Chapter 5

Hybridization of Fuzzy c-means with Random Forest, Adaptive Neuro fuzzy and Genetic algorithm for the prediction of defects

5.1 Introduction

In chapter 3 established the strengths and limitations of various clustering algorithms. This chapter also discussed clustering algorithms such as FCM and k-means which are good candidates for defect prediction. But the basic problem with all clustering algorithms is the lack of stability criterion, which means the ability to generate same partition of data irrespective of the order in which the patterns are presented to the algorithm.

To alleviate this issue, cascading clustering and classification approach have been established in chapter 4 with a view to accomplishing better performance. However, these clustering algorithms suffer from high computational time requirements. To mitigate this issue, in this chapter hybridization of clustering and classification is being proposed in order to reduce the computational complexity. Hybrid models are expected to improve the accuracy of clustering and classification.

It has been observed that hybridization in which more than one technique is combined is a promising approach. Increasing research activities focus on hybridization for improving the performance of a classifier. Fuzzy, random forest (RF), adoptive neuro fuzzy inference system (ANFIS) and genetic algorithm (GA) techniques are used in combination to improve the accuracy. This chapter is organized as follows; section 5.2 provides survey on hybrid model of RF, FCM and ANFIS. Section 5.3 includes collection of empirical data. Section 5.4 discusses a novel hybrid method of Random Forest (RF) and Fuzzy C Means (FCM) clustering for building defect prediction model. Experimental results have been provided. Section 5.5 presents the adaptive-neuro fuzzy c-means clustering for fault classification using fuzzy c-means clustering. Experimental studies are presented.. The contents of the chapter are summarized in section 5.6.

5.2 Collection of empirical data

An empirical investigation related to this work is carried on several product and service based software industries of varying production capabilities. The empirical data of various projects developed at the companies under study is CMM Level 5 and PCMM (Process CMM), which
are expected to maintain very high quality standards. Data is also collected from companies following TQM model for quality management in the software industry. Complete set of data for this part of the work is obtained through Document Management Repository of respective companies. Additionally, modes of data collection include interactions with project developing team, quality assurance departments, defect management centers etc. The projects that have been collected for the investigation purpose are for non-critical applications which include projects related to ERP (Enterprise Resource Planning), Business, Retail, Medical and web applications.

The empirical data which is collected from the industries are pre-processed and formatted using Waikato Environment for Knowledge Analysis (WEKA) data mining tool. The selected data is converted into Attribute Relation File Format (ARFF) which is readable from WEKA. The transformed data thus obtained is subjected to data mining process. The data set obtained from various software industries consists of 1696 samples with 23 attributes some of which are project development time (Person Hours), experience level of persons, project effort, function point, defect captured, defect estimation, project success rate and real valued features. The class can have two possible values namely success (non-defect) and failure (defect).

5.3 Hybrid method of Random Forest (RF) and FCM

Random Forest (RF) is fast becoming an attractive method in data mining. Random forest (RF) has been found to be one of the most successful ensemble learning techniques. It is also a very powerful and popular technique in pattern recognition and machine learning for high-dimensional classification and skewed problems. As a classifier integration method, RF possesses the features necessary for classifying training simple fast. It is also suitable for feature selection. The method combines Breiman’s “bagging” idea and the random selection of features in order to construct a collection of decision trees with controlled variation [107]. Bagging RF algorithm can be stated as follows:

1. For a given training dataset, extract a new sample set by repeating n times the random sampling using bootstrap method. For example, from the data \((x_1, y_1),..., (x_n, y_n)\) build a sample \((x_1^*, y_1^*),..., (x_n^*, y_n^*)\). Samples which are not being extracted consist of Out-Of Bag data (OOB).

2. Build a decision tree or regression tree based on sample set resulting from step 1;
3. Repeat steps 1 and 2 till it covers all the data. It might result in many trees forming a forest.

4. Let every tree in the forest vote for $x_i$.

5. Compute the sum of votes for every class, the class with highest number of votes is the classification label for $x_i$.

