Journal of Theoretical and Applied Information Technology
10th August 2013. Vol. 54 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GENETIC ALGORITHM AND PROGRAMMING BASED CLASSIFICATION: A SURVEY

[1]DHARMINDER KUMAR, [1,2*] SUNITA BENIWAL

[1]Department of Computer Science & Engineering, Guru Jambheshwar University of Science & Technology, Hisar-125001, Haryana, India
[2]Department of Information Technology, Maharishi Markandeshwar University, Mullana-Ambala- 133203, Haryana, India
*E-mail: [1]sunita_beniwal@rediffmail.com

ABSTRACT

Classification is the process of finding a model or a function that describes and distinguishes data classes and concepts, for the purpose of being able to use the model to predict the classes of objects whose class label is not known. The process of data analysis becomes time consuming and tedious as volume of data increases. So to make the process of data classification faster, soft computing techniques have been applied. Great deal of work has been done in the area of classification using evolutionary techniques. This survey gives an insight into the work done on classification using genetic algorithms and genetic programming and their applications in different problems and areas.

Keywords: *Classification, Genetic algorithm, Genetic programming, Fitness function*

1. INTRODUCTION

In recent years there has been an explosive growth in the generation and storage of electronic information as more and more operations are computerized due to technology advances in data acquisition and a continued decline in the cost of data storage [1]. The digital revolution has made digitized information easy to capture and fairly inexpensive to store [2]. The manual process of data analysis becomes tedious as size of data grows and the number of dimensions increases, so the process of data analysis needs to be computerised. The term KDD refers to the automated process of knowledge discovery from databases. KDD has many steps for analysis namely data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation and knowledge representation. As can be seen data mining is a step in the whole process of knowledge discovery which can be explained as a process of extracting or mining knowledge from large amounts of data [3]. It can also be explained as a collection of techniques for efficient automated discovery of previously unknown, valid, novel, useful and understandable patterns in large databases [4]. Data mining is the non trivial process that automatically collects the useful hidden information from the data and is taken on as forms of rule, concept, pattern and so on [5]. The extracted knowledge from data mining, allows the user to find interesting patterns and regularities deeply buried in the data to help in the process of decision making. According to different goals, the mining task can be mainly divided into four types: class/concept description, association analysis, classification or prediction and clustering analysis [6]. Before applying any kind of data mining algorithm to the available data, data needs to be prepared. A robust preprocessing system is required in order to extract any kind of knowledge [7]. Data preprocessing is done using steps like data integration, data cleaning, discretization, and attribute selection. Detailed descriptions of above preprocessing techniques have been widely reported [8-12].

The present article provides an overview of the available literature on classification based on genetic algorithms. Section II provides a brief overview of all classification techniques. Section III describes the basics of genetic algorithms. In Section IV the available literature on genetic algorithms based classification strategies.

2. CLASSIFICATION

Classification is the process of finding a model or a function that describes and distinguishes data classes and concepts, for the purpose of being able

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

to use the model to predict the classes of objects whose class label is not known. In the classification task the data being mined is divided into two mutually exclusive and exhaustive data sets: the training set and the test set. The data mining algorithm has to discover model by accessing the training set only. Data classification can be viewed as a two step process: learning step in which a classifier is built describing a predetermined set of classes or concepts by analyzing the training set made up of database tuples and their associated labels. In the second step model is used for classification by first estimating the predictive accuracy of classifier built during the first step. It is done using the test data. The accuracy of classifier on a given test set tuples is percentage of tuples that are correctly classified by the classifier. If the accuracy is above some acceptable level, the classifier can be used to predict future tuples whose class label is not known [3]. Classifiers used are of many categories like rule based classifiers, decision trees, support vector machines, bayesian classifiers, genetic algorithms, neural networks, fuzzy logic etc.

Genetic algorithms are used with other classification algorithms to improve their performance. Great deal of work has been done using hybrid evolutionary classifiers. In this paper a review of existing classifiers using genetic algorithms is presented with the respective fitness functions and reproduction operators.

3. GENETIC ALGORITHMS AND PROGRAMMING

Genetic Algorithms (GA) are search algorithms based on natural genetics that provide robust search capabilities in complex spaces, and thereby offer a valid approach to problems requiring efficient and effective search processes [13]. Initially GA was not designed as machine learning algorithm but can be easily dedicated to this task [14]. GA is an iterative process that operates on a population, i.e., a set of candidate solutions. Each solution is obtained by means of an encoding/decoding mechanism, which enables us to represent the solution as a chromosome and vice versa. Encoding of the chromosomes depends mainly on the problem to be solved. Initially, the population is randomly generated. Every individual in the population is assigned, by means of a fitness function, a fitness value that reflects its quality with respect to solving the particular problem. In each cycle, the fitness of each candidate solution is determined. The next stage is selection, where a temporary population is created in which the fittest individuals are likely to have a higher number of instances than less fit individuals. The reproductive operators like crossover and mutation are then applied to the individuals in this population yielding a new population. The whole process is repeated until a certain termination criterion is achieved, usually after completing a predetermined number of iterations or reaching steady state [15]. To speed up GA for harder and bigger problems, parallel and distributed GA's are used. Literature related to work done using parallel and distributed GA's can be found in [16,17,18,19,20,21].

