Information and Software Technology 55 (2013) 1710–1725

Contents lists available at SciVerse ScienceDirect

Information and Software Technology

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

AREION: Software effort estimation based on multiple regressions with adaptive recursive data partitioning

Yeong-Seok Seo [a,*], Doo-Hwan Bae [a], Ross Jeffery [b]

[a] Department of Computer Science, College of Information Science & Technology, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea
[b] NICTA, Locked Bag 9013, Alexandria, NSW 1435, Australia

ARTICLE INFO

Article history:
Received 13 March 2012
Received in revised form 15 March 2013
Accepted 16 March 2013
Available online 4 April 2013

Keywords:
Software project management
Software cost estimation
Software effort estimation
Data distribution
Least squares regression
Adaptive recursive data partitioning

ABSTRACT

Context: Along with expert judgment, analogy-based estimation, and algorithmic methods (such as Function point analysis and COCOMO), Least Squares Regression (LSR) has been one of the most commonly studied software effort estimation methods. However, an effort estimation model using LSR, a single LSR model, is highly affected by the data distribution. Specifically, if the data set is scattered and the data do not sit closely on the single LSR model line (do not closely map to a linear structure) then the model usually shows poor performance. In order to overcome this drawback of the LSR model, a data partitioning-based approach can be considered as one of the solutions to alleviate the effect of data distribution. Even though clustering-based approaches have been introduced, they still have potential problems to provide accurate and stable effort estimates.
Objective: In this paper, we propose a new data partitioning-based approach to achieve more accurate and stable effort estimates via LSR. This approach also provides an effort prediction interval that is useful to describe the uncertainty of the estimates.
Method: Empirical experiments are performed to evaluate the performance of the proposed approach by comparing with the basic LSR approach and clustering-based approaches, based on industrial data sets (two subsets of the ISBSG (Release 9) data set and one industrial data set collected from a banking institution).
Results: The experimental results show that the proposed approach not only improves the accuracy of effort estimation more significantly than that of other approaches, but it also achieves robust and stable results according to the degree of data partitioning.
Conclusion: Compared with the other considered approaches, the proposed approach shows a superior performance by alleviating the effect of data distribution that is a major practical issue in software effort estimation.

1. Introduction

Software effort estimation has always been a challenge for software engineering communities. Accurate and reliable software effort estimation is important for allocating resources and creating a reasonable schedule during software project planning. Underestimating software project effort causes schedule delays and cost over-runs, which, in the end, can lead to project failure. Conversely, overestimating software project effort can also be detrimental in effectively utilizing software development resources. Thus, over the course of the last decade, a number of software effort estimation methods have been developed by using different theoretical concepts [1] and combining existing effort estimation methods [2,3].

Among existing methods, Least Squares Regression (LSR) is one of the widely studied methods [3–5] and is known to provide competitive accuracy for effort estimation [6,7]. However, an effort estimation model by LSR, a single LSR model, is highly impacted by the distribution of historical software project data. Specifically, if the data set is scattered and the data points[1] are located inconsistently away from the single LSR model, then the model does not work properly.[2] This problem is generally caused in widely scattered data sets rather than data sets with small scatter. That is, this is escalated typically for data sets collected from a variety of organizations, such as the ISBSG data set [8], which exhibit high data dispersion. In

* Corresponding author. Tel.: +82 42 350 8489; fax: +82 42 350 8488.
E-mail addresses: ysseo@se.kaist.ac.kr (Y.-S. Seo), bae@se.kaist.ac.kr (D.-H. Bae), Ross.Jeffery@nicta.com.au (R. Jeffery).

[1] A data point refers to one software project data in a historical software project data set.
[2] Note that even though the data set is scattered, the data points in the tail of a skewed distribution still sit on the LSR model that fits the main body of the distribution, then the model will perform more than usefully.

0950-5849/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2013.03.007

this case, a single LSR model usually leads to poor accuracy of effort estimation[3] [9,10], without the use of defensible subsets of the data being used in model building. So the issue being addressed here is one of looking to make use of the entire available data set while alleviating the issue of high data dispersion.

