International Journal of Computer Science and Informatics

Volume 1 | Issue 3 Article 14

January 2012

Improved Genetic Algorithm Based Classification

Keshavamurthy B. N
Department of Electronics & Computer Engineering, Indian Institute of Technology, Roorkee, Uttarakhand, India, kesavdec@iitr.ernet.in

Asad Mohammed Khan
Department of Electronics & Computer Engineering, Indian Institute of Technology, Roorkee, Uttarakhand, India, asadmuec@iitr.ernet.in

Durga Toshniwal
Deptt. of Electronics and Computer, Indian Institute of Technology, Roorkee, India, durgafec@iitr.ernet.in

Improved Genetic Algorithm Based Classification

Keshavamurthy B. N , Asad Mohammed Khan & Durga Toshniwal

Department of Electronics & Computer Engineering, Indian Institute of Technology, Roorkee, Uttarakhand, India
E-mail : kesavdec@iitr.ernet.in, asadmuec@iitr.ernet.in, durgafec@iitr.ernet.in

Abstract - Classification is the supervised learning technique of data mining which is used to extract hidden useful knowledge over a large volume of databases by predicting the class values based on the predicting attribute values. Of the various techniques, the most widely talked ones include decision tree, probabilistic model and evolutionary algorithms. Recently, the probabilistic and evolutionary techniques are most worked upon, because of the fact that probabilistic models yields high accuracy when there is no attribute dependency in the existing problem and evolutionary algorithms are used to obtain optimal solution over a large search space very quickly when there is less information known about a problem and problem posses attribute dependency. Though there are tradeoffs in each model still there are scopes to improve upon the existing. The proposed approach improves the evolutionary technique such as genetic algorithm by improving the fitness function parameters. Also, in this we compare the genetic algorithm results with Naïve Bayes algorithm results. For the experimental work we have used the nursery data set taken from the UCI Machine Learning Repository.

Keywords - *Naïve Bayes, classification, genetic algorithm, data mining.*

I. INTRODUCTION

With the rapid advancement of technology, large volume of data easily collected from day to day management of the recent applications such as retail-business, social and health services administration and universities. Intuitively, this large amount of raw stored data contains valuable hidden knowledge, which could be used to improve the decision-making process of an organization. It is tedious and difficult to analyse such large voluminous data and establishing relationship between multiple features manually.

Thus data mining techniques such as classification, association and clustering are generally used to extract the hidden, previously unseen knowledge from voluminous of databases.

Of the various data analysis technique classification is a supervised machine learning technique which makes predictions about future class instances by mapping instances of testing data to the predefined class labels which is learn from the supplied instances of classes with class labels. There are several models in classifications such as probabilistic model, evolutionary algorithmic model etc. In the present work we analyse the voluminous data using Naïve Bayes and genetic algorithm.

A lot of research papers have discussed the classification analysis. One of important the key issue of applying the probabilistic model and evolutionary to the same dataset is not been covered in existing research work. This paper address issues related to the data analysis using efficient probabilistic model such as Naïve Bayes and effective evolutionary algorithm such as genetic algorithm. We have applied above mentioned algorithms to the nursery dataset.

The rest of the paper is organized as follows: Section II gives the basic definitions followed by related research work. Section III presents the proposed approach. Section IV gives experimental results. Section V includes conclusion

II. BACKGROUND

Definition 1(**Fitness Function**): Fitness Function for rule discovery is the key to the convergence of the GA. The basic parameters to measure the fitness function are Confidence and Completeness.

Definition 2(**Confidence Factor**): Let a rule be of the form: IF A THEN C. Confidence factor (CF) of the rule defined as:

$$CF = |A \& C| / |A| \quad (1)$$

where |A| is the number of tuples satisfying all the conditions in the antecedent A of the rule and |A&C| is the no of tuples satisfying both the antecedent A and the Consequent C i.e. the predicted class.