Genetic programming (GP) has also emerged as a promising technique to discover useful and interesting knowledge from the database. Genetic programming has been formulated originally as an autonomous method for breeding computer programs using tree structures [22]. The principle elements of the GP are a set of functions and terminals that are able to represent the solution of the problem. For every generation, each individual will be evaluated for its fitness, and individuals of the next generation will be produced from the parents selected based on their fitness value. The population evolves through genetic reproduction, crossover and mutation over a number of generations until the termination criteria is met. At the end of the GP run, the best individual is presented as the solution to the problem.

4. GENETIC ALGORITHMS FOR CLASSIFICATION AND FEATURE SELECTION

4.1 Genetic Algorithms for Feature Selection

Genetic algorithms have been used with different classification methods to improve their performance. GA can also be used for selection of relevant features for classification [23]. Feature Selection techniques which use classifier error rates as the criterion are called wrapper type algorithms [24]. In a wrapper approach, the classifier for which features are being selected is itself used as the evaluation function. Since the suitability of features depends upon the concerned learning algorithm or classifier, the wrapper type algorithms usually perform better compared to other type called filter type techniques. In a filter-type method, the selection is done independent of the learning algorithms. In this case, the relevant features are filtered out from the irrelevant ones prior to the learning. Tseng and Yang [25] proposed a genetic algorithm for the clustering and classification problem. They also proposed another genetic

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithm for the feature selection problem which can not only search for a good set of features but also find the weight of each feature such that the application of these features associated with their weights to the classification problem will achieve a good classification rate. Raymer et al. [26] also presented a GA feature extractor for feature extraction in which feature selection, feature extraction, and classifier training are performed simultaneously using a genetic algorithm. The main purpose of feature selection is to reduce the number of features by eliminating less significant features while maintaining acceptable classification accuracy.

4.2 Genetic Algorithms for Rule Based Classification

Rule based classification classify data by using a collection of "if…then…" rules. A rule r covers an instance x if the attributes of the instance satisfy the condition of the rule. Classification rules can be build using two methods- direct method and indirect method. Direct methods are those methods that extract rules directly from data. Indirect methods are those that extract rules from other classification model like decision trees. [27,28]. GA has been used extensively for rule based classification. While using GA for rule optimisation there are two approaches to encode the rules: Michigan approach and Pitts Approach. The Michigan approach uses GA to evolve individual rules, a collection of which comprises the solution of the classification problem [29]. In the Pitt's approach, each individual represents a rule set. In this case a chromosome evolves a complete rule base and they compete among them along the evolutionary process [30].

GABIL system implemented by De Jong et al.[31] continually learns and refines concept classification rules using binary encoding of chromosomes representing a set of rules. Noda et al.[32] have reported a GA based approach to discover interesting rules using degree of interestingness of the rule and its predictive accuracy in this fitness function. Muntean and Valean [33] used genetic algorithms for rule induction by using Michigan approach for encoding and extracted a small set of comprehensible rules. Shi and Lei [34] presented a genetic algorithm based approach for mining classification rules from large database having higher classification performance to unknown data. The approach presented by Yang et al. [35] for learning classification rules outperformed traditional approaches on both the average prediction accuracy and the standard deviation. Pitt's approach is used for encoding and adaptive asymmetric mutation which biases the population toward generating rules with more coverage on training examples is used. HGAc proposed by Zhongyang and Lei [36] combines the capabilities of taboo search and genetic algorithms for rule discovery while using classification accuracy of each rule as the fitness function. Traditional genetic algorithm lacks the local search ability so a taboo criterion and flexible memory structure is used to avoid loop search. HGAc discovers rules much less in number and more precise. GACA (Genetic Ant Colony Algorithm) proposed by Zhang and Wu [37] for pattern classification outperforms the normal GA and ACA, while taking a bit longer training time and is capable of escaping from local optima. GACA consists of two stages. GA is used to search the optimal solutions in first stage, and when the efficiency decreases, the improved ACA is used to search.

For discovering comprehensible classification rules, methods using genetic programming as well as genetic algorithms have been proposed. Fidelis et al. [38] proposed a method which seems to be particularly effective in finding a concise set of comprehensible rules, since it discovers only a single rule for each class. Flexible chromosome encoding is used where a chromosome is divided into genes, each gene corresponding to a condition involving one attribute, subdivided into three fields: weight, operator and value. Stochastic tournament selection, two-point crossover and elitist reproduction strategy were used. Bojarczuk et al. [39] used Genetic Programming to find some short, very comprehensible rules having a good performance concerning predictive accuracy for diagnosing certain pathologies. In a given run of the GP all individuals represent rules predicting the same class. Genetic Programming Classifier (GPc) developed by Tan et al. [40] used Genetic Programming to evolve multiple comprehensible if-then classification rules which evolved the expected rules easily with high classification accuracy for clean data. The GP starts with an initial population which is created with ramped half- and-half method [41]. A covering- algorithm and token competition technique are used to penalize the redundant individuals and to promote diversity. The complete GP process is performed for every class of the dataset. The rule vector of every GP run is combined into a global decision list vector which is presented as the final solution of the problem at the end of the training process. The proposed GP

