Applied Soft Computing 15 (2014) 113–120

Contents lists available at ScienceDirect

Applied Soft Computing

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

Software reliability prediction model based on support vector regression with improved estimation of distribution algorithms

Cong Jin [a,*], Shu-Wei Jin [b]

[a] School of Computer Science, Central China Normal University, Wuhan 430079, PR China
[b] Département de Physique, École Normale Supérieure, 24, rue Lhomond, 75231 Paris Cedex 5, France

ARTICLE INFO

Article history:
Received 10 July 2012
Received in revised form
30 September 2013
Accepted 23 October 2013
Available online 31 October 2013

Keywords:
Support vector regression
Improved estimation of distribution algorithms
Software reliability prediction
Parameters optimization

ABSTRACT

Software reliability prediction plays a very important role in the analysis of software quality and balance of software cost. The data during software lifecycle is used to analyze and predict software reliability. However, predicting the variability of software reliability with time is very difficult. Recently, support vector regression (SVR) has been widely applied to solve nonlinear predicting problems in many fields and has obtained good performance in many situations; however it is still difficult to optimize SVR's parameters. Previously, some optimization algorithms have been used to find better parameters of SVR, but these existing algorithms usually are not fully satisfactory. In this paper, we first improve estimation of distribution algorithms (EDA) in order to maintain the diversity of the population, and then a hybrid improved estimation of distribution algorithms (IEDA) and SVR model, called IEDA-SVR model, is proposed. IEDA is used to optimize parameters of SVR, and IEDA-SVR model is used to predict software reliability. We compare IEDA-SVR model with other software reliability models using real software failure datasets. The experimental results show that the IEDA-SVR model has better prediction performance than the other models.

1. Introduction

Reliability is the ability of software system to perform its required functions under stated conditions for a specified period of time, and it is an important characteristic inherent in the concept of software quality. It is intimately connected with defects and faults. As more and more faults are encountered, the software reliability will decrease. Software reliability generally changes with time, and these changes can be treated as a time series process.

Artificial neural networks (ANN) have general nonlinear mapping capabilities, and have increasingly attracted attention in the field of time series predicting [1–3]. In [4], the reliability of the systems can be predicted by feed-forward multi-layer ANN and radial basis function ANN respectively. The ANN technology has better prediction performance than the autoregressive integrated moving average (ARIMA) approach. In [5], ANN has contributed significantly to software reliability prediction, and which achieved better prediction performance than traditional statistical models. In [6], the counter-propagation and back-propagation ANN models were used to estimate parameters of a reliability distribution with only a small dataset. The experimental results show that the proposed approach improves the accuracy of reliability predicting. In [7], the system reliability may be predicted by a hybrid learning neural fuzzy system. Numerical results demonstrate that the proposed model achieved more accurate predicting results than ARIMA and generalized regression ANN model (GRNN). However, the ANN suffers from a number of weaknesses, e.g., it is based on gradient descent, and it is easy to local minima.

Recently, support vector machines (SVMs) [8–11] have been widely applied to solve nonlinear predicting problems in many fields. With the introduction of ε-insensitive loss function, it has been also extended to solve nonlinear regression estimation problems, such as new techniques known as support vector regression (SVR) [12]. In [13], the SVM was used to solve financial time series problems. The experimental results demonstrate that SVM forecasts better than back propagation (BP) algorithm. In [14], a two-step kernel learning method based on SVR was proposed for predicting financial time series. The results confirm the advantage of SVR. However, although SVR has very good learning performance and generalization ability, there is no structured way to determine the parameters of SVR.

Estimation of distribution algorithms (EDA) [15], sometimes called probabilistic model-building genetic algorithm (GA) [16], have emerged as a generalization of GA, for overcoming the two main problems: poor performance in certain deceptive problems and the difficulty of mathematically modeling a huge number of algorithm variants [17]. In GA, a population of candidate solutions

* Corresponding author. Tel.: +86 02788664026.
E-mail address: jincong@mail.ccnu.edu.cn (C. Jin).

1568-4946/$ – see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.asoc.2013.10.016

to a problem is maintained as part of the search for an optimum solution. This population is typically represented explicitly as an array of objects. Depending on the specifics of the GA, the objects might be bit strings, vectors of real numbers or some custom representation. In EDA, this explicit representation of the population is replaced with a probability distribution over the choices available at each position in the vector that represents a population member. Moreover, in GA, new candidate solutions are often generated by combining and modifying existing solutions in a stochastic way. The underlying probability distribution of new solutions over the space of possible solutions is usually not explicitly specified. In EDA, a population may be approximated with a probability distribution and new candidate solutions can be obtained by sampling this distribution. Compared with traditional GA, EDA can solve nonlinear variable coupling problems for complex optimization.

Software reliability predictions are used for various purposes, such as software planning, reliability assessment, detecting faults in manufacturing processes, and evaluating risks. As reliability prediction plays an increasingly important role in assessing the performance of software systems, intensive studies have been carried out to ensure software reliability. The rest of this paper is organized as follows. Section 2 describes SVR model, expressing it as a combinatorial optimization problem with constraints. Section 3 explains the improved EDA (IEDA) and gives a model of optimizing SVR parameters based on IEDA. In Section 4, we give some assessing methods of the software reliability. Section 5 describes the numerical experiments and the results. Finally, Section 6 shows the conclusions from the experiment results.

