



A study of project selection and feature weighting for analogy based software cost estimation

Y.F. Li^{*}, M. Xie, T.N. Goh

Department of Industrial and Systems Engineering, National University of Singapore, Singapore 119 260, Singapore

ARTICLE INFO

Article history:

Received 14 May 2007

Received in revised form 4 June 2008

Accepted 4 June 2008

Available online 17 June 2008

Keywords:

Software cost estimation

Analogy based estimation

Feature weighting

Project selection

Genetic algorithm

Artificial datasets

ABSTRACT

A number of software cost estimation methods have been presented in literature over the past decades. Analogy based estimation (ABE), which is essentially a case based reasoning (CBR) approach, is one of the most popular techniques. In order to improve the performance of ABE, many previous studies proposed effective approaches to optimize the weights of the project features (feature weighting) in its similarity function. However, ABE is still criticized for the low prediction accuracy, the large memory requirement, and the expensive computation cost. To alleviate these drawbacks, in this paper we propose the project selection technique for ABE (PSABE) which reduces the whole project base into a small subset that consist only of representative projects. Moreover, PSABE is combined with the feature weighting to form FWPS-ABE for a further improvement of ABE. The proposed methods are validated on four datasets (two real-world sets and two artificial sets) and compared with conventional ABE, feature weighted ABE (FWABE), and machine learning methods. The promising results indicate that project selection technique could significantly improve analogy based models for software cost estimation.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

Software cost estimation is critical for the success of software project management. It affects almost management activities including resource allocation, project bidding, and project planning (Pendharkar et al., 2005; Auer et al., 2006; Jorgensen and Shepperd, 2007). The importance of accurate estimation has led to extensive research efforts to software cost estimation methods. From a comprehensive review (Boehm et al., 2000), these methods could be classified into the following six categories: *parametric models* including COCOMO (Boehm, 1981; Huang et al., 2007), SLIM (Putnam and Myers, 1992), and SEER-SEM (Jensen, 1983); *expert judgment* including Delphi technique (Helmer, 1966) and work breakdown structure based methods (Tausworthe, 1980; Jorgensen, 2004); *learning oriented techniques* including machine learning methods (Heiat, 2002; Shin and Goel, 2000; Oliveira, 2006) and analogy based estimation (Shepperd and Schofield, 1997; Auer et al., 2006; Huang and Chiu, 2006); *regression based methods* including ordinary least square regression (Mendes et al., 2005; Costagliola et al., 2005) and robust regression (Miyazaki et al., 1994); *dynamics based models* (Madachy, 1994); *composite methods* (Chulani et al., 1999; MacDonell and Shepperd, 2003).

The analogy based estimation (ABE) which is essentially a case-based reasoning (CBR) approach (Shepperd and Schofield, 1997) was first proposed by Sternberg (1977). Due to its concep-

tual simplicity and empirical competitiveness, ABE has been extensively studied and applied (Shepperd and Schofield, 1997; Walkerden and Jeffery, 1999; Angelis and Stamelos, 2000; Mendes et al., 2003; Auer et al., 2006; Huang and Chiu, 2006; Chiu and Huang, 2007). The basic idea of ABE is simple: when provided a new project for estimation, compare it with historical projects to retrieve the most similar projects which are then used to predict the cost of new project. Generally, the ABE (or CBR) consists of four parts: a historical project dataset, a similarity function, a solution function and the associated retrieval rules (Kolodner, 1993). One of the associated central parts in ABE is the similarity function, which measures the level of similarity between two different projects. Since each project feature (or cost driver) has one position in the similarity function and therefore largely determines which historical projects should be retrieved for final prediction, there are several approaches focusing on searching the appropriate weight of each feature, such as Shepperd and Schofield (1997), Walkerden and Jeffery (1999), Angelis and Stamelos (2000), Mendes et al. (2003), Auer et al. (2006), Huang and Chiu (2006).

However, some difficulties are still confronted by ABE methods. Such as the non-normal characteristics (includes skewness, heteroscedasticity and excessive outliers) of the software engineering datasets (Pickard et al., 2001) and the increasing sizes of the datasets (Shepperd and Kadoda, 2001). The large and non-normal datasets always lead ABE methods to low prediction accuracy and high computational expense (Huang et al., 2002). To alleviate these drawbacks, many research works in the CBR literature (Lipowetzky,

^{*} Corresponding author. Tel.: +65 83442816.

E-mail address: liyanfu@nus.edu.sg (Y.F. Li).

1998; Babu and Murty, 2001; Huang et al., 2002) have been devoted to the case selection technique. The objective of case selection (CS) is to identify and remove redundant and noisy projects. By reducing the whole project base into a smaller subset that consist only of representative projects, the CS could save the computing time searching for most similar projects and produce quality prediction results. Moreover, the simultaneous optimization of feature weighting and case selection in CBR has been investigated in several studies (Kuncheva and Jain, 1999; Rozsypal and Kubat, 2003; Ahn et al., 2006) and significant improvements are reported from these studies.

From the discussion above, it is worthwhile to investigate case selection technique in the context of analogy based software cost estimation. In this study, we propose genetic algorithm for project selection for ABE (PSABE) and the simultaneous optimization of feature weights and project selection for ABE (FWPSABE). The proposed two techniques are compared against the feature weighting ABE (ABE), the conventional ABE and other popular cost estimation methods including ANN, RBF, SVM and CART. For the consistency of terminology, in rest of this paper we refer the case selection as project selection for ABE.

To compare different estimation methods, the empirical validation is very crucial. This has led to many studies use various real datasets to conduct comparisons of different cost estimation methods. However most published real datasets are relatively small (Mair et al., 2005) and the small real dataset could be problematic if we would like to show the significant differences between the estimation methods. Another drawback of the real world datasets is that the true properties of them may not be fully known. The artificially generated datasets (Pickard et al., 2001; Shepperd and Kadoda, 2001; Foss et al., 2003; Myrtveit et al., 2005) with known characteristics provide a feasible way to the above problems. Thus, we generate two artificial datasets and select two well known real-world datasets for controlled experiments.

The rest of this paper is organized as follows: Section 2 presents a brief overview on the conventional ABE method. In Section 3, the general framework of feature weight and project selection system for ABE is described. Section 4 presents the real world datasets and the experiments design. In Section 5, the results on two real world data sets are summarized and analyzed. In Section 6, two artificial datasets are generated, experiments are conducted on these two datasets, and results are summarized and analyzed. The final section presents the conclusion, and future works.

2. Overview on analogy based cost estimation

Analogy based method is a pure form of case based reasoning (CBR) with no expert used. Generally, ABE model comprises of four components: a historical dataset, a similarity function, a solution function and the associated retrieval rules (Kolodner, 1993). The ABE system process also consists of four stages:

1. Collect the past projects' information and prepare the historical dataset.
2. Select new project's relevant features such as function points (FP) and lines of source code (LOC), which are also collected for past projects.
3. Retrieval the past projects, estimate the similarities between new project and the past projects, and find the most similar past projects. The commonly used similarities are functions of weighted Euclidean distance and the weighted Manhattan distance.
4. Predict the cost of the new project from the chosen analogues by the solution function. Generally the un-weighted average is used as solution function.

The historical dataset which keeps all information of past projects is a key component in ABE system. However, it often contains noisy or redundant projects. By reducing the whole historical dataset into a smaller but more representative subset, the project selection technique positively affects the conventional ABE systems. First, it reduces the search space, thus more computing resources searching for most similar projects are saved. Secondly, it also produces quality predictions because it may eliminate noise in the historical dataset.

In the following sections, other components of ABE system including similar function, the number of most similar projects, and solution function are presented.

2.1. Similarity function

The *similarity function* measures the level of similarity between projects. Among different types of similarity functions, euclidean similarity (ES) and manhattan similarity (MS) based similarities are widely accepted (ES: Shepperd and Schofield, 1997. MS: Chiu and Huang, 2007). The Euclidean similarity is based on the Euclidean distance between two projects:

$$\text{Sim}(p, p') = 1 / \sqrt{\sum_{i=1}^n w_i \text{Dis}(f_i, f'_i) + \delta} \quad \delta = 0.0001$$

$$\text{Dis}(f_i, f'_i) = \begin{cases} (f_i - f'_i)^2, & \text{if } f_i \text{ and } f'_i \text{ are numeric or ordinal} \\ 1 & \text{if } f_i \text{ and } f'_i \text{ are nominal and } f_i = f'_i \\ 0 & \text{if } f_i \text{ and } f'_i \text{ are nominal and } f_i \neq f'_i \end{cases} \quad (1)$$

where p and p' denote the projects, f_i and f'_i denote the i th feature value of their corresponding projects, $w_i = [0, 1]$ is the weight of the i th feature, $\delta = 0.0001$ is a small constant to prevent the situation the denominator equals 0, and n is the total number of features.