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

classifier also demonstrates its capability of discovering comprehensible rules from datasets without background knowledge on that dataset. Robu and Holban [42] suggested a genetic algorithm with a new fitness function for mining the classification rules after studying earlier used fitness functions and the results obtained by the algorithm are good and comparable with other algorithms. Table 1 summarizes various fitness functions, crossover operators, mutation techniques used for genetic algorithms designed for rule based classification.

Table 1: Rule Based Classification Using Genetic Algorithms

Fitness Function	Crossover	Mutation	Reference
Percentage of correctly classified examples	Two point crossover	Bit inversion	[31]
Degree of interestingness of the rule and its predictive accuracy	Uniform Crossover	Transforms the value of an attribute into another value	[32]
Classification accuracy of a rule	Two point crossover	-	[33]
Arithmetic weighted mean of comprehensibility , predictive accuracy and interestingness of rule	In case of common attributes :values exchanged In case of no common attributes: Swapping attributes	Deleting an attribute or replacing by another value if fitness is improved.	[34]
Classification accuracy of each rule	-	-	[36]
Error rate, entropy, rule consistency, hole ratio	-	Adaptive asymmetric mutation	[35]
Accuracy and coverage	Single point crossover	Bit Inversion	[37]
Sensitivity and specificity	Two-point crossover	Weight mutation, relational-operator mutation, value mutation	[38]
Sensitivity and specificity	Swapping selected subtrees	-	[39]
True positive, false positive, true negative and false negative.	-	-	[40]
Predictive accuracy, comprehensibilit, and sensitivity.	One point crossover	Bit inversion	[42]

4.3 Genetic Algorithms for Decision Trees

Decision tree is a classifier in the form of a tree structure, where each node is either: a leaf node which indicates the value of the target attribute of examples, or a decision node which specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test. A decision tree can be used to classify an example by starting at the root of the tree and moving through it until a leaf node, which provides the classification of the instance. Decision tree induction is a typical inductive approach to learn knowledge on classification. (http://dms.irb.hr/tutorial/tut_dtrees.php)[43].
Genetic algorithms have been employed for construction of decision trees having improved classification performance. Papagelis and Kalles[44] proposed an algorithm called GATree which used genetic algorithms to evolve decision trees for classification. Decision trees are build so that one decision node leads to two leaves. Mutation chooses a random node and replaces that node's test-value with a new random value. The crossover operator chooses two random nodes and swaps those nodes' sub-trees. For the objective function that assigns utility to candidate solutions, accuracy and size of decision trees are balanced. GATree produced a dynamic, small-biased, accuracy size based tree optimisation. Fu [45] proposed an evolutionary computation approach GAIT, combining statistical sampling, genetic algorithm, and decision tree, to develop intelligent decision trees. The initial population of the genetic algorithm is generated by C4.5 which is used to generate a set of diverse decision trees. GAIT produces about the same level of accuracy as a standard decision tree algorithm at significantly lower sampling percentages which indicates that GAIT is likely to scale well and is effective for large-scale data mining. Bala et al. [46] introduced a hybrid learning methodology that integrates genetic algorithms and decision tree learning in order to evolve optimal subsets of discriminatory features for robust pattern classification. The representation used for chromosomes is a fixed-length binary string. The GA-ID3 process iterates until a feature subset is found with satisfactory classification performance which is then recommended to be used in the actual design of the pattern classification system. The results revealed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

significant improvements in classification performance and reduced description complexity when compared against standard methods for feature selection. Information regarding fitness functions, crossover and mutation techniques are presented in table 2.

Table 2: Genetic Algorithms For Decision Trees, SVM And KNN

Fitness Function	Crossover	Mutation	Reference
Accuracy and size	Swapping selected subtrees.	Choosing a random node and replacing the node's test-value with a new random value.	[44]
Percentage of correctly classified observations	Exchanging subtrees or leaf nodes of different trees	Exchanging subtrees or leaf nodes of same tree.	[45]
Accuracy and the size of the feature subset			[46]
Classification accuracy, the number of selected features, and the feature cost	Simple crossover	Bit inversion	[48]
Classification accuracy	-	-	[49]
Based on rank ordering	Uniform crossover	Real number mutation	[53]
Accuracy	Uniform crossover	Population-adaptive mutation	[54]
Number of incorrect predictions, unmasked features, incorrect votes and difference in accuracy between classes.	-	-	[26]

