Knowledge-Based Systems 000 (2016) 1–13

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Instance selection of linear complexity for big data

Álvar Arnaiz-González, José-Francisco Díez-Pastor, Juan J. Rodríguez, César García-Osorio*

University of Burgos, Spain

ARTICLE INFO

Article history:
Received 18 December 2015
Revised 3 April 2016
Accepted 30 May 2016
Available online xxx

Keywords:
Nearest neighbor
Data reduction
Instance selection
Hashing
Big data

ABSTRACT

Over recent decades, database sizes have grown considerably. Larger sizes present new challenges, because machine learning algorithms are not prepared to process such large volumes of information. Instance selection methods can alleviate this problem when the size of the data set is medium to large. However, even these methods face similar problems with very large-to-massive data sets.

In this paper, two new algorithms with linear complexity for instance selection purposes are presented. Both algorithms use *locality-sensitive hashing* to find similarities between instances. While the complexity of conventional methods (usually quadratic, $\mathcal{O}(n^2)$, or log-linear, $\mathcal{O}(n \log n)$) means that they are unable to process large-sized data sets, the new proposal shows competitive results in terms of accuracy. Even more remarkably, it shortens execution time, as the proposal manages to reduce complexity and make it linear with respect to the data set size. The new proposal has been compared with some of the best known instance selection methods for testing and has also been evaluated on large data sets (up to a million instances).

1. Introduction

The k nearest neighbor classifier (kNN) [11], despite its age, is still widely used in machine learning problems [9,17,20]. Its simplicity, straightforward implementation and good performance in many domains means that it is still in use, despite of some of its flaws [37]. The kNN algorithm is included in the family of instance based learning, in particular within the *lazy learners*, as it does not build a classification model but just stores all the training set [8]. Its classification rule is simple: for each new instance, assign the class according to the majority vote of its k nearest neighbors in the training set, if $k = 1$, the algorithm only takes the nearest neighbor into account [45]. This feature means that it requires a lot of memory and processing time in the classification phase [48]. Traditionally, two paths have been followed to speed up the process: either accelerate the calculation of the closest neighbors [3,4], or decrease training set size by strategically selecting only a small portion of instances or features [38].

Regarding the acceleration of algorithms, perhaps one of the most representative approaches is to approximate nearest neighbors, a broadly researched technique in which the nearest neighbor search is done over a sub-sample of the whole data set [56]. In this field, many algorithms have been proposed for approximate nearest neighbor problems [3,4,30,34,39].

The focus of this paper is on the second path, the reduction of data set size. The reason is that this reduction is beneficial for most methods rather than only those based on nearest neighbors. Although we will only consider the reduction of instances (instance selection) in this paper, the reduction could also be applied to attributes (feature selection), or even both at the same time [51]. The problem is that the fastest conventional instance selection algorithms have a computational complexity of at least $\mathcal{O}(n \log n)$ and others are of even greater complexity.

The need for rapid methods for instance selection is even more relevant nowadays, given the growing sizes of data sets in all fields of machine learning applications (such as medicine, marketing or finance [43]), and the fact that the most commonly used data mining algorithms for any data mining task were developed when the common databases contained at most a few thousands of records. Currently, millions of records are the most common scenario. So, most data mining algorithms find many serious difficulties in their application. Thus, a new term has emerged, “Big Data”, in reference to those data sets that, by volume, variability and speed, make the application of classical algorithms difficult [44]. With regard to instance selection, the solutions that have appeared so far to deal with big data problems adopt the ‘divide and conquer’ approach [13,22]. The algorithms proposed in the present paper offer a different approach, just a sequential but very quick and simple processing of each instance in the data set.

* Corresponding author.Fax: +34947258910.
E-mail addresses: alvarag@ubu.es (Á. Arnaiz-González), jfdpastor@ubu.es (J.-F. Díez-Pastor), jjrodriguez@ubu.es (J.J. Rodríguez), cgosorio@ubu.es (C. García-Osorio).

http://dx.doi.org/10.1016/j.knosys.2016.05.056

In particular, the major contribution of this paper is the use of Locality-Sensitive Hashing (LSH) to design two new algorithms, which offers two main advantages:

- Linear complexity: the use of LSH means a dramatic reduction in the execution time of the instance selection process. Moreover, these methods are able to deal with huge data sets due to their linear complexity.
- *On-the-fly* processing: one of the new methods is able to tackle the instances in one step. It is not necessary for all instances fit in memory: a characteristic that offers a remarkable advantage in relation to big data.

The paper is organized as follows: Section 2 presents the reduction techniques background, with special emphasis on the instance selection methods used in the experimental validation; Section 3 introduces the concept of *locality-sensitive hashing*, the basis of the proposed methods which are presented in Section 4; Section 5 presents and analyzes the results of the experiments and, finally, Sections 6 and 7 set out the conclusions and future research, respectively.

