Optimizations of LSSVR parameters using GA for software effort estimation from clustered data

Wen Zhou Wang a, Dengsheng Wu b, *

a. School of Economics and Business Administration, Beijing Normal University, Beijing 100875, P.R. China
b. Institute of Policy and Management, Chinese Academy of Sciences, Beijing 100190, P.R. China

* Corresponding author. Email: wds@casipm.ac.cn, Tel: +86 10 59358805

Abstract: As an important part in software process, software effort estimation plays a central role in controlling software cost, reducing software risk and guaranteeing software quality. Both the software industry and academic communities have become more and more concerned about a reliable and accurate estimation of software effort. However, no model is proved successful at effectively and consistently predicting software development cost. This paper examines the potentials of a software effort estimation model by integrating a genetic algorithm (GA) to the least squares support vector regression (LS-SVR) and analyzes the influence of different fitness function and number of categories to accuracy of the GA-LS-SVR model. The GA is adopted to optimize the parameters in LS-SVR for software effort estimation. Moreover, four evaluation criterions are regarded as fitness function in the GA-LS-SVR model, and K-means method is employed to cluster historical dataset to reduce the influence of data
heterogeneity. The ISBSG dataset is used to confirm the validity of the model. Firstly, we get the best number of clusters in K-means by calculating minimum number of clusters with best estimation accuracy in GA-LS-SVR model. Secondly, according to different fitness function, GA-LS-SVR model is divided into four different types: MGLS, DGLS, PGLS, and MPGLS. The empirical results from ISBSG show that the GA is suitable to optimize the parameters in LS-SVR, and GA-LS-SVR significantly outperforms other methods in ISBSG dataset. Moreover, different fitness functions in GA-LS-SVR generate similar results. The results also indicate that clustering historical projects into a homogeneous group by K-means is an effective measure to improve estimation accuracy in GA-LS-SVR model.

**Keywords:** software effort estimation, least squares support vector regression, parameter optimization, k-means clustering, genetic algorithm

### 1 Introduction

As an important part in software process, software effort estimation plays a central role in controlling software cost, reducing software risk and guaranteeing software quality [1]. Therefore, estimating the software effort accurately has become a significant issue in the progress of software planning and managing. In the light of the significant role of software effort estimation in the process of software development, various estimation methods are used to improve the estimation accuracy of software effort, especially in the early stage of software
development life cycle [2, 3].

From a comprehensive review [4], software effort estimation methods could be classified into the following six categories: parametric models, expert judgment, learning oriented techniques including CBR method, regression based methods, dynamics based models and composite methods [5-7]. Recently, regression based methods, like OLS regression [8], support vector regression (SVR) [9], ordinal regression [10], multiple additive regression trees [11], are adopted to estimate software effort frequently. Among these methods, SVR is a kernel regression method based on the principle of structural risk minimization and usually applied to small sample which is entirely different from the traditional statistical methods based on large sample [12]. In order to reduce computational complexity of SVR model, Least Squares Support Vector Regression (LS-SVR) is proposed by Suykens, which is reformulation to standard SVR that lead to solving linear KKT systems [13].

This study utilizes the LS-SVR model to estimate the software effort, given its lower computational complexity and small sample feature. Two problems needed to be solved in the LS-SVR model are determining suitable parameters value and choosing a fitness function to assess the model performance. The study employs Generic algorithm (GA) to optimize the parameters value in the LS-SVR model, thereby constructing a GA-LS-SVR model for software effort estimation. Meanwhile, traditional measurement indicators, such as MMRE, MdMRE, and Pred(0.25), are adopted as fitness function respectively in the GA-LS-SVR model.
In order to validate the feasibility of GA-LS-SVR model in software effort estimation, a standard data set (ISBSG R10) provided by the International Software Benchmarking Standards Group (http://www.isbsg.org) is used. As we know that ISBSG dataset is a public, multi-organization dataset, and some research indicate that multi-organization dataset may influence the accuracy of software effort estimation [14]. Therefore, the K-means method is used in the study to cluster the sample into sets of several categories to reduce the influence of data heterogeneity.

