Expert Systems With Applications 58 (2016) 1–9

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Improving the performance of inductive learning classifiers through the presentation order of the training patterns

Gonzalo A. Ruz*

[a] Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Av. Diagonal Las Torres 2640, Peñalolén, Santiago, Chile
[b] Center of Applied Ecology and Sustainability (CAPES), Santiago, Chile

ARTICLE INFO

Article history:
Received 28 February 2015
Revised 1 April 2016
Accepted 2 April 2016
Available online 5 April 2016

Keywords:
Inductive learning
Rules family
Clustering
Classification

ABSTRACT

Although the development of new supervised learning algorithms for machine learning techniques are mostly oriented to improve the predictive power or classification accuracy, the capacity to understand how the classification process is carried out is of great interest for many applications in business and industry. Inductive learning algorithms, like the Rules family, induce semantically interpretable classification rules in the form of if-then rules. Although the effectiveness of the Rules family has been studied thoroughly and new and improved versions are constantly been developed, one important drawback is the effect of the presentation order of the training patterns which has not been studied in depth previously. In this paper this issue is addressed, first by studying empirically the effect of random presentation orders in the number of rules and the generalization power of the resulting classifier. Then a presentation order method for the training examples is proposed which combines a clustering stage with a new density measure developed specifically for this problem. The results using benchmark datasets and a real application of wood defect classification show the effectiveness of the proposed method. Also, since the presentation order method is employed as a preprocessing stage, the simplicity of the Rules family is not affected but instead it enables the generation of fewer and more accurate rules, which can have a direct impact in the performance and usefulness of the Rules family in an expert system context.

1. Introduction

Pattern classification consists in assigning (automatically) a class label to a vector of attributes. These attributes, hopefully, have sufficient discriminatory power in order for the classifier to correctly assign each pattern to its appropriate class. The construction or learning of a classifier from data, consists in a *training* phase, where a training set of attribute vectors are used to adjust the parameters of the classifier, for example in a neural network, it would be the weights and the biases, or in a naive Bayes classifier, the distribution of each class and class conditional probabilities of each attribute. Then, the performance of a trained classifier is measured during a *testing* phase, using what is called a test set, which contains attribute vectors that were not used during the training. Throughout the years we have seen new or improved classifiers coming from the machine learning community that are more powerful, in the sense of accuracy, such as support vector machines (SVM) and its variants (Cortes & Vapnik, 1995) and random forests (RF) (Breiman, 2001). In general, one can find many applications of classifiers in different domains, for example, SVM have been used for mechanical faults diagnostic in induction motors (Baccarini, Rocha e Silva, Rodrigues de Menezes, & Caminhas, 2011), neural networks for fraud detection in medical claims (Ortega, Figueroa, & Ruz, 2006), Bayesian network classifiers in the recognition of control chart patterns (Ruz & Pham, 2009) and policy making for broadband adoption (Ruz, Varas, & Villena, 2013), a naive Bayes classifier is used to predict job performance in a call center (Valle, Varas, & Ruz, 2012), and customer churn prediction using random forests (Xie, Li, Ngai, & Ying, 2009).

While research in some areas of machine learning is devoted to generate more powerful classifiers, there is a trade-off between the quantitative aspect of the classifier, correct classification or accuracy, and the qualitative aspect of the classifier, i.e., understand how the classification process is carried out. Most of the more powerful classifiers are considered as black-box models, where although achieving high quantitative performances, they tend to have a rather low qualitative aspect of the model.

For real world expert systems and data mining applications, in different areas such as retail, banking, finance, health, etc. many of these machine learning techniques are still considered as *black*

* Corresponding author. Tel.: +56223311200.
E-mail address: gonzalo.ruz@uai.cl

http://dx.doi.org/10.1016/j.eswa.2016.04.003
0957-4174/

magic, in the sense that they are not able to explain or give explicit classification rules. Therefore, this skepticism for these techniques can in many cases, even after a successful prototype or trial period, never consolidate in a real working application in the company. One approach to overcome this problem, is to work with more simple classifiers, known as white-box models, where the classification process is made explicit through rules, such as IF-THEN rules or decision trees. In this case, quantitative performance is usually reduced, but the qualitative aspect of the model is enhanced. Therefore, in many applications where one can compromise the quantitative performance of the model in order to gain more qualitative characteristics of how the classification is conducted, are in much need in real industry applications. Of course one could always approach the problem by training a black-box model for high accuracy and a white-box model to use to explain in a more broader sense, in a board meeting for example, how the classification is carried out.

Amongst the many white-box classifiers, in this paper, we revisit *inductive learning*, or the generation of IF-THEN rules for automatic classification purposes. There have been a number of inductive learning programs developed throughout the years, like the well-known programs ID3 (Quinland, 1983), which is a divide-and-conquer program, and the AQ program (Michalski, 1990). For this paper, the simple inductive learning algorithms belonging to the RULES (RULe Extraction System) family developed by Pham and Aksoy (1993); 1995a); 1995b) are of interest.