2. Reduction techniques

Available data sets are progressively becoming larger in size. As a consequence, many systems have difficulties processing such data sets to obtain exploitable knowledge [23]. The high execution times and storage requirements of the current classification algorithms make them unusable when dealing with these huge data sets [28]. These problems can be decisive, if a lazy learning algorithm such as the nearest neighbor rule is used, and can even prevent results from being obtained. However, reducing the size of the data set by selecting a representative subset has two main advantages: it reduces the memory required to store the data and it accelerates the classification algorithms [19].

In the scientific literature, the term “reduction techniques” includes [61]: prototype generation [32]; prototype selection [52] (when the classifier is based on kNN); and (for other classifiers) instance selection [8]. While prototype generation replaces the original instances with new artificial ones, instance selection and prototype selection attempt to find a representative subset of the initial training set that does not lessen the predictive power of the algorithms trained with such a subset [45]. In the paper, prototype generation is not addressed, however a complete review on it can be found in [57].

2.1. Instance selection

The aforementioned term “instance selection” brings together different procedures and algorithms that target the selection of a representative subset of the initial training set. There are numerous instance selection methods for classification, a complete review of which may be found in [21]. Instance selection has also been applied to both regression [2,33] and time series prediction [26,55].

According to the order in which instances are processed, instance selection methods can be classified into five categories [21]. If they begin with an empty set and they add instances to the selected subset, by means of analyzing the instances in the training set, they are called incremental. The decremental methods, on the contrary, start with the original training data set and they remove those instances that are considered superfluous or unnecessary. Batch methods are those in which no instance is removed until all of them have been analyzed, instances are simply marked from removal if the algorithm determines that they are not needed, and at the end of the process only the unmarked instances are kept. Mixed algorithms start with a preselected set of instances. The process then decides either to add or to delete the instances. Finally, fixed methods are a sub-family of mixed ones, in which the number of additions and removals are the same. This approach allows them to maintain a fixed number of instances (more frequent in prototype generation).

Table 1
Summary of state-of-the-art instance selection methods used in the experimental setup (taxonomy from [21]; computational complexity from [31] and authors' papers).

Strategy	Direction	Algorithm	Complexity	Year	Reference
Condensation	Incremental	CNN	$\mathcal{O}(n^3)$	1968	[27]
	Incremental	PSC	$\mathcal{O}(n \log n)$	2010	[46]
	Decremental	RNN	$\mathcal{O}(n^3)$	1972	[25]
	Decremental	MSS	$\mathcal{O}(n^2)$	2002	[6]
Hybrid	Decremental	DROP1-5	$\mathcal{O}(n^3)$	2000	[60]
	Batch	ICF	$\mathcal{O}(n^2)$	2002	[8]
	Batch	HMN-EI	$\mathcal{O}(n^2)$	2008	[41]
	Batch	LSBo	$\mathcal{O}(n^2)$	2015	[37]

Considering the type of selection, three categories may be distinguished. This criterion is mainly correlated with the points that they remove: either border points, central points, or otherwise. Condensation techniques try to retain border points. Their underlying idea is that internal points do not affect classification, because the boundaries between classes are the keystone of the classification process. Edition methods may be considered the opposite of condensation techniques, as their aim is to remove those instances that are not well-classified by their nearest neighbors. The edition process achieves smoother boundaries as well as noise removal. In the middle of those approaches are hybrid algorithms, which try to maintain or even to increase the accuracy capability of the data set, by removing both: internal and border points [21].

Evolutionary approaches for instance selection have shown remarkable results in both reduction and accuracy. A complete survey of them can be found in [16]. However, the main limitation of those methods is their computational complexity [36]. This drawback is the reason why they are not taken into account in this study, because the methods it proposes are oriented towards large data sets.

In the remaining part of this section, we give further details of the most representative methods used in the experimental setup. A summary of the methods considered in the study can be seen in Table 1.

2.1.1. Condensation

The algorithm of Hart, *Condensed Nearest Neighbor* (CNN) [27] is considered the first formal proposal of instance selection for the nearest neighbor rule. The concept of training set consistency is important in this algorithm and is defined as follows: given a non empty set X ($X \neq \varnothing$), a subset S of X ($S \subseteq X$) is consistent with respect to X if, using the subset S as training set, the nearest neighbor rule can correctly classify all instances in X. Following this definition of consistency, if we consider the set X as the training set, a condensed subset should have the properties of being consistent and, ideally, smaller than X. After CNN appeared, other condensation methods emerged with the aim of decreasing the size of the condensed data set, e.g.: Reduced Nearest Neighbor (RNN) [25]. One of the latest is the Prototype Selection by Clustering (PSC) [46], which uses clustering to speed up the selection process. So, the use of clustering gives a high efficiency to PSC, if compared against state-of-the-art methods, and better accuracy than other clustering-based methods such as CLU [40].

In [6], the authors proposed a modification to the definition of a selective subset [54], for a better approximation to decision borders. The selective subset can be thought of as similar to the idea of the condensed algorithm of Hart, but applying a condi-