The Manhattan similarity is based on the Manhattan distance which is the sum of the absolute distances for each pair of features

$$\text{Sim}(p, p') = 1 / \left[\sum_{i=1}^n w_i \text{Dis}(f_i, f'_i) + \delta \right] \quad \delta = 0.0001$$

$$\text{Dis}(f_i, f'_i) = \begin{cases} |f_i - f'_i| & \text{if } f_i \text{ and } f'_i \text{ are numeric or ordinal} \\ 1 & \text{if } f_i \text{ and } f'_i \text{ are nominal and } f_i = f'_i \\ 0 & \text{if } f_i \text{ and } f'_i \text{ are nominal and } f_i \neq f'_i \end{cases} \quad (2)$$

An important issue in the similarity functions is how to assign appropriate weight w_i to each feature pair, because each feature may have different relevance to the project cost. In the literature, several approaches were focusing on this topic: Shepperd and Schofield (1997) set each weight to be either 1 or 0 then apply a brute-force approach choosing optimal weights; Auer et al. (2006) extent Shepperd and Schofield's approach to the flexible extensive search method. Walkerden and Jeffery (1999) use human judgment to determine the feature weights; Angelis and Stamelos (2000) choose a value generated from statistical analysis as the feature weights. More recently, Huang and Chiu (2006) propose the genetic algorithm to optimize feature weights.

2.2. K number of similar projects

This parameter refers to the K number of most similar projects that is close to the project being estimated. Some studies suggested $K = 1$ (Walkerden and Jeffery, 1999; Auer et al., 2006; Chiu and Huang, 2007). However, we sets $K = \{1, 2, 3, 4, 5\}$ since many studies recommend K equals to two or three (Shepperd and Schofield, 1997; Mendes et al., 2003; Jorgensen et al., 2003; Huang and Chiu,

2006) and $K = \{1, 2, 3, 4, 5\}$ could cover most of the suggested numbers.

2.3. Solution functions

After K most similar projects are selected, the final prediction for the new project is determined by computing certain statistic based on the selected projects. The solution functions used in this study are: the closet analogy (most similar project) (Walkerden and Jeffery, 1999), the mean of most similar projects (Shepperd and Schofield, 1997), the median of most similar projects (Angelis and Stamelos, 2000) and the inverse distance weighted mean (Kadoda et al., 2000).

The mean is the average of the costs of K most similar projects, where $K > 1$. It is a classical measure of central tendency and treats all most similar projects as being equally influential on the cost estimates.

The median is the median of the costs of K most similar projects, where $K > 2$. It is another measure of central tendency and a more robust statistic when the number of most similar projects increases (Angelis and Stamelos, 2000).

The inverse distance weighted mean (Kadoda et al., 2000) allows more similar to have more influence than less similar ones. The formula for weighed mean is shown in (3):

$$\hat{C}_p = \frac{\sum_{k=1}^K \text{Sim}(p, p_k) C_{p_k}}{\sum_{k=1}^K \text{Sim}(p, p_k)} \quad (3)$$

where p denotes the new project being estimated, p_k represents the k th most similar project, $\text{Sim}(p, p_k)$ is the similarity between project p_k and p , C_{p_k} is the cost value of the k th most similar project p_k , and K is the total number of most similar projects.

3. Project selection and feature weighting

In this section, we construct the FWPSABE system (stands for feature weighting and project selection analogy based estimation) which can perform feature weighting analogy based estimation (FWABE) alone, project selection analogy based estimation (PSABE) alone, and the simultaneous optimization of feature weights and project selection (FWPSABE). Genetic algorithm (Holland, 1975) is selected as the optimization tool for the FWPSABE system, since it is a robust global optimization technique and has been applied to optimize the model parameters by several cost estimation papers (Dolado, 2000; Shukla, 2000; Dolado, 2001; Huang and Chiu, 2006). The framework and detailed description of FWPSABE system are presented in Section 3.2. In order to introduce the fitness function in GA operation, performance metrics for model accuracy are firstly presented in Section 3.1.

3.1. Performance metrics

To measure the accuracies of cost estimation methods, three widely used performance metrics are considered: Mean magnitude of relative error (MMRE), median magnitude of relative error (MdMRE) and PRED (0.25). The MMRE is defined as

$$\text{MMRE} = \frac{1}{n} \times \sum_{i=1}^n \text{MRE} \quad (4)$$

$$\text{MRE} = \left| \frac{C_i - \hat{C}_i}{C_i} \right|$$

where n denotes the number of projects, C_i denotes the actual cost of the i th project, and \hat{C}_i denotes the estimated cost of the i th project. Small MMRE value indicates low level of estimation error.

However, this metric is unbalanced and penalizes overestimation more than underestimation. The MdMRE is the median of all the MREs.

$$\text{MdMRE} = \text{median}(\text{MRE}) \quad (5)$$

It exhibits a similar pattern to MMRE but it is more likely to select the true model especially in the underestimation cases since it is less sensitive to extreme outliers (Foss et al., 2003). The PRED (0.25) is the percentage of predictions that fall within 25% of the actual cost.

$$\text{PRED}(q) = \frac{k}{n} \quad (6)$$

where n denotes the total number of projects and k represents the number of projects whose MRE is less than or equal to q . Normally, q is set to be 0.25. The PRED (0.25) identifies the cost estimations that are generally accurate, while MMRE is a biased and not always reliable as a performance metric. However, MMRE has been the de facto standard in the software cost estimation literature. Thus, the MMRE is selected for the fitness function in GA. More specifically, for each chromosome generated in GA, MMRE is computed across the training dataset. Then GA searches through the parameters space to minimize MMRE.

3.2. GA for project selection and feature weighting

The procedure of the project selection and feature weighting via genetic algorithm is presented in this section. The system consists of two stages: the first one is the training stage (as shown in Fig. 2) and the second is the testing stage (as shown in Fig. 3). In the training stage, a set of training projects are presented to the system, the ABE model is configured by the candidate parameters (feature weights and selection codes) to produce the cost predictions, and GA explores the parameters space to minimize the error (in terms of MMRE) of ABE on the training projects by the following steps:

i. Encoding.

To apply GA for optimization, the candidate parameters are coded as a binary code chromosome. As shown in Fig. 1, each individual chromosome consists of two parts. The first part is the codes for feature weights with the length of $14 \times n$, where n is the number of features. Since the feature weights in ABE model are decimal numbers, the binary codes have to be transformed into decimal values before entering ABE model. As many authors (Michalewicz, 1996; Ahn et al., 2006) suggested, the features weights is set as precisely as $1/10,000$. Thus, 14 binary bits are required to express this precision level because $8192 = 2^{13} < 10,000 \leq 2^{14} = 16,384$. After transformation, all decimal weight values are normalized into the interval $[0, 1]$ by the following formula (Michalewicz, 1996):

$$w_i = \frac{w'_i}{2^{14} - 1} = \frac{w'_i}{16,383} \quad (7)$$

where w'_i is the decimal conversion of i th feature's binary weight. For example, the binary code for feature 1 of the sample chromosome in Fig. 1 is $(10000000000001)_2$. Its decimal value is $(8193)_{10}$ and its normalized value is $8193/16,383 \approx 0.5001$. The second part of the codes is for project selection. The value of each bit is set to be either 0 or 1: 0 means the corresponding project in not selected and 1 means it is selected. The length of first part is m , and m is the total number of projects in the historical project base.

ii. Population generation.

After the encoding of the individual chromosome, the algorithm generates a population of chromosomes. For GA process, larger population size often results in higher chance for

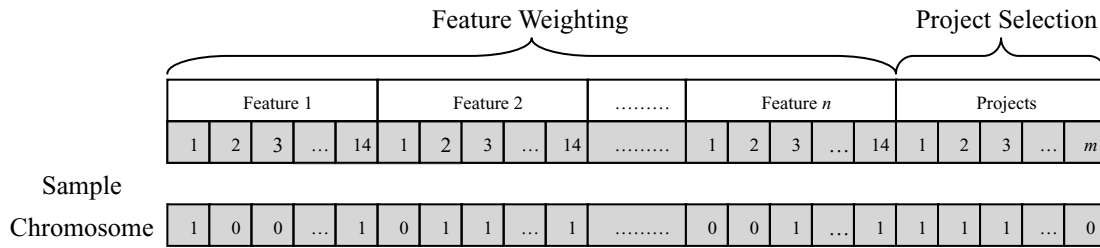


Fig. 1. Chromosome for FWPSABE.

