2014 IEEE 38th Annual International Computers, Software and Applications Conference

A Mutual Information-Based Hybrid Feature Selection Method for Software Cost Estimation Using Feature Clustering

Qin Liu, Shihai Shi, Hongming Zhu*, Jiakai Xiao
School of Software Engineering
Tongji University
Shanghai, China
qin.liu@tongji.edu.cn myshishihai@gmail.com hongming.zhu@gmail.com waitingxjk@gmail.com

***Abstract* —Feature selection methods are designed to obtain the optimal feature subset from the original features to give the most accurate prediction. So far, supervised and unsupervised feature selection methods have been discussed and developed separately. However, these two methods can be combined together as a hybrid feature selection method for some data sets. In this paper, we propose a mutual information-based (MI-based) hybrid feature selection method using feature clustering. In the unsupervised learning stage, the original features are grouped into several clusters based on the feature similarity to each other with agglomerative hierarchical clustering. Then in the supervised learning stage, the feature in each cluster that can maximize the feature similarity with the response feature which represents the class label is selected as the representative feature. These representative features compose the feature subset. Our contribution includes 1)the newly proposed feature selection method and 2)the application of feature clustering for software cost estimation. The proposed method employs wrapper approaches, so it can evaluate the prediction performance of each feature subset to determine the optimal one. The experimental results in software cost estimation demonstrate that the proposed method can outperform at least 11.5% and 14.8% than the supervised feature selection method INMIFS and mRMRFS in ISBSG R8 and Desharnais data set in terms of PRED (0.25) value.**

Keywords: software cost estimation; feature selection; mutual information; feature clustering

I. INTRODUCTION

Feature selection is a procedure to select an optimal feature subset from the original feature set. It needs to eliminate the redundant and irrelevant features, and keep the informative ones. Feature selection methods have been widely used in pattern recognition, machine learning and data mining because it is an effective solution to the "curse of dimensionality" problem. Feature selection not only provides a better understanding of the underlying process that generates data but also helps predictors do more accurate and cost-effective prediction[1].

Feature selection methods can be classified into supervised and unsupervised methods based on whether response feature is available or not[2]. Supervised feature selection methods using mutual information have been studied in[3-7]. They take both feature relevance and feature redundancy into consideration. The feature relevance can be defined as the feature similarity between candidate feature and the response feature while the feature redundancy refers to the feature similarity between candidate feature and the selected feature subset. However, the exact value of feature redundancy is not easy to calculate because it involves the calculation of high-dimensional mutual information. The proposed supervised feature selection methods in [5-7] approximate it by averaging the feature similarity between candidate feature and each feature in the selected feature subset, which may lead to estimation error and is unable to eliminate feature redundancy. Another concern of above supervised feature selection methods is the "nesting effect"[8]. Since the features that are selected into the feature subset will not be eliminated, the next selected feature will be affected by those selected ones. The "nesting effect" in feature selection always provides a sub-optimal solution instead of the optimal one.

Recently unsupervised feature selection methods using feature clustering have been proposed to overcome the disadvantage in those supervised feature selection methods. Feature clustering is one of the popular approaches to deal with high dimensional data in unsupervised feature selection methods. The original features are clustered based on the feature similarity and one representative feature will be selected from each cluster to form the best discriminative feature subset. The redundant features are grouped together and eliminated. Therefore feature clustering can handle feature redundancy quite well. As far as we’ve concerned, little work has been done on applying this kind of feature selection methods for software cost estimation.

An unsupervised feature selection method named FSA is proposed in[9]. FSA uses symmetric uncertainty[10] as the measurement of feature similarity to do feature clustering and predefines the number of selected features in the final

0730-3157/14 $31.00 © 2014 IEEE
DOI 10.1109/COMPSAC.2014.99

IEEE computer society

feature subset. However, the number of selected features always differs when using different data set. The predefined number cannot guarantee to produce the optimal feature subset. Another unsupervised feature selection method named FSFC is proposed in[11]. FSFC applies MICI[12] as the measurement of feature similarity and also predefines the number of selected features in the final feature subset. So it has the same drawback as FSA. Another drawback of both FSA and FSFC is the selection of representative features in the cluster. When choosing the representative features in the cluster, both methods do not take the response feature into consideration. So the final feature subset may be irrelevant to the response feature, which will not do help to prediction at all. In order to solve this problem, A new feature selection method is proposed in[13]. It employs conditional mutual information and hierarchical clustering to group the features and then selects the feature which has the highest value of mutual information with the response feature as the representative feature. So it employs both unsupervised learning and supervised learning in the feature selection method.

