An Approach to Predict Software Project Success by Data Mining Clustering

Suma.V, Pushpavathi T.P, and Ramaswamy.V

Abstract— Generation of successful project is one of the core challenges of the day. Since information is always a valuable resource, data of software projects comprises of wealth of information about the project status, progress and its evolution. Mining software engineering data has therefore emerged as a successful research since two decades. Further, predicting the success of software projects based upon information related to defect management is deemed to be one of the vital activities in software engineering research. Data mining techniques assures exploring the defect management information from the historical projects depending on which the project manager plans, controls and executes the software development in order to produce high quality software. This paper presents an empirical investigation carried on several projects developed at various software industries having different production capabilities. The aim of this investigation is to comprehend the effectiveness of appropriate clustering techniques for efficient defect management and henceforth accurate prediction of project success. The investigation results infer that K-Means clustering technique is more efficient than other clustering techniques in terms of processing time, scalability and efficiency. This awareness of right choice of clustering technique further accelerates the developing team to accurately plan towards generation of improved quality software.

Keywords— Data Mining Clustering, Software Engineering, Defect management, Project Management.

I. INTRODUCTION

MINING software engineering data has emerged as one of the successful research directions since few decades. Software repositories contain a wealth of valuable information about software projects. Using the information stored in these repositories which act as predictive pattern for estimating, planning and controlling the future project, the software developing team can now depend less on their intuition and operate more with realistic and field data.

Data mining has emerged as one of the assuring techniques through which valuable information can be mined from the population of empirical projects.

Clustering is considered to be one of the most significant techniques of data mining which is applicable in various domains such as life sciences, medical sciences, engineering and so on. Clustering technique can be viewed in various perspectives depending on operational environment which includes unsupervised learning in pattern recognition, numerical taxonomy in biology and typology in social sciences and partition in graph theory etc [1].

However, software is one of the domains which have laid its strong foot in almost all domains of applications. Nevertheless, the advancement in technology, software life cycle which is a human activity is prone to produce software having defects. Software defects include product defects such as defects at requirement analysis phase, design flaws, implementation errors and so on. Defects however can also be due to process imperfections. Additionally, defects have varying degrees of complexity and severity as all defects do not have same nature. Hence, to deliver defect free software, it is imperative to predict and fix all defects before product deployment to the customers, which is yet another key challenge of the software industry. However, software repositories have enormous information which is highly beneficial in assessing the software quality. Hence, data mining techniques and machine learning algorithms can be applied on these repositories to extract the useful information on effective defect management.

The most habitually discussed hitch in any software organization is therefore defect prediction to enhance quality and reliability. To accomplish the above-said objective, organizations are now integrating data mining techniques in software engineering process effectively which includes defect management. Clustering is one of the main tasks of explorative data mining and a common technique for statistical data analysis [2][11]. One of the vital characteristic of the clustering process is to categorize the software projects into sensible groups. It facilitates in discovering similarities and differences upon which useful conclusions can be drawn from the mined information which in turn improves the quality of the software.

However, quality is not a stable state but varies depending upon applications, nature of projects, industrial environment and so on. Hence, the choice of clustering algorithms is based on availability of type of input data, purpose and application domain of the project.

The aim of this paper is to analyze the effectiveness of popular clustering algorithms in predicting the success of the projects. Organization of the paper is as follows. Section II specifies the related work in the domains of data mining which
includes clustering and software engineering. Section III provides research methodology followed during this investigation. Section IV presents details about data mining clustering techniques on various software projects. Section V indicates the performance measurements of various clusters thereby indicating the choice of apt cluster technique for effective project management. Section VI specifies the experimental results and summary of this part of research is briefed in Section VII.

II. RELATED WORK

The growing complexities of software and increasing demand of software projects have led to the progress of continual research in the areas of effective project management. Data mining has proven as one of the established techniques for effective project management since two decades.

Authors in [1] presented different clustering algorithms of Waikato Environment for Knowledge Analysis (WEKA), which is the most popularly used data mining tool. Every algorithm has its unique importance.

Authors in [7] conducted an empirical comparison of four clustering algorithms (i.e. CLARA, CLARANS, GAC-R3, and GAC-RARw) over a wide range of data characteristics. Experimental results of CLARANS outperform its counterparts both in terms of clustering quality and execution time.

However, demand for effectively mining the valuable information of software has focused research to integrate data mining with software engineering. Authors in [4] provide understanding of mining software engineering data into three perspectives namely data sources being mined, tasks being assisted and mining techniques being used. It is an open issue to design a good data mining prediction model [3].

Authors in [8] have shown that there is no particular learning technique that performs the best for all the data sets. However, IBL (Instance Based Learning) and 1R (1Rule), which are some of the widely used learning techniques, have proven to exhibit relatively better consistency in predicting accuracy of data sets when compared with other learning techniques.

Further, authors in [9] suggest the association rule mining to be an attractive technique for the software engineering community due to its relative simplicity, transparency, and seemingly effective in constructing prediction systems. Since, defect prediction is one of the most significant activities of software engineering, application of data mining techniques for effective prediction of software defects is one of the core needs of the day.

Authors in [5] surveyed different data mining algorithms used for defect prediction and also discuss the performance and effectiveness of data mining algorithms. Authors of [6] made a comparative analysis of performance of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Neuro-Fuzzy System for prediction of level of severity of faults present in Java based object oriented software systems. Their results indicate no variations in accuracy of DBSCAN and Neuro-Fuzzy based system.

Interestingly, the work of authors in [10] triggers few research issues which throw light for research to progress towards existence of common characteristics within the data related to software defects.

Thus, this part of the research aimed at exploring the choice of right clustering algorithm for effective prediction of software success using defect count as one of the influencing parameter.

III. RESEARCH METHODOLOGY

This research focused upon the selection of clustering technique that yields effective results in prediction of project success through defect profile. In order to achieve the aforementioned objective, a deep investigation is carried out upon several projects from various software industries.

However, software is a huge population which needs to be sampled and analyzed. Therefore, this work directed towards formulation of hypothesis for overcoming the intrinsic complexities involved within the universe of software projects which is as explained in subsequent sections of this paper.

Further, the empirical data related to defect management aspects were collected from the sampled software industries through their Document Management Repository and Defect Management Centres. Modes of data collection includes interactions with project developing team, quality assurance departments, defect management team etc. in the form of interviews, face to face communication and so on.

Empirical data analysis includes application of most popular clustering techniques on the software projects to evaluate the efficiency of each clustering algorithms for effective prediction of project success based on defects. The observational results indicate that K-means clustering is most effective when compared to other clustering approaches in terms of processing time, scalability and efficiency.

IV. RESEARCH WORK

This research was carried out on several software projects which were developed in various software industries within the framework of formulated hypotheses.

Hypothesis 1: All projects are developed using similar technology, operating environment and programming languages.

Hypothesis 2: The projects are categorized as small project, medium projects and large projects based on total developmental hours required for the project, which is one of the popular complexity measurements of software projects in industries. A project that requires less than 1000 person hours is considered to be small project and that which needs up to 5000 person hours is medium project. A project is considered to be large project when it requires more than 5000 person hours of total developmental time.

Hypothesis 3: Defect associated data is used for analyzing the effectiveness of clustering algorithms towards prediction of success of software projects.
Threat to hypothesis 1: Projects developed with dissimilar environment, operating systems and programming languages may have a divergent pattern of defect count and hence varying project success.

Threat to hypothesis 2: Legacy projects, innovative projects may not fall under the category of measurement of project complexity based on total person hours.

Threat to hypothesis 3: Analysis of clustering algorithm depends upon the right choice of data, its purpose and its application in the project.

Steps to resolve threat to hypothesis 1: The software industries under study are CMMI Level 4 and Level 5 certified software industries.

Steps to resolve threat to hypothesis 2: Projects that were developed from the year 2005 onwards are considered for the investigation purpose. Projects developing non critical applications such as ERP, web applications and so on using Linux operating system and object supporting languages such as Java is further considered for this research purpose.

Steps to resolve threat to hypothesis 3: The clustering algorithms which are operated using WEKA tool is applied upon the defect count as the data since defect count is also considered to be one of the modulating factors for the achievement of project success.

V. EMPIRICAL DATA ANALYSIS USING VARIOUS CLUSTERING ALGORITHMS

There exist several clustering techniques for mining of information from the massive amount of data ([11], [12],[13],[14]). They include partitioning methods, hierarchical methods, density based methods, grid based methods, model based methods, methods for high dimensional data and constraint based clustering. Clustering approach segments data from large data sets into groups based on similarity. Table 1 provides details about the sampled type of data chosen for the analysis purpose. Table 1 illustrates projects which are sampled for medium category of complexity. Some of the clustering approaches are used in this work are provided below.

A. K-means Clustering

The K-means algorithm is one of the most popular partitioning clustering techniques which are implemented in 4 steps: Step 1- Partition objects into k nonempty subsets. In this research empirical projects are collected for the analysis purpose indicating the non empty subsets. Step 2- Compute seed points as the centroids of the clusters of the current partition where centroid is the center (mean point) of the cluster. In this research, defect count attribute is considered as centroid to partition the data into two groups namely success of project or failure of projects.

Step 3- Assign each object to the cluster with the nearest seed point. In this investigation, projects having nearest defect count to the arbitrary chosen defect count are assigned to form clusters.