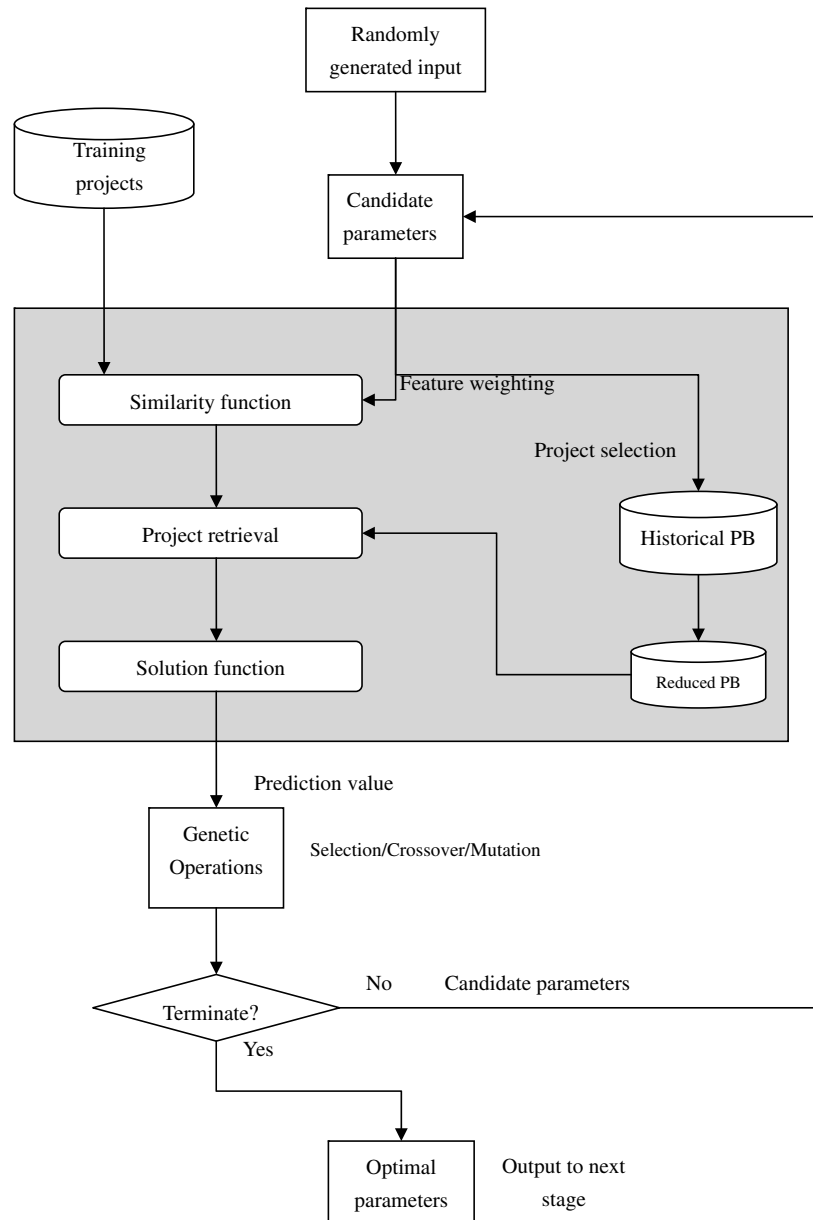


Fig. 2. The training stage of FWPSABE.

good solution (Doval et al., 1999). Since GA is computationally expensive, a trade-off between the convergence time and the population size must be made. In general, the minimum effective population size grows with problem size. Based on previous works (Huang and Chiu, 2006; Chiu and Huang, 2007), the size of the population is set to be $10V$ where V is the total number of input variables of GA search, which partially reflects the problem size.

iii. Fitness function.

Each individual chromosome is evaluated by the fitness function in GA. As mentioned in Section 3.1 MMRE is chosen for the fitness function and GA is designed to maximize the fitness function, as the sake of simplicity we set the fitness function as the reciprocal of MMRE.

$$f = \frac{1}{\text{MMRE}} \quad (8)$$

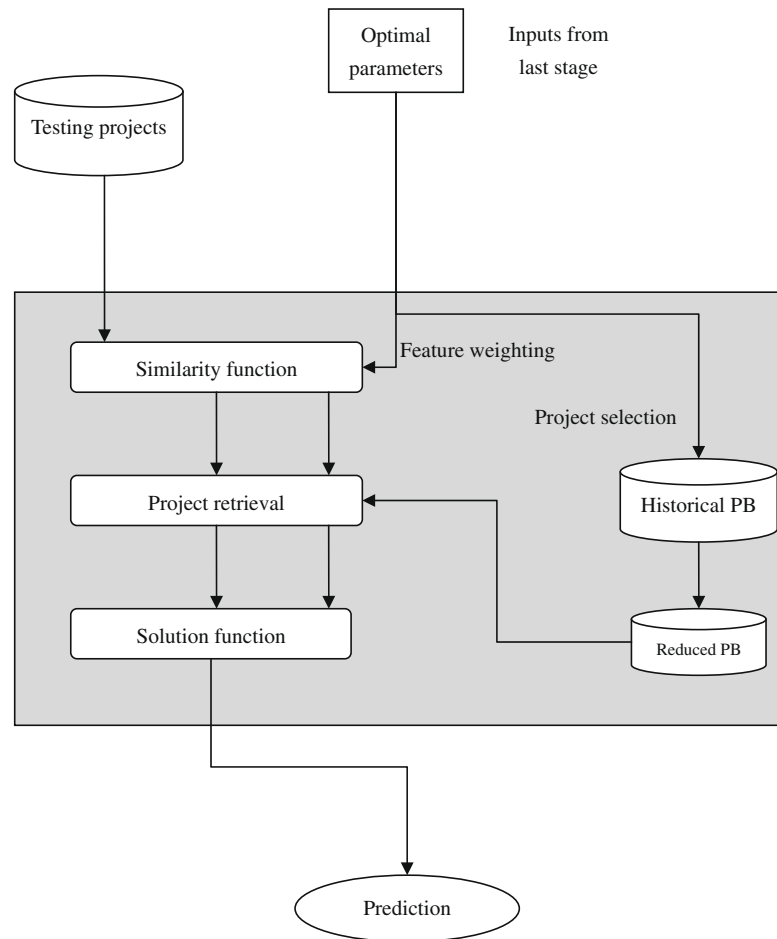


Fig. 3. The testing stage of FWPSABE.

iv. Fitness evaluation.

After transforming the binary chromosomes into the feature weighting and project selection parameters (see step i), the procedures of ABE are executed as follows:

Given one training project, the similarities between the training project and historical projects are computed by applying the feature weights into the similarity functions in (1) or (2). Simultaneously, the project selection part of the chromosome is used to generate the reduced historical project bases (reduced PBs). Then, ABE uses 1–5 most similar projects (1–NN to 5–NN) matching to search through the reduced PB for 1–5 most similar historical projects. Finally, the ABE model assigns a prediction value to the training project by adopting different solution functions. The error metric MMRE, PRED(0.25), and MdMRE are applied to evaluate the prediction performance on the training project set. Then, the reciprocal of MMRE is used as the fitness value for each parameter combination (or chromosome).

v. Selection.

The standard roulette wheel is used to select 10V chromosomes from the current population.

vi. Crossover.

The selected chromosomes were consecutively paired. The 1-point crossover operator with a probability of 0.7 was used to produce new chromosomes in each pair. The newly created chromosomes constituted a new population.

vii. Mutation.

Each bit of the chromosomes in the new population is chosen to change its value with a probability of 0.1, in a way that a bit '1' is changed to '0' and a bit '0' is changed to '1'.

viii. Elitist strategy.

Elitist strategy is used to overcome the defect of the slow convergence rate of GA. The elitist strategy retains good chromosomes and ensures they are not eliminated through the mechanism of crossover and mutation. Under this strategy, if the minimum fitness value of the new population is smaller than that of the old population, then the new chromosome with the minimum fitness value will be replaced with the old chromosome with the maximum fitness value.

ix. Stopping criteria.

