The Journal of Systems and Software 101 (2015) 127–135

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Neural networks for predicting the duration of new software projects

Cuauhtémoc López-Martín[a,*], Alain Abran[b]

[a] Information Systems Department, CUCEA, Universidad de Guadalajara, Periférico Norte #799, Zapopan, Jalisco, P.O. Box 45100, Mexico
[b] Department of Software Engineering and Information Technology, École de technologie supérieure, Université du Québec, 1100 Rue Notre-Dame Ouest, Montréal, Québec, QC H3C 1K3, Canada

ARTICLE INFO

Article history:
Received 9 July 2014
Revised 21 November 2014
Accepted 3 December 2014
Available online 9 December 2014

Keywords:
Software project duration prediction
Radial basis function neural network
Multilayer feedforward neural network

ABSTRACT

The duration of software development projects has become a competitive issue: only 39% of them are finished on time relative to the duration planned originally. The techniques for predicting project duration are most often based on expert judgment and mathematical models, such as statistical regression or machine learning. The contribution of this study is to investigate whether or not the duration prediction accuracy obtained with a multilayer feedforward neural network model, also called a *multilayer perceptron* (MLP), and with a radial basis function neural network (RBFNN) model is statistically better than that obtained by a multiple linear regression (MLR) model when functional size and the maximum size of the team of developers are used as the independent variables. The three models mentioned above are trained and tested by predicting the duration of new software development projects with a set of projects from the International Software Benchmarking Standards Group (ISBSG) release 11. Results based on absolute residuals, Pred(*l*) and a Friedman statistical test show that prediction accuracy with the MLP and the RBFNN is statistically better than with the MLR model.

1. Introduction

Software engineering (SE) involves a number of product development activities, including engineering management. Software project planning (SPP) addresses the activities undertaken to prepare for a successful SE project from a management perspective. SPP involves process planning, determination of deliverables, software prediction (also referred to as software estimation), risk management, quality management, and plan management (Abran and Moore, 2004).

The term *development* is frequently used with reference to three generic stages: design, construction, and testing; and the term *maintenance* refers to anomalies uncovered, operating environments changed, and additional user requirements put forward after the software product has been delivered (Abran and Moore, 2004). A maintained software project is also referred to as an *enhancement* development project (ISBSG, 2011).

Both development and maintenance are typically accompanied by other activities, such as documentation, risk analysis, verification, validation, and measurement. For both development and maintenance, the following types of predictions can be made once the software requirements have been specified (Abran and Moore, 2004):

a) The number of person-hours required to complete the development or maintenance (effort).
b) The duration of tasks with projected start times, individual duration by task, and end times. In SPP, the critical path is usually called duration (Berlin et al., 2009), schedule (Alyahya et al., 2009), or cycle time (Agrawal and Chari, 2007).
c) The cost of the project based on the resource requirements, such as people or tools.

Duration prediction is necessary for budgeting purposes, and is typically handled on a monthly basis in software development organizations (i.e. building rental, employee health or life insurance, and so on). The duration of projects has also been used as a reference for the maturity of processes in software enterprises (Agrawal and Chari, 2007; Alyahya et al., 2009; Harter et al., 2000), since the under prediction or over prediction of project duration at the planning stage can negatively impact budgets.

A 2013 study based on an analysis of 50,000 projects developed between the years 2003 and 2012 in real environments from USA (60%), Europe (25%) and the remaining 15% representing the rest of the world, reports that only 39% of projects were delivered on time, on budget, and with the required features and functions; 43% of them were challenged (late, over budget, and/or with less than the required features and functions); and 18% failed (cancelled prior to completion or delivered and never used) (Chaos Report, 2013).

We found only a few studies on duration prediction in the literature, published in the past 15 years: 2002 (Kitchenham et al., 2002),

* Corresponding author. Tel.: +52 33 36 56 24 07.
E-mail addresses: cuauhtemoc@cucea.udg.mx (C. López-Martín), alain.abran@etsmtl.ca (A. Abran).

http://dx.doi.org/10.1016/j.jss.2014.12.002
0164-1212/

2007 (Bourque et al., 2007), 2009 (Berlin et al., 2009), 2012 (Wang et al., 2012), and 2013 (López-Martín et al., 2013; Zapata and Chaudron, 2013). Bourque et al. (2007) report that duration studies prior to 2002 were published in the late 1970s and during the 1980s. However, a 2013 study analyzing 171 projects developed by 1000 practitioners working in 50 countries (Zapata and Chaudron, 2013) notes that, while most of the studies on accuracy have focused on effort prediction, the main prediction issue in practice is duration. Actually, development duration has become a competitive issue in many industries (Alyahya et al., 2009; Harter et al., 2000; Zapata and Chaudron, 2013), therefore, this study investigates the prediction of the duration of software development projects.

The techniques for predicting the duration have been based upon expert judgment (Kitchenham et al., 2002; Zapata and Chaudron, 2013), statistical regression (Berlin et al., 2009; Bourque et al., 2007; Kitchenham et al., 2002; López-Martín et al., 2013; Oligny et al., 1997; Wang et al., 2012), artificial neural networks (Berlin et al., 2009; López-Martín et al., 2013; Wang et al., 2012) and support vector machines (Wang et al., 2012).

