ISSN:0975-9646

A Proposal to Enhance A-KNN Clustering Method

Kamaljeet Kaur[1]
M.Tech (CSE) Student,
[1]Department of Computer Science and Engineering,
BBSBEC Fatehgarh Sahib,
Punjab (India)

Dr. Satwinder Singh[2]
Assistant Professor,
[2]Department of Computer Science and Engineering,
BBSBEC Fatehgarh Sahib,
Punjab (India)

Abstract: **To maintain the quality of software and to better understand it, software architecture is decomposed. Software decomposition is done by various clustering methods. Each method provides different results of clustering on datasets. This paper presents the review of various clustering methods with A-KNN method in terms of efficiency and accuracy. It also proposes the way to enhance AKNN clustering technique so that each component of the software can be clustered properly. This enhancement will be done on Euclidean distance by normalization method.**

Keywords— **Clustering, A-KNN, Euclidean distance, Normalization**

I. INTRODUCTION

Clustering is an unsupervised classification method aims at creating groups of objects, or clusters, in such a way that objects in the same cluster are very similar and objects in different clusters are quite distinct. The key concept of clustering is to group similar things together to form a set of clusters, such that intra-cluster similarity (cohesion) is high and inter-cluster (coupling) similarity is low. Refactoring or clustering aims to improve various factors of software quality such as understandability, maintainability etc. It also makes developers program faster and helps in finding bugs. Since refactoring changes the internal structure of the code, the internal quality attributes such as coupling and cohesion will change. So designers should try to maximize cohesion and minimize coupling Coupling refers to the interdependencies among software modules and Cohesion is the functional strength of a module. Software should have low coupling and high cohesion. Cluster analysis has been widely used in numerous applications, including market research, pattern recognition, data analysis, and image processing. Clustering is a method used to group similar documents, but it differs from categorization of documents results. Various clustering methods are available for making clusters such as Single Linkage algorithm (SLINK), Weighted Pair-Group Method using Arithmetic averages (WPGMA), K- means, Adaptive K-Nearest Neighbor (A-KNN) algorithm etc. In this paper, we present an approach to enhance the software refactoring method.

The objective is to provide a way so that the clustering algorithm that is A-KNN is to be enhanced to cluster the various components properly.

II. LITERATURE REVIEW

Chung-Horng Lung, Samuel Ajila et al [1] used two Hierarchical Agglomerative Clustering (SLINK and WPGMA) and adaptive K-nearest neighbour algorithms. SLINK, WPGMA and A-KNN methods were applied on the requirement attribute dataset. The performance of A-KNN is compared with SLINK and WPGMA. The results show that A-KNN algorithm is competitive with these two techniques.

Abdulaziz Alkhalida, Mohammad Alshayebb and Sabri Mahmoudb,[2] had proposed a Adaptive K-Nearest Neighbor (A-KNN) algorithm to perform clustering with different attribute weights. The technique is used to refactoring at the function/method level. This is achieved by identifying ill-structured software entities. They also compare the proposed technique with three function-level clustering techniques SLINK, CLINK and WPGMA. Refactoring reduce the complexity of the software by changing internal structure of the software without effecting on external functionality.

Abdulaziz Alkhalid, Duo Liu, et al[3] had explained that, the software designer have two problems while applying the clustering techniques :(1) determination of number of clusters (2) determination of specific cluster or software module for some highly coupled or fuzzy components. This paper presented approach for finding solution to these two issues. Fuzzy C-means clustering with three hierarchical agglomerative clustering techniques and the adaptive K- nearest neighbor algorithm is used on two industrial software systems. Results of this approach shows A-KNN is competitive than other three agglomerative clustering techniques and FCM provide valuable information, which is helpful to handle the two issues for clustering techniques [2].

Mohammad Alshayeb and Sabri A. Mahmoud, et al [4] had explained that refactoring is used to increase the software quality and reduce the software complexity. They proposed a technique to maintain the balance between cohesion and coupling by pattern recognition techniques at class level. Clustering techniques were used and compared the results. This paper proposed two approaches. First one is “software refactoring at the class level with a fixed number of classes” and the second is “software refactoring at the class level using an adaptive number of classes”.

Mark Shtern and Vassilios Tzerpos, [8] had explained that many different software clustering algorithms have been developed with its properties, qualities and restrictions. These algorithms used for specific software systems, but the question is how to choose a clustering algorithm that is best suited for a particular software system. They provide a method for the selection of a software clustering algorithm for particular needs.