There are few theoretical guidelines for determining when to terminate the genetic search. By following the previous works (Huang and Chiu, 2006; Chiu and Huang, 2007) on GA combining with ABE method, step v to step viii are repeated until the number of generations equal to or excess 1000V trials or the best fitness value does not change in the past 100V trails. After the stopping criteria are satisfied, the system moves to the second stage and the optimal parameters or chromosome are entered into the ABE model for testing.

In the above procedure, the population size, crossover rate, mutation rate and stopping condition are the controlling parameters of the GA search. However, there are few theories to guild the assign-

ments of these values (Ahn et al., 2006). Hence, we determine the value of these parameters in the light of previous studies that combines ABE and GAs. Most prior studies use 10V chromosomes as the population size, their crossover rate ranges from 0.5 to 0.7, and the mutation rate ranges from 0.06 to 0.1 (Ahn et al., 2006; Huang and Chiu, 2006; Chiu and Huang, 2007). However, because the search space for our GA is larger than these studies, we set the parameters to the higher bounds of those ranges. Thus, in this study the population size is 10V, the crossover rate is set at 0.7 and the mutation rate is 0.1.

The second stage is the testing stage. In this stage system receives the optimized parameters from the training stage to configure ABE model. The optimal ABE is then applied to the testing projects to evaluate the trained ABE.

4. Datasets and experiment designs

In this section, two real world software engineering datasets are firstly utilized for empirical evaluation of our methods. Additionally, all the cost estimation methods included in our experiments are described in Section 4.2 and the detailed experiments procedure is presented in Section 4.3.

4.1. Dataset preparation

The Albrecht dataset (Albrecht and Gaffney, 1983) includes 24 projects developed by using third generation languages. 18 of the projects were written in COBOL, 4 were written in PL1, and 2 were written in DMS languages. Six independent features of this dataset are 'input count', 'output count', 'query count', 'file count', 'function points', and 'source lines of code'. The dependent feature 'person hours' is recorded in 1000 h. The descriptive statistics of all the features shown in Table 1.

The Desharnais dataset (Desharnais, 1989) includes 81 projects and 11 features, 10 independent and one dependent. Since 4 out of 81 projects contain missing feature values, they have been excluded from the dataset. This process results in the 77 complete projects for our study. The ten independent features of this dataset are 'TeamExp', 'ManagerExp', 'YearEnd', 'Length', 'Transactions', 'Entities', 'PointsAdjust', 'Envergure', 'PointsNonAdjust', and 'Language'. The dependent feature 'person hours' is recorded in 1000 h. The descriptive statistics of all the features are shown in Table 2.

Before the experiments, all types of features are normalized into the interval [0, 1] in order to eliminate their different influences. In addition, the two real datasets (Albrecht and Desharnais) are randomly split into three nearly equal sized sub-sets for training and testing. The detail partitions of each dataset are provided in Table 3. The historical dataset is utilized by ABE model to retrieve similar past projects. The training set is treated as the targets for the optimization of feature weights and project subsets. The testing set is exclusively used to evaluate the optimized ABE models.

4.2. Cost estimation methods

Four ABE based models are included in our experiments. The first model is the conventional ABE. The second model is feature

Table 2
Descriptive statistics for Desharnais dataset

Feature	Minimum	Maximum	Mean	Standard deviation
TeamExp	0.00	4.00	2.30	1.33
ManagerExp	0.00	7.00	2.65	1.52
YearEnd	83.00	88.00	85.78	1.14
Length	1.00	36.00	11.30	6.79
Language	1.00	3.00	1.56	0.72
Transactions	9.00	886.00	177.47	146.08
Entities	7.00	387.00	120.55	86.11
PointsAdjust	73.00	1127.00	298.01	182.26
Envergure	5.00	52.00	27.45	10.53
PointsNonAdjust	62.00	1116.00	282.39	186.36
Person hours	0.55	23.94	4.83	4.19

Table 3
The partition of real datasets

Dataset	Sample size of Albrecht	Sample size of Desharnais
Historical	8	25
Training	8	25
Testing	8	27
Total	24	77

weighing analogy based estimation (FWABE) which assigns optimal feature weights via GA (Huang and Chiu, 2006). FWABE does not include project selection technique. The third model, project selection analogy based estimation (PSABE) uses GA to optimize the historical project subsets. PSABE excludes of feature weighting. The forth model is FWPSABE which uses GA for simultaneous optimization of features weighting and projects Selection. The latter two are the proposed by our study.

For a comprehensive evaluation of the proposed models, we compare them with other popular machine learning methods including artificial neural network ANN (Heiat, 2002), radial basis functions RBF (Shin and Goel, 2000), support vector machine regression SVR (Oliveira, 2006), and classification and regression trees CART (Pickard et al., 2001). The best variants of machine learning methods are obtained by training these methods and tuning their parameters on the historical datasets and training datasets presented in Section 3.1 respectively.

In ANN model, the number of hidden layers, the number of hidden nodes and the transfer functions are three predefined parameters and they have a major impact on the prediction performance (Martin et al., 1997). Among these parameters, one hidden layer is often recommended since multiple hidden layers may lead to an over parameterized ANN structure. Thus, one hidden layer is utilized in this study. The search spaces for the number of hidden neurons and hidden layer transfer functions are set to be {1, 3, 5, 7, 9, 10} and {linear, tan-sigmoid, log-sigmoid} respectively. During the training process, the ANN models with different parameter configurations are firstly trained on the historical dataset. Then, all ANN structures are implemented on the training set and the one producing the lowest MMRE value is selected for the comparisons against ABE models.

For RBF network, the forward selection strategy is utilized since forward selection has the advantages of flexible number of hidden nodes in advance, the tractable model selection criteria and the relatively low computational expense (Orr, 1996). In this case, the regularization parameter λ is introduced. To determine λ , the search space is defined as $\lambda = \{10^i | i = -10, -9, \dots, 0, \dots, 10\}$. Similar to ANN's training procedure, all RBFs with different λ values are trained on historical dataset and the one yielding the lowest MMRE on training data is selected for comparisons.

For SVR model, the common Gaussian function $K(x, y) = \exp\{-(x - y)^2 \delta^2\}$ is used as the kernel function. The predefined parameters δ , C and ε , are selected from the same search space

Table 1
Descriptive statistics for Albrecht dataset

Feature	Minimum	Maximum	Mean	Standard deviation
Input count	7.00	193.00	40.25	36.91
Output count	12.00	150.00	47.25	35.17
Query count	3.00	60.00	17.38	15.52
File count	0.00	75.00	16.88	19.34
Function points	3.00	318.00	61.08	63.68
SLOC	199.00	1902.00	647.63	488.00
Person hours	0.50	105.20	21.88	28.42

$\{10^i - i | i = -10, -9, \dots, 0, \dots, 10\}$. SVR models with all kinds of parameters combinations ($10 \times 10 \times 10 = 1000$ combinations) are trained on the historical dataset. The combination producing the minimal MMRE on the training set is chosen for comparisons.

To train CART model, we first use the historical set to fit the model and obtain a decision tree T . The tree T then is applied to the training set, and returns a vector of cost values computed for the training projects. The cost vector is then used to prune the tree T into a minimized size. The tree with optimal size is adopted for comparisons.

4.3. Experiment procedure

For the purpose of validations and comparisons, the following experiments procedures are conducted:

Firstly, the performances of FWPSABE are investigated by varying ABE parameters other than feature weights and project subsets. As mentioned in Section 2, ABE has three components exclusive of historical project base: similarity functions, K number of most similar projects, and the solution functions. In line with the common settings of these parameters, we define the search spaces for similarity function as {Euclidean distance, Manhattan distance}, K number of similar projects as {1, 2, 3, 4, 5}, and solution functions as {closest analogy, mean, median, inverse distance weighted mean} respectively. All kinds of parameter combinations are executed on both the training dataset and the testing. The best configuration on training dataset is selected out for the comparisons with other cost estimation methods.

Secondly, other ABE based methods are trained by the similar procedure described in the first step and the best variants on training set are selected as the candidate for comparisons. In addition, the optimizations of machines learning methods are conducted on the training dataset by searching through their parameter spaces.

Thirdly, the training and testing results of the best variants of all estimation methods are summarized and compared. The experiments results and analysis are presented in next section.