4.4 Genetic Algorithms for SVM

The support vector machine (SVM) is a training algorithm for learning classification and regression rules from data. SVMs arose from statistical learning theory; the aim being to solve only the problem of interest without solving a more difficult problem as an intermediate step. SVMs are based on the structural risk minimisation principle, closely related to regularisation theory[47]. Genetic algorithms have been used with SVM for feature reduction and classification. Rough sets and genetic algorithms based method called RGSC(Rough Sets and Genetic Algorithms) for SVM classifier was presented by Wang et al. [48], to reduce the dimension of feature vectors, optimizing the parameters to improve the SVM classification accuracy and speed. A genetic algorithm is used for feature selection and parameter optimization to improve classification accuracy. A rough set feature reduction algorithm is used for finding a reduction of a decision table which reduces the features by constructing the binary discernibility matrix. It is indicated in the approach that the RGSC yields better accuracy even with a large data set. In the approach presented by Tahayna et al. [49], for video events classification, GA is employed to optimize the feature and instance subset and SVM kernel parameters simultaneously. The system generates the initial population which is used to find global optimum factors: feature and instance selection variables, and kernel parameters. The chromosome is encoded as a binary string. After generating the initial population, the system performs a typical SVM process using the assigned value of the factors in the chromosomes, and assesses the performance of each chromosome, which is determined through the fitness function. Classification results on sport videos show significant improvement over conventional SVM. Frohlich and Chapelle [51] presented a special genetic algorithm, which takes into account the existing bounds on the generalization error for support vector machines. The genetic encoding allows optimizing different parameters of the SVM in parallel. Chromosomes are represented by a standard binary encoding where a 1 indicates the selection of the feature at the corresponding position. The GA used here is the CHC algorithm proposed by Eshelman [52], which was reported to perform a more aggressive and faster search strategy than the traditional Simple GA. The results showed that the selection of a feature subset and kernel parameters of the SVM can be optimized by means of GAs. Summarized information related to fitness function, crossover operators and mutation operators used with SVMs is given in table 2.

4.5 GA for KNN

Nearest neighbor classifiers are based on learning by analogy. The training samples are described by n dimensional numeric attributes.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

k–NN classification has two stages; the first is the determination of the nearest neighbours and the second is the determination of the class using those neighbors [52]. Kelly and Davis [53] proposed a method called GA-WKNN to learn an attribute weight vector which improves KNN classification. Real valued encoding is used for encoding by associating a vector value with each classification attribute and with each of the k neighbours. The initial population of chromosomes in each run of the GA-WKNN algorithm is randomly generated. Fitness was assigned by first rank ordering the population using one of the two ranking functions either number of misclassifications or multiple value ranking and then assigning fitness to each member from the series. The effect of this technique is to produce mild pressure in favour of the best population members when the run begins. The proposed method requires computational capabilities above that of the KNN algorithm, but achieves improved classification performance in a reasonable time. Peterson et al. [54] explored the effectiveness of GA weight and offset optimization for knowledge discovery using KNN classifiers employing Pearson correlation as a similarity measure which follows the population-adaptive mutation to accomplish feature selection instead of using bit mask. Tournament selection is used for selection respectively. A fairly high probability for mutation is employed as mutation drives dimensionality reduction. However, the flexibility provided by offset optimization for KNN may provide an improvement in accuracy, justifying increased computational effort for some applications. Approach presented by Raymer et al. [26] performs feature selection, feature extraction, and classifier training simultaneously using a genetic algorithm. The integrated feature extraction and classification approach described was tested on different data sets and its performance was compared with SFSS(sequential floating forward selection) presented by Pudil et al. [55] and it was found GA–KNN was more effective of the approaches, and required less features to make the classification. For optimizing the performance of KNN using genetic algorithms different fitness functions and crossover and mutation operators have been used about which summarized information can be found in table 2.

4.6 Genetic Algorithms with |Neural Networks

An artificial neural network, often just called a "neural network" is a mathematical model or computational model based on biological neural networks, in other words, is an emulation of biological neural system. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase [56]. Genetic algorithms have been used to optimize neural networks for classification. Foster et al. [57] used genetic algorithm with neural networks to investigate the tradeoff between the number of genetic algorithm generations, the number of neural network passes, and the size of the population used by the genetic algorithm. They reported that smaller population sizes are effective because they allow a greater number of generations permitting the evolution of better neural network parameters and that the relationship between the number of genetic algorithm trials and number of neural network passes was found to be problem dependent. Chen and Yao [58] proposed a method called SGA for optimizing classifiers of neural network using genetic algorithm based on principle of gene reconfiguration, and implement classification by training the weight. Results show that the improved genetic algorithm optimizes neural network to improve classification correctness. Table 3 provides summarized information related to fitness functions, crossover operators of genetic algorithms used in combination with neural networks.