Definition 3 (**Completeness**): Completeness of the rule is the fraction of tuples having the predicted class C that is actually covered by the rule antecedent.

Completeness = TP / (TP + FN). (2)

Where TP = True Positives = Number of tuples satisfying A and C, FP = False Positives = Number of tuples satisfying A but not C, FN = False Negatives = Number of tuples not satisfying A but satisfying C and TN = True Negatives = Number of tuples not satisfying A nor C.

Using False Positive and True Positive parameters we rewrite the confidence as follows:

Confidence = TP / (TP + FP) (3)

A. *Naïve Bayes technique.*

Bayesian networks are often used for classification problems, in which a learner attempts to construct a classifier from a given set of training examples with class labels. Bayes theorem is a statistical principle for combining prior knowledge of the class with new instances gathered from data. Basically Bayes theorem works with joint conditional probability. Conditional probability $P(A = a/B = b)$ refers to the probability of variable A takes a value a given that variable B is observed to have the values b. A joint conditional probability for A and B is given by:

$$P(A, B) = P(B/A) \times P(A) \quad (4)$$

$$P(A, B) = P(A/B) \times P(B) \quad (5)$$

By rearranging above expression (4) & (5) we get

$$P(B/A) = \frac{P(A/B)P(B)}{P(A)} \quad (6)$$

The Naïve Bayes classifier given below in equation (7) estimates the class conditional probabilities with the basic assumption that the attributes are conditionally independent [1]:

$$P(A/B = b) = \prod_{i=1}^{n} P(A_i / \ B = b) \quad (7)$$

where each attribute set A ={a_1,a_2,…a_n} contains n attributes.

There is a tremendous work has been done on Naïve Bayes classification. The initial literature on Naïve Bayes classification found in [2]. In [3], The Naïve Bayes classification has been applied for decision tree. In [4], Naïve Bayes classification has been applied with K- nearest neighbor algorithm. In [5] [6] [7], Naïve Bayes applied for attribute subspace computation. In [5], it uses forward greedy technique to select an attribute through the entire attribute space. In [6], it selects the attribute subspace by applying evolutionary search space process. In [7], optimal wrapper method of subset selection has been introduced.

B. *Genetic Algorithmic Technique*

Classification using Genetic Algorithm discover the high-level prediction rules by performing a global search in cope with better attribute interaction than the greedy rule induction algorithms where there is no attribute dependency. Genetic algorithm requires less information about the problem. The genetic algorithm uses rule based classifiers in general. One can design an individual to represent prediction (IF-THEN) rules. Classification rules can be considered a particular kind of prediction rules where the rule antecedent ("IF part") contains a combination - typically, a conjunction - of conditions on predicting attribute values, and the rule consequent ("THEN part") contains a predicted value for the goal attribute [8].

Examples:

i. IF (Housing=convenient) $\wedge$ (Finance=convenient) $\wedge$ (Social=slightly problematic) $\wedge$ (Health= recommended) $\Rightarrow$ (class = very recommended)

ii. IF (Health=not recommended) $\Rightarrow$ (class = not recommended)

The classification problems have been well studied as a major category of data analysis data analysis in genetic algorithm which generally uses rule based approach [9]. The initial literature on rule based genetic algorithm was found in [10]. It describes the kind of rule assessment schemes which have been proposed for rule discovery systems. In [11], it reviews the definition, theory and extent applications of classifier systems. The survey of classification using genetic algorithm rule based approach which includes Michigen versus Pittsburg approach in [12]. The genetic algorithms have been associated with greedy techniques in [13] [14] [15] [16] [17]. The recent ten years survey of evolutionary algorithms described in [18].

III. PROPOSED APPROACH

The present work improves the rule based genetic algorithm classifier by improve upon the fitness function parameter modification. Also, it compares the results with the probabilistic approach such as Naïve Bayes which is always gives better results and very efficient in case there is no attribute dependency in the problem, which is not true in most of the real world problem including nursery dataset we have considered for our work.

