

Implementing Clustering Based Approach for Evaluation of Success of Software Reuse using K-means algorithm

Jagmeet Kaur¹ , Dr. Dheerendra Singh²

Assistant Professor CSE/IT, RBCEBTW

Jagmeet.bhangu03@gmail.com

Professor and HOD CSE, SUSCET

professordsingh@gmail.com

Abstract

A great deal of research over the past several years has been devoted to the development of methodologies to create reusable software components and component libraries. But the issue of how to find the contribution of the factor towards the successfulness of the reuse program is still in the naïve stage and very less work is done on the modeling of the success of the reuse. The success and failure factors are the key factors that predict the successful reuse of software. An algorithm has been proposed in which the inputs can be given to K-Means Clustering system in form of tuned values of the Data Factors and the developed model shows the high precision results , which describe the success of software reuse.

Keywords: Kmeans, Reuse and Machine learning.

1. Introduction:

Software Reuse and Success factors: - Systematic reuse is generally recognized as a key technology for improving software productivity and quality (Mili et al. 1995), possibly with a higher payoff than process improvement or process automation(Boehm 1993). *Software reuse* is the process whereby an organization defines a set of systematic operating procedures to specify, produce, classify, retrieve, and adapt software artefacts for the purpose of using them in its development activities. In the April 2002 TSE article *Success and Failure Factors in Software Reuse* [1], Morisio et.al. sought key factors that predicted for successful software reuse. Their data came from a set of structured interviews conducted with project managers of 24 European projects from 19 companies in the period 1994 to 1997. Those projects were trying to achieve company-wide reuse of between one to a hundred assets. Nine of those 24 projects were judged by their respective managers as failures. Morisio et.al. employed a well-designed interview process to collect a wide set of project attributes (for a complete listing

of those attributes, see the appendix). There is much that is exemplary in the approach taken by Morisio et.al. For example, their data collection method is well-documented.

The main causes of failure was a lack of commitment by top management, or non-awareness of the importance of those factors, often coupled with the belief that using the object-oriented approach or setting up a repository seamlessly is all that is necessary to achieve success in reuse[3].

Conversely, successes were achieved when, given a potential for reuse because of commonality among applications, management committed to introducing reuse processes, modifying non-reuse processes, and addressing human factors.

While addressing those three issues turned out to be essential, the lower-level details of how to address them varied greatly: for instance, companies produced large-grained or small-grained reusable assets, did or did not perform domain analysis, did or did not use dedicated reuse groups, used specific tools for the repository or no tools. As far as these choices are concerned, the key point seems to be the sustainability of the approach and its suitability to the context of the company.

There are three types of Factors that are considered here as:

- High Level Control variables
- State Variables
- Low Level Control Variables

Table 1. High Level Control Variable

Attributes	Response for Reuse	
Top management commitment	Yes	No
Key Reuse Roles	Yes	No
Reuse Process	Yes	No
Non Reuse Process Modified	Yes	No
Human Factors	Yes	No
Repository of assets	Yes	No

Note that all 23 projects seen in this data set used a repository; i.e. this data set could never be used to refute claims that a repository is useless. Nevertheless, like Morisio et.al., we believe that reuse products have to be kept in some sort of repository to enable reuse.

State Variables:

The state Variables, these are the attributes over which a company has no control.

Table 2. State variables

Attributes			
No. Of Staff	Large	Medium	Small
Overall Staff	Large	Medium	Small
Type of Production	Product-family	Project related	Isolated
Product type	Embedded	Standalone	Embedded in Process
SP Maturity Level	CMM Level 3	ISO 9001	Low
Application Domain	TLC	Manufacturing	ATC
Size of Baseline	Large	Medium	Small
Staff Experience	High	Medium	Low

Low Level Control Variables:

These are the specific approaches to the implementation of reuse

Table 3. Low level control variable

Attributes		
Reuse Approach	Loosly Coupled	Tightly Coupled
Domain Analysis	Yes	No
Configuration Management	Yes	No
Rewards Policy	Yes	No

Kmeans Clustering Algorithm:

Clustering (or cluster analysis) aims to organize a collection of data items into clusters, such that items within a cluster are more "similar" to each other than they are to items in the other clusters. This notion of similarity can be expressed in very different ways, according to the purpose of the study, to domain-specific assumptions and to prior knowledge of the problem. Clustering is usually performed when no information is available concerning the membership of data items to predefined classes.

