The International Journal of Engineering and Science (IJES)
|| Volume || 8 || Issue || 10 Series I || Pages || PP 18-21 || 2019 ||
ISSN (e): 2319 – 1813 ISSN (p): 23-19 – 1805

Incorporating feature derivation into machine learning to predict software project duration

Qingkang Zhu, Hongbo Li *
School of Management,Shanghai University, China
** Corresponding author: Hongbo Li*

----------***ABSTRACT***----------
A common problem in software development is that many projects cannot be finished on time. Inaccurate estimation of the project duration is one important reason to this phenomenon. In view of this, we incorporate feature derivation into three machine learning algorithms (i.e., multiple linear regression, multi-layer perceptron and decision tree) to predict software project duration as accurately as possible. We use cross-validation to compare the performance of the duration prediction algorithms. Two evaluation criteria are adopted to measure the algorithms: mean magnitude of relative error (MMRE) and PRED(25%). Based on data of143 projectsthat are collected from an open source code management platform, we conduct experiments to test the duration prediction algorithms.
KEYWORDS-*Software project duration prediction; Feature derivation; Machine learning*

Date of Submission: 30-09-2019 Date of acceptance: 15-10-2019

I. INTRODUCTION

Software projects suffer from time delays and budget overruns worldwide. A report shows that only 39% of software development projects are completed within the planned deadline (Chaos Report, 2013[1]). Software project duration determines the cost and resource allocation. Therefore,accurate prediction of software project duration will greatly reduce cost and improve resource efficiency. However, due to the lack of relevant information and the unclear definitions of requirements at the planning stage of the software project, the prediction of software project durationis verydifficult. In view of this, we incorporate feature derivation into machine learning algorithms to predict software project duration as accurately as possible.

In the next section, we give a literature review on existingsoftware project duration prediction methods.Section 3explainsthe duration prediction algorithms adopted in this paper. Section 4 describesdata preprocessing, feature derivation and algorithm performance measure. experimental results are presented in the Section 5. The last section concludes the paper.

II. RELATED RESEARCH

The prediction of software project duration has always been one of the most difficult problems. Existing studies mainly focus on using expert judgment, parameter models and machine learning algorithms to estimate software duration.

Zapata et al.(2013) use expert judgment techniques to analyze the relationship between budget, duration and funding[2]. Berlinet al.(2009) compare the estimation capacity of statistical regression and multi-layer perceptron algorithm for software project duration on the basis of excluding outliers[3]. They conclude that there is no significant difference in the estimation accuracy of the two algorithms on two datasets at 95.0% confidence level. Lopez-Martin et al.(2013) applymulti-layer perceptron and statistical regression algorithm to estimate software project duration based on ISGSB R11 dataset[4]. Wang et al. (2012) use the multi-layer perceptron and support vector machine to estimate software project duration. The results of their experiments show that the multi-layer perceptron algorithm has better estimation accuracy than the other two algorithms[5]. Lopez-Martin et al.(2015) input datasets of ISGSB into the multi-layer feedforward neural network algorithm, radial basis function neural network algorithm and multivariate linear regression algorithm, respectively[6]. Cross-validation results show that the multi-layer perceptron algorithm has better prediction performance than the other two algorithms. Przemyslaw et al.(2018) propose a practical scheme for software duration and effort estimation based on multi-layer perceptron, support vector machine and linear regression algorithm [7].

It is worth noting thatprior predictionalgorithms mainly use the features directly given by the dataset. Few studies use the idea of feature derivation and ignore the impact of new features on the prediction of software project duration. In addition, few studies compare the traditional parametric algorithms(e.g., constructive cost model(COCOMO)) with machine learning algorithms when estimating software project duration. Therefore, in

this paper, based on the existing features, the idea of feature derivation[8, 9] is used to derive a new feature that has an impact on project duration and three machine learning algorithms are compared with the traditional parameter algorithm COCOMO.

III. TECHNIQUES

To predict software project duration, the following 4 algorithms are used:constructive cost model(COCOMO), multiple linear regression(MLR), multi-layer perceptron(MLP) and decision tree(DT).

Constructive Cost Model(COCOMO)

COCOMO is aneasy-to-use and accurate cost-estimation method based on parameter factors, which was first proposed by Boehm in 1981. Its essence is a parameterized estimation method for software projects. Because software projects usually record lines of codeor function points in the process of development which conforms to the research scope of COCOMO, so it can take these factors as parameters to predict software cost and duration. This algorithm has become a common method for software effort and duration prediction since it was proposed. The next two formulas are used to calculate software effort and duration in COCOMO algorithm based on the lines of code.

$$E = a * KLOC^b \quad (1)$$

$$D = c * E^d \quad (2)$$

WhereEisthe software effort, Disthe software project duration, KLOC represents thousands of lines of code for a software project, and the remaining a, b, c and d are parameters of COCOMO. The values of these parameters[10] are shown in Table 1.

