

Chapter 78

Machine Learning Classification to Effort Estimation for Embedded Software Development Projects

Kazunori Iwata

Department of Business Administration, Aichi University, Nagoya, Japan

Toyoshiro Nakashima

Department of Culture-Information Studies, Sugiyama Jogakuen University, Nagoya, Japan

Yoshiyuki Anan

Process Innovation H.Q, Omron Software Co., Ltd., Kyoto, Japan

Naohiro Ishii

Department of Information Science, Aichi Institute of Technology, Nagoya, Japan

ABSTRACT

This paper discusses the effect of classification in estimating the amount of effort (in man-days) associated with code development. Estimating the effort requirements for new software projects is especially important. As outliers are harmful to the estimation, they are excluded from many estimation models. However, such outliers can be identified in practice once the projects are completed, and so they should not be excluded during the creation of models and when estimating the required effort. This paper presents classifications for embedded software development projects using an artificial neural network (ANN) and a support vector machine. After defining the classifications, effort estimation models are created for each class using linear regression, an ANN, and a form of support vector regression. Evaluation experiments are carried out to compare the estimation accuracy of the model both with and without the classifications using 10-fold cross-validation. In addition, the Games-Howell test with one-way analysis of variance is performed to consider statistically significant evidence.

DOI: 10.4018/978-1-6684-3702-5.ch078

INTRODUCTION

The growth and expansion of our information-based society has resulted in an increasing number of information products. In addition, the functionality of these products is becoming ever more complex (Hirayama, 2004; Takagi 2003). Guaranteeing the quality of software is particularly important, because this relates to reliability. It is, therefore, increasingly important for corporations that develop embedded software to realize efficient methods while guaranteeing timely delivery, high quality, and low development costs (Boehm, 1976; Komiyama, 2003; Nakashima, 2004; Ogasawara & Kojima, 2003; Takagi 2003; Tamaru, 2004; Ubayashi, 2004; Watanabe, 2004). Companies and divisions involved in the development of such software are focusing on a variety of improvements, particularly in their processes. Estimating the amount of effort (man-days cost) required for new software projects and guaranteeing product quality are especially important, because the amount of effort is directly related to cost and the product quality affects the reputation of the corporation. In the field of embedded software, there have been various studies on development techniques, management techniques, tools, testing techniques, reuse techniques, real-time operating systems, and other elements. However, there has been little research on the relationships among the scale of the development, amount of effort, and number of errors based on data accumulated from past projects. Previously, we investigated the estimation of total effort and errors using an artificial neural network (ANN), and showed that ANN models are superior to regression analysis models for estimating effort and errors in new projects. We proposed a method to estimate intervals for the number of errors using a support vector machine (SVM) and an ANN. These models were created with data that excluded outliers. The outliers can be identified in practice once the projects have been completed, and so they should not be excluded during the creation of models and when estimating the effort required. This paper presents classifications for embedded software development projects based on whether the amount of effort is an outlier or not using an ANN and an SVM. After the classification stage, we establish effort estimation models for each class using linear regression (LR), an ANN, and ϵ -support vector regression (ϵ -SVR). Evaluation experiments are carried out to compare the estimation accuracy of the model both with and without classification using 10-fold cross-validation and by applying the Games-Howell test with one-way analysis of variance (ANOVA).

The rest of this paper is organized as follows. First, we explain the related works. Second, we show datasets used in this paper. After that, we place our work and evaluation experiments. As a result, we conclude the paper.

RELATED WORKS

Support Vector Regression

SVR uses the same principles as SVM for classification, albeit with a few minor differences. The ϵ -SVR (Alex & Bernhard, 2004) regression method uses an ϵ -insensitive loss function to solve regression problems. This approach attempts to find a continuous function in which as many data points as possible lie within the ϵ -wide insensitivity tube. ϵ -SVR is used to estimate the amount of effort required for software projects (Oliveira, 2006). This approach has been tested using the well-known NASA software project dataset (John & Victor, 1981; Shin & Goel, 2000). However, these studies did not investigate the parameters of ϵ -SVR. The effectiveness of the SVM (and SVR) using the resulting continuous function

12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/machine-learning-classification-to-effort-estimation-for-embedded-software-development-projects/294535

Related Content

A Design Method for Real-Time Object-Oriented Systems Using Communicating Real-Time State Machines

Eduardo B. Fernandez, Jie Wuand Debera R. Hancock (2002). *Successful Software Reengineering* (pp. 171-185).

www.irma-international.org/chapter/design-method-real-time-object/29975

A Systematic Approach for Designing Educational Recommender Systems

Patrick H. S. Brito, Ig Ibert Bittencourt, Aydano Pamponet Machado, Evandro Costa, Olavo Holanda, Rafael Ferreiraand Thiago Ribeiro (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 1264-1288).

www.irma-international.org/chapter/systematic-approach-designing-educational-recommender/77757

Method Using Command Abstraction Library for Iterative Testing Security of Web Applications

Seiji Munetohand Nobukazu Yoshioka (2015). *International Journal of Secure Software Engineering* (pp. 26-49).

www.irma-international.org/article/method-using-command-abstraction-library-for-iterative-testing-security-of-web-applications/136451

Software Change Impact Analysis: An Approach to Compute and Prioritize Impacted Functions in Software Systems

Chetna Guptaand Varun Gupta (2015). *International Journal of Systems and Service-Oriented Engineering* (pp. 44-55).

www.irma-international.org/article/software-change-impact-analysis/126637

Computer-Aided Management of Software Development in Small Companies

Lukáš Pichland Takuya Yamano (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 2379-2390).

www.irma-international.org/chapter/computer-aided-management-software-development/29512