The present paper aims to examine the potential benefits of software effort estimation using the GA-LS-SVR model and analyzes the influence of different fitness functions and the number of categories to accuracy of the GA-LS-SVR model. The rest of the paper is organized as follows: Section 2 presents the related works on regression based methods and data clustering methods in software effort estimation. Section 3 provides the concept of GA-LS-SVR method and describes the evaluation criterion used in our study. Clustering data based on K-means method is introduced in Section 4. Section 5 presents an experiment from ISBSG dataset and analyzes the accuracy of GA-LS-SVR model with different fitness functions and clustered data. Last section concludes.

2 Related work

According to Boehm’s opinion, regression-based techniques are the most popular ways of estimating software effort [4]. Among these regression-based
techniques, the ordinary least square (OLS) regression is one of the most well-known methods used in establishing software effort estimation model. Huang et al. employ the OLS regression to establish effort estimation model based on ISBSG dataset and their empirical results show that OLS regression can fit the relationship between effort and other drivers well [8]. Miyazaki et al. used robust regression to estimate software effort, which is an improvement over the standard OLS approach and can alleviate the common problem of outliers in observed software data [15]. Sentas et al. explored the possibility of using ordinal regression to model software effort estimation and their results show that the fitting and predictive results using ordinal regression in the ISBSG R7 database are quite encouraging [10]. Elish adopted multiple additive regression trees (MART) to model software effort estimation based on NASA software project dataset, and his results indicate that improved estimation accuracy of software project effort has been achieved using MART [11].

Another approach that can be classified as regression-based techniques is SVR model. The small sample feature of SVR is suitable for the limited number of software effort data. However, suitable parameters value needed to be determined for SVR model before application, which can influence the model accuracy seriously. Traditional method to solve this problem is determining the model parameters according to personal experience and expert knowledge. For example, to solve this problem, Oliveira chose different initial value of parameters $C$, $\varepsilon$
and $\gamma$ for his SVR model, and determined the suitable value of parameters by applying various combination of parameter value to test the model [9]. Recently, GA is adopted to optimize parameters of SVR given its wide and successful application in various types of optimization problems [16]. Empirical results indicate that GA based optimizing parameters of SVR is a feasible approach to improving the accuracy of software effort estimates [17,18]. Meanwhile, empirical results also show that GA-SVR model is time-consuming especially when k-fold cross validation is used to establish and validate the model. Therefore, we adopts LS-SVR model proposed by Suykens to estimate the effort of software with optimized parameters derived from GA, which is called GA-LS-SVR in the study. LS-SVR is reformulated to standard SVR to solve linear KKT systems. A significant advantage of LS-SVR is the lower computational complexity than the standard SVR [13].

Another common problem in software effort estimation is the lack of a reliable and accessible historical database. Therefore, ISBSG dataset was used for many studies and applications. The ISBSG dataset is multi-company dataset. So some studies compared the accuracies of software effort estimate derived from cross-company data and single-company data. Although some research found that the cross-company model gives similar prediction accuracy to that of the single-company model [19], more research indicated that the cross-company model can't give accurate predications compared with the single-company model. Additionally, some research clustered the dataset from other perspectives [14, 20,
Heiat divided the dataset into two groups by programming languages, and the results indicated that effort estimating from these two datasets reveal slightly different accuracies [22]. Chen and Kao used support vector machine to classify the training data set [23]. Lee et al. tried to predict temperature with fuzzy logical relationships and genetic algorithms [24]. Huang et al. adopted K-means and Scheffe to cluster dataset into sets of categories according to different effort drivers and their empirical results showed that the estimation accuracies do not reveal significant differences within the respective dataset clustered by each software effort driver [8]. However, the past researchers did not estimate software effort from sets of categories clustered by all effort drivers. The ISBSG dataset is clustered in different categories by K-means according to all effort divers in this study. Moreover, the best K value in K-means is calculated by GA-LS-SVR model.

