Information and Software Technology 52 (2010) 1155–1166

Contents lists available at ScienceDirect

Information and Software Technology

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

GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation

Adriano L.I. Oliveira *, Petronio L. Braga, Ricardo M.F. Lima, Márcio L. Cornélio

Center of Informatics, Federal University of Pernambuco, Recife 50.732-970, Brazil

ARTICLE INFO

Article history:
Received 26 August 2009
Received in revised form 11 May 2010
Accepted 14 May 2010
Available online 31 July 2010

Keywords:
Software effort estimation
Genetic algorithms
Feature selection
Support vector regression
Regression

ABSTRACT

Context: In software industry, project managers usually rely on their previous experience to estimate the number men/hours required for each software project. The accuracy of such estimates is a key factor for the efficient application of human resources. Machine learning techniques such as radial basis function (RBF) neural networks, multi-layer perceptron (MLP) neural networks, support vector regression (SVR), bagging predictors and regression-based trees have recently been applied for estimating software development effort. Some works have demonstrated that the level of accuracy in software effort estimates strongly depends on the values of the parameters of these methods. In addition, it has been shown that the selection of the input features may also have an important influence on estimation accuracy.
Objective: This paper proposes and investigates the use of a genetic algorithm method for simultaneously (1) select an optimal input feature subset and (2) optimize the parameters of machine learning methods, aiming at a higher accuracy level for the software effort estimates.
Method: Simulations are carried out using six benchmark data sets of software projects, namely, Desharnais, NASA, COCOMO, Albrecht, Kemerer and Koten and Gray. The results are compared to those obtained by methods proposed in the literature using neural networks, support vector machines, multiple additive regression trees, bagging, and Bayesian statistical models.
Results: In all data sets, the simulations have shown that the proposed GA-based method was able to improve the performance of the machine learning methods. The simulations have also demonstrated that the proposed method outperforms some recent methods reported in the recent literature for software effort estimation. Furthermore, the use of GA for feature selection considerably reduced the number of input features for five of the data sets used in our analysis.
Conclusions: The combination of input features selection and parameters optimization of machine learning methods improves the accuracy of software development effort. In addition, this reduces model complexity, which may help understanding the relevance of each input feature. Therefore, some input parameters can be ignored without loss of accuracy in the estimations.
© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Experienced software project managers develop the ability to find the trade-off between software quality and time-to-market. The efficiency in resource allocation is one of the main aspects to find out such equilibrium point. In this context, estimating software development effort is essential.

An study published by the Standish Group's Chaos states that 66% of the software projects analyzed were delivered with delay or above the foreseen budget, or worse, they were not finished [8]. Failures rate of software projects is still very high [9,10]; it is estimated that over the last 5 years the impact of such software project failures on the US economy have cost between 25 billion and 75 billion [9,10]. In this context, both overestimates and underestimates of the software effort are harmful to software companies [7]. Indeed, one of the major causes of such failures is inaccurate estimates of effort in software projects [10]. Hence, investigate novel methods for improving the accuracy of such estimates is essential to strengthen software companies' competitive strategy.

Several methods have been investigated for software effort estimation, including traditional methods such as the constructive cost model (COCOMO) [11], and, more recently, machine learning techniques such as radial basis function (RBF) neural networks [12], MLP neural networks [26], multiple additive regression trees [30], wavelet neural networks [27], bagging predictors [13] and support vector regression (SVR) [10]. Machine learning techniques use data from past projects to build a

* Corresponding author. Tel.: +55 81 32668939.
E-mail addresses: alio@cin.ufpe.br (A.L.I. Oliveira), plb@cin.ufpe.br (P.L. Braga), rmfl@cin.ufpe.br (R.M.F. Lima), mlc2@cin.ufpe.br (M.L. Cornélio).

0950-5849/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2010.05.009

regression model that is subsequently employed to predict the effort of new software projects.

Genetic algorithms (GAs) were shown to be very efficient for optimum or approximately optimum solution search in a great variety of problems. They avoid problems found in traditional optimization algorithms, such as returning the local minimum [14]. Recently, Huang and Wang proposed a genetic algorithm to simultaneously optimize the parameters and input feature subset of support vector machine (SVM) without loss of accuracy in *classification problems* [15]. Two factors substantially influence the accuracy and computation time of machine learning techniques: (1) the choice of the input feature subset and (2) the choice of the parameters values of machine learning techniques. Hence, according to Huang and Wang, the simultaneous optimization of these two factors improves the accuracy of machine learning techniques for classification problems.

Oliveira employed grid selection for optimizing SVR parameters for software effort estimation [10]. His work did not investigate feature selection methods; all input features were used for building the regression models. Huang and Wang demonstrated that the simultaneous optimization of the parameters and feature selection improves the accuracy of SVM results for classification problems [15]. Their results showed that GA-based method outperforms grid selection for SVM parameter optimization for classification problems [15]. These results motivated us to adapt the ideas of Huang and Wang for machine learning regression methods. In particular, we aim at reducing the number of input features, keeping the accuracy level for software effort estimates.

In this context, this paper adapts the method proposed by Huang and Wang [15] for feature selection and parameters optimization of machine learning methods applied to software effort estimation (a regression problem). The idea behind our method is to adapt the fitness function of the genetic algorithm and the parameters to be optimized. Notice that support vector machines for regression (SVR) has three important parameters, whereas SVM for classification has only two. Furthermore, our method generalizes the method of Huang and Wang [15], since we apply it to three different machine learning techniques (SVR, MLP neural networks and M5P model trees), whereas their method was developed and investigated solely for SVMs [15].

