Pattern Recognition 48 (2015) 1878–1889

Contents lists available at ScienceDirect

Pattern Recognition

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

IRAHC: Instance Reduction Algorithm using Hyperrectangle Clustering

Javad Hamidzadeh [a,*], Reza Monsefi [b], Hadi Sadoghi Yazdi [b]

[a] Faculty of Computer Engineering and Information Technology, Sadjad University of Technology, Mashhad, Iran
[b] Department of Computer Engineering, Ferdowsi University of Mashhad (FUM), Mashhad, Iran

ARTICLE INFO

Article history:
Received 23 January 2013
Received in revised form
28 July 2014
Accepted 8 November 2014
Available online 18 November 2014

Keywords:
Instance reduction
Instance selection
Hyperrectangle
Instance-based classifiers
k-Nearest neighbor (*k*-NN)

ABSTRACT

In instance-based classifiers, there is a need for storing a large number of samples as training set. In this work, we propose an instance reduction method based on hyperrectangle clustering, called Instance Reduction Algorithm using Hyperrectangle Clustering (IRAHC). IRAHC removes non-border (interior) instances and keeps border and near border ones. This paper presents an instance reduction process based on hyperrectangle clustering. A hyperrectangle is an n-dimensional rectangle with axes aligned sides, which is defined by min and max points and a corresponding distance function. The min–max points are determined by using the hyperrectangle clustering algorithm. Instance-based learning algorithms are often confronted with the problem of deciding which instances must be stored to be used during an actual test. Storing too many instances can result in a large memory requirements and a slow execution speed. In IRAHC, core of instance reduction process is based on set of hyperrectangles. The performance has been evaluated on real world data sets from UCI repository by the 10-fold cross-validation method. The results of the experiments have been compared with state-of-the-art methods, which show superiority of the proposed method in terms of classification accuracy and reduction percentage.

1. Introduction

Instance reduction is a crucial task in instance based learning algorithms. Instance-based learning algorithms are often confronted with the problem of deciding which instances must be stored to be used during an actual test. Storing too many instances can result in a large memory occupation and an increase in execution time. In practice, some training sets may contain non-informative instances for the purpose of classification task, which can be either noisy or redundant. Therefore, a process is needed to discard superfluous instances from the training set. In literature, this discarding process is known as instance reduction. Through instance reduction, the training set size is reduced, which could be useful for reducing the runtimes in the actual test in classification process, particularly for instance-based classifiers when they use large data sets.

Instance reduction is an effective approach to increase the performance of instance-based classifiers when data sets are large, such as data mining, text categorization, financial forecasting, Web documents and information filtering in E-commerce [1]. The effectiveness of some instance-based methods for data classification, such as *k*-NN and DDC [2] are related to the size of the training set.

One choice in designing an instance reduction algorithm is to decide whether to maintain a subset of the original instances or to modify the instances using a new representation, which are called Selection and Abstraction methods respectively [3]. Some algorithms are based on the combination of Selection and Abstraction methods, called hybrid. One important difference among instance reduction methods is whether to maintain border or central instances. The reason to maintain border instances is that the internal instances do not affect the decision boundaries as much as border instances. Thus, internal instances can be discarded without having much effect on classification accuracy. Some algorithms remove noisy or outlier instances that do not agree with their neighbors [4]. Although due to the lack of enough border instances, some algorithms are urged to maintain some central instances in order to use them to classify properly.

In the reduction of instances, we often face a trade-off between the size of the sample and the classification quality [5]. A successful algorithm often significantly reduces the size of the training set without a significant reduction of generalization accuracy. In some cases generalization accuracy can increase with instance reduction, such as when noisy instances are removed and when decision boundaries are smoothed. A variety of instance reduction, distance measure, and weighted NN techniques has been proposed [6]. A survey of different methods for instance reduction can be seen in Refs. [7,8]. Fast nearest neighbor classification has been investigated by exploiting metric structures efficiently and hardware resources

* Corresponding author. Tel.: +98 51 36029000; fax: +98 51 36029110.
E-mail addresses: J_hamidzadeh@sadjad.ac.ir, Hamidzadehj@gmail.com (J. Hamidzadeh), monsefi@um.ac.ir (R. Monsefi), h-sadoghi@um.ac.ir (H. Sadoghi Yazdi).

http://dx.doi.org/10.1016/j.patcog.2014.11.005
0031-3203/

[9]. Weighted NN aims to weigh the discriminative capability of different features or different nearest neighbors [10].

In this paper, a novel hybrid algorithm is proposed to abstract and select a new subset of instances. It aims to preserve instances near or within the boundaries of classes, since they affect the decision surface. For non-boundary and internal instances, the algorithm calculates their mean and then retains it. The aim of this paper is to reduce the size of training set while trying to maintain or possibly to improve generalization accuracy. In this paper, we propose a hybrid Instance Reduction Algorithm based on Hyperrectangle Clustering, namely IRAHC.

Although a variety of instance reduction methods have been proposed in the literature, no single approach can be considered superior or a guarantee for satisfactory results in terms of classification accuracy or efficiency. While there are enormous methods for finding proper instances, it seems that a method superior in certain domains is inferior in other domains [8]. Hence, searching for efficient approaches in instance reduction is still an active field of research.

The remainder of the paper is organized as follows: In Section 2, a survey of instance reduction algorithms is presented. The proposed method for instance reduction is introduced in Section 3. The experimental results of the proposed method have been shown in Section 4. Finally, Section 5 contains conclusions and future works.

2. Survey of instance reduction algorithms

Several methods have been proposed for instance reduction. This section surveys some important instance reduction methods. Most of the algorithms discussed here use T as original instances in the training set and $S \subseteq T$ (S as a subset of T) as their representation.