III. METHODOLOGY

Various research papers concluded that AKNN method is better in terms of accuracy and efficiency than other clustering methods. The methodology to make enhancement in A-KNN method is explained in this section. The working of A-KNN method is stated as below:
In A-KNN clustering algorithm, probability of the most relevant function is calculated and using Euclidean distance formula the functions are clustered. Before the proposed A-KNN [2] algorithm is applied, the code is parsed to extract the intended attributes. These attributes are organized into an entity–attribute data matrix. The entities are the source code statements and the attributes are the data and control variables. Based on this entity–attribute matrix, the similarity matrix will be extracted. This is done by calculating the similarity between each entity and all other entities.
The resemblance coefficient is used to measure the similarity. A resemblance coefficient can be qualitative or quantitative. In the proposed approach, the coefficient resemblance matrix will be calculated once. It will be used in all further iterations. The algorithm starts by considering each entity as a cluster (i.e. each entity will be labeled with a unique identifier representing the cluster identity). In the second iteration, for K = 3, the algorithm selects the three nearest neighbors to the entity that will be clustered, and then checks their labels. If at least two out of the three clusters have the same label, the algorithm will label the current entity with the same label of those two entities. If the three entities have different labels, the algorithm will label the current entity with the same label of the closest entity (NN). The algorithm repeats the clustering process until no more changes occur in the clustering tree. Then the algorithm outputs one cluster at the highest level of the hierarchy.
In the shown algorithm, the calculation of the similarity matrix is calculated once. Thus, the number of computations is decreased significantly. A-KNN has an advantage over other techniques because it reduces the amount of required computations. In A-KNN, there is no need to calculate the distance between two clusters directly; which consumes significant amounts of computation. A-KNN depends mainly on the original similarity matrix between entities. Consequently, the amount of computation in each clustering step is less. The similarity matrix is calculated only once and the same matrix is used in all successive steps.

The A-KNN algorithm works as under:

```
Algorithm: A-KNN for K=3
Input: Entity-Attribute matrix (n*m); n: number of entities; m: number of attributes
Output: Hierarchy of clusters

Steps:
  1. Assign each entity to a single cluster and label each cluster. (L ($C_i$)= unique label, i belongs to {1,2,……n}; where n is the number of entities)

  2. Calculate the similarity matrix(n*n)   (Calculate the similarity between each cluster and all other clusters using the formula(1) and fill the (n*n) matrix)

  3. While the number of clusters is more than one Do
       Find the most similar pair of clusters {$C_a$ , $C_b$} , {$C_d$ , $C_e$} , {$C_f$ , $C_g$} Where: Coeff ($C_a$,$C_b$) > ($C_d$,$C_e$) > ($C_f$,$C_g$)

       If L ($C_a$) = L ($C_d$) = L ($C_f$)   (The same cluster)
       Then
         If L ($C_e$) = L ($C_g$) Then
           L ($C_a$) = L ($C_e$) (Merge clusters of $C_a$ and $C_e$ in one cluster)
         Else
           L ($C_a$) = L ($C_b$) (Merge clusters of $C_a$ and $C_b$ in one cluster)
       Else
         L ($C_a$) = L ($C_b$) (Merge clusters of $C_a$ and $C_b$ in one cluster)

  4. End While

  5. Return "Hierarchy of Clusters".
```

This method is mainly based on the distance calculated between the components. Euclidean Distance is defined as:

$$D(x,y)= \sqrt{} (x_2-x_1)^2 + (y_2-y_1)^2$$

The problem exists when all the components in the software are not clustered properly. To make them clustered properly, the Euclidean distance will be enhanced by normalization method. This method will minimize the distance between the functions so that they will be closed to each other to be clustered. AKNN technique will be applied based on function point analysis (FPA). Function Point Analysis is a method to measure the functional size reflects the amount of functionality that is relevant to and recognized by the user in the business. The unit of measurement is "function points". Function point can be calculated by following formula:

FP = Unadjusted FP * Value adjustment factor

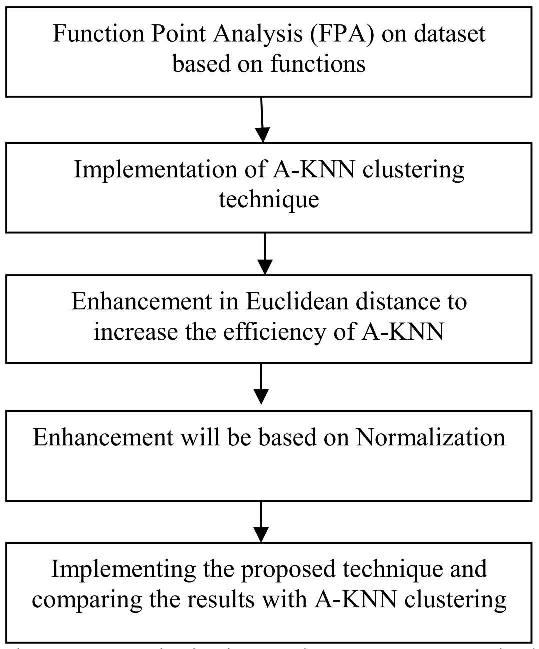
Function Point Analysis (FPA) on dataset based on functions
Implementation of A-KNN clustering technique
Enhancement in Euclidean distance to increase the efficiency of A-KNN
Enhancement will be based on Normalization
Implementing the proposed technique and comparing the results with A-KNN clustering

Fig 1. Proposed criteria to enhance A-KNN method

The enhancement will be done by following the above proposal. In the end, the results of A-KNN method and the enhanced A-KNN method will be compared to check the variation between both methods.

IV. Discussion

1. The main objective of this paper is to make the A-KNN clustering technique more efficient so that all the components can be clustered properly.
2. The enhancement will be done in order to increase the performance of A-KNN method. Normalization will be done in order to make enhancement.
3. This normalization will reduce the distance between the two functions to make them closer to each other to be clustered in a particular cluster. This will make the clustering technique more efficient.

References

[1] Chung-Horng Lung, Samuel Ajila, Abdulaziz Alkhalid, " Software Decomposition Using Adapative K-Nearest Neighbour Algorithm", 26th IEEE Canadian Conference Of Electrical And Computer Engineering (CCECE), 2013.

[2] Abdulaziz Alkhalida, Mohammad Alshayebb, Sabri Mahmoudb," Software refactoring at the function level using new Adaptive K-Nearest Neighbor algorithm",2010.

[3] Abdulaziz Alkhalid, Duo Liu, Chung-Horng Lung, Samuel Ajila, "Software Architecture Decomposition Using Clustering Techniques", IEEE 37th Annual Computer Software and Applications Conference, 2013.

[4] Mohammad Alshayeb, Sabri A. Mahmoud, Abdulaziz Alkhalid, "Software Refactoring at the Class Level using Clustering Technique", 2011

[5] Marzia Zaman, Xia Xu, Chung-Horng Lung, "Software Architecture Decomposition Using Attributes", 2005.

[6] Chung-Horng Lung, Marzia Zaman, " Applications of Clustering Techniques to Software Partitioning, Recovery and Restructuring ", 2004.

[7] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource ReSerVation Protocol" (RSVP), 1997.

[8] Mark Shtern and Vassilios Tzerpos, "Methods for Selecting and Improving Software Clustering Algorithms", 2014.

[9] Anwiti Jain, Anand Rajavat , Rupali Bhartiya, " Design, Analysis and Implementation of Modified K- Mean Algorithm for Large Data-set to Increase Scalability and Efficiency",2012.

[10] Timothy C. Lethbridge and Robert Laganiere, "Object-Oriented Software Engineering".

[11] Len Bass, paul clements and Rick Kazman, "Software Architecture in Practice".

[12] Shifa-e-Zehra Haidry and Tim Miller, " Using Dependency Structures for Prioritisation of Functional Test Suites"IEEE, 2012

[13] S.raju , G. V. Uma , "Factors Oriented Test Case Prioritization Technique in Regression Testing using Genetic Algorithm" European Journal of Scientific Research ISSN 1450-216X Vol.74 No.3, 2009

[14] Ryan Carlson , Hyunsook Do, Anne Dento, "A Clustering Approach to Conference on Software Maintenance (ICSM)", 2011

[15] Timothy C.Lethbridge,Robert Langaniere, "Object oriented software Engineering practical software development using UML and java", Tata McGraw Hill Eduction Private Limited.

[16] Gregg Rothermel,Roland H.untch, Chengyun Chu "prioritize the test case for Regression testing" *IEEE Transactions on Software Engineering*, 2011

[17] Nguyen Van Vy. Nguyen Vietnam – curriculum software engineering, *IT Facility – Hanoi University of Technology,* VNU, 2006.

[18] Jehad Al Dallal- A Design-Based Cohesion Metric for Object-Oriented Classes- *World Academy of Science, Engineering and Technology International Journal of Computer, Information, Systems and Control Engineering,* 2007.

[19] Payal Khurana & Puneet Jai Kaur- Dynamic metric at design Level-*International Journal of Information Technology and Knowledge Management* July-December 2009, Volume 2, No. 2, pp. 449-454.

[20] Kuljit Kaur and Hardeep Singh-An Investigation of Design Level Class Cohesion Metrics-*Department of Computer Science and Engineering, Guru Nanak Dev University, India* January 2012.

[21] N.Sasirekha, A.Edwin Robertand Dr.M.Hemalatha-Program slicing techniques and its applications -International Journal of Software Engineering & Applications (IJSEA), *Vol.2, No.3, July 2011.*

[22] S. Revathi, Dr.T.Nalini, "Performance Comparison of Various Clustering Algorithm", *International Journal of Advanced Research in Computer Science and Software Engineering* , Volume 3, Issue 2, February 2013