In order to alleviate the effect of data distribution on LSR, we can consider the following three approaches: (1) a robust regression-based approach [11,12], (2) an outlier elimination-based approach [13–15], and (3) a data partitioning-based approach [16–18] for the generation of multiple LSR models. The robust regression-based approach is used to build a single LSR model that is more resilient to influential data points (e.g., outliers). Furthermore, the outlier elimination-based approach is used to eliminate outliers in advance to prevent model distortion before building the single LSR model. Although these approaches overcome the data distribution problem to which the naïve use of a single LSR model is vulnerable, they do not significantly improve the accuracy of effort estimation [13,14]. This is because the single LSR model still cannot provide accurate effort estimates for data points that lie largely outside the model [19]. As an alternative to the single LSR model, data partitioning-based approaches have been proposed to generate multiple regression models. These approaches focus on using different regression models for different parts of a data set to improve the accuracy of effort estimation even in a widely scattered data set. In existing studies, a software project data set is partitioned by clustering algorithms that consider data similarities based on data distribution, and a regression model is then built on each cluster. Although these approaches have demonstrated more accurate effort estimates in comparison to that achieved by the single LSR model, they can still potentially be poorly accurate if the data points are not partitioned into proper clusters for achieving accurate effort estimates [20]. That is, the results can fluctuate according to the number of clusters that have been selected in the data set.[4] However, the most significant problem encountered during clustering is that it is difficult to determine the qualified number of clusters for improving the accuracy of effort estimation achieved by the regression model. Thus, these approaches sometimes provide unfavorable results even in a single-company data set that has a relatively small amount of scatter in its data distribution[5] [19] (the previous studies also present the importance of the selection of the qualified number of clusters and discuss the likelihood of the improper clusters for achieving accurate effort estimates by the selected number of clusters [19,20]. Note that although there are cluster validity indices that are designed to assess goodness of clusters, the recommended number of clusters can be different in each index [21,22]). Moreover, a reliable number of data points to build regression models should be considered to guarantee the reliability of the models. If the reliability of the model is not guaranteed, then the effort estimation results are not reliable as well. For example, if one of the clusters includes a small number of data points (e.g., 2 or 3) after clustering has been completed on a data set, then it is not appropriate to build a regression model using the data points because there is a risk of overfitting [23].

In order to overcome the drawbacks of the data partitioning-based approaches and improve the performance of the LSR model, we propose a new data partitioning-based approach, Adaptive REcursive data partitIONing (AREION), for generating multiple reliable LSR models by mitigating the effect of data distribution. The basic idea of AREION is to generate data partitions that consist of data points that are well estimated by the LSR models. AREION recursively partitions a data set according to a threshold value that is determined as an accuracy indicator for the data partition, including a procedure to guarantee the reliability of the LSR model. Based on the derived LSR model for each data partition, the effort for a new software project is obtained by the selective use of the effort estimates from the models. With AREION, we can obtain an effort prediction interval (PI) that is useful to describe the uncertainty of the estimates [24,25], as well as an effort estimate.

We have focused on one of the practical issues in estimation; that of the data distribution strongly affecting the performance of the effort estimation models. By alleviating the effect of data distribution, AREION can provide sound effort estimates in addition to the effort prediction interval that is useful to set software project budgets devoted to risk management and to avoid unfair criticism in situations with high uncertainty [24,25]. Thus, AREION can help to achieve efficient software project planning and resource allocation. Furthermore, in our empirical experiments, we have tried to guide practitioners to the degree of a data partitioning for deriving more accurate effort estimates. Finally, we have used real industrial data sets in this work. This can be a better reference for a practical use of the approaches.

The remainder of this paper is organized as follows: Section 2 introduces the related work. Section 3 explains the characteristics of the most common evaluation criteria for the software effort estimation model as background to our work. Section 4 generally describes the proposed approach. Section 5 shows the experimental design and the intermediate results from the experiments, and Section 6 presents the experimental results and analysis. Section 7 discusses threats to validity. Finally, Section 8 concludes this paper and suggests future work.

2. Related work

In order to improve the accuracy of effort estimation in the single regression model, there have been several data partitioning-based studies on deriving multiple regression models [16–18]. These studies focus on overcoming weaknesses that are common when using a single regression model, such as poor model fitting and low accuracy of effort estimation in software project data sets coming from heterogeneous projects.

Cuadrado-Gallego et al. [16,17] proposed an approach to generating multiple regression models through clustering using the Expectation–Maximization (EM) algorithm, and they analyzed the influence of two process-related attributes as drivers of clustering: the use of CASE tools and the use of methodologies. According to the experimental results validated with the ISBSG (Release 8) data set, the accuracy of effort estimation that was achieved via the multiple regression models was highly improved compared to that of the single model. For example, in [17], MMRE and Pred(0.3) were used as evaluation criteria for the models, and MMRE of the single model and the multiple models through clustering were 2.17 and 1.03, respectively. Similarly, Pred(0.3) of the single model and the multiple models through clustering were 26.75 and 35.60, respectively. Although the results were not accurate enough to achieve the desired software effort estimate, they showed the possibility of improving the accuracy of effort estimation via multiple models.

Aroba et al. [18] presented an approach to generating multiple standard LSR models based on fuzzy clustering. The use of fuzzy clustering allows generating different LSR models for each cluster and also allows data points to be contained in more than one cluster with different degrees of fuzziness (membership value). Taking

[3] The criticism being leveled at LSR here generally focuses on the weak point caused when conducting actual application of LSR for software effort estimation, rather than the inadequacy of LSR itself.

[4] Note that the number of clusters is closely related to the dispersion of data points within/across the clusters. That is, the dispersion is determined according to the selection of the number of clusters.

[5] All of the single-company data sets are not always inherently homogeneous. Note that there are some software organizations that provide bespoke solutions across a wide range of domains, countries, and so on.