3 LS-SVR method and parameters optimization

3.1 Introduction of LS-SVR method

Motivated by statistical learning theory based on the principle of structural risk minimization, SVR was developed by Vapnik and his colleagues in 1995 [12, 25], which has received increasing attention. Since it is used to solve a quadric optimization problem, the computational complexity of SVR model is very high, especially for the problems which deal with mass data or need on-line computation. LS-SVR proposed by Suykens [13] which results in a set of linear
equations instead of a quadratic programming problem in SVR model is a modification version of SVR and a more simple method than SVR [26]. LS-SVR is equivalent to solve a linear optimization problem as the \( \varepsilon \)-insensitive loss function used by SVR is replaced by a sum square error loss function and accordingly the inequality restriction is replaced by the equation restriction, which gives LS-SVM the lower computational complexity.

In LS-SVR, training samples are assumed to be \( \{(x_k, y_k)\}_{k=1}^{N} \), in which \( x_k \) is the input vector and \( y_k \) is the target vector of the \( k^{th} \) sample. To find out the underlying relationship between input vector and target output vector, the data space is mapped to high dimension feature space via a nonlinear mapping function \( \phi(x) \). The nonlinear mapping relationship between input variable and target output variable takes the following form

\[
y = f(x; w, b) = w^T \phi(x) + b
\]

where \( \phi(x) \) is a nonlinear mapping function, \( w \) is the \( i^{th} \) weights, and \( b \) is a bias. Given training set \( \{(x_k, y_k)\}_{k=1}^{N} \) the following optimization problem is formulated in the LS-SVR model

\[
\min J(w, e) = \frac{1}{2} w^T w + \frac{C}{2} \sum_{k=1}^{N} e_k^2
\]

s.t. \( y_k = w^T \phi(x) + b + e_k \), for \( k = 1, \ldots, N \).

This is a form of ridge regression, where \( C \) is the regularization parameter, \( e_k \) is the \( k^{th} \) approximation error between predicted and actual values. The polynomial of Lagrange duality problem is presented as
\[ L(w, b, e; a) = \frac{1}{2} w^T w + \frac{C}{2} \sum_{k=1}^{N} e_k^2 - \sum_{k=1}^{N} a_k \{ w^T \phi(x) + b + e_k - y_k \}, \] (3)

where \( a_k \) is the \( k^{th} \) Lagrange multiplier. The KKT conditions can be expressed by

\[ \begin{align*}
\frac{\partial L}{\partial w} &= 0 \Rightarrow w = \sum_{k=1}^{N} a_k \phi(x_k) \\
\frac{\partial L}{\partial b} &= 0 \Rightarrow \sum_{k=1}^{N} a_k = 0 \\
\frac{\partial L}{\partial e_k} &= 0 \Rightarrow a_k = C e_k \\
\frac{\partial L}{\partial a_k} &= 0 \Rightarrow w^T \phi(x_k) + b + e_k - y_k = 0 \\
\end{align*} \] (4)

Eliminate \( e \) and \( w \), and the solution can also be written as the following linear equations set

\[ y_j = \sum_{j=1}^{n} \sum_{k=1}^{m} \alpha_{j,k} K(x_{j,k}, x_{j,j}) + b + \sum_{k=1}^{m} \alpha_{j,k} / C \quad k = 1, \ldots, N. \] (5)

Using the mercer condition, the kernel function can be defined as \( K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \). Eq. (5) can be rewritten as

\[ \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} w^T \\ \Omega + \frac{I}{C} \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} \] (6)

where \( b \) is a scalar, \( y = (y_1, y_2, \ldots, y_N)^T \), \( \tilde{1} = (1, 1, \ldots, 1)^T \), \( a = (a_1, a_2, \ldots, a_N)^T \) and \( \Omega = \phi(x_i)^T \phi(x_l) = K(x_i, x_l) \) \( (k, l = 1, 2, \ldots, N) \). Commonly used kernel functions include: linear kernel \( K(x_i, x_j) = x_i^T x_j \), Polynomial kernel \( K(x_i, x_j) = (x_i^T x_j + 1)^d \), Gaussian kernel or KBF kernel \( K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / \sigma^2) \), and MLP kernel \( K(x_i, x_j) = \tanh(kx_i^T x_j + \theta) \), where \( d, \sigma, k \) and \( \theta \) are kernel parameters.

