Knowledge-Based Systems 000 (2016) 1–19

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Centralized vs. distributed feature selection methods based on data complexity measures

L. Morán-Fernández*, V. Bolón-Canedo, A. Alonso-Betanzos

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

ARTICLE INFO

Article history:
Received 10 February 2016
Revised 6 September 2016
Accepted 26 September 2016
Available online xxx

Keywords:
Distributed learning
Feature selection
Data complexity measures
Classification

ABSTRACT

In the era of Big Data, many datasets have a common characteristic, the large number of features. As a result, selecting the relevant features and ignoring the irrelevant and redundant features has become indispensable. However, when dealing with large amounts of data, most existing feature selection algorithms do not scale well, and their efficiency may significantly deteriorate to the point of becoming inapplicable. Moreover, data is often distributed in multiple locations, and it is not economic or legal to gather it in a single site. For these reasons, we propose a distributed approach for partitioned data using two techniques: horizontal (i.e. by samples) and vertical (i.e. by features). Unlike than existing procedures to combine the partial outputs obtained from each partition of data, we propose a merging process using the theoretical complexity of these feature subsets. The novel procedure tested in 11 datasets has proved to be useful, showing competitive results both in terms of runtime and classification accuracy.

1. Introduction

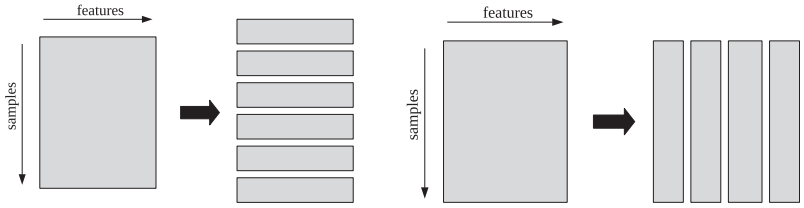
Terabytes of data are collected every second on Internet while scientific data from surveys and simulations are generated faster than researchers can analyze it. In this scenario, where data is not only big in volume, but also in complexity and variety, machine learning techniques face a major challenge since it is difficult to deal with a high number of input features due to the curse of dimensionality [1]. The scaling up problem appears in any traditional data mining algorithm when data size increases beyond capacity, damaging performance and efficiency. This issue can also affect negatively in some other aspects such as excessive storage requirements, increase of time complexity and, finally, generalization accuracy due to over-fitting and noise. To confront the problem of the extremely high dimensionality it is advisable to investigate the effects of the application of feature selection. The use of an adequate feature selection method can avoid over-fitting and improve model performance, providing faster and more cost-effective models and a deeper insight into the underlying processes that generated the data [2]. However, we will have to deal with a scability problem if we apply these techniques to large datasets due to their high computational complexities. The advantages of feature selection come at a certain price, as the search for a relevant feature subset introduces an extra layer of complexity. This new layer increases runtime and memory requirements, making algorithms very inefficient when they are applied to problems that involve very large datasets.

Traditionally, feature selection methods have been designed to run in a centralized computing environment. However, over the last few years many distributed methods have been developed instead of the centralized approaches. The first reason is that —with the advent of network technologies— there has been an increase in the size of datasets in all fields of application, but particularly affecting privacy concerns, data is often distributed across institutional, geographical and organizational boundaries, and it is not economic or legal to gather it in a single location. And, second, when dealing with large amounts of data, most existing feature selection algorithms do not scale well, and their efficiency may significantly deteriorate to the point of becoming inapplicable. Therefore, a possible solution might be to distribute the data, run a feature selection method on each subset of data and then combine the results. Data can be distributed either horizontally or vertically. In horizontal partitioning, the dataset is divided into several packets that have the same features as the original dataset, each containing a subset of the original instances (see Fig. 1(a)). In vertical partitioning, the original dataset is divided into several packets that have the same number of instances as the original dataset, each containing a subset of the original set of features (see Fig. 1(b)).

In this work, we will present a methodology in which several rounds of feature selection are performed on different partitions

* Corresponding author.
E-mail addresses: laura.moranf@udc.es (L. Morán-Fernández), vbolon@udc.es (V. Bolón-Canedo), ciamparo@udc.es (A. Alonso-Betanzos).

http://dx.doi.org/10.1016/j.knosys.2016.09.022
0950-7051/



(a) Horizontal partition (b) Vertical partition

Fig. 1. Techniques to partition data.

of the data. Then, the partial outputs are combined into a single subset of relevant features according to the theoretical complexity of these features using data complexity measures. These measures are a relatively recent proposal by Ho and Basu [3] to identify data particularities which imply some difficulty for the classification task beyond estimates of error rates. The experimental results on several datasets demonstrate important savings in runtime and satisfactory performance, since the classification accuracy matches in some cases is even better than when using the original feature selection algorithms over the whole datasets.

