Knowledge-Based Systems 98 (2016) 1–29

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Tutorial on practical tips of the most influential data preprocessing algorithms in data mining

Salvador García[a,b,*], Julián Luengo[a], Francisco Herrera[a,b]

[a] Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
[b] Faculty of Computing and Information Technology North Jeddah, King Abdulaziz University, 21589 Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 24 April 2015
Revised 11 December 2015
Accepted 14 December 2015
Available online 21 December 2015

Keywords:
Data preprocessing
Data reduction
Missing values imputation
Noise filtering
Dimensionality reduction
Instance reduction
Discretization
Data mining

ABSTRACT

Data preprocessing is a major and essential stage whose main goal is to obtain final data sets that can be considered correct and useful for further data mining algorithms. This paper summarizes the most influential data preprocessing algorithms according to their usage, popularity and extensions proposed in the specialized literature. For each algorithm, we provide a description, a discussion on its impact, and a review of current and further research on it. These most influential algorithms cover missing values imputation, noise filtering, dimensionality reduction (including feature selection and space transformations), instance reduction (including selection and generation), discretization and treatment of data for imbalanced preprocessing. They constitute all among the most important topics in data preprocessing research and development. This paper emphasizes on the most well-known preprocessing methods and their practical study, selected after a recent, generic book on data preprocessing that does not deepen on them. This manuscript also presents an illustrative study in two sections with different data sets that provide useful tips for the use of preprocessing algorithms. In the first place, we graphically present the effects on two benchmark data sets for the preprocessing methods. The reader may find useful insights on the different characteristics and outcomes generated by them. Secondly, we use a real world problem presented in the ECDBL'2014 Big Data competition to provide a thorough analysis on the application of some preprocessing techniques, their combination and their performance. As a result, five different cases are analyzed, providing tips that may be useful for readers.

1. Introduction

Data preprocessing for Data Mining (DM) [48] focuses on one of the most meaningful issues within the famous Knowledge Discovery from Data process [57,149]. Data compilation is usually a relatively controlled task. Data will likely have inconsistencies, errors, out of range values, impossible data combinations, missing values or most substantially, data is not suitable to start a DM process. In addition, the growing amount of data in current business applications, science, industry and academia, demands to the requirement of more complex mechanisms to analyze it. With data preprocessing, converting the impractical into possible is achievable, adapting the data to accomplish the input requirements of each DM algorithm.

The data preprocessing stage can take a considerable amount of processing time [106]. Data preprocessing includes data preparation, compounded by integration, cleaning, normalization and transformation of data; and data reduction tasks, which aim at reducing the complexity of the data, detecting or removing irrelevant and noisy elements from the data through feature selection, instance selection or discretization processes. The outcome expected after a reliable connection of data preprocessing processes is a final data set, which can be contemplated correct and useful for further DM algorithms.

In an effort to identify some of the most influential data preprocessing algorithms that have been widely used in the DM community, we enumerate them according to their usage, popularity and extensions proposed in the research community. We are aware that each nominated algorithm should have been widely cited and used by other researchers and practitioners in the field. The selection of the algorithms is based entirely on our criteria and expertise, especially after the composition of a recent book on this topic [48]. The criteria considered are:

* Corresponding author at: Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain. Tel.: +34 958 240598; fax: +34 958 243317.
E-mail addresses: salvagl@decsai.ugr.es (S. García), julianlm@decsai.ugr.es (J. Luengo), herrera@decsai.ugr.es (F. Herrera).

http://dx.doi.org/10.1016/j.knosys.2015.12.006
0950-7051/

- Usage: the algorithm is frequently used in a previous step of a DM process or it is included in DM software packages.
- Referable: it must be described in a publication in the specialized literature.
- Popularity: the associated publication is considered as a highly cited one in well-known data bases: Web of Knowledge, Google Scholar, Scopus, etc.
- Standardization: the algorithm has been the baseline of inspiration of several modern and hybrid extensions.
- Smart: it must somehow incorporate a smart procedure in its definition, for the sake of not including direct and basic mechanisms as algorithms.
- Variability: there have been a minimum number of representatives belonging to each data preprocessing family.

