

## A study of the non-linear adjustment for analogy based software cost estimation

Y. F. Li · M. Xie · T. N. Goh

Published online: 12 February 2009

© Springer Science + Business Media, LLC 2009

**Editor:** Emilia Mendes

**Abstract** Cost estimation is one of the most important but most difficult tasks in software project management. Many methods have been proposed for software cost estimation. Analogy Based Estimation (ABE), which is essentially a case-based reasoning (CBR) approach, is one popular technique. To improve the accuracy of ABE method, several studies have been focusing on the adjustments to the original solutions. However, most published adjustment mechanisms are based on linear forms and are restricted to numerical type of project features. On the other hand, software project datasets often exhibit non-normal characteristics with large proportions of categorical features. To explore the possibilities for a better adjustment mechanism, this paper proposes Artificial Neural Network (ANN) for Non-linear adjustment to ABE (NABE) with the learning ability to approximate complex relationships and incorporating the categorical features. The proposed NABE is validated on four real world datasets and compared against the linear adjusted ABEs, CART, ANN and SWR. Subsequently, eight artificial datasets are generated for a systematic investigation on the relationship between model accuracies and dataset properties. The comparisons and analysis show that non-linear adjustment could generally extend ABE's flexibility on complex datasets with large number of categorical features and improve the accuracies of adjustment techniques.

**Keywords** Software cost estimation · Analogy based estimation · Case based reasoning · Non-linear adjustment · Artificial neural network · Artificial datasets

---

Y. F. Li (✉) · M. Xie · T. N. Goh  
Department of Industrial and Systems Engineering, National University of Singapore,  
Singapore 119 260, Singapore  
e-mail: liyanfu@nus.edu.sg

M. Xie  
e-mail: isexiem@nus.edu.sg

T. N. Goh  
e-mail: isegoh@nus.edu.sg

## Abbreviations

<i>MRE</i>	Magnitude of Relative Error
<i>MMRE</i>	Mean Magnitude of Relative Error
<i>PRED(0.25)</i>	PREDiction at level 0.25
<i>MdMRE</i>	Median Magnitude of Relative Error
<i>ABE</i>	Analogy Based Estimation
<i>NABE</i>	Non-linear adjusted ABE
<i>GABE</i>	GA optimized linear adjusted ABE
<i>LABE</i>	Linear adjusted ABE
<i>RABE</i>	Regression Toward the Mean adjusted ABE
<i>ANN</i>	Artificial Neural Network
<i>CART</i>	Classification and Regression Trees
<i>OLS</i>	Ordinary Least Square regression
<i>SWR</i>	Stepwise Regression

## 1 Introduction

Software cost estimation is a continuous activity which often starts at the first stage of the software life cycle and continues throughout the life time. Since software cost estimation affects most software development activities, it has become a critical practice in software project management. The importance of accurate cost estimation has led to extensive research efforts onto estimation methods in the past decades. These methods can be classified into three basic types (Angelis and Stamelos 2000): expert judgment (Jorgensen 2004, 2005, 2007), algorithmic estimation (Jun and Lee 2001; Heiat 2002; Pendharkar et al. 2005; Van Koten and Gray 2006), and analogy based estimation (Shepperd and Schofield 1997; Auer et al. 2006; Lee and Lee 2006; Chiu and Huang 2007; Li et al. 2007; Li and Ruhe 2008)

The analogy based estimation (ABE) was first proposed by Shepperd and Schofield (1997) as a valid alternative to expert judgment and algorithmic estimation. ABE is partially motivated by the obvious connections between project managers making estimation based on the memories of past similar projects and the formal use of analogies in Case Based Reasoning (CBR) (Kolodner 1993). The fundamental principle of ABE is simple: when provided a new project for estimation, the most similar historical projects (analogies) are retrieved, the solutions (cost values) of the retrieved projects are used to construct a ‘retrieved solution’ to the new project, with the expectation that the cost values of the retrieved projects will be similar to the real cost of the new project.

However, the adjustment on the retrieved solution is of necessity since it can capture the differences between the new project and the retrieved projects, and refine the retrieved solution into the target solution (Walkerden and Jeffery 1999). In the literature, many works (Walkerden and Jeffery 1999; Jorgensen et al. 2003; Chiu and Huang 2007; Li et al. 2007; Li and Ruhe 2008) have been focusing on the adjustments to the retrieved solution. However, most of these adjustment mechanisms are based on predetermined linear forms without learning ability to adapt to more complex situations such as non-normality. In addition, these adjustment techniques are limited to the numeric features despite that the categorical features also contain valuable information to improve the cost estimation accuracies (Angelis et al. 2000). In contrast, software project datasets often exhibit non-normal characteristics (Pickard et al. 2001) and contain large proportion of categorical features (Sentas and Angelis 2006; Liu and Mintram 2005).

To improve the existing adjustment mechanisms, we propose a more flexible non-linear adjustment with learning ability and including categorical features. The Non-linearity adjusted Analogy Based Estimation (NABE) is realized by adding a non-linear component (Artificial Neural Network) onto the retrieved solution of the ABE system. In this approach, the ordinary ABE procedure is first executed to produce an un-adjusted retrieval solution to the new project. Then, the differences between the new project's features and its analogies' features are treated as inputs to ANN model to generate the non-linear adjustment. Finally, the retrieved solution and the adjustment from ANN are summed up to form the final prediction.

The rest of this paper is organized as follows: Section 2 presents the related work on the adjustments of analogy based cost estimation, the detailed comparisons of the existing adjustment mechanisms and how they are related to the properties of the project datasets. Section 3 describes the details of non-linearity adjusted ABE system (NABE). Section 4 presents four real world data sets and the evaluation criteria for experiments. Section 5 provides an illustrative example of the application procedure of NABE. In Section 6, the NABE is tested on the real world datasets and is compared against the linear adjusted ABEs, ANN CART and SWR. In Section 7, eight artificial data sets are generated and a systematic analysis is conducted to explore how the model accuracies are related to dataset properties. In Section 8, the threats to validity are presented. The final section presents the conclusions and future works.

## 2 Related Work and Motivations

### 2.1 Related Work

Analogy based software cost estimation is essentially a case-based reasoning (CBR) approach (Shepperd and Schofield 1997). This approach identifies one or more historical projects that are similar to the present project and then derives the cost estimates from the similar projects. Generally, the ABE consists of four parts (Fig. 1): a case/project data base, a similarity function, a retrieved solution and the associated retrieval rules (Kolodner 1993).

**Fig. 1** The general framework of analogy based estimation

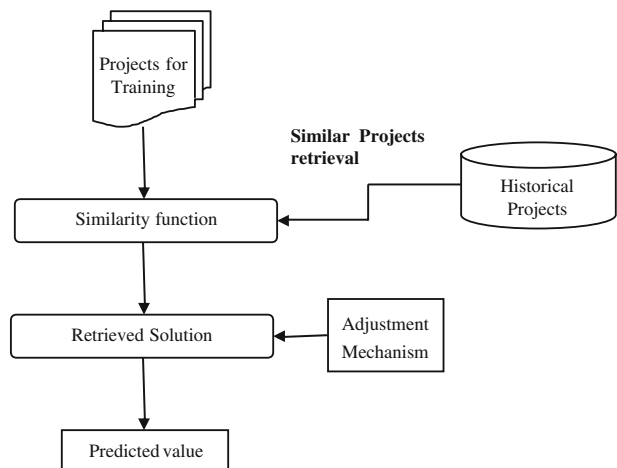


Figure 1 shows that the retrieved solution function is a crucial component in ABE, since it obtains the adjustment and produces the final prediction. Retrieved solution has a general mathematical form shown the following formula:

$$\hat{C}_x = g(C_1, C_2, \dots, C_n) \quad (1)$$

where  $\hat{C}_x$  denotes the estimated cost for the new project  $x$ ,  $C_i$  is the cost value of the  $i$ th closest analogy to project  $x$ , and  $n$  denotes the total number of closest analogies. The retrieved solution function (1) only includes the ‘cost’ values as its variables while other project features such as ‘lines of source code’ and ‘function points’ do not appear in this function. In the literature several retrieved solutions have been proposed, such as un-weighted mean (Shepperd and Schofield 1997; Jorgensen et al. 2003), weighted mean (Mendes et al. 2003), and median (Angelis and Stamelos 2000; Mendes et al. 2003). However, these solution functions can be rarely directly applied to predict  $\hat{C}_x$ . Instead, they need to be adjusted in order to fit the situations of the new project (Walkerden and Jeffery 1999). Therefore the adjustment mechanisms should first identify the differences between the new project (features) and the retrieved projects (features) and then convert these differences into the amount of change in the cost value. In the literature, many adjustment techniques have been proposed:

Walkerden and Jeffery (1999) first proposed the linear size adjustment. This approach performs a linear extrapolation along a single metric—the number of function points (FP) which is supposed to be strongly correlated with the project cost.

$$\hat{C}_x = \frac{FP_x}{FP_{CA}} C_{CA} \quad (2)$$

where  $\hat{C}_x$  denotes the estimated cost of a new project  $x$ ,  $C_{CA}$  is the cost value of the closest analogy (CA) or the most nearest neighbor,  $FP_x$  is the number of function points of the new project  $x$ , and  $FP_{CA}$  is the number of function points of the closest project. However, this adjustment technique may not be applicable when the size measure is not function point or there are many size measures other than function points such as the size measures of website projects (Mendes et al. 2001). Thus, based on Walkerden and Jeffery’s work, Mendes and Mosley (2003) further extended the naive form into an arbitrary number of size related features with multiple closest analogies.

$$\hat{C}_x = \frac{1}{K} \sum_{i=1}^K \frac{1}{Q} \left( \sum_{p=1}^Q \frac{s_{qx}}{s_{qi}} C_i \right) \quad (3)$$

where  $C_i$  is the project cost of the  $i$ th closest analogy,  $K$  is the total number of retrieved closest analogies,  $s_{px}$  is the  $q$ th size related feature of the new project  $x$ ,  $s_{pi}$  is the  $q$ th size related feature of the  $i$ th closest project, and  $Q$  is the total number of size related features. In a later study, Kirsopp et al. (2003) conducted an empirical investigation into the above two types of linear adjustments on ABE system.

Later, Jorgensen et al. (2003) proposed the ‘Regression Toward the Mean’ (RMT) adjustment mechanism:

$$\begin{aligned} \hat{C}_x &= FP_{CA} \times \hat{P}_x \\ \hat{P}_x &= P_{CA} + (M - P_{CA}) \times (1 - r) \end{aligned} \quad (4)$$

where  $\hat{P}_x$  denotes the adjusted productivity (Productivity = Cost / Function Points) of the new project  $x$ ,  $P_{CA}$  is the productivity of the closest analogies,  $M$  is the average productivity of the similar projects, and  $r$  is the historical correlation between the non-adjusted analogy based productivity and the actual productivity as a measure of the expected estimation accuracy. This method can also be regarded as an extension of Walkerden and Jeffery’s model, as it adjusts the ratio  $P_{CA}=C_{CA}/FP_{CA}$  in (2) by adding a component  $(M-P_{CA})\times(1-r)$  representing ‘regression toward the mean’.

At a later stage, heuristic method has been applied onto adjustment techniques. Chiu and Huang (2007) proposed Genetic Algorithm (GA) to optimize a linear adjustment model:

$$\hat{C}_x = C_{CA} + Adj \tag{5}$$

where  $C_{CA}$  denotes the cost value of the closest analogy,  $Adj = \sum_{i=1}^m \alpha_i \times (s_{xi} - s_{CAi})$  is the linear adjustment term,  $S_{xi}$  is the  $i$ th feature of the new project  $x$ , and  $S_{CAi}$  denotes the  $i$ th feature of the closest analogy. Genetic Algorithm is used to optimize the coefficients  $\alpha_i$  in this equation.

More recently, the categorical features are included into the adjustment model. Li et al. (2007) and Li and Ruhe (2008) proposed AQUA and AQUA+ for cost estimation. In their works, the following similarity adjusted solution function is proposed:

$$\hat{C}_x = \sum_{i=1}^K \left[ \frac{Sim(x, i)}{\sum_{i=1}^K Sim(x, i)} \cdot C_i \right] \tag{6}$$

where  $C_i$  is the project cost of the  $i$ th closest analogies,  $Sim(x, i)$  is the similarity between project  $x$  and its  $i$ th analogy, and  $K$  is the total number of closest analogies. The similarity measure  $Sim(x, i)$  can deal with both numerical and categorical features. In AQUA, the similarity measure assigns equal weights to the project features to eliminate the impact of different features, while AQUA+ employs the rough set approach to weight each project feature.

## 2.2 Motivations

Subsequent to the short descriptions of the published adjustment techniques, we present the motivations of this study in this section. Table 1 characterizes each adjustment method from

**Table 1** Comparison of published adjustment mechanisms

Source	Adjustment function	Adjustment feature	Categorical feature	Learning ability	Number of analogies
Walkerden and Jeffery (1999)	Linear	Function point (FP)	No	No	One
Mendes et al. (2003)	Linear	Size related features	No	No	Multiple
Jorgensen et al. (2003)	Linear	Function point (FP)	No	No	Multiple
Chiu and Huang (2007)	Linear	Size related features	No	Yes	One
Li et al. (2007) and Li and Ruhe (2008)	Linear	All relevant features	Yes	No	Multiple

six aspects. The first column contains the source of the adjustment. The second column is the type of function (linear / non-linear) that the adjustment bases on. The third column describes the type of project feature used in the adjustment function. The fourth column indicates whether the categorical features are considered in the adjustment. The fifth column shows whether the adjustment function can learn from the training dataset to approximate a complex relationship. The last column presents the number of closest analogies (one / multiple) used in the adjustment function. The reasons for selecting these criteria are as follows. The *function type* reflects the basic structure of the adjustment model. The *adjustment feature*, *categorical feature*, and *number of analogies* together determine the inputs of adjustment model. The *learning ability* indicates whether the adjustment mechanism has the flexibility to adapt to complex relationships.

We can tell from Table 1 that most works are restricted on the linear functions without learning ability except the GA adjusted approach (Chiu and Huang 2007) and most works do not consider the categorical features except the similarity adjusted function (Li et al. 2007; Li and Ruhe 2008). To improve the adjustments mechanism, we propose the more flexible non-linear adjustment mechanism with learning ability and incorporating categorical features.