Step 4- Repeat Step 2, stop when no more new assignment [12]. The process is iterative till all the projects are analyzed.

B. Expectation–Maximization (EM) algorithm

An Expectation–Maximization (EM) algorithm is an iterative method for finding maximum likelihood estimates of parameters in statistical models which is used having satisfied the results of K-means method. The two stages involving this approach alternates between performing an expectation (E) and maximization (M) [16]. This research manages E stage by calculating the project grouping probabilities based on success or failure of projects. Further the M stage of the technique is achieved by the calculation of the distribution of defect count parameters.

C. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN is one of the wildly used density based clustering approach. Euclidean distance metric is used on defect count metric to form clustering of projects but it determines number of clusters automatically unlike K-means, find arbitrarily shaped clusters and incorporate a view of outlier.

D. Hierarchical clustering

Forming an initial pair of clusters which recursively considers the option of worth splitting of groups or merging of groups leads to hierarchical clustering technique [15]. In this investigation, software projects are initially merged based on success and failure count using single linkage clustering principle.

E. Cobweb

The Cobweb algorithm compares four different ways of treating a new instance and chooses the best. It generates
hierarchical clustering, where clusters are described probabilistically [16]. In this research approach according to the standard features available WEKA tool, software projects grouping using defect related data which is based on model and evaluation of training set scheme where 66 percent of projects are considered as training set and remaining projects to be deemed as test set.

F. Ordering Points to Identify the Clustering Structure (OPTICS)

The OPTICS technique is an extension of DBSCAN by introducing values that are stored with each data object, an attempt to overcome the necessity to supply different input parameters. Its time complexity is $O(n \log n)$ also same. In our investigation, grouping of software projects is based on density parameter such as epsilon (threshold value) and minpts (minimum number of points). According to the standard WEKA tool, threshold value for the density parameter which is success count of the project is fixed to be 0.9 and minpts is 6.

G. Farthest first

It is modelled on K means that places each cluster centre in turn at the point farthest from the existing cluster centers. This point must lie within the data area. It fastens the clustering in most cases because of the need for less reassignment and adjustment. As per the WEKA tool, empirical data related to defect profile of projects are subjected to training and cross validation to form software project groups to evaluate success and failure of projects.

H. MakeDensityBasedClustererAlgorithm

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas which are required to separate clusters are usually considered to be noise and border points. In this MakeDensityBasedClusterer Algorithm, each cluster is produced by Simple K-means algorithm.

VI. OBSERVATIONAL INFERENCES

These algorithms are implemented using source code in the Weka 3.6.4 version upon which simulations are carried out in order to measure the effectiveness of the algorithms over the empirical datasets. The results are depicted in the following figures.

The major strength of K-means is its relatively scalable and efficient ability in processing large data sets because computational complexity of the algorithm is $O(nkt)$. Additionally, it may produce tighter clusters than other clustering, especially if the clusters are global. However, the major deficiency of this approach is difficulty in comparing quality of the clusters since different initial partitions can result in diverse final clusters.

Figure 1 represents clusters having success (cluster 0) and failure (cluster 1) for the projects based on defect count.

**Fig. 1 Output of K-means algorithm**

<table>
<thead>
<tr>
<th>Scheme:</th>
<th>weka.clusterer:EM-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation:</td>
<td>whatever</td>
</tr>
<tr>
<td>Instances:</td>
<td>25</td>
</tr>
<tr>
<td>Attributes:</td>
<td>6</td>
</tr>
<tr>
<td>Clustered Instances:</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4 (44%)</td>
</tr>
<tr>
<td>1</td>
<td>2 (22%)</td>
</tr>
<tr>
<td>2</td>
<td>3 (33%)</td>
</tr>
<tr>
<td>Log likelihood:</td>
<td>-24.81157</td>
</tr>
<tr>
<td>RATE</td>
<td>SUCCESS: 1</td>
</tr>
<tr>
<td>SUCCESS: 2</td>
<td></td>
</tr>
<tr>
<td>SUCCESS: 5</td>
<td></td>
</tr>
<tr>
<td>Test mode: split 66% train, remaining test</td>
<td></td>
</tr>
</tbody>
</table>

The strength of EM is that the technique ensures the results to converge to a local maximum and may not be feasible for the global maximum. However, the repetitive process yields better global maximum. However, weakness of this approach is its technical difficulties with equating probability value of success/failure. In figure 2, log likelihood of the model has shown number of instances assigned to each cluster whenever the learned model is applied to the data as a classifier. The table entries in figure 2 shows the parameters of numeric attributes (defect count) or frequency counts for the values of nominal attributes (success/failure) and fractional values reveal the soft nature of the clusters produced by EM indicating that any defect can split between several clusters. Figure 2. further infers that the algorithm converges towards fixed point but never actually gets there, However K-means algorithm stops when it reaches a fixed point(do not change from one iteration to the next).

