Thisara - GA Optimized Nearest Neighbor Classification Framework

L.P.Priyabashitha, A.A.B.Amarasinghe, M.S.G.Vithana, A.P.D.S.K.Gunarathne,
Chulaka Gunasekara and A. Shehan Perera
Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka

Abstract – **Information plays such an important role in almost every human life. We have plenty of data available and for the most effective and efficient use, we need to extract information from them. Data classification comes to the top in that scenario, where it will indicate its value in terms of business assets. For the problems exist in classification field, such as data mining and other third party applications which involve classifications/predictions in their domains, use their own procedures of classifying the data available, hence slowing down the efficiency due to increase of development time which ultimately results in high development cost. Thisara addresses the above issue by providing a common platform for the classification problems to be implemented upon, without worrying about the underlying complexity of the application, hence reduce bulkiness and provides a stable framework which offers a high performance in real time operation of classifications.**

Keywords – **Clustering, Classification, Normalization, GA Optimization**

I. Introduction

Data classification plays a major role in today's world in the context of determining the importance and the value of data present in various forms. In real life, application developers encounter many situations where they require classification algorithms for their systems. Some key areas lies in data mining field which are inherently complex [1]. Most data mining applications require the help of a Nearest Neighbor application to classify the data retrieved by the data mining module. This will increase the complexity of the data mining application unnecessarily. Therefore it is very useful to have a supportive Nearest Neighbor framework to reduce the bulkiness and the complexity of the data mining application source code.

Thisara would replace the need for developing background classification modules in larger business applications, while providing additional functionalities which in most cases are not considered specifically in application development.

The poor performance and inefficiencies associated with classifications are reduced in the new system by introducing clustering prior to the classification execution, so that scanning of all the data available is avoided and, analyzing only interested data is preferred, hence reduced the time spent for the task, resulting high performance.

To maintain higher accuracy in classification, the relative importance of the attributes and the parameters of the classification algorithm are optimized using Genetic Algorithm in the system [2]. The optimization plays a major role in this context, since the accuracy of the algorithm is quite considerable.

Thisara consists of three major modules, namely Clustering module, Classification module and GA module.

A. Clustering Module

The module categorizes the given n-dimensional data in such a manner that similar data points lie in the same group [3]. A suitable similarity measurement depending on the problem domain would use for this purpose. The clustered data would provide faster access on interested data items for a particular classification problem, even though the available data space is huge.

B. Classification Module

The module predicts the class of a given data item in which the class is unknown. The most nearest neighbors of the given data item would retrieve searching through the clustered data set and then it would be classified in to the most occurred class label of nearest neighbors.

C. GA Module

The module optimizes the operation of the classification for a given data item providing the most suitable weightings of the attributes and k parameter of the classification algorithm. The module uses an initial population from the available data space and evaluate for an exact or near exact solution to a preferred extent.

This paper discusses the research carried out to introduce a new classification framework which operates in high performance. Section II describes the key research areas underlying the implementation of the framework. Section III focuses on the methodology used for the proper implementation of the system and section IV explains the implementation details of the framework. The results gained for the operations associated with the framework are analyzed and discussed under section V.

II. Theory

This section describes the key research areas focused on building up the system. Only the basic theoretical concepts behind those techniques are explained briefly, to up front the implementation of the system.

A. CF Tree

The main data structure uses to cluster the incoming data items in the system. It is a height balanced tree with two parameters; branching factor (B) and threshold (T). The least cluster represents by an entry of a leaf node [4].

The information of clusters is stored in Clustering Feature (CF), which is a triple;

$$CF = (N, \overrightarrow{LS}, SS)$$

$$\overrightarrow{LS} = \sum_{i=1}^{N} x_i$$

$$SS = \sum_{i=1}^{N} x_i^2$$

Where, N is the number of data points in the cluster, LS is the linear sum of the N data points and SS is the square sum of the data points. Figure 1 illustrates the structure of a typical CF Tree.

