Adaptive ridge regression system for software cost estimating on multi-collinear datasets

Yan-Fu Li\textsuperscript{a,b}, Min Xie\textsuperscript{b}, Thong-Ngee Goh\textsuperscript{b}

\textsuperscript{a} University of Tennessee, USA
\textsuperscript{b} National University of Singapore, Singapore

Abstract

Cost estimation is one of the most critical activities in software life cycle. In past decades, a number of techniques have been proposed for cost estimation. Linear regression is yet the most frequently applied method in the literature. However, a number of studies point out that linear regression is prone to low prediction accuracy. The low prediction accuracy is due to a number of reasons such as non-linearity and non-normality. One less addressed reason is the multi-collinearities which may lead to unstable regression coefficients. On the other hand, it has been reported that multi-collinearity spreads widely across the software engineering datasets. To tackle this problem and improve regression's accuracy, we propose a holistic problem-solving approach (named adaptive ridge regression system) integrating data transformation, multi-collinearity diagnosis, ridge regression technique and multi-objective optimization. The proposed system is tested on two real world datasets with the comparisons with OLS regression, step-wise regression and other machine learning methods. The results indicate that adaptive ridge regression system can significantly improve the performance of regressions on multi-collinear datasets and produce more explainable results than machine learning methods.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

The production of low cost and high quality software in a short time has become the ultimate goal of software industry. To achieve this goal, the software development processes need to be well managed and controlled. One of the most important activities is that of estimating the cost devoted to build the software. This task is known as Software Cost Estimation (Boehm, 1981). Aiming at accurate estimation, several techniques have been published in the past decades. Such as expert judgment (Jorgensen, 2004; Gruschke and Jorgensen, 2008), parametric models (Boehm, 1981; Albrecht and Gaffney, 1983; Puntam and Myers, 1991), machine learning methods (Shepperd and Schofield, 1997; Heiat, 2002; Pendaraxkar et al., 2005; Kumar et al., 2008; Keung et al., 2008; Mendes and Mosley, 2008; Li et al., 2009a,b) and linear regression methods (Miyazaki et al., 1994; Costagliola et al., 2005; Berlin et al., 2009; Huang et al., 2008).

According to a recent overview conducted by Jorgensen and Shepperd (2007), the linear regression is yet the most frequently applied method in the cost estimation literature. A large number of studies used linear regression methods (especially the OLS regression) as a benchmark against their proposed methods and concluded that regressions have lower accuracies than the new methods. However, as Kitchenham and Mendes (2009) recently pointed out that, these conclusions might be biased due to the inappropriate use of regression methods. Some restrictions of linear regressions such as homoscedasticity, moderate outliers, and normal errors might be ignored by the users. Among all restrictions, independence of the explanatory variables (or project features) is a less addressed one in the cost estimation literature. This restriction is violated when the multi-collinearity phenomenon appears. Multi-collinearity means two or more explanatory variables in a multiple regression model are highly correlated in linear form and multi-collinearity often causes linear regression losing stability and effectiveness (Kutner et al., 2005).

On the other hand, the inter-correlated explanatory variables (or project features) widely spreads across the software engineering datasets (Shepperd and Kadoda, 2001; Mendes et al., 2003). For example, the 'number of lines of source code' is often regarded to be positively related to the 'number of inputs', 'number of outputs', etc. In literature, there are some techniques to alleviate the effects of multi-collinearity: such as getting additional or dropping some explanatory variables (Neter et al., 1983). However, it is either difficult to get appropriate variables out from large amount of project features such as the ISBSG provided (ISBSG, 2007), or...
time-consuming to gather the information needed to create a new variable. On the other hand, dropping some variables might lead to only a few explanatory variables in the regression equation. Too few variables will reduce the explainability of the regression equation and make it vulnerable to the changing collinearities among explanatory variables.

Ridge regression technique (Hoerl and Kennard, 1970) provides a good alternative to deal with multi-collinearity which makes full use of the existing data and avoids adding or dropping explanatory variables. It also holds the potential (via ridge parameter) to improve the fitting and prediction accuracy of the regression methods. As ridge regression implements the central idea of the regularization theory (Tikhonov and Arsenin, 1977), some studies (Agarwal et al., 2007; Yu and Liong, 2007) have pointed out that it can achieve equally good or even better prediction performances than some machine learning methods such as support vector machine and artificial neural networks. In addition, the ridge regression equation is more transparent than the black-box liked machine learning methods especially neural networks.

Ridge regression has been successfully applied in a number of research areas such as biology (Goeman, 2008), environmental science (Hessami et al., 2008), hydrology (Chokmani et al., 2008), and nuclear science (Zhou et al., 2001). More recently, ridge regression has been introduced for the estimation issues in software engineering. Nguyen et al. (2008) applied ridge regression technique to estimate the constrained coefficients for the COCOMO models. Papadopoulos et al. (2009) utilized ridge regression to generate confidence intervals for effort estimation. Parsa et al. (2008) applied ridge regression to produce classification scores to filter out the redundant features. However, none of the previous studies have been focused on using ridge regression to resolve the multi-collinearity problem in the software cost dataset.

Based on ridge regression, we propose a novel problem-solving approach, namely adaptive ridge regression (ARR) system, combining different techniques for cost estimation on multi-collinear datasets. The ARR system consists of data transformation, multi-collinearity diagnosis, ridge regression technique and a multi-objective optimization to train the ridge parameter. The rest of this paper is organized as follows: Section 2 introduces multi-collinearity diagnosis and ridge regression; Section 3 describes the proposed adaptive ridge regression (ARR) system; Section 4 presents the real world datasets and procedures for the empirical validation; Section 5 describes the experimental results and comparisons; Section 6 presents the threads to validity; the last section summarizes this work and points out possible future directions.

2. Multi-collinearity diagnosis and ridge regression

Prior to the detailed description of multi-collinearity, the multiple linear regression equation is presented. In general, a multiple linear regression equation has the following form:

\[ y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon \]  

where \( y \) denotes the dependent variable, \( x_i, i = 1, \ldots, p \) stands for the \( i \)th explanatory variable, \( \beta_i, i = 1, \ldots, p \) is the \( i \)th regression coefficient, \( \beta_0 \) is the intercept, and the error term \( \epsilon \) is a random noise following a standard normal distribution.