4.7 GA Based Fuzzy Classifiers

Fuzzy set theory provides excellent means to model the fuzzy boundaries of linguistic terms by introducing gradual memberships. In contrast to classical set theory, in which an object or a case either is a member of a given set (defined, e.g., by some property) or not, fuzzy set theory makes it possible that an object or a case belongs to a set only to a certain degree, thus modelling the penumbra of the linguistic term describing the property that defines the set [59]. Ishbuchi et al. [60] and Zong-yi et al. [61] used GA for construction of compact fuzzy classification system. Ishbuchi et al. [60] proposed a genetic-algorithm based approach for the construction of fuzzy classification systems with rectangular fuzzy rules. Compact fuzzy classification systems are automatically constructed from numerical data by selecting a small number of significant fuzzy rules using genetic algorithms. A rule set is represented as concatenation of bits. The possible values are 1,-1 or 0 which denotes whether rth rule is included, not included or dummy rule respectively. Mutation is performed by assigning a larger probability to inversion from 1 to -1 in order to reduce the number of fuzzy rules in each individual. The proposed approach can be viewed as a knowledge

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

acquisition tool for classification problems as significant fuzzy rules are selected and unnecessary fuzzy rules are removed. Zong-yi et al. [61] used multi objective genetic algorithm to obtain an initial fuzzy system. The multi-objective genetic algorithm is used to accomplish feature selection and fuzzy partition simultaneously. A genetic algorithm is used to select significant fuzzy rules and to exclude the redundant fuzzy rules to achieve a compact fuzzy system which is done in the same manner as done by Ishbuchi et al. [60]. A genetic algorithm is also used to optimize the parameters of the fuzzy system. The proposed fuzzy classification system has higher classification performance and lesser number of features and less number of fuzzy rules. For selecting best and minimal fuzzy rule set using genetic algorithms, different types of operators are used in genetic algorithms summarized information related to them can be found in table 3.

Table 3: Neural Network And Fuzzy Classifiers Using Genetic Algorithms

Fitness Function	Crossover	Reference
Performance of neural networks in classification and size of the network	-	[57]
Reciprocal of error function	Shift reverse logic crossover operation	[58]
Number of fuzzy rules and the number of mistakenly classified training patterns	Two point crossover	[60]
Weighted sum of classification performance, interpretability and the number of fuzzy rules	-	[61]

4.8 Parallel, Distributed and Incremental Genetic Algorithm for Classification

Parallel and distributed genetic algorithms find their application in classification for large data sets. PC-HGA i.e. parallel classification algorithm based on hybrid genetic algorithm was presented by Zhongyang et al. [62] for improving the performance of HGAc [36] for large data sets using Master-slave parallel computing mode. The experimental results show that PC-HGA has high performance and speedup. Efficient Distributed Genetic Algorithm for Rule Extraction (EDGAR) presented by Rodriguez et al [63] based on dynamic data partitioning shows advantages in scalability for exploring high complexity search spaces with comparable classification quality. EDGAR shows a considerable speed up without compromising the accuracy and quality of the classifier. Araujo et al. [64] presented a method called GA-PVMINER, a parallel genetic algorithm that uses Parallel Virtual Machine (PVM) to discover rules in a database by dividing the global population into several subpopulations and partitioning each subpopulation and data being mined across the available processors so that different subpopulations evolve in parallel. The discovered rules have a good generalization performance on the unseen test set.

The problems in which new training data, attributes and classes may become available or some existing elements may get changed i.e. incremental learning also used Genetic algorithms. Guan and Zhu [65] developed a GA-based incremental learning scheme for classification purposes using one or more classifier agents in a multiagent environment such that that there is no need to re-evolve the rule set from scratch in order to adapt to the ever-changing environment. Classifier agents can exchange information of new attributes and classes. If available, the agents can also exchange evolved rule sets and provide each other with new training/test data, or challenge other agents with unsolved problems. They suggested that an incremental type approach was better suited for satisfying the time and resource constraints imposed in typical real-world applications. Li et al. [66] also proposed an incremental genetic algorithm (IGA) for real-world datasets subject to concept drift. Michigan approach is used for encoding. In the proposed approach, an assessment is made of the percentage change in the environment, and a corresponding number of new chromosomes are then randomly generated and used to replace an equivalent number of chromosomes in the stored population pool. A Rank based Roulette Wheel Selection (RRWS) scheme [67] is used for the selection process. The IGA presented is more computationally efficient than the non-incremental GA, but achieves a virtually identical classification performance. Two problems in the IGA [66] methods were identified by Vivekanandan and Nedunchezhian[68]. First is increase in learning cost because of application genetic algorithm incrementally without monitoring concept drift. Next, if the data distribution changes, IGA may also forget some of the rules, if the data of that rules does not reappear. A new incremental genetic algorithm is proposed by Vivekanandan and Nedunchezhian [68] to rectify the above two

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

problems using a core method similar to the method proposed by Li et al. [66]. In the proposed method each record of the incoming dataset is monitored. Correctly classified records are dropped and misclassified records are added to a window. When the window is full, the genetic algorithm is applied to the records in the window and new rules are generated based only on the misclassified examples and on the examples of new classes. The new method ensures that the problems present in IGA [66] are removed greatly reducing the computational overhead particularly when there is no concept drift or when there is a slow drift.