5. Experiment results

Table 4 presents FWPSABE's results on Albrecht dataset with different parameter configurations mentioned in Section 2. The results show that in general Euclidean distance achieves slightly more accurate performances than Manhattan distance on both the training and testing dataset. As to the solution function, there is no clear observation which function is most preferable. The choice of K value has some influence on the accuracies. The smaller errors mostly appear when $K = 3$ and $K = 4$. Among all configurations, the setting {Euclidean similarity, $K = 4$, and mean solution function} produces best results on training dataset and so it is selected for the comparisons with other cost estimation methods.

Table 5 summarizes the results of the best variants of all cost estimation methods on Albrecht dataset. It is observed that the FWPSABE achieves the best testing performance (0.30 for MMRE, 0.63 for PRED(0.25) and 0.27 for MdMRE) among all methods, and followed by PSABE, and FWABE. For a better illustration, the corresponding testing performs are presented in Fig. 4.

The results of FWPSABE with different configurations on Desharnais dataset are summarized in Table 6. The results show that on this dataset the choice of different similarity functions has little influence on both the training and testing performances. As to the solution functions, there is no clear conclusion which solution function is the best. The choice of K value has slight influence on the accuracies. The smaller errors are achieved by setting $K = 3$. In all configurations, the setting {Euclidean similarity, $K = 3$, and mean solution function} produces best results on training dataset

Table 4
Results of FWPSABE on Albrecht dataset

Similarity	K value	Solution	Training			Testing		
			MMRE	PRED(0.25)	MdMRE	MMRE	PRED(0.25)	MdMRE
Euclidean	$K = 1$	CA	0.39	0.25	0.35	0.40	0.38	0.45
		Mean	0.37	0.54	0.34	0.55	0.13	0.58
		IWM	0.40	0.58	0.34	0.57	0.32	0.42
	$K = 3$	Mean	0.56	0.38	0.34	0.41	0.33	0.39
		IWM	0.55	0.42	0.32	0.42	0.42	0.29
		Median	0.55	0.38	0.33	0.38	0.46	0.32
	$K = 4$	Mean	0.31	0.54	0.32	0.30	0.63	0.27
		IWM	0.35	0.52	0.33	0.44	0.50	0.32
		Median	0.40	0.54	0.37	0.37	0.58	0.28
	$K = 5$	Mean	0.58	0.42	0.32	0.39	0.38	0.45
		IWM	0.54	0.33	0.38	0.51	0.25	0.42
		Median	0.51	0.38	0.45	0.42	0.25	0.45
Manhattan	$K = 1$	CA	0.50	0.25	0.41	0.45	0.25	0.53
		Mean	0.56	0.38	0.42	0.43	0.13	0.44
		IWM	0.55	0.40	0.44	0.59	0.28	0.45
	$K = 3$	Mean	0.55	0.52	0.45	0.39	0.38	0.35
		IWM	0.51	0.44	0.42	0.42	0.25	0.40
		Median	0.53	0.32	0.43	0.51	0.33	0.32
	$K = 4$	Mean	0.53	0.38	0.32	0.41	0.54	0.45
		IWM	0.51	0.36	0.35	0.51	0.50	0.42
		Median	0.50	0.34	0.44	0.53	0.32	0.46
	$K = 5$	Mean	0.54	0.34	0.42	0.59	0.13	0.58
		IWM	0.52	0.36	0.48	0.52	0.23	0.48
		Median	0.53	0.34	0.45	0.51	0.13	0.46

Table 5
The results and comparisons on Albrecht dataset

Models	MMRE		PRED(0.25)		MdMRE	
	Training	Testing	Training	Testing	Training	Testing
ABE	0.38	0.49	0.50	0.13	0.36	0.49
FWABE	0.48	0.42	0.38	0.25	0.34	0.46
PSABE	0.40	0.39	0.25	0.38	0.35	0.45
FWPSABE	0.31	0.30	0.54	0.63	0.32	0.27
SVR	0.46	0.45	0.50	0.25	0.22	0.43
ANN	0.39	0.49	0.38	0.25	0.35	0.51
RBF	0.79	0.49	0.50	0.25	0.25	0.39
CART	4.77	1.70	0.13	0.13	0.58	0.89

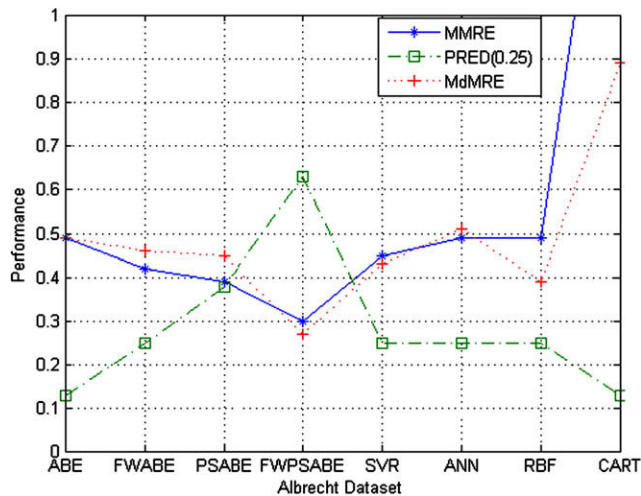


Fig. 4. The testing results on Albrecht dataset.

Table 6
Results of FWPSABE on Desharnais dataset

Similarity	K value	Solution	Training			Testing		
			MMRE	PRED(0.25)	MdMRE	MMRE	PRED(0.25)	MdMRE
Euclidean	K = 1	CA	0.54	0.24	0.47	0.52	0.27	0.51
		Mean	0.57	0.26	0.45	0.62	0.37	0.50
		IWM	0.55	0.24	0.44	0.97	0.42	0.67
	K = 3	Mean	0.40	0.36	0.36	0.32	0.44	0.29
		IWM	0.55	0.36	0.38	0.42	0.42	0.36
		Median	0.56	0.34	0.36	0.38	0.42	0.34
	K = 4	Mean	0.59	0.16	0.39	0.40	0.26	0.39
		IWM	0.55	0.36	0.41	0.64	0.17	0.46
		Median	0.53	0.34	0.37	0.57	0.38	0.42
	K = 5	Mean	0.55	0.24	0.56	0.43	0.28	0.48
		IWM	0.54	0.26	0.56	0.52	0.25	0.42
		Median	0.59	0.29	0.55	0.64	0.27	0.53
Manhattan	K = 1	CA	0.39	0.28	0.37	0.67	0.30	0.44
		Mean	0.54	0.32	0.48	0.47	0.25	0.51
		IWM	0.55	0.40	0.34	0.52	0.25	0.53
	K = 3	Mean	0.45	0.28	0.49	0.46	0.22	0.38
		IWM	0.56	0.24	0.43	0.41	0.42	0.37
		Median	0.58	0.20	0.46	0.51	0.20	0.45
	K = 4	Mean	0.51	0.24	0.48	0.57	0.33	0.51
		IWM	0.53	0.26	0.55	0.58	0.27	0.52
		Median	0.60	0.30	0.53	0.54	0.28	0.52
	K = 5	Mean	0.54	0.24	0.50	0.52	0.26	0.48
		IWM	0.56	0.34	0.58	0.64	0.18	0.59
		Median	0.63	0.36	0.55	0.55	0.23	0.52

and so it is selected for the comparisons against other cost estimation methods.

Table 7 presents the results of the best variants of all cost estimation methods on Desharnais dataset. It is shown that the FWPSABE achieves the best testing performance (0.32 for MMRE, 0.44 for PRED(0.25) and 0.29 for MdMRE), and followed by SVR and PSABE. Fig. 5 provides an illustrative version of the testing results in Table 7.

6. Artificial datasets and experiments results

To compare different cost estimation methods, the need for empirical validation is very crucial. This has led to the collection of various real world datasets for experiments. Mair et al. (2005) conducted an extensive survey of the real datasets for cost estimation from 1980 onwards. As reported, most published real world datasets are relatively small for the tests of significance and the true properties of them may not be fully known. For example, it might be difficult to distinguish different types of distribution in the presence of extreme outliers in a small dataset (Shepperd and Kadoda, 2001).