2.1. Multi-collinearity and diagnosis of multi-collinearity

Multi-collinearity is a statistical phenomenon in which two or more explanatory variables in a multiple regression model are highly correlated in linear form. When the explanatory variables are highly correlated, the ordinary least square (OLS) estimates are likely to be unstable or even with a wrong sign. The reason is that OLS estimates involve inverting the matrix \( XX \)

\[
X = \begin{bmatrix}
1 & x_{1,1} & \cdots & x_{p,1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{1,n} & \cdots & x_{p,n}
\end{bmatrix}
\]  

and \( XX \) is rank deficient and not invertible when serious multi-collinearity presents. In the matrix \( X \), \( n \) is the total number of observations, and \( (p - 1) \) is the total number of explanatory variables.

For example, suppose that \( y \) has the underlying function with \( X \) in the form \( y = 1 + 2x_1 + 3x_2 \), \( x_1 \) and \( x_2 \) are perfectly related by \( x_1 = 2x_2 \), and the regression equation takes the form \( y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon \). Then infinite sets of \( \beta_0, \beta_1, \beta_2 \) can be obtained by using the relation \( x_1 = 2x_2 \). As a result, the underlying function can be written into infinite number of equivalent forms, such as \( y = 1 + 7x_2, y = 1 + 3.5x_1 \), and \( y = 1 + x_1 + 5x_2 \).

To quantify the severity of the multi-collinearity, several diagnosis indicators have been proposed. Generally, they are divided into two groups (Brauner and Shacham, 1998): the indicators based merely on explanatory variables (e.g. Variance Inflation Factor (VIF)), and the indicators based on both explanatory variables and the dependent variable (e.g. confidence intervals (Brauner and Shacham, 1998)). Diagnosis using only the explanatory variables is frequently employed by the researchers because it distinguishes collinearity from the prediction error (in the dependent variable). In addition, VIF is also a popular criterion for ridge regression parameter selection (which will be discussed in Section 2.2) (Belsley, 1991). Hence, in this study we mainly focus on VIF as the indicator of multi-collinearity.

The Variance Inflation Factor (VIF) measures how much the variance of an estimated regression coefficient is increased because of collinearity. It is mathematically defined as:

\[
\text{VIF}_i = \frac{1}{1 - R_i^2}
\]

\[
R_i^2 = \frac{\sum_{j=1}^{n}(\bar{x}_{ij} - \bar{x}_i)^2}{\sum_{j=1}^{n}(x_{ij} - \bar{x}_i)^2}
\]

where \( R_i^2 \) is the coefficient of the determination when the \( i \)th explanatory variable is regressed on the remaining explanatory variables. \( x_{ij} \) is the \( j \)th observed value of the \( i \)th explanatory variable, \( \bar{x}_i \) is the calculated value of \( x_i \) when it is regressed on the other explanatory variables, and \( \bar{x}_i \) is the mean value of the observed values of the \( i \)th explanatory variable. If the \( i \)th variable is not linearly related to the remaining variables, then \( R_i^2 \) is close to 0 and VIF is close to 1, while if the \( i \)th variable is linearly related to some subset of the remaining variables, \( R_i^2 \) is close to 1 and the VIF is greater than 1.

As discussed above, VIF appears to be a good indicator of the severity of the multi-collinearity. Thus several researchers (Chatterjee and Price, 1977; Neter et al., 1983) have used VIF for ridge regression parameter selection (one important application of the diagnosis indicator is to guide the selection of the ridge regression parameter). This process involves varying the ridge regression parameter until all the VIFs approach 1. However, the problem of this method is that the VIFs for different variables may not all approach 1 simultaneously.

Alternatively, the mean value of the VIFs (MVIF) is often regarded as a better diagnosis indicator (Kutner et al., 2005), because it avoids the problem faced by using all VIFs and it also provides the information about the severity of multi-collinearity in terms of how far the estimated regression coefficients are from the true values of the coefficients. The mean value of the VIFs (MVIF) is
defined as following:

$$\text{MVIF} = \frac{\sum_{j=1}^{p-1} \text{VIF}_j}{p - 1}$$

(4)

The expected value of the sum of the squared errors (which reflects how far the estimated coefficients are from the true coefficients) is given as below:

$$E[\sum_{j=1}^{p-1} (\hat{\beta}_i - \beta_i)^2] = \sum_{j=1}^{p-1} \text{var}(\hat{\beta}_j) = \sigma^2 \sum_{j=1}^{p-1} \text{VIF}_j = \sigma^2 \cdot \text{MVIF} \cdot (p - 1)$$

(5)

where $\sigma^2$ is the variance of the dependent variable. Eq. (5) shows that MVIF is positively related to the expectation of the sum of the squared errors. According to the discussions above, this study uses MVIF to diagnose the severity of multi-collinearity and to guide the selection of ridge regression parameter.

2.2. Ridge regression

For the ease of presentation, we rewrite regression Eq. (1) into the following matrix form:

$$y = X\beta + e$$

(6)

where $y$ is a $(n \times 1)$ vector of the observation values of the dependent variable, $X$ is a $(n \times p)$ matrix of the $p$ explanatory variables or $p - 1$ explanatory variables with the first column being a column vector of 1s as the regression intercept, $\beta$ is a $(p \times 1)$ vector of the unknown regression coefficients, and $e$ is a $(n \times 1)$ vector of normally distributed errors with $E(e) = 0$ and $\text{Var}(e) = \sigma^2 I_n$ where $I_n$ is a $(n \times n)$ identity matrix.

The ordinary least square (OLS) regression is the fundamental method in the regression family. Though OLS regression is one of the oldest methods for cost estimation, it is still being continuously improved or frequently applied by many researchers (Angelis et al., 2001; Sentas et al., 2005; Jeffery et al., 2000; Jorgensen, 2004; Mendes et al., 2005; Costagliola et al., 2005; Huang et al., 2008; Berlin et al., 2009; Li et al., 2009a). The least square estimate of $\beta$ is usually given by:

$$\hat{\beta} = (XX')^{-1}X'y$$

(7)

The two key properties of $\hat{\beta}$ are (1) it is unbiased, which is $E(\hat{\beta}) = \beta$ and (2) it has the minimum variance among all linear unbiased estimators, which is assured by Gauss–Markov theorem. But there is no guarantee that this variance will be small. The total sum of the variance of each element in $\hat{\beta}$ can be written as:

$$\text{Var}(\hat{\beta}) = \text{Trace}\{\sigma^2(X'X)^{-1}\} = \sigma^2 \sum_{i=1}^{p} \frac{1}{\lambda_i}$$

(8)

where $\lambda_i$ s are the eigenvalues of the matrix $XX'$ (if $X$ is normalized then $XX'$ is the so-called covariance matrix). It is noted that a small value of $\lambda_i$ will lead to a large value of $\text{Var}(\hat{\beta})$.