Table 4: Parallel, Distributed And Incremental Genetic Algorithm For Classification

Fitness Function	Crossover	Mutation	Reference
Degree of interestingness of a rule	In case of common attributes swapping the value element If no common attribute: randomly choosing a term of the first parent and inserting it into the second.	Attribute mutation and value mutation.	[64]
Weighted sum of simplicity, support, confidence and accuracy	According to attribute	According to attribute	[62]
Simplicity, Quality	Two-point crossover	Bit inversion	[63]
Percentage of correctly classified by the chromosome's rule set	Biased crossover.	Biased Mutation	[65]
Sensitivity and specificity	Two-point crossover	Bit inversion is used.	[66]
Sensitivity and specificity	Two point crossover	Bit inversion is used.	[68]
Predictive accuracy and comprehensibility	Exchanging the genes	Changing the value of an attribute to another random value	[69]

Vivekanandan and Nedunchezhian (2010a) also proposed an incremental genetic algorithm that builds the rule based classification model in a fine granular manner by independently evolving tiny components based on the evolution of the data set which reduces the learning cost and makes it scalable to large data sets. The proposed method's execution time is faster as compared to the simple genetic algorithm and the parallel genetic algorithm and accuracy of the rules mined is better than or similar to that of the rules mined by the simple genetic algorithm.

Table 4 provides information related to different fitness functions and crossover operators used with parallel and distributed genetic algorithms used for classification.

5. CONCLUSION

The optimization capability of Genetic Algorithms was applied to correctly classify the given data sets both with known and unknown data. The use of genetic programming for discovering comprehensible rules was also discussed. Both small and large data set were considered. One parameter has been mainly used in designing of fitness function that is classification accuracy. GA was applied on classification algorithms like rule based classification, support vector machines, K-nearest neighbour and decision trees. Different types of genetic algorithm like incremental, multi objective and parallel genetic algorithm were applied. Genetic algorithms were also combined with other techniques to improve classification accuracy like ant colony and tabu search.

REFERENCES

[1] G.K. Gupta, Introduction to data mining with case studies, Prentice hall of India Pvt. Ltd., New Delhi, 2006.

[2] U. Fayyad, and R. Uthuruswamy, "Data mining and knowledge discovery in databases," Commun. ACM, vol. 39, pp. 24-27, November 1996.

[3] J. Han and M. Kamber, Data mining concepts and techniques, Morgan Kaufmann, San Francisco, 2006.

[4] I.H. Witten, E. Frank and M.A. Hall, Data mining practical machine learning tools and techniques, Morgan Kaufmann publisher, Burlington, 2011.

[5] T.J. Shan, H. Wei and Q. Yan, "Application of genetic algorithm in data mining" in First international workshop on education technology and computer science. IEEE, vol. 2, pp. 353- 356, 7-8 March 2009.

[6] Z.Z. Shi, Knowledge discovery, Tsinghua University Press, Beijing, 2001.

[7] S. Mitra, S.K. Pal and P. Mitra, "Data mining in soft computing framework: a survey,"

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IEEE T Neural Networ., vol. 13, pp. 3-14, January 2002.

[8] J. Catlett, On changing continuous attributes into ordered discrete attributes, Machine Learning—EWSL-91, Springer Berlin Heidelberg, 1991.

[9] I. Guyon, N. Matic and V. Vapnik, Discovering informative patterns and data cleaning, in Advances in knowledge discovery and data mining, AAAI/MIT Press, California, 1996.

[10] B. Pfahringer, "Compression-based discretization of continuous attributes," in Proceeding of 12th International Conference on Machine Learning, San Francisco , July 1995, pp. 456-463.

[11] D. Pyle, Data preparation for data mining, Morgan Kaufmann publisher, San Francisco,1999.

[12] E. Simoudis, B. Livezey and R. Kerber, Integrating inductive and deductive reasoning for data mining, in Advances in knowledge discovery and data mining, AAAI/MIT Press, California,1996.

[13] J.H. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT Press, Michigan, 1992.

[14] D.E. Goldberg, Genetic algorithms in search, optimization and machine learning, Addison-Wesley Longman publishing Co., Boston, 1989.

[15] E. Alba, and C. Carlos, The On-Line Tutorial on Evolutionary Computation, (2003) http://www.lcc.uma.es/~ccottap/semEC. Accessed 21 August 2011.

[16] E. Alba, and M. Tomassini, "Parallelism and evolutionary algorithms," IEEE T. Evolut. Comput., vol. 6, pp. 443-461, October 2002.

[17] E. Alba, and J.M. Troya, "A survey of parallel distributed genetic algorithms," Complexity, vl. 4, pp. 31-52, May 1999.

[18] S, Baluja, "Structure and Performance of fine grain parallelism in genetic search," in Proceedings of the 5th International conference on genetic algorithms, San Francisco, Kaufmann Publishers Inc., pp. 155-162, 1993.

[19] T.C. Belding, "The distributed genetic algorithm revisted," in Proceedings of the 6th International conference on genetic algorithms, San Francisco, Morgan Kaufmann Publishers Inc., pp. 114-121, 1995.