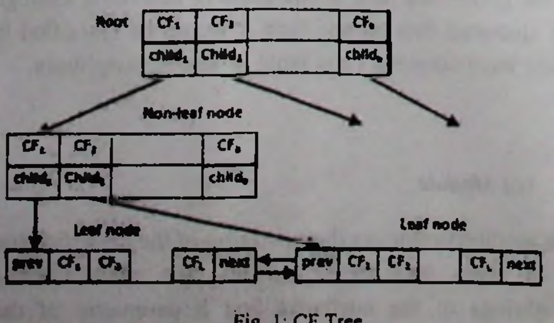

Fig. 1: CF Tree

The Radius (R) and the Diameter (D) of a cluster are calculated as below, where x_0 is the centroid of the data items in that particular cluster.

$$\overrightarrow{x_0} = \frac{\sum_{i=1}^{N} \overrightarrow{x_i}}{N}$$

$$R = \left(\frac{\sum_{i=1}^{N} (\overrightarrow{x_i} - \overrightarrow{x_0})^2}{N} \right)^{\frac{1}{2}}$$

$$D = \left(\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (x_i - x_j)^2}{N(N-1)} \right)^{\frac{1}{2}}$$

B. k – Means

k - Means is an unsupervised clustering algorithm to cluster a given data set by grouping them into k number of clusters [5].

The squared error ($J(c_k)$) between the centroid and the data points in cluster c_k and, objective function ($J(c)$) which is the sum of the squared error over all the k clusters is defined as below, where C_j is the cluster center.

$$J(c_k) = \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

$$J(c) = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

C. k – Nearest Neighbor

k – Nearest Neighbor (k-NN) is a supervised learning algorithm where a new object is classified by the majority of the votes of its k nearest neighbors [6]. The performance of the algorithm depends on the characteristics of the data set.

D. Genetic Algorithm

GA is a search technique use to find exact or near exact solutions to a problem, which is a type of evolutionary algorithm, but in contrast, it is not designed only to solve specific problems, but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop mechanisms of natural adaptation into computer systems [2].

Algorithm is started with a set of solutions called population. It proceeds with generating populations applying selection, mutation and crossover operations to reach the best optimal solution.

III. Methodology

The following section discusses the methodology followed in order to implement the project.

The goal of Thisara is to provide a common classification platform with higher accuracy and lower classification time. It is acquired by integrating clustering and genetic algorithm [7] to the system. Therefore the methodology of Thisara can be described by using three key areas.

1. Thisara is built as a framework to provide a common classification platform to third party developers and data mining applications to work with. It provides common and widely used operations to the user and user can customize the parameters and override method implementations to customize the framework to suit user's needs.

2. The most important feature of the Thisara framework is the constant execution time for any classification despite of the size of the data set. This is achieved by implementing a clustering [8] mechanism prior to the classification.

    - It reduces the number of data points to be accessed for the classification module

      Since the clustering module clusters initial data set and categorizes them based on a suitable similarity measurement, classification module is fed only with the data items which belong to the same cluster or neighboring clusters of the test sample, hence cut down significant amount of data items to be accessed for the classification.

    - It does not compromise the accuracy of the classification

      Classification module only interested in the nearest neighbors of a given test data item. Since the accessing of cluster/clusters contain(s) the most nearest neighbors (similar data) of the test item, the accuracy of the classification result preserves even though there is a considerable reduction of training data.

3. Thisara framework contains GA module for the purpose of gaining higher accuracy when classifying. The GA module would optimize the parameters in the classification algorithm [9]. The most important parameters for the classification are the individual significance or the weights of the conditions of situations. GA module is used to analyze the present significance values and output with the original result and optimize them to have a minimum error. Genetic algorithm is integrated by using Java Genetic Algorithm Package which is an existing library.

IV. Implementation

The system comprise with three major modules, namely clustering, classification and GA. The implementation details of the system will be discussed here along with these three modules taking in to consideration.

The system undergoes a two step process.

- Training phase
- Classification phase

The training phase is implemented by deviating from the classical implementation of training, which is done by simply adding training data samples into a knowledge base. The clustering module and GA module are mainly responsible for the implementation of training phase. The concept of clustering was introduced for the purpose of improving performance of the classification and GA is used for optimizing the operation in order to gain high accuracy.

A. Clustering Module

The module implements three major functionalities mainly.

- Building the cluster tree

  The training data is fed to the system and build up the CF Tree to cluster the data items. At the beginning, an empty CF tree is built with only the root and single child to prevent the root from becoming a leaf node. As data is fed to the tree, data is added to the first cluster in the child of the root. When a data node which violates the threshold limitation of the cluster appears, the cluster splits. Data nodes which were there on the cluster and the data node that appears last get added to the respective closest cluster.