Ridge regression was first proposed by Hoerl and Kennard (1970) to deal with the multi-collinearity. This method simply modifies the OLS by adding one number onto the diagonal of the covariance matrix $XX'$. Consequently, the calculated parameters are biased but nevertheless more stable than those of the OLS regression.

Mathematically, ridge regression parameter estimation has the following form:

$$\hat{\beta}(k) = (XX' + kI_p)^{-1}X'y, \quad k > 0$$

(9)

$\hat{\beta}(k)$ in (9) is obtained by minimizing the combination of both the sum of square error and the norm of the $\hat{\beta}$ vector:

$$\min(y - X\hat{\beta})(y - X\hat{\beta}) + k\hat{\beta}'\hat{\beta}$$

(10)

The norm $\hat{\beta}'\hat{\beta}$ is the length of the vector $\hat{\beta}$ in a vector space. The purpose of adding one norm term to the sum of square errors is to restrict the values of $\hat{\beta}$ into a certain boundary. It is noted that the methodology of minimizing Eq. (10) is also referred as ‘regularization’ (Tikhonov and Arsenin, 1977) in the literature of statistical learning. Many modern machine learning techniques such as SVM (Vapnik, 1995) and RBF (Hardy, 1971) are based on the regularization theory (Evgeniou et al., 2000).

As mentioned in the Section 2.1, multi-collinearity often brings large variances to the OLS estimations. In ridge regression, the ridge parameter $k$ provides the flexibility to reduce the coefficients variances. The mathematical form of ridge regression’s coefficients variances is presented as follows:

$$\text{Var}[\hat{\beta}(k)] = \text{Trace}\{\sigma^2(X'X + kI_p)^{-1}(X'X)(X'X + kI_p)\}$$

$$= \sigma^2 \sum_{i=1}^{p} \frac{\lambda_i}{(\lambda_i + k)^2}$$

(11)

Eq. (11) shows that the variance will decrease as $k$ becomes large. On the other hand, a larger value of $k$ will result to more biased coefficients, as shown in Eq. (12):

$$\text{Bias}[\hat{\beta}(k)] = E[\hat{\beta}(k)] - \beta = k^2 \beta'(XX' + kI_p)^{-2}\beta$$

(12)

The biasness and variance together define the mean square error (MSE) of the ridge regression parameters $\hat{\beta}(k)$:

$$\text{MSE}[\hat{\beta}(k)] = \text{bias}[\hat{\beta}(k)]^2 + \text{var}[\hat{\beta}(k)]$$

$$= k^2 \beta'(XX' + kI_p)^{-2}\beta + \sigma^2 \sum_{i=1}^{p} \frac{\lambda_i}{(\lambda_i + k)^2}$$

(13)

Similar to the MSE of ridge regression coefficients, the MSE of OLS regression coefficients can be obtained as follows:

$$\text{MSE}[\hat{\beta}(k)]_{\text{OLS}} = \text{bias}[\hat{\beta}]^2 + \text{var}[\hat{\beta}] = \sigma^2 \sum_{i=1}^{p} \frac{1}{\lambda_i}$$

(14)

Hoerl and Kennard (1970) have proved that there always exist a set of $k$ values that make the ridge regression coefficients smaller than the MSE of OLS regression coefficients:

$$\text{MSE}[\hat{\beta}(k)]_{\text{RR}} < \text{MSE}[\hat{\beta}]_{\text{OLS}}.$$ A graphic illustration of this conclusion is shown in Fig. 1. The above discussions indicate that ridge regression, by reducing the effects of multi-collinearity, is able to generate more stable coefficients with smaller mean square error than the OLS does. Thus, how to select the appropriate $k$ values is critical to the success of ridge regression.

In the literature, several selection criteria have been proposed. Ridge trace (Hoerl and Kennard, 1970), Variance Inflation Factor (VIF) (Belsley, 1991) and Generalized Cross-Validation (GCV) (Golub et al., 1979) are the most frequently applied criteria. In this study, we employ VIF for parameter selection because it is a useful indicator for multi-collinearity diagnosis (see Section 2.1) while ridge trace and GCV do not show direct relation to the multi-collinearity property.

Besides ridge regression, multi-collinearity may be also alleviated by getting additional explanatory variables or dropping some explanatory variables. The stepwise regression (Neter et al., 1983)
accuracy metrics are introduced. Prior to the details about ARR system, different regression parameter which can maximize the accuracy and minimize the multi-collinearity. This equation may generate poor predictions when the explanatory variables are more independent from each other. In this case, more explanatory variables are expected to be added into the regression equation.

3. Adaptive ridge regression for cost estimation

Based upon the concepts of ridge regression and the indicator of multi-collinearity diagnosis, this section proposes the adaptive ridge regression (ARR) system for cost estimation. Different from the previous studies on ridge regression, this system includes a multi-objective optimization problem to search for the ridge regression parameter which can maximize the accuracy and minimize the multi-collinearity. Prior to the details about ARR system, the accuracy metrics are introduced.

3.1. Accuracy metrics

Accuracy evaluation criteria are essential to the experiments. In the cost estimation literature, several quality metrics have been proposed to assess the performances of estimation methods. Among them, mean magnitude of relative error (MMRE) (Conte et al., 1986), median magnitude of relative error (MdMRE) (Jorgensen, 1995), and percentage of predictions (PRED) (Conte et al., 1986) are three popular metrics.

The MMRE is defined as below:

\[
\text{MMRE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|
\]

where \( n \) denotes the total number of projects, \( y_i \) denotes the actual cost of project \( i \), and \( \hat{y}_i \) 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 (Foss et al., 2003). Nevertheless, MMRE has been the de facto standard in the software cost estimation literature.

The MdMRE (Kitchenham et al., 2001) is the median of all the MRES:

\[
\text{MdMRE} = \text{median} \left( \left| \frac{y_i - \hat{y}_i}{y_i} \right| \right)
\]

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 is the percentage of predictions that fall within a specified percent of the actual cost.

\[
PRED(q) = \frac{k}{n}
\]

where \( n \) denotes the total number of projects and \( k \) represents the number of projects whose MRE is less than or equal to \( q \). In our paper, \( q \) is set to be 0.25.