Resulting LS-SVR model for function estimation can be obtained as
\[ y(x) = \sum_{i=1}^{N} a_i K(x_i, x) + b. \quad (7) \]

A LS-SVR model contains two or three parameters. One is regularization parameter and the others are the kernel function parameters. LS-SVR solves linear equations and will lead to important reduction in calculation complexity. Compared with SVR, LS-SVR characterizes higher stability, faster training speed and better control strategy, so LS-SVR can also be well used in non-positive nuclear matrix function.

3.2 Parameters optimization of LS-SVR method using GA

We employs Gaussian kernel in LS-SVR model, and the generalization performance of LS-SVR and its efficiency are dependent upon penalty parameter C and kernel function parameters \( \sigma \)[27]. Choosing the best parameters is an important step for LS-SVR model. In our work, GA is used to optimize the model parameters in order to improve the accuracy and generalization of LS-SVR model for software effort estimation. GA can find the best parameters by exploring all regions of state space and exploiting potential areas through mutation, crossover and selective operations applied to individuals in the population [28]. In the model, the effort drivers are first normalized in the range between 0 and 1. Meanwhile, three-fold cross validation is used to measure the model performance. We divide the total available data into three subsets and then select two of them randomly as train dataset and the rest of samples are treated as test dataset. The procedure is repeated three times so that each subset will be used once for validation and the
average criterion value of three times is treated as model performance. In the study, the detailed steps of GA-LS-SVR can be summarized as follows:

Step 1: Initialize the parameters used in GA, which includes the number of evolutionary generations, population size, and the number of subpopulations, individuals per subpopulation, and the range of the LS-SVR parameters. The two LS-SVR parameters, C and $\sigma$, are directly coded to generate the chromosome randomly.

Step 2: Setting $t=0$, where $t$ is for evolulutional generation.

Step 3: In this context, a three-fold cross validation technique is used to overcome over fitting phenomenon. The performance of the set of parameters is measured by MMRE (MdMRE or Pred(0.25)), which will be introduced detailed in section 3.3.

Step 4: Setting $t = t + 1$.

Step 5: Selection, crossover and mutation method are implemented to generate new chromosome.

Step 6: If the executed generation number equals to the special generation number, the algorithm terminates. The best C and $\sigma$ would be output according to the optimum fitness function value. On the contrary, to obtain the best C and $\sigma$, steps 3-5 are repeatedly executed until C and $\sigma$ are satisfied with minimum model error.

The range of $\sigma$ and C are set as [0, 1000] and [0, 1000] respectively. The flow diagram of GA-LS-SVR model with clustered data from K-means is shown in Fig. 1.
3.3 Assessment of model fitness and evaluation criterion

A fitness function assessing the performance for each chromosome must be designed before searching for the optimal values of the LS-SVR parameters. Traditional measurement indicators have been proposed and employed to evaluate the prediction accuracy of models such as MAPE, RMSE and RMS. We put forward some new fitness function according to the characteristic in estimating software effort. The MMRE, MdMRE, Pred(0.25) and the combination of these evaluation criterions are used to be fitness function in LS-SVR model. Moreover, The MMRE, MdMRE, Pred(0.25) are also used to evaluate the performance of estimation methods [8].

The MMRE is defined as below:

\[ MRE_i = \frac{|Y_i - \hat{Y}_i|}{Y_i} \quad i = 1, \ldots, n \]  \hspace{1cm} (8)

\[ MMRE = \frac{1}{n} \sum_{i=1}^{n} MRE_i \]  \hspace{1cm} (9)

where \( n \) denotes the number of projects, \( Y_i \) denotes the actual effort of \( i^{th} \) project, and \( \hat{Y}_i \) denotes the estimated effort of \( i^{th} \) project. Small MMRE value indicates the low level of estimation error. However, this metric is unfair since it penalizes overestimation more seriously than underestimation.