From a practical point of view, we will carry out a practical study including numerous tips. It will be divided into two parts:

- We will include graphical representations of data preprocessed by the techniques reviewed in this tutorial. We will use the banana data set to illustrate most of the selected techniques, as it is a non-linearly separable 2D data set with two classes, often used for this purpose. For the techniques focused on dimensionality reduction (as Feature Selection and Space Transformations), the sonar data set is used instead, as it provides a large amount of features. The effects on the data and a comparison among related techniques for these data sets are thus easily attainable, as the effect on the data is visually depicted for every technique.
- We will consider a complex real-world problem from the ECDBL'14 Big Data competition [10,132]. This problem is related to bioinformatics, posing an appropriate scenario to show the importance of correctly applying several preprocessing techniques. In order to do so, we will select one preprocessing technique of each type, and we will combine them in order to solve different problems present in the data, and thus obtaining a benefit in the model obtained by a representative classifier as C4.5. By varying the order in which the preprocessing algorithms are applied and their parameters' values, five different cases are analyzed, along with the performance obtained by C4.5 in each step. From these five cases the reader will may find useful tips and insights that will help him/her to better understand the behavior and combination of preprocessing techniques.

The reader may consult and obtain the data sets, figures, results and complementary material in the website associated to this paper.[1]

The rest of the paper is organized according to the data preprocessing families, subfamilies and algorithms nominated. Thus, Section 2 will gather and present the preprocessing algorithms categorized by their type. Section 3 will show the effects of applying the preprocessing techniques with some illustrations, while Section 4 will illustrate the practical usage of the algorithms over a complex real-world data set. Finally, Section 5 will conclude the paper.

2. Most influential data preprocessing algorithms

In this section we present the most influential data preprocessing algorithms in Data Mining according to their relevance as previously stated. They are categorized as in shown in Fig. 1, attending to their type and purpose.

The section is organized following the categories illustrated in Fig. 1 as follows. Section 2.1 is devoted to imperfect data preprocessing algorithms, both including missing values and noise data techniques. Next Section 2.2 gathers the techniques used for data reduction. Finally, Section 2.3 addresses the preprocessing for imbalanced data.

2.1. Imperfect data

The data obtained from real-world problems is rarely clean and complete. The usage of techniques for either removing the noisy data or filling in the missing values (or even both) is common. In this section we tackle the preprocessing techniques used for completing (imputing) the missing values in Section 2.1.1 and for filtering the noise in Section 2.1.2.

2.1.1. Missing values imputation

Most techniques in DM rely on a data set that is supposedly complete. In order to extract and infer knowledge from the examples gathered, DM algorithms expect to process a series of complete instances sampled from the real-world. However, due to a faulty sampling process or limitations in the data acquisition process many existing, industrial and research data sets contain missing values (MVs). An MV is a value that has not been recorded in the data set for any reason, or that was never sampled.

Cleaning and preparing the data is required to make it clear and useful for the knowledge extraction process. Preprocessing the data is the most commonly used approach to perform this repairing task, and many options are available. Discarding the MVs is the simplest way to deal with them, but this approach is rarely practical. Only when the data has a low number of MVs should it be applied, and we must assure that the analysis carried out over the remaining complete examples will not produce an inference bias [83].

The presence of MVs in data analysis cannot be overlooked, and they usually create severe difficulties for practitioners. Handling the MVs in an inappropriate manner will lead to wrong conclusions taken from the knowledge extraction process [139]. As the main problems associated with MVs in the literature we may point out losses of efficiency in the extraction process, biases due to incomplete examples and complications in the data handling.

A first approach to deal with MVs is to discard them, either removing the incomplete examples or the attributes that present MVs. While the first approach is very usual, the second one is only applied in those cases in which the MVs are concentrated in a few attributes. This simplistic approach has the advantage of avoiding the modification of the posterior DM technique. However, blindly erasing the instances will probably result in a biased model, as the appearance of MVs can rarely be ignored. Traditional solutions to MVs come from statistics, and they are based on the use of maximum likelihood procedures. Starting from a probability model for the data, its parameters' values are adjusted to better fit the data set by taking into account the distributions and nature of the MVs, while sampling the probability functions in order to obtain values to fill in the MVs. Since the real probability models for a given data set are rarely known, maximum-likelihood methods are difficult to be correctly applied. On the other hand, the use of machine learning techniques is a straightforward solution to induce a model without providing details of the data distribution. The ultimate goal of these techniques is to provide estimates of the MVs, commonly known as imputations, and thus they are also known as imputation techniques. Please note that maximum likelihood methods also provide an imputation of the values, but these imputations are usually part of the model adjusting process and then exploited when the model convergence has stagnated.

We will focus our attention on maximum likelihood and imputation methods. A broad family of imputation methods is available, from simple imputation techniques like mean substitution, *k*NN,

[1] http://sci2s.ugr.es/most-influential-preprocessing .