3.2. Adaptive ridge regression

The ARR system consists of four stages: data transformation, multi-collinearity diagnosis, training, and predicting (shown in Fig. 2). The following paragraphs provide the details of each stage.

3.2.1. Data transformation

The data transformation stage includes the conversion of the categorical explanatory variables into dummy variables, the standardization of the numerical variables, and the transformation of the categorical variable.

Categorical explanatory variables (e.g. ‘language’ and ‘application type’) often exist in cost estimation datasets. However, conventional regressions cannot directly make use of the categorical variables with more than two categories. In literature, dummy variables representation (Hardy, 1993; Kutner et al., 2005; Wissmann et al., 2007; Papadopoulos et al., 2009) has been regarded as the de facto approach to transform categorical variables for regressions. The dummy variable approach uses a set of dichotomous variables to represent one categorical variable. A dummy variable is defined as:

\[
D_j = \begin{cases} 
1 & \text{if in category } j \\
0 & \text{else}
\end{cases}
\]

where \( j \) denotes the total number of categories, and \( D_j \) represents the presence or absence of a category.

A categorical variable with \( m \) categories is represented by \((m - 1)\) dummy variables. The reference/baseline category is where all dummy variables are equal to zero (Hardy, 1993). For instance, a categorical variable ‘Language’ with three categories (‘C++’, ‘JAVA’, ‘C#’) can be represented by two dummy variables:

\[
D_1 = \begin{cases} 
1 & \text{if } C++ \\
0 & \text{else}
\end{cases}
\]

\[
D_2 = \begin{cases} 
1 & \text{if } JAVA \\
0 & \text{else}
\end{cases}
\]

where the condition \( D_1 = 0 \) and \( D_2 = 0 \) represent the category that ‘Language’ = (‘C#’).

In cost estimation, standardization is a necessary step to normalize numerical variables (Angelis and Stamelos, 2000; Li et al., 2009a). This makes the values of a transformed explanatory variable into the interval [0,1], where zero is assigned to the minimum observed value and one is assigned to the maximum observed value.

The dependent variable transformation is also crucial for linear regression techniques (Kitchenham and Mendes, 2009), because the normality and homoscedasticity assumptions are not valid in
many cases, and the relationships between project features and project costs are often non-linear in nature (Boehm, 1981). Various kinds of transformation techniques have been proposed, such as logarithm transformation, square root transformation, and reciprocal transformation. To select an appropriate transformation under some given conditions is a difficult task (Ryan, 1997). Box and Cox (1964) proposed a method to determine the appropriate transformation in a way that the transformed dependent variable $y$ is a linear function of the explanatory variables with a normally distributed constant error variance. The mathematically form of Box–Cox transformation is presented as follows:

$$y^{(\lambda)} = \begin{cases} 
\frac{y^{\lambda} - 1}{\lambda} & \lambda \neq 0 \\
\log(y) & \lambda = 0 
\end{cases}$$

(20)

where $y$ is the original dependent variable, $y^{(\lambda)}$ is the transformed dependent variable, and $\lambda$ is the transformation parameter. $\lambda$ is often determined by maximizing the likelihood function given by Draper and Smith (1981)

$$L(\lambda) = -\frac{n}{2} \log \left( \frac{SSE}{n} \right) + (\lambda - 1) \sum \log(y)$$

(21)

where SSE is the total variance of $y$. After the data transformation, the system enters the multi-collinearity diagnosis stage.

3.2.2. Multi-collinearity diagnosis

As discussed in Section 2.1, the Mean Variance Inflation Factor (MVIF) is used for multi-collinearity diagnosis. At this stage, the MVIF value of the training sample data $X_{\text{train}}$ is first calculated (using OLS regression). If MVIF value is larger than 1, then the multi-collinearity is regarded to be serious and the system will enter the ridge regression loop to optimize the ridge parameter $k$ and obtain the final regression equation. If MVIF is not larger than 1, the system will perform least square estimation to obtain the regression equation. The threshold value ‘1’ is commonly used as an indicator that the multi-collinearity may be influencing the least square estimates (Kutner et al., 2005).
3.2.3. Training

There are two different training schemes at this stage: one is based on least square estimation, and the other is based on ridge regression. The decision on performing which training scheme is made in the multi-collinearity diagnosis stage. It is shown that the training scheme under the condition ‘MVIF ≤ 1’ is relatively straightforward (by applying OLS estimation). Thus this section will focus on the training scheme under the condition ‘MVIF > 1’ which indicates serious multi-collinearity in the training dataset. The goal of this training scheme is to obtain the appropriate ridge parameter \( k \) to minimize the effects of multi-collinearity and maximize the fitting accuracy. To achieve this goal, we formulize a multi-objective optimization problem.

\[
\begin{align*}
\min \quad & \left[ \text{MMRE}(k) - \text{PRED}_{0.25}(k), \frac{\text{MVIF}(k)}{\text{MVIF}(0)} \right] ^T \\
\text{s.t.} \quad & 0 \leq k \leq U_k
\end{align*}
\]

(22)

where the first objective function represents the training accuracy, the second objective function represents the degree of multi-collinearity and \( U_k \) in the constraint is the upper limit of \( k \). It is noted that the metrics MMRE, PRED(0.25) and MVIF are all associated with the ridge parameter \( k \). In the first objective, the accuracy metrics MMRE and PRED(0.25) are combined together because they are widely accepted for cost estimation and Kitchenham et al. (2001) demonstrated that MMRE and PRED(0.25) are used to predict multi-collinearity, but MVIF is the indicator MVIF is normalized by dividing it with the largest MVIF value: MVIF(0) which is obtained by using ridge regression with \( k = 0 \) (or OLS regression).

To solve this multi-objective problem, we adopt the well-known weight aggregation method (Gass and Saaty, 1955) because it is effective, efficient, and easy to be understood by practitioners. In addition, the weight aggregation method is the foundation of many other optimization techniques such as normal-boundary intersection (Das and Dennis, 1998) and normal constraint (Messa et al., 2003).

The weight aggregation method associates a weight to each objective function and minimizes the weighted linear sum of the objectives. By multiplying a positive coefficient \( \alpha \) to the second objective, the problem in (22) can be rewritten as follows:

\[
\begin{align*}
\min \quad & \text{MMRE}(k) - \text{PRED}_{0.25}(k) + \alpha \times \frac{\text{MVIF}(k)}{\text{MVIF}(0)} \\
\text{s.t.} \quad & 0 \leq k \leq U_k
\end{align*}
\]

(23)

where the coefficient \( \alpha \) controls the weights of training accuracy and multi-collinearity. Coefficient \( \alpha \) can be provided by the decision manager or obtained from historical information.