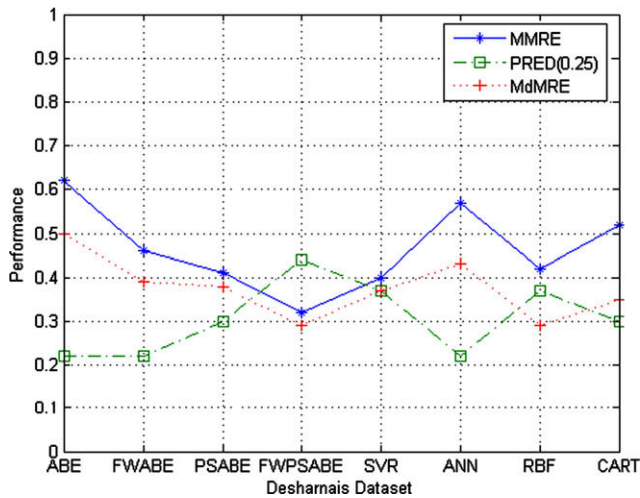
Artificially generated datasets provide a feasible solution to the above two difficulties. Firstly, the researchers can generate reasonable amount of artificial data to investigate the significant differences among the competing techniques. Secondly, it provides the control over the characteristics of the artificial dataset. Especially, researchers could design a systematic way to vary the properties for their research purposes (Pickard et al., 1999). In order to evaluate the proposed methods in a more controlled way, we generate two artificial datasets for further experiments.

From each of the two real datasets, we extract a set of characteristics describing its property, or more specifically its non-normality. The non-normality considered in our study includes

Table 7

The results and comparisons on Desharnais dataset

Models	MMRE		PRED(0.25)		MdMRE	
	Training	Testing	Training	Testing	Training	Testing
ABE	0.62	0.62	0.28	0.22	0.51	0.50
FWABE	0.51	0.46	0.12	0.22	0.48	0.39
PSABE	0.39	0.41	0.28	0.30	0.37	0.38
FWPSABE	0.40	0.32	0.36	0.44	0.36	0.29
SVR	0.42	0.40	0.28	0.37	0.45	0.37
ANN	0.45	0.57	0.36	0.22	0.44	0.43
RBF	0.57	0.42	0.24	0.37	0.49	0.29
CART	0.97	0.52	0.28	0.30	0.50	0.35

**Fig. 5.** The testing results on Desharnais dataset.

skewness, variance instability, and excessive outliers (Pickard et al., 2001). We then by using the two sets of characteristics generate two sets of artificial data. Section 6.1 presents the details for artificial datasets generation.

6.1. Generation of the artificial datasets

To explore the non-normal characteristics of the real world dataset, the “cost-size” scatter plot for Albrecht dataset is drawn as Fig. 6. The scatter plot indicates the slight skewness, moderate outliers, and slight variance instability of the Albrecht dataset.

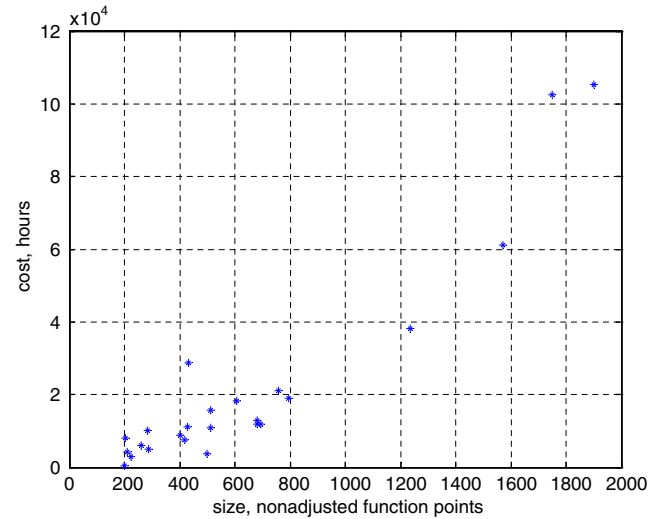
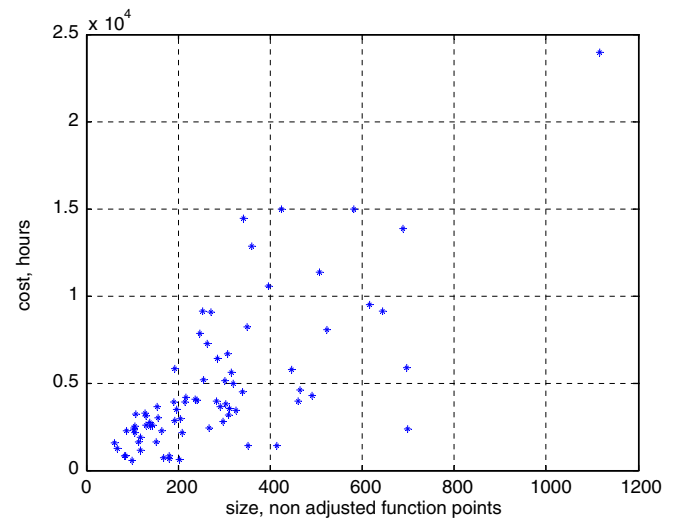
The “cost-size” scatter plot of the Desharnais dataset is illustrated in Fig. 7 which shows weak skewness, extreme outliers, and highly variance instability of this dataset.

From the analysis above, software dataset often exhibits a mixture of several non-normal characteristics such as skewness, variance instability, and excessive outliers (Pickard et al., 2001). These characteristics do not always appear in the same degree. In some cases they are moderately non-normal such as the Albrecht dataset, while in other cases they are severely non-normal such as the Desharnais dataset. Without loss of generality, we adopted Pickard’s way of modeling non-normality in this work. Other types of techniques for artificial dataset generation are also available in recent literature. For more details, readers can refer to Shepperd and Kadoda (2001), Foss et al. (2003) and Myrtevit et al. (2005).

By Pickard’s way, we simulate the combination of non-normal characteristics: skewness, unstable variance and outliers in (7):

$$y = 1000 + 6x_1sk + 3x_2sk + 2x_3sk + e_{het} \quad (9)$$

The independent variables (x_1sk , x_2sk , x_3sk) are generated by Gamma distributed random variables x'_1 , x'_2 , and x'_3 with mean 4

**Fig. 6.** Cost versus size of Albrecht dataset.**Fig. 7.** Cost versus size of Desharnais dataset.

and variance 8. And the skewness is explicit by the Gamma distributions. In order to vary the scale of the independent variables, we then multiply the x'_1 by 10 to create variable x_1sk , the x'_2 by 3 to create variable x_2sk and x'_3 by 20 to create the variable x_3sk .

The last term e_{het} in the formula simulates a special form of unstable variance: heteroscedasticity. The heteroscedasticity occurs where the error term is related to one of the variables in the model and either increase or decreases depending on the value of the independent variable. The error term e_{het} is related to x_1sk by the relationship $e_{het} = 0.1 \times e \times x_1sk$ for the moderate heteroscedasticity, and $e_{het} = 6 \times e \times x_1sk$ for the severe heteroscedasticity (Pickard et al., 2001).

The outliers are generated by multiplying or dividing the dependent variable y by a constant. We select 1% of the data to be the outliers. Half of the outliers are obtained by multiplying while half of them are got by dividing. For the moderate outliers, we set the constant value as 2, while for the severe outliers, 6 is chosen to be the constant.

The combination of moderate heteroscedasticity and moderate outliers is used to generate the moderate non-normality dataset (Fig. 8). The joint of severe heteroscedasticity and severe outliers is used to obtain the severe non-normality dataset (Fig. 9).

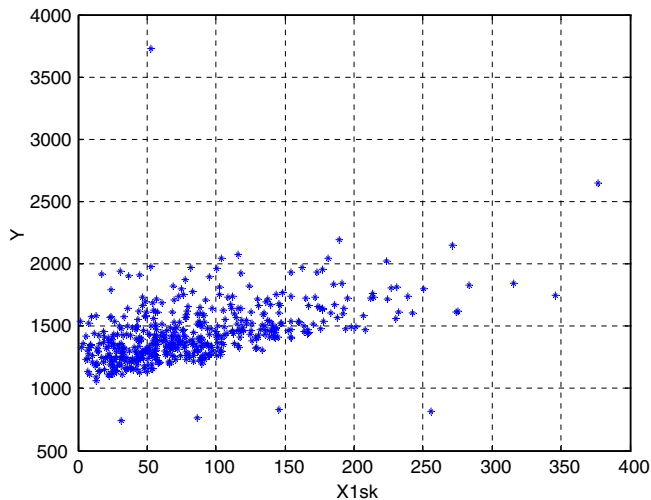


Fig. 8. Y versus x_{1sk} of moderate non-normality dataset.