The data set used in the software cost estimation contains many features including the response feature. Therefore, our proposed method takes advantage of supervised and unsupervised feature selection methods to obtain a hybrid feature selection method. In the unsupervised learning stage, the original features are grouped into several clusters based on the feature similarity without using the response feature. Then in the supervised stage, the feature which is most relevant with the response feature will be selected from each cluster as the representative feature. These representative features form the optimal feature subset. The proposed method guarantees that the selected features are relevant to the response feature and they are not redundant features to each other.

Feature selection methods can be divided into three categories: filter, wrapper and embedded. The filter methods evaluate the goodness of the feature set by using the intrinsic characteristic of the data. It is relatively general and computationally cheap since it does not involve the particular prediction problem. Wrapper approach has to evaluate the feature subset with some criteria of particular prediction problem. Therefore, it is more specific and computationally expensive. The wrapper methods use a specific classifier to assess the feature subsets. So they usually outperform the filter in terms of prediction accuracy but the computational complexity is large. The embedded methods try to include the feature selection as a part of the classifier training process. The proposed method is a wrapper method.

The rest of this paper is organized as follows. Section 2 gives the background of mutual information and software cost estimation with case based reasoning. In Section 3, the hybrid feature selection method is introduced. Experimental settings and result analysis are presented in Section 4. Finally, the conclusion is drawn in Section 5.

II. Preliminaries

A. Entropy and Mutual Information

Entropy is an essential concept in the information theory. It was introduced into information theory by Shannon[14] as the measure of uncertainty of a random variable.

Let X be a random variable with discrete values, and $p(x)$ is the probability density function, then the entropy of X is defined as:

$$\mathrm{H(X)} = -\sum_{x\epsilon S_x} p(x) log P(x)$$

Joint entropy is a measure of the uncertainty associated with two random variables X and Y. $p(x,y)$ is the joint probability density function and the joint entropy can be defined as:

$$\mathrm{H(X,Y)} = -\sum_{x\epsilon S_x}\sum_{y\epsilon S_y} p(x,y) log\, p(x,y)$$

Conditional entropy is used to measure the remaining uncertainty of a random variable X when the entropy value of another variable Y is given.

$$\mathrm{H(X|Y)} = -\sum_{x\epsilon S_x}\sum_{y\epsilon S_y} p(x,y) log\, p(x|y)$$

Mutual information is a quantity that measures the mutual dependence of two random variables X and Y. If X and Y are discrete random variables, the mutual information is as follows:

$$\mathrm{I(X;Y)} = \sum_{y\epsilon S_y}\sum_{x\epsilon S_x} p(x,y) log \frac{p(x,y)}{p(x)p(y)}$$

Otherwise, if X and Y are continuous random variables, the mutual information is defined as:

$$\mathrm{I(X;Y)} = \int_{S_y}\int_{S_x} p(x,y) log \frac{p(x,y)}{p(x)p(y)} dx\, dy$$

The value of $\mathrm{I(X;Y)}$ is higher, the two random variables X and Y are more relevant. If X and Y are independent to each other, then $\mathrm{I(X;Y)}$=0.

The motivation to consider the mutual information in software cost estimation is its capability of measuring arbitrary relations between features and it does not depend on transformations acting on different features[3].

B. Case Based Reasoning

Software cost estimation is one of the most crucial processes in software development management because it involves many management activities such as project planning, resource allocation and risk assessment.

Software cost estimation using case based reasoning is proposed by Shepperd and Schofield[15]. Case based reasoning is the process to select one or more completed software projects that are similar to the new project from the historical software projects and then derive the new project's cost from the selected projects. There are three major stages to build an estimation model when using case based reasoning: feature selection, case selection and case adaptation. Feature selection is to select a best informative feature subset from the original features. Case selection refers to use selected feature subset in the feature selection stage to calculate the global distance between two projects and select one or more projects that are most similar to the new project. Case adaptation makes use of the selected projects in case selection stage to do the estimation for the new project. Therefore, it can be seen that the feature selection is of great importance in the case based reasoning. The quality of the selected feature subset can determine the performance of the estimation model.