The framework for the proposed approach is shown in Figure1 and proposed approach of GA uses the following steps:

1. Representation of the rule

In general, let a rule be of the form: *IF* $A \Rightarrow C$, where A is the antecedent and C is the consequent. The rule antecedent A may be consisting of attributes A_1 A_2.... A_i.... A_n. Let us consider attributes $A_1, A_2 and\ A_3$ i.e *'Student_of', 'year', 'Semester'* and their attributes values are *{Arts, Engineering, Medical, Law},{1st , 2nd , 3rd, 4th },{1st , 2nd , 3rd, 4th , 5th , 6th 7th, 8th }*. Gene representation for the different attribute values, in case of binary chromosomes will be as follows:

TABLE I : BINARY GENE REPRESENTATION FOR STUDENT_OFF

Student_of	Bit1	Bit2	Bit3	Bit4
Arts	1	0	0	0
Engineering	0	1	0	0
Medical	0	0	1	0
Law	0	0	0	1

TABLE II : BINARY GENE REPRESENTATION FOR YEAR

Year	Bit1	Bit2	Bit3	Bit4
1st	1	0	0	0
2nd	0	1	0	0
3rd	0	0	1	0
4th	0	0	0	1

TABLE III : BINARY GENE REPRESENTATION FOR SEMESTER

Sem	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8
1st	1	0	0	0	0	0	0	0
2nd	0	1	0	0	0	0	0	0
3rd	0	0	1	0	0	0	0	0
4th	0	0	0	1	0	0	0	0
5th	0	0	0	0	1	0	0	0
6th	0	0	0	0	0	1	0	0
7th	0	0	0	0	0	0	1	0
8th	0	0	0	0	0	0	0	1

The gene representation for the different attribute values, in case of numeric chromosomes will be as follows:

TABLE IV : NUMERIC GENE REPRESENTATION FOR STUDENT_OF AND YEAR

Student_of	Bit1/gene	Year	Bit1/gene
Arts	1	1st	1
Engineering	2	2nd	2
Medical	3	3rd	3
Law	4	4th	4

TABLE V : NUMERIC GENE REPRESENTATION FOR SEMESTER

Semester	Bit1/gene	Semester	Bit1/gene
1st	1	5th	5
2nd	2	6th	6
3rd	3	7th	7
4th	4	8th	8

a) For Example: IF (*Student_of = Engineering*) ∧ (*Year = 2nd*) ∧ (*Semester = 6th*). Then the rule Antecedent can be represented as in binary chromosome representation: 0100| 0100|00000100. In numeric chromosome representation attribute values can be assigned numeric value as shown in Table IV and Table V:

b) A flag bit may be associated with every attribute, which indicates presence or absence of an attribute in the antecedent part of the rule.

For example: IF *(Student_of = Engineering)* ∧ *(Year = 2nd)* ∧ *(Semester = 6th)*. The binary and numeric chromosome representation for this expression is as follows: Binary chromosome: 0100|***1***|0100|***1***|00000100|***1***| and its Numeric chromosome: 2|***1***|2|***1***|6|***1***|.

IF *(Student_of = Engineering)* ∧ *(Semester = 6th).* Binary Chromosome Representation: 0100|***1***|0100|***0***|00000100|***1***| and its Numeric chromosome: 2|***1***|2|***0***|6|***1***|. Here entries of bold and italics show the flag bit.

The presence of a flag bit helps in the process of chromosomes crossover. For example in the absence of the flag bit the following rule will have the binary or numeric representation as follows:

IF (*Student_of = Engineering)* ∧ *(Year = 2nd).* Binary Chromosome: 0100|***1***|0100|***1***|. Numeric chromosome: 2|***1***|2|***1***|.