- Merging the clusters
  Splitting of the clusters creates very sparsely populated clusters. This makes the clustering inefficient and increase the load on the system memory. To prevent the execution of the merging phases after every split operation, the following procedure is followed.
  - Identifying two sub clusters of a node in which inter cluster distance is least, calculated by brute force checking of distances between each and every pair of sub clusters.
  - If the two clusters are the clusters that are the result of the immediate previous split operation then the merge operation exits, otherwise check the radius of the merged cluster.
  - If the radius of the merged cluster is lower than the threshold value then the two clusters get merged.

- Refining the clusters
  The global clustering takes place in this step, implemented using the k – Means clustering algorithm.
  The algorithm traverses through the CF Tree from its root till the leaf nodes and cluster each leaf node one at a time. The clustering process scans through all the leaf node entries of the clustering leaf node and listed down each and every data item in a separate list. These data items are subjected to the clustering process and clustered items are then added to the CF Tree as it is.
  The refining phase doesn't change the number of clusters the data points are resides in, but the centroids of the initially formed clusters may get changed after performing the refining operation on those clusters.

Figure 2 illustrates a cluster generated by Thisara clustering module.

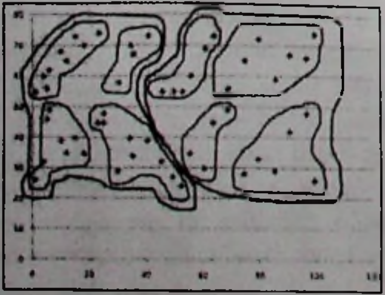

Fig. 2: A cluster generated by clustering module

B. GA Module

Potential solutions for weights value represent as chromosomes in GA. Chromosomes evolve using multiple genetic operations to obtain best solution. Evolving process happen until best fittest chromosome's fitness value in generations would reach an unaltered value.
The module implements two major functionalities primarily.

- Optimize weights value

Chromosomes values initialized randomly and evolve by using genetic operations. Sample data sets extracted from given dataset are classified using the weights produced by each chromosome. The process continues until the best fitness values in consecutive generations reach an unaltered value. The chromosome corresponding to the best fitness value is considered as the optimized weights for given K value.

- Optimize k-value

Final generation evolve by GA when k=1 taken as the initial sample chromosome when k=3. This step would reduce the period of time to seek an optimized weights value for each K value. The process repeated until upper bound of k value provided by the system/user. The best fitness value in fittest chromosome for each K value would record in a list and Optimum K value for the dataset retrieve for the list considering corresponding fitness values.

The classification phase is responsible for classifying test data and determining the respective class label. The classification module provides the functionality for the operation of classification in the system.

C. Classification Module

The module is implemented using k–Nearest Neighbor classification algorithm. The k–NN algorithm classifies a data item by finding the simple majority of the result appears in its nearest neighbors.

The nearest data items would retrieve by traversing the CF Tree and getting the clusters in which most nearest neighbors of the given test data item lies.

The steps of classifying a data item are described below.

- Determine parameter k (number of nearest neighbors)

- Calculate the distance between the test item and retrieved nearest neighbor data items
- Sort the distance and determine the nearest neighbors based on the kth minimum distance
- Gather the category (result / class) of the nearest neighbors
- Use simple majority of the category of nearest neighbors as the prediction value of the query instance

The parameter k, is configurable by the user, or the user could give the priority to the system to determine, where the GA module would provide the best fitted value. The similarity measurement is also configurable and the framework provides defaults as Euclidean distance and Manhattan distance.

V. Results

Two major types of tests were carried out to assess the functionality and evaluate the performance of the framework, as follows.

A. Effect of Clustering
B. Effect of GA

All the tests were carried out and obtained the results using sample data sets extracted from the KDD network intrusion detection data set [10].

A. Effect of Clustering

The framework operates in two major operational modes.

1. Memory mode: relies on the available system memory
2. Database mode: based on a supporting database

The performance in each mode inspected separately for the tests carried.

1. Memory mode (data is stored in the RAM)

In system training, introduction of clustering to the framework makes the framework perform much slower than the classical k-NN implementation.