On the other hand, three relevant dataset characteristics are considered in our study: *non-normality*, *categorical feature*, and *dataset size*. These properties are likely to be relevant to the differences between the adjustment models. First, the *non-normality* is a commonly referred characteristic cross the software engineering datasets (Pickard et al. 2001). Many existing studies (Myrtveit and Stensrud 1999; Shepperd and Kadoda 2001; Mendes et al. 2003) have considered the non-normality as an influential factor to the accuracies of the models including analogy based methods. Generally, the higher degree of non-normality leads to lower modeling accuracy. This property appears to relate to the *function type* of the adjustment models since the linear models usually work well under the normal condition and non-linear models with adaptive abilities seems to produce better results under non-normal conditions. Several application studies of ANN in other research fields show that ANN model or ANN based models are robust to the non-normal datasets (Guh 2002; Chang and Ho 1999; Cannon 2007) and theoretically ANN is capable to approximate arbitrary relationships (Lawrence 1994). Therefore, it is expected that ANN based adjustment might enhance ABE model's robustness to non-normality.

Given the fact that *categorical features* frequently appear in software engineering datasets (Sentas and Angelis 2006; Liu and Mintram 2005) and they may enclose useful information which could distinguish the projects (Angelis et al. 2000), many papers start to incorporate the *categorical feature* into consideration (Angelis et al. 2000; Sentas et al. 2005; Li et al. 2007; Li and Ruhe 2008). However, most existing adjustment techniques do not consider categorical features. NABE aims incorporate categorical features into adjustment mechanism for improvement of total performance. Therefore, the appearance of categorical features is regarded as one important data set property in our study.

The *dataset size* is also an influential factor on ABE methods. The ABE system retrieves the similar cases from the historical project dataset. The dataset with more projects could provide larger searching space for ABE. If the data is not very heterogeneous, it might lead to a higher chance for good prediction. Several papers (Auer et al. 2006; Shepperd and Kadoda 2001; Shepperd and Schofield 1997) studied dataset size as one major factor on the accuracy of analogy based method. In both Shepperd and Schofield's paper and Auer's paper, the authors analyzed the trends in estimation accuracy as the datasets grow. While in Shepperd and Kadoda's work, they confirmed that ABE benefits from having larger training sets. In addition, Shepperd and Kadoda also found that ANN can achieve better

performance from large training sets. Hence, the *dataset size* seems to have some connection with the *learning ability* of ANN and ABE.

As discussed above, dataset characteristics have large impacts on the estimation results and therefore it is more fruitful to identify which is the preferable estimation system in a particular context rather than to search for the ‘best’ prediction system for any case.

### 3 Artificial Neural Networks for Non-Linear Adjustment

In this section, a detailed description of the non-linear adjusted analogy based estimation (NABE) is presented.

First of all, the non-linear component—ANN is briefly introduced. Artificial Neural Network (ANN) is one type of machine learning technique that have played an important role in approximating complex relationships (Lawrence 1994). Due to its excellent approximation capability, ANN has been widely applied for software cost estimation research (Gray and MacDonell 1997; Heiat 2002; De Barcelos Tronto et al. 2007).

In the ANN architecture there are typically three layers: the input layer, the hidden layers, and the output layer. All the layers are composed of neurons. The connections between neurons across layers represent the transmission of information between neurons. ANN has the following mathematical form:

$$y = y(x) = \sum_{j=1}^J w_j f \left( \sum_{i=1}^I v_{ij} f(x_i) + \alpha_j \right) + \beta + \varepsilon \quad (7)$$

where  $x$  is an  $I$ -dimensional vector with  $\{x_1, x_2, \dots, x_I\}$  as its elements,  $f(\cdot)$  is the user defined transfer function,  $\varepsilon$  is a random error with 0 as mean,  $J$  is the total number of hidden neurons,  $v_{ij}$  is the weight on the connection between the  $i$ th input neuron and the  $j$ th hidden neuron,  $\alpha_j$  is the bias in the  $j$ th hidden neuron,  $w_j$  is the weight on the connection between the  $j$ th hidden neuron and the output neuron, and  $\beta$  is the bias in the output neuron. The weights and biases are determined by the training procedure which minimizes the training error. The commonly used training error function Mean Square Error (MSE) is presented as follow:

$$E = \frac{1}{I} \sum_{i=1}^I (t^s - y^s)^2 \quad (8)$$

where  $y^s$  is the output of the network when the  $s$ th sample is the ANN input, and  $t^s$  is the  $s$ th training target. The classical Back Propagation (BP) algorithm is often used to update the weights and biases to minimize the training error.

As shown by formula (7), ANN has three user-defined parameters: the number of hidden layers, the number of hidden nodes and the type of transfer function. These parameters have a major impact on ANN’s prediction performance (Hagan et al. 1997). Among these parameters, one hidden layer is often recommended since multiple hidden layers may lead to an over parameterized ANN structure. For the number of hidden nodes, too few hidden nodes can compromise the ability of network to approximate a desired function. On the contrary, too many hidden nodes can lead to over-fitting. In our study, ANN is used as the adaptive non-linear adjustment component in NABE system. The NABE method and its system procedure are described in Section 3.1.

### 3.1 Non-linear Adjusted Analogy Based System

From the explanations in Section 2, the adjustment mechanism should capture the ‘update’ that transforms the solution from the retrieved projects into the target solution. Based on the linear adjustment model proposed by Chiu and Huang (2007), we extend the linear adjustment model to the following additive form:

$$\widehat{C}_x = C_{w/o} + f(\mathbf{s}_x, \mathbf{S}_k) \tag{9}$$

where  $f(\cdot)$  is an arbitrary function approximating the update that is necessary to change the retrieved solution into the target solution (in our study,  $f(\cdot)$  is the ANN model),  $\mathbf{s}_x$  is the feature vector of project  $x$ ,  $\mathbf{S}_k$  is the feature matrix of the  $K$  closest analogies and  $C_{w/o}$  is the cost value obtained from the ABE without adjustment (or the retrieved solution).

The NABE system consists of two stages. In the first stage NABE system obtains the retrieved (un-adjusted) solution and trains the non-linear component—ANN. In the second stage the non-linear component is used to produce the update and then the update is added up with the retrieved solution to generate the final prediction.

#### 3.1.1 Stage I—Training

The procedures of stage I are shown in Fig. 2. The jackknife approach (Angelis and Stamelos 2000) (also known as leave one out cross-validation) is employed for the training of the non-linear adjustment (ANN). For each project in the training dataset, the following steps are performed:

- Step 1: the  $i$ th project is extracted from the training dataset as the new project being estimated, and the rest projects are treated as the historical projects in ABE system.
- Step 2: the ABE system finds the  $K$  closest analogies from the historical projects by the similarity measure. In this study, the Euclidean distance is used to construct the similarity function  $Sim(i, j)$ :

$$Sim(i, j) = 1 / \left[ \delta + \sqrt{\sum_{q=1}^Q Dist(s_{iq}, s_{jq})} \right] \quad \delta = 0.0001$$

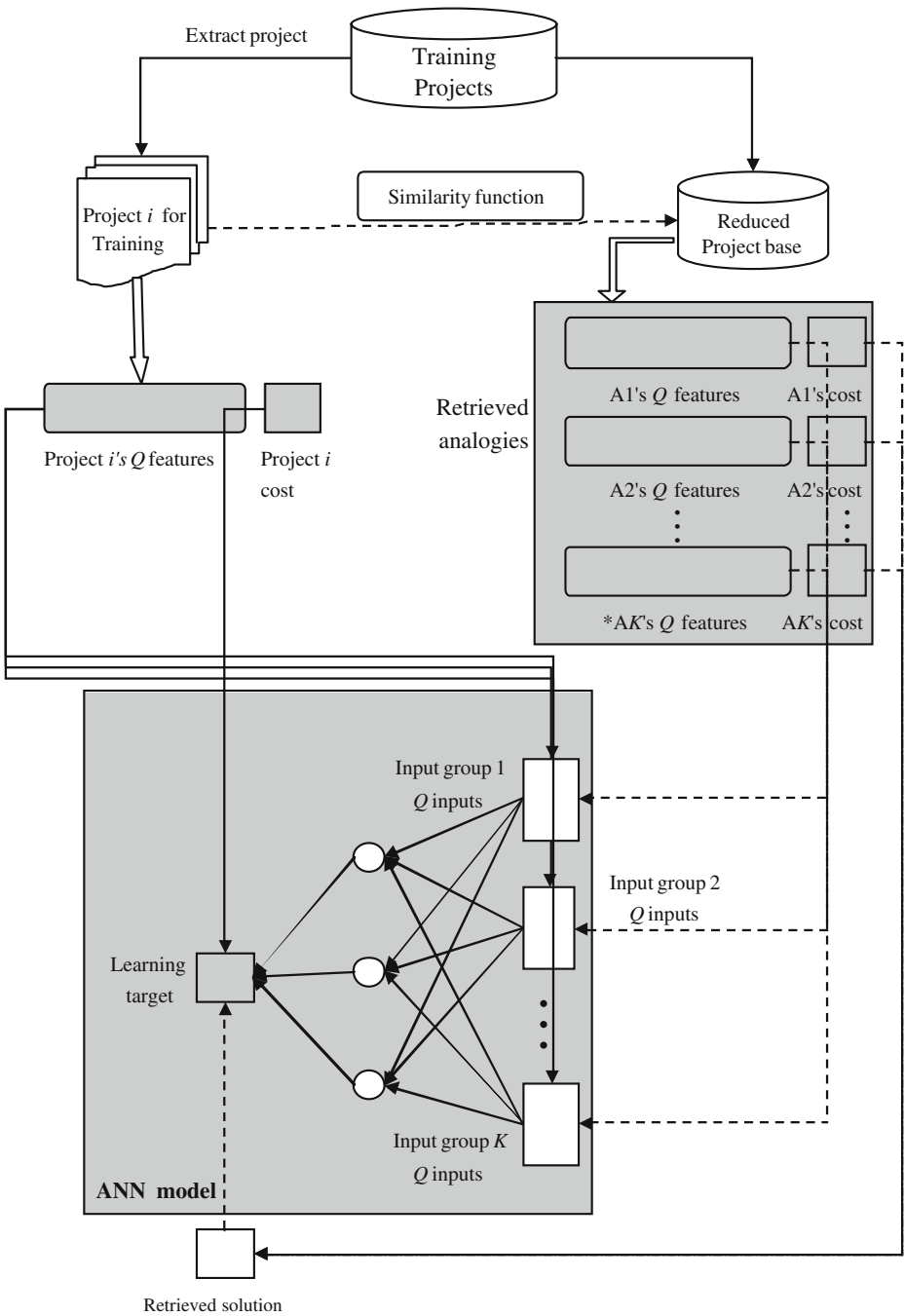
$$Dist = \begin{cases} (s_{iq} - s_{jq})^2 & \text{if } s_{iq} \text{ and } s_{jq} \text{ are numeric} \\ 1, & \text{if } s_{iq} \text{ and } s_{jq} \text{ are categorical and } s_{iq} = s_{jq} \\ 0, & \text{if } s_{iq} \text{ and } s_{jq} \text{ are categorical and } s_{iq} \neq s_{jq} \end{cases} \tag{10}$$

where  $i$  represents the project being estimated,  $j$  denotes one historical project,  $S_{iq}$  is the  $q$ th feature value of project  $i$ ,  $S_{jq}$  denotes the  $q$ th feature value of project  $j$ ,  $Q$  is the total number of features in each project and  $\delta=0.0001$  is a small constant to prevent the situation that

$\sqrt{\sum_{q=1}^Q Dist(s_{iq}, s_{jq})} = 0$ . In our similarity function, we use un-weighted Euclidean distance to eliminate the impacts of different feature weights.

After obtaining the  $K$  analogies, the retrieved solution (cost value) to the  $i$ th project is generated. For the sake of simplicity, the un-weighted mean (Shepperd and Schofield 1997) is used as retrieved solution in this study.





**Fig. 2** Training stage of the ANN adjusted ABE system with *K* closest analogies \* *AK* means the *K*<sup>th</sup> closest analogy of project *i*

Step 3: after obtaining the retrieved solution, the inputs and the training target are prepared to train the ANN model in (9). The inputs of ANN are the residuals between the features of project  $i$  and the features of its  $K$  analogies. The training target of ANN is the residual between the  $i$ th project’s real cost value and the retrieved solution from its  $K$  analogies:

$$C_i - \sum_{k=1}^K \frac{C_k}{K} = \sum_{j=1}^J w_j f \left( \sum_{k=1}^K \sum_{q=1}^Q v_{kqj} f (s_{iq} - s_{kq}) + \alpha_j \right) + \beta + \varepsilon \tag{11}$$

The left hand side of (11) is the training target: the difference between the real cost of project  $i$  and the retrieved solution of project  $i$ . The right hand side of (11) is the ANN model with  $s_{iq}$  as the  $q$ th feature of project  $i$ ,  $s_{kq}$  as the  $q$ th feature of its  $k$ th analogy (if  $s_{iq}$  and  $s_{kq}$  are categorical features, then  $s_{iq} - s_{kq} = 1$  when  $s_{iq} = s_{kq}$ , and  $s_{iq} - s_{kq} = 0$  when  $s_{iq} \neq s_{kq}$ ), with  $w_j, v_{kqj}, \alpha_j$  and  $\beta$  as ANN weights and biases, with  $f(\cdot)$  as the transfer function, with  $J$  as the number of hidden neurons, with  $K$  as the total number of analogies, and with  $Q$  as the total number of features in each project. For example, if the  $i$ th project’s real cost is 40 and the retrieved solution is 21, then the targeting output of ANN is  $40 - 21 = 19$ .

Step 4. given the inputs and the targeting output, the Back Propagation (BP) algorithm is performed to update the parameter in (11) to minimize the training error MSE (8).

After repeating the above procedure to all the projects in training dataset, the training stage is completed and the system moves to the testing stage.

