Information and Software Technology 67 (2015) 108–127

Contents lists available at ScienceDirect

Information and Software Technology

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

An empirical analysis of data preprocessing for machine learning-based software cost estimation

Jianglin Huang [a,*], Yan-Fu Li [b], Min Xie [a]

[a] Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong
[b] Department of Industrial Engineering, CentraleSupelec, Paris, France

ARTICLE INFO

Article history:
Received 28 August 2014
Received in revised form 30 June 2015
Accepted 6 July 2015
Available online 13 July 2015

Keywords:
Software cost estimation
Data preprocessing
Missing-data treatments
Scaling
Feature selection
Case selection

ABSTRACT

Context: Due to the complex nature of software development process, traditional parametric models and statistical methods often appear to be inadequate to model the increasingly complicated relationship between project development cost and the project features (or cost drivers). Machine learning (ML) methods, with several reported successful applications, have gained popularity for software cost estimation in recent years. Data preprocessing has been claimed by many researchers as a fundamental stage of ML methods; however, very few works have been focused on the effects of data preprocessing techniques.
Objective: This study aims for an empirical assessment of the effectiveness of data preprocessing techniques on ML methods in the context of software cost estimation.
Method: In this work, we first conduct a literature survey of the recent publications using data preprocessing techniques, followed by a systematic empirical study to analyze the strengths and weaknesses of individual data preprocessing techniques as well as their combinations.
Results: Our results indicate that data preprocessing techniques may significantly influence the final prediction. They sometimes might have negative impacts on prediction performance of ML methods.
Conclusion: In order to reduce prediction errors and improve efficiency, a careful selection is necessary according to the characteristics of machine learning methods, as well as the datasets used for software cost estimation.

1. Introduction

Software *project managers often need to* estimate the cost/effort of developing a software system at the early stage of its life-cycle [1] in order to plan the project management activities. The ability to accurately estimate the development cost plays an important role in the success of software project management. In the past decades, numerous studies have been published on software cost estimation (SCE) methods, which can be classified into the following three main categories.

1. *Expert judgment:* It requires the consultation of one or more experts to derive the cost estimate. With the experience and available information of past projects and the understanding of a new project, the experts could obtain the estimation by a non-explicit and subjective reasoning process.
2. *Parametric models:* They often involve the utilization of analytical or statistical equations relating software project cost to a number of project features. The well-known ones include COCOMO [2] and SLIM Model [3].
3. *Machine learning (ML) methods*: They involve at least one modeling method, taking a number of project features and producing a cost prediction, making no or minimal assumptions about the form of the relation under study. Thus they can provide higher approximation capabilities to solve complex problems. Recently, they have been adopted as an alternative or together with the first two methods [4–7]. Representative ML methods include artificial neural networks (ANN) [6,8,9], case-based reasoning (CBR) [9,10] (also referred to as analogy-based estimation [11,12] or estimation by analogy [13]), and classification and regression trees (CART) [5,14,15].

When targeting estimation accuracy, considerable effort has been devoted to improving ML methods [1,16–21]. For the empirical validations, ML algorithms are routinely tested on the SCE datasets. Data preprocessing (DP) is a fundamental stage of the

* Corresponding author. Tel.: +852 56013641.
E-mail addresses: jianhuang7-c@my.cityu.edu.hk (J. Huang), yanfu.li@centralesupelec.fr (Y.-F. Li), minxie@cityu.edu.hk (M. Xie).

http://dx.doi.org/10.1016/j.infsof.2015.07.004
0950-5849/

ML application, which has been reported to have significant impacts onto the performances of ML methods [18].

To the knowledge of the authors, there is very few research work focused on the DP techniques in the SCE literature. In many situations the DP techniques, such as feature selection (FS) [7,22–24] and case selection (CS) [4,11,12,25], have been considered as a necessary step for CBR while for other ML methods, such as ANN and CART, they might be ignored. In the literature some studies focus on analyzing DP techniques. Strike et al. [26] simulated various incomplete data and found that the best regression model could be obtained from missing-data imputation with Z-score standardization. The combination of scaling scheme and missing-data treatment (MDT) is firstly analyzed; however, their impacts onto the ML method were not studied. Many studies propose one or more DP techniques to deal with a specific issue in SCE, such as data missingness [27–29], redundant or irrelevant features [10,25], or abnormal cases [30,31]. But they did not study the effectiveness of different DP techniques. Keung et al. [32] first time concluded that the performance of a ML method could be significantly altered by a DP technique, such as scaling and FS. But the number of DP techniques they considered is limited and the effectiveness of combined DP techniques are not investigated.

From the analysis above, an empirical study on multiple DP techniques for ML methods is needed to promote much more careful use of the DP techniques rather than taking one or more DP approaches as granted. The obtained results would be beneficial to the future ML-based SCE studies.