Because it is difficult to obtain the closed form of the objective in (23), the following algorithm is proposed to search for optimal \( k \) within a predefined range (see the gray portion of Fig. 2).

1. Specify the weighting coefficient \( \alpha \), the upper boundary \( U_k \), and the incremental step of \( \Delta \). Set \( k = \Delta \).
2. Compute the regression coefficient \( \hat{\beta}(k) = (X'X + kI)^{-1}Xy \) given the \( k \) value.
3. Compute the output of regression model \( \hat{y} = \hat{\beta}(k)X \) and use reverse Box–Cox transformation to obtain the final output.
4. Calculate the object value in (22) given the final output of the regression model.
5. Increase \( k \) by a small amount \( \Delta \): \( k = k + \Delta \).
6. Repeat steps (2–5) until \( k \) reaches its upper limit.
7. Select the optimal \( k_{opt} \) that minimize the object and the corresponding regression coefficient \( \hat{\beta}(k_{opt}) \).

Once the optimal solution is obtained, the training phase is completed and the system is ready to make predictions.

3.2.4. Predicting

In this stage, the ARR system receives new data for predicting. The following procedures are performed:

1. Standardize the explanatory variables.
2. Compute the output of regression model \( \hat{y} = \hat{\beta}(k_{opt})X \) with the optimized regression coefficient.
3. Apply the reversed Box–Cox transformation to obtain final prediction.

4. Experiment set-up

In this section, we describe the structured experiment procedures on real world cost estimation datasets.

4.1. Data sets

Two representative public datasets are selected for experiments. They are Ablrecht dataset (Ablrecht and Gaffney, 1983) and Desharneis dataset (Desharnais, 1989). The analysis in the following part of this section indicates that these datasets have serious multi-collinearity problems.

The Ablrecht dataset is a popular dataset utilized by many recent studies (Shepperd and Schofield, 1997; Heiat, 2002; Auer et al., 2006; Li et al., 2009a,b). 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 six explanatory variables: ‘Inpcount’, ‘Outcount’, ‘Quecount’, ‘Filcount’, ‘SLOC’ and ‘Fp’, and one dependent variable ‘Effort’. The ‘Effort’ is recorded in 1000 personnel hours. The detailed descriptions of the variables are shown in Appendix A. The descriptive statistics is presented in Table 1. Among these statistics, the ‘Skewness’ and ‘Kurtosis’ are used to quantify the degree of non-normality of the features, ‘VIF’ is the indicator of multi-collinearity among the explanatory variables. In summary, Ablrecht is a relatively small dataset with high order multi-collinearities and non-normality.

Desharnais dataset is another popular public datasets and it has been frequently used by recent studies, such as Mair et al. (2000), Burgess and Lefley (2001), Auer et al. (2006) and Li et al. (2009a,b).

<table>
<thead>
<tr>
<th>Features</th>
<th>Mean</th>
<th>Std. dev</th>
<th>Min</th>
<th>Max</th>
<th>Skewness</th>
<th>Kurtosis</th>
<th>VIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inpcount</td>
<td>40.25</td>
<td>36.91</td>
<td>7.00</td>
<td>193.00</td>
<td>3.07</td>
<td>13.44</td>
<td>4.74</td>
</tr>
<tr>
<td>Outcount</td>
<td>47.25</td>
<td>35.17</td>
<td>12.00</td>
<td>150.00</td>
<td>1.28</td>
<td>4.29</td>
<td>9.78</td>
</tr>
<tr>
<td>Quecount</td>
<td>17.38</td>
<td>15.52</td>
<td>3.00</td>
<td>60.00</td>
<td>1.40</td>
<td>3.96</td>
<td>5.71</td>
</tr>
<tr>
<td>Filcount</td>
<td>16.88</td>
<td>19.34</td>
<td>0</td>
<td>75.00</td>
<td>1.94</td>
<td>6.46</td>
<td>3.82</td>
</tr>
<tr>
<td>Fp</td>
<td>61.08</td>
<td>63.68</td>
<td>3.00</td>
<td>318.00</td>
<td>2.90</td>
<td>12.19</td>
<td>7.21</td>
</tr>
<tr>
<td>SLOC</td>
<td>647.63</td>
<td>488.00</td>
<td>195.00</td>
<td>1902.00</td>
<td>1.44</td>
<td>4.02</td>
<td>39.31</td>
</tr>
<tr>
<td>Effort</td>
<td>21.88</td>
<td>28.42</td>
<td>0.50</td>
<td>105.20</td>
<td>2.16</td>
<td>6.51</td>
<td>NA</td>
</tr>
</tbody>
</table>
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 six numerical explanatory variables are ‘TeamExp’, ‘ManagerExp’, ‘Length’, ‘Transactions’, ‘Adjust’, and ‘PointsAdjust’. One categorical explanatory variable is ‘Language’. The dependent variable ‘Effort’ is recorded in 1000 h. The descriptions of the features are summarized in Appendix B. The descriptive statistics of the features are provided in Table 2. It is worth noting that in Table 2 the categorical explanatory variable ‘Language’ has been replaced by two dummy variables: ‘Language1’ and ‘Language2’. Table 2 shows that Desharnais dataset contains categorical feature, and has high order multi-collinearity and moderate non-normality.

‘Language1’ and ‘Language2’ are the dummy variables representing the categorical variable ‘Language’ in the original dataset.

### 4.2. Methods specifications

In this section, other cost estimation methods used for the comparative studies on the real world datasets are introduced. Two regression methods and three machine learning techniques are chosen. The regression methods include OLS and stepwise regression, the stepwise regression (SWR) is based on an important assumption that some explanatory variables in a multiple regression do not have an important explanatory effect on the dependent variable. Usually, stepwise procedure takes the form of a sequence of F-tests (or other tests such as t-tests). The stepwise regression approaches include: (1) forward selection, which involves starting with no variables in the model, trying out the variables one by one and including them if they are statistically significant; and (2) backward selection, which starts with all candidate variables and testing them one by one for statistical significance, deleting any variables that are not significant. In our case, we employed the forward selection strategy. For more details about stepwise regression, readers can refer to Neter et al. (1983).

