Applied Soft Computing 30 (2015) 136–150

Contents lists available at ScienceDirect

Applied Soft Computing

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

Distributed feature selection: An application to microarray data classification

V. Bolón-Canedo*, N. Sánchez-Maroño, A. Alonso-Betanzos

Laboratory for Research and Development in Artificial Intelligence (LIDIA), Computer Science Department, University of A Coruña, 15071 A Coruña, Spain

ARTICLE INFO

Article history:
Received 14 March 2013
Received in revised form 15 January 2015
Accepted 16 January 2015
Available online 7 February 2015

Keywords:
Feature selection
Distributed learning
Microarray data

ABSTRACT

Feature selection is often required as a preliminary step for many pattern recognition problems. However, most of the existing algorithms only work in a centralized fashion, i.e. using the whole dataset at once. In this research a new method for distributing the feature selection process is proposed. It distributes the data by features, i.e. according to a vertical distribution, and then performs a merging procedure which updates the feature subset according to improvements in the classification accuracy. The effectiveness of our proposal is tested on microarray data, which has brought a difficult challenge for researchers due to the high number of gene expression contained and the small samples size. The results on eight microarray datasets show that the execution time is considerably shortened whereas the performance is maintained or even improved compared to the standard algorithms applied to the non-partitioned datasets.

1. Introduction

Over the last decade, feature selection, which consists of detecting the relevant features and discarding the irrelevant ones [1,2], has been an active research area due to the high dimensionality of the datasets. Feature selection methods usually come divided into three types: filters, wrappers and embedded methods. While wrapper models involve optimizing a predictor as part of the selection process, filter models rely on the general characteristics of the training data to select features independently of any predictor. The embedded methods generally use machine learning models for classification, and then an optimal subset of features is built by the classification algorithm. In the last few years, ensemble methods represented a new type of methods for feature selection. They aim to cope with the instability issues observed in many techniques for feature selection when the characteristics of the data change [3–5]. In previous works, several feature selection methods are applied and the obtained features are merged into a more stable subset of features prior to classification or the different predictions obtained with the different subsets of features are somewhat combined [6,7]. However, as stated in [8], even when the subset of features is not optimal, filters are preferable due to their computational and statistical scalability, so they will be the focus of this research.

The filter approach is commonly divided into two different subclasses: individual evaluation and subset evaluation [9]. Individual evaluation is also known as feature ranking and assesses individual features by assigning them weights according to their degrees of relevance. On the other hand, subset evaluation produces candidate feature subsets based on a certain search strategy. Each candidate subset is evaluated by a certain evaluation measure and compared with the previous best one with respect to this measure. While the individual evaluation is incapable of removing redundant features because these are likely to have similar rankings, the subset evaluation approach can handle feature redundancy with feature relevance. However, methods in this framework can suffer from an inevitable problem which is caused by searching though the possible feature subsets. This stage, required in the subset generation step, usually increases the computational time. Having said that, a revision of the existing literature showed that the subset evaluation approach outperformed the ranking methods [9–12].

Feature selection is usually applied in a centralized manner, i.e. a single learning model is used to solve a given problem. However, if the data is distributed, feature selection may take advantage of processing multiple subsets in sequence or concurrently. The need to use distributed feature selection can be two-fold. On the one hand, with the advent of network technologies data is sometimes distributed in multiple locations and may consequently be biased. On the other hand, most of the existing feature selection algorithms do not scale well and their efficiency significantly deteriorates or even becomes inapplicable when dealing with large-scale data. In order to increase efficiency, learning can be parallelized by

* Corresponding authors. Tel.: +34 981167000; fax: +34 981167160.
E-mail addresses: vbolon@udc.es (V. Bolón-Canedo), ciamparo@udc.es (A. Alonso-Betanzos).

http://dx.doi.org/10.1016/j.asoc.2015.01.035
1568-4946/

distributing the subsets of data to multiple processors, learning in parallel and then combining them. There are two main techniques for partitioning and distributing data: vertically, i.e. by features, and horizontally, i.e. by samples. Distributed learning has been used to scale up datasets that are too large for batch learning in terms of samples [13–15]. While not common, there are some other developments that distribute the data by features [16,17]. In this study, the data are distributed vertically in order to have the feature selection process distributed. After having the data distributed in small feature subsets and selecting the relevant features from each subset, a merging procedure is performed which updates the feature subset according to improvements in the classification accuracy.