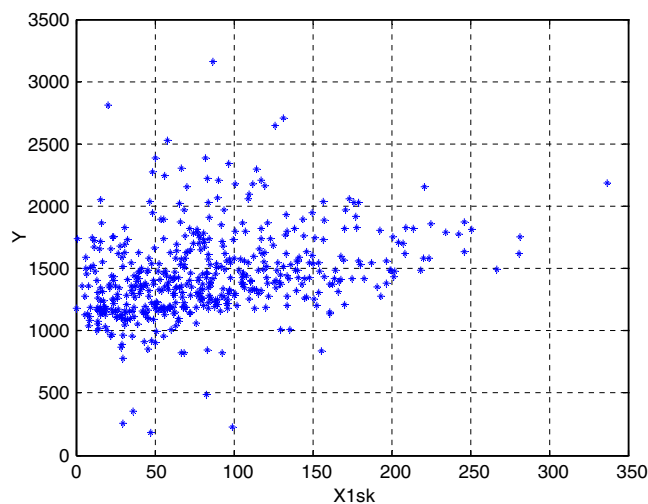


Fig. 9. Y versus x_{1sk} of severe non-normality dataset.

6.2. Experiments results on artificial datasets

By using the equation mentioned in Section 6.1, we generate two artificial datasets, each with 500 projects. For a better assessment of accuracy, we make the data for testing much larger by dividing the artificial datasets into: historical set with 50 projects, training set with 50 projects, and the testing set with 400 projects (see Table 8).

We apply all the methods onto the two artificial datasets by following the same procedure presented in Section 4.3. The results and comparisons are summarized as following.

The results on artificial moderate non-normality dataset are in Table 9. It is shown that FWPSABE achieves the best performances

Table 8
The partition of artificial datasets

Dataset	Sample size of artificial moderate non-normality data	Sample size of artificial severe non-normality data
Historical	50	50
Training	50	50
Testing	400	400
Total	500	500

Table 9
The results and comparisons on artificial moderate non-normality dataset

Models	MMRE		PRED(0.25)		MdMRE	
	Training	Testing	Training	Testing	Training	Testing
ABE	0.068	0.116	0.98	0.94	0.048	0.093
FWABE	0.090	0.110	1.00	0.98	0.081	0.098
PSABE	0.057	0.086	1.00	0.98	0.043	0.068
FWPSABE	0.055	0.079	1.00	0.98	0.044	0.060
SVR	0.069	0.095	0.98	0.98	0.055	0.077
ANN	0.065	0.088	1.00	0.99	0.061	0.077
RBF	0.099	0.115	0.94	0.93	0.075	0.092
CART	0.099	0.109	0.98	0.95	0.074	0.090

in MMRE at 0.079 and MdMRE at 0.06 and the second best value 0.98 for PRED(0.25), while ANN gets the highest PRED(0.25) value at 0.99. Compare the prediction error curves in Fig. 4 for Albrecht dataset to the error curves in Fig. 10 for moderate non-normality set, it is observed that all the methods achieve much better performance on the artificial dataset and the differences among the candidate methods are much smaller on the artificial dataset. These findings imply that estimation methods in our study may converge to good prediction results on the moderately non-normal dataset with large size and FWPSABE is slightly better than other methods as it eliminate the noise in the historical dataset.

Table 10 shows the results on artificial severe non-normality dataset. FWPSABE achieves the best performances in MMRE at 0.16 and MdMRE at 0.11 and the second best value 0.80 for PRED(0.25), while CART obtains the highest PRED(0.25) value at 0.81. Compare Figs. 10, and 11, it is shown that the all methods obtain poorer performances on severe non-normal dataset. This

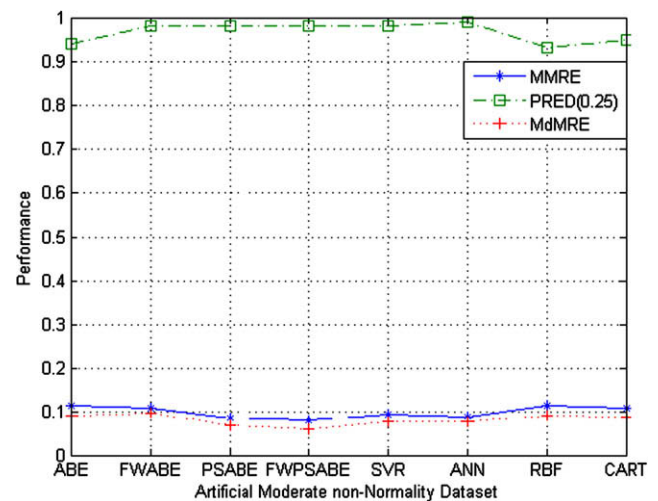


Fig. 10. The testing results on artificial moderate non-normality dataset.

Table 10
The results and comparisons on artificial severe non-normality dataset

Models	MMRE		PRED(0.25)		MdMRE	
	Training	Testing	Training	Testing	Training	Testing
ABE	0.32	0.20	0.68	0.73	0.18	0.14
FWABE	0.34	0.19	0.72	0.77	0.14	0.13
PSABE	0.31	0.18	0.70	0.75	0.11	0.12
FWPSABE	0.30	0.15	0.74	0.80	0.14	0.10
SVR	0.34	0.18	0.62	0.76	0.19	0.12
ANN	0.34	0.17	0.70	0.79	0.16	0.12
RBF	0.37	0.18	0.66	0.80	0.18	0.13
CART	0.38	0.18	0.72	0.81	0.16	0.14

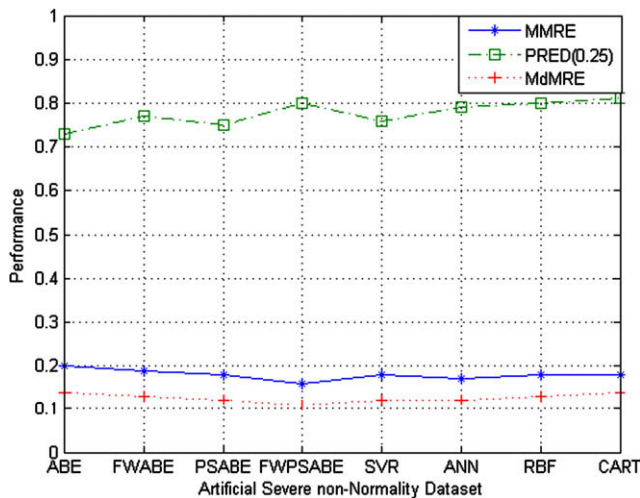


Fig. 11. The testing results on artificial severe non-normality dataset.

observation indicates that high degree of non-normality has negative impacts on the performance of estimation methods in our study.

7. Conclusions and future works

In this study, we introduce the project selection technique to refine the historical project database in ABE model. In addition, the simultaneous optimization of feature weights and project selection (FWPSABE) is proposed to further improve the performance of ABE. To evaluate our methods, we apply them on two real-world dataset and two artificial datasets. The error indicators for methods evaluations are MMRE, PRED(0.25), and MdMRE. The promising results of the proposed FWPSABE system indicate that it can significantly improve the ABE model and enhance ABE as a successful method among software cost estimation techniques.

One major conclusion of this paper is that FWPSABE system may produce more accurate predictions than other advanced machine learning techniques for software cost estimation. In the literature, ABE is already regarded as a benchmarking method for cost estimation (Shepperd and Schofield, 1997). First, it is not complex for implementation and it is more transparent to the users than most machine learning methods. Moreover, ABE's prediction can update in real time; once a project is completed, its information can be easily inserted into the historical project database. However, many studies reported that in practice ABE has been hindered by the low prediction accuracy. According to the results in this study, FWPSABE may be useful in practical situations because it has the advantages of ABE and the ability to produce more accurate cost estimation results.

However, there are still some limitations of study. For example, the two real-world datasets in our experiments are quite old though they have been frequently used by many recent studies. Experiments on recent and large size datasets such as ISBSG database are essential for more rigorous evaluations on our methods. In addition, our methods are only validated on the projects developed by the traditional waterfall based approach. Software projects developed by new type of approaches such as agile methods have additional features indicating the characteristics of their development approaches. The accuracies of FWPSABE for projects under newly development types should be further investigated. Moreover, ABE based methods are intolerant of missing features. If information of some historical projects is incomplete, then the data imputation techniques should be taken

to process the miss values the FWPSABE system starts. Furthermore, only MMRE is used for optimization objective function, and there is no guarantee that other quality metrics such as PRED(0.25) and MdMRE can be optimized while optimizing the single objective MMRE. Multi-objective optimization techniques can be investigated in future works.