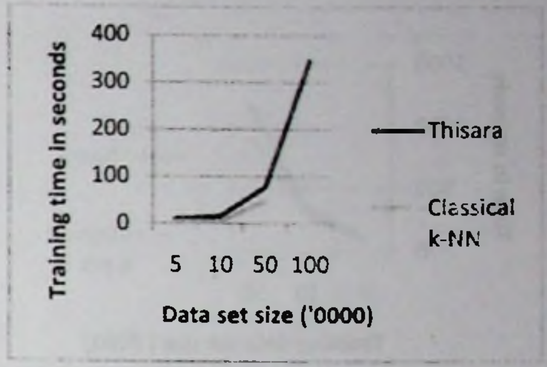

Fig. 3: Training time comparison

This happens as classical k-NN algorithm does not involve any specific operations for training but, Thisara has a specific tree building phase to be completed. This takes processing power and time. Therefore training phase of the Thisara framework is much slower than classical k-NN algorithm.

But the performance gain can be seen clearly in the classification phase. While classical k-NN shows exponential growth in the classification time, Thisara shows almost constant time operation.

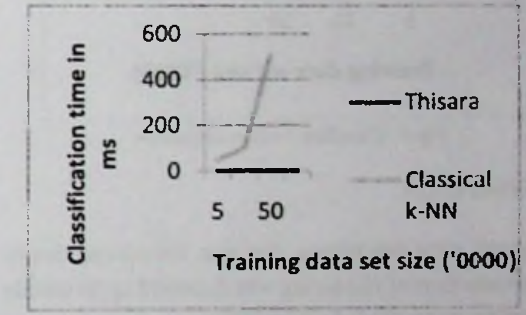

Fig. 4: Classification time comparison

This is due to the fact that the classical k-NN sweeps through whole data set to find the nearest neighbors, but Thisara examines only a small fraction of the whole data set to find them.

- Database mode

Same as the memory mode, Thisara performs slower than the classical k-NN implementation in training phase.

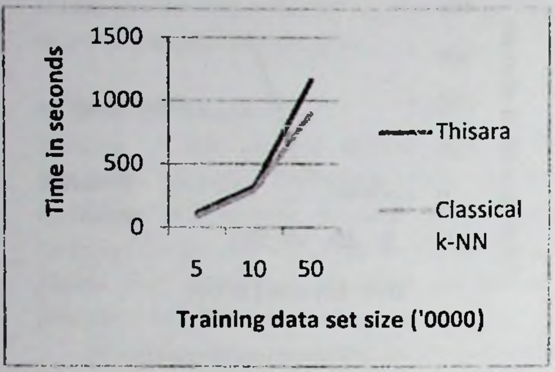

Fig. 5: Training time comparison

But in the classification phase, Thisara outperforms the classical k-NN as in the memory mode.

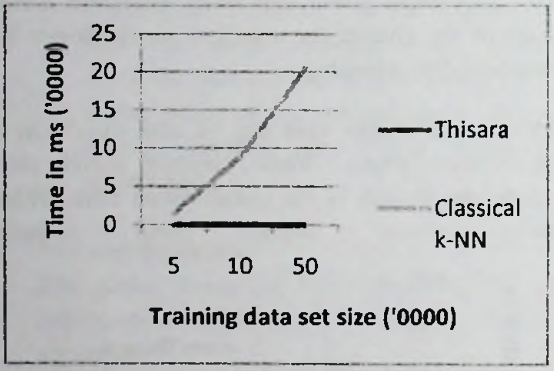

Fig. 6: Classification time comparison

1. Effect of GA

The small error percentage that was introduced due to the introduction of clustering was removed up to certain amount by using the GA optimization.

Size of training data: 35000 [11]
Size of test data: 2000
Without GA: 19.1%;
With GA: 10%

The tests were carried out with a sample population of 200 and maximum number of evolutions as 100. The data set was labeled to have around 20% error rate under normal classifiers.

2. Feasibility of the framework

Even though classification time is significantly lower in Thisara, training time of Thisara is quite high. Therefore for small data sets, the usability of the framework is low. Considering both training time and average classification time, it is possible to deduce a point where Thisara performs better than the classical implementation. Following tests were carried out with a training data set of 100000 training instances.