Table 1 Parameters of COCOMO

Project type	a	b	c	d
Foundation project	2.4	1.05	2.5	0.38
Semi-independent project	3.0	1.12	2.5	0.35
Embedded Project	3.6	1.20	2.5	0.32

Multiple Linear Regression(MLR)

There are two or more independent variables in the MLR, which has the following form:

$$D = \alpha + \beta_1 x_1 + \beta_2 x_2 \cdots + \beta_n x_n + \varepsilon \quad (3)$$

Where$x_i (i = 1, \ldots, n)$ denotes the value of multiple independent variables, D denotes the target variable, α is a constant, $\beta_1, \beta_2 \cdots, \beta_n$ denotes the regression coefficients of the corresponding independent variables and ε is a random value that cannot be observed directly.Most studies use least squares method, generalized least squares method or gradient descent method to obtain regression coefficients.

Multi-Layer Perceptron (MLP)

MLP is an extension of the perceptron model, which overcomes the shortcomings that perceptron cannot recognize data of linear non-separable. MLP is a trend-structured artificial neural network, which maps a set of input vectors to a set of output vectors. MLP consists of a set of neurons constituting the input layer, one or more hidden layers and one output layer. In addition to the input nodes, each node is a neuron (or processing unit) is associated with a non-linear activation function.

The back propagation algorithm is often used to train the MLP. Meanwhile, MLP is also a parameterized algorithm and focuses on recognizing local minimum value rather than global minimum. Because it consists of hidden layer and deviation parameters, MLP is robust relative to noisy data[11].However, in the extended training process, it may have the problem of over-fitting[7].

Decision Tree (DT)

DT is a divide-and-conquer decision-making process. DT regression algorithms (i.e., CART)can be used to solve prediction tasks. The process for constructinga DT can be divided into two steps: the first step is the generation of DT, which is process of generating the structure of DT from training datasets. The second step isDT pruning.DT pruning is the process of checking and correcting the DT generated in the previous stage. It uses the data from test datasets to verify the preliminary rules generated before and prune the branches that affect the accuracy of pre-balancing. The main advantages of DT are readability and fast regression speed.

IV. EXPERIMENTSETUP

DataSets and Data Preprocessing

We collected data of 143 projects from an open source code management platform.The missing data is replaced by 0.The abnormal data is eliminated by the box chart method.

There are four direct features in the dataset: the number of developers, thousand lines of code, the start date and the latest update date of the software project. In addition, we created a new feature based on the idea of feature derivation: thousand lines of code per person. We define the software project that has not been updated in the last six months as completed project. Therefore, the software duration is equal to the latest update date minus the start date.

Therefore, the number of developers, thousand lines of code and thousand lines of code per person constitute the feature set forour software duration prediction algorithms. Table 2 presents the range of values for each feature.

Table 2 Features and their values

Features	**Range of values**
Input:	
number of developers	2-82
thousand lines of code	3.02-481.41
thousand lines of code per person	0.13-240.7
Output:	
Software project duration	13.07-141.43 months

Cross-Validation

The "Leave-one-out" method has better performance on small-scale datasets and issuitable for the evaluation of software project duration prediction[6][12].Therefore, we select "Leave-one-out " method as the evaluation method. In each experiment, one project data sample is selected as test set, and the remaining are training set. After 143 experiments, the predicted data can be used to compare the performance of different algorithms.

Evaluation Criteria

Mean magnitude of relative error (MMRE) and PRED(25%)are chosen as indicators to measure the prediction performance of different algorithms.To get MMRE, we first calculate MRE as follows:

$$\mathrm{MRE_i} = \frac{\left|\mathrm{D_i} - \widehat{\mathrm{D}}_\mathrm{i}\right|}{\mathrm{D_i}} \qquad 4)$$

Where $\mathrm{D_i}$and $\widehat{\mathrm{D}}_\mathrm{i}$are the real and predicted durations, respectively. According to the "Leave-one-out" method, 143 prediction value are obtained from 143 experiments. Then, the MMRE of 143 experiments is calculated according to (5).

$$\mathrm{MMRE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{MRE_i} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{\left|\mathrm{D_i} - \widehat{\mathrm{D}}_\mathrm{i}\right|}{\mathrm{D_i}} \qquad (5)$$

The value of PRED(25%)is equal to the number of MRE indicators greater than 0.25 divided by the total number of samples:

$$\mathrm{PRED}(25\%) = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \begin{cases} 1 & \text{if } \mathrm{MRE_i} \le 0.25 \\ 0 & \text{otherwise} \end{cases} \qquad 6)$$

V. EXPERIMENTAL RESULTS

Table 3 presents the duration prediction results of four algorithms and the best results are shown in bold. It can be seen that: (1) Although MLP performs well in some other duration prediction literature, it shows poor prediction performance for our dataset. (2) Compared with MLR and DT, the MMRE of COCOMO is larger andPRED(25%) is smaller, which means that most of the machine learning algorithms have better prediction performance than COCOMO. (3) The PRED(25%) of MLR is the largest, and the MMRE of DT is the smallest. Therefore, when facing the problem of software project duration prediction, MLR and DT should be given priority, and we recommend to combine them.