6. The percentage of incorrect classification is the classification error ratio of random forest.

**Fuzzy C- Means (FCM) Clustering**

FCM analyses the distribution of defects. FCM assigns different degrees of membership to the clusters to eliminate the ill effect of hard membership introduced by K-means clustering. This approach employs fuzzy measures as the basis for membership matrix calculation and for cluster center identification. The algorithm consists of four steps as mentioned in section 3.4.1 of chapter 3.

**Hybrid of FCM and Random Forest**

This chapter considers the combination of RF and FCM approaches to build the prediction model for software project defect analysis. FCM is used to perform preliminary grouping of similar defect variables. Variable importance effectively ranks all variables in the dataset in the order of their ability to predict defect patterns. 19 predictor sets together with cross-validated data sets are being used. The new dataset is generated using these 19 predictors which become input to RF procedure. It is used to find interpretable models for predicting defect distribution. RF and FCM clustering have been implemented using MATLAB.

The following cross-validation procedure is used to estimate the prediction accuracy:

- Randomly divided the dataset into 10 disjoint subsets (folds) each fold containing approximately the same number of records.
- The sampling is stratified by the class labels to ensure that the subset class proportions are roughly the same as those in the whole dataset.
- For each subset, a classification model is constructed using nine of the 10 folds and tested on the tenth one to obtain a cross-validation estimate of its prediction accuracy.
• The 10 fold cross-validation estimates are then averaged to provide an estimate for the classifier accuracy constructed from all the data.

5.3.1. Performance measures and experimental Results

In the combined FCM and RF study, the accuracy, sensitivity and specificity were used to evaluate the performance of prediction model. FCM, RF and combined FCM are three commonly used performance measurements and are computed based on the confusion matrix. With the confusion matrices, the accuracy, sensitivity and specificity can be computed as shown below.

The accuracy of classifiers is the percentage of correctness of prediction among the test sets. It is defined in equation (5.1). The sensitivity is referred to as the true positive rate and the specificity as the true negative rate. Both sensitivity and specificity are used for measuring the factors which affect the performance. They are computed using (5.2) and (5.3).

\[
\text{Accuracy} = \frac{TP + TN}{(TP + FN + TN + FN)} \quad (5.1)
\]

Sensitivity which is the Probability of Detection (PD) is simply the percentage of defect-free modules that are correctly predicted by the classification algorithm. It is defined as:

\[
\text{PD} = \frac{TP}{(TP + FN)} \quad (5.2)
\]

Specificity which is nothing but the Probability of False positives is the percentage of defect modules that are correctly predicted by the classification algorithm. It is defined as:

\[
\text{PF} = \frac{TN}{(TN + FP)} \quad (5.3)
\]

Area Under Curve (AUC) / Receiver Operating Characteristic (ROC) curves can be used to evaluate the performance of classifiers. An ROC curve plots PF against PD and provides a visual tool for examining the tradeoff between the ability of a classifier to correctly identify positive cases and the number of negative cases that are incorrectly classified. The higher the PD at low PF values is measured to be the better model. An area of 1.0 suggests near perfect accuracy whereas an area of less than 0.5 shows that the model is worse than simple random guessing. Table 5.1 shows the comparison of the performance of RF and FCM besides showing the results of accuracy measurements.
Table 5.1 Comparison of performance of RF and FCM

<table>
<thead>
<tr>
<th>Projects</th>
<th>AUC (RF)</th>
<th>AUC (FCM)</th>
<th>AUC (RF&amp;FCM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web app</td>
<td>0.751</td>
<td>0.760</td>
<td>0.817</td>
</tr>
<tr>
<td>Business</td>
<td>0.867</td>
<td>0.861</td>
<td>0.910</td>
</tr>
<tr>
<td>Retail</td>
<td>0.813</td>
<td>0.820</td>
<td>0.890</td>
</tr>
<tr>
<td>Medical</td>
<td>0.831</td>
<td>0.821</td>
<td>0.980</td>
</tr>
<tr>
<td>ERP</td>
<td>0.792</td>
<td>0.801</td>
<td>0.879</td>
</tr>
</tbody>
</table>

- 81.7% of web applications, 91% of business, 89% of retail, 98% of medical and 87.9% of ERP applications.

- One can infer that both RF and FCM do not provide adequate accuracy level whereas as hybrid model is showing better accuracy when compared to RF and FCM considered individually.