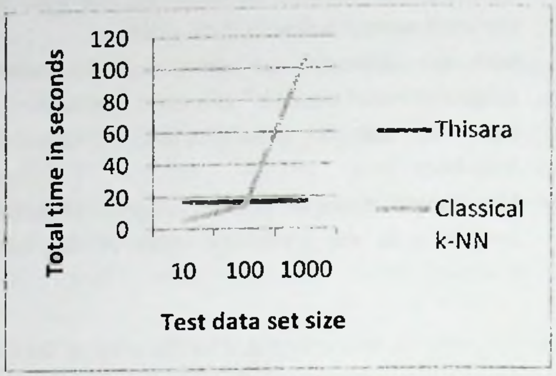

Fig. 7: Total operation time with changing testing dataset size in memory mode

This characteristic is also visible in same manner in the database mode. Therefore it is apparent that framework is viable to classify large data sets. These tests were carried out without adding the GA optimization timing. Therefore the crossover point will go higher when GA time is added to this. But even by that, Thisara has higher performance than the classical k-NN implementation for large data sets, because training phase is one time activity.

Speedup achieved using introduction of clustering is calculated as below.

$$\text{Speedup} = \frac{(T - t)}{T} \times 100\%$$

Where, T is the classification time before clustering and t is the classification time after clustering.

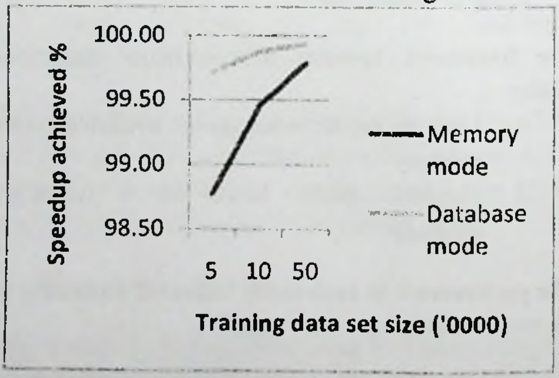

Fig. 8: Speedup achieved using clustering

VI. Conclusion

The main objective of the research is to introduce a classification framework which operates in a less time and increase the performance without compromising the accuracy. The system uses a novel approach of classifying data items by introducing clustering prior to the classification, using an efficient data structure which can handle sample data items effectively without consuming much time. It is observed that Thisara achieved significant performance gain over conventional classification with the novel approach.

REFERECNES

[1] Jiawei Han and Micheline Kamber, *Data Mining Concepts and Techniques*, Second Edition, Morgan Kaufmann Publishers, 2006

[2] Melanie Mitchell. (1999). An Introduction To Genetic Algorithms [Online]. Available: http://mitpress.mit.edu/catalog/item/default.asp?tid=5974&ttype=2 [Accessed: 26.08.2009]

[3] Michael Steinbach, Levent Ertöz and Vipin Kumar, "The Challenges of Clustering High Dimensional Data", [Online], Available: http://www-users.cs.umn.edu/~kumar/papers/high_dim_clustering_19.pdf [Accessed: 10.08.2009]

[4] Tian Zhang, Raghu Ramakrishnan, and Miron Livny, "Birch: An Effective Data Clustering Method for Very Large Databases", [Online], Available: http://www.lans.ece.utexas.edu/course/ee380l/2000sp/papers/p103-zhang.pdf [Accessed: 27.08.2009]

[5] "K-Means Clustering." [Online]. Available: http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html [Accessed: 27.08.2009]

[6] Kardi Teknomo, "K Nearest Neighbors Tutorial", [Online]. Available: http://people.revoledu.com/kardi/tutorial/KNN [Accessed: 14.06.2009]

[7] Anupam Kumar Nath, Syed M. Rahman, and Akram Salah. An Enhancement of k-Nearest Neighbor Classification Using Genetic Algorithm. [Online]. Available: http://komar.cs.stthomas.edu/MICS2005/papers/paper92.pdf [Accessed: 30.08.2009]

[8] Rui Xu and Donald C. Wunsch II, *Clustering*, IEEE Press. New Jersey. John Wiley & Sons Inc, 2008

[9] Hongxing He, et al. Application of Genetic Algorithm and k-Nearest Neighbor Method in Real World Medical Fraud Detection Problem. [Online]. Available: http://www.springerlink.com/index/5WXH6MYCCRHYH0E1.pdf [Accessed: 10.08.2009]

[10] "UCI Machine learning repository" [Online], Available: http://archive.ics.uci.edu/ml/databases/kddcup99/kddcup99.html [Accessed: 15.03.2010]

[11] "UCI Machine learning repository" [Online], Available: http://archive.ics.uci.edu/ml/datasets/Adult [Accessed: 15.03.2010]