International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.9 (2015), pp.1-8
http://dx.doi.org/10.14257/ijmue.2015.10.9.01

Comparison of Machine Learning Algorithms for Software Project Time Prediction

WanJiang Han[1], LiXin Jiang[2], TianBo Lu[1] and XiaoYan Zhang[1]

[1]*School of Software Engineering, Beijing University of Posts and Telecommunication, Beijing 100876, China*
[2]*Department of Emergency Response, China Earthquake Networks Center, Beijing 100036, China*
hanwanjiang@bupt.edu.cn

Abstract

Software Project Management (SPM) is one of the primary factors to software success or failure. Prediction of software development time is the key task for the effective SPM. The accuracy and reliability of prediction mechanisms is also important. In this paper, we compare different machine learning techniques in order to accurately predict the software time. Finally, by comparing the accuracy of different techniques, it has been concluded that Gaussian process algorithm has highest prediction accuracy among other techniques studied. Experimental results show this prediction approach is more effective.

Keywords: *Software time prediction, Software Project Management, Machine Learning Algorithm*

1. Introduction

A software project is more difficult to visually monitor and control than project that creates a physical product like a building that you can see. For software projects, it becomes difficult to meet the deadlines and to deliver the product features "as promised" and within the budget. Before starting a project, software project planning is one of the most critical activities in modern software development. Without a realistic and objective plan, a software development project cannot be managed in an effective way [1-2].

When historical project effort data, duration data and other project data are used for effort or duration prediction, the basic motivation behind this attempt is that the knowledge stored in the historical datasets can be used to develop predictive models, by either statistical methods such as linear regression and correlation analysis or machine learning methods such as ANN (Artificial Neural Network) and SVM (Support Vector Machine) [3], to predict the effort or duration of projects [4].

In order to develop prediction-based software mechanisms, it is required a high-accurate prediction function. So, prediction approaches must to be able to predict project effort and duration, with high accuracy.

Prediction-based approaches require a prediction function that according to the past data of the project, will predict the future project effort and duration. However, due to the vast number of machine learning algorithms, there are still different machine learning algorithms not analyzed.

In this paper, we will compare the performance of several different Machine learning algorithms to predict Software Project time. At last, we evaluate the algorithms according to Magnitude Relative Error (MRE), Mean magnitude relative error (MMRE) and MdMRE [5-7].

ISSN: 1975-0080 IJMUE

2. Related Work

In this section, we will briefly describe Neural networks and the main characteristics of the machine learning algorithms compared [10-17].

2.1. Neural Networks

A neural network (NN) is a model of computing and signal processing that is inspired in the processing done by a network of biological neurons [16]. The basis for the construction of a neural network is the artificial neuron. The input of an artificial neuron is a vector of numeric values.

A neural network learns by adjusting its parameters. The parameters are the values of bias and weights in its neurons. Some neural networks learn constantly during their application, whereas most of them have two distinct periods: a training period and an application period. During the training period a network processes prepared inputs and adjusts its parameters. In order to improve its performance, it is guided by some learning algorithm. Once the performance is acceptably accurate, or precise, the training period is finished. The parameters of the network are then fixed to the learned values, and the network starts its period of application for the intended task [9].

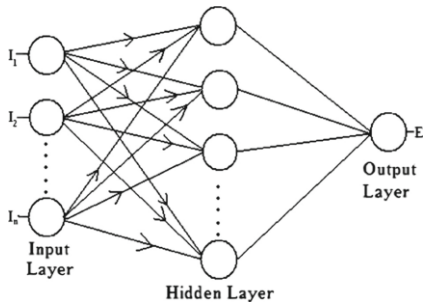

Figure1 Simple Neural Network

The simplest form of ANN is shown in the Figure 1. It has three layers: Input layer, Hidden Layer and Output Layer. Input layer has as much number of neurons as number of input parameters to the ANN model, the next Layer is Hidden Layer, which gets inputs from input layer neurons and it is fully or partially connected to the input layer. Hidden Layer is normally fully connected with the Output Layer. Output Layer has one or more output neurons depending on the output of the model [10].