C. Project Similarity Measurement

The project similarity measurement in case selection stage in this study is the Euclidean distance. The Euclidean distance between two projects i and j can be written as follows:

$$D_{ij} = \sqrt{\sum_{k=1}^{k=n} w_k Dis(f_{ik}, f_{jk})}$$

$$\mathrm{Dis}(f_{ik}, f_{jk}) = \begin{cases} (f_{ik} - f_{jk})^2, If\ f_{ik} and\ f_{jk}\ are\ numeric \\ 1, If\ f_{ik}, f_{jk}\ are\ nominal\ and\ f_{ik} \neq\ f_{jk} \\ 0, If\ f_{ik}, f_{jk}\ are\ nominal\ and\ f_{ik} =\ f_{jk} \end{cases}$$

where f_{ik}, f_{jk} means the value of kth feature in project i and j, respectively. The feature weight w_k here is either 1 or 0. When the value of w_k is 1, it means that the kth feature is selected in the feature selection. Otherwise, the kth feature is discarded in the feature selection.

D. Evaluation Criteria

Evaluation criteria are necessary and important to assess the quality of cost estimation model. In this study we adopt the mean magnitude of relative error (MMRE), and PRED (0.25) as evaluation metrics because they are widely used in the software cost estimation[16].

The MMRE is defined as below:

$$\mathrm{MMRE} = \frac{1}{n}\sum_{i=1}^{n} MRE_i$$

$$MRE_i = \left|\frac{AE_i - EE_i}{AE_i}\right|$$

where n, AE_i and EE_i donate the number of projects, real cost of project i and estimated cost of project i, respectively.

The PRED (0.25) is the percentage of estimated effort that falls within 25% of the actual effort:

$$\mathrm{PRED}\ (0.25) = \frac{1}{n}\sum_{i=1}^{n}(MRE_i \leq 0.25)$$

III. Hybrid Feature Selection Method

A. Feature Similarity Measurement

Mutual information can be used as the feature similarity measurement but the normalized mutual information[6] proposed by Estévez et al is better. It restricts its value from 0 to 1. By employing the normalized mutual information, the well-known mutual information bias can be solved[6]. The definition of normalized mutual information is as follows:

$$\mathrm{NI}(f_i; f_j) = \frac{I(f_i;f_j)}{\min\{H(f_i), H(f_j)\}}$$

If the value of $\mathrm{NI}(f_i; f_j)$ is higher, the ith feature and the jth feature is more relevant.

B. Feature Clustering

All the features except the predicted variable are partitioned into different clusters based on the feature distance with hierarchical clustering in agglomerative mode. Here the feature distance in hierarchical clustering is defined as follows:

$$\mathrm{FDis}(f_i; f_j) = 1 - \mathrm{NI}(f_i; f_j)$$

In hierarchical clustering, two nearest clusters should be merged into one bigger cluster in each step until remaining only one cluster. Single link, complete link and group average are three common measurements to define the distance between two clusters[17]. In our study, complete link will act as the measurement of cluster distance for it is not sensitive to extreme values and outliers. It can be defined as follows:

$$\mathrm{CDis}(C_x; C_y) = \max\{\mathrm{FDis}(f_i; f_j), f_i \in C_x \text{ and } f_j \in C_y\}$$

where C_x and C_y are two clusters X and Y.

C. Number of Representative Feature K

Our proposed feature selection employs wrapper approaches. So our hybrid feature selection method has to evaluate the performance of each feature subset based on the criteria in software cost estimation in order to determine the optimal number of representative features K. The feature subset that can yield highest PRED(0.25) value will be the optimal feature subset and the K is the cardinality of that optimal feature subset.

D. Representative Feature Selection

The representative features will be selected from K clusters to compose the optimal feature subset. We need to recognize the top K clusters after hierarchical clustering. The order of top K clusters is a reversed order of hierarchical clustering sequence. The top 1 cluster is the cluster obtaining in the final step of the clustering sequence which contains all the features and the top 2 clusters are the clusters obtaining in the last 2 steps. Then, the criterion for representative feature selection is that the feature in each cluster that can maximize the feature similarity with the predicted variable will be selected. The representative feature selection procedure will iterate until all K features have been selected.

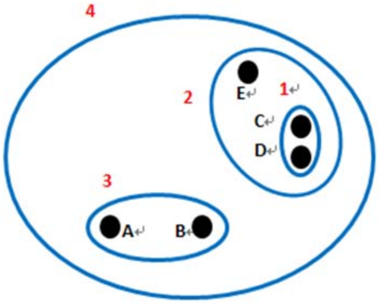

Figure 1. Hierarchical clustering result

Here we give an example of representative feature selection. After hierarchical clustering, we obtain the clustering result as shown in Figure 1. There are 4 clusters as drawn in 4 circles. Assume we need to select 2 representative features from the original feature set with 5 features A, B, C, D and E. The top 2 clusters are represented by the circle with number 4 and 3. Then, the first representative feature will be selected from cluster number 4, which contains all 5 features. The second representative feature will be selected from cluster number 3 with feature A and B. If the feature selected in the cluster number 4 is A, then the feature A cannot be selected again in the cluster number 3.

