IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 6, Ver. V (Nov – Dec. 2014), PP 01-06
www.iosrjournals.org

A Survey on: Stratified mapping of Microarray Gene Expression datasets to decision tree algorithm aided through Evolutionary Design

Ms. Neha V.Bhatambarekar[1], Prof. Payal S. Kulkarni[2]
[1]*Student, Computer Engg. Department, GES's R. H. Sapat COE Nashik Pune University, India*
[2]*Professor, Computer Engg. Department, GES's R. H. Sapat COE Nashik Pune University, India*

***Abstract*:** *Analyzing gene expression data is a challenging task since the large number of features against the shortage of available examples can be prone to over fitting. In order to avoid this pitfall and achieve high performance, some approaches construct complex classifiers, using new or well-established strategies. In medical decision making (classification, diagnosing) there are many situations where decision must be made effectively and reliably. Decision tree induction is one of the most employed methods to extract knowledge from data, since the representation of knowledge is very intuitive and easily understandable by humans. Conceptual simple decision making models with the possibility of automatic learning are the most appropriate for performing such tasks. Decision trees are a reliable and effective decision making technique that provide high classification accuracy with a simple representation of gathered knowledge and they have been used in different areas of medical decision making. Following recent breakthroughs in the automatic design of machine learning algorithms, a novel approach of hyper-heuristic evolutionary algorithm called HEAD-DT that evolves design components of top-down decision tree induction algorithms can be used to improve the classification accuracy.*

***Keywords*:** *Automatic algorithm design, Decision trees, Evolutionary Algorithms, Genetic programming, Hyper-heuristics, Machine Learning, Random forest*

I. Introdution:

The DNA microarray technology allows monitoring the expression of thousands of genes simultaneously [1] .Thus, it can lead to better understanding of many biological processes, improved diagnosis, and treatment of several diseases. However data collected by DNA microarray's are not suitable for direct human analysis, since a single experiment contains thousands of measured expression values. Several approaches have been suggested towards exploiting data mining from microarray data [2] including supervised and unsupervised machine learning algorithms where supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples [3]. Decision tree is a classifier depicted in a flowchart like tree structure which has been widely used to represent classification models, due to its comprehensible nature that resembles the human reasoning. Decision tree induction algorithms present several advantages over other learning algorithms, such as robustness to noise, low computational cost for generating the model, and ability to deal with redundant attributes. Besides, the induced model usually presents a good generalization ability [3] [4]. Most decision tree induction algorithms are based on a greedy top-down recursive partitioning strategy for tree growth. They use different variants of impurity measures, like, information gain [5], gain ratio [6], gini-index [7] and distance-based measures [6], to select an input attribute to be associated with an internal node. One major drawback of greedy search is that it usually leads to sub-optimal solutions. Several alternatives have been proposed to overcome these problems, including the induction of an ensemble of trees. Ensembles are created by inducing different trees from training samples and the ultimate classification is frequently given through a voting scheme [8], [9]. However, a disadvantage of ensembles is that the comprehensibility of analyzing a single decision tree is lost. Indeed, classification models being combined in an ensemble are often, to some extent, inconsistent with each other; an inconsistency that is necessary to increase the predictive accuracy of the ensemble [10].Therefore, ensembles are not a good option for applications where comprehensibility is crucial. Hence, an approach that has been increasingly used is the induction of decision trees through Evolutionary Algorithms (EAs). Instead of local search, EAs perform a robust global search in the space of candidate solutions. As a result, EAs tend to cope better with attribute interactions than greedy methods [11]. They are essentially algorithms inspired by the principle of natural selection and genetics. In nature, individuals are continuously evolving, adapting to their living environment. In EAs, each "individual" represents a candidate solution to the target problem. Each individual is evaluated by a

fitness function, which measures the quality of its corresponding candidate solution. At each generation, the best individuals have a higher probability of being selected for reproduction. The selected individuals undergo operations inspired by genetics, such as crossover and mutation, producing new offspring which will replace the parents, creating a new generation of individuals. This process is iteratively repeated until a stopping criterion is satisfied [12], [13].