The machine learning methods include artificial neural network (ANN) (Heiat, 2002), Case Based Reasoning (CBR) (Shepperd and Schofield, 1997), and Classification and Regression Trees (CART) (Pickard et al., 2001).

Artificial neural network (ANN) plays an important role in solving problems with difficult or unknown analytical solutions. In the ANN model, the number of hidden layers, the number of hidden nodes and the type transfer functions are the three predefined parameters and they have a major impact on the prediction performance of ANN (Martin et al., 1997). Among these parameters, one hidden layer structure is often recommended since multiple hidden layers may lead to an over-parameterized ANN structure. Thus, one hidden layer is utilized in this study. Additionally, the search spaces for the number of hidden neurons and hidden layer transfer functions are set to be 1, 3, 5, 7, 9, 10 and {Linear, Tan-Sigmoid, Log-Sigmoid} respectively. During the training process, the ANN models with different parameter combinations are trained on the historical dataset. Then, all ANN structures are implemented on the training set and the one producing the lowest MMRE–PRED value is selected for the comparisons against other models.

The Case Based Reasoning (CBR) (Kolodner, 1993) approach was firstly proposed by Shepperd and Schofield (1997) as a valid alternative to expert judgment and algorithmic method for software cost estimations. The procedure of CBR algorithm is simple but effective: given a new project for estimation, the most similar projects are retrieved out from historical dataset to derive the cost of the new project. Generally, there are three alternative parameters for CBR method: the similarity function, the number of most similar projects (analogy), and the analogy adaptation (Angelis and Stamelos, 2000). In line with the common settings of these parameters, we define the search spaces for similarity function as {Euclidean distance, Manhattan distance}. K number of similar projects as {1, 2, 3, 4, 5}, and solution functions as {Closest Analogy, Mean, Median, Inverse Distance Weighted Mean} respectively.

The CART is a non-parametric and tree structured analysis procedure that can be used for classification and regression. The construction of the CART involves recursively splitting the data set into (usually two) relatively homogeneous subsets until the terminate conditions are satisfied. CART is often treated as a non-parametric method, however we consider one parameter in this study: the depth of the tree (l = 1, 2, 3, 4, 5). The l is controlled in the range from 1 to 5, to balance modeling accuracy and overfitting. All the methods are implemented via MATLAB. The best variants of machine learning methods are obtained by training these methods and tuning their parameters following the procedures presented in Section 4.3.

### 4.3. Experiment procedure

The predefined parameters (such as the ridge parameter in ANN, depth of tree in CART) have large impacts on the cost estimation performances, it is possible that a good parameter combination of method ‘A’ can generate a better prediction than method ‘B’ with the poor parameter setting even if ‘B’ might be generally more accurate than ‘A’. Therefore, a systematic selection of the predefined parameters should be included in the experiments. A simple but effective way for parameter selection is the so-called holdout validation scheme. Under this scheme, the entire dataset is randomly split (often 2/3 to 1/3 split) into two mutually exclusive subsets: training subset (2/3 split) and testing subset (1/3 split). The training subset is used to construct models and select parameters. The testing subset is used to test the trained methods with optimal parameters to evaluate the prediction performances which reflect their abilities to generalize to unknown conditions. The above validation scheme is performed 10 times on each dataset to eliminate the biasness from different splits, similarly to Briand et al. (1999), Jeffery et al. (2001), and Mendes et al. (2003).

After determining the validation scheme, the following procedures are executed on each data split:

### Table 2

Descriptive statistics of numeric features of Desharnais dataset.

<table>
<thead>
<tr>
<th>Features</th>
<th>Mean</th>
<th>Std. dev</th>
<th>Min</th>
<th>Max</th>
<th>Skewness</th>
<th>Kurtosis</th>
<th>VIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>TeamExp</td>
<td>2.30</td>
<td>1.33</td>
<td>0.00</td>
<td>4.00</td>
<td>−0.05</td>
<td>1.73</td>
<td>1.66</td>
</tr>
<tr>
<td>ManagerExp</td>
<td>2.65</td>
<td>1.52</td>
<td>0.00</td>
<td>7.00</td>
<td>0.22</td>
<td>3.01</td>
<td>1.43</td>
</tr>
<tr>
<td>Length</td>
<td>11.30</td>
<td>6.79</td>
<td>1.00</td>
<td>36.00</td>
<td>1.43</td>
<td>5.49</td>
<td>2.72</td>
</tr>
<tr>
<td>Transactions</td>
<td>177.47</td>
<td>146.08</td>
<td>9.00</td>
<td>886.00</td>
<td>2.34</td>
<td>10.09</td>
<td>47.59</td>
</tr>
<tr>
<td>Entities</td>
<td>120.55</td>
<td>86.11</td>
<td>7.00</td>
<td>387.00</td>
<td>1.36</td>
<td>4.37</td>
<td>15.71</td>
</tr>
<tr>
<td>Adjust</td>
<td>27.45</td>
<td>10.53</td>
<td>5.00</td>
<td>52.00</td>
<td>−0.19</td>
<td>2.58</td>
<td>1.65</td>
</tr>
<tr>
<td>PointsAdjust</td>
<td>284.71</td>
<td>187.10</td>
<td>62.00</td>
<td>1116.00</td>
<td>1.65</td>
<td>6.89</td>
<td>75.59</td>
</tr>
<tr>
<td>Language1*</td>
<td>0.57</td>
<td>0.50</td>
<td>0.00</td>
<td>1.00</td>
<td>−0.29</td>
<td>1.08</td>
<td>2.26</td>
</tr>
<tr>
<td>Language2*</td>
<td>0.30</td>
<td>0.46</td>
<td>0.00</td>
<td>1.00</td>
<td>0.88</td>
<td>1.77</td>
<td>2.03</td>
</tr>
<tr>
<td>Effort</td>
<td>4.83</td>
<td>4.189</td>
<td>0.55</td>
<td>23.94</td>
<td>2.00</td>
<td>7.89</td>
<td>NA</td>
</tr>
</tbody>
</table>
Table 3
Overall averages and standard deviations of error values of all metrics.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Error metrics</th>
<th>MMRE</th>
<th>PRED (0.25)</th>
<th>MdMRE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
<td>Std</td>
</tr>
<tr>
<td>ARR</td>
<td>0.36</td>
<td>0.05</td>
<td>0.44</td>
<td>0.11</td>
</tr>
<tr>
<td>SWR</td>
<td>0.43</td>
<td>0.07</td>
<td>0.33</td>
<td>0.06</td>
</tr>
<tr>
<td>OLS</td>
<td>0.63</td>
<td>0.44</td>
<td>0.23</td>
<td>0.16</td>
</tr>
<tr>
<td>ANN</td>
<td>0.90</td>
<td>0.27</td>
<td>0.21</td>
<td>0.08</td>
</tr>
<tr>
<td>CBR</td>
<td>0.58</td>
<td>0.20</td>
<td>0.24</td>
<td>0.09</td>
</tr>
</tbody>
</table>

1. ARR is performed on training sets and the ridge parameter is optimized on training set by the procedures described in Section 3.2 with the weighting coefficient $\alpha = 1$, the upper limit $U_k = 20$, and the incremental $\Delta = 0.1$. The regression equation with optimal ridge parameter is obtained and tested on testing datasets.