A neural network can learn complex (nonlinear) functions (Anderson, 1995), and nonlinear relationships are common among dependent and independent variables in software projects (Chao-Jung and Chin-Yu, 2011). The kind of neural network used for predicting the duration of software projects has been the multilayer feedforward neural network, also termed multilayer perceptron (MLP) (Berlin et al., 2009; López-Martín et al., 2013; Wang et al., 2012). In this study, another kind of neural network referred to as radial basis function neural network (RBFNN) is proposed.

In the software prediction field, it is common practice to use software size as the independent variable for predicting project effort, and the predicted effort as the independent variable for predicting the duration of the project (Ahmed et al., 2013; Berlin et al., 2009; Boehm et al., 2000; Bourque et al., 2007). The size of a software product is mainly measured in function points or source lines of code (Sheetz et al., 2009), while the duration of a software project is usually measured in months.

Taking into consideration that actual effort is not known at the start of the project and cannot be used as an independent variable in a duration prediction model at prediction time (Bourque et al., 2007; Kitchenham et al., 2002), and that the duration of a software project also depends on the number of developers it involves (Kitchenham et al., 2002), the models proposed in this study are trained and tested using the size of the projects and the number of developers as independent variables instead of using development effort as the independent variable. Sixteen publicly available datasets of software projects were analyzed for identifying the availability of these two independent variables for our research purposes: 15 datasets from the PROMISE repository (PROMISE, 2014) and release 11 of the International Software Benchmarking Standards Group (ISBSG) (ISBSG, 2011). In the PROMISE repository, all of the 15 datasets had an attribute related to the size of the projects (either in function points or lines of code), while 8 of them (China, COCOMO 81, COSMIC, ISBSG release 10, Kemerer, Kitchenham, Maxwell, and Nasa93) included an attribute related to duration (reported either in months or days). Excluding the 2 datasets (COSMIC and ISBSG release 10) corresponding to a subset of the ISBSG dataset, none 13 PROMISE datasets had an attribute related to the number of developers. In the release 11 of the ISBSG dataset, the required attributes related to the size of projects, the number of developers, and the duration of software projects were available. Therefore, only the ISBSG dataset could be used for our research purposes. In addition, the ISBSG dataset also allowed to identify the software projects based upon their type of development (*new* and *enhancement*), development platform, and programming language type (ISBSG, 2011).

The contribution of this study is to investigate whether or not the duration prediction accuracy obtained with an MLP and with a RBFNN model is better than that obtained by a multiple linear regression (MLR) model when functional size and the maximum size of the team of developers are used as the independent variables.

The comparison among prediction accuracies of the MLP, RBFNN and MLR model, was achieved using absolute residuals (AR) and Pred(l) as the accuracy criteria.

Specifically, the hypothesis investigated in this research is the following:

H1. The accuracy of duration prediction with an MLP and a RBFNN is statistically better than the accuracy obtained by MLR when adjusted function point data and the maximum team size of developers are used as the independent variables.

The rest of this study is organized as follows: Section 2 presents related works on duration prediction for software projects. Section 3 describes and compares the MLP and RBFNN models. Section 4 presents the criteria for evaluating the accuracy of the models, as well as for selecting the sample data from the ISBSG dataset. Section 5 describes the training and testing for the three models. Section 6 compares the accuracy results obtained for the models. Finally, section 7 presents a discussion, including our conclusions, the limitations of our study, and directions for future work.

2. Related work

In the software project prediction field, techniques have mainly been used for predicting software product size, project effort, and project duration. These prediction techniques have been based on informal models, such as expert judgment, or on mathematical models, such as statistical and machine learning techniques.

Regarding software product size prediction, the techniques reported in the literature are either expert judgment (Wilkie et al., 2011) or mathematical models, such as statistical regression (Laranjeira, 1990), neural networks (Hakkarainen et al., 1993), and fuzzy logic (MacDonnel, 2003).

In accordance with project effort prediction, the techniques used are either expert judgment (Halkjelsvik and Jørgensen, 2012; López-Martín and Abran, 2012) or mathematical models based on probability density function (Ahmed et al., 2013), statistical regression (Yang et al., 2013; Yeong-Seok et al., 2013), as well as machine learning models, such as case-based reasoning, artificial neural networks, decision trees, Bayesian networks, support vector regression, genetic algorithms, genetic programming, and association rules (Wen et al., 2012). Most studies related to software prediction have focused on predicting effort: Jørgensen and Shepperd (2007) analyzed 304 studies in 76 journals published up to early 2004, and Wen et al., (2012) analyzed 84 studies published from 1991 to 2010. Jørgensen and Shepperd (2007) surveyed studies in which any prediction technique had been used, whereas Wen et al. (2012) analyzed only studies involving machine learning techniques. Neural networks have been the second most widely used technique in the software effort prediction field to date, at 26% (Wen et al., 2012). The MLP has been used to predict the development effort of software projects at least since 1997 (Finnie et al., 1997) and in 2008, the MLP was reported as the most commonly used (Park and Baek , 2008); and it has been applied until the year 2009 (Berlin et al. 2009), 2012 (González-Carrasco et al., 2012), 2013 (Chou and Wu , 2013; Kocaguneli and Menzies, 2013; Nassif et al., 2013) and 2014 (López-Martín, 2014; Mittas et al., 2014).

As for duration prediction of software projects, we only identified seven recent studies which can be analyzed from three points of view: the technique used, the independent variables used, and the dataset used as the source of the software projects for training and testing prediction models: