Applied Soft Computing 37 (2015) 416–443

Contents lists available at ScienceDirect

Applied Soft Computing

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

Simultaneous instance and feature selection and weighting using evolutionary computation: Proposal and study☆

Javier Pérez-Rodríguez, Alexis Germán Arroyo-Peña, Nicolás García-Pedrajas*

Department of Computing and Numerical Analysis, Edificio Einstein, 3ª Planta, University of Córdoba, Campus de Rabanales, 14014 Córdoba, Spain

ARTICLE INFO

Article history:
Received 9 July 2013
Received in revised form 2 October 2014
Accepted 18 July 2015
Available online 13 August 2015

Keywords:
Instance selection
Feature selection
Instance weighting
Feature weighting
Class-imbalanced datasets

ABSTRACT

Current research is constantly producing an enormous amount of information, which presents a challenge for data mining algorithms. Many of the problems in some of the most relevant research areas, such as bioinformatics, security and intrusion detection or text mining, involve large or huge datasets. Data mining algorithms are seriously challenged by these datasets. One of the most common methods to handle large datasets is data reduction. Among others, feature and instance selection are arguably the most commonly used methods for data reduction. Conversely, feature and instance weighting focus on improving the performance of the data mining task.

Due to the different aims of these four methods, instance and feature selection and weighting, they can be combined to improve the performance of the data mining methods used. In this paper, a general framework for combining these four tasks is presented, and a comprehensive study of the usefulness of the 15 possible combinations is performed.

Using a large set of 80 problems, a study of the behavior of all possible combinations in classification performance, data reduction and execution time is carried out. These factors are also studied using 60 class-imbalanced datasets.

1. Introduction

The overwhelming amount of data available in any research field poses new problems for data mining and knowledge discovery combinations. This huge amount of data makes most existing algorithms inapplicable to many real-world problems. Two approaches have been used to face this problem: scaling up data mining algorithms [1] and data reduction. However, scaling up a certain algorithm is not always feasible. Data reduction consists of removing the missing, redundant, information-poor and/or erroneous data to obtain a tractable problem size. Data reduction techniques use different approaches, including feature selection [2], feature-value discretization [3] and instance selection [4].

The feature and instance weighting processes aim to improve the performance of the data mining task. Because no data reduction is performed, the main target is a better result of the data mining process, e.g., a better clustering or a more accurate classifier.

These four processes have been used separately, and in a few cases in combinations of two, as in simultaneous instance and feature selection. However, there is no general framework for combining all of them. In this paper, first a general framework for combining the four processes using evolutionary computation is proposed, and a complete study of the performance of all possible combinations is performed. Considering the possible combinations and the four methods separately, the 15 possibilities using standard datasets and class-imbalanced problems are evaluated.

The underlying idea of this paper is to answer the question of whether combining these four processes might improve performance. To achieve this goal, a unified approach for the four tasks that allows combining any of them is needed. We must then perform a thorough experimental comparison of the different combinations to study their behavior.

To provide the necessary focus, the study is restricted to the classification problem, although a similar study could be performed for other data mining tasks, such as clustering. A nearest neighbor rule is used as the base classification method. We have used this classification method because it is the most commonly used when instance selection is applied. Furthermore, due to the large number

☆ This work was supported in part by the Project TIN2011-22967 of the Spanish Ministry of Science and Innovation and the Project P09-TIC-4623 of the Junta de Andalucía.

* Corresponding author. Tel.: +34 957211032; fax: +34 957218630.
E-mail addresses: javier.perez@uco.es (J. Pérez-Rodríguez), i52arpea@uco.es (A.G. Arroyo-Peña), npedrajas@uco.es (N. García-Pedrajas).

http://dx.doi.org/10.1016/j.asoc.2015.07.046
1568-4946/

of experiments reported, using more than a classifier method was not feasible.

Although many methods have been proposed for each of these tasks, we restrict ourselves to evolutionary approaches for two reasons. First, evolutionary approaches perform better when they are compared with other algorithms [5] as a general rule. Wrapper approaches usually offer better performance than filters [6]. Second, as we intend to try all possible combinations of the four tasks, only powerful metaheuristics offer the necessary flexibility for such fusion. Among those metaheuristics, evolutionary computation has shown arguably the best performance across a wide variety of problems.

This paper is organized as follows. Section 2 presents a short review of instance and feature selection and weighting. Section 3 describes our proposed framework. Section 4 shows the experimental setup. Section 5 presents and discusses the experimental results, and Section 6 states the conclusions of our work.

