Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 10; April-June, 2015 pp. 24-28
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/acsit.html

Data Mining Methodologies to Predict Defects in Data Sets

Niharika Chaudhary[1], Gaurav Mehta[2] and Karan Bajaj[3]

[1]Student, Chitkara University, Himachal Pradesh, India
[2,3]Chitkara University, Himachal Pradesh, India
E-mail: [1]ncniharika9@gmail.com, [2]gaurav.mehta@chitkarauniversity.edu.in,
[3]karan.bajaj@chitkarauniversity.edu.in

Abstract—*As the quantity of data is growing exponentially, there has been an increase in the demand for reliable data. Therefore, there is a need to make the data free of defects or minimize them. The National Aeronautics and Space Administration Metrics Data Program (NASA MDP) and PROMISE repositories provide the software metrics and associated error data to the research community. However, this data is not totally clean and hence poses some quality problems such as repeated data points, missing values, etc. But with the use of various machine learning methods such as Decision Trees, Neural Networks, etc. the defects can be minimized and it is possible to predict error early in the lifecycle. A number of open source tools like Waikato Environment for Knowledge Analysis (WEKA) and Rapid Miner are freely available for this purpose. In this paper, the concept of defect prediction is described along with some defects present in data sets. A brief description of J48 and Random Forest classifiers and Cobweb and K-means clusterers present in WEKA is also given. Evaluation of the performance of these algorithms on the PROMISE data sets to find out the best algorithm among them will be done by us in the future and cleaner versions of these data sets will be produced with the hope that other fields such as banking, education, business, medical diagnosis and many more be able to draw benefit from our research work.*

Keywords*: Machine learning, defect prediction, WEKA*

1. INTRODUCTION

Data in the real-world is noisy (contains missing values), incomplete and also inconsistent. Noisy data means meaningless data or in more appropriate words we can say that any data which is beyond the ability of a machine to understand and interpret it correctly is called noisy data. The results of any data mining analysis can be affected unfavorably by noisy data and it also increases the storage space required. Reliable data is of great significance and has a lot of demand. It is essential in almost every field like medical diagnosis, education, research, banking, business, etc. Noisy data can be removed by using information from historical data. The data cleaning process removes noise and hence makes the data sets suitable for machine learning. Therefore, an essential task during mining the data is smoothing out noise.

A defect is an error that leads to production of unanticipated results. The main cause of these defects is mistakes made by programmers in the source code of a program or in the frameworks used by such programs. Some are caused by production of wrong code by compilers. Defect prediction is a sub-domain of data mining which is the process of finding erroneous components in data before the beginning of the testing stage. Occurrence of defects is unavoidable, but we should try our best to keep their count to the minimum. Defect prediction has the merits of reducing the cost, decreased rework, higher customer satisfaction and more authentic data. Hence, defect prediction is important to enhance quality of data and to learn from past mistakes.

A number of models can be used for defect prediction, each having its own set of advantages as well as disadvantages. Information collected during testing and defect prediction can be useful for predicting defects in similar types of projects. A large number of data mining tools such as WEKA and Rapid Miner are freely available which can be used for this purpose.

The rest of this paper is presented as follows: in the next section we discuss related work; papers where defect prediction has been discussed. In Section 3, we give an overview of some common defects in data sets. Section 4 describes two data mining techniques, classification and clustering. Section 5 gives the details about implementing these techniques in future. Section 6 concludes the paper and Section 7 lists the references.

2. RELATED WORK

The quality of data has become a very important parameter to focus upon. But the prerequisite is to find out why the data that you want to clean needs to be cleaned. Gray et al. [1] found out that the NASA data sets contained repeated data points and hence it was necessary to clean this data both to make it suitable for machine learning and to remove noise. They also presented a data cleansing process and implemented it on all the 13 original NASA data sets. After the cleaning process, the number of recorded values decreased by 6 to 90 percent in all the data sets. They concluded that experiments based on the NASA data sets which included the repeated data points may have led to erroneous findings. The proportion of recorded values removed during data cleaning is shown in Fig. 1.