E. Hybrid Feature Selection Scheme

Our hybrid feature selection method combines unsupervised learning and supervised learning. In unsupervised learning stage, assume there are n features in the original feature set excluding the response feature. First the entire feature set is grouped into several clusters based on the cluster distance using hierarchical clustering without employing the response feature. In supervised learning stage, the K representative features, where K varies from 1 to n, are selected and the corresponding feature subset is evaluated. The feature subset that can yield the best performance of cost estimation model will be kept as the result of the hybrid feature selection method.

F. Computational Complexity Analysis

Assume that the original data set contains n features. It has computational complexity $O(n^2)$ in the filter approach for feature clustering and $O(n^2)$ in the wrapper approach for determining the optimal number of representative features. So the total complexity of HFSFC is $O(n^2) = O(n^2) + O(n^2)$.

However, there is still one limitation in this algorithm. If the data set contains n features, then we need to yield n feature subset and have to evaluate these subsets one by one to determine the best one as the final result.

```
Hybrid Feature Selection using Feature Clustering (HFSFC)
Input: Original feature set with n featuresF = $\{f_1, f_2, \ldots, f_n\}$, predicted variable $f_c$.
Output: the optimal feature subset S;
Step 1: S = $\emptyset$
        Calculate pair-wise feature distance
Step 2: $C_i = f_i$, each feature in F represents a cluster.
Step 3: Repeat
        Merge $C_i$ and $C_j$ if the their cluster distance is minimal
        Until all clusters are merged into one cluster.
Step 4: For K=1 to n
            Recognize the top K clusters $S_k$ from the hierarchical clustering result.
                $FS_k = \emptyset$
            For each cluster $C_x$ in $S_k$
                The unselected feature $f_x$ that can maxmize the feature similarity with the predicted variable $f_c$ is selected as a representative feature
                    $FS_k = FS_k \cup f_x$
            EndFor
            Evaluate the performance of subset $FS_k$
        EndFor
Step 5: The feature subset $FS_o$ that can achieve best performance is kept as the final result of the hybrid feature selection method
                S = $FS_o$
```

IV. EXPERIMENTS AND ANALYSIS

A. Data sets Description

ISBSG Release 8[18] and Desharnais[19] data set are two popular real world data sets made publicly available in order to encourage improvable predictive models in software cost estimation.

ISBSG data set contains 2008 samples with 61 features. It rates the reliability of historical project samples as 4 level from A to D where A represents the most reliable and useful data. We first have to do data cleaning as in[20] to obtain an refined data set. After eliminating the samples that containing missing values, there are 345 A-rated projects with 11 refined features remaining, namely "CouTech", "DevType", "FP", "Rlevel", "PrgLang", "DevPlatform","Time", "MethAcquired", "OrgType", "Method" and the response feature "SoftwareEffort".

Desharnais data set is much smaller than ISBSG R8 data set with 77 completed projects and 11 features. The features used in this study are “TeamExp”, “ManagerExp”, “YearEnd”, “Length”, “Language”, “Transactions”, “Entities”, “PointsNonAdjust”, “Adjustment”, “PointAdjust” and the predicted variable “SoftwareEffort”.

B. Experiment Settings

In order to make sure that each feature has the same degree influence in the project similarity measurement, all feature values except the predicted variable in the data set should be normalized into range (0, 1). When employing KNN to select the most similar projects in case selection, we choose K=3 because it can yield best performance according to our preliminary experiments.

In our experiments, we adopt 10-fold cross validation as the validation scheme. Projects in each testing set are randomly generated from the original data set and the remaining ones are treated as the training set. All of the experiments are conducted in R[21], which is an open source statistics tool for scientific calculation and data visualization.

