Estimating Software Cost with a Weighted Feature Selection and Support Vector Regression with Mixture of Kernels Ensemble Learning Method

Tiejun Jiang*, Chengjie Zhou, Huaiqiang Zhang

Department of Management Engineering and Equipment Economy Naval University of Engineering Wuhan, China

*Corresponding author e-mail: tiejunjiang@126.com

Abstract. In traditional feature selection methods, there are only two possible outcomes: the feature is selected or the feature is not selected, which will lead to the loss of feature information. In this paper, considering the deficiencies of traditional methods and the requirement of software cost estimation, a weighted feature selection (WFS) method with the supervised wrapper mode is used in software cost estimation, which can effectively distinguish the influence of different features on the cost. In view of the good application effect of support vector regression (SVR), as well as a good performance of the mixture of kernels, the relationship model among the features and the software cost is established based on SVR with the mixture of kernels. In addition, considering the consistency of feature selection and the establishment of cost estimation model, a joint optimization method based on hybrid particle swarm optimization (HPSO) is adopted, which can achieve the influence analysis of features and the optimization of cost estimation model. Experiments show that the proposed ensemble learning method is effective.

1. Introduction

Software cost estimation has long been an important topic in software project management. In order to make the accurate estimation, various statistical and artificial intelligence techniques have been developed in software cost estimation. These techniques may be grouped into two major categories: algorithmic models and non-algorithmic models. The first are the most popular techniques and are illustrated by estimation models such as COCOMO, PUTNAM-SLIM and function points analysis etc. Algorithmic models are derived from the statistical or numerical analysis of historical project data. With the increasing software complexity, non-algorithmic models are developed. Many researchers have diverted their attention to this alternative and in the particular to a set of methods based on neural networks (NNs), regression trees, rule induction and case-based reasoning (CBR), support vector regression (SVR) [1, 2].

Generally speaking, in order to avoid the loss of information, the features, which have impacts on the cost, are collected as much as possible in software cost estimation. However, in this condition, the introduction of unrelated features will reduce the estimation effect and bring the difficulties to the data collection, particularly in the application with a small sample. Therefore, it is necessary to select the
feature before the construct of cost estimation model, which can obtain the cost estimation model with a better estimation effect and realize the importance identification of features.

In the traditional feature selection method proposed, the effective feature subset is extracted from the candidates in order to eliminate the interaction among features, which benefits the construct of cost estimation model. However, only a simple choice of features is done in this method, as a result, the loss of some feature information will often happen because the feature is removed. Moreover, all of features can be simply attributed to two types: the important or the unimportant, and it is impossible to determine the differences among various features further just like which is more important than others in the same selected features.

However, in software cost estimation, it is important to study the influence degree of features on the cost. Therefore, in this paper, a weighted feature selection (WFS) method based on the supervised wrapper mode is put forward to establish the model among the features and the cost, which can effectively distinguish the influence of features on the estimation performance and improve the effect of the estimation model.

In this paper, WFS is combined with SVR and a WFS-based SVR ensemble adaptive learning methodology is proposed to estimate software cost. Firstly, different weights distinguish the influence of different features on the estimated cost; secondly, SVR with the mixture of kernels is used to establish the cost estimation model; thirdly, a hybrid particle swarm optimization algorithm (HPSO) is used to achieve the optimization of feature weight and estimation model, which greatly improve the efficiency and accuracy of the learning and training.

2. Methodologies

2.1. Support Vector Regression with the Mixture of Kernels
As an effective nonlinear system modeling tool, SVR has been widely used in many fields [3]. In this paper, software cost estimation model based on SVR is established to describe the relationship among the features and the cost. At the same time, considering the importance of kernel function to the performance of cost estimation, a kind of SVR modeling method based on the mixture of kernels is adopted.

The mixture of radial basis function (RBF) and polynomial kernels can be defined as [4]:

\[ K = \rho K_{\text{poly}} + (1 - \rho)K_{\text{rbf}} \]  

Where \( K_{\text{poly}} \) is the polynomial kernel and \( K_{\text{rbf}} \) is the RBF kernel. The characteristics of the mixture of kernels are determined by different values of \( \rho \) for different regions of the input space. \( \rho \) is a vector. Through this method, the relative contribution of both kernels to the model can be varied over the input space. In this paper, a uniform \( \rho \) over the entire input space is used.

2.2. Weighted Feature Selection
Feature selection is a very important problem in the cost estimation modeling, which can improve the model performance and provide a better explanation for the relationship among the features and the cost [5].