The MdMRE is the median of all the MREs, which is defined as follow:

\[ MdMRE = \text{median}(MRE). \]  \hspace{1cm} (10)

It is an aggregate measure which is less sensitive to extreme values. It exhibits a similar pattern to MMRE but is more likely to select the true model.
especially in the underestimation cases. The Pred (0.25) is the percentage of predictions that fall within 25% of the actual value,

\[
Pred(0.25) = \frac{m}{n}
\]

(11)

where \( n \) denotes the number of projects and \( m \) is the number of projects whose MRE is less than or equal to 25 percent.

4 Clustering data based on K-means method

4.1 Introduction of K-means method

This section describes an outline of K-means algorithm used for clustering the historical software dataset. K-means algorithm is based on the minimization of performance index, which is defined as the sum of squared distances from all points in a cluster domain to the cluster center [29].

Given a set of \( n \) objects denoted as \( X = \{x_i\}_{i=1}^{n} \). Suppose \( K \) is a positive integer. The objective of clustering \( X \) is to find a partition, which divides the objects in \( X \) into \( K \) disjoint clusters. Let \( \mu_k \) be the mean of cluster \( c_k \) \((k = 1, 2, \ldots, K) \). The squared error between \( \mu_k \) and the points in cluster \( c_k \) is defined as

\[
J(c_k) = \sum_{x_i \in c_k} \|x_i - \mu_k\|^2
\]

(12)

The goal of K-means is to minimize the sum of the squared error over all \( K \) clusters

\[
J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \|x_i - \mu_k\|^2
\]

(13)

K-means starts with an initial partition with \( K \) clusters selected from the n
objects, which initially represents a cluster center. For the remaining objects, an object is assigned to a cluster to which it is the most similar, based on the distance between the object and the cluster mean. Then the new mean for each cluster is calculated. This process iterates until the criterion function converges.

4.2 Hybrid GA-LS-SVR model with K-means based clustered data

The system structure diagram for GA-LS-SVR with K-means based clustered data is illustrated in Fig. 1. All input and output data were standardized and normalized to [0, 1] in order to eliminate the influence of different dimension. Here, population size of GA is set as 50; hybrid probability and mutation probability are set as 0.4 and 0.2; evolutional generation is set as 1000. The kernel functions used in LS-SVR is Gaussian kernel \( K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / \sigma^2) \).

The calculation is accomplished by using LS-SVMlab and GAOT MATLAB toolbox.

Fig. 1 the flow diagram of GA-LS-SVR model with K-means based clustered data
5 Experiment analysis

5.1. Dataset and Data pre-processing

To support repeatability, the database used in the study comes from the International Software Benchmarking Standards Group (http://www.isbsg.org). The ISBSG R10 is released in 2007 containing data from 4106 software projects and each historical software project has 106 data items. Generally, the software projects in this dataset are heterogeneity data.

In order to assess the accuracy of the effort estimation models established from the data with clustering, a suitable subgroup of software projects was derived from the ISBSG data repository for modeling. As recommended by ISBSG, only Quality Rating of A or B data are selected for the study. There are 3811 sets of data that fulfill the above condition. To ensure a fair comparison of all the applied techniques and the use of the same projects as the basis for comparing predictions, no missing values will be considered in the study. After removing the projects and attributes containing missing values, we finally select 523 sets of project data with 8 project attributes for the study. The description of the variables is shown in Table 1.
Table 1 summary of the variables in ISBSG dataset

<table>
<thead>
<tr>
<th>ID</th>
<th>Feature</th>
<th>Full name or explanation</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Effort</td>
<td>Effort</td>
<td>Numeric</td>
</tr>
<tr>
<td>2</td>
<td>FunSize</td>
<td>Functional Size</td>
<td>Numeric</td>
</tr>
<tr>
<td>3</td>
<td>MTSize</td>
<td>Max Team Size</td>
<td>Numeric</td>
</tr>
<tr>
<td>4</td>
<td>DevType</td>
<td>Development Type</td>
<td>Nominal</td>
</tr>
<tr>
<td>5</td>
<td>DevPlatform</td>
<td>Development Platform</td>
<td>Nominal</td>
</tr>
<tr>
<td>6</td>
<td>LanType</td>
<td>Language Type</td>
<td>Nominal</td>
</tr>
<tr>
<td>7</td>
<td>RMethod</td>
<td>How Methodology Acquired</td>
<td>Nominal</td>
</tr>
<tr>
<td>8</td>
<td>YearP</td>
<td>Year of Project</td>
<td>Numeric</td>
</tr>
</tbody>
</table>