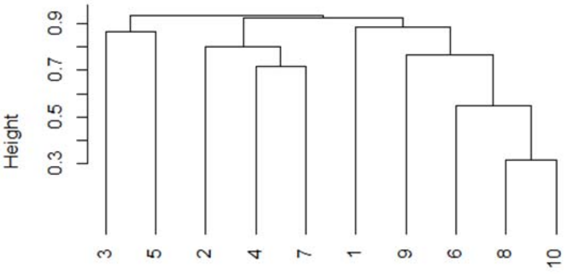

Figure 2. The hierarchical cluster tree of Desharnais data set

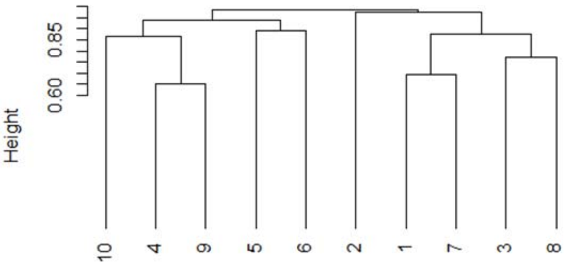

Figure 3. The hierarchical cluster tree of ISBSG R8 data set

C. Experiment Result Analysis

Our proposed HFSFC method will be compared with the mRMRFS[5] method, INMIFS[7] method and FSFC[11] method. Among these three feature selection methods, mRMRFS and INMIFS are supervised feature selection method while the FSFC is unsupervised feature selection method.

MMRE and PRED (0.25) are two important evaluation indices in software cost estimation. Smaller MMRE value indicates the lower level of estimation error[16]. The higher PRED (0.25) value implies better performance of estimating model because it means more estimated cost error falls within 25% of the real cost.

Table I. Experimental results on ISBSG R8 data set.

Feature Selection Method		MMRE	PRED (0.25)
Supervised Method	mRMRFS	1.6098	0.2285
	INMIFS	1.6383	0.2355
Unsupervised Method	FSFC	1.4660	0.2318
	HFSFC	**1.3938**	**0.2627**

Table II. Experimental results on Desharnais data set.

Feature Selection Method		MMRE	PRED (0.25)
Supervised Method	mRMRFS	0.4325	0.4400
	INMIFS	0.4644	0.4437
Unsupervised Method	FSFC	0.7425	0.3625
	HFSFC	**0.3406**	**0.5062**

In Figure 2 and Figure 3, it illustrates the hierarchical clustering results on two data sets by cluster tree. According to the HFSFC method, the optimal feature subset in Desharnais data set contains 2 features while in ISBSG R8 data set it contains 6 features. In Desharnais data set the features subset is {5,10}({“Language”, “PointAdjust”}) and in ISBSG R8 data set the feature subset is {2,3,5,6,7,9} ({“DevType”, “FP”, “PrgLang”, “DevPlatform”, “Time”, “OrgType”}).

Table I and Table II summarize the experiment results of four feature selection methods on two data sets. As shown in Table I, HFSFC method performs better than other three methods in terms of MMRE and PRED (0.25) value. The MMRE value of HFSFC is 14.9% lower than that of INMIFS method meanwhile PRED (0.25) value is 11.5% higher. The experimental results in Table II demonstrate that HFSFC method still has better performance over other three methods. It achieves 26.6% lower in MMRE value than that of INMIFS method as well as 14.1% higher in PRED (0.25) value.

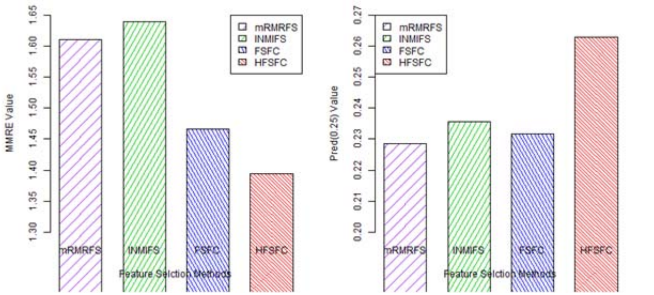

Figure 4. MMRE and PRED (0.25) value of each feature selection method in ISBSG R8 data set

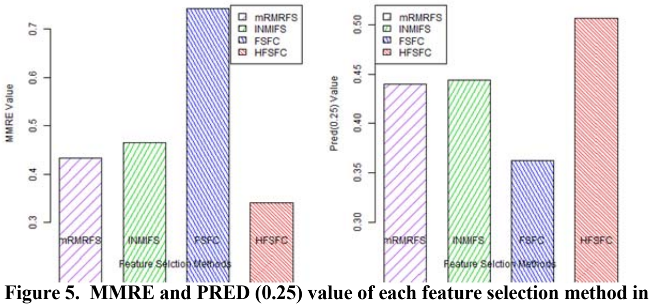

Figure 5. MMRE and PRED (0.25) value of each feature selection method in Desharnais data set

It is reasonable that HFSFC method yields better experimental results over other three methods for it takes full advantage of both supervised feature selection and unsupervised feature selection. The HFSFC can handle feature redundancy problem better than mRMRFS method and INMIFS method by using feature clustering. In addition, HFSFC method employs predicted variable to select the representative feature from each cluster, which is easier to pick out the feature subset that is more relevant to the predicted variable than the FSFC method.