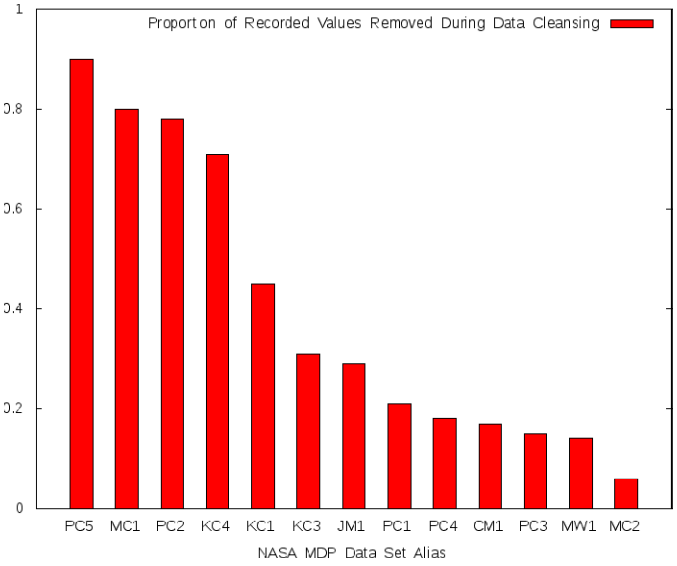

Fig. 1: The proportion of recorded values removed during data cleansing [1]

Their work was taken a step further by Shepperd et al. [2] who described in detail a preprocessing algorithm "NASA MDP Data Preprocessing Approach" applied to original NASA datasets and made the cleaned versions of these datasets available. They made valid comparisons between studies which have used the same datasets. They concluded that some differences and data quality may seem less important and affect only a small proportion of the observations. Hence, addressing such problems is not optional. Table 1 shows the comparison between different data sets in MDP and Promise repositories based on number of cases and features. They found that the two versions are significantly different and the ordering of cases also differs which may affect n-fold cross-validation technique.

Table 1: Comparison of the two versions of NASA defect data sets [2]

Data Set	Cases		Features	
	MDP	Promise	MDP	Promise
CM1	505	498	43	22
JM1	10878	10885	24	22
KC1	2107	2109	27	22
KC2	n. a.	522	n. a.	22
KC3	458	458	43	40
KC4	125	n. a.	43	n. a.
MC1	9466	9466	42	39
MC2	161	161	43	40
MW1	403	403	43	38
PC1	1107	1109	43	22
PC2	5589	5589	43	37
PC3	1563	1563	43	38
PC4	1458	1458	43	38
PC5	17186	17186	42	39

A large number of data mining techniques help in the process of defect prediction. There is no single technique that is the best among all as each one has its own set of advantages and disadvantages. Many researchers in this field have made comparisons between the different techniques and algorithms. 37 classification algorithms were compared on 5 different NASA data sets by Haghighi et al. [3]. They found that Bagging classifier showed the best performance in fault detection. Also, they proposed a fault detection system showing better performance as well as reducing the cost of fault detection. Performance of Bagging was compared against two more classifiers which verified that Bagging has the highest performance on fault detection systems as shown in Table 2. Mittal and Dubey [4] depicted defect handling life cycle models and the use of Cost Constructive COQUALMO model for defect handling which is a two-step defect prediction model. They used a Process Improvement model including Defect Identification, Classification, Analysis, Prediction, Prevention and finally Process Improvement. But the stages of work flow that they have used in their paper are more complex which increases the number of stages of Defect Handling.