### 3.1.2 Stage II—Predicting

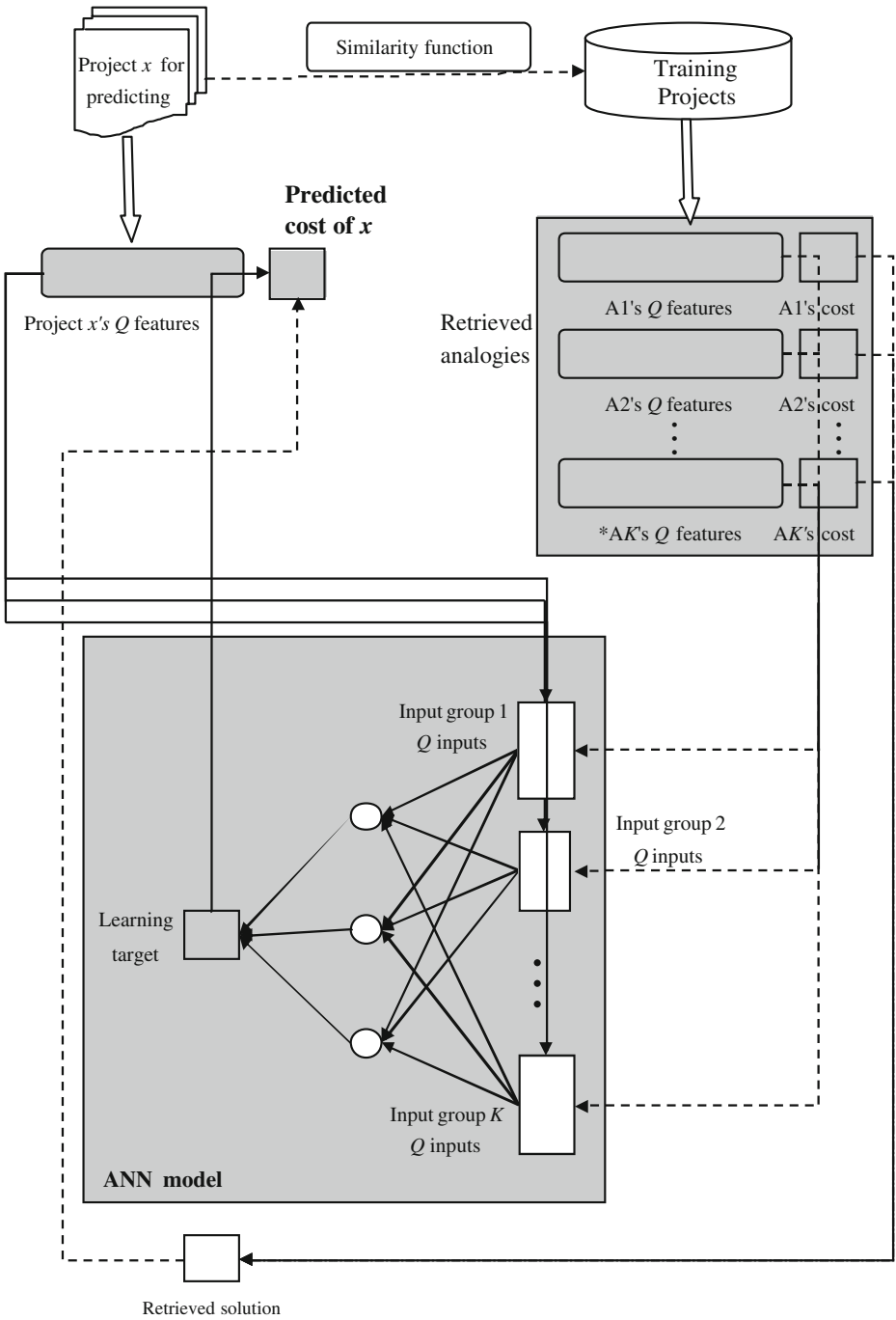
The predicting stage is illustrated in Fig. 3. At this stage, a new project  $x$  is presented to the trained NABE system. Then, a set of  $K$  analogies are retrieved from the training dataset by applying (10) to calculate the similarities. After obtaining the  $K$  analogies, the retrieved solution function is used to generate the un-adjusted prediction, and the differences between features of project  $x$  and its  $K$  analogies are inputted into the trained ANN model to generate the adjustment. Finally, the ABE prediction and the ANN adjustment are summed up as the final prediction:

$$\hat{y}(x) = \sum_{k=1}^K \frac{C_k}{K} + \sum_{j=1}^J w_j f \left( \sum_{k=1}^K \sum_{q=1}^Q v_{kqj} f (s_{xq} - s_{kq}) + \alpha_j \right) + \beta \tag{12}$$

For a better illustration of the NABE system procedure, an application example is given in Section 5. Before the example, the accuracy evaluation criteria and the real world datasets are presented in Section 4.

## 4 Evaluation Criteria and Data Sets

The evaluation criteria and real world data sets for experiments are presented in Section 4.1 and Section 4.2 respectively.



**Fig. 3** Predicting stage of the ANN adjusted ABE system with  $K$  closest analogies \*  $AK$  means the  $K^{\text{th}}$  analogies of project  $x$

#### 4.1 The Evaluation Criteria

Evaluation criteria are essential to the experiments. In the literature, several quality metrics have been proposed to assess the performances of estimation methods. More specifically, Mean Magnitude of Relative Error (MMRE), PRED(0.25) (Conte et al. 1986), and Median Magnitude of Relative Error (MdMRE) (Jorgensen et al. 1995) are three popular metrics.

The MMRE is defined as below:

$$MMRE = \frac{1}{n} \times \sum_{i=1}^n MRE \quad (13)$$

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

where  $n$  denotes the total number of projects,  $C_i$  denotes the actual cost of project  $i$ , and  $\hat{C}_i$  denotes the estimated cost of project  $i$ . Small MMRE value indicates the low level of estimation error. However, this metric is unbalanced and penalizes overestimation more than underestimation.

The MdMRE (Kitchenham et al. 2001) is the median of all the MREs.

$$MdMRE = median \left( \left| \frac{C_i - \hat{C}_i}{C_i} \right| \right) \quad (14)$$

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 value.

$$PRED(0.25) = \frac{1}{n} \times \sum_{i=1}^n (MRE \leq 0.25) \quad (15)$$

The PRED(0.25) identifies 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. In addition to the metrics mentioned above, there are several metrics available in the literature, such as Adjusted Mean Square Error (AMSE) (Burgess and Lefley 2001), Standard Deviation (SD) (Foss et al. 2003), Relative Standard Deviation (RSD) (Foss et al. 2003), and Logarithmic Standard Deviation (LSD) (Foss et al. 2003).

#### 4.2 Data Sets

Four well known real world datasets are chosen for experiments. The Albrecht dataset (Albrecht and Gaffney 1983) is a popular dataset used by many recent studies (Shepperd and Schofield 1997; Heiat 2002; Auer et al. 2006). This dataset includes 24 projects developed by third generation languages. Eighteen out of 24 projects were written in

COBOL, four were written in PL1, and two were written in DMS languages. There are five independent features: ‘Inpcout’, ‘Outcount’, ‘Quecount’, ‘Filcount’, and ‘SLOC’. The two dependent features are ‘Fp’ and ‘Effort’. The ‘Effort’ which is recorded in 1,000 person hours is the targeting feature of cost estimation. The detailed descriptions of the features are shown in Appendix A. The descriptive statistics is presented in Table 2. Among these statistics, the ‘Skewness’ and ‘Kurtosis’ are used to quantify the degree of non-normality of the features (Kendall and Stuart 1976). It is noted that Albrecht is a relatively small dataset with high order non-normality comparing to the rest three datasets.

The Desharnais dataset was collected by Desharnais (1989). Despite the fact that Desharnais dataset is relatively old, it is one of the large and publicly available datasets. Therefore it still has been employed by many recent research works, such as Mair et al. (2000), Burgess and Lefley, (2001), and Auer et al. (2006). This data set includes 81 projects (with nine features) from one Canadian software company. Four of 81 projects contain missing values, so they have been excluded from further investigation. The eight independent features are ‘TeamExp’, ‘ManagerExp’, ‘Length’, ‘Language’, ‘Transactions’, ‘Entities’, ‘Envergure’, and ‘PointsAdjust’. The dependent feature ‘Effort’ is recorded in 1,000 *h*. The definitions of the features are provided in Appendix B. The descriptive statistics of all features are presented in Table 3. Table 3 shows that Desharnais is a larger dataset with relatively lower order non-normality comparing with Albrecht dataset.

The Maxwell dataset (Maxwell 2002) is a relative new dataset and has already been used by some recent research works (Sentas et al. 2005; Li et al. 2008b). This dataset contains 62 projects (with 26 features) from one of the biggest commercial banks in Finland. In this dataset, four out of 26 features are numerical and the rest features are categorical. The categorical features can be further divided into ordinal features and nominal features, and they have to be distinguished. When calculating the similarity measure, the ordinal features are treated as ‘numerical features’ since they are sensitive to the order while the nominal features are regarded as ‘categorical’ type. (See formula (10))

In Maxwell dataset, the numerical features are ‘Time’, ‘Duration’, ‘Size’ and ‘Effort’. The categorical features are ‘Nlan’, ‘T01’-‘T15’, ‘App’, ‘Har’, ‘Db’, ‘Ifc’, ‘Source’ and ‘Telonuse’. The ordinal features are ‘Nlan’, and ‘T01’-‘T15’. The nominal features are ‘App’, ‘Har’, ‘Db’, ‘Ifc’, ‘Source’ and ‘Telonuse’. The definitions of all the features are presented in Appendix C. The descriptive statistics of all features are provided in Table 4. It is shown that Maxwell is a relative large dataset with relatively lower order non-normality and larger proportion of categorical features comparing with Albrecht set and Desharnais set.

**Table 2** Descriptive statistics of all features of Albrecht dataset

Features	Mean	Std Dev	Min	Max	Skewness	Kurtosis
Inpcout	40.25	36.91	7.00	193.00	3.07	13.44
Outcount	47.25	35.17	12.00	150.00	1.28	4.29
Quecount	17.38	15.52	3.00	60.00	1.40	3.96
Filcount	16.88	19.34	0	75.00	1.94	6.46
Fp	61.08	63.68	3.00	318.00	2.90	12.19
SLOC	199.00	1,902.00	647.63	488.00	1.44	4.02
Effort	21.88	28.42	0.50	105.20	2.16	6.51

**Table 3** Descriptive statistics of all features of Desharnais dataset

Features	Mean	Std Dev	Min	Max	Skewness	Kurtosis
TeamExp	2.30	1.33	0	4.00	-0.05	1.73
ManagerExp	2.65	1.52	0	7.00	0.22	3.01
Length	11.30	6.79	1.00	36.00	1.43	5.49
Language	1.56	0.72	1.00	3.00	0.88	2.45
Transactions	177.47	146.08	9.00	886.00	2.34	10.09
Entities	120.55	86.11	7.00	387.00	1.36	4.37
Envergure	27.45	10.53	5.00	52.00	-0.19	2.58
PointsAdjust	298.01	182.26	73.00	1,127.00	1.81	7.67
Effort	4.83	4.189	0.55	23.94	2.00	7.89

The ISBSG (International Software Benchmarking Standards Group) has developed and refined its data collection standard over a ten-year period based on the metrics that have proven to be very useful to improve software development processes. The latest data release of this organization is the ISBSG R10 data repository (ISBSG 2007a) which contains totally 4,106 projects (with 105 features) coming from 22 countries and various

**Table 4** Descriptive statistics of all features of Maxwell data set

Features	Mean	Std Dev	Min	Max	Skewness	Kurtosis
Time	5.58	2.13	1.00	9.00	-0.42	2.25
App	2.35	0.99	1.00	5.00	0.96	4.11
Har	2.61	1.00	1.00	5.00	1.43	4.09
DbA	1.03	0.44	0.00	4.00	4.74	35.13
Ifc	1.94	0.25	1.00	2.00	-3.55	13.57
Source	1.87	0.34	1.00	2.00	-2.21	5.90
Telonuse	2.55	1.02	1.00	4.00	-0.04	1.91
Nlan	0.24	0.43	0.00	1.00	1.21	2.45
T01	3.05	1.00	1.00	5.00	-0.20	2.05
T02	3.05	0.71	1.00	5.00	-0.07	3.57
T03	3.03	0.89	2.00	5.00	0.51	2.51
T04	3.19	0.70	2.00	5.00	0.02	2.60
T05	3.05	0.71	1.00	5.00	0.48	4.98
T06	2.90	0.69	1.00	4.00	-0.46	3.49
T07	3.24	0.90	1.00	5.00	-0.08	2.52
T08	3.81	0.96	2.00	5.00	-0.17	1.97
T09	4.06	0.74	2.00	5.00	-0.58	3.32
T10	3.61	0.89	2.00	5.00	0.00	2.22
T11	3.42	0.98	2.00	5.00	0.12	2.02
T12	3.82	0.69	2.00	5.00	-0.66	3.83
T13	3.06	0.96	1.00	5.00	-0.24	2.35
T14	3.26	1.01	1.00	5.00	-0.15	2.37
T15	3.34	0.75	1.00	5.00	0.09	3.99
Duration	17.21	10.65	4.00	54.00	1.25	4.34
Size	673.31	784.08	48.00	3,643.00	2.28	7.80
Effort	8,223.21	10,499.90	583.00	63,694.00	3.27	15.52

organizations such as banking, communications, insurance, business services, government and manufacturing.

Due to the heterogeneous nature and the huge size of the entire repository, ISBSG recommends extracting out a suitable subset for any cost estimation practice (ISBSG 2007b). At the first step, only the relevant features characterizing projects should be considered to create the subset. Thus, we select out 14 important features (include project effort) suggested by ISBSG (ISBSG 2007b): ‘DevType’, ‘OrgType’, ‘BusType’, ‘AppType’, ‘DevPlat’, ‘PriProLan’, ‘DevTech’, ‘ProjectSize’ (consisting of six sub features: ‘InpCont’, ‘OutCont’, ‘EnqCont’, ‘FileCont’, ‘IntCont’, and ‘AFP’), and ‘NorEffort’. Then, the projects with missing values in any of the selected feature are excluded from the subset. Afterward, a further step is taken to refine the subset. In ISBSG dataset, project data quality is rated and only projects with A or B rating are used in published research works. Therefore the projects with the ratings other than A and B are excluded from the subset. Moreover, since the normalized effort (‘NorEffort’) is used as the target for estimation, the risk of using normalized effort should be noted. For project covering less than a full development life cycle, normalized effort is an estimate of the full development effort and this may introduce biasness. Hence the normalized ratio (normalized effort / summary effort) is used to refine the project subset. As suggested by ISBSG that the ratio up to 1.2 is acceptable (ISBSG 2007b), we filter out the projects with normalized ration larger than 1.2. Finally, the subset is further reduced to the projects with ‘Banking’ as ‘OrgType’. After all, the above procedures results to a subset with 118 projects.

The definitions of the project features are presented in Appendix D. The descriptive statistics of all features are summarized in Table 5. Table 5 and Appendix D show that the ISBSG subset is the largest dataset with high order non-normality and large proportion of categorical features comparing with the datasets above.