The results are depicted in the fig. 5.1 given below:

Figure 5.1 Performance of combination of RF and FCM

Figure 5.1 infers that hybrid model of RF and FCM give better results. This section has analyzed the characteristics of software project process data and has proposed a novel method of hybrid of RF and FCM for software project success prediction. The proposed method has been implemented on MATLAB and has been tested on the empirical dataset. The conclusion is that hybrid model of RF and FCM yield comparatively better performance. The experimental result shows that the combination of RF and FCM is appropriate for prediction of software project success, not only has good classification accuracy but also results in relatively simple and interpretable model. From the software perspective, defect can be either an error or a failure or a fault. In section 5.4 we are analyzing the software fault or defect prediction by FCM clustering and adaptive neuro FCM clustering.
5.4 Defect prediction by modified FCM and adaptive neuro fuzzy c-means clustering

The system faults are the defects that brim in executable files. Conventional approaches expect direct navigation into the source code errors. As the system size grows, the complexity of task also grows exponentially. This has necessitated new methods in fault classification. Experimental studies show that miniature bugs are the reasons for faults. In a system of moderate size, the defect labels and non-defect labels are marked during modular phase. Software reliability is the essential parameter while taking decisions about software standard classification [109]. The program attributes are scaled by quantitative representation from software metrics that play crucial role in detecting the software quality based on evaluation parameters [110]. In late 20th century, immense research in this field was oriented towards the prediction of relationship between faults detected and complexity metrics. Finding the necessary metrics depends on multiple variable models [111] in addition to fault size.

A software system is the composition of a number of modules which are dependent on each other. Any module with faults in its functionality will adversely affect the output and will lower its reliability. In this scenario, the detection of faulty modules in early stage (developmental stage) is mandatory for minimizing faults in operation phase. Hence the systems can be classified into faulty / non-faulty categories in their testing phase. This classification diverts the focus to neutralize faulty sections for achieving high reliability and accuracy.

Software fault or error is one of the reasons for failure in execution stage. The error message at each stage of executing the program indicates the fault in programming. Conventionally, errors are the logical errors in software program. The prediction models of software fault proneness technique estimate the amount of faulty modules in a program. Software metrics are the attributes for process and execution of the software system. Various other attributes such as, fault proneness, maintainability, reusability etc. determines the quality of software.

The attributes are the inputs for self-learning model when co-related with weighted error or defect data. The learning model is a system that employs the previous results of performance measure to upgrade itself so as to enhance the performance when compared to previous results. The learning system is modeled in two phases of categorization in its working mechanism i.e. the testing dataset and the training data set. Some predictor functions in
software fault proneness systems simulate the Multilayer Perceptron and Decision Tree algorithm for training and evaluation of effects with respect to testing data set.

**Modified objective function based FCM clustering**

The Fuzzy C-Means clustering for classification of defects in software defect prediction is a conventional approach. Fuzzy C-Means clustering method is the reference of adaptive method that grooms the performance index in defect classification sector for software systems. The enhancement of this method is the collective co-relation of feed-forward neural network [108][112] with fuzzy c-means. It outperforms the assignment of mean deviation and absolute error to a cluster in the process minimizing the distance for defect prediction.

Some of the limitations of FCM algorithm are:

a. Since the sum of the degrees of belongingness of a data point to all clusters is always equal to unity, there may be a tendency to give high membership values for the outlier points. This might result in the algorithm having difficulty while handling outlier points.

b. Since the membership of a data point in a cluster depends directly on its membership values in other cluster centers, there is a possibility that impractical results are produced.

c. The algorithm might not be able to compute the membership value if the distance of a data point is zero.

d. Another shortcoming of FCM algorithm is the difficulty in selecting appropriate parameters. One of the important parameters is the index of fuzziness m which influences the performance of FCM algorithm when clusters in the data set have different densities. When \( m = 1 \), FCM algorithm degenerates into HCM algorithm. Care should be taken while choosing good value for \( m \) and this aspect has been considered in this chapter.

e. It is possible that FCM algorithm gets struck in the local minima when our aim is to find the global extreme. To increase the probability of finding global extreme, various alternative methods for the optimization of clustering algorithm have been suggested. One possibility is the integration of genetic algorithm (GA) with FCM algorithm which is discussed in chapter 6.
The results reported are based on datasets obtained from NASA (National Aeronautics and Space Administration) public MDP (Modular toolkit for Data Processing) repository. This is a public repository for NASA datasets. NASA datasets are composed of several static code attributes.