I. Following represents a common algorithmic framework for both Genetic Algorithms (GAs) and Genetic Programming (GP), well-known EAs.
II.
III. 1: Create initial population of individuals
IV. 2: Compute the fitness of each individual
V. 3: repeat
VI. 4: Select individuals based on fitness
VII. 5: Apply genetic operators to selected individuals, creating new individuals
VIII. 6: Compute fitness of each new individual
IX. 7: Update the current population (new individuals replace previous individuals)
X. 8: until (stopping criteria)
XI.
XII. The number of EAs for decision tree induction has grown in the past few years, mainly because they report good predictive accuracies whilst keeping the comprehensibility of decision trees. In this context, we provide a detailed survey of EAs to evolve decision trees for classification.

II. Literature Survey:

Finding a (near)-optimal decision tree for a given data set is a challenging problem because the number of possible trees for a given data set grows exponentially with the number of attributes. It has been shown that constructing a decision tree with a minimal number of tests for classifying a new instance is a NP-complete problem [14]. It has also been proved that finding a minimal decision tree consistent with the training set is NP-hard [15], which is also the case for the task of finding the minimal decision tree equivalent to a given decision tree [16] and building the optimal decision tree from decision tables [17]. The aforementioned studies indicate that finding the optimal decision tree for a given data set using a brute force approach that evaluates all possible decision trees is feasible only for very simple problems. Therefore, heuristic procedures are necessary for finding a near-optimal decision tree in difficult problems. Many different types of heuristic procedures have been proposed in the literature for building decision trees.

R. C. Barros, R. Cerri, P. A. Jaskowiak, and A. C. P. L. F. de Carvalho [18] proposed hill-climbing bottom-up induction an iterative searching methodology. Hill climbing is simply a loop that continually moves in the direction of increasing value. The algorithm does not maintain a search tree, so the node data structure need only record the state and its evaluation. One important refinement is that when there is more than one best successor to choose from, the algorithm can select among them at random.

But this simple policy has three well-known drawbacks: [19]

- Local maxima: a local maximum, as opposed to a global maximum, is a peak that is lower than the highest peak in the state space. Once on a local maximum, the algorithm will halt even though the solution may be far from satisfactory.
- Plateaux: a plateau is an area of the state space where the evaluation function is essentially flat. The search will conduct a random walk.
- Ridges: a ridge may have steeply sloping sides, so that the search reaches the top of the ridge with ease, but the top may slope only very gently toward a peak. Unless there happen to be operators that move directly along the top of the ridge, the search may oscillate from side to side, making little progress. If the problem is NP-complete, then in the algorithm cannot do better than exponential time.

Deborah R. Carvalho and Alex A. Freitas [20] proposed a hybrid decision tree algorithm method. The central idea of this hybrid method involved the concept of small disjuncts in data mining.

A hybrid approach using rule induction and clustering techniques maximizes the accuracy resulting in fast processing time. This approach can obtain better result than previous work. This can also improves the traditional algorithms with good result. Due to their nature, small disjuncts are error prone. It covers only a few small disjunct of examples.

R. C. Barros, D. D. Ruiz, and M. P. Basgalupp [21] suggested use of Model Trees. These trees are similar to regression trees, which are hierarchical structures for predicting continuous dependent variables. At each leaf

node they hold a linear regression model that predicts the class value of instances that reach the leaf. The only difference between regression tree and model tree in duction is that, for the latter, each node is replaced by a regression plane instead of a constant value. Hence model trees are more sophisticated than either regression trees or linear regression. The tree induction of model trees through the greedy algorithms is sequential in nature and locally optimal at each node split, which means that convergence for a global optimal solution is hardly feasible. In addition, minor modifications in the training set often lead to large changes in the final model due to the intrinsic instability of these algorithms

L. Breiman, "Random forests," Hong Hu, Jiuyong Li, Hua Wang ,Grant Daggard ,Mingren Shi Ensemble methods [22][23] were proposed to take advantage of these unstable algorithms by growing a forest of trees from the data and averaging their predictions. Techniques that aggregate the prediction of multiple models in order to improve predictive performance are known as ensemble methods. They tend to perform better than any single model alone when in conformity with two necessary conditions: (i) the base models should be independent of each (ii) the base models should do better than a model that performs random guessing. Well-known examples of ensembles are:

1. Bagging [Leo Breiman in 1996] - the training set is sampled (with replacement) according to a uniform probability distribution, and for each sample a base model is trained. Bagging's effectiveness depends on the (in)stability of the base models. If a base modelis unstable, bagging helps to reduce the errors associated with random fluctuations in the training data (reduction of variance). Inversely, if a base model is stable, it means that the error of the ensemble is caused by bias in the base model, which cannot be solved by bagging.
2. Boosting [Freund and Schapire]-the distribution of the training set is changed iteratively so that the base models will focus on examples that are hard to predict. Unlike bagging, boosting assigns a weight to each training instance and dynamically changes the weights at the end of each boosting iteration. Generally, the weights assigned to the training instances are used as a sampling distribution to draw a set of samples from the original data. Data is predicted through a weighted voting scheme, where models with a poor predictive performance are penalized.
3.Bagging and Boosting [Tan and Gilbert] - in 2003 used Bagging and Boosting C4.5 decision trees,for Microarray data classification, the results showed that both methods outperform C4.5 single tree on some Microarray cancer data sets.
4. Statistik and Surich [2004]-developed a new BagBoosting method. Their experiments showed that BagBoosting outperforms constantly over Boosting and Bagging methods and achieved a better accuracy result on some Microarray data sets compared with some well-known single classification algorithms such as SVM and kNN.
5. Zhang and et al. (Zhang et al. 2003) proposed a new ensemble decision tree method called deterministic forest which was a modified version of random forests. Instead of re-sampling the training data set, this method selects a specified number of the top splits of the root node and then generates a number of alternative trees. The accuracy of results from deterministic forests are comparable to random forests.

III.Motivation:

DNA microarray is an important technology for studying gene expression. With a single hybridization, the level of expression of thousands of genes, or even entire genome, can be estimated for a sample of cells. Microarray data classification is a complex classification problem that involves a decision-making process. This decision making process has lot of uncertainties since the information related to gene expressions are vague in nature and are very hard to predict as the boundaries between them cannot be well defined. Classification is a machine learning (ML) task that aims at building class distribution models taking into account a set of instances characterized by predictive attributes.

Among the most well-known classifiers are artificial neural networks (ANN), support vector machines (SVMs), decision rules, and decision trees.Decision trees, in particular, are widely used as a comprehensible classification model,since they can be easily represented in a graphical form and also be represented as a set of classification rules, which generally can be expressed in natural language in the form of IF-THEN rules.

Instead of manually improving the design components of a decision tree algorithm as it has been done for the past 40 years, a novel approach of a hyper-heuristic evolutionary algorithm for optimally combining design components from decision-tree algorithms is proposed. By doing so, the hyper-heuristic automatically designs new decision-tree algorithms, which can be tailored to a particular type of data sets, associated with a given application domain. A hyper-heuristic is capable of providing a faster, less-tedious and at least equally effective strategy for improving decision-tree algorithms for particular application domains.

IV.A Novel Approach HEAD-DT:

The newly encountered problems or even new instances of known problems face some stumbling block despite the success of heuristic search methods in solving real-world computational search problems. These difficulties arise mainly from the remarkable range of parameters or algorithm choices involved when using this approaches, and the lack of guidance as to how to proceed for select them. Another hitch of these techniques is that state-of-the-art approaches for real-world problems tend to represent bespoke problem-specific methods which are expensive to develop and maintain. Hyper-heuristics comprise a set of approaches with the common goal of automating the design and tuning of heuristic methods to solve hard computational search problems [24]. A hyper-heuristic is a heuristic search method that solicits to automate, often by the embodying the machine learning techniques, which includes the process of selecting, combining, generating or adapting several simpler heuristics to efficiently solve computational search problems. One of the motivations for studying hyper-heuristics is to build systems which can handle classes of problems rather than solving just one problem, which can be achieved by combing through an evolutionary algorithm, components or building-blocks of human designed heuristics.

HEAD-DT individuals are collections of building blocks of DT algorithms. Every individual component is encoded as an integer string as shown in figure below , and each gene has a different range of supported values. We divided the genes in four categories that represent the major building blocks of a DT algorithm: (i) split genes (ii) stopping criteria genes (iii) missing values genes and (iv) pruning genes.