References

- Ahn, H., Kim, K., Han, I., 2006. Hybrid genetic algorithms and case-based reasoning systems for customer classification. *Expert Systems* 23 (3).
- Albrecht, A.J., Gaffney, J., 1983. Software function, source lines of code, and development effort prediction. *IEEE Transactions on Software Engineering* 9 (6), 639–648.
- Angelis, L., Stamelos, I., 2000. A simulation tool for efficient analogy based cost estimation. *Empirical Software Engineering* 5, 35–68.
- Auer, M., Trendowicz, A., Graser, B., Haunschmid, E., Biffl, S., 2006. Optimal project feature weights in analogy-based cost estimation: improvement and limitations. *IEEE Transactions on Software Engineering* 32 (2), 83–92.
- Babu, T.R., Murty, M.N., 2001. Comparison of genetic algorithm based prototype selection schemes. *Pattern Recognition* 34, 523–525.
- Boehm, B., 1981. *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs, NJ.
- Boehm, B., Abts, C., Chulani, S., 2000. Software development cost estimation approaches – a survey. *Annals of Software Engineering* 10, 177–205.
- Chiu, N.H., Huang, S.J., 2007. The adjusted analogy-based software effort estimation based on similarity distances. *Journal of Systems and Software* 80 (4), 628–640.
- Chulani, S., Boehm, B., Steece, B., 1999. Bayesian analysis of empirical software engineering cost models. *IEEE Transactions on Software Engineering* 25 (4), 573–583.
- Costagliola, G., Ferrucci, F., Tortora, G., Vitiello, G., 2005. Class point: an approach for the size estimation of object-oriented systems. *IEEE Transactions on Software Engineering* 31 (1), 52–74.
- Desharnais, J.M., 1989. *Analyse statistique de la productivité des projets informatiques a partie de la technique des point des fonction*, University of Montreal, Masters thesis.
- Dolado, J.J., 2000. A validation of the component-based method for software size estimation. *IEEE Transactions on Software Engineering* 26 (10), 1006–1021.
- Dolado, J.J., 2001. On the problem of the software cost function. *Information and Software Technology* 43, 61–72.
- Doval, D., Mancoridis, S., Mitchell, B.S., 1999. Automatic clustering of software systems using a genetic algorithm. *Proceedings of the 9th International Workshop Software Technology and Engineering Practice*, 73–81.
- Foss, T., Stensrud, E., Kitchenham, B., Myrtevit, I., 2003. A simulation study of the model evaluation criterion MMRE. *IEEE Transactions on Software Engineering* 29 (11).
- Heiat, A., 2002. Comparison of artificial neural network and regression models for estimating software development effort. *Information and Software Technology* 44, 911–922.
- Helmer, O., 1966. *Social Technology*. Basic Books, NY.
- Holland, J., 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, USA.
- Huang, S.J., Chiu, N.H., 2006. Optimization of analogy weights by genetic algorithm for software effort estimation. *Information and Software Technology* 48, 1034–1045.
- Huang, Y.S., Chiang, C.C., Shieh, J.W., Grimson, E., 2002. Prototype optimization for nearest-neighbor classification. *Pattern Recognition* 35, 1237–1245.
- Huang, X.S., Ho, D., Ren, J., Capretz, L.F., 2007. Improving the COCOMO model using a neuro-fuzzy approach. *Applied Soft Computing Journal* 7 (1), 29–40.
- Jensen, R., 1983. An improved macrolevel software development resource estimation model. In: *Proceedings of 5th Conference of International S Parametric Analysts*, pp. 88–92.
- Jorgensen, M., 2004. Top-down and bottom-up expert estimation of software development effort. *Information and Software Technology* 46, 3–16.
- Jorgensen, M., Indahl, U., Sjoberg, D., 2003. Software effort estimation by analogy and 'regression toward the mean'. *Journal of Systems and Software* 68 (3), 253–262.
- Jorgensen, M., Shepperd, M., 2007. A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering* 33 (1), 33–53.
- Kadoda, G., Cartwright, M., Chen, L., Shepperd, M., 2000. Experiences using case-based reasoning to predict software project effort. In: *Proceedings EASE 2000 conferences – 4th International Conference on Empirical Assessment and Evaluation in Software Engineering*. Staffordshire, U.K.
- Kolodner, J.L., 1993. *Case-Based Reasoning*. Morgan Kaufmann Publishers Inc.
- Kuncheva, L.I., Jain, L.C., 1999. Nearest neighbor classifier: simultaneous editing and feature selection. *Pattern Recognition Letters* 20, 1149–1156.
- Lipowezky, U., 1998. Selection of the optimal prototype subset for 1-NN classification. *Pattern Recognition Letters* 19, 907–918.
- MacDonell, S.G., Shepperd, M.J., 2003. Combining techniques to optimize effort predictions in software project management. *Journal of Systems and Software* 66, 91–98.

- Madachy, R., 1994. A Software Project Dynamics Model for Process Cost, Schedule and Risk Assessment, Ph.D. Dissertation, University of Southern California.
- Mair, C., Shepperd, M., Jorgensen, M., 2005. An analysis of data sets used to train and validate cost prediction systems. PROMISE'05.
- Hagan, Martin T., Demuth, Howard B., Beale, Mark H., 1997. Neural Network Design. PWS Publishing Co., Boston, MA.
- Mendes, E., Watson, I., Triggs, C., Mosley, N., Counsell, S., 2003. A comparative study of cost estimation models for Web hypermedia applications. *Empirical Software Engineering* 8, 163–196.
- Mendes, E., Mosley, N., Counsell, S., 2005. Investigating Web size metrics for early Web cost estimation. *Journal of Systems and Software* 77 (2), 157–172.
- Michalewicz, Z., 1996. Genetic Algorithms + Data Structures = Evolution Programs, third ed. Springer, Berlin.
- Miyazaki, Y., Terakado, K., Ozaki, K., Nozaki, H., 1994. Robust regression for developing software estimation models. *Journal of Systems and Software* 27, 3–16.
- Myrtveit, I., Stensrud, E., Shepperd, M., 2005. Reliability and validity in comparative studies of software prediction models. *IEEE Transactions on Software Engineering* 31 (5), 380–391.
- Oliveira, A.L.I., 2006. Estimation of software project effort with support vector regression. *Neurocomputing* 69, 1749–1753.
- Orr, M.J.L. 1996. Introduction to Radial Basis Function Network. Technical Reports, Centre for Cognitive Science, University of Edinburgh, 2, Buccleuch Place, Edinburgh, Scotland.
- Pendharkar, P.C., Subramanian, G.H., Rodger, J.A., 2005. A probabilistic model for predicting software development effort. *IEEE Transactions on Software Engineering* 31 (7), 615–624.
- Pickard, L., Kitchenham, B., Linkman, S., 1999. An investigation analysis techniques for software datasets. In: *Proceeding of Sixth IEEE International Software Metrics Symposium*.
- Pickard, L., Kitchenham, B., Linkman, S., 2001. Using simulated data sets to compare data analysis techniques used for software cost modeling. *IEE Proceeding of Software* 148 (6), 165–174.
- Putnam, L., Myers, W., 1992. Measures for Excellence. Yourdon Press Computing Series.
- Rozsypal, A., Kubat, M., 2003. Selecting representative examples and attributes by a genetic algorithm. *Intelligent Data Analysis* 7, 291–304.
- Shepperd, M., Kadoda, G., 2001. Comparing software prediction techniques using simulation. *IEEE Transactions on Software Engineering* 27 (11), 1014–1022.
- Shepperd, M., Schofield, C., 1997. Estimating software project effort using analogies. *IEEE Transactions on Software Engineering* 23 (12), 733–743.
- Shin, M., Goel, A.L., 2000. Empirical data modeling in software engineering using radial basis functions. *IEEE Transactions on Software Engineering* 26 (6), 567–576.
- Shukla, K.K., 2000. Neuro-genetic prediction of software development effort. *Information and Software Technology* 42, 701–713.
- Sternberg, R., 1977. Component processes in analogical reasoning. *Psychological Review* 84 (4), 353–378.
- Tausworthe, R.C., 1980. The work breakdown structure in software project management. *Journal of Systems and Software* 1 (3), 181–186.
- Walkerden, F., Jeffery, R., 1999. An empirical study of analogy-based software effort estimation. *Empirical Software Engineering* 4, 135–158.