[20] E. Cantu-Paz, "A survey of parallel genetic algorithms," Calculateurs, Paralleles, Reseaux et. Systemes Repartis, vol. 10, pp. 141-171, 1998.

[21] R. Tanese, Distributed genetic algorithms, Morgan Kaufmann publishers Inc., San Francisco, 1989.

[22] R. Koza, Genetic Programming: on the Programming of Computers by Means of Natural Selection, MA: MIT Press,Cambridge, 1992.

[23] A.L. Blum, and P. Langley, "Selection of relevant features and examples in machine learning," Artificial Intelligence, vol. 97, pp. 245–271, December 1997.

[24] R. Kohavi, and G.H. John, "Wrappers for feature subset selection," Artificial Intelligence, vol. 97, pp. 273–324, December 1997.

[25] L.Y. Tseng, and S.B. Yang, "Genetic algorithms for clustering, feature selection and classification," Neural Networks, vol. 3, pp. 1612–1616, June 1977.

[26] M.L. Raymer, W.F. Punch, E.D. Goodman, L.A. Kuhn, and A.K. Jain, "Dimensionality reduction using genetic algorithms," IEEE T. Evolut. Comput., vol. 4,pp. 164-171, July 2000.

[27] J.R. Quinlan, C4.5: Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann, San Francisco, 1993.

[28] W.W. Cohen, Fast Effective Rule Induction, in Machine Learning-International Workshop Then Conference, Morgan Kaufmann Publishers, Inc., 1995.

[29] A.A. Freitas, A survey of evolutionary algorithms for data mining and knowledge discovery, in Advances in evolutionary computation, Springer-Verlag, New York, 2002.

[30] S. Smith, A learning system based on genetic algorithms. Dissertation, University of Pittsburgh, 1980.

[31] K. DeJong, W.M. Spears and D.F. Gordon, Using genetic algorithms for concept learning, Springer US, 1994.

[32] E. Noda, A.A. Freitas and H.S. Lopes, "Discovering interesting prediction rules with a genetic algorithm," in Proceedings of 1999 congress on evolutionary computation (CEC' 99), Washington, July 1999, pp. 1322-1329.

[33] M. Muntean and H. Valean, "Learning classification rules with genetic algorithm," Communications (COMM), 2010 8th

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

International Conference on, Bucharest, Romania, June 2010, pp. 213-216

[34] X. Shi and H. Lei, "A genetic algorithm-based approach for classification rule discovery," in Proceedings of International conference on information management, innovation management and industrial engineering (IEEE), Taipei, Dec 2008, pp. 175-178.

[35] L. Yang, D.H. Widyantoro, T. Ioerger and J. Yen, "An entropy-based adaptive genetic algorithm for learning classification rules," in Proceeding of the 2001 congress on evolutionary computation, Seoul, May 2001, pp. 790-796.

[36] X. Zhongyang, Z. Lei and Z. Yufang, A classification rule mining method using hybrid genetic algorithms, IEEE TENCON, Thailand, 2004.

[37] Y.D. Zhang and L.N. Wu, "A genetic ant colony classifier," in Proceeding of World Congress on Computer Science and Information Engineering, Los Angeles, April 2009, pp. 744 - 748 .

[38] M.V. Fidelis, H.S. Lopes and A.A. Freitas, "Discovering comprehensible classification rules with a genetic algorithm," in Proceedings of congress on evolutionary computation, La Jolla, July 2000, pp. 805 – 810.

[39] C.C. Bojarczuk, H.S. Lopes, and A.A. Freitas, "Discovering comprehensible classification rules using Genetic Programming: a case study in a medical domain," in Proceedings of genetic and evolutionary computation conference (GECCO- 99), pp. 953-958, 1999.

[40] K.C. Tan, A. Tay, T.H. Lee, C.M. Heng, "Mining multiple comprehensible classification rules using genetic programming," in Proceedings of the 2002 congress on evolutionary computation, Honolulu, May 2002, pp.1302–1307.

[41] M.L. Wong and K.S. Leung, Data mining using grammar based genetic programming and applications, Kluwer Academic Publishers, London, 2000.

[42] R. Robu and S. Holban, "A genetic algorithm for classification," in Proceedings of recent researches in computers and computing, pp. 52-56, 2011.

[43] http://dms.irb.hr/tutorial/tut_dtrees.php

[44] A. Papagelis and D. Kalles, "GATree: Genetically evolved decision trees," in proceedings of 12th International conference on tools with artificial intelligence, Vancouver, November 2000, pp. 203-206.

[45] Z. Fu, "A Computational Study of Using Genetic Algorithms to Develop Intelligent Decision Trees," in Proceedings of the 2001 IEEE congress on evolutionary Computation, Seoul, May 2001, 1382 - 1387.

[46] J. Bala, J. Huang, H. Vafaie, K. DeJong and H. Wechsler, "Hybrid learning using genetic algorithms and decision trees for pattern classification," in International joint conference on artificial intelligence, pp. 719-724, 1995.