In the traditional feature selection method, the feature is set as a discrete variable, 0 or 1. If some feature is 1, then the feature is selected, otherwise the feature is discarded.

As for WFS, the selected feature is no longer a simple 0 or 1, but some value between 0 and 1. Therefore, it can be seen that the general feature selection is only a special case of WFS.

In our proposal, we will use a supervised wrapper feature selection algorithm which utilizes the precision estimation provided by SVR.
In view of the complexity of feature selection and parameter optimization in the wrapper mode, a joint optimization method based on HPSO is presented in this paper, which means the features and the parameters in the cost estimation model can be determined simultaneously.

2.3. Hybrid Particle Swarm Optimization

PSO is a parallel population-based computation technique, which is applied in the function optimization, neural network training, fuzzy system control and many other areas [6, 7].

The continuous PSO algorithm can be seen in [6] for details. The binary PSO algorithm is similar to the continuous PSO algorithm, where each particle takes the values of binary vectors with the length $n$ and the velocity represents the probability that a bit will take the value 1. The velocity remains unchanged, but the position is update as follows:

$$X_i(k+1) = \begin{cases} 0 & \text{if } \nu \geq f(V_i(k+1)) \\ 1 & \text{otherwise} \end{cases}$$

Where $r \in \text{Rand}(0,1)$, $f(V_i(k+1))$ is the sigmoid function.

In this paper, HPSO algorithm is used to implement the joint optimization to meet the needs of different types of variable optimization.

Because of the need for the joint optimization of the features and the parameters in the cost estimation model, the definition of variables adopts a hybrid form, which contains the features and the model parameters.

In order to fully enhance the generalization ability of the model, the 3-fold cross-validation method is used to evaluate the training effectiveness. For the evaluation of the test sample set, the error is measured by the root mean square error (RMSE) criterion, which is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

Where $\hat{y}_i$ ($i=1,2,\cdots,n$) is the estimated value, $y_i$ ($i=1,2,\cdots,n$) is the actual value, $n$ is the number of data points.

2.4. The Procedure

The overall structure for the estimation procedure of the proposed method in this paper is depicted in Fig. 1. The procedure consists of the preprocessing, training, and estimation stage.

Firstly, in the preprocessing stage, the data normalization is applied before the model establishment. In this paper, each attribute is scaled by the following method:

$$x_i = \frac{(x_i - \mu)}{\delta}$$

Where $x_i$ is a certain attribute value, $\mu$ is the mean value of the attribute, and $\delta$ is the standard deviation.
Figure 1. The structure of the proposed method.

Secondly, in the training stage, a training method based on WFS, SVR and HPSO are applied to obtain the optimized estimation model. The fitness value of each individual is evaluated by measuring the accuracy from the estimator of SVR. The 3-fold cross validation is used to evaluate the fitness of each individual to reduce the over-fitting.

Finally, in the estimation stage, the optimized estimation model obtained in the training stage is used for the new samples and the performance of the model is evaluated against the hold-out samples.

3. Research Design and Experiments

3.1. Application Data
In this study, the desharnais database, a commercial project data set mainly from Canada Software Park, is investigated. The database consists of 4 incomplete projects (No. 38, 44, 66, 75). Therefore, 77 complete cases are studied. In each case, there are 10 features and a corresponding software actual effort, which can refer to Table Ⅰ. In this dataset, 70 cases are selected as the modeling data and the remaining 7 cases as the hold-out data.

3.2. Research Design and System Development
In order to validate the performance of the proposed method, nine different models are experimented for the same data set. According to the type of feature selection and kernel function, the models are SVR-RBF (SVR with the RBF kernel), SVR-POL (SVR with the polynomial kernel), SVR-MK (SVR with the mixture of kernels), FS-SVR-RBF (Feature selection and SVR-RBF), FS-SVR-POL, FS-SVR-MK, WFS-SVR-RBF (WFS and SVR-RBF), WFS-SVR-POL and WFS-SVR-MK.

In order to compare the performance of different models, some parameters in HPSO are set uniform. The population size is set at 50; The maximal velocity of particles $V_{\text{max}}$ is set at 20; The acceleration constant $C_1 = 2.0$, $C_2 = 2.0$; The maximal iteration $k_{\text{max}} = 200$. The inertia weight $w$ is very important for the convergence behaviour of HPSO, which usually provides a balance between global and local exploration abilities and consequently results in a better optimum solution. The linearly decreasing weight algorithm is adopted to adjust $w$ to enable quick convergence: $w_{\text{max}} = 1.2$, $w_{\text{min}} = 0.4$. Considering the randomicity of HPSO algorithm, the experiments are carried out 20 times and the best results are chosen.
4. Experimental Results

4.1. The Results of HPSO-optimized Models

According to the above experiment design, different software cost estimation models with different parameters can be built. The optimized parameters of nine models are shown in Table II.