## 5 Application Example of NABE

This section presents an application example of the NABE system on Albrecht dataset.

**Table 5** Descriptive statistics of all features of ISBSG data set

Features	Mean	Std Dev	Min	Max	Skewness	Kurtosis
DevType	1.52	0.50	1.00	2.00	-0.07	1.00
BusType	7.55	6.36	2.00	15.00	0.29	1.11
AppType	5.76	2.14	1.00	9.00	0.18	1.85
DevPlat	6.25	4.50	1.00	12.00	0.03	1.12
PriProLan	1.45	0.77	1.00	4.00	1.87	6.07
DevTech	10.19	3.96	4.00	16.00	0.10	1.66
InpCont	75.05	128.38	0	780.00	3.37	15.78
OutCont	68.90	96.81	0	648.00	3.42	17.50
EnqCont	41.49	75.80	0	398.00	2.70	10.23
FileCont	61.25	79.03	0	383.00	2.24	8.23
IntCont	28.07	36.74	0	172.00	1.83	6.02
AFP	284.41	340.65	10.00	2,190.00	2.81	12.63
NorEffort	4,309.08	5,520.68	508.00	36,046.00	2.86	13.29

### 5.1 Stage I—Training

Step 1, suppose project  $i$  is used as the training target for the non-linear component (all feature values are normalized in to region  $[0, 1]$ ).

Features	Inpcount	Outcount	Quecount	Filcount	Fp	SLOC	Effort
Project $i$	0.22	0.38	0.08	0.16	0.13	0.32	0.17

Step 2, the ABE algorithm searches through the training dataset for the  $K=1$  analogy of the project  $i$ . Suppose that the retrieved closest analogy is project  $j$ :

Features	Inpcount	Outcount	Quecount	Filcount	Fp	SLOC	Effort
Project $j$	0.23	0.43	0.27	0.19	0.15	0.36	0.12

Step 3, the training target and inputs are calculated for the ANN component. The target output is: (project  $i$ 's effort)—(project  $j$ 's effort)= $0.17-0.12=0.05$ ; the inputs of ANN are the features of project  $i$  subtracting the corresponding features of project  $j$ :

Inputs	1	2	3	4	5	6
Residual	-0.01	-0.05	-0.19	-0.03	-0.02	-0.04

Step 4, the back-propagation algorithm is performed to train ANN by using the target and inputs from step 3.

Step 1 to step 4 are repeated cross all the projects in the training set. The trained ANN model is the non-linear adjustment that adapts to the relationship that transforms the retrieved solution to the target solution.

### 5.2 Stage II—Predicting

Suppose a new project  $x$  is being estimated and the effort value of this project is unknown.

Features	Inpcount	Outcount	Quecount	Filcount	Fp	SLOC	Effort
Project $x$	0.36	0.18	0.20	0	0.06	0.23	?

The ABE system firstly retrieves the closest analogy: project  $k$  from the training dataset.

Features	Inpcount	Outcount	Quecount	Filcount	Fp	SLOC	Effort
Project $k$	0.18	0.09	0.08	0	0.11	0.11	0.08



Therefore, the effort = 0.08 is the retrieved solution. Then the residuals between the feature values of project  $x$  and the feature values of project  $k$  are calculated and inputted into the ANN model.

Inputs	1	2	3	4	5	6
Residual	0.18	0.09	0.12	0	-0.05	0.12

After processing the inputs, the ANN generates an output = 0.07, according to its network structure obtained from the training stage. At last, the final prediction is produced by adding the ANN output to the retrieved solution ( $0.08+0.07=0.15$ ).

## 6 Experiments and Results

In this section, the proposed NABE is tested on the four datasets introduced in Section 4, comparing with the linear adjusted ABEs and other estimating methods including ANN, CART and SWR. In Section 6.1, the experiments design and the methods parameterizations are presented. From Section 6.2 to Section 6.5 the experimental results on the four datasets are summarized and analyzed.

### 6.1 Experiments Design

#### 6.1.1 Three-Fold Cross Validation

Prior to the experiments setup, all types of features are normalized into  $[0, 1]$  by dividing each feature value by that feature range, similarly to ANGEL (Shepperd and Schofield 1997). The three-fold cross-validation is used to assess the accuracies of the methods, similarly to Briand et al. (1999), Jeffery et al. (2001), and Mendes et al. (2003). Under this scheme, the data set is randomly divided into  $k=3$  equally sized subsets. At each time, one of the three subsets is used as the testing set exclusively for evaluating model prediction, and the rest two subsets are integrated to form a training set which is only used to construct the estimating model. This process is repeated three times and each subset has been used as testing set only one time. Finally the average training error and testing error across all three trials are computed. The advantage of cross validation scheme is that it matters little how the data is divided since every data point is assigned into a test set exactly once, and into a training set twice.

#### 6.1.2 Experiments Procedures

After determining the cross-validation scheme, the following procedures are performed to validate the proposed NABE system with comparisons against other methods on each dataset.

1. The performances of NABE are analyzed on both training set and testing set by varying the  $K$  number of analogies from 1 to 5 while keeping the similarity measure as the formula in (10) and the retrieved solution function as the ‘un-weighted mean’. The reason for changing  $K$  values is that  $K$  is an important parameter which determines the number of inputs to the non-linear adjustment. The similarity measure and retrieved

- solution are fixed because the focus of this study is on non-linear adjustment but these two parameters may not have direct impacts to the non-linear adjustment.
2. The optimal  $K$  value of the training practice ( $K$  minimizes the *MMRE* on training set) is selected to configure NABE for comparisons. Similarly, the best variants of other methods on the training sets are also obtained to compare with NABE. The training and testing results are summarized and analyzed.
  3. The Wilcoxon signed-rank tests ( $\alpha=0.05$ ) are performed to quantitatively identify the significance of difference in each pair-wised comparisons on testing sets.

### 6.1.3 Methods Specifications

Many cost estimation techniques are included for comparisons. They are: the standard ABE (Shepperd and Schofield 1997), the Linear size adjusted ABE (LABE) (Walkerden and Jeffery 1999), Regression toward the mean adjusted ABE (RABE) (Jorgensen et al. 2003), GA optimized linear adjusted ABE (GABE) (Chiu and Huang 2007), Similarity adjusted ABE (SABE) (Li and Ruhe 2008), and other popular cost estimation methods including the Classification and Regression Trees (CART) (Stensrud 2001), the Artificial Neural Network (ANN) (Mair et al. 2000) and Stepwise Regression (SWR) (Mendes et al. 2003).

To eliminate the impacts from different parameters, all types of ABE methods are implemented with fixed similarity measure (Euclidean) and retrieve solution (un-weighted mean). The only changeable parameter  $K$  number of analogies varies from 1 to 5. It is noted that, in SABE method the un-weighted similarity function is applied since the feature weighting is not included in this study.

For ANN, there are generally three parameters: the number of hidden nodes, the number of hidden layers and the types of hidden transfer functions. In our study, only one hidden layer is considered in order to avoid the over-parameterized ANN structure. The number of hidden nodes is chosen from the set  $\{1, 3, 5, 7, 10\}$  and the type of hidden transfer function is chosen from the set  $\{\text{Linear, Tan-Sigmoid, Log-Sigmoid}\}$ . Every combination of hidden node and hidden transfer function is evaluated on the training data. The optimal combination (minimize *MMRE*) is used for testing and comparisons.

The CART (Brieman et al. 1984) is a non-parametric and tree structured analysis procedure that can be used for classification and regression. When the tree structure is applied for numerical targets they are often called regression trees. CART has the following advantages: the capability of dealing with categorical features, the easily understandable diagram of complex data and the ability to identify the major subsets in the total dataset (Srinivasan and Fisher 1995). The construction of the CART involves recursively splitting the data set into (usually two) relatively homogeneous subsets until the terminate conditions are satisfied. The best tree is obtained by applying cross-validation on the training set using a spread minimization criterion. The best tree model is used in testing and comparisons.

For the stepwise regression method (SWR), the optimal regression model is determined from the forward stepwise procedure on the training dataset. Then the optimal linear equation is used in testing and comparisons. When the categorical features appear in the dataset, the optimal scaling (or CATREG) technique by Angelis et al. (2000) is utilized to build regression model based on both numerical and categorical features.

Finally, the random model (RAND) is also included in the comparisons as the control group to produce the estimation by randomly selecting any project's cost value from the dataset (training set or testing set).

All the methods are implemented via MATLAB code. The ANN component in NABE system and the ANN method in comparisons are trained by BP algorithm. The MSE error in (8) is used to determine how well the network is trained. The training stops when the MSE drops below the specified threshold = 0.01 in this study.

## 6.2 The Results on Albrecht Dataset

This section presents the results and comparisons on Albrecht dataset. Table 6 summarizes the three-fold cross validation results of NABE with different  $K$  values. It is observed that the setting  $K=4$  minimizes the training MMRE. Thus, the NABE system with  $K=4$  is chosen for the comparisons with other methods. In order to provide more insight on the magnitude of adjustment proposed by ANN, the ratios of absolute adjustment / non-adjusted values are calculated across the testing sets. The mean value of these ratio is 0.41 by NABE system with  $K=4$ .

Table 7 collects the training and testing results of the best variants of all cost estimation models. The configurations for ABE based methods are  $K=2$  for ABE,  $K=3$  for RABE,  $K=1$  for LABE,  $K=2$  for GABE and  $K=1$  for SABE. The testing results in Table 7 show that the NABE achieves the best values in MMRE, PRED(0.25) and MdmRE. Among other types of ABEs, LABE obtains the smallest MMRE, ABE achieves the maximum PRED (0.25), and SABE has the minimal MdmRE. In addition, it is noted that all methods have better performances than the random model. Another interesting observation is that some testing results are better than the training results. Some published cost estimation works (such as Chiu and Huang (2007) and Huang and Chiu (2006)) also reported similar patterns. This may be due to the fact that the machine learning techniques are data driven methods and they learn from examples without any knowledge of the model type. If the testing data happens to fit well to the model constructed on training data, then it is possible to have better testing results than training results.

To further analyze the testing performances, we draw out the box plots of absolute residuals, because absolute residuals are less sensitive to bias than the asymmetric MRE values (Stensrud et al. 2003). The plots in Fig. 4 show that NABE has a lower median, a shorter inter-quartile range, and fewer outliers than other methods. It is also observed that the distributions of absolute residuals are heavily skewed. This implies that the standard  $t$ -test is no longer valid for significance testing. Thus, the assumption-free Wilcoxon signed-rank tests are performed instead. The  $p$ -values of Wilcoxon signed-rank tests are summarized in Table 8.

Table 8 summarizes the  $p$ -values of Wilcoxon tests of NABE versus other method. Four paired comparisons have  $p$ -values smaller than 0.05. They are NABE v.s. RABE, NABE v.

**Table 6** Results of NABE on Albrecht dataset

$K$ value	Training			Testing		
	MMRE	PRED(0.25)	MdmRE	MMRE	PRED(0.25)	MdmRE
$K=1$	0.84	0.13	0.64	0.70	0.50	0.28
$K=2$	0.87	0.33	0.40	0.48	0.38	0.41
$K=3$	0.89	0.46	0.28	0.59	0.46	0.29
$K=4$	0.82	0.29	0.31	0.41	0.36	0.25
$K=5$	0.93	0.42	0.29	1.01	0.33	0.39

**Table 7** Accuracy comparison on Albrecht dataset

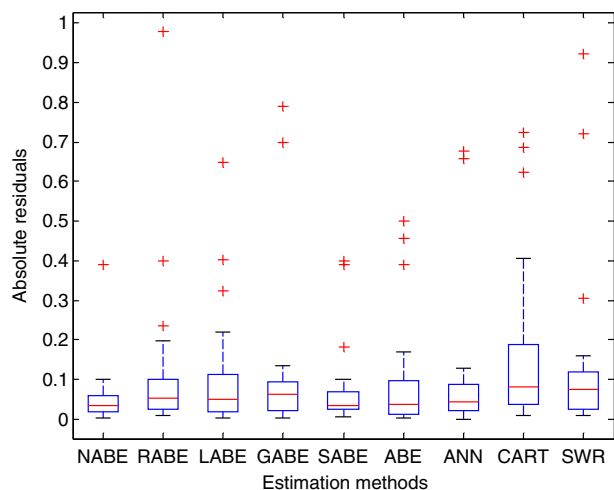
Methods	MMRE		PRED(0.25)		MdMRE	
	Training	Testing	Training	Testing	Training	Testing
NABE	0.82	0.41	0.29	0.36	0.31	0.25
RABE	0.85	0.66	0.37	0.21	0.36	0.45
LABE	0.81	0.61	0.29	0.21	0.39	0.53
GABE	0.92	0.77	0.40	0.33	0.45	0.48
SABE	0.84	0.81	0.33	0.25	0.41	0.46
ABE	0.93	0.87	0.29	0.33	0.46	0.43
ANN	0.97	0.85	0.46	0.33	0.30	0.39
CART	3.36	1.44	0.13	0.17	0.93	0.66
SWR	1.19	0.94	0.25	0.17	0.81	0.55
RAND	4.47	1.71	0.17	0.13	0.74	0.72

s. GABE, NABE v.s. CART, and NABE v.s. SWR. In addition, the improvements of NABE to other methods in terms of MMRE values are presented in Table 8. Four of the MMRE improvements are larger than 30% and the largest improvement is 60% on CART. The smallest improvement is 6% on LABE.

### 6.3 The Results on Desharnais Dataset

In this section, we present the results on Desharnais dataset in a way that similar to the presentations on Albrecht dataset. Table 9 illustrates the training errors and testing errors of NABE with regard to different  $K$  values. The  $K=2$  achieves the minimal training MMRE, thus NABE with  $K=2$  is chosen for comparisons with other methods. The average of the ratios of (absolute adjustment / non-adjusted prediction) is 0.03 on the testing sets.

Table 10 summarizes the training and testing errors of the best variants of all cost estimation models. The optimal parameters for ABE based methods other than NABE are:

**Fig. 4** Boxplots of absolute residuals on Albrecht dataset

**Table 8** NABE vs. other methods: *p*-values of the Wilcoxon tests and the improvements in percentages

	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR
<i>p</i> -value	0.02	0.19	0.04	0.08	0.08	0.12	0.00	0.00
Improvement on MMRE (%)	13	6	26	29	34	32	60	39

ABE with  $K=1$ , RABE with  $K=1$ , LABE with  $K=1$ , GABE with  $K=2$  and SABE with  $K=4$ . The testing results show that NABE achieves smallest MMRE and MdmRE, and second largest PRED(0.25). Among other types of ABEs, GABE obtains the smallest MMRE, RABE achieves the largest PRED(0.25) and the minimal MdmRE. It is also observed that the differences between NABE and other methods are not as apparent as those on Albrecht datasets. This observation maybe attribute to the characteristic of Desharnais dataset: moderate non-normality. It implies that all methods tend to perform equally good when the data set is close to normal distribution. As to the control group, all other methods have better predictions than the random model.