The main contributions of this paper are threefold: (1) to develop a novel method for software effort estimation based on genetic algorithms applied to input feature selection and parameters optimizations of machine learning methods; (2) to investigate the proposed method by applying it to three machine learning techniques, namely, (i) support vector regression (SVR), (ii) multi-layer perceptron (MLP) neural networks, and (iii) model trees; and (3) to show that our method outperforms recent methods proposed and investigated in the literature for software effort estimation [5,30,26,33,13,25,10].

This paper is organized as follows. Section 2 reviews the regression methods used in this paper and Section 3 reviews some basic genetic algorithm (GA) concepts. In Section 4, we present our GA-based method for feature selection and optimization of machine learning parameters for software effort estimation. The experiments and results are discussed in Section 5. Finally, Section 6 concludes the paper.

2. Regression methods

The goal of regression methods is to build a function $f(x)$ that adequately maps a set of independent variables $(X_1, X_2, \ldots, X_n)$ into a dependent variable Y. In our case, we aim to build regression models using a training data set to use it subsequently to predict the total effort for the development of software projects in man-months.

2.1. Support vector regression

Support vector machines (SVMs) are a set of machine learning methods used in many applications, such as classification and regression [16]. SVMs are based on structural risk minimization (SRM); instead of attempting to minimize only the empirical error, SVMs simultaneously minimize the empirical error and maximize the geometric margin [16]. This method has outperformed previous ones in many classification and regression tasks.

Support vector regression (SVR) was designed for regression problems [17]. SVR is a machine learning technique based on statistical learning theory, developed by Cortes and Vapnik [18]. The investigation of the application of SVR for software project effort estimation was originally carried out by Oliveira [10]. Oliveira showed that SVR outperforms both linear regression and radial basis functions neural networks (RBFNs) for software effort estimation in a data set of NASA software projects [10,12].

Consider a training data set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i \in \Re^d$ denotes an input vector and $y_i \in \Re$ its corresponding target value.

In ε-SVR, the aim is to find a function $f(x)$ that has at most ε deviation from the actually obtained targets y_i for the training data set, and simultaneously is as flat as possible. This type of loss function defines a margin around the true outputs. The idea is that errors smaller than a certain threshold $\varepsilon > 0$ are rejected. That is, errors inside the margin are considered to be zero. On the other hand, errors caused by points outside the margin are measured by variables ξ and ξ^*.

In SVR for linear regression, $f(x)$ is given by $f(x) = \langle w, x \rangle + b$, with $w \in \Re^d$, $b \in \Re$ where $\langle .,. \rangle$ denotes the dot product in $\Re^d$. For the case of nonlinear regression, $f(x) = \langle w, \phi(x) \rangle + b$, where ϕ is some nonlinear function which maps the input space to a higher dimensional feature space ($\Re^d$). In ε-SVR, the weight vector w and the threshold b are chosen to optimize the following problem [16]:

$$\begin{aligned} &\text{minimize}_{w,b,\xi,\xi^*} \quad \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ &\text{subject to} \quad \begin{cases} (\langle w, \phi(x_i)\rangle + b) - y_i & \leqslant \varepsilon + \xi_i \\ y_i - (\langle w, \phi(x_i)\rangle + b) & \leqslant \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geqslant 0 \end{cases} \end{aligned} \tag{1}$$

The constant $C > 0$ determines the trade-off between the model complexity, that is, the flatness of $f(x)$ and the amount up to which deviations larger than ε are tolerated. ξ and ξ^* are called *slack variables* and measure the cost of the errors on the training points. ξ measures deviations exceeding the target value by more than ε

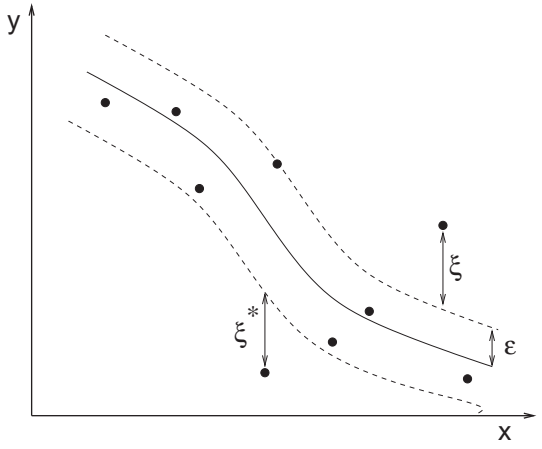
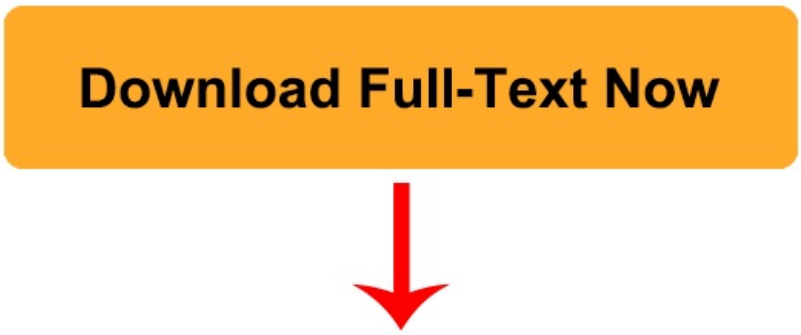


Fig. 1. Regression using ε-SVR.

ID	Title	Pages
550501	GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation	12



http://fulltext.study/article/550501

FullText.study

Categorized Journals
Thousands of scientific journals broken down into different categories to simplify your search

Full-Text Access
The full-text version of all the articles are available for you to purchase at the lowest price

Free Downloadable Articles
In each journal some of the articles are available to download for free

Free PDF Preview
A preview of the first 2 pages of each article is available for you to download for free

http://FullText.Study