The data of PC1 (NASA) is input to the system. Fuzzy C-Means for clustering the faulty data requires pre-processing of data in order to reduce time consumption. The output of fuzzy C-Means is fed to Adaptive Neuro-Fuzzy C-Means clustering algorithm. The algorithm tuned by Neural Network trains the data and improves the performance index. The output of the algorithm is compared with the output of Fuzzy C-Means for accuracy, reliability, RMSE and MAE.

Table 5.2 NASA PC1 Dataset

<table>
<thead>
<tr>
<th></th>
<th>Defect</th>
<th>Defect-free</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC1 Dataset</td>
<td>23</td>
<td>77</td>
</tr>
</tbody>
</table>

PCI dataset in the form of m x n matrices with distinguish features are being considered as input,

\[ X_{ij} = A \times B \]

\( X_{ij} \) is the initial PC1 data matrix.
A is the size of dataset.
B is the number of distinguish features.

\[ s = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2} \]  \hspace{1cm} (5.4)

Where
\( \bar{x} \) is \( \frac{1}{n} \sum_{i=1}^{n} x_i \)
n is the number of elements in sample.
\( x_i \) is the random value of B taken from a finite data set \( x_1, x_2, ..., x_n \).
s is the standard deviation.

\[ s = \frac{x_1 + x_2 + ... + x_n}{n} \]  \hspace{1cm} (5.5)
Where
t is the mean value of distinguish features
\[ B = x_1, x_2, \ldots, x_n \]

From eq. (5.4) and eq. (5.5) the new preprocessed dataset will be (matrix representation):
\[ \phi_y = s \times t \]  (5.6)

Where
s is the standard deviation and
t is the mean of distinguish features.

Fuzzy C-Means Clustering (FCM) determines the degree of membership of each data point to a cluster. Generally, the restrictive constraint scales the performance of various hard clustering algorithms to cluster original data into different cluster groups. Bezdek [113] proposed Fuzzy C-Means Clustering to improve the Hard C-Means Clustering (HCM). In FCM, the overlapping of data points into different groups can happen. The flow diagram of FCM is given below:

![Flow diagram of FCM](figure)

Fuzzy C-Means clustering has been considered in [114] by Zhiwei Gao et al. In this method, the output vector of preprocessing (eq. 5.6) taken as input \( (\phi_y \text{ for } i = 1, 2, \ldots, n) \) are classified into c clusters and the cluster centers are found. The primary difference between HCM and
FCM lies in fuzzy partition. This partition is responsible for determining the degree to which every data point belongs to every group. The membership matrix \( U \) which is a matrix of 0s and 1s helps in deriving the Fuzzy partition. Based on the rules of normalization the sum of the membership grades of all elements in a cluster is equal to 1.

\[
\sum_{j=1}^{c} u_{ij} = 1, \forall i = 1, ..., n \quad (5.7)
\]

Here \( c \) denotes the number of clusters.

The objective function (or cost function) of FCM is the generalization of equation:

\[
J(U, c_1, ..., c_n) = \sum_{i=1}^{n} \sum_{j=1}^{c} \mu_{ij}^p d_{ij}^2 
\]

\[
J(U, c_1, ..., c_n) = \sum_{i=1}^{n} \sum_{j=1}^{c} \mu_{ij}^p d_{ij}^2 = \sum_{i=1}^{n} \left( \sum_{j=1}^{c} \mu_{ij}^p d_{ij}^2 \right) \quad (5.8)
\]

where, \( u_{ij} \) lies between 0 and 1, \( c_i \) is the clustering center of fuzzy group \( i \), \( d_{ij} = \| c_i - x_j \| \) denotes the Euclidean distance between the \( i^{th} \) clustering center and \( j^{th} \) data point, \( x_j \) is the \( j^{th} \) data point, \( m \in [1, \infty] \) denotes the weighted index.

