Voronoi Diagrams - An exhaustive study with applications.

Samir Sheriff
Student 8th Semester
Department of Computer Science
RV College of Engineering
Email : samiriff@gmail.com

Vinay BS.
Student 8th Semester
Department of Computer Science
RV College of Engineering
Email : bsv0099@gmail.com

Satvik N
Student 8 semester
Department of Computer Science
RV College of Engineering
Email : nsatvik@gmail.com

***Abstract*—A Voronoi diagram is a decomposition of a given space, determined by distances to a specied family of objects (subsets) in space. These objects are usually called the sites or the generators and to each such object, one associates a corresponding Voronoi cell, namely the set of all points in the given space whose distance to the given object is not greater than their distance to the other objects. Voronoi diagrams can be found in a large number of elds in science and technology, even in art, and they have found numerous practical and theoretical applications. It is the technique that enables the division of such multi-dimensional spaces into subspaces.**

This paper deals with an exhaustive study of Voronoi Diagrams, different algorithms pertaining to it and its real-world applications, which will be explained with the aid of multimedia technology.

***Index Terms*—decision tree, machine learning, genetic algorithms, decision problems.**

I. INTRODUCTION

Voronoi diagram is a decomposition of a given space, determined by distances to a speci

ed family of objects (subsets) in the space. These objects are usually called the sites or the generators and to each such object one associates a corresponding Voronoi cell, namely the set of all points in the given space whose distance to the given object is not greater than their distance to the other objects. It is named after Georgy Voronoi, and is also called a Voronoi tessellation, a Voronoi decomposition, or a Dirichlet tessellation (after Lejeune Dirichlet). Voronoi diagrams can be found in a large number of

elds in science and technology, even in art, and they have found numerous practical and theoretical applications. It is the technique that enables the division of such multi-dimensional spaces into subspaces.

II. INTRODUCTION TO VORONOI DIGRAMS

A. Decision Trees

- **ID3** (Iterative Dichotomiser 3)
- **C4.5** algorithm, successor of ID3
- **CART** (Classification And Regression Tree)
- **CHi-squared Automatic Interaction Detector** (CHAID). Performs multi-level splits when computing classification trees.
- **MARS**: extends decision trees to better handle numerical data

B. Genetic Algorithms

A GA generally has four components.

The following is a typical GA procedure:

- Create an initial population of random genomes.
- Loop through the genetic algorithm, which produces a new generation every iteration.
  - Assess the fitness of each genome, stopping if a solution is found.
  - Evolve the next generation through natural selection and reproduction.
    - Select two random genomes based on fitness.
    - Cross the genomes or leave them unchanged.
    - Mutate genes if necessary.
  - Delete the old generation and set the new generation to the current population.
- When a solution is found or a generation limit is exceeded, the loop breaks and the genetic algorithm is complete.

C. GA-Based Feature Selection for Decision Trees

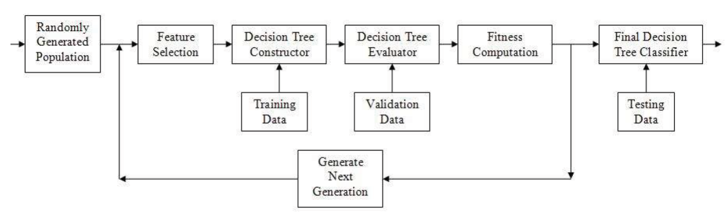

Fig. 1. The data flow in DT/GA Hybrid Classifier.

III. OBJECT-ORIENTED DESIGN AND RESULTS

The results of the GA-based optimization scheme is shown in the Figure 2.

The results obtained with manual feature selection is shown in Figure 3.

IV. CONCLUSION

The genetic algorithm and decision tree hybrid was able to outperform the decision tree algorithm which was based on manual feature selection. We believe that this improvement

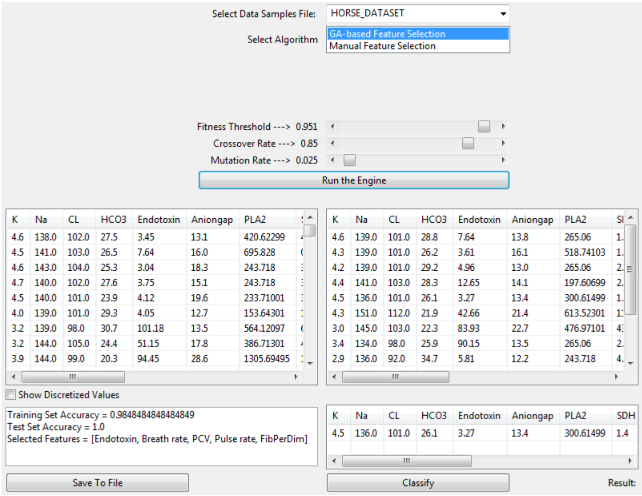

Fig. 2. DT built with GA based feature selection.

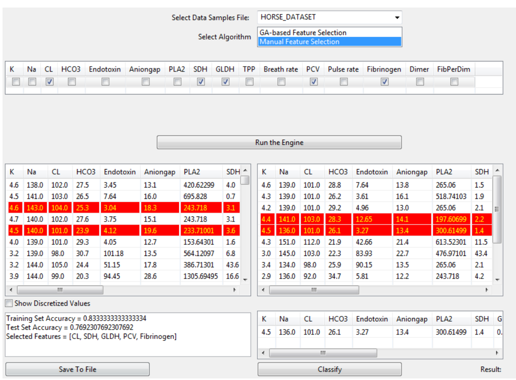

Fig. 3. DT built with manual feature selection.

is due to the fact that the hybrid approach is able to focus on relevant features and eliminate unnecessary or distracting features. This initial filtering is able to improve the classification abilities of the decision tree. The algorithm does take longer to execute than the standard decision tree; however, its non-deterministic process is able to make better decision trees. The training process needs only to be done once. The classification process takes the same amount of time for the hybrid and non-hybrid systems.

V. Future Work

The hybrid GA /decision tree algorithm needs to be tested further to realize its true potential. Clearly more work needs to be done. The test results show that the Decision Trees constructed using the Genetic algorithm-based feature selector, were more efficient and accurate in classifying the data than the Decision Trees constructed by selecting features manually. A forest of decision trees will be constructed from the combination of four final decision trees, each for one major attack category. The final decision will be made through a voting algorithm. We will then compare the overall classification ability of the hybrid algorithm with other machine learning algorithms in the literature.

Some other future enhancements could include one of the following:

1) The application could be made more responsive by using Threads and Parallel/Cloud Computing
2) The Decision Tree Classifier of this application could be optimized using Neural Networks which are more efficient than Decision Trees.
3) An interesting extension to be explored is the possibility of additional feedback from ID3 concerning the evaluation of a feature set. Currently only classification accuracy is returned. However, there is potentially exploitable information with respect to which features were actually used to build the decision tree and their relative positions in the tree.

References

[1] Hua;Kien A., Wu;Annie S.; Chen;Bing, Stein;Gary *Decision Tree Classifier For Network Intrusion Detection With GA-based Feature Selection*

[2] Jing Xu, Ming Zhao,Jos Fortes, Robert Carpenter, Mazin Yousif. *Autonomic resource management in virtualized data centers using fuzzy logic-based approaches* Cluster Comput (2008) 11: 213227 DOI 10.1007/s10586-008-0060-0

[3] T. Wood et al., *Sandpiper.Black-box and gray-box resource management for virtual machines*, Computer Network (2009), doi:10.1016/j.comnet.2009.04.014