Table 2: Comparison of the appropriate classifier (Bagging) and one of the most commonly used classifiers in fault detection systems (Naïve Bayes) [3]

Dataset	Classifier					
	Bagging		Naïve Bayes		Classification via Regression	
	ACC	AUC	ACC	AUC	ACC	AUC
KC1	85.68	0.807	83.59	0.757	85.20	0.791
KC2	83.33	0.839	83.90	0.806	82.95	0.835
PC1	93.32	0.811	89.90	0.641	92.87	0.856
CM1	89.95	0.733	86.14	0.615	88.95	0.699
JM1	81.04	0.733	80.58	0.646	81.13	0.713
PC4	90.60	0.907	89.50	0.814	89.36	0.907
PC3	89.25	0.817	63.46	0.764	88.80	0.814
PC2	99.58	0.778	98.31	0.770	99.58	0.760
MW1	91.81	0.674	86.35	0.696	91.56	0.831
MC1	99.41	0.931	95.25	0.868	99.42	0.949
KC3	89.51	0.806	88.42	0.794	90.39	0.814

A survey on the existing data mining techniques for defect prediction in software was carried out by Kaur and Bajaj [5]. Various models and techniques were studied which have their own advantages and disadvantages. They introduced the concept of neural networks which is one of the promising techniques for predictive models. They mentioned that designing Neural Networks is difficult as the optimal number of nodes, hidden layers, activation function, etc. is to be determined. Also, they have high computational burden. Sanyal and Singh [6] studied various fault prediction techniques like Decision Trees, Neural Networks, Density based clustering approach, Bagging method and Naïve Bayes. They concluded that fault prediction is necessary to decrease the cost of testing and to improve reliability. Paramshetti and Phalke [7] discussed the existing techniques used for defect prediction. A brief description of different code metrics such as Cyclomatic Complexity, Halsted's Product Metrics and Product Metrics is also given in this paper. They analyzed the advantages and limitations of various defect prediction techniques – Artificial Neural Network, Support Vector Machine, Decision Tree, Association Rule and Clustering as listed in Table 3.

Table 3: Comparative Analysis [7]

Techniques	Data sets used	Advantages	Limitations
Artificial Neural network	NASA AR1,AR6 and MDP	No need to know metrics relationships. It has self learning capability therefore get more accuracy	It cannot manage imprecise information
Support Vector Machine	NASA AR1, AR6	Using different kernel function it gives better prediction result	Not suitable for large number of software metrics
Decision Tree	NASA AR1,AR6	Performing operation on tree structure therefore more accurate result compared to others	Construction of decision tree is complex
Association Rule	NASA MDP repository	Generated rules using historical data and predict defect	Require continuous value of software metrics
Clustering	NASA MDP repository	It is suitable for small dataset	Dataset should be unlabeled

3. SOME COMMON DEFECTS IN DATA SETS

A defect is nothing but an error in data. Defects can occur in any type of data and at any time. For example, consider the NASA software defect data sets which can be downloaded from NASA MDP or PROMISE data repositories. Research has shown that these data sets are not clean and contain some inconsistencies. Therefore, it is necessary that more and more work be undertaken on them so that cleaner data sets are available for future work. Some common types of defects that are present in these data sets are listed below:

3.1 Missing Values

The attributes for which at least one instance value is not present are known as attributes with missing values. In [1], it was mentioned that missing values in data sets can occur due to division by zero error. One way is either to discard all instances which contain missing values or replace the missing values by zero. They concluded that it is preferable to clean the data rather than remove it so that the data is not reduced.

3.2 Identical Values

If two or more attributes have the same value for all instances then those attributes are said to contain identical values. Only one of them can be kept and the rest can be deleted as redundant data adds to the storage space.

3.3 Constant Values

It refers to those attributes in which every instance has the same value. Since such attributes contribute no information to the data, they can be deleted.

4. DATA MINING TECHNIQUES FOR DEFECT PREDICTION

Data mining converts raw data into meaningful data [5]. It is useful in finding patterns in data which are further used to extract patterns in new sets of data. The main aim of data mining is prediction using automated data analysis and finding future outcomes based on past results. In this paper, an overview of some data mining techniques along with their merits and demerits is discussed for identifying fault prone modules as data quality is a very important parameter for researchers who derive results from existing data. Classification and clustering techniques which are discussed below play a vital role in data cleaning. Waikato Environment for Knowledge Analysis (WEKA), an open source and platform-independent data mining software houses these algorithms and provides an automated way to apply them on data sets.