The Condensed Nearest Neighbor (CNN) [11] is the first method of instance reduction. The CNN begins by randomly selecting one instance which belongs to each class from T and puts them in S. Then each instance in T is classified using only the instances in S. If an instance is misclassified, it will be added to S in order to ensure that it will be classified correctly. This process repeats until there is no instance in T that is misclassified. Thus, the subset S, produced by CNN, is consistent. CNN has several problems. CNN assigns noisy and outlier instances to S. Thus, this hurt the classification accuracy. Also, CNN does not guarantee to produce a minimal consistent subset S. CNN is dependent on instance order in training set T. The time complexity of CNN is $O(n^2)$, where n is the size of the training set.

Gates [13] introduced the Reduced Nearest Neighbor algorithm (RNN). The RNN algorithm assigns all instances in T to S first and then removes each instance from S, till further removal causes no other instances in T to be misclassified by the remaining instances in S. The time complexity of RNN is $O(n^3)$, where n is the size of the training set. Thus, RNN produces subset S with smaller cardinality than it in CNN. Thus, it is better than CNN in terms of instance reduction percentage and classification speed. RNN is less sensitive to noise than CNN, because it is able to remove noisy and outlier instances, but RNN is more expensive than CNN in terms of learning time, $O(n^3)$ versus $O(n^2)$.

A recent extension of CNN is the Generalized Condensed Nearest Neighbor (GCNN) that has been introduced in Ref. [5]. GCNN is the same as CNN, but GCNN assigns instances which satisfy an absorption criterion to S. The absorption is calculated in terms of the nearest neighbors and enemies (the nearest instances of the other classes). An instance is absorbed or included in S if its difference distance, compared to its nearest neighbor and enemy, is not more than a defined threshold. GCNN by utilizing the absorption criterion could reduce the size of S compared to CNN algorithm, but determining value for the threshold could be a challenge in this algorithm depending to each classification problem.

In the Edited Nearest Neighbor (ENN) algorithm [12], all training instances are assigned to S first, and then each instance in S will be removed if it does not agree with the majority label of its k nearest neighbors. This algorithm removes noisy and outlier instances. Thus, it could enhance classification accuracy. It keeps internal instances in contrast to removing boundary instances. Therefore, it cannot reduce instances as much as most other reduction algorithms. A variant of this method is the Repeated ENN (RENN). The RENN applies the ENN algorithm repeatedly until all instances which remain agree with the majority label of their neighbors. Another extension of ENN is the All k-NN method [14]. This algorithm works as follows: for $i=1$–k, flag any instance which does not agree with the majority label of their k nearest neighbors as bad. After finishing the loop, all the instances from S that has been flagged as bad will be removed.

Lowe [15] introduced a Variable Similarity Metric (VSM) algorithm. An instance t is discarded if the label of all k of its nearest neighbors is the same. Using this conservative rule, noisy and some internal instances could be removed. VSM exhibits good classification accuracy, but it does not have a good instance reduction percentage. Wilson and Martinez [4] introduced five algorithms, the Decremental Reduction Optimization Procedure algorithms, called DROP1–DROP5. Unlike most previous methods, these algorithms pay more attention to the order in which instances have been removed. In those, each instance P has k nearest neighbors and those instances that have P as one of their k nearest neighbors are called the associates of P. Among them, DROP3 is the most successful method. As an initial step, it applies a noise filter algorithm such as ENN and then removes an instance if its associates in the original training set are correctly classified without that. The main drawback of DROP3 is high order of computation time in large scale classification problems or non-scalability.

The Iterative Case Filtering algorithm (ICF) was proposed in Ref. [16]. ICF is based on the Coverage and Reachable sets which are the neighborhood set and associates set, respectively. The neighborhood set for an instance such as P is all the instances between P and the nearest enemy of P. The nearest enemy of P is the nearest instance from the other classes. In this method, an instance P is flagged for removing if $|Reachable(p)| > |Coverage(p)|$, which means P instance would be removed, where the number of other instances that can classify P correctly are more than the number of instances that P can correctly classify. Then, all instances flagged for removing will be discarded.

Lumini and Nanni [17] introduced a clustering method for instance selection, called CLU. In CLU, after splitting T in some clusters, the centers of the clusters will be used as instances. Another method that is based on clustering idea is Nearest Subclass Classifier (NSB) [7]. In NSB, a different number of instances in each class could be selected by using the maximum variance cluster algorithm. Raicharoen and Lursinsap introduced a divide and conquer approach to the Pairwise Opposite Class Nearest Neighbor algorithm, namely POC-NN, [18]. In POC-NN, border instances are selected based on the mean of the instances in each class. Another method that finds border instances is proposed in Ref. [3], namely Prototype Selection by Clustering (PSC), which applies clustering algorithm. Two types of clustered regions are in PSC method, namely homogeneous and heterogeneous clusters. In homogeneous clusters all the instances are from the same class and in heterogeneous clusters all the instances are not from the same clusters. Thus, two types of instances are in PSC. One is the mean of the instance in each homogeneous cluster and the others are from heterogeneous clusters as border instances.

Another method that removes internal instances and keeps border instances is introduced in Ref. [19], as Template Reduction for k-NN (TRk-NN). TRk-NN is an iterative routine for removing redundant instances. TRk-NN uses nearest enemy concept to determine close to border instances.

Marchiori [20] proposed a graph-based algorithm for instance selection, namely HMN. Hit Miss Network (HMN), which is a directed graph of instances. In this graph, each instance is represented by a node and is adjacent to nearest neighbor instances from all other classes or there is an edge between each instance and its nearest neighbor from all other classes. The edge between two instances