2. OLS and SWR are performed on training set. The regression equations are then tested on testing datasets.

3. ANN, CART, and CBR are trained and optimized on training set. The optimal models are tested on the testing datasets.

4. To evaluate the prediction performance of ARR, the results of all methods on testing datasets are compared and analyzed.

The results and analysis are presented in Section 5.

5. Results and comparisons

5.1. Results on Albrecht dataset

Table 3 summarizes the testing results across of all 10 experiments. The table shows that ARR achieves the best average MMRE, average PRED(0.25) and average MdMRE. In terms of standard deviation of error metrics, ARR has the smallest std. of MMRE, the fifth smallest std. of PRED(0.25) and the third smallest std. of MdMRE.

To further analyze the error metrics, we draw out the box plots of MMRE, PRED(0.25), and MdMRE in Fig. 3.

The boxplots show that ARR has best medians, shortest interquartiles range among all methods. Although boxplots are useful graphical tool of visually comparing predictions, they cannot statistically confirm whether one technique is significantly better than another. Therefore, we perform the significant tests on the error values.

Because the boxplots show that some result distributions are heavily skewed thus do not satisfy the requirements of the classical t-tests, the assumption free test Wilcoxon sign-rank test (significance level $\alpha = 0.05$ with two tails) is performed to quantify the statistical significance of the comparisons. The $p$-values of the Wilcoxon tests are presented in Table 4. The entry in the table contains the $p$-value of the paired test between ARR and the methods in the row. We can tell from the table that ARR performs significantly better than all other methods with $p$-values smaller than 0.05, expect for two cases where ARR is statistically equal to ANN under PRED(0.25) and MdMRE, respectively.

5.2. Results on Desharnais dataset

Table 5 summarizes the testing performances across 10 experiments. The table shows that ARR achieves the best average MMRE, average PRED(0.25) and average MdMRE. In terms of standard deviation of the error metrics, ARR has the smallest std. on MMRE and MdMRE.
Table 5
Overall averages and standard deviations of error values of all metrics.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Error metrics</th>
<th>MMRE</th>
<th>PRED (0.25)</th>
<th>MdMRE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
<td>Std</td>
</tr>
<tr>
<td>ARR</td>
<td>0.37</td>
<td>0.08</td>
<td>0.45</td>
<td>0.08</td>
</tr>
<tr>
<td>SWR</td>
<td>0.68</td>
<td>0.29</td>
<td>0.32</td>
<td>0.09</td>
</tr>
<tr>
<td>OLS</td>
<td>0.47</td>
<td>0.12</td>
<td>0.36</td>
<td>0.07</td>
</tr>
<tr>
<td>ANN</td>
<td>0.52</td>
<td>0.15</td>
<td>0.36</td>
<td>0.13</td>
</tr>
<tr>
<td>CART</td>
<td>0.92</td>
<td>0.25</td>
<td>0.28</td>
<td>0.07</td>
</tr>
<tr>
<td>CBR</td>
<td>0.80</td>
<td>0.16</td>
<td>0.35</td>
<td>0.11</td>
</tr>
</tbody>
</table>

Table 6
Results of Wilcoxon sign-rank tests.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MMRE</th>
<th>PRED(0.25)</th>
<th>MdMRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWR</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>OLS</td>
<td>0.01</td>
<td>0.01</td>
<td>0.03</td>
</tr>
<tr>
<td>ANN</td>
<td>0.01</td>
<td>0.11</td>
<td>0.13</td>
</tr>
<tr>
<td>CART</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>CBR</td>
<td>0.00</td>
<td>0.09</td>
<td>0.11</td>
</tr>
</tbody>
</table>

To further analyze the error metrics, box plots of MMRE, PRED(0.25), and MdMRE are drawn out in Fig. 4. The graphs show that ARR has best medians, shortest inter-quartiles range in the plots of MMRE. ARR has the highest median in PRED(0.25) plot and MdMRE plot. We then perform the significant tests on the error values to quantitatively evaluate differences between the methods.

The Wilcoxon sign-rank test (significance level $\alpha = 0.05$ with two tails) is used to test the paired methods. The $p$-values of the Wilcoxon tests are given in Table 6. We can tell from the table that ARR performs significantly better than all other methods under MMRE. For PRED(0.25) and MdMRE, ARR is better than SWR, OLS and CART, and equal to ANN and CBR. These findings imply that ARR could achieve better predictions than other regression methods and equally good predictions to the machine learning methods on Desharnais dataset.

6. Discussions

This section discusses the threats to validity, and the theoretical and practical implications of this study.

6.1. Threats to validity

The threats to validity include the threads to internal validity and the threads to external validity. The internal validity represents to what extent conclusions can be drawn about the causal effect of the explanatory variables on the dependent variable (e.g. project cost). In this study, the threats to internal validity include the following aspects: first the dummy variable representation is utilized for the transformation of categorical variable. However, this approach may lead to an over-parameterized regression equation (e.g. the number of regression coefficients is larger than the number of observations) when the categorical variables contain large numbers of categories. Therefore, under these situations necessary steps (e.g. hierarchical discriminant analysis (Lachenbruch, 1975)) have to be taken to reduce the number of dummy variables.

Secondly, the weight aggregation method is used to solve the multi-objective optimization problem to obtain the optimal ridge parameter. However this method is essentially subjective, where a decision manager needs to supply the weighting coefficient $\alpha$. To further improve ARR system, the objective way of solving multi-objective problems is worth to investigate. The objective approach requires a Pareto-compliant ranking method, as seen in current multi-objective evolutionary approaches such as NSGA-II and SPEA2 (Deb, 2001), where no weight is required.