V. Conclusions

In this paper, a hybrid feature selection method which makes use of the advantage of supervised feature selection and unsupervised feature selection method is proposed. It contains two stages. First, it employs feature clustering to partition the original features into several clusters. Then, a representative feature is selected from each cluster based on the feature similarity with the predicted variable. The feature subsets are made up of these representative features. Each feature subset will be evaluated with the direct goal to minimize the prediction error and the optimal feature subset will be kept as the final result of our proposed hybrid feature selection method. The experimental results confirm that our proposed method performs better over some supervised and unsupervised feature selection methods in two data sets for software cost estimation in terms of both MMRE and PRED (0.25) value.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (approval number is 61103006) and Fundamental Research Funds for the Central Universities.

References

[1] I. Guyon, and A. Elisseeff, “An introduction to variable and feature selection,” *The Journal of Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.

[2] L. Faivishevsky, and J. Goldberger, "Unsupervised feature selection based on non-parametric mutual information." pp. 1-6.

[3] R. Battiti, “Using mutual information for selecting features in supervised neural net learning,” *Neural Networks, IEEE Transactions on,* vol. 5, no. 4, pp. 537-550, 1994.

[4] N. Kwak, and C.-H. Choi, “Input feature selection for classification problems,” *Neural Networks, IEEE Transactions on,* vol. 13, no. 1, pp. 143-159, 2002.

[5] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 27, no. 8, pp. 1226-1238, 2005.

[6] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, “Normalized mutual information feature selection,” *Neural Networks, IEEE Transactions on,* vol. 20, no. 2, pp. 189-201, 2009.

[7] V. La The, T. Nguyen Duc, and L. Young-Koo, "An Improved Maximum Relevance and Minimum Redundancy Feature Selection Algorithm Based on Normalized Mutual Information." pp. 395-398.

[8] P. Pudil, J. Novovičová, and J. Kittler, “Floating search methods in feature selection,” *Pattern recognition letters,* vol. 15, no. 11, pp. 1119-1125, 1994.

[9] F. Zhang, Y.-J. Zhao, and J. Fen, "Unsupervised feature selection based on feature relevance." pp. 487-492.

[10] B. P. Flannery, W. H. Press, S. A. Teukolsky, and W. Vetterling, “Numerical recipes in C,” *Press Syndicate of the University of Cambridge, New York*, 1992.

[11] G. Li, X. Hu, X. Shen, X. Chen, and Z. Li, "A novel unsupervised feature selection method for bioinformatics data sets through feature clustering." pp. 41-47.

[12] P. Mitra, C. Murthy, and S. K. Pal, “Unsupervised feature selection using feature similarity,” *IEEE transactions on pattern analysis and machine intelligence,* vol. 24, no. 3, pp. 301-312, 2002.

[13] J. Martínez Sotoca, and F. Pla, “Supervised feature selection by clustering using conditional mutual information-based distances,” *Pattern Recognition,* vol. 43, no. 6, pp. 2068-2081, 2010.

[14] T. M. Cover, and J. A. Thomas, *Elements of information theory*: John Wiley & Sons, 2012.

[15] M. Shepperd, and C. Schofield, “Estimating software project effort using analogies,” *Software Engineering, IEEE Transactions on,* vol. 23, no. 11, pp. 736-743, 1997.

[16] Y. Li, M. Xie, and T. Goh, “A study of mutual information based feature selection for case based reasoning in software cost estimation,” *Expert Systems with Applications,* vol. 36, no. 3, pp. 5921-5931, 2009.

[17] P.-N. Tan, *Introduction to data mining*: Pearson Education India, 2007.

[18] ISBSG, “http://www.isbsg.org/.”

[19] Desharnais, “http://promise.site.uottawa.ca/SERepository/datasets/desharnais.arff.”

[20] Q. Liu, and R. Mintram, “Preliminary Data Analysis Methods in Software Estimation,” *Software Quality Journal,* vol. 13, no. 1, pp. 91-115, 2005/03/01, 2005.

[21] "R," http://www.r-project.org/.