Because LS-SVR handles only numerical data, nominal data has to be converted to numeric data in the GA-LS-SVR model. 1, 2 and 3 indicate New Development, Enhancement and Re-development respectively in DevType feature. PC, MR and MF in DevPlatform feature are replaced with 1, 2 and 3. Moreover, 1, 2, 3 and 4 represent 2GL, 3GL, 4GL and APG in LanType feature. At last, Developed In-house, Purchased Combined and Developed/Purchased are replaced with 1, 2 and 3 in RMethod feature.

5.2. Estimating software effort using GA-LS-SVR model with clustered data

5.2.1 Determining best number of clusters in K-means

A critical problem needed to be solved is determining the best number of clusters when K-means is used to cluster the ISBSG dataset for GA-LS-SVR model. Given that clustered data have reduced the influence of data heterogeneity,
GA-LS-SVR model can improve estimation accuracy based on clustered data. In theory, the accuracy of GA-LS-SVR model will increase with the number of sample categories increasing. However, it is difficult to classify the new sample into the suitable category when the number of clusters is too big. The paper determines the best number of clusters in K-means method by comparing the estimation accuracy of different number of clusters. Firstly, the ISBSG dataset is assigned to $k$ different clusters by K-means ($k = 1, \ldots, 20$). The accuracy of each cluster is calculated by GA-LS-SVR shown in Fig. 2 and Fig. 3 in terms of the error measure which are MMRE, MdMRE and Pred(0.25).

![Fig. 2 the accuracy of GA-LS-SVR form different number of cluster on train dataset](image-url)
Fig. 3 the accuracy of GA-LS-SVR form different number of cluster on test dataset

In Fig. 2 and Fig. 3 we can get the conclusion that clustering historical projects into some homogeneous groups to build effort estimation model is an effective measure to improve estimation accuracy. The figures also show comparison results between the clustered data and the data without clustering. When \( k=1 \), which indicates that the historical data isn’t clustered, the GA-LS-SVR model generates the worst estimation result. Its value of MMRE is 0.78 in train dataset and 0.81 in test dataset. With increasing of the number of sample categories, GA-LS-SVR model obtains more precise estimation result. However, too many clusters will result in new problem when estimating the effort of new software project. It is difficult to classify the new software project into suitable cluster. Therefore, we need improve the estimation accuracy through as few clusters as possible.

From Fig. 2 and Fig. 3, we can also see that the optimal clustering number for
ISBSG data is 13, which is minimum number of clusters with best estimation accuracy. In order to analyze the influence of cluster fact to estimation accuracy, this paper investigates the sample character and estimation accuracy under each cluster when cluster number is 13. Table 2 show the result including number of samples, mean of effort and MMRE of each cluster under the circumstances.

<table>
<thead>
<tr>
<th>cluster ID</th>
<th>number of samples</th>
<th>mean of effort</th>
<th>MMRE of each cluster on test dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>maximum</td>
</tr>
<tr>
<td>1</td>
<td>44</td>
<td>4681</td>
<td>0.189</td>
</tr>
<tr>
<td>2</td>
<td>39</td>
<td>12405</td>
<td>0.275</td>
</tr>
<tr>
<td>3</td>
<td>37</td>
<td>33990</td>
<td>0.178</td>
</tr>
<tr>
<td>4</td>
<td>61</td>
<td>767</td>
<td>2.968</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>6074</td>
<td>0.220</td>
</tr>
<tr>
<td>6</td>
<td>42</td>
<td>1941</td>
<td>0.196</td>
</tr>
<tr>
<td>7</td>
<td>47</td>
<td>1356</td>
<td>0.238</td>
</tr>
<tr>
<td>8</td>
<td>39</td>
<td>20415</td>
<td>0.323</td>
</tr>
<tr>
<td>9</td>
<td>66</td>
<td>8358</td>
<td>0.525</td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td>279</td>
<td>0.371</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>2574</td>
<td>0.291</td>
</tr>
<tr>
<td>12</td>
<td>21</td>
<td>2261</td>
<td>0.496</td>
</tr>
<tr>
<td>13</td>
<td>34</td>
<td>3529</td>
<td>0.310</td>
</tr>
</tbody>
</table>