Our aim is to minimize equation (5.8).

\[
J(U, c_1, ..., c_n) = \sum_{i=1}^{n} \sum_{j=1}^{c} \mu_{ij}^p d_{ij}^2 = \sum_{i=1}^{n} \left( \sum_{j=1}^{c} \mu_{ij}^p d_{ij}^2 \right) + \sum_{i=1}^{n} \lambda_i \left( \sum_{j=1}^{c} \mu_{ij} - 1 \right)
\]

\[
J(U, c_1, ..., c_n) = \sum_{i=1}^{n} \sum_{j=1}^{c} \mu_{ij}^p d_{ij}^2 + \sum_{i=1}^{n} \lambda_i \left( \sum_{j=1}^{c} \mu_{ij} - 1 \right) \quad (5.9)
\]

Where \( \lambda_i, j = 1, 2, ..., n \) is the Lagrange multiplier of \( n \) inhibited formula described in equation (5.7). The values that minimize equation (5.8) are as follows:

\[
c_i = \frac{\sum_{j=1}^{n} \mu_{ij}^p x_j}{\sum_{j=1}^{n} \mu_{ij}^p} \quad (5.10)
\]

\[
\mu_{ij} = \frac{1}{\sum_{k=1}^{n} \left( \frac{\mu_{ik}^p d_{ik}^2}{\sum_{j=1}^{n} \mu_{jk}^p d_{kj}^2} \right)^{(p-1)}} \quad (5.11)
\]

Fuzzy C-Means is an iterative process which is made simple by virtue of equations (5.10) and (5.11). The steps given below can be used to compute the center \( c_i \) and membership matrix \( U \).

Step1: Select a number anonymously in the range 0 to 1 for calculation of membership matrix \( U \).

Step2: Compute the various clustering centers from eq (5.7).

Step3: In accordance with equation (5.8), compute the cost function. The iteration will be terminated, when it is less than the threshold value in comparison to the change in last cost function value.

Step4: Compute the next matrix \( U \) (based on eq. 5.11) and return to step 2.
The weighted index \( m \) decides the flexibility of the algorithm. If it is high, then the cluster effect will be poor and if it is too small, the algorithm will be nearer to Hard C-Means Clustering (HCM).

**Adaptive Neuro Fuzzy Inference**

Figure 5.3 given below shows the Adaptive Neuro fuzzy Inference System architecture.

![Adaptive Neuro Fuzzy Inference System Architecture](image)

Figure 5.3 Adaptive Neuro Fuzzy Inference System Architecture [115]

It is a reasoning fuzzy system that is trained by neural network for computation of membership function parameters. The method tracks the input output data as a non-linear relation with inputs \( x, y \) and \( f \) as output. By training the system to a number of epochs, the knowledge base is developed. The fuzzy inference system can be explained using the following layers and can be used for minimization of error rate.

Layer 1: In this layer, the degree of membership is upgraded with reference to the parameters of fuzzy sets [116].

\[
l = \mu_{A_i}(x), l = 1, 2 \quad (5.12)
\]

\[
l = \mu_{B_i}(y), l = 1, 2 \quad (5.13)
\]

Layer 2: In this layer, the fuzzy value of inputs is calculated. In the range \([0, 1]\) the membership value of fuzzy set is determined [116].

\[
w_t = \mu_{A_i}(x)\mu_{B_j}(y), l = 1, 2 \quad (5.14)
\]

Layer 3: This layer normalizes the strength [116].

\[
w_t = \frac{w_t}{w_t + u_v}, l = 1, 2 \quad (5.15)
\]
Layer 4: Input to this layer is the fuzzy output of layer 2 whereas the output is a single number. The parameters are given by eq. 5.16 [116].

\[
\mu_{y|z} = \mu_y(x_1 + q(y_1 + \eta)) \quad (5.16)
\]

5. Layer 5: Overall output is computed by summing each incoming signal [116].

\[
\sum_i \mu_{yi} = \sum_i \frac{\mu_{yi}}{\sum_i \eta} \quad (5.17)
\]

The Flow Diagram of Adaptive Neuro Fuzzy C-Means Clustering is given below.