4.1 Classification

It is a form of supervised learning which facilitates prediction of a certain outcome from the given input. Firstly, a training set, in which all class labels are already assigned to each variable, is used to build a classification model. This model is then applied on a test set in which class labels are not given. Thus, first the classification algorithm learns from the training set and then performs classification of the test set [8].

4.1.1 J48 Decision Tree. A decision tree is a predictive model in which classification is performed in the form of a tree structure. There are internal/decision and external/leaf nodes in a decision tree which are connected through branches. If the value of an attribute is to be determined, the internal nodes make a decision as to which node should be visited next based on the branch values. The leaf nodes represent a value or label that the attribute should have. J48 is a decision tree classifier which constructs a decision tree based on the training set which contains known attribute values. The internal node is split based on the attribute which tells us the most about data instances so that we can split it. In [9], it was seen that J48 performs very well. J48 has the advantage that it can handle attributes with missing values by marking them as '?'.

4.1.2 Random Forest. The Random Forest algorithm is one of the best classification algorithms which classify large amounts of data accurately. It creates not one but a number of decision trees during training. All X cases in the training set are sampled at random. This sample acts as the training set for growing the tree. If N input variables are there, a number n<<N is chosen such that n variables are selected at random out of the N and the best split on these n is used to split the node. Random Forest estimates missing data efficiently and performs well even if large proportion of data is missing [10]. It can handle thousands of input variables without variable deletion and run efficiently on large databases.

4.2 Clustering

The process of grouping data points such that those in one group possess some similarity which is different from those in other groups is called clustering. Clusters are subsets that are formed as a result of grouping. It is a form of unsupervised learning in which no class labels are provided [8]. Clustering facilitates in assigning class labels to data as the groups as a whole can be assigned different labels. The quality of a clustering method is measured by its ability to discover some or all of the hidden patterns.

4.2.1 Cobweb. Developed in the 1980s, Cobweb is an incremental algorithm for performing hierarchical clustering. It incrementally produces a classification tree known as a 'dendrogram'. The class at each node is labeled by a probabilistic concept. The construction of the classification tree is based upon a measure known as 'category utility'. Objects are inserted into the tree such that the highest category utility is obtained [11].

4.2.2 Simple K-means. It is one of the most common and efficient clustering algorithms used nowadays. K-means algorithm partitions m observations into k clusters such that each observation belongs to a cluster in which it is closest to the mean of that cluster. Euclidean distance is used as a metric in clustering using K-means. First, the k means or centroids are defined for each of the k clusters [12]. Then, each point is associated with the closest mean or centroid.

5. IMPLEMENTATION DETAILS

Data quality is of utmost importance nowadays. It is a very important parameter for researchers who derive results from existing data. Data mining can be applied to improve the quality of data sets. Also, the implementation of different machine learning algorithms can be used for predicting fault proneness in software. Our approach is to analyze the performance of the classification and clustering algorithms described above on the NASA defect data sets in the near future. A defect prediction model to enhance the quality of these data sets may also be built.

STEP 1: A survey and comparison of classification and clustering algorithms on NASA defect data sets

A survey and comparison of the algorithms discussed in this paper i.e. J48 Decision Tree, Random Forest, COBWEB and Simple K-means will be done. Their pros and cons will be listed and evaluation of their performance will be done based on the results produced.

STEP 2: Model designing for defect prediction

A data mining model for defect prediction may also be built. First, it will be trained using a training set and then tested on a test set to check its accuracy.

STEP 3: Implementation of the designed model on the data sets

The model designed in step II will be used to predict defects in new data sets. It will be applied to test data to generate predictions and make inferences about relationships.

STEP 4: Performance evaluation of new model

The performance of the built data mining model will be evaluated on the basis of accuracy of the resultant data sets produced on which it was implemented in step 3.