The maximum sample number and minimum sample number of the clusters are 66 and 11 respectively, and the average sample number of the cluster is 40.

From table 3, we find that the worst estimation result is obtained in cluster 4, whose mean of effort is 767. According to the statistical characteristic of the whole historical dataset, the projects in cluster 4 belong to small software project, which illustrates that we should pay more attention when small new software project needed to estimated. Moreover, the maximum MMRE and mean MMRE in
cluster 4 are 2.968 and 0.487, which are far greater than the others. We believe that there are some abnormal samples in historical dataset, which result in the phenomenon. So we should detect and delete abnormal samples before estimation in further research.

5.2.2 Comparative result based on different methods

After determining best number of clusters in K-means, GA-LS-SVR method for software effort estimation should be compared with other methods in order to find its advantage and shortcoming. Four evaluation criterions are regarded as fitness function in GA-LS-SVR model, which are MMRE, MdMRE, Pred(0.25) and combination of them. So we can divide the GA-LS-SVR model into four different types based on different fitness function used in estimation process. MGLS method employs MMRE to measure the fitness in GA-LS-SVR model; DGLS employs MdMRE to measure the fitness in GA-LS-SVR model; PGLS employs Pred(0.25) to measure the fitness in GA-LS-SVR model; And MPGLS employs a combination of MMRE and Pred(0.25) to measure the fitness in GA-LS-SVR, which is MMRE+ (1 - Pred(0.25)).

Moreover, some other methods used in published paper are used to compare with the four methods proposed in the study, which are UWA, LWA, NWA proposed by Huang in 2006[30]; OWC, CMA proposed by Huang in 2008[8]; CART and ANN proposed in other research but used by Huang in 2006[30]. UWA, LWA and NWA belong to analogy method and are abbreviation of unequally weighted analogy, linearly weighted analogy and nonlinearly weighted analogy.
OWC is a traditional OLS method and CMA is improved OLS method based on clustered dataset by methodology acquired in ISBSG dataset. The comparison result is shown in table 3.

Table 3 the comparison of estimation results based on different method

<table>
<thead>
<tr>
<th>method</th>
<th>Training</th>
<th>Testing</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MMRE</td>
<td>MdMRE</td>
<td>Pred(0.25)</td>
</tr>
<tr>
<td>MGLS</td>
<td>0.128</td>
<td>0.087</td>
<td>0.950</td>
</tr>
<tr>
<td>DGLS</td>
<td>0.147</td>
<td>0.082</td>
<td>0.925</td>
</tr>
<tr>
<td>PGLS</td>
<td>0.090</td>
<td>0.052</td>
<td>0.965</td>
</tr>
<tr>
<td>MPGLS</td>
<td>0.121</td>
<td>0.074</td>
<td>0.952</td>
</tr>
<tr>
<td>UWA</td>
<td>0.49</td>
<td>0.37</td>
<td>0.41</td>
</tr>
<tr>
<td>LWA</td>
<td>0.50</td>
<td>0.30</td>
<td>0.47</td>
</tr>
<tr>
<td>NWA</td>
<td>0.42</td>
<td>0.33</td>
<td>0.48</td>
</tr>
<tr>
<td>CART</td>
<td>0.85</td>
<td>0.29</td>
<td>0.41</td>
</tr>
<tr>
<td>ANN</td>
<td>1.30</td>
<td>0.64</td>
<td>0.17</td>
</tr>
<tr>
<td>OWC</td>
<td>0.63</td>
<td>0.36</td>
<td>0.40</td>
</tr>
<tr>
<td>CMA</td>
<td>0.65</td>
<td>0.37</td>
<td>0.37</td>
</tr>
</tbody>
</table>