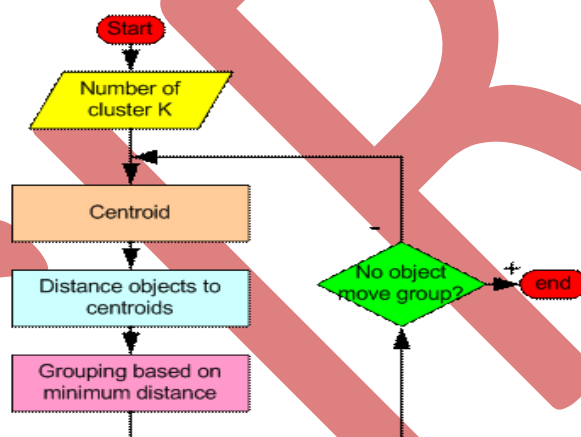


Fig.1. Flowchart of Kmeans Algorithm

2. Problem Formulation:

A great deal of research over the past several years has been devoted to the development of methodologies to create reusable software components and component libraries. But the issue of how to find the contribution of the factor towards the successfulness of the reuse program is still in the naive stage and very less work is done on the modeling of the success if the reuse. Our approach, for evaluation success of software reuse, is based on software models and metrics. As the exact relationship between the attributes of the reuse success is difficult to establish so a Clustering based approach could serve as an economical, automatic tool to generate ranking of reuse success by formulating the relationship based on its training.

3. PROPOSED METHODOLOGY

The success of software reuse can be measured by following steps.

I. Selection of Dataset and Factors:

The datasets are generated from the interviews and questionnaires with the organization, related to the software to be developed. There are three types of factors that are considered are:

- a) High level control variables
- b) State variables
- c) Low-level control variables

Collect or create the relevant data:

Collect the relevant data from the dataset, which are required for the success of software reuse.

II. Perform clustering:

The Clustering is an approach that uses software measurement data for analyzing software quality. In this step, K-Means clustering algorithm is used for partitioning the data into different level of reusability value based on the structural metric values as K-means is the well known approach that classify data into different K groups where K is a positive integer, based on the attributes or some features. Grouping of data is done on the basis of minimizing sum of squares of distances between data and their cluster centroid.

III. Comparison :

The comparisons are made on the basis of the least value of *Accuracy, Precision, and Recall* values. In case of the two-cluster based problem, the confusion matrix has four categories: True positives (TP) are modules correctly classified as faulty modules. False positives (FP) refer to fault-free modules incorrectly labelled as faulty modules. True negatives (TN) correspond to fault-free modules correctly classified as such. Finally, false negatives (FN) refer to faulty modules incorrectly classified as fault-free modules as shown in Table 1.

TABLE 1 CONFUSION MATRIX OF PREDICTION OUTCOMES

Predicted Project	Real Data Value of Project Status	
	Success	Failure
Success	TP	FP
Failure	FN	TN

With help of the confusion matrix values the precision and recall values are calculated described below:

Precision:

Precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class). The equation is:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Recall:

Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. sum of true positives and false negatives, which are items which were not labelled as belonging to the positive class but should have been.) The Recall can be calculated as:

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

Accuracy:

The percentage of the predicted values that match with the expected values for the given data. The best system is that having the high *Accuracy, High Precision and High Recall* value.

IV. Conclusion:

The conclusions are made on the basis of the comparison made in the previous section.

4. RESULTS & DISCUSSION

The implementation of algorithm is done in open source tool known as WEKA 3.2

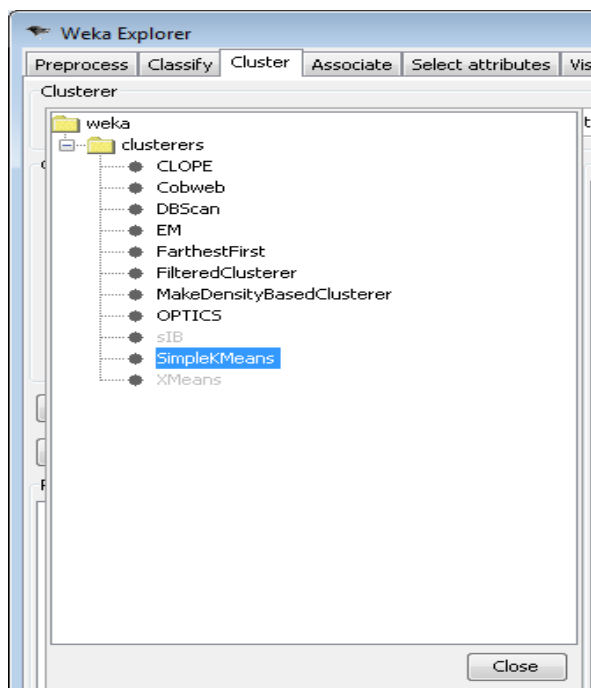


Fig. 1. Snapshot of the Kmeans also used.

