio, A. de Haro-Garcia, and N. Garcia-Pedrajas, "Democratic instance selection: a linear complexity instance selection algorithm based on classifier ensemble concepts," *Artificial Intelligence*, vol. 174, no. 5/6, pp. 410-441, 2010.

[37] I. Triguero, D. Peralta, J. Bacardit, S. Garcia, and F. Herrera, "MRPR: a MapReduce solution for prototype reduction in big data classification," *Neurocomputing*, vol. 150, pp. 331-345, 2015.

[38] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130-137, 1980.

[39] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the 14th International Conference on Machine Learning*, San Francisco, CA, 1997, pp. 412-420.

Fatiha Barigou https://orcid.org/0000-0001-5444-4000

She graduated from Department of Computer Science, University of Oran 1 Ahmed Ben Bella, Algeria. In 2012, she received her Ph.D. degrees in Computer Science from the University of Oran 1. She is currently a research member of Laboratory of Computer Science of Oran. Her research interests include natural language processing, information extraction, information retrieval, knowledge-based system, pattern recognition and text mining.