For further analysis, the box plots of absolute residuals on testing datasets are presented in Fig. 5. The plots in Fig. 5 show that NABE’s median is close to those of RABE, ANN and SWR; NABE has the shortest inter-quartile range, NABE gets five outliers while SABE and CART have fewer ones though their outliers are more extreme. The distributions of absolute residuals are skewed and therefore Wilcoxon tests are used to quantitatively investigate the differences between NABE and other methods.

In Table 11, the *p*-values from the Wilcoxon tests are presented together with the improvements on MMRE. Six out of eight *p*-values are larger than 0.05, and the rest two *p*-values are NABE vs. LABE = 0.02 and NABE vs. CART = 0.03. All the MMRE improvements are not larger than 30%. The largest improvement is 30% on SWR while the smallest improvement is 7% on GABE. These observations confirm the previous observation that on Desharnais dataset NABE does not perform significantly better than most methods and the performances of different methods are very close.

### 6.4 The Results on Maxwell Dataset

This section presents the results and comparisons on Maxwell dataset. Table 12 presents the three-fold cross validation results of NABE with different  $K$  values. The best setting  $K=3$  which minimizes the training MMRE is chosen for comparisons with other methods. The

**Table 9** Results of NABE on Desharnais dataset

<i>K</i> value	Training			Testing		
	MMRE	PRED(0.25)	MdmRE	MMRE	PRED(0.25)	MdmRE
$K=1$	0.41	0.44	0.28	0.66	0.34	0.45
$K=2$	0.39	0.40	0.35	0.52	0.36	0.32
$K=3$	0.51	0.36	0.36	0.73	0.27	0.49
$K=4$	0.52	0.30	0.42	0.64	0.21	0.50
$K=5$	0.42	0.38	0.33	0.69	0.23	0.46

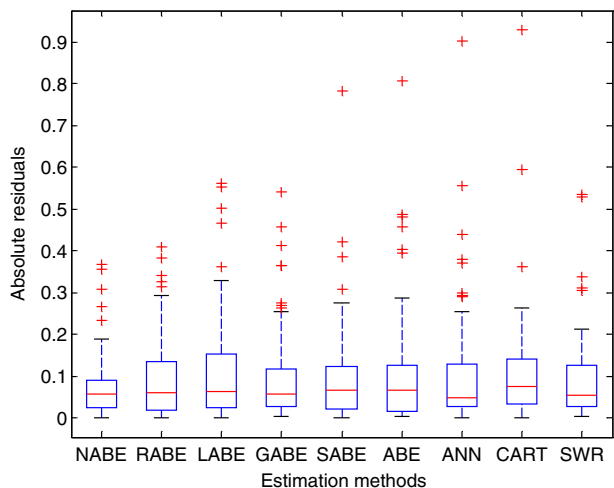
**Table 10** Accuracy comparisons on Desharnais dataset

Methods	MMRE		PRED(0.25)		MdmRE	
	Training	Testing	Training	Testing	Training	Testing
NABE	0.39	0.52	0.40	0.36	0.35	0.32
RABE	0.68	0.68	0.38	0.39	0.34	0.34
LABE	0.75	0.62	0.29	0.29	0.41	0.51
GABE	0.72	0.55	0.28	0.32	0.38	0.43
SABE	0.76	0.65	0.31	0.36	0.41	0.41
ABE	0.38	0.60	0.44	0.34	0.29	0.42
ANN	0.89	0.67	0.29	0.31	0.47	0.38
CART	0.58	0.71	0.31	0.25	0.41	0.44
SWR	0.67	0.73	0.35	0.35	0.39	0.34
RAND	1.81	1.14	0.12	0.18	0.67	0.60

mean of the ratios of (absolute adjustment / non-adjusted prediction) is 0.37 on the testing sets.

Table 13 presents the training and testing accuracies of different cost estimation models. The results from best variants of all methods are collected in this table. The configurations for ABE based methods are: ABE with  $K=3$ , RABE with  $K=3$ , LABE with  $K=2$ , GABE  $K=3$  and SABE  $K=4$ . The results show that NABE achieves best testing MMRE, PRED (0.25) and MdmRE. Among other types of ABEs, SABE obtains the smallest MMRE, LABE achieves the largest PRED(0.25), and SABE has the minimal MdmRE. As to the control group, all other methods seem to be better than the random model

To further analyze the testing results, we draw out the box plots of absolute residuals. The plots in Fig. 6 show that NABE has a median close to those of GABE and SABE; NABE has a inter-quartile range close to those of GABE, SABE and CART; NABE gets five outliers while RABE, GABE, ABE, ANN and SWR have fewer outliers though some

**Fig. 5** Boxplots of absolute residuals on Desharnais dataset

**Table 11** NABE vs. other methods: *p*-values of the Wilcoxon tests and the improvements in percentages

	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR
<i>p</i> -value	0.28	0.02	0.23	0.20	0.23	0.25	0.03	0.22
Improvement on MMRE (%)	24	17	7	20	14	23	27	30

of their outliers are more extreme. The distributions of absolute residuals suggest the Wilcoxon tests to identify the differences between NABE and other methods.

Table 14 summarizes the *p*-values of Wilcoxon tests and the improvements on MMRE values. Four out of eight *p*-values are smaller than 0.05. Two of the MMRE improvements are larger than 30%. The largest improvement is 48% on CART and the smallest improvement is 7% on GABE. These observations confirm the finding that NABE performs significantly better than other methods except SABE and GABE, on Maxwell dataset.

### 6.5 The Results on ISBSG Dataset

In this section, we present the results and comparisons on ISBSG dataset. Table 15 illustrates the training and testing errors of NABE with different *K* values. The *K*=2 achieves the minimal training MMRE and therefore NABE with *K*=2 is chosen for comparisons with other methods. The mean value of the ratios of (absolute adjustment / non-adjusted prediction) is 0.43 on the testing sets, which is close to those on Albrecht data set and Maxwell data set.

Table 16 summarizes the comparisons among the best variants of different cost estimation models. The optimal parameters for ABE based methods are: ABE with *K*=3, RABE with *K*=3, LABE with *K*=1, GABE with *K*=3 and SABE with *K*=5. The results show that the NABE achieves best testing MMRE, PRED(0.25), and MdmRE. Among other types of ABEs, SABE obtains the smallest MMRE, RABE achieves the largest PRED (0.25) and the minimal MdmRE. Comparing to the control group, all methods appear to be better than the random model.

The box plots of absolute residuals on testing sets are provided for further analysis. The plots in Fig. 7 show that NABE achieves a lower median, the shorter inter-quartile range than other methods. Another observation is that all methods are prone to extreme outliers. This maybe attribute to the fact that ISBSG dataset were collected inter-organizationally and internationally. Due to the diverse sources of data, even two similar projects might have quite different amount of cost. In the next step, Wilcoxon tests are used to assess the differences between NABE and other methods.

**Table 12** Results of NABE on Maxwell dataset

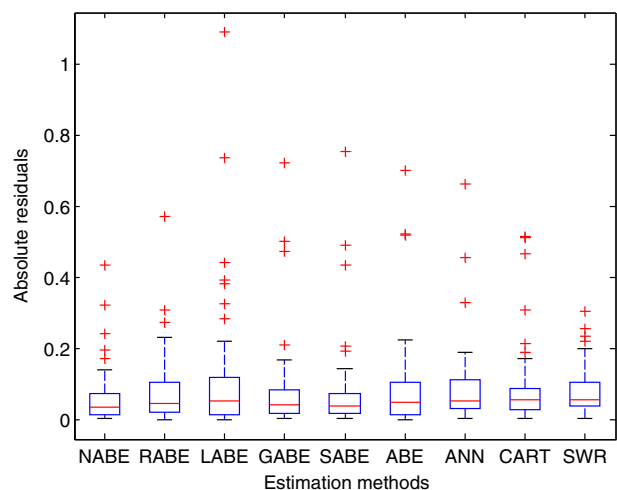
<i>K</i> value	Training			Testing		
	MMRE	PRED(0.25)	MdmRE	MMRE	PRED(0.25)	MdmRE
<i>K</i> =1	0.91	0.23	0.61	1.21	0.16	0.57
<i>K</i> =2	0.87	0.27	0.61	1.22	0.21	0.58
<i>K</i> =3	0.80	0.23	0.51	0.80	0.35	0.45
<i>K</i> =4	0.89	0.21	0.54	0.77	0.19	0.49
<i>K</i> =5	0.89	0.24	0.56	0.93	0.19	0.56

**Table 13** Accuracy comparisons on Maxwell dataset

Methods	MMRE		PRED(0.25)		MdmRE	
	Training	Testing	Training	Testing	Training	Testing
NABE	0.80	0.80	0.23	0.35	0.51	0.45
RABE	0.78	0.88	0.23	0.16	0.57	0.62
LABE	0.74	1.08	0.24	0.31	0.44	0.63
GABE	0.92	0.98	0.21	0.26	0.45	0.52
SABE	0.94	0.85	0.15	0.23	0.60	0.50
ABE	0.92	1.04	0.23	0.21	0.63	0.62
ANN	1.19	1.32	0.34	0.13	0.52	0.62
CART	1.60	1.52	0.23	0.26	0.61	0.65
SWR	1.53	1.09	0.18	0.23	0.65	0.76
RAND	2.49	1.70	0.16	0.05	0.66	0.81

In Table 17, the  $p$ -values from the Wilcoxon tests are presented together with the improvements on MMRE. In this table, all  $p$ -values are not larger than 0.05. As to the MMRE improvement, four MMRE improvements are larger than 30%. The largest improvement is 48% on RABE while the smallest improvement is 14% on SWR.

Besides the three-fold-cross validation, we conduct a different round of testing on ISBSG dataset. In this experiment, we select the test subset to consist of 33% more recent projects (completed in year 2000 and 2001) and the training subset to consist of 66% older projects (completed from year 1993 to 1999), because this would provide a more realistic setting: in real life applications the experimenter of NABE would train the method on old projects and apply it to the incoming projects. The results on training and testing subsets are summarized in Table 18. It is shown that most methods achieve better results than in three-fold cross-validation. NABE achieves best results under all error metrics. Among other ABEs, ABE obtains the smallest MMRE, and LABE achieves the largest PRED(0.25) and the minimal MdmRE. Comparing to the control group, all methods appear to perform better than the random model.

**Fig. 6** Boxplots of absolute residuals on Maxwell dataset



**Table 14** NABE vs. other methods:  $p$ -values of the Wilcoxon tests and the improvements in percentages

	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR
$p$ -value	0.09	0.02	0.14	0.15	0.06	0.00	0.02	0.02
Improvement on MMRE (%)	11.00	27.00	7.00	20.00	24.00	40.00	48.00	28.00

## 7 Analysis on Dataset Characteristics

In Section 6, results and comparisons are presented on each real dataset individually. However, the results vary dramatically from one dataset to another. For instance, NABE is statistically better than RABE on ISBSG dataset ( $p=0.02$ ) but their performances are statistically equal to each other on Desharnais dataset ( $p=0.28$ ). This is probably due to the fact that model accuracies are not only affected by the parameters selections but also affected by other factors such as the dataset characteristics (Shepperd and Kadoda 2001). In this section, we conduct a systematic investigation in order to explore the relationship between model accuracy and the dataset characteristics, and identify under which conditions NABE is the preferred prediction system and under what conditions other methods is also recommendable.

Table 19 summarizes a set of characteristics of the real world datasets. The columns in this table list the dataset ID, the number of projects, total number of features, the number of categorical features, and average of absolute skewness and kurtosis of each feature. The skewness and kurtosis values together reflect the degree of non-normality of the dataset.

This table provides some insights to each dataset. It is shown that software dataset often exhibits a mixture of several characteristics such as skewness and excessive outliers (kurtosis). These characteristics do not always appear in the same degree. In some cases they are moderate such as the Albrecht dataset, while in other cases they are severe such as the ISBSG dataset. It is also noted that the data sets are largely contrast to each other, for example Albrecht dataset has a relatively small size and small proportion of categorical features while Maxwell dataset is larger and has large proportion of categorical features. However, based on only the real world datasets, there are still some difficulties for a systematic analysis. The real dataset properties are uncontrollable and the real world datasets can not cover the full range of the combinations of the properties being studied.

Artificially generated dataset by simulation (Pickard et al. 2001; Shepperd and Kadoda 2001) is a feasible solution to the above difficulties. This approach generates artificial dataset from predefined distributions and equations. The simulated dataset provides the researcher with more control over the characteristics of a dataset. Especially, it enables the researcher to vary one property at a time and thus allows a more systematic exploration of

**Table 15** Results of NABE on ISBSG dataset

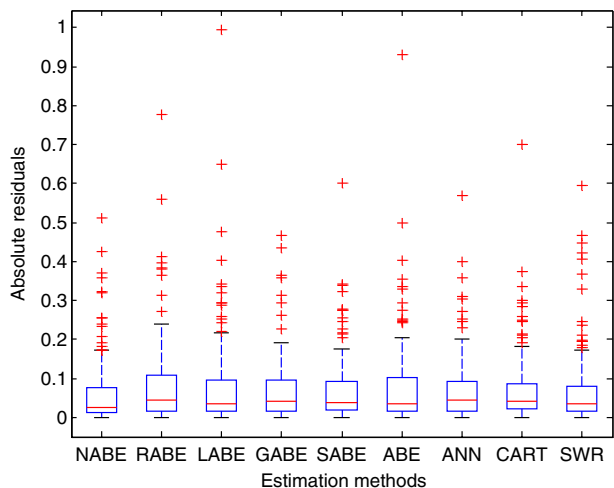
$K$ value	Training			Testing		
	MMRE	PRED(0.25)	MdMRE	MMRE	PRED(0.25)	MdMRE
K=1	0.98	0.25	0.60	0.87	0.26	0.55
K=2	0.89	0.33	0.46	0.74	0.31	0.42
K=3	0.97	0.26	0.53	0.89	0.22	0.49
K=4	1.00	0.15	0.63	0.95	0.22	0.58
K=5	1.10	0.10	0.69	1.03	0.23	0.61

**Table 16** Accuracy comparisons on ISBSG dataset

Methods	MMRE		PRED(0.25)		MdMRE	
	Training	Testing	Training	Testing	Training	Testing
NABE	0.89	0.74	0.33	0.30	0.46	0.42
RABE	1.16	1.36	0.28	0.28	0.51	0.54
LABE	1.19	1.13	0.29	0.17	0.53	0.58
GABE	1.13	1.09	0.25	0.21	0.54	0.60
SABE	0.91	0.85	0.23	0.18	0.51	0.58
ABE	0.97	0.98	0.16	0.22	0.63	0.59
ANN	0.82	0.96	0.27	0.25	0.49	0.60
CART	1.26	1.07	0.19	0.18	0.73	0.61
SWR	0.77	0.82	0.29	0.19	0.54	0.60
RAND	2.17	2.29	0.13	0.09	0.73	0.70

the relationship between dataset characteristics and model accuracies. As a simple but powerful tool for empirical evaluations, this technique has been frequently implemented by several recently published studies (Myrtveit et al. 2005; Li et al. 2008a).