2.2. Linear Regression

Linear regression (LR) is a method to model the linear relationship between a scalar dependent variable y and one or more independent variables x [11].

2.3. Least Median Square

Least Median Square (LMS) method is one of the statistical methods for solving the equations which are more than unknown. This method almost used in analytical regression. In fact, Least Median Square is a method for fitting the dataset. The least Median Square must yield the smallest value for the median of squared residuals computed for the entire data set. It means the residuals, the difference between real data and predicted data [12].

2.4. Gaussian Process

In probability theory, the Gaussian process (GP) is realization consist of random values associated with every point in a range of times. The random variable has a normal distribution. It has been mentioned that "a Gaussian process is a stochastic process whose generalization of a Gaussian distribution over a finite vector space to a function space of infinite dimension".

Gauss algorithm is a method used to solve non-linear least squares problems. The method is named after the mathematicians Carl Friedrich Gauss and Isaac Newton [17-18].

Non-linear least squares problems arise for instance in non-linear regression, where parameters in a model are sought such that the model is in good agreement with available observations.

Given m functions r = (r1,…,rm) of n variables β= (β1,…,βn), with m ≥ n, the Gauss algorithm finds the minimum of the sum of squares, as in (1).

$$S(\beta) = \sum_{i=1}^{m} r_i^2(\beta) \tag{1}$$

Starting with an initial guess β (0) for the minimum, the method proceeds by the iterations, as in (2).

$$\beta^{(s+1)} = \beta^{(s)} + \Delta, \tag{2}$$

Where

$$S(\beta^{(S)} + \Delta) = S(\beta^{(S)}) + \left[\frac{\partial S}{\partial \beta_i}\right]\Delta + \frac{1}{2}\Delta^T\left[\frac{\partial^2 S(\beta)}{\partial \beta_i \partial \beta_j}\right]\Delta$$

Δ is a small step. We then have.

If we define the Jacobian matrix as in (3).

,

$$J_r(\beta) = \frac{\partial r_i}{\partial \beta_j}\Big|_{\beta} \tag{3}$$

We can replace

$\left[\frac{\partial S}{\partial \beta_i}\right]$ with $J_r^T r$

and the Hessian matrix can be approximated by (4).

$$S(\beta^{(S)} + \Delta) \approx S(\beta^{(S)}) + J_r^T r\Delta + \frac{1}{2}\Delta^T J_r^T J_r \Delta \tag{4}$$

(assuming small residual), giving: $J_r^T J_r$ .We then take the derivative with respect to Δ and set it equal to zero to find a solution as in (5).

$$S'(\beta^{(S)} + \Delta) \approx J_r^T r + J_r^T J_r \Delta = 0 \tag{5}$$

This can be rearranged to give the normal equations which can be solved for Δ as in (6).

$$(J_r^T J_r)\Delta = -J_r^T J_r \tag{6}$$

In data fitting, where the goal is to find the parameters β such that a given model function y=f(x, β) fits best some data points (xi, yi), the functions ri are the residuals.

$$r_i(\beta) = y_i - f(x_i, \beta)$$

Then, the increment Δ can be expressed in terms of the Jacobian of the function f, as in (7).

$$(J_f^T J_f)\Delta = J_f^T r \tag{7}$$

2.5. M5P

M5P is a tree learner and consists of binary decision tree which learns a "model" tree. This is a decision tree with linear regression functions at the leaves. It is used to predict a numeric target attribute. It may be piecewise fit to the target [2].

2.6. REPtree

REPtree (RT) Builds a decision tree using information variance and prunes it using reduced-error pruning (with back fitting). Only sorts values for numeric attributes once. Missing values are dealt with by splitting the corresponding instances into pieces [2].

2.7. Sequential Minimal Optimization

Sequential minimal optimization (SMO) was proposed as an iterative algorithm. This algorithm uses SVM for solving regression problem and SVM classifier design. The SMO algorithm has two worthy aspects: implementation is easy and computational speed is fast [14-15].