**Fig. 2 Output for the EM algorithm**

<table>
<thead>
<tr>
<th>Scheme:</th>
<th>weka.clusterers:DBScan-0.0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation:</td>
<td>whatever</td>
</tr>
<tr>
<td>Attributes:</td>
<td>6</td>
</tr>
<tr>
<td>Clustered Instances:</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>16 (70%)</td>
</tr>
<tr>
<td>1</td>
<td>7 (30%)</td>
</tr>
<tr>
<td>Unclustered instances:</td>
<td>1</td>
</tr>
<tr>
<td>Test mode: evaluate on training data</td>
<td></td>
</tr>
</tbody>
</table>

The strength of DBSCAN is that unlike K-means it is not required for the prior information about the number of clusters. DBSCAN finds arbitrarily shaped clusters. DBSCAN needs
only two parameters and is insensitive to the ordering of the points since the edge of two different clusters can swap cluster membership if the ordering of the points is altered. However, the weakness in this approach is that it cannot cluster data sets accurately with large differences in densities, due to high dimensionality.

Figure 3. depicts the results of DBSCAN (minpts = 6 and ε = 0.9). K-means method is comparatively better than DBSCAN since with increase in project features, finding appropriate value for epsilon (ε) using DBSCAN algorithm is difficult.

**The strength of hierarchical clustering is more robust to its input parameters, less influenced by cluster shapes and less sensitive to large differing point densities of clusters. It can represent nested clusters. Weakness is that clusters are not explicit in the output of the algorithms and have to be determined by dendrogram representation. Figure 4 indicates the results of dendrogram as tree visualise structure.**

**The strength of DBSCAN is that it automatically adjusts the number of classes in a partition. DBSCAN provides merging and splitting of classes based on category utility and is additionally proficient to do bidirectional search which is distinct from K-means method. However, the weakness of DBSCAN approach is that the probability distribution representation of clusters is expensive whenever it is required for updating information and storage in the form of clusters. Further, increasing number of attributes equally increases the time and space complexity. Figure 6. infers that K-means methods are free of issues that considers the probabilities and independence of clustering and considers only the Euclidian distance into consideration.**

**The strength of farthest first clustering algorithm is that it is a heuristic based method, modelled on K-means which is fast and suitable for large-scale data mining applications. Farthest-first clustering expensive in terms of its computation and storage requirements. Further merging causes the problem for noisy and high dimensional data, these two issues are resolved by using K-means.**
point heuristic based method has the time complexity O(nk), where n is number of objects in the dataset and k is number of desired clusters. Figure 8. Infers that farthest first method is simple, fast. Weakness is most of the local minima are average good only.

Various clustering algorithms are applied on the empirical influencing parameters to predict the success of the projects. Retention of high quality software which can predict the success/failure to analyze it. However, K-means algorithm finds the centers of natural clusters in the training data, which is important factor need to consider for effective defect management.

VII. CONCLUSIONS

Software has become of the critical feature of any application even since three decades. However, the advancement in technology and wide spread need of software has created enforced situation where massive amount of data needs to be managed efficiently. Data mining has proven as one of the most efficient techniques to effectively manage huge amount of data since information is highly expensive and valuable in nature.

Retention of high quality software which can predict the success of projects is one of the core challenges in any industry. Hence, predicting software success using the empirical knowledge of projects can be achieved through effective data mining algorithms.

In this paper, an empirical analysis of several projects developed at various software industries is brought out. In this analysis, defect count is considered to be one of the influencing parameters to predict the success of the projects. Various clustering algorithms are applied on the empirical projects to evaluate the above said objective. Observational inferences indicate that K-means is more efficient than other clustering techniques in terms of processing time, efficiency and reasonably scalable. The experimental results of K-means with rest of the clustering algorithms have ensured its continued relevance and progressively increased its effectiveness as well.

ACKNOWLEDGMENT

The authors would like to acknowledge the software companies involved in this study and the project managers for their invaluable help in providing necessary information for our work under the framework of the Non-Disclosure

REFERENCES


[2] Chih-Ping Wei, Yen-Hsien Lee and Che-Ming Hsu, Empirical Comparison of Fast Clustering Algorithms for Large Data Sets, Proceedings of the 33rd Hawaii International Conference on System Sciences 2000, 0-7695-0493-0/00 $10.00 (c) 2000 IEEE.


[8] Qinbao Song, Martin Shepperd, Michelle Cartwright, and Carolyn Mair, Software Defect Association Mining and Defect Correction Effort Prediction, 0098-5589/06/$20.00 2006 IEEE Published by the IEEE Computer Society.