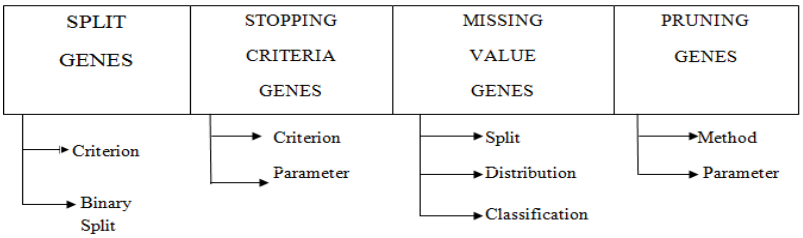

Fig : Major building blocks of Decision Tree

1. Split Genes

These genes are concerned with the task of selecting the attribute to split the data in the current node of the decision tree. A decision rule based on the selected attribute is thus generated, and the input data is filtered according to the outcomes of this rule, and the process continues recursively. The two genes to model the split component of a DT algorithm are Criterion gene and Binary Split Gene. The first gene, criterion, is an integer that indexes one of the splitting criteria. The motivation behind hyper-heuristics is to raise the level of generality at which search methodologies can operate. In the context of decision trees, instead of having an evolutionary algorithm searching for the best decision tree to a given problem (a regular meta-heuristic approach [21][25], the generality level is raised by having an evolutionary algorithm searching for the best decision-tree induction algorithm that may be effectively applied to several different classification problems (a hyper-heuristic approach).

2. Stopping Criteria Genes

The top-down induction of a DT is recursive and it keeps on going until a stopping criterion (per-pruning) is satisfied. Different stopping criterion used are as :

A) Class homogeneity: when all instances that reach a given node belong to the same class, then the nodes are not split further

B) Maximum tree depth: a parameter tree depth can be specified to avoid deep trees.

C) Minimum number of instances for a non-terminal node: a parameter minimum number of instances for a non-terminal node can be specified to alleviate the data fragmentation

D) Accuracy threshold within a node: when a parameter reaches the accuracy which is specified for halting the growth of the tree within a node i.e. majority of instances have reached a defined threshold.

3. Missing Values Genes

Missing values can be an issue during tree induction and also during classification therefore handling missing values is an important task in decision tree induction. During tree induction there are two instances where missing value are need to be dealt with are as :

A) Split Gene

B) Distributed Gene

During tree deduction i.e. while classification of tree , it is important to deal with missing values in the test set :

C) Classification gene

4. Pruning Genes :

XIII. Pruning is usually performed in decision trees for enhancing tree comprehensibility by reducing its size while maintaining (or even improving) accuracy. It was originally conceived as a strategy for tolerating noisy data, however it was found to improve decision tree accuracy in many noisy data sets [5][27].Pruning helps to avoid over-fitting to the training set and to reduce the size of decision tree which makes simpler interpretation for users. The well-known approaches for pruning a decision tree are :

- Error-based pruning (EBP) [27]
- Cost-complexity pruning (CCP) [28]
- Pessimistic error pruning (PEP) [29]
- Reduced error pruning (REP) [29]
- Minimum error pruning (MEP) [30], [31]

V.Conclusion

In this paper, we studied about previously implemented techniques to classify microarray data using decision trees. Decision trees are one of the most recurrently used representations for classifiers .We discussed the relative strengths and weaknesses of the several methodologies proposed in literature. These algorithms have been manually improved for the last 40 years, resulting in a great number of approaches for each of their design components. HEAD-DT, a hyper-heuristic algorithm improves designs top-down decision-tree induction algorithm by automating the manual classification process.

Acknowledgement

We are glad to express our sentiments of gratitude to all who rendered their valuable guidance to us. We would like to express our appreciation and thanks to Prof. Dr. P. C. Kulkarni, Principal, Gokhale Education Society's R. H. Sapat College Of Engineering, Nashik. We are also thankful to Prof. N. V. Alone, Head of Department, Computer Engg., G. E. S. R. H. Sapat College of Engg., Nashik. We thank the anonymous reviewers for their comments.