Table 3 shows the estimation results of different methods based on ISBSG projects. The results show that there is no significant difference between four different methods proposed in this study. PGLS can achieve slightly more accurate performances than others in train dataset and MPGLS is more accurate in test.
dataset. In all, the performance of four methods proposed in this study is similar, which indicates that different fitness function in GA-LS-SVR model for software effort estimation will generate similar result. So, MMRE should be chosen to calculate the fitness value in optimization model for further research according to traditional opinion.

When comparing GA-LS-SVR model with other methods, the advantages of GA-LS-SVR model is very obvious. The mean MMRE of four methods based on GA-LS-SVR in train dataset and test dataset are 0.121 and 0.194, which are far below those achieved in other methods. As to MdMRE and Pred(0.25), four methods based on GA-LS-SVR share same advantages as well. For the purpose of better comparison between each method, this study draw the accuracy figure of the models based on train dataset and test dataset separately. In comparison figure, CART and ANN are excluded for low accuracy. The MMRE of CART and ANN are 0.85, 1.30 in train dataset and 1.89, 1.70 in test dataset. Their accuracies are far worse than the other methods, so they are excluded from comparison figure.
Fig. 4 shows the accuracy of each method at train dataset of ISBSG. The MMRE measures average error and small MMRE value indicates the low level of estimation error. In train stage, MGLS, DGLS, PGLS and MPGLS are superior to other models obviously for MMRE criterion. The maximum MMRE value of the four methods proposed in this study is 0.147, which is less than half what in other models. The MdMRE value measures median error and small MdMRE value indicates the low level of estimation error. The four methods proposed in this study generate low MdMRE value, which mean that they have high accuracy. The Pred(0.25) value is the proportion of project estimates whose MREs are equal to or less than 0.25 and large Pred(0.25) value indicates the low level of estimation error. The maximum value of the method in Fig. 4 approximates 1. These analyses lead us to a conclusion that the MGLS, DGLS, PGLS and MPGLS proposed in this study are superior to other methods used by Huang in 2006 and 2008.
Figure 5 accuracy comparisons on testing dataset of ISBSG

The accuracy of each method at train dataset is shown in Fig. 5. MGLS, DGLS, PGLS and MPGLS generate similar result according to MMRE and MdMRE. MPGLS generates slightly better result according to Pred(0.25). The four methods proposed in this study are superior to other models obviously as in train dataset. Moreover, we can get a conclusion that UMA, LWA, NWA and OWC will generate “over learning” phenomenon. There isn’t “over learning” phenomenon in the four methods proposed in this study regarding that they can generate accurate estimation in train dataset as well as in test dataset.

6 Conclusions

The study aims to examine the potential benefits of software effort
estimation using the GA-LS-SVR method, and analyze the influence of different fitness function and the number of categories simultaneously. The empirical study from ISBSG dataset shows that clustering historical projects into a homogeneous group to build effort estimation models is an effective measure to improve estimation accuracy. Moreover, by calculating minimum number of clusters with best estimation accuracy in GA-LS-SVR model, we can get the best number of clusters in K-means. According to different fitness function, GA-LS-SVR model is divided into four different types: MGLS, DGLS, PGLS and MPGLS. The calculation from ISBSG dataset indicates that there isn’t significant difference between four different methods proposed in this study. When comparing with some other methods used by Huang in 2006 and 2008, we find that MGLS, DGLS, PGLS and MPGLS are superior to other models obviously for MMRE, MdMRE and Pred(0.25) criterion.

Acknowledgements

The research is partially supported by the National Natural Science Foundation of China (71201156, 71571179, 91218302), the Fundamental Research Funds for the Central Universities (SKZZY2013014), and Youth Innovation Promotion Association of the Chinese Academy of Sciences.

References