2.8. Multilayer Perceptron

The multilayer perceptron (MLP) is a very simple model of biological neural networks. The network is structured in a hierarchical way. Multilayer Perceptron includes some nodes located in different layers where the information flows only from one layer to the next layer. The first and the last layers are the input and output. Layers between the input and output layer are called hidden layers. This network is trained based on back propagation error in that the real outputs are compared with network outputs, and the weights are set using supervised back propagation to achieve the suitable model [16].

3. Comparison of Machine Learning Algorithms to Predict Project Time

The main goal of this paper is to compare the effectiveness of different machine learning techniques to predict the time of software project.

We have collected lots of software projects. Table 1 describes the attributes of projects, which are used to build the time prediction model by a machine learning technique.

Table 1. Information of the Projects Collected

Number of projects	Number of Workers	Time of projects(Month)	Effort of projects(PM)	KLOC
59	6-28	3-28	6-220	3-320

From the total 59 project data, as in Table 1, we used 50 randomly selected data points for training and building the model and the other 9 of them for test and evaluate the model. Then the same training data and testing data was used for all the models. Figure 1 illustrates the overall flow of the work conducted.

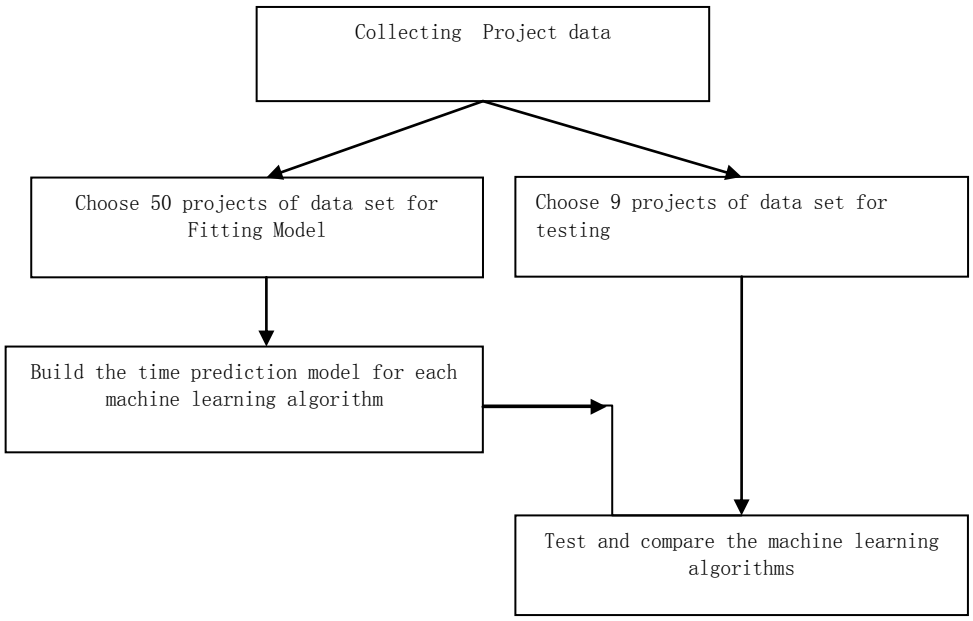

Figure 1. Flow Chart of Building Model

It is desired to find time prediction model for software project. We fit the models by seven kinds of machine learning algorithms. So we can obtain seven time prediction model, shown as in Eq.(8)-(14), which are fitted respectively by linear regression, SMO, multilayer Perceptron, M5P, Leastmedsqr, REPtree and Gaussian process.

$$T = 2.81 \times E^{0.38} \quad (8)$$

$$T = 3.12 \times E^{0.29} \quad (9)$$

$$T = 2.98 \times E^{0.31} \quad (10)$$

$$T = 4.02 \times E^{0.39} \quad (11)$$

$$T = 3.62 \times E^{0.41} \quad (12)$$

$$T = 2.91 \times E^{0.21} \quad (13)$$

$$T = 3.99 \times E^{0.35} \quad (14)$$

Where T is project prediction time(Month) ,and E is Project effort(Person Month). These functions are fitted by collected project data shown in Table 1.