References

[1]. Schena, M., Shalon, D., Davis, R.W., and Brown, P. O. (1995).Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science, 270(5235):467–470.
[2]. Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M.Mesirov, J. P., Coller, H., Loh, M., Downing, J. R., Caligiuri, M. A., Bloomfield,C. D., and Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science, 286:531–537.
[3]. J. Han, Data Mining: Concepts and Techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.
[4]. P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining,(First Edition). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. 2005.
[5]. J. R. Quinlan, "Induction of decision trees," Mach. Learn., vol. 1, no. 1, pp. 81–106, 1986
[6]. C4.5: programs for machine learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
[7]. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Wadsworth, 1984
[8]. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, 2nd ed. Springer Series in Statistics. Springer, September 2009.
[9]. G. Seni and J. Elder, Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions. Morgan and Claypool Publishers, 2010.
[10]. A. A. Freitas, D. C. Wieser, and R. Apweiler, "On the importance of comprehensible classification models for protein function prediction," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 7, no. 1, 2010.
[11]. A. A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms. Secaucus, NJ, USA: Springer-Verlag New York, Inc.2002.
[12]. Soft Computing for Knowledge Discovery and Data Mining. Springer US, 2008, ch. A Review of evolutionary Algorithms for Data Mining, pp. 79–111.
[13]. A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing (Natural Computing Series). Springer, October 2008
[14]. L. Hyafil and R. Rivest, "Constructing optimal binary decision trees is np-complete," Information Processing Letters, vol. 5, no. 1, pp. 15–17,1976.
[15]. T. Hancock, T. Jiang, M. Li, and J. Tromp, "Lower bounds on learning decision lists and trees," Information and Computation, vol. 126, no. 2, May 1996.
[16]. H. Zantema and H. Bodlaender, "Finding small equivalent decision trees is hard," International Journal of Foundations of Computer Science,vol. 11, no. 2, pp. 343–354, 2000.
[17]. G. E. Naumov, "Np-completeness of problems of construction of optimal decision trees," Soviet Physics: Doklady, vol. 36, no. 4, 1991.
[18]. R. C. Barros, R. Cerri, P. A. Jaskowiak, and A. C. P. L. F. de Carvalho, "A Bottom-Up Oblique Decision Tree Induction Algorithm," in 11th International Conference on Intelligent Systems Design and Applications,2011, pp. 450 –456
[19]. Artificial Intelligence A Modern Approach Stuart J. Russell and Peter Norvig
[20]. Deborah R. Carvalho and Alex A. Freitas , "A hybrid decision tree/genetic algorithm for coping with the problem of small disjuncts in Data Mining". Proc 2000 Genetic and Evolutionary Computation Conf. (Gecco-2000), 2000, 1061-1068. Las Vegas, NV, USA. July.
[21]. R. C. Barros, D. D. Ruiz, and M. P. Basgalupp, "Evolutionary model trees for handling continuous classes in machine learning," Information Sciences, vol. 181, pp. 954–971, 2011.
[22]. L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp.5–32, 2001.

[23]. Hong Hu, Jiuyong Li, Hua Wang ,Grant Daggard ,Mingren Shi "A Maximally Diversified Multiple Decision Tree Algorithm for Microarray Data Classification."
[24]. A Survey of Hyper-heuristics Edmund K. Burke, Matthew Hyde, Graham Kendall Gabriela Ochoa, Ender Ozcan and Rong Qu
[25]. R. C. Barros, M. P. Basgalupp, A. C. P. L. F. de Carvalho, and A. A. Freitas, "A survey of evolutionary algorithms for decision tree induction," IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, vol. 42, pp. 291–312, May 2012.
[26]. J. R. Quinlan. Simplifying decision trees. International Journal of Man-Machine Studies, 27:221–234, 1987
[27]. J. R. Quinlan, C4.5: programs for machine learning. San Francisco, CA, USA: Morgan Kaufmann, 1993.
[28]. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Wadsworth, 1984.
[29]. "Simplifying decision trees," International Journal of Man-Machine Studies, vol. 27, pp. 221–234, 1987.
[30]. T. Niblett and I. Bratko, "Learning decision rules in noisy domains," in 6th Annual Technical Conference on Expert Systems, 1986, pp. 25–34.
[31]. B. Cestnik and I. Bratko, "On estimating probabilities in tree pruning," in EWSL'91. Springer, 1991, pp. 138–150.