[47] R. Burbidge and B.F. Buxton, "An introduction to support vector machines for data mining," in: Sheppee, M, (ed) Keynote Papers, Young OR12, University of Nottingham, Operational Research Society, pp. 3–15, 2001.

[48] X. Wang, Z. Hua and R. Bai, "A hybrid text classification model based on rough sets and genetic algorithms," in Ninth ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing, Phuket, August 2008, pp. 971-977.

[49] B. Tahayna, M. Belkhatir, S.M. Alhashmi and T. O'daniel, "Optimizing support vector machine based classification and retrieval of semantic video events with genetic algorithms," In: Proceedings of 17th international conference on image processing, Hong Kong, 2010, pp.1485-1488.

[50] H. Frohlich and O. Chapelle, "Feature selection for support vector machines by means of genetic algorithms," in Proceedings of 15th IEEE international conference on tools with artificial intelligence, November 2003, pp.142–148.

[51] L.J. Eshelman, "The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination," in Proceedings of foundations of genetic algorithms (FOGA), pp. 265-283,1990.,

[52] P. Cunningham and S.J. Delany, K-Nearest Neighbour Classifiers, Technical Report UCD-CSI-2007, 2007.

[53] J.D. Kelly and L. Davis, "A hybrid genetic algorithm for classification," in Proceedings of 12th international joint conference on artificial intelligence, Sydney, Morgan Kaufmann, pp. 645–650, 1991..

[54] M.R. Peterson, T.E. Doom and M.L. Raymer, "GA-facilitated KNN classifier optimization with varying similarity measures," in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Proceedings of 2005 IEEE Congress on Evolutionary Computation, New York, September 2005, pp. 2514 – 2521.

[55] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," Pattern Recognition Letters, vol 15, pp. 1119–1125, November 1994.

[56] Y. Singh, and A.S. Chauhan, "Neural Networks in Data Mining," J. Inform. Tech. Theor. Appl., pp. 37-42. 2005.

[57] D. Foster, J. McCullagh and T. Whitfort, "Evolution versus training: An investigation into combining genetic algorithms and neural networks," in Proceedings of international conference on neural information processing and intelligent information systems (ICONIP), Perth, November 1999, pp. 848-854.

[58] M. Chen and Z. Yao, "Classification techniques of neural networks using improved genetic algorithms," in proceedings of second international conference on genetic and evolutionary computing, Washington, September 2008, pp. 115-119.

[59] R. Kruse, J. Gebhardt and F. Klawonn, Foundations of Fuzzy Systems, J. Wiley & Sons, Chichester, United Kingdom, 1994.

[60] H. Ishbuchi, K. Nozalu, N. Yamamoto and H. Tanaka, "Acquisition of fuzzy classification knowledge using genetic algorithms," in Proceedings of 3rd IEEE conference on fuzzy systems, Florida, June 1994, pp. 1963-1968.

[61] X. Zong-yi, H. Yuan-long, T. Zhong-zhi and J. Li-min, " Construction of fuzzy classification system based on multi-objective genetic algorithm," in Proceedings of the 6th international conference on intelligent systems design and applications, Jinan , October 2006, pp. 1029 - 1034 .

[62] X. Zhongyang, Y. Zhang, L. Zhang and S. Niu, "A parallel classification algorithm based on hybrid genetic algorithm," in Proceedings of the 6th world congress on intelligent control and automation, Dalian, China, 2006, pp. 3237 – 3240.

[63] M.A. Rodriguez, D.M. Escalante, and A. Peregrin, "Efficient distributed genetic algorithm for rule extraction," Applied Soft Computing, vol. 11, pp. 733-743, January 2011.

[64] D.L.A. Araujo, H.S. Lopes and A.A. Freitas, "A parallel genetic algorithm for rule discovery in large databases," in International conference Proceedings of systems, man and cybernetics conference IEEE, Tokyo, October 1999, pp. 940-945.

[65] S.U. Guan and F. Zhu, "An incremental approach to genetic-algorithms-based classification," IEEE Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 35, pp. 227 – 239, April 2005

[66] I.H. Li, I.E. Liao and W.Z. Pang, "Mining classification rules in the presence of concept drift with an incremental genetic algorithm," J. Theor. Appl. Info. Tech., vol. 4, pp. 608-623, July 2008.

[67] A.J. Omar, R. Lakishmi, and C.R. Rao, "Improved selection operator for GA," J Theor. Appl. Inform. Technol.,, vol. 4, pp. 269-277, April 2008.

[68] P. Vivekanandan, and R. Nedunchezhian, "A new incremental genetic algorithm based classification model to mine data with concept drift," J. Inform. Tech. Theor. Appl., vol 21, pp. 36-42, November 2010.

[69] P. Vivekanandan, and R. Nedunchezhian, "A fast genetic algorithm for mining classification rules in large datasets," Int. j. soft computing, vol. 1, pp. 10-20, 2010.