Besides the simulation approach, bootstrapping (Efron and Gong 1983) is often used to produce artificial datasets to study the uncertainties in the predictions (Angelis and Stamelos 2000). Its principle is to generate several new datasets with the same size as the original dataset by randomly sampling original data with replacement. Each new dataset may have some items from the original dataset appearing more than once while some not appearing at all. However, bootstrapping is not considered for artificial dataset generation in this study. The reason is that our study mainly emphasizes on varying dataset properties to investigate the relationships between dataset properties and model accuracies but bootstrapping only generates a series of datasets based on original data and offers limited variability to change the dataset properties. On the other hand, the simulation technique provides a more explicit control over the dataset properties such as adjusting the distribution parameters to vary the skewness and kurtosis of the variable distribution.

**Fig. 7** Boxplots of absolute residuals on ISBSG dataset

**Table 17** NABE vs. other methods:  $p$ -values of the Wilcoxon tests and the improvements in percentages

	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR
$p$ -value	0.02	0.05	0.04	0.05	0.02	0.02	0.01	0.02
Improvement on MMRE (%)	48	31	35	17	29	27	34	14

In Section 7.1, we simulate eight artificial datasets to match the eight different combinations of the three data characteristics. Due to the computational limits, we only considered two levels for each characteristic: such as Large/Small for the ‘Dataset size’, Large/Small for the ‘Proportion of categorical features’, and Severe/Moderate for the ‘Non-normality’.

### 7.1 Artificial Datasets Generation

In this section, we present the procedures of artificial datasets generation. We extend Pickard’s equation of artificial dataset generation in this work. Other types of simulation techniques for artificial dataset are also available in the literature. For more details, readers can refer to Shepperd and Kadoda (2001), Foss et al. (2003), and Myrtveit et al. (2005).

Based on Pickard’s method, we simulate the combinations of characteristics from the equation (16):

$$y = 1000 + 6x_1sk + 3x_2sk + 2x_3sk + 5x_4sk + 10x_5sk + x_6sk + e \quad (16)$$

The independent variables are  $x_1sk$ ,  $x_2sk$ ,  $x_3sk$ ,  $x_4sk$ ,  $x_5sk$ , and  $x_6sk$ . Among them,  $x_1sk$ ,  $x_2sk$ , and  $x_3sk$  are continuous variables, and  $x_6sk$  is categorical variable. The first variable  $x_1sk$  is treated as the feature ‘function point’ for the linear adjustment methods. The last term  $e$  in (16) is the normally distributed noise with mean 0 and variance 1. To simulate different proportions of categorical features (Large/Small),  $x_4sk$  and  $x_5sk$  are defined as categorical variables for the situation of large proportion (50%) while  $x_4sk$  and  $x_5sk$  are set to be continuous to represent the situation of small proportion of categorical features (16.7%).

The non-normality is represented by skewness and outliers (kurtosis). For the continuous variables, the skewnesses are generated by five independent Gamma distributed random

**Table 18** Accuracy comparisons on ISBSG dataset

Methods	MMRE		PRED(0.25)		MdMRE	
	Training	Testing	Training	Testing	Training	Testing
NABE	0.67	0.57	0.40	0.31	0.50	0.41
RABE	0.55	0.81	0.38	0.20	0.32	0.55
LABE	0.57	0.90	0.38	0.25	0.47	0.54
GABE	0.83	1.00	0.21	0.18	0.54	0.70
SABE	0.91	0.85	0.23	0.18	0.51	0.58
ABE	0.82	0.71	0.16	0.22	0.62	0.58
ANN	0.74	0.75	0.31	0.18	0.51	0.66
CART	0.80	0.87	0.28	0.17	0.41	0.57
SWR	0.67	0.86	0.15	0.15	0.61	0.86
RAND	1.43	2.93	0.12	0.13	0.70	0.95

**Table 19** Characteristics of the four real world datasets

Dataset	Number of Projects	Number of Features	Number of Categorical Features	Avg. Skewness	Avg. Kurtosis
Albrecht	24	7	0	2.03	7.27
Desharnais	77	9	1	1.18	5.03
Maxwell	62	26	6	0.97	5.42
ISBSG	118	14	7	1.67	7.42

variables  $x_1'$ ,  $x_2'$ ,  $x_3'$ ,  $x_4'$ , and  $x_5'$  with scale parameter  $\theta=2$  and shape parameter  $k=3$  representing the moderate skewness, and  $\theta=2$  and  $k=1$  for the severe skewness. For the categorical variables, the moderate skewnesses are simulated by the independent discrete random variables  $x_4'$ ,  $x_5'$ , and  $x_6'$  with the distribution  $\{P(X=1)=0.1; P(X=2)=0.1, P(X=3)=0.5, P(X=4)=0.2, P(X=5)=0.1\}$  and the severe skewnesses are simulated by the distribution  $\{P(X=1)=0.7; P(X=2)=0.1, P(X=3)=0.1, P(X=4)=0, P(X=5)=0.1\}$ . To vary the magnitude of the independent variable, we then multiply  $x_1'$  by ten to create variable  $x_1sk$ ,  $x_2'$  by three to create  $x_2sk$ ,  $x_3'$  by 20 to create  $x_3sk$ ,  $x_4'$  by five to create the variable  $x_4sk$ ,  $x_5'$  by two to create  $x_5sk$ , and  $x_6'$  by one to create  $x_6sk$ . The outliers are generated by multiplying or dividing the dependent variable  $y$  by a constant. We select 1% of the data points to be the outliers. Half of the outliers are obtained by multiplying while half of them are obtained by dividing. For the moderate outliers, we set the constant value as 2, while for the severe outliers, six is chosen to be the constant.

For dataset sizes, we generate 400 projects to form the large sized dataset and 40 projects to construct the small sized dataset. Table 20 summarizes the properties of the eight artificial datasets.

## 7.2 Comparisons on Modeling Accuracies

The experimental procedures presented in Section 6.1.2 are applied on all artificial datasets. The comparisons between NABE and other models are presented first, since the relative performances of NABE to other methods could provide more insights about how to choose appropriate cost estimation method under a certain condition. Table 21 summarizes the results of Wilcoxon signed rank tests. These significance tests assess the differences between the absolute residuals of NABE's predictions and the absolute residuals of other methods' predictions. The confidence limit is set at  $\alpha=0.05$ . In Table 21, the entry with 'Y'

**Table 20** Artificial datasets and properties

Dataset ID	Size (number of projects)	Number of Categorical features (proportion)	Degree of Non-normality (Avg. skewness, Avg kurtosis)
#1	Small (40)	Small (16.7%)	Moderate (0.75, 3.10)
#2	Small (40)	Small (16.7%)	Severe (2.32, 9.87)
#3	Small (40)	Large (50%)	Moderate (0.61, 3.37)
#4	Small (40)	Large (50%)	Severe (2.84, 9.71)
#5	Large (400)	Small (16.7%)	Moderate (0.93, 3.72)
#6	Large (400)	Small (16.7%)	Severe (3.21, 13.9)
#7	Large (400)	Large (50%)	Moderate (0.82, 3.63)
#8	Large (400)	Large (50%)	Severe (3.32, 10.09)

**Table 21** Comparative performance of NABE to other methods

Dataset ID	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR	Totals
#1									0
#2		Y						Y	2
#3	Y	Y				Y			3
#4	Y	Y	Y	Y	Y	Y	Y	Y	8
#5		Y	Y						2
#6		Y						Y	2
#7	Y	Y	Y		Y				4
#8	Y	Y	Y	Y	Y	Y	Y	Y	8

indicates that NABE performs significantly better than the method located in this entry's corresponding column. The last column summarizes the total number of 'Y's in each row (dataset).

The results in Table 21 show that NABE achieves better performance than all rest methods on datasets #4 and #8. Both have large proportions of categorical features and severe non-normality. This observation suggests that NABE might be the best choice among all methods in our study, when the dataset is highly non-normal and with large proportion of categorical features. This observation also confirms the findings on ISBSG dataset which has similar properties to dataset #8. Another interesting observation is that NABE obtains the equally good predictions to other methods on dataset #1 which has small size, small number of categorical features and moderate non-normality. Comparing to the real world datasets, Dataset #1's properties are closest to those of Desharnais set on which NABE also performs equally to other methods except LABE and CART.

The analysis above clarifies the conditions under which NABE is preferable to other methods. To further study the relationship between dataset property and model accuracy, we analyze the model predictions under single dataset characteristic.

### 7.3 Analysis on 'Size'

Table 22 summarizes the testing MMREs of each cost estimation model on the artificial datasets grouped under different 'size'. The results show that, NABE achieves the lowest MMREs on dataset #2, #4, #5, #6, #7, and #8. It is also observed that the dataset size might largely influence the prediction accuracies. More specifically, almost all the methods obtain smaller MMRE values on larger datasets.

**Table 22** Testing MMREs under different dataset size

Dataset	Size	NABE	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR
#1	Small	0.13	0.13	0.13	0.12	0.10	0.15	0.10	0.13	0.17
#2		0.14	0.19	0.25	0.14	0.15	0.18	0.14	0.22	0.44
#3		0.19	0.28	0.27	0.24	0.18	0.19	0.28	0.24	0.15
#4		0.28	0.45	0.44	0.48	0.45	0.47	0.41	0.44	0.68
#5	Large	0.08	0.15	0.17	0.18	0.08	0.09	0.08	0.09	0.14
#6		0.10	0.14	0.19	0.13	0.14	0.12	0.12	0.17	0.47
#7		0.08	0.23	0.20	0.21	0.13	0.20	0.14	0.12	0.12
#8		0.24	0.40	0.32	0.35	0.35	0.39	0.34	0.45	0.65

To further investigate the ‘size’ property, we compare the absolute residuals of predictions using the small datasets and the large datasets. The difference is tested by using the Mann-Whitney U test setting the confidence limit at  $\alpha=0.05$ , since the sample sizes are not equal (40 data points vs. 400 data points). The results are presented in Table 23. The entry with ‘Y’ means the difference between the datasets pair in its row is significant when using the model in its column. Table 23 shows that a larger dataset size may significantly reduce prediction error measured by absolute residuals. Most approaches including NABE could benefit from having larger datasets. However, SWR seems to be un-influenced by the dataset size. This maybe attribute to the fact that SWR constructs a hyperplane from the data with only a few critical data points. This finding also confirms the suggestion from Shepperd and Kadoda (2001) that for the machine learning methods large dataset size could reduce the prediction errors when other properties are fixed.

#### 7.4 Analysis on ‘Proportion of Categorical Features’

This section presents the analysis on the proportion of categorical features. Table 24 is essentially a re-arrangement of the rows in Table 22. In Table 24, the artificial datasets are grouped under different ‘proportion of categorical features’. It is observed that large proportion of categorical features may have negative impacts on the prediction accuracy. This finding is reflective of the fact that categorical features may have less statistical power compared with numerical features (Kirsopp et al. 2003).

Table 25 presents the results of Wilcoxon signed rank tests with confidence level at  $\alpha=0.05$  on the absolute residuals of predictions using the datasets with smaller number of categorical features and the datasets with larger number of categorical features. In general, all methods are more or less affected by this property. Among them, NABE, SABE and SWR are least sensitive to the categorical values. The probable reason is that CATREG technique is adopted in SWR model, and NABE and SABE both can make use of the categorical features in their adjustment mechanism.

#### 7.5 Analysis on ‘Degree of Non-Normality’

This section provides the analysis on degree of non-normality. Table 26 is also a re-arrangement of the rows in Table 22. In Table 26, the artificial datasets are grouped under different ‘degree of non-normality’. It is noted that most methods obtain larger MMRE values under severe non-normal conditions. This indicates a trend that the increase of non-normality may result in a decrease of the prediction accuracy. However, NABE appear least sensitive to non-normality while SWR seems to be most sensitive to non-normality. This observation supports our argument in Section 2.2 that ANN could enhance ABE’s robustness to non-normal data.

**Table 23** Mann-Whitney U tests of dataset size influences

Datasets pair	NABE	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR
#1 vs. #5	Y			Y		Y			
#2 vs. #6	Y	Y	Y			Y		Y	
#3 vs. #7	Y	Y	Y		Y		Y	Y	
#4 vs. #8	Y	Y	Y	Y	Y	Y	Y		

**Table 24** Testing MMREs under different proportions of categorical features

Dataset	Proportion of categorical features	NABE	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR
#1	Small	0.13	0.13	0.13	0.12	0.10	0.15	0.10	0.13	0.17
#2		0.14	0.19	0.25	0.14	0.15	0.18	0.14	0.22	0.44
#5		0.08	0.15	0.17	0.18	0.08	0.09	0.08	0.09	0.14
#6		0.10	0.14	0.19	0.13	0.14	0.12	0.12	0.17	0.47
#3	Large	0.19	0.28	0.27	0.24	0.18	0.19	0.28	0.24	0.15
#4		0.28	0.45	0.44	0.48	0.45	0.47	0.41	0.44	0.68
#7		0.08	0.23	0.20	0.21	0.13	0.20	0.14	0.12	0.12
#8		0.24	0.40	0.32	0.35	0.35	0.39	0.34	0.45	0.65

Table 27 presents the results of Wilcoxon signed rank tests with confidence level at  $\alpha=0.05$  on the absolute residuals of predictions using moderate non-normal datasets and severe non-normal datasets. The results confirm the finding from Table 26 that NABE is least sensitive to non-normality while SWR is most sensitive to the non-normal property. Table 27 also can partially support Shepperd and Kadoda’s (2001) argument that ABE is preferred to SWR if the dataset contains large proportion of outliers.