The rest of this paper is organized as follows: Section 2 presents a literature survey on DP applications; Section 3 presents the four datasets used in this study, an overview on the ML methods (*i.e.* ANN, CBR and CART), and the experimental design; Section 4 presents the experiment results and analysis; Section 5 discusses the threats to four types of validity; Section 6 concludes this work and points out future research directions.

2. Related work

2.1. Literature survey

The application of ML algorithm requires the presence of data in a mathematically feasible format through data preprocessing. DP techniques consist of data reduction, data projection and missing-data treatment. Data reduction aims to decrease the size of the datasets by means of feature selection (FS) or case selection (CS). Data projection intends to transform the appearance of the data, *e.g.* scaling, which scales all features into a predefined same range. Missing-data treatments (MDTs) include deleting missing values [12,15,16,33,34] and/or replacing them with the estimates [13,35]. Moreover, the logarithm transformation [36,37] is frequently applied for linear regression to retain the normality assumption for the correct implementation of linear regression. It is indeed an important step for regression models to ensure the normality of the residual [38–41]. On the other hand, logarithm does not frequently appear in ML studies. In our survey, there are only three publications [32,42–44] that clearly used logarithm for ML methods. This study aims to investigate the effectiveness of the mainstream DP techniques for ML methods. Consider both scaling and logarithmic transformation could help reduce ranges, we choose to include scaling as a candidate DP method in our experiments.

To reveal the situations of DP technique utilization in the literature, we first conduct a survey of relevant ML papers from 2005 to present published on the following journals: *IEEE Transactions on Software Engineering* (*IEEE TSE*), *Empirical Software Engineering* (*ESE*), *Journal of Systems Software* (*JSS*), *Information and Software Technology* (*IST*), and *Software Quality Journal* (*SQJ*), and the following major conference proceedings: *International Conference on Software Engineering* (*ICSE*), *International Symposium on Empirical Software Engineering and Measurement* (*ESEM*), and *International Conference on Predictive Models in Software Engineering* (*PROMISE*). Both the individual studies of ML methods and the comparative studies (within ML methods or between ML and other methods) are included. We summarize the publications according to the ML methods and the DP techniques applied. In specific, we explore the use of MDT, scaling, FS and CS. These 48 publications are presented in Table 1. It is shown that most publications have employed certain DP techniques. 12 works [4,8,11,44–52] only mention single step of DP. Table 1 also shows that many studies use combined DPs. For examples, there are 7 of totally 48 works combined only scaling and FS/CS [17,24,31,53–56], and 7 of 48 works combined only MDTs and FS/CS [16,20,33,57–60]. FS and CS have been considered as a necessary step for CBR in several studies [4,11,20,23,30,34,48,49,53,56–58,61–64]. However, there is no empirical study to investigate more DPs and their combinations.

The following section presents an overview on the four types of DP methods and summarizes the evidence and arguments in the literature that lead to the research questions of this study and serve as the foundation for the empirical work.

2.2. Data preprocessing techniques

2.2.1. Missing-data treatments

Due to the high cost of gathering and reporting data from projects, development teams are less focused on data collection [28]. The incomplete datasets also frequently appear across the SCE studies (*e.g.* the ISBSG database and PROMISE datasets) [13,21,28,29,34,59,72]. The missing values have significant impacts on ML estimation performances, as reported by [19,28,46].

There are many MDTs in the literature. They often include: deletion methods (*listwise deletion* and *pairwise deletion* [28,50,71,72]), and imputation methods (*mean imputation*, *hot-deck imputation*, *cold-deck imputation*, *regression imputation*, etc.) [13,20,35,56,57,61]. It is noted that the deletion methods, especially listwise deletion (LD), widely used as a default approach for dealing with missing values, can result in discarding large proportions of datasets in cases and introducing biasness [28,34]. As another solution of MDT, imputation requires more extensive and complicated statistical and computational analysis [26,28] and also includes natural prediction error [34]. Mean imputation (MI) imputes each missing value with the mean of observed values and preserves the information of data. However, as the simplest imputation method it may cause to diminish the variance of variables [26].

According to results of our survey in Table 1, LD is the most popular method followed by MI. Particularly, 13 works [12,15,16,21,31,33,34,58–60,64,66,68] regarded LD as the default DP method for missing values. However, some studies show that MI or *k*-NN imputations are better than LD [26–28,73]. In this study, we will validate the superiority of MI over LD.

2.2.2. Scaling

Scaling generally refers to measurements or assessments conducted under exact, specified and repeatable conditions. In ML, scaling transforms feature values according to a defined rule so that all scaled features have the same degree of influence [36] and thus the method is immune to the choice of units [71], which is a major stage for ML methods. Normally, the intervals of [0,1] and [−1,1] are used to be the target of scaling, as shown in Eq. (1).