Table III shows the finally selected features of each model. As a result of WFS-SVR-MK, 10 optimized weights of each feature are obtained to maximize the estimation result for the modeling data set.

As shown in the Table III, for the same kernel function, the results of different feature selection strategies (FS or WFS) are basically consistent. However, WFS can more effectively distinguish the importance of different features on the software cost than FS.

With the same feature selection strategy, different features have the different importance on the software cost with different kernel functions.

From the perspective of the selected features, there are four features that always are selected (or have a greater importance) in either model. They are TeamExp, ManagerExp, Transactions and PointsNonAdjust, which have more important influence on software cost than other features.

| Number | Feature Name | Feature type |
|--------|--------------|--------------|
| 1      | TeamExp     | Numeric      |
| 2      | ManagerExp  | Numeric      |
| 3      | YearEnd     | Numeric      |
| 4      | Length      | Numeric      |
| 5      | Transactions| Numeric      |
| 6      | Entities    | Numeric      |
| 7      | PointsNonAdjust | Numeric |
| 8      | Adjustment  | Numeric      |
| 9      | PointsAdjust| Numeric      |
| 10     | Language    | {1, 2, 3}    |
| 11     | Effort      | Numeric      |

TABLE 1. Variables in desharnais data set

| Model     | Optimized parameters |
|-----------|----------------------|
| SVR-RBF   | $\gamma=10.82$, $\sigma=99.03$ |
| FS-SVR-RBF | $\gamma=62.91$, $\sigma=2.28$ |
| WF-SSVR-RBF | $\gamma=84.87$, $\sigma=92.80$ |
| SVR-POL   | $\gamma=63.51$, $d=0.55$ |
| FS-SVR-POL | $\gamma=2.07$, $d=0.13$ |
| WFS-SVR-POL | $\gamma=11.89$, $d=0.06$ |
| SVR-MK    | $\gamma=42.19$, $\sigma=97.07$, $d=1.00$, $\rho=0.78$ |
| FS-SVR-MK | $\gamma=50.70$, $\sigma=52.04$, $d=1.00$, $\rho=0.78$ |
| WF-SSVR-MK | $\gamma=98.95$, $\sigma=45.21$, $d=1.00$, $\rho=0.85$ |
TABLE 3. Feature selection results of different models.

| Feature       | RBF |   |   | POL |   |   | MK |   |   |
|---------------|-----|---|---|-----|---|---|----|---|---|
|               | SVR | FS | WFS | SVR | FS | WFS | SVR | FS | WFS |
| TeamExp       | 1   | 1 | 0.87 | 1   | 1 | 0.87 | 1   | 1 | 1.00 |
| ManagerExp    | 1   | 1 | 0.61 | 1   | 1 | 0.67 | 1   | 1 | 0.74 |
| YearEnd       | 1   | 1 | 0.59 | 1   | 0 | 0.29 | 1   | 0 | 0.41 |
| Length        | 1   | 0 | 0.05 | 1   | 0 | 0.21 | 1   | 1 | 0.53 |
| Transactions  | 1   | 1 | 0.83 | 1   | 1 | 1.00 | 1   | 1 | 0.95 |
| Entities      | 1   | 0 | 0.00 | 1   | 1 | 0.68 | 1   | 0 | 0.47 |
| PointsNonAdjust| 1   | 1 | 0.72 | 1   | 1 | 0.81 | 1   | 1 | 0.90 |
| Adjustment    | 1   | 0 | 0.49 | 1   | 1 | 0.75 | 1   | 1 | 0.52 |
| PointsAdjust  | 1   | 0 | 0.02 | 1   | 1 | 0.81 | 1   | 0 | 0.49 |
| Language      | 1   | 1 | 0.62 | 1   | 1 | 0.98 | 1   | 0 | 0.14 |

As can be seen from the above experiment, WFS can implement the more detailed feature selection, distinguish the influence of different features and create basic conditions for software cost control.

4.2. Comparison of the Estimation Performance

Table VI describes the estimation accuracy of each model which is produced when applying the parameters in Table II.

In Table VI, a clear comparison of various models for software cost estimation is reported via RMSE. Generally speaking, the results obtained from the table also indicate that the estimation performance of the proposed WFS-SVR-MK technique is better than the performance of the other modeling techniques.