A common drawback that the Rules family algorithms have is that the generalization power of the rules generated during the training process is affected by the presentation order of the patterns used in this process. Random presentation orders of the training data can yield to different results, which may not be always efficient and accurate. This issue arises from the fact that the Rules family during its rule induction process considers the class of a selected seed example as the target class. Then it tries to induce rules that cover as many examples of the target class as possible through a rule evaluation function. No formal criterion is given on efficient ways to select the seed example.

The problem of the presentation order for the Rules family was first considered in Pham, Dimov, and Salem (2000), where a clustering approach was used to generate different presentation orders with the aim of reducing the RULES-4 execution time and the generation of fewer rules. Later in Pham and Salem (2004), a reorganization of the training data is proposed that essentially consists in introducing an example of each different class in the sequence of the training examples, this way the algorithm would learn from examples of at least two classes from the beginning of the training. This would improve the performance of the Rules algorithm by avoiding the generation of one default rule from the initial examples in the training data.

Apart from these two works, there have been no other recent efforts to analyze and propose new methods as a preprocessing stage to minimize the effects of the presentation order of the training patterns. In fact, more recent research concerned with the Rules family algorithms have been concentrated in handling continuous class labels (ElGibreen & Aksoy, 2015a). For example in ElGibreen and Aksoy (2015b), a new version of the Rules family y proposed called RULES-3C which incorporates properties from reinforcement learning to handle continuous classes. Another extension to the Rules family is RULES-IT (ElGibreen & Aksoy, 2014b), which is an incremental covering algorithm that integrates transfer learning to enable the use of past experiences from different domains. RULES-IT has proven to be effective for handling incomplete data and missing labels (ElGibreen & Aksoy, 2014a). It is important to point out that none of these new versions of the Rules family address the issue of the effect of the presentation order of the training examples in their results, which is the main contribution of this paper.

Another preprocessing approach alternative to changing the presentation order of the training examples is to apply instance reduction before carrying out the rule induction as in Othman and Bryant (2015). It was found that by applying instance reduction methods fewer rules were generated without compromising the classification performance.

Fuzzy min-max neural networks (Simpson, 1992) also can be affected by the presentation order. FMMN grow decision boundaries called hyperboxes based on the presentation of the training examples one by one, thus, the resulting hyperboxes and the predictive power may vary depending on the sequence of the training data. To overcome this problem two training algorithms have been proposed for this type of neural networks (Rizzi, Panella, & Mascioli, 2002). Nowadays, new versions have focused in handle mixed attributes (continuous and discrete) such as in Shinde and Kulkarni (2016) or a modified version for clustering (Seera, Lim, Loo, & Singh, 2015).

More recent techniques have been developed for fuzzy ART-type networks such as fuzzy ARTMAP (FAM) (Pourpanah, Lim, & Saleh, 2016), that also suffer from data presentation order issue (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992). One alternative to overcome this problem has been the use of genetic algorithm. Like in Loo, Liew, Seera, and Lim (2015), where the presentation order is coded from 1 to N, where N is the number of instances in the training set. Then the training sequence, together with some other parameters to be optimized were coded in a chromosome. The fitness function corresponds to the average classification accuracy of the test set, within a ten-fold cross validation scheme. Overall the results using five benchmark data sets showed the effectiveness of the proposed approach. Other examples using a genetic algorithm for the selection of the training pattern order can be found in Baek, Lee, Lee, Lee, and Kim (2014); Palaniappan and Eswaran (2009). In the case of a FAM ensemble, in Oong and Isa (2014), a data presentation method is proposed based on the ascending order of the value from the most uncorrelated input features. The results using benchmark data sets showed that the proposed ordering algorithm obtained better generalization performance in seven out of the eleven data sets.

It is desirable to overcome this presentation order problem without compromising the simplicity of the Rules family algorithms, therefore, in this work, a preprocessing stage is proposed to reduce the variability of the generalization of the Rules family algorithms. In particular, the contributions of this work are

- The quantification in terms of the number of rules and generalization power of the Rules family when random order presentation of the training data is used.
- A new density measure that uses a fuzzy membership function to select representative seed examples from each class.
- A presentation order method of the training examples that combines clustering techniques with the proposed density measure.
- The evaluation of the proposed approach not only on benchmark data sets but also on a real application of wood defect classification.

The outline of this paper is as follows. A brief description of the Rules family and the clustering techniques employed by the proposed presentation order of the training patterns are specified in Section 2, the density measure used to rank the data is developed in Section 3. Section 4 introduces the presentation order of the training patterns method, whereas Section 5 describes the data sets used to test the proposed technique and how the simulations were conducted. Results and discussions are carried out in