The remainder of this paper is organized as follows. Section 2 describes the background and early attempts to deal with distributed feature selection. Section 3 presents our proposed method to combine the partial outputs which makes use of data complexity measures. Section 4 provides a brief description of the specific datasets that will be dealt with, as well as the data complexity measures, the classification algorithms and the filters used to reduce data dimensionality. Section 5 describes an experimental study with the two techniques for partitioning the data. Section 6 presents several case studies in order to extract some recommendations on the appropriate approaches to use in specific scenarios. Finally, Section 7 contains our concluding remarks and proposals for future research.

2. Background

Although distributed learning is a fairly new field, it has been receiving a growing amount of attention since its inception. There exist in the literature several works to scale up datasets that are too large for machine learning in terms of samples. Chan and Stolfo [4] propose several meta-learning strategies for integrating independently learned classifiers by the same learner in a parallel and distributed computing environment. Experiments demonstrate that parallel learning by meta-learning can achieve comparable prediction accuracy in less space and time than purely serial learning. Other authors [5] have designed a partitioned distributed architecture and an efficient distributed association rule-mining algorithm based on the pattern tree called PC-tree. In Tsoumakas et al. [6], a new classifier combination strategy is presented. It scales up efficiently and achieves both high predictive accuracy and tractability of problems with high complexity. Kamalika et al. [7] have developed a local distributed privacy preserving algorithm for feature selection in large peer-to-peer environment. While not common, there are some other developments that distribute the data by features. In these works [8,9], authors described a novel ensemble approach, in which data is partitioned by features. Results show that this technique is simple, predicts almost as well as a centralized approach, reduces the amount of communication required, distributes computation and data access well, and allows each local site to keep its raw data private. Furthermore, Banerhee et al. [10] proposed a distributed privacy preserving method to perform feature selection that handles both horizontal and vertical data partitioning, whose efficiency has been demonstrated for real life datasets including time series data. Other works [11] present a feature selection method based on evolutionary computation which uses the MapReduce paradigm to obtain subsets of features from big datasets.

In Bolón-Canedo et al. [12], we presented a methodology for distributing the data vertically which combined partial feature subsets based on improvements in classification accuracy. Although the experiments showed that execution time was considerably shortened whereas performance was maintained or even improved compared to standard algorithms applied to the non-partitioned datasets, the drawback of this methodology was its dependence on the classifier used. In order to overcome this issue, a new framework for distributing the feature selection process [13,14] was proposed, which performed a merging procedure to update the final feature subset according to the theoretical complexity of these features, by using data complexity measures instead of the classification error. Such measures provides a basis for analyzing classifier performance beyond estimates of error rates. The most commonly employed measures are those proposed by Ho and Basu [3]. Lorena et al. [15] investigated the capability of the data complexity measures to explain the difficulty in the classification of cancer gene expression data before and after applying feature selection, demonstrating that this procedure could reduce the influence of data characteristics on classifier error rates. Macià et al. [16] proposed the characterization of datasets using data complexity measures, as helpful both in guiding experimental design and in explaining learner behavior. Luengo et al. [17] presented an automatic extraction method to determinate the domains of competence of a classifier using a set of data complexity measures. In this way, we provided a framework for distributed feature selection which not only was independent of the classifier, but also reduced drastically the computational time needed by the algorithm, thus paving the way for its application in high dimensional datasets.

Unlike in our previous works, in this paper we (a) include experimental results for both horizontal and vertical partitioning strategies; (b) perform a more intensive study, making use of several complexity measures to combine the partial rankings of features; (c) include datasets of different sample sizes, with different number of classes, and also microarray datasets, trying to check the behavior of the approach on high input dimensionality and (d) examine the effects of including different levels of overlap in the feature subsets.

3. Distributed feature selection based on complexity measures (DFS-CM)

Our proposed framework for distributed feature selection (DFS-CM) can be summarized in the three following stages:

1. Partition of the training datasets in several packets (by samples or features).
2. Application of the distributed algorithm to the subsets in several rounds.
3. Combination of the results into a single feature subset.

The pseudocode for the distributed algorithms is shown in Algorithm 1 (horizontal partitioning) and Algorithm 2 (vertical

Please cite this article as: L. Morán-Fernández et al., Centralized vs. distributed feature selection methods based on data complexity measures, Knowledge-Based Systems (2016), http://dx.doi.org/10.1016/j.knosys.2016.09.022