Focusing on the RMSE indicator, our proposed WFS-SVR-MK model performs the best in all the cases, followed by the WFS-SVR-POL and WFS-SVR-RBF; the SVR-POL is the worst from a general point of view. Of models built with the same kernel function, the WFS-SVR can consistently outperform other models.

With different kernels, from the experiment analysis above, it is seen that the estimation performance is the best with the mixture of kernels. This indicates that the proposed modeling technique is an effective and promising method to software cost estimation.

In another comparative model, a linear regression model (LRM) is applied to our data set. For the desharnais data, the linear model is shown as

\[
\text{Effort} = -433.25 \times \text{TeamExp} + 408.8057 \times \text{ManagerExp} \\
+ 201.2701 \times \text{Length} + 4.2361 \times \text{Transactions} \\
+ 7.9056 \times \text{Entities} + 4.4594 \times \text{PointsAdjust} \\
+ 92.1389 \times \text{Adjustment} - 1777.4579 \times \text{Language} - 278.0786
\] (5)

Table V shows the performance of the LRM model. As we can see above, the estimation accuracy of the WFS-SVR-MK is higher than the performance of the LRM model.
### TABLE 4. Estimation accuracy of different models

| Model     | Modeling data set | Hold-out data set |
|-----------|------------------|-------------------|
| SVR-RBF   | 2.85*10^3        | 1.02*10^3        |
| FS-SVR-RBF| 2.35*10^3        | 925.24           |
| WFS-SVR-RBF| 2.72*10^3       | 673.19           |
| SVR-POL   | 2.93*10^3        | 1.27*10^3        |
| FS-SVR-POL| 2.95*10^3        | 947.48           |
| WFS-SVR-POL| 2.52*10^3       | 623.02           |
| SVR-MK    | 2.75*10^3        | 772.48           |
| FS-SVR-MK | 2.74*10^3        | 755.55           |
| WFS-SVR-MK| 2.45*10^3        | 611.48           |

### Table 5. Estimation accuracy of LRM and WFS-SVR-MK

| Model     | Modeling data set | Hold-out data set |
|-----------|------------------|-------------------|
| LRM       | 2.53*10^3        | 1.29*10^3        |
| WFS-SVR-MK| 2.45*10^3        | 611.48           |

### 5. Conclusion

In this study, a software cost estimation ensemble learning method based on WFS and SVR with the mixture of kernels is put forward, by which not only the relationship among the features and the cost can be obtained, but also the influence of the features on the cost can be distinguished so as to meet the requirement for project management and cost control. At the same time, the proposed joint optimization method can achieve the feature selection and the model optimization simultaneously.

In terms of empirical results, we find that across different models for the experiment cases of desharnais data on the basis of RMSE evaluation criteria, our proposed WFS-SVR-MK model performs the best, indicating that the proposed modeling technique can be used as a viable solution to software cost estimation.

Further, although this method is put forward for the software cost estimation, it is also applicable to the similar problems in other areas.

### Acknowledgments

This work is partially supported by the military science project of national social science fund (NO. 16GJ003-105) in China and the autonomous project of naval university of engineering (NO. 20161632).

### References

[1] Tayyab, Muhammad Raza, M. Usman, and W. Ahmad, A Machine Learning Based Model for Software Cost Estimation. Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016, 2018: pp. 402-414.

[2] Arora, Shaina, and N. Mishra, Software Cost Estimation Using Artificial Neural Network. Soft Computing: Theories and Applications, 2018.

[3] Aljahdali, Sultan, A. F. Sheta, and N. C. Debnath, "Estimating software effort and function point using regression, Support Vector Machine and Artificial Neural Networks models." Computer Systems and Applications IEEE, 2016: pp.1-8.

[4] Tian, Dongping, X. Zhao, and Z. Shi. Support Vector Machine with Mixture of Kernels for Image Classification. Intelligent Information Processing VI. Springer Berlin Heidelberg, 2012, pp. 68-76.

[5] Guyon, Isabelle, and A. Elisseeff, “An Introduction to Variable Feature Selection”, Journal of Machine Learning Research, vol. 3, Jan 2003, pp: 1157-1182, doi: 10.1162/153244303322753616

[6] M. Clerc, J. Kennedy, “The particle swarm explosion, stability, and convergence in a
multidimensional complex space”, IEEE Transactions on Evolutionary Computer, vol. 6, Feb. 2002, pp. 58-73, doi: 10.1109/4235.985692

[7] Couceiro, Micael, and P. Ghamisi, Particle Swarm Optimization, Fractional Order Darwinian Particle Swarm Optimization, Springer International Publishing, 2016: pp. 149-150.