The external validity represents the degree to which the findings of our study can be generalized to environments outside of our research settings. The threats to external validity are presented as

![Fig. 4. Boxplots of error metric values on Desharnais dataset.](image-url)
follows. First, the limitations on real world datasets make some difficulties to generalize our findings. Although the two datasets used in our experiments contrast to each other, additional real world datasets are required for a more comprehensive comparisons. In addition, missing data is a very common phenomenon in software engineering datasets. Many studies (Myrtev 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. But our study is mainly focused on solving the multi-collinearity problems. This might cause some difficulties to generalize our results to the datasets with missing values. However, simple ways (such as listwise deletion and variable deletion) could be adopted to deal with missing data before data reduction and data projection.

6.2. Implications

This work aims to improve regression’s performance on the multi-collinear software cost datasets. To achieve this goal, we proposed a holistic approach (ARR) combining data transformation, multi-collinearity diagnosis, ridge regression and multi-objective optimization. The results of this study reveal the theoretical and practical values of the proposed ARR system. Theoretically speaking, to the best of our knowledge this is the first study that introduce ridge regression to systematically resolve multi-collinearity in cost estimation. Furthermore, the estimation accuracy is also considered in ARR system. With the multi-objective optimization technique, ARR is capable of maximizing estimation accuracy and minimizing multi-collinearity with certain predetermined balance between these two criteria.

The empirical results show that ARR system is capable of achieving accurate cost estimation. In practice, this is very important for software enterprises to maintain a better control of the budget through their software development processes. Moreover, the final regression equation of ARR is transparent to the practitioners. This is one valuable advantage that the regression methods often possess, as comprehensible models provide the opportunity to discover the causalities beneath the experiment results.

7. Summary and future works

Linear regression is the most frequently applied method for cost estimation. A number of studies pointed out that linear regression is prone to low prediction accuracy. One less addressed reason for the low prediction accuracy is the multi-collinearity. To tackle this problem and improve regression accuracy, we propose a holistic approach: adaptive ridge regression system (ARR), which consists of data transformation, multi-collinearity diagnosis, ridge regression technique and a multi-objective optimization to train the ridge parameter.

The proposed system is validated on two real world datasets with the comparisons to OLS, SWR, ANN, CART, and CBR. The comparisons against OLS and SWR illustrate ARR’s advantages on data fitting and prediction. The comparisons against ANN, CART and CBR reveal that ridge regression can achieve equally good or even better predictions than the machine learning methods and ridge regression has better interpretations than machine learning methods. The results show that ridge regression could be a promising improvement of the regressions in the context of cost estimation. However, this study also has some limitations. First, the optimization of ridge parameter can be further improved, as the objective function only includes MMRE, PRED(0.25) and MVIF. Other type of error metrics (e.g. MdMRE) or multi-collinearity indicators may be considered to guide the optimal search. Additionally, to solve the multi-objective problem, more advanced optimization algorithms can be considered such as evolutionary algorithms (Deb, 2001) and normal constraint algorithm (Messac et al., 2003).

Acknowledgement

This research was partially supported by a grant from A*Star (SERC grant number 072 1340050) in Singapore.

Appendix A. Appendix A. Variable definitions of Albrecht dataset.

<table>
<thead>
<tr>
<th>Features</th>
<th>Full name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inpcount</td>
<td>Input count</td>
<td>Numerical</td>
<td>Count of inputs</td>
</tr>
<tr>
<td>Outcount</td>
<td>Output count</td>
<td>Numerical</td>
<td>Count of outputs</td>
</tr>
<tr>
<td>Queccount</td>
<td>Query count</td>
<td>Numerical</td>
<td>Count of queries</td>
</tr>
<tr>
<td>Filecount</td>
<td>File count</td>
<td>Numerical</td>
<td>Count of files</td>
</tr>
<tr>
<td>Fp</td>
<td>Function points</td>
<td>Numerical</td>
<td>Number of function points</td>
</tr>
<tr>
<td>SLOC</td>
<td>Lines of source code</td>
<td>Numerical</td>
<td>Lines of source code</td>
</tr>
<tr>
<td>Effort</td>
<td>Development effort</td>
<td>Numerical</td>
<td>Measured in 1000h</td>
</tr>
</tbody>
</table>

Appendix B. Appendix B. Variable definitions of Desharnais dataset

<table>
<thead>
<tr>
<th>Features</th>
<th>Full name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TeamExp</td>
<td>Team experience</td>
<td>Numerical</td>
<td>Measured in years</td>
</tr>
<tr>
<td>ManagerExp</td>
<td>Manager’s experience</td>
<td>Numerical</td>
<td>Measured in years</td>
</tr>
<tr>
<td>Length</td>
<td>Length of project</td>
<td>Numerical</td>
<td>Actual project schedule in months</td>
</tr>
<tr>
<td>Transactions</td>
<td>Transactions</td>
<td>Numerical</td>
<td>Count of basic logical transactions in the system</td>
</tr>
<tr>
<td>Entities</td>
<td>Entities</td>
<td>Numerical</td>
<td>Number of entities in the systems data model</td>
</tr>
<tr>
<td>Adjust</td>
<td>Adjustment</td>
<td>Numerical</td>
<td>Function point complexity adjustment factor (Total Processing Complexity)</td>
</tr>
<tr>
<td>PointsAdjust</td>
<td>Adjusted function points</td>
<td>Numerical</td>
<td>This is the function points adjusted by the adjustment factor</td>
</tr>
<tr>
<td>Language</td>
<td>Language</td>
<td>Categorical</td>
<td>This is the type of major programming language used in the project</td>
</tr>
<tr>
<td>Effort</td>
<td>Development effort</td>
<td>Numerical</td>
<td>Measured in 1000h</td>
</tr>
</tbody>
</table>

References


Yanfu Li received his PhD in 2010 from Department of Industrial & Systems Engineering at National University of Singapore. He is currently a Research Associate at University of Tennessee, USA. His research interests include software cost estimation, data mining, software reliability and quality engineering, and cloud computing. He has publications on Journal of Systems and Software, Empirical Software Engineering, IEEE Transactions on Reliability, Expert Systems with Applications, Applied Software Computing and several international conferences. He is an invited reviewer of six international journals. Dr. Li is a member of IEEE.
**Min Xie** received his PhD 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 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.