In order to evaluate the accuracy of the Model, we use the common evaluation criteria in the field They are Magnitude Relative Error (MRE), Mean magnitude relative error (MMRE) and MdMRE [5-7].

1) Magnitude Relative Error (MRE) computes the absolute percentage of error between actual and predicted time for each reference project.

$$MRE_i = \frac{Actual_i - Estimated_i}{Actual_i}$$

2) Mean magnitude relative error (MMRE) calculates the average of MRE over all reference projects. Despite of the wide use of MMRE in estimation accuracy, there has been a substantive discussion about the efficacy of MMRE in estimation process. MMRE has been criticized that is unbalanced in many validation circumstances and leads often to overestimation (Shepperd and Schofield, 1997). Moreover, MMRE is not always reliable to compare between prediction methods because it is quite related to the measure of MRE spread (Foss *et al*., 2003).

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i$$

3) Median of MRE (MdMRE): MMRE is sensitive to individual predictions with excessively large MREs. The median of MREs (MdMRE), is less sensitive to extreme values and is used as another measure [18]. We adopt median of MREs for the n projects (MdMRE) which is less sensitive to the extreme values of MRE.

$$MdMRE = \underset{i}{median} \; MRE_i .$$

4. Experimental Results

For comparing machine learning algorithms, we use MRE, MMRE and MdMRE to evaluate the results. Based on this value, we rank the algorithms according to each metric of interest. At the end, we compute the overall ranked position by the same approach [19-21]. Presents the result of each algorithm. We can state that predicting values by Gaussian Process are directly and closely related to training dataset values, comparing other algorithms. The ranked list of the algorithms will be as follows considering the four scenarios together: (1) GP, (2) M5P and SMO, (3) LR, (4) RT, (5) LMS and (6) MLP.

This indicates that Gaussian Process algorithm is able to adapt better to different projects. After analyzing the four performance metrics for all algorithms, we can observe that GP is the best candidate.

Table 2. The Evaluation for Machine Learning Algorithms

	LR	MLP	M5P	SMO	GP	LMS	RT
MRE	0.910	0.921	0.897	0.956	0.976	0.898	0.921
MMRE	0.875	0.936	0.934	0.895	0.951	0.921	0.937
MdMRE	0.915	0.899	0.911	0.901	0.960	0.932	0.945

5. Conclusions Second and Following Pages

In this paper, we used different machine learning algorithms to build software time prediction model, based on lots of project data. Then we comprehensively compared different machine learning algorithms. Our experimental results show that Gaussian Process obtains the better overall performance among the studied algorithms. As a future work, it would necessary to compare the algorithms under more complex projects.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61170273).

References

[1] C. S. Wu, W. C. Chang and I. K. Sethi, "A Metric-Based Multi-Agent System for Software Project Management", 2009 Eight IEEE/ACIS International Conference on Computer and Information Science.
[2] I. H. Witten and E. Frank, "Data Mining Practical Machine Learning Tools and Techniques", Morgan Kaufmann Publishers Is an Imprint of Elsevier, **(2005)**, pp. 187-427.
[3] K. Nigam, A. Mccallum and T. Mitchell, "Text classification from labeled and unlabeled documents using em", Machine Learning, vol. 39, no. 2, **(2000)**.
[4] Z. Y. Yang and Q. Wang, "Handling missing data in software effort prediction with naive Bayes and EM algorithm", PROMISE '11, Banff, Canada, September 20-21, **(2011)**.
[5] Foss T., Stensrud E., Kitchenham B. and Myrtveit I., 2003. "A simulation study of the model evaluation criterion MMRE", IEEE Trans Softw Eng, vol. 29, no. 11, pp. 985-995.