2. Support vector regression

The performance of SVR depends on the rational optimization of parameters, and the optimization of these parameters is important to predict accurately. The traditional methods of optimizing parameters are: experience selection method (ESM), gradient descent method (GDM), and Bayesian method (BM). However, these methods have their own disadvantages. For example, ESM requires a large amount of experience and domain knowledge in order to obtain the appropriate parameters, and otherwise it is difficult to obtain the appropriate parameters. GDM is very sensitive to the initial point. In addition, GDM is a linear search method, and it is easy to fall into local minimum. Disadvantage of BM is to need some priori knowledge of parameter space for optimizing parameters, and it also needs more computation and computational complexity. In addition, this technique does not guarantee the outcome of better parameters. In fact, some researches have studied how to apply intelligence method to optimize parameters of SVR [18–20].

Suppose $\{(x_1, d_1), (x_2, d_2), \ldots, (x_n, d_n)\} \subset R^m \times R$ is training set, where R^m is the space of the input features x_i, and d_i is the phenomenon under investigation, i.e., the actual value. In ε-SVR [19], the goal is to find a function $f(x)$ whose deviation from each target d_i is at most ε for all training data, and at the same time, is as "flat" as possible. For the sake of clarity, we consider the following objective function in the linear case, i.e., F: $R^m \to R$, such that

$$y = f(x) = w\phi_i(x) + b \tag{1}$$

where $\phi_i(x)$ is the input features, and w and b are coefficients. The coefficients (w and b) are estimated by minimizing the following regularized risk function:

$$R_{\mathrm{SVR}}(C) = R_{\mathrm{emp}} + \frac{1}{2}\|w\|^2 = C\frac{1}{n}\sum_{i=1}^{n} L_\varepsilon(d_i, y_i) + \frac{1}{2}\|w\|^2 \tag{2}$$

$$L_\varepsilon(d, y) = \begin{cases} \left|d - y\right| - \varepsilon, & \text{if } \left|d - y\right| \geq \varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where R_{SVR} and R_{emp} represent the regression and empirical risk, respectively, C and ε are two parameters. In Eq. (2), $L_\varepsilon(d, y)$ is called the ε-insensitive loss function. $\|w\|^2/2$ is used as a measure of the flatness of the function.

Two positive slack variables ξ and ξ^*, which represent the distance from actual values to the corresponding boundary values of ε-tube, are introduced. Then, Eq. (2) is transformed into the following convex optimization problem:

$$\text{Min} \quad R_{\mathrm{SVR}}(w, \xi, \xi^*) = C\sum_{i=1}^{n}(\xi_i + \xi_i^*) + \frac{1}{2}\|w\|^2 \tag{4}$$

$$s.t. \begin{cases} w\phi(x_i) + b_i - d_i \leq \varepsilon + \xi_i^*, \\ d_i - w\phi(x_i) - b_i \leq \varepsilon + \xi_i, \qquad i = 1, 2, \ldots, n \\ \xi_i, \xi_i^* \geq 0. \end{cases} \tag{5}$$

By introducing Lagrange multipliers and exploiting the optimality constraints, the decision function given by Eq. (1) has the following explicit form [21]

$$f(x, \alpha_i, \alpha_i^*) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x, x_i) + b \tag{6}$$

where $K(x_i, x_j)$ is called the kernel function, α_i and α_i^* are the so-called Lagrange multipliers. In Eq. (6), they satisfy the equality $\alpha_i * \alpha_i^* = 0$. α_i and α_i^* are calculated by maximizing the dual function of Eq. (4), and the maximal dual function in Eq. (4), which has the following form:

$$\text{Max} \quad R(\alpha_i, \alpha_i^*) = \sum_{i=1}^{n} d_i(\alpha_i - \alpha_i^*) - \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) \tag{7}$$

under the constraints, $\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0$; $0 \leq \alpha_i \leq C, i = 1, 2, \ldots, n$; $0 \leq \alpha_i^* \leq C, i = 1, 2, \ldots, n$.

The value of the kernel is the inner product of the two vectors x_i and x_j in the feature space $\phi(x_i)$ and $\phi(x_j)$, so $K(x_i, x_j) = \phi(x_i) * \phi(x_j)$.

Any function that satisfies Mercer condition [21] can be used as the kernel function. Generally, the Gaussian function will yield better prediction performance [15]. Thus, in this work, the Gaussian function, $\exp(-\left\|x_i - x_j\right\|^2/2\sigma^2)$, is used in the SVR. Where, σ^2 represents the bandwidth of Gaussian kernel.

So, to build a SVR model efficiently, we need to select three positive parameters ε, σ and C.

3. IEDA and IEDA-SVR model

Although performance of the EDA is better than GA's, the EDA still has drawbacks. For example, in EDA evolutionary process, the individuals in the population are easy to trend to the same solution and the population diversity declines rapidly. These drawbacks affect the performance of the EDA. In order to maintain population diversity, we improve EDA, and obtain the IEDA, and then the IEDA is used to optimize parameters of the SVR.

3.1. Improved EDA

The chaotic sequence has the characteristics of ergodicity, randomness, initial sensitivity and regularity, and the chaotic mutation operation is an important way to maintain population diversity [22,23]. In this paper, the chaotic mutation was introduced into the traditional EDA. IEDA is described in detail as follows.