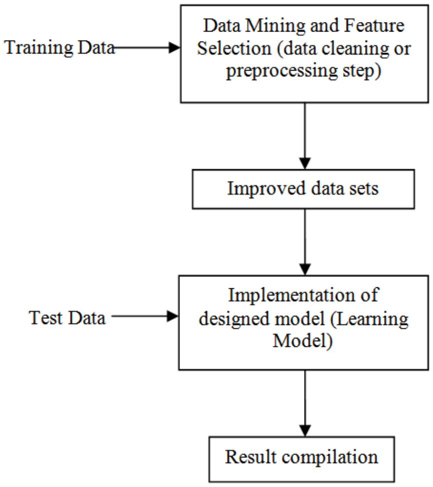
Training Data
Data Mining and Feature Selection (data cleaning or preprocessing step)
Improved data sets
Test Data
Implementation of designed model (Learning Model)
Result compilation

Fig. 2: Data cleaning and data mining model generation

6. CONCLUSION AND FUTURE WORK

In this paper, we studied the NASA data sets and identified some common defects present in them like missing values, identical values and constant values which are described above and also discussed how defect prediction minimizes or reduces them. Classification algorithms such as J48 Decision Tree and Random Forest and clustering algorithms such as COBWEB and Simple K-means are discussed. These classification and clustering techniques help clean the data before applying a defect prediction model on them. These algorithms will be implemented on the NASA defect data sets available from the PROMISE data repository and their performance will be analyzed so as to produce improved versions of these data sets. In addition to that, the best algorithm among these will be found out based on the performance of the algorithms which will benefit researchers in other fields such as medicine, banking, business, etc.

REFERENCES

[1] Gray, D., Bowes, D., Davey, N., Sun, Y., and Christianson, B., "The Misuse of the NASA Metrics Data Program Data Sets for Automated Software Defect Prediction", *15th Annual Conference on Evaluation and Assessment in Software Engineering (EASE 2011)*, 2011, pp.96-103.

[2] Shepperd, M., Song, Q., Sun, Z., and Mair, C., "Data Quality: Some Comments on the NASA Software Defect Data Sets", *IEEE Transactions on Software Engineering*, 2013, 39, 9, pp.1208-1215.

[3] Haghighi, A. A. S., Dezfuli, M. A., and Fakhrahmad, S. M., "Applying mining schemes to software fault prediction: A proposed approach aimed at test cost reduction", in *Proceedings of the World Congress on Engineering*, 2012, 1, pp.415-419.

[4] Mittal, A. and Dubey, S. K., "Defect handling in software metrics", *International Journal of Advanced Research in Computer and Communication Engineering*, 2012, 1, 3, pp.251–274.

[5] Kaur, R. and Bajaj, P., "A Review on Software Defect Prediction Models Based on Different Data Mining Techniques", *International Journal of Computer Science and Mobile Computing (IJCSMC)*, 2014, 3, pp.879-886.

[6] Sanyal, A. and Singh, B., "A Systematic Literature Survey on Various Techniques for Software Fault Prediction", *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, 2014, 4, pp.1210-1212.

[7] Paramshetti, P. and Phalke, D. A., "Survey on Software Defect Prediction Using Machine Learning Techniques", *International Journal of Science and Research (IJSR)*, 2014, 3, pp.1394-1397.

[8] Kaur, P. J. and Pallavi, "Data mining techniques for software defect prediction", *International Journal of Software and Web Sciences (IJSWS)*, 2013, pp.54-57.

[9] http://www.d.umn.edu/~padhy005/Chapter5.html

[10] http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home .htm#prox

[11] http://en.wikipedia.org/wiki/Cobweb_(clustering)

[12] Sharma, N., Bajpai, A., and Litoriya, R., "Comparison the various clustering algorithms of WEKA tool", *International Journal of Emerging Technology and Advanced Engineering (IJETAE)*, 2012, 2, pp.73-80.