[6] M. A. Ahmed, I. Ahmad, J. S. AlGhamdi, "Probabilistic size proxy for software effort prediction: A framework", Information and Software Technology, vol. 55, **(2013)**, pp. 241–251.
[7] N. S. Gill and P. S. Grover, "Software size prediction before coding, ACM SIGSOFT", Software Engineering, Notes, vol. 29, no. 5, **(2005)**.
[8] M. Y. Yerina and A. F. Izmailov, "The Gauss-Newton method for finding singular solutions to systems of nonlinear equations", Computational Mathematics and Mathematical Physics, vol. 47, no. 5, May **(2007)**, pp. 748-759.
[9] C. L. Martín, A. Chavoya and M. E. M. Campaña, "Use of a Feed forward Neural Network for Predicting the Development Duration of Software Projects", 2013 12th International Conference on Machine Learning and Applications.
[10] Vachik S. Dave · Kamlesh Dutta ,Neural network based models for software effort estimation: a review, Artif Intell Rev (2014) 42:295–307
[11] A. Andrzejak and L. Silva, "Using Machine Learning for Non-Intrusive Modeling and Prediction of Software Aging", Network Operations and Management Symposium, IEEE, April 7-11, **(2008)**, pp. 25-32.
[12] J. A. Lopez, J. L. Berral, R. Gavalda and J. Torres, "Adaptive on-line software aging prediction based on machine learning", in Procs. 40th IEEE/IFIP Intl. Conf. on Dependable Systems and Networks, June 28, July 1, **(2010)**, pp. 497-506.
[13] Y. Wang, and I. H. Witten, "Inducing Model Trees for Continuous Classes", the European Conf. on Machine Learning Poster Papers, **(1997)**.
[14] Weka 3.6.8 http://www.cs.waikato.ac.nzlmllwekal.
[15] A. J. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression", Royal Holloway College, London, U.K., NeuroCOLT Tech. Rep.TR 1998-030, **(1998)**.
[16] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya and K. R. K. Murthy, "Improvements to the SMO Algorithm for SVM Regression", IEEE Transactions ON Neural Networks, vol. 11, no. 5, September **(2000)**.
[17] D. J. C. Mackay, "Introduction to Gaussian Processes", Dept. of Physics, Cambridge University, UK, **(1998)**.
[18] W. Zhang, Y. Yang and Q. Wang, "Handling missing data in software effort prediction with naive Bayes and EM algorithm", PROMISE '11, Banff, Canada, September 20-21, **(2011)**.
[19] A. Macedo, T. B. Ferreira and R. Matias, "The Mechanics of Memory-Related Software Aging", Second IEEE International Workshop on Software Aging and Rejuvenation, November 2-2, **(2010)**.
[20] S. J. Huang and W. M. Han, "Exploring the relationship between software project duration and risk exposure: A cluster analysis", Information & Management, vol. 45, **(2008)**, pp. 175–182.
[21] P. K. Subramanian, "Gauss-Newton methods for the complementarity problem", Journal of Optimization Theory and Applications, vol. 77, no. 3, June **(1993)**, pp. 467-482.

Authors

Wan-Jiang Han, was born in Hei Long Jiang province, China, 1967. She received her Bachelor Degree in Computer Science from Hei Long Jiang University in 1989 and her Master Degree in Automation from Harbin Institute of Technology in 1992.

She is an assistant professor in School Of Software Engineering, Beijing University of Posts and Telecommunication, China. Her technical interests include software project management and software process improvement.

Li-Xin Jiang, was born in Hei Long Jiang province, China, 1966. He received his Bachelor Degree and Master Degree in physical geography from Beijing University in 1989 and 1992.

He is a professor in the department of Emergency Response of China Earthquake Networks Center. His technical interests include software cost estimation and Emergency Response software development.

Tian-Bo Lu, was born in Guizhou Province, China, 1977. He received his Master Degree in computer science from Wuhan University in 2003 and his PH.D Degree in computer science from the Institute of Computing Technology of the Chinese Academy of Sciences in 2006.

He is an Associate professor in School of Software Engineering, Beijing University of Posts and Telecommunications, China. His technical interests include information and network security, trusted software and P2P computing.

Xiao-Yan Zhang, was born in Shandong Province, China, 1973. She received her Master Degree in Computer Application in 1997 and her PH.D Degree in Communication and information system from Beijing University of Posts and Telecommunication, China, in 2011.

She is an Associate professor in School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, China. Her technical interests include software cost estimation and software process improvement.