First the dataset is loaded in WEKA environment. The metadata view of the input dataset is shown in the figure 2. The dataset includes all the factors or variables that are considered in the reusability of software such as:

- High level control variables
- State variables
- Low level control variables

The screenshot shows the 'Metadata view' of a dataset in Weka Explorer. It displays a table with 15 rows and 6 columns. The columns are: 'id', 'Project ID', 'Software Staff', 'Overall Staff', 'Type of Software Production', and 'Software and Product'. The 'id' column contains values from 1 to 15. The 'Project ID' column contains values like '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15'. The 'Software Staff' column contains values like '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15'. The 'Overall Staff' column contains values like '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15'. The 'Type of Software Production' column contains values like 'product-family', 'product', 'isolated', 'process', 'product-family', 'product', 'product-family', 'product', 'product-family', 'product', 'product-family', 'product', 'product-family', 'product', 'product-family'. The 'Software and Product' column contains values like 'product', 'product', 'alone', 'alone', 'alone', 'alone', 'process', 'product', 'product', 'product', 'product', 'product', 'product', 'product', 'product'. The 'ST status' column contains values like 'high', 'high', 'middle', 'middle', 'middle', 'middle', 'low', 'high', 'middle', 'middle', 'middle', 'high', 'high', 'middle', 'middle'.

Fig.2. View of the input dataset

Thereafter, Kmeans clustering algorithm is applied on the dataset. In the Kmeans clustering algorithm the value of K is set to 2 means the total number of clusters that Kmeans clustering algorithm going to generate will be 2.

The Text view of Cluster Assignment is shown in Fig.3. The figure shows that the 15 examples are assigned to cluster 0, 9 examples are assigned to cluster 1.

Cluster Model

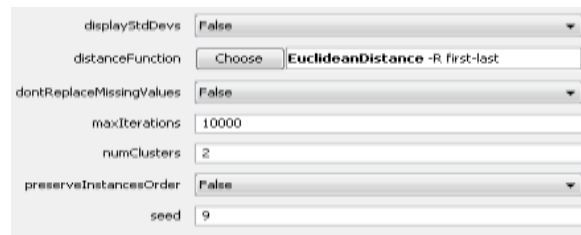
Cluster 0: 15 items

Cluster 1: 9 items

Total number of items: 24

Fig.3. Text View of Cluster Assignments

When Kmeans Algo applied Various parameters used as:



displayStdDevs	False
distanceFunction	Choose EuclideanDistance -R first-last
dontReplaceMissingValues	False
maxIterations	10000
numClusters	2
preserveInstancesOrder	False
seed	9

Fig.4.

4. Conclusion

Reuse based approaches emphasize cost reduction as a means of increasing productivity. From an accounting perspective there are different ways of achieving this. One way is the amortization of the development and maintenance cost of assets over multiple projects. Another way is the avoidance of cost in later projects through the use of results of earlier projects. As evidenced by the results, Kmeans Clustering algorithm is proved to be best as compared to the Multi-preceptron algorithm for evaluating the success of software reuse in an organization. It is concluded that for non linear and complex engineering applications involving decision and analysis by and large Kmeans clustering is an efficient technique.

5. References:

1. Alpaydin, Ethem. *Introduction To Machine Learning*. Cambridge, Massachusetts: MIT Press. 2004.
2. Lazlo, Michael, and Mukherjee, Sumitra. "A Genetic Algorithm Using Hyper-Quadrees for Low-Dimensional K-means Clustering." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 28. No. 4. April 2006. 533-543.
3. M. Morisio, M. Ezran, and C. Tully, "Success and failure factors in software reuse," *IEEE Transactions on Software Engineering*, vol. 28, no. 4, pp. 340-357, 2002.
4. Mili, H., Mili A, Yacoub, S, Addy E. (2002). *Reuse-Based Software Engineering: Techniques, Organization, and Controls*. John Wiley & Sons.
5. Zhong Wei, et al. "Improved K-Means Clustering Algorithm for Exploring Local Protein Sequence Motifs Representing Common Structural Property." *IEEE Transactions on Nanobioscience*. Vol. 4. No. 3. September 2005. 255-265.
6. Zhao, Tong, Nehorai, Arye, and Porat, Boaz. "K-Means Clustering-Based Data Detection and Symbol-Timing Recovery for Burst-Mode Optical Receiver." *IEEE Transactions on Communications*. Vol. 54. No 8. August 2006. 1492-1501.
7. Hamerly, Greg, and Elkan, Charles. "Learning the k in k-means." Retrieved from the World Wide Web at http://books.nips.cc/papers/files/nips16/NIPS2003_AA36.pdf through Google Scholar on November 25, 2006. Found.
- 8.