### 7.6 Summary of Analysis

This section summarizes the findings from the real world datasets and artificial datasets:

- Generally, NABE achieves better MMRE, PRED(0.25), and MdMRE values than most cost estimation methods on the four real world datasets. Its prediction performances (in terms of absolute residuals) are significantly better than other methods in 18 out of totally 32 comparisons (eight methods  $\times$  four datasets = 32 comparisons). For the artificial datasets, NABE achieves the lowest MMRE values on six out of eight datasets. Its performances in terms of absolute residuals are significantly better than other methods in 29 out of totally 64 comparisons.
- More specifically, NABE outperforms other methods on ISBSG dataset, artificial dataset #4 and artificial dataset #8. All have large proportion of categorical features and severe non-normality. This observation indicates that NABE could largely improve ABE on the dataset with high degree non-normality and large proportion of categorical features.
- NABE obtains equally good accuracies to most methods on Desharnais dataset and artificial dataset #1. This indicates that NABE may not be an ideal option on the dataset of small size, small proportion of categorical features and moderate non-normality comparing with other linear based adjustment mechanisms. Besides accuracy, simplicity

**Table 25** Wilcoxon tests of proportion of categorical features influences

Datasets pair	NABE	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR
#1 vs. #3		Y	Y	Y			Y	Y	
#2 vs. #4	Y	Y	Y	Y	Y	Y	Y	Y	Y
#5 vs. #7						Y			
#6 vs. #8	Y	Y	Y	Y	Y	Y	Y	Y	Y

**Table 26** Testing MMREs under different degrees of non-normality

Dataset	Non-normality	NABE	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR
#1	Moderate	0.13	0.13	0.13	0.12	0.10	0.15	0.10	0.13	0.17
#3		0.19	0.28	0.27	0.24	0.18	0.19	0.28	0.24	0.15
#5		0.08	0.15	0.17	0.18	0.08	0.09	0.08	0.09	0.14
#7		0.08	0.23	0.20	0.21	0.13	0.20	0.14	0.12	0.12
#2	Severe	0.14	0.19	0.25	0.14	0.15	0.18	0.14	0.22	0.44
#4		0.28	0.45	0.44	0.48	0.45	0.47	0.41	0.44	0.68
#6		0.10	0.14	0.19	0.13	0.14	0.12	0.12	0.17	0.47
#8		0.24	0.40	0.32	0.35	0.35	0.39	0.34	0.45	0.65

is also a critical criterion for model evaluation. Especially, when the models are equally accurate, the simpler ones become more preferable.

- There are significant relationships between the successes of NABE and dataset properties. First, the large size can generally improve NABE's performance. Second, comparing with linear based adjustments NABE appears to be less sensitive to the proportion of categorical features and the degree of non-normality.

## 8 Threats to Validity

This section presents the comments on the validities of our study based on the internal threats to validity and external threats to validity.

### 8.1 Internal Validity

The threats to internal validity include the following aspects: to focus on different adjustment mechanisms, we pre-determined the similarity measure and the retrieved solution function in ABE system. However, there are many other options. For the similarity measures there are alternatives based on Manhattan and Minkowski distances (Mendes et al. 2003; Huang and Chiu 2006; Li and Ruhe 2008), and for the retrieved solutions there are weighted mean and median (Angelis and Stamelos 2000; Mendes et al. 2003).

Moreover, feature selection (Kirsopp et al. 2003) and project selection (Li et al. 2008a) are important preprocessing steps of ABE method since there are often many irrelevant features and noisy projects in the software engineering datasets. The possibility of further improvement of the NABE systems also lies in the appropriate selection of relevant features and representative projects.

**Table 27** Wilcoxon tests of non-normality influences

Datasets pair	NABE	RABE	LABE	GABE	SABE	ABE	ANN	CART	SWR
#1 vs. #2			Y						Y
#3 vs. #4		Y	Y	Y	Y	Y	Y	Y	Y
#5 vs. #6									Y
#7 vs. #8	Y	Y		Y	Y	Y	Y	Y	Y



Furthermore, missing values often appear in the software engineering datasets. Many studies (Myrtveit et al. 2001; Strike et al. 2001; Jonsson and Wohlin 2006; Song and Shepperd 2007) have proposed different data imputation techniques to recover missing data by estimating replacement values. However, the miss values are excluded from our study. This might cause some difficulties for practitioners to apply the proposed NABE system to the datasets with missing values. For example during the ISBSG subset preparation, we realize that missing values cause the deletion of too many projects.

## 8.2 External Validity

The external validity represents the possibilities of generalizing the findings of our comparative studies.

The threats to external validity are as follows. First, the limitations on real world datasets make some difficulties to generalize our findings. Although the four real world datasets largely contrast to each other (from the simplest Albrecht to the largest and most complex ISBSG), additional real world datasets are required for a more comprehensive evaluation of our method. For the artificial datasets, eight artificial datasets are systematically generated to match the eight combinations of the three dataset properties in this study. However, other types of dataset characteristics such as multi-collinearity (Shepperd and Kadoda 2001) and heteroskedasticity (Pickard et al. 2001) may also influence the performance of NABE. Further studies on additional dataset characteristics are necessary to increase the external validity from this aspect.

Moreover, the non-linear adjustment proposed in our study is based on artificial neural networks; but other types of non-linear approximations such as Radius Basis Functions (Hardy 1971) and Support Vector Machines (Vapnik 1995) can also be employed as the non-linear adjustment. They may achieve better performance than ANN does, since they have fewer parameters than ANN and they have the regularization mechanism to prevent the over-fitting problem confronted by ANN. The reason to choose ANN in this study is that it has been widely accepted in cost estimation literature and it has the flexibility to adapt to complex relationships and the capability to process the categorical inputs.

## 9 Conclusions and Future Works

Analogy based estimation is one of the most widely studied methods in the software cost estimation literature. Given a new project, the ABE system retrieves similar projects from its historical project database and derives the cost prediction from the similar projects. The adjustment to the retrieved solution is of necessity since the adjustment recognizes the difference between the new project and historical information, and refines the retrieved solution into the target solution. However, most published adjustment mechanisms are based on predetermined linear forms without learning ability to adapt to more complex situations. In addition, these adjustments techniques often restrict to numeric features despite that the categorical features contain valuable information to improve the cost estimation accuracies. Moreover, given the fact that software project datasets often exhibit non-normal characteristics, it is hard to approximate the relationships among the projects by linear adjustments.

To improve the adjustments mechanism, this paper proposes a more flexible non-linear adjustment mechanism with learning ability and incorporating categorical features. The non-linearity adjusted Analogy Based Estimation (NABE) is implemented by adding a non-

linear component (Artificial Neural Network) onto the retrieved solution of the ABE system. The proposed NABE is validated on four real world datasets with the comparisons against the published linear adjusted ABEs and three well established methods: CART, ANN and SWR. The results and comparisons show that NABE generally achieves best MMRE, PRED(0.25) and MdmRE values on the real world datasets.

To answer the question: under what conditions NABE is preferred, we generate eight artificial datasets to analyze the relationships between model accuracies and dataset characteristics (non-normality, categorical feature, and dataset size). The analyses show that NABE performances significantly better than other methods on the artificial datasets with severe non-normality and large proportion of categorical features.

In the domain of cost estimation, the lessons learnt via this study are as follows:

- The non-linear based adjustment to ABE system is generally an effective approach to extend ABE's flexibility on complex datasets and improve the accuracy of ABE.
- NABE is likely to be a more accurate method than other types of ABE methods on the dataset with high degree of non-normality and large proportion of categorical features.
- On the dataset with a relatively small size, a relatively small proportion of categorical features and a moderate non-normality, NABE may not be an ideal option, since it is likely to have equal accuracy to other ABE methods and it has a more complex structure than other ABE methods.
- There are strong relationships between the successes of NABE and dataset properties (non-normality, categorical feature, and dataset size). Thus, the practitioners should be aware of the tradeoffs among datasets properties, model complexity and model accuracy, when implementing NABE.

There are some limitations of NABE. The similarity measure and retrieved solution function are pre-determined in this study. Future works can be done to investigate the sensitivities of these ABE parameters. Moreover, additional real world datasets and additional dataset characteristics can be explored to enhance the external validity of the current work. Thirdly, other types of non-linear approximators such as RBF and SVM could be considered in the future works.

**Acknowledgement** We would like to thank the anonymous reviewers for their insightful and constructive comments. This research was partially supported by a grant from A\*Star (SERC grant number 072 1340050) in Singapore.

## Appendix

### Appendix A Feature definition of Albrecht dataset

Features	Full name	Type	Description
Inpcount	Input count	<i>Numerical</i>	Count of inputs
Outcount	Output count	<i>Numerical</i>	Count of outputs
Quecount	Query count	<i>Numerical</i>	Count of queries
Filcount	File count	<i>Numerical</i>	Count of files
Fp	Function points	<i>Numerical</i>	Number of function points
SLOC	Lines of source code	<i>Numerical</i>	Lines of source code
Effort	Development effort	<i>Numerical</i>	Measured in 1,000 h

**Appendix B** Feature definition of Desharnais dataset

Features	Full name	Type	Description
TeamExp	Team experience	<i>Numerical</i>	Measured in years
ManagerExp	Manager's experience	<i>Numerical</i>	Measured in years
Length	Length of project	<i>Numerical</i>	Year of development
Transactions	Transactions	<i>Numerical</i>	Number of transactions
Entities	Entities	<i>Numerical</i>	Number of entities
PointsAdjust	Adjusted function points	<i>Numerical</i>	Number of adjusted function points
Envergure	Development environment	<i>Numerical</i>	Development environment
Language	Programming language	<i>Categorical</i>	1 = 1st generation 2 = 2nd generation 3 = 3rd generation
Effort	Development effort	<i>Numerical</i>	Measured in 1,000 h

**Appendix C** Feature definition in Maxwell dataset

Features	Full name	Type	Description
Time	Time	<i>Numerical</i>	Time = syear-1985 + 1, with levels: 1,2,3,4,5,6,7,8,9..
App	Application type	<i>Categorical</i>	1 = Information/on-line service (infServ) 2 = Transaction control, logistics, order processing (TransPro) 3 = Customer service (CustServ) 4 = Production control, logistics, order processing (ProdCont) 5 = Management information system (MIS)
Har	Hardware platform	<i>Categorical</i>	1 = Personal computer (PC) 2 = Mainframe (Mainfrm) 3 = Multi-platform (Multi) 4 = Mini computer (Mini) 5 = Networked (Network)
Dbas	Database	<i>Categorical</i>	1 = Relatnl (Relational) 2 = Sequentl (Sequential) 3 = Other (Other) 4 = None (None)
Ifc	User interface	<i>Categorical</i>	1 = Grafical user interface (GUI) 2 = Text user interface (TextUI)
Source	Where developed	<i>Categorical</i>	1 = In-house (Inhouse) 2 = Outsourced (Outsrccd)
Telonus	Telom use	<i>Categorical</i>	0 = No 1 = Yes
Nlan	Number of different development languages used	<i>Ordinal</i>	1 = one language used 2 = two languages used 3 = three languages used 4 = four languages used
T01	Customer participation	<i>Ordinal:</i>	1 = Very low
T02	Development environment adequacy		2 = Low
T03	Staff availability		3 = Nominal
T04	Standards use		4 = High
T05	Methods use		5 = Very high

**Appendix C** (continued)

Features	Full name	Type	Description
T06	Tools use		
T07	Software's logical complexity		
T08	Requirements volatility		
T09	Quality requirements		
T10	Efficiency requirements		
T11	Installation requirements		
T12	Staff analysis skills		
T13	Staff application knowledge		
T14	Staff tool skills		
T15	Staff team skills		
Duration	Duration	<i>Numerical</i>	Duration of project from specification until delivery, measured in months
Size	Application size	<i>Numerical</i>	Function points measured using the experience method
Effort	Effort	<i>Numerical</i>	Work carried out by the software supplier from specification until delivery, measured in hours

**Appendix D** Feature definition in ISBSG dataset

Features	Full name	Type	Description
DevType	Development type	<i>Categorical</i>	1 = Enhancement 2 = New development 3 = Re-development
OrgType	Organization type	<i>Categorical</i>	1 = Banking 2 = Communication 3 = Community services 4 = Computer, Software, ISP 5 = Electricity, Gas, Water; 6 = Financial, Property & Business Services; 7 = Insurance; 8 = Manufacturing; 9 = Government, Public Administration 10 = Transport & Storage; 11 = Wholesale & Retail Trade; 12 = Others.
BusType	Business Area Type	<i>Categorical</i>	1 = Accounting; 2 = Banking; 3 = Engineering; 4 = Financial; 5 = Insurance, Actuarial; 6 = Inventory; 7 = Legal; 8 = Logistics; 9 = Manufacturing 10 = Personnel; 11 = Research & Development; 12 = Sales & Marketing; 13 = Telecommunications;

## Appendix D (continued)

Features	Full name	Type	Description
AppType	Application Type	<i>Categorical</i>	14 = Others. 1 = Billing; 2 = Office information system, Executive information system, Decision support system; 3 = Electronic Data Interchange; 4 = Financial; 5 = Management Information System; 6 = Network Management, Communications; 7 = Process control, sensor control, real time; 8 = Transaction/Production System; 9 = Others.
DevPlat	Development Platform	<i>Categorical</i>	1 = Mainframe 2 = Mid-range 3 = Multi; 4 = Personal Computer.
PriProLan	Primary Programming Language	<i>Categorical</i>	1 = ABAP; 2 = Access; 3 = ASP; 4 = C; 5 = C++; 6 = COBOL; 7 = JAVA; 8 = Lotus Notes; 9 = NATURAL; 10 = ORACLE; 11 = PL/I; 12 = PL/SQL; 13 = PowerBuilder; 14 = SQL; 15 = Visual Basic; 16 = Others.
DevTech	Development Techniques	<i>Categorical</i>	1 = Business area modeling; 2 = Data Modelling; 3 = Event Modelling 4 = Joint Application Development; 5 = Multifunction teams 6 = Object Oriented Analysis; 7 = Object Oriented Design; 8 = Process Modelling; 9 = Prototyping; 10 = Rapid Application Development; 11 = WaterFall; 12 = Others.
InpCont	Input Count	<i>Numerical</i>	The count of inputs
OutCont	Output count	<i>Numerical</i>	The count of outputs
EnqCont	Enquiry count	<i>Numerical</i>	The count of enquiries
FileCont	File count	<i>Numerical</i>	The count of files
IntCont	Interface count	<i>Numerical</i>	The count of interfaces
AFP	Adjusted function points	<i>Numerical</i>	The adjusted function point-count number

## Appendix D (continued)

Features	Full name	Type	Description
NorEffort	Normalized work effort	<i>Numerical:</i>	For project covering less than a full development life cycle, this value is an estimate of the full development effort in hours.