Table 3 Software project duration prediction results

Algorithms	**MMRE**	**PRED(25%)**
COCOMO	0.73	0.03
MLR	0.57	**0.38**
MLP	0.93	0.01
DT	**0.51**	0.35

In order to see the value of incorporating feature derivation into machine learning algorithms. Table 4 shows the duration prediction results with and without feature derivation. The numbers shown in bold indicate that

incorporating feature derivation into MLR do improve its performance. For other algorithms, the impact of feature derivation is not obvious.

Table 4 Comparison results with and without feature derivation

Algorithms	MMRE (with feature derivation)	MMRE (without feature derivation)	PRED(25%)(with feature derivation)	PRED(25%) (without feature derivation)
MLP	0.925	0.933	0.014	0.014
MLR	**0.567**	0.572	**0.378**	0.349
DT	0.511	0.511	0.350	0.350

VI. CONCLUSIONS AND FUTURE RESEARCH

This paper incorporates feature derivation into machine learning algorithms to predict software project duration. Leave-one-out cross-validation are used to compare the performance of the duration prediction algorithms. Two evaluation criteria are adopted to measure the algorithms: MMRE and PRED(25%). Experimental results on data of 143 projects collected from an open source code management platform shows that multiple linear regression and decision tree techniques outperform the traditional COCOMO method. We also find that incorporating feature derivation into multiple linear regression can improve its performance. In the future, more data need to be used to test the duration prediction algorithms.

REFERENCE

[1]. Carroll Cf. IT Success and Failure—the Standish Group CHAOS Report Success Factors [J]. Chris F Carroll, 2013.
[2]. Zapata Andres H, Michel Rv Chaudron. An empirical study into the accuracy of IT estimations and its influencing factors [J]. International Journal of Software Engineering and Knowledge Engineering, 2013, 23(04):409-432.
[3]. Berlin Stanislav, Tzvi Raz, Chanan Glezer, Moshe Zviran. Comparison of estimation methods of cost and duration in IT projects [J]. Information and software technology, 2009, 51(4):738-748.
[4]. López-Martín Cuauhtémoc, Arturo Chavoya, Maria Elena Meda-Campaña. Use of a feedforward neural network for predicting the development duration of software projects [C].2013 12th International Conference on Machine Learning and Applications. 2013. IEEE, 156-159.
[5]. Wang Yu-Ren, Chung-Ying Yu, Hsun-Hsi Chan. Predicting construction cost and schedule success using artificial neural networks ensemble and support vector machines classification models [J]. International Journal of Project Management, 2012, 30(4):470-478.
[6]. López-Martín Cuauhtémoc, Alain Abran. Neural networks for predicting the duration of new software projects [J]. Journal of Systems and Software, 2015, 101:127-135.
[7]. Pospieszny Przemyslaw, Beata Czarnacka-Chrobot, Andrzej Kobylinski. An effective approach for software project effort and duration estimation with machine learning algorithms [J]. Journal of Systems and Software, 2018, 137:184-196.
[8]. Pan Zhisong, Siqi Tang, Junyang Qiu, Guyu Hu. Survey on Online Learning Algorithms [J]. Journal of Data Acquisition & Processing, 2016, 31:1067-1082.
[9]. Dimitrov Alexander G, Tomas Gedeon, Brendan Mumey, Ross Snider, Zane Aldworth, Albert E Parker, John P Miller. Derivation of natural stimulus feature set using a data-driven model [C].International Conference on Computational Science. 2003. Springer, 337-345.
[10]. Boehm Barry, Bradford Clark, Ellis Horowitz, Chris Westland, Ray Madachy, Richard Selby. Cost models for future software life cycle processes: COCOMO 2.0 [J]. Annals of software engineering, 1995, 1(1):57-94.
[11]. Larose Daniel T. Data mining and predictive analytics [M]. John Wiley & Sons. 2015.
[12]. Kocaguneli Ekrem, Tim Menzies. Software effort models should be assessed via leave-one-out validation [J]. Journal of Systems and Software, 2013, 86(7):1879-1890.

Hongbo Li" Incorporating feature derivation into machine learning to predict software project duration" The International Journal of Engineering and Science (IJES), 8.10 (2019): 18-21