![Flow Diagram of Adaptive Neuro Fuzzy C-Means Clustering](image)

**Figure 5.4 Flow Diagram of Adaptive Neuro Fuzzy C-Means Clustering**

### i) Software Fault Detection Using Modified objective function based FCM clustering

The matrix we consider is a 100x21 matrix which indicates that there are 100 cases of faulty or non-faulty modules with 21 properties of each. Pre processing of the data (computation of mean and standard deviation) is done for each software stream.

This reduces the size of the matrix to \(100 \times 21\) i.e. 100 cases with 2 properties of each either defect or defect free.

\% of false: 6.94
% of true: 93.05

The pre-processed data is fed to Fuzzy C-Means clustering to obtain 25 clusters. The first cluster of data is labeled as non-faulty while the remaining 24 fields indicate faulty data. Fuzzy C-Means finds the location of cluster center and assigns each stream to a cluster. The mean and standard deviation of the stream is computed and then assigned to the cluster to which the distance is minimum. The stream is classified as faulty if the cluster is faulty else true.

ii) Software Fault Detection Using Adaptive Neuro-Fuzzy C-Means Clustering

The dimension of the matrix is again 100 x 21 which indicates 100 cases with 21 properties each. Pre processing of data reduces the size of the matrix to 100 x 2 i.e. 100 cases with 2 properties each.

A fuzzy inference system is generated using FCM. Both systems share the same level of accuracy. This inference system is fed to feed forward neural network with training data. In the training stage, neural network modifies the structure of fuzzy inference system to obtain higher level of accuracy.

5.4.1 Performance measures and experimental Results

Figure 5.5 is a graphical representation of PC1 data distribution of defect and defect free modules. In training, dataset is taken as input for class distribution: the class value (defects) is discrete in nature.

![Graphical representation of PC1 data set defect and defect free modules](image)

Figure 5.5 Graphical representation of PC1 data set defect and defect free modules
Predictive Model for Software Project Success

Dept of CSE, Jain University, Bangalore

% data with positive attribute: 23
% data with negative attribute: 77 = 77%

Figure 5.6 Adaptive Neuro-Fuzzy Inference Model for PC1 Dataset

Figure 5.6 shows Adaptive Neuro-Fuzzy Inference Model for PC1 Dataset with three rules. Figure 5.6 presents Membership Function Plot of Fuzzy Inference System and Figure 5.7 shows Membership Function Plot of Adaptive Neuro Fuzzy Inference System.

Figure 5.7 Membership Function Plot of Fuzzy Inference System
The performance factors considered here are accuracy, reliability, RMSE (Root Mean Square Error) and MAE (Mean Absolute Error). Table 5.3 compares the performance of Fuzzy C-Means and Adaptive Neuro Fuzzy C-Means Clustering.

Table 5.3 Comparison of Fuzzy C-Means and Adaptive Neuro Fuzzy C-Means Clustering

<table>
<thead>
<tr>
<th></th>
<th>Fuzzy C-Means</th>
<th>Adaptive Neuro Fuzzy C-Means</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td>77%</td>
<td>91%</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>72.98%</td>
<td>73.98%</td>
</tr>
<tr>
<td><strong>RMSE</strong></td>
<td>0.068</td>
<td>0.09</td>
</tr>
<tr>
<td><strong>MAE</strong></td>
<td>0.23</td>
<td>0.13</td>
</tr>
</tbody>
</table>

This work empirically evaluates and compares the performance of Fuzzy C-Means Clustering technique and adaptive Neuro-Fuzzy C-Means Clustering for software fault prediction. Testing has been carried out using MATLAB 2010A on the PC1 testing database. The proposed Adaptive Neuro-Fuzzy C-Means Clustering based prediction technique shows an accuracy rate of 91%.
5.5 Summary

This chapter has analyzed the characteristics of software project process data and has proposed a novel method which is the hybrid of RF and FCM for software project success prediction. The hybrid model of RF and FCM appear to give comparatively better performance. Experimental results show that the combination of RF and FCM is appropriate for prediction of software project success. It not only has good classification accuracy but also results in relatively simple and interpretable model.

We have also carried out empirical evaluation and comparison of the performance of modified objective function based FCM clustering and adaptive Neuro-Fuzzy C-Means clustering for software defect prediction. The proposed technique shows 91% accuracy in results. The implementation shows that for the prediction of defect prone classes, hybrid models perform better.