2. Selecting and weighting instances and features

As a general statement of the problem, consider a problem involving K classes and N training instances with M features whose class membership is known. Let $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_N, y_N)\}$ be the set of N training samples, where each instance $\mathbf{x}_i$ belongs to a domain X. Each label is an integer from the set $Y = \{1, \ldots, K\}$.

Considering this setup, the feature selection, feature weighting, instance selection and instance weighting processes can be carried out. Each process has different aims. In the following, each task is briefly explained.

2.1. Instance selection

Instance selection [7] consists of the selection a subset of the whole available dataset to achieve the original objective of the data mining application as if all of the data were used. Several different instance selection variants exist. Two major models are distinguishable [5]: instance selection as a method for prototype selection for algorithms based on prototypes (i.e., K-nearest neighbors) and training set selection to obtain the training data for a learning algorithm (i.e., classification trees or neural networks).

The instance selection problem for instance-based learning can be defined as [8] "the isolation of the smallest set of instances that enable us to predict the class of a query instance with the same (or higher) accuracy than the original set".

Different groups of learning algorithms have different learning/search bias [9] that must be addressed by instance selection algorithms. This may make many instance selection algorithms useless when their design philosophy is not appropriate for the problem at hand. Wrapper approaches do not assume any data structure or classifier behavior, adapting the instance selection to classifier performance. This, they usually are the best performing methods.

Brighton and Mellish [8] stated that the structure of the classes for a given dataset can differ greatly; an instance selection procedure can thus have good performance in one problem and a poor performance in another. Thus instance selection algorithms must gain some insight into the class structures to perform an efficient instance selection. However, this insight is rarely available and very difficult to obtain. However, approaches based on evolutionary computation do not assume any special form of the space, classes or boundaries between classes; and are only guided by the ability of each subset of instance to solve the data mining problem. The algorithm thus learns the relevant instances from the data without imposing any constraint in the form of classes or boundaries between them. García-Pedrajas and Pérez-Rodríguez [10] proposed selecting the instances more than once for a better accuracy and reduction.

Evolutionary computation has been shown [5,11,12] to be the most efficient combination for instance selection. However, it suffers from scalability problems. The computational cost, even for moderately large datasets [13], is a serious problem for this approach. For very large or huge datasets, evolutionary computation-based methods are not applicable. Stratification [14] and democratization [13,15] have been proposed to solve this scalability problem. A similar approach was constructed to scalable instance selection specially focused on class-imbalanced datasets [16].

2.2. Instance weighting

Instance weighting is arguably the least frequent of these tasks. Problems in instance weighting include the large search space and the difficulty of devising algorithms. There are two alternatives for instance weighting. Instance weights can be used for case-based methods, such as nearest neighbors:

$$d(\mathbf{q}, \mathbf{x}) = \sqrt{\sum_{j=1}^{M} v_{\mathbf{x}}^2 (q_j - x_j)^2}, \tag{1}$$

where $v_{\mathbf{x}}$ is the weight associated with instance $\mathbf{x}$. Instances with a higher value of $v_{\mathbf{x}}$ are less relevant for classification, as they tend to be effectively less closer to the query instance. With instance weighting, more complex decision surfaces can be constructed [17].

A second alternative may be specifically designed for many classifiers, such as decision trees [18], and consists of weighting the relevance of each instance in the classification. These weights can be used for classifier boosting [19] or class-imbalanced datasets [20,21]. However, this alternative is not applicable for a 1-NN; this paper thus does not consider it.

2.3. Feature selection

Feature selection is an important and frequently used data preprocessing technique for data mining [22]. In contrast with other dimensionality reduction techniques, feature selection preserves the original variable semantics, thus offering the advantage of interpretability by a domain expert [23].

Feature selection has been a fertile research and development field since the 1970s in statistical instance recognition [24], machine learning [4,25], and data mining [26], and it has been widely applied to many fields, including text categorization [27], image retrieval [28] customer relationship management [29], intrusion detection [30], and genomic analysis [31].

Feature selection can be defined as selecting a subset of M' features from a set of M features, $M' < M$, such that the value of a criterion function is optimized over all subsets of size M' [32]. The feature selection objectives are manifold, the most important ones being the following [23]:

- To avoid over-fitting and improve model performance, i.e., prediction performance for supervised classification and better cluster detection for clustering.
- To provide faster and more cost-effective models.
- To gain a deeper insight into the underlying processes that generate data.

Feature selection usually has two main goals: reducing datasets and gaining knowledge about the most important problem features. Many algorithms have been developed for feature selection.