## References

- Albrecht AJ, Gaffney J (1983) Software function, source lines of code, and development effort prediction. *IEEE Trans Softw Eng* 9:639–648. doi:[10.1109/TSE.1983.235271](https://doi.org/10.1109/TSE.1983.235271)
- Angelis L, Stamelos I (2000) A simulation tool for efficient analogy based cost estimation. *Empir Softw Eng* 5:35–68. doi:[10.1023/A:1009897800559](https://doi.org/10.1023/A:1009897800559)
- Angelis L, Stamelos I, Morisio M (2000) Building a software cost estimation model based on categorical data. *Proceedings of Seventh International Software Metrics Symposium*, 4–15
- Auer M, Trendowicz A, Graser B, Haunschmid E, Biffel S (2006) Optimal project feature weights in analogy-based cost estimation: Improvement and limitations. *IEEE Trans Softw Eng* 32:83–92. doi:[10.1109/TSE.2006.1599418](https://doi.org/10.1109/TSE.2006.1599418)
- Briand LC, El-Emam K, Surmann D, Wieczorek I, Maxwell KD (1999) An assessment and comparison of common cost estimation modeling techniques. *Proceeding of the 1999 International Conference on Software Engineering*, 313–322
- Brieman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth, Belmont
- Burgess CJ, Lefley M (2001) Can genetic programming improve software effort estimation? A comparative evaluation. *Inf Softw Technol* 43:863–873. doi:[10.1016/S0950-5849\(01\)00192-6](https://doi.org/10.1016/S0950-5849(01)00192-6)
- Cannon AJ (2007) Nonlinear analog predictor analysis: a coupled neural network/analog model for climate downscaling. *Neural Netw* 20(4):444–453. doi:[10.1016/j.neunet.2007.04.002](https://doi.org/10.1016/j.neunet.2007.04.002)
- Chang SI, Ho ES (1999) Two-stage neural network approach for process variance change detection and classification. *Int J Prod Res* 37(7):1581–1599. doi:[10.1080/002075499191148](https://doi.org/10.1080/002075499191148)
- Chiu NH, Huang SJ (2007) The adjusted analogy-based software effort estimation based on similarity distances. *J Syst Softw* 80:628–640. doi:[10.1016/j.jss.2006.06.006](https://doi.org/10.1016/j.jss.2006.06.006)
- Conte S, Dunsmore H, Shen VY (1986) *Software engineering metrics and models*. Benjamin Cummings, Menlo Park, CA
- Efron B, Gong G (1983) A leisurely look at the bootstrap, the jackknife, and cross-validation. *Am Stat* 37(1):36–48. doi:[10.2307/2685844](https://doi.org/10.2307/2685844)
- De Barcelos Tronto IF, Da Silva JDS, Sant Anna N (2007) An investigation of artificial neural networks based prediction systems in software project management. *J Syst Softw* (in press). Corrected Proof
- Desharnais JM (1989) *Analyse statistique de la productivité des projets informatique a partie de la technique des point des foncti on*. University of Montreal
- Foss T, Stensrud E, Kitchenham B, Myrteit I (2003) A simulation study of the model evaluation criterion MMRE. *IEEE Trans Softw Eng* 29:985–995. doi:[10.1109/TSE.2003.1245300](https://doi.org/10.1109/TSE.2003.1245300)
- Gray AR, Macdonell SG (1997) A comparison of techniques for developing predictive models of software metrics. *Inf Softw Technol* 39:425–437. doi:[10.1016/S0950-5849\(96\)00006-7](https://doi.org/10.1016/S0950-5849(96)00006-7)
- Guh RS (2002) Robustness of the neural network based control chart pattern recognition system to non-normality. *Int J Qual Reliab Manage* 19(1):97–112. doi:[10.1108/02656710210415749](https://doi.org/10.1108/02656710210415749)
- Hagan MT, Demuth HB, Beale MH (1997) *Neural network design*. PWS, Boston, MA
- Hardy RL (1971) Multiquadratic equations of topography and other irregular surfaces. *J Geophys Res* 76:1905–1915. doi:[10.1029/JB076i008p01905](https://doi.org/10.1029/JB076i008p01905)
- Heiat A (2002) Comparison of artificial neural network and regression models for estimating software development effort. *Inf Softw Technol* 44:911–922. doi:[10.1016/S0950-5849\(02\)00128-3](https://doi.org/10.1016/S0950-5849(02)00128-3)
- Huang SJ, Chiu NH (2006) Optimization of analogy weights by genetic algorithm for software effort estimation. *Inf Softw Technol* 48:1034–1045. doi:[10.1016/j.infsof.2005.12.020](https://doi.org/10.1016/j.infsof.2005.12.020)
- ISBSG (2007a) *International software benchmark and standard group*, Data CD Release 10, [www.isbsg.org](http://www.isbsg.org), 2007
- ISBSG (2007b) *Guidelines for use of ISBSE data*, available from web link: <http://www.isbsg.org/isbsg.nsf/weben/Repository%20info>

- Jeffery R, Ruhe M, Wiczorek I (2001) Using public domain metrics to estimate software development effort. Proceedings Seventh International Software Metrics Symposium, 16–27
- Jonsson P, Wohlin C (2006) Benchmarking k-nearest neighbour imputation with homogeneous Likert data. *Empir Softw Eng* 11:463–489. doi:10.1007/s10664-006-9001-9
- Jorgensen M (1995) An empirical study of software maintenance tasks. *J Softw Mainten* 7:27–48. doi:10.1002/smr.4360070104
- Jorgensen M (2004) A review of studies on expert estimation of software development effort. *J Syst Softw* 70:37–60. doi:10.1016/S0164-1212(02)00156-5
- Jorgensen M (2005) Evidence-based guidelines for assessment of software development cost uncertainty. *IEEE Trans Softw Eng* 31:942–954. doi:10.1109/TSE.2005.128
- Jorgensen M (2007) Forecasting of software development work effort: evidence on expert judgement and formal models. *Int J Forecast* 23(3):449–462. doi:10.1016/j.ijforecast.2007.05.008
- Jorgensen M, Indahl U, Sjoberg D (2003) Software effort estimation by analogy and “regression toward the mean”. *J Syst Softw* 68:253–262. doi:10.1016/S0164-1212(03)00066-9
- Jun ES, Lee JK (2001) Quasi-optimal case-selective neural network model for software effort estimation. *Expert Syst Appl* 21:1–14. doi:10.1016/S0957-4174(01)00021-5
- Kendall M, Stuart A (1976) The advanced theory of statistics, 4th Edition, Vol. I. Griffin, London
- Kirsopp C, Mendes E, Premraj R, Shepperd M (2003) An empirical analysis of linear adaptation techniques for case-based prediction. *ICCBR* 2003:231–245
- Kitchenham BA, Pickard LM, MacDonell SG, Shepperd MJ (2001) What accuracy statistics really measure. *IEE Proc Softw* 148(3):81–85. doi:10.1049/ip-sen:20010506
- Kolodner JL (1993) Case-Based Reasoning. Kaufmann
- Lawrence J (1994) Introduction to neural networks: Design, theory, and applications. California Scientific Software, Nevada City, CA
- Lee JK, Lee N (2006) Least modification principle for case-based reasoning: a software project planning experience. *Expert Syst Appl* 30:190–202. doi:10.1016/j.eswa.2005.06.021
- Li JZ, Ruhe G (2008) Analysis of attribute weighting heuristic for analogy-based software effort estimation method AQUA+. *Empir Softw Eng* 13(1):63–96. doi:10.1007/s10664-007-9054-4
- Li JZ, Ruhe G, Al-Emran A, Richter M (2007) A flexible method for software effort estimation by analogy. *Empir Softw Eng* 12(1):65–106. doi:10.1007/s10664-006-7552-4
- Li YF, Xie M, Goh TN (2008a) A study of project selection and feature weighting for analogy based software cost estimation. *J Syst Softw* (in press). Accepted Manuscript
- Li YF, Xie M, Goh TN (2008b) A study of mutual information based feature selection for case based reasoning in software cost estimation. *Expert Syst Appl* (in press). Accepted Manuscript
- Liu Q, Mintram RC (2005) Preliminary data analysis methods in software estimation. *Softw Qual J* 13:91–115. doi:10.1007/s11219-004-5262-y
- Mair C, Kadoda G, Lefley M, Phalp K, Schofield C, Shepperd M, Webster S (2000) An investigation of machine learning based prediction systems. *J Syst Softw* 53:23–29. doi:10.1016/S0164-1212(00)00005-4
- Maxwell K (2002) Applied statistics for software managers. Englewood Cliffs, NJ, Prentice-Hall
- Mendes E, Mosley N, Counsell S (2001) Web metrics—Estimating design and authoring effort. *IEEE Multimedia*, Special Issue on Web Engineering, 50–57
- Mendes E, Watson I, Triggs C, Mosley N, Counsell S (2003) A comparative study of cost estimation models for web hypermedia applications. *Empir Softw Eng* 8:163–196. doi:10.1023/A:1023062629183
- Myrtveit I, Stensrud E (1999) A controlled experiment to assess the benefits of estimating with analogy and regression models. *IEEE Trans Softw Eng* 25(4):510–525. doi:10.1109/32.799947
- Myrtveit I, Stensrud E, Olsson UH (2001) Analyzing data sets with missing data: an empirical evaluation of imputation methods and likelihood-based methods. *IEEE Trans Softw Eng* 27:999–1013. doi:10.1109/32.965340
- Myrtveit I, Stensrud E, Shepperd M (2005) Reliability and validity in comparative studies of software prediction models. *IEEE Trans Softw Eng* 31(5):380–391. doi:10.1109/TSE.2005.58
- Pendharkar PC, Subramanian GH, Rodger JA (2005) A probabilistic model for predicting software development effort. *IEEE Trans Softw Eng* 31(7):615–624. doi:10.1109/TSE.2005.75
- Pickard L, Kitchenham B, Linkman S (2001) Using simulated data sets to compare data analysis techniques used for software cost modeling. *IEE Proc Softw* 148(6):165–174. doi:10.1049/ip-sen:20010621
- Sentas P, Angelis L (2006) Categorical missing data imputation for software cost estimation by multinomial logistic regression. *J Syst Softw* 79(3):404–414. doi:10.1016/j.jss.2005.02.026
- Sentas P, Angelis L, Stamellos I, Bleris G (2005) Software productivity and effort prediction with ordinal regression. *Inf Softw Technol* 47:17–29. doi:10.1016/j.infsof.2004.05.001

- Shepperd M, Schofield C (1997) Estimating software project effort using analogies. *IEEE Trans Softw Eng* 23:736–743. doi:10.1109/32.637387
- Shepperd M, Kadoda G (2001) Comparing software prediction techniques using simulation. *IEEE Trans Softw Eng* 27(11):1014–1022. doi:10.1109/32.965341
- Song QB, Shepperd M (2007) A new imputation method for small software project data sets. *J Syst Softw* 80:51–62. doi:10.1016/j.jss.2006.05.003
- Srinivasan R, Fisher D (1995) Machine learning approaches to estimating software development effort. *IEEE Trans Softw Eng* 21(2):126–137. doi:10.1109/32.345828
- Stensrud E (2001) Alternative approaches to effort prediction of ERP projects. *Inf Softw Technol* 43(7):413–423. doi:10.1016/S0950-5849(01)00147-1
- Stensrud E, Foss T, Kitchenham B, Myrvtveit I (2003) A further empirical investigation of the relationship between MRE and project size. *Empir Softw Eng* 8(2):139–161. doi:10.1023/A:1023010612345
- Strike K, El-Emam K, Madhavji N (2001) Software cost estimation with incomplete data. *IEEE Trans Softw Eng* 27(10):890–908. doi:10.1109/32.962560
- Van Koten C, Gray AR (2006) Bayesian statistical effort prediction models for data-centred 4GL software development. *Inf Softw Technol* 48:1056–1067. doi:10.1016/j.infsof.2006.01.001
- Vapnik V (1995) *The nature of statistical learning theory*. Springer, New York
- Walkerden F, Jeffery DR (1999) An empirical study of analogy-based software effort Estimation. *Empir Softw Eng* 4(2):135–158. doi:10.1023/A:1009872202035



**Yanfu Li** received his B.Eng. degree in Software Engineering in 2005 from Wuhan University in China. He is currently a Ph.D. candidate in Department of Industrial & Systems Engineering at National University of Singapore. His research interests include software cost estimation, artificial intelligence, software reliability and quality engineering, and simulation optimization. He has publications on *Journal of Systems and Software*, *Expert Systems with Applications*, and several international conferences. He is a student member of IEEE.





**Min Xie** received his Ph.D. in Quality Technology in 1987 from Linkoping University in Sweden. He was awarded the prestigious LKY research fellowship in 1991 and currently he is a Professor at National University of Singapore. Prof Xie has authored or co-authored numerous papers and six books on quality and reliability engineering, including *Software Reliability Modelling* in 1991 by World Scientific Publisher, *Weibull Models* by John Wiley in 2003, and *Computing Systems Reliability* by Kluwer Academic in 2004. He is an Editor of *Int Journal of Reliability, Quality and Safety Engineering*, Department Editor of *IIE Transactions*, Associate Editor of *IEEE Transactions on Reliability*, and on the editorial board of a number of other international journals. Prof Xie is an elected fellow of IEEE.



**T N Goh** holds a BE from the University of Saskatchewan, Canada and a PhD from the University of Wisconsin-Madison. Positions that he has held include Dean, Faculty of Engineering; Head, Department of Industrial and Systems Engineering; Director, NUS Office of Quality Management, and CEO, Design Technology Institute International. Dr Goh is an elected Academician of the International Academy for Quality, Fellow of the American Society for Quality (ASQ), and Honorary Member of the Singapore Quality Institute. He is author or co-author of more than a hundred research papers and three books and currently serves on the editorial boards of eight international research journals.