Although this approach can be applied to any feature-abundant classification problem, it is especially suitable for application to microarray data. This type of data has become very popular in the past decade since it poses a difficult challenge for machine learning researchers due to its high number of features and its small sample size. In this domain, features represent gene expression coefficients corresponding to the abundance of mRNA – messenger ribonucleic acid – in a sample (e.g. tissue biopsy), for a number of patients. Although there are usually very few samples, the number of features in the raw data ranges from 6000 to 60,000. A typical classification task is to separate healthy patients from cancer patients based on their gene expression "profile". By applying the proposed distributed methodology to this domain, we will be able to deal with subsets with a more balanced feature/sample ratio and avoid overfitting problems. The experimental results from eight different databases demonstrate that our proposal can improve the performance of original feature selection methods and show important savings in running times.

The remainder of the paper is organized as follows: Section 2 describes the state of the art on distributed feature selection and feature selection methods on microarray data, Section 3 introduces our distributed approach, Section 4 reveals the experimental setup and Section 5 visualizes the experimental results. Finally, Sections 6 and 7 provide the discussion and conclusions, respectively.

2. State of the art

The literature about feature selection for microarray data is abundant. Different feature selection strategies have been proposed over the last years for feature/gene selection. Ferreira and Figueiredo [18] proposed combining unsupervised feature discretization and feature selection techniques which have improved previous related techniques over several microarray datasets. In [19] a new framework for feature selection based on dependence maximization between the selected features and the labels of an estimation problem is presented and tested over microarray data, showing promising results. Wang et al. [20] also proposed a novel filter framework to select optimal feature subsets based on a maximum weight and minimum redundancy criterion. Hybrid methods have been recently tested on this type of data [21,22] obtaining high classification accuracies. Embedded methods have also been proposed, such as in [23], where the authors introduced an algorithm that simultaneously selects relevant features during classifier construction by penalizing each feature's use in the dual formulation of support vector machines. On the other hand, trying to overcome the problem that a weakly ranked gene could be relevant within an appropriate subset of genes, Sharma et al. [24], introduced an algorithm that first distributes genes into relative small subsets, then selects informative smaller subsets of genes from a subset and merges the chosen genes with another gene subset to update the final gene subset. Their method showed promising classification accuracy for all the test datasets.

As we have said, many filter approaches were applied to successfully classify microarray data [8,12,25], however according to the authors' knowledge, there has been no attempt in the literature to tackle this problem with distributed feature selection, apart from the aforementioned proposal of Sharma et al. [24]. In fact, feature selection in distributed environments is a poorly explored field and very few references were found in the literature, most of which undertaken over the few last years. It is worth mentioning the research of Das et al. [26], where an algorithm is presented which, based on a horizontal partition (by samples), performs feature selection in an asynchronous fashion with a low communication overhead by which each peer can specify its own privacy constraints. A vertical partition of the data (by features) to generate the diverse components of an ensemble [27] is also present in the literature. However in these cases, feature selection is not applied to the different partitions of the data and therefore the model may be constructed based on irrelevant features. More recently, Banerjee and Chakravarty [28] proposed a distributed feature selection method evolved from a method called virtual dimension reduction, where the partition of the data can be done both vertically or horizontally. Zhao et al. [29] presented a distributed parallel feature selection algorithm based on maximum variance preservation. The algorithm can read data in a distributed form and perform parallel feature selection in both symmetric multiprocessing mode via multithreading and massively parallel processing.

Still, DNA microarray data prevents the use of horizontal partitioning because of the small sample size. The distributed methods mentioned above based on vertical partitioning have not been designed specifically for dealing with microarray data, so they do not tackle the particularities of this type of data, such as the high redundancy present among the features. The method proposed in [24] does address these issues, but it has the disadvantage of being computationally expensive by interacting with a classifier to select the genes in each subset. In this work we will propose a distributed filter method, suitable for application to microarray data and with a low computational cost.

3. The proposed method

Distributed feature selection has not been deeply explored yet. So, in this paper we present a distributed filter approach trying to improve upon previous accuracy results over microarray data as well as reducing the running time. Our proposal consists of performing several fast filters over several partitions of the data, combined afterwards into a single subset of features. Thus, we divide each dataset D into several small disjoint subsets D_i. The filter is applied to each of them, generating a corresponding selection S_i. After all the small datasets D_i have been used (which could be done in parallel, as all of them are independent of each other), the combination method builds the final selection S as the result of the filtering process. To sum up, there are three main steps in this methodology:

1. Partition of the datasets.
2. Application of filtering to the subsets.
3. Combination of the results.

The partition of the dataset consists of dividing the original dataset into several disjoint subsets of approximately the same size that cover the full dataset (see Fig. 1). As mentioned in Section 1, in this research the partition is made vertically. Two different methods are used for partitioning the data: (a) performing a random partition and (b) ranking the original features before generating the subsets. The second option was introduced to try to improve the performance obtained by the first one. By having an ordered ranking, features with similar relevance to the class will be in the same subset, which will facilitate the task of the subset filter which