IF *(Student_of = Engineering)* ∧ *(Semester = 6th).* Binary Chromosome: 0100|***1***|00000100|***1***|. Numeric

chromosome: 2|***1***|6|***1***|. As a result of a crossover operation, one of the children might be an invalid individual (i.e. a rule with contradicting conditions), such as:

IF *(Student_of = Engineering)* $\wedge$ *(Year = 6th).*

IF *(Student_of = Engineering)* $\wedge$ *(Semester = 2nd).*

Here domain of attribute Year is restricted to 4. For the current Problem the Numeric representation of the chromosomes is considered.

2. Crossover

Crossover selects genes from parent chromosomes and creates a new offspring. The single point crossover is performed by choosing randomly some crossover point and everything before this point copy from a first parent and then everything after a crossover point copy from the second parent. Crossover can then look like this (| is the crossover point) as follows:

TABLE VI : CROSSOVER COMPUTATION

Chromosome 1	11011 | 00100110110
Chromosome 2	11001 | 11000011110
Offspring 1	11011 | 11000011110
Offspring 2	11001 | 00100110110

3. Mutation

This is to prevent falling all solutions in population into local optima. Mutation changes randomly the new offspring.

For example: 2|***0***|2|***1***|4|***0***|3|***1***|. For mutation a random number is chosen using the *random()* function. For example an attribute 'parent' has values range {*0, 1, 2*} for usual, pretentious and great_pretentious. Expression *rand()%3* will result in the mutated value of the attribute '*parent'*. For the current problem 'multipoint non-uniform mutation' is performed.

4. Selection

Roulette wheel selection is performed. In roulette wheel selection, individuals are given a probability of being selected that is directly proportionate to their fitness. Two individuals are then chosen randomly based on these probabilities and produce offspring.

5. Fitness Functions for Rule Discovery

The rule discovered should have high Confidence as well as high Completeness factor, but there is always a trade of between the two, as they are inversely related. So a good rule will be one with high Confidence factor and a significant amount of Completeness.

A fitness function can be defined by combining Confidence and Completeness such as:

Fitness $= Confidence^n * Completeness$; Where n>=2 (8)

Since more weight should be given to the Confidence then Completeness, thus an exponential power of more than 1 is associated with Confidence.

Example: (a) Rule R1 with a Confidence of 0.6 and Completeness of 0.4

(b) Rule R2 with a Confidence of 0.4 and completeness of 0.6

If n=2, Then for R1 Fitness = 0.6*0.6*0.4=0.144. For R2 Fitness= 0.4*0.4*0.6=0.096. So rule R1 will be selected rather than rule R2. More the value of n more will be the weight assigned to the Confidence.

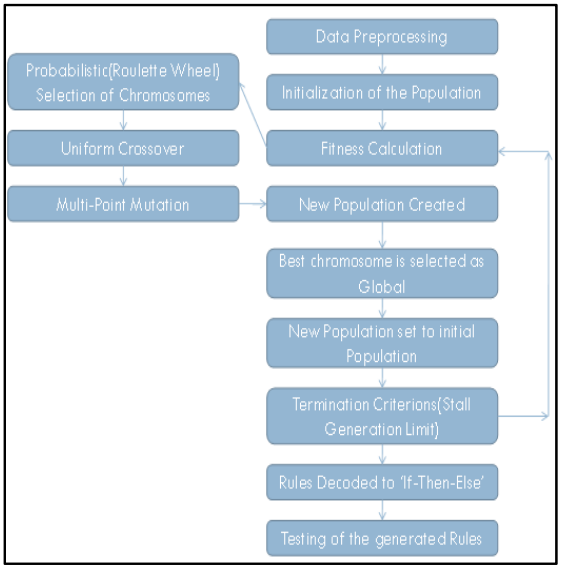

Fig. 1 : Frame work for evolutionary rule-based-classification

IV. RESULTS

Nursery database [20] was developed to rank the applications for nursery school admission in Europe countries such as Ljubljana and Slovenia, and the rejected applications needed an objective explanation. This dataset contains 12960 instances of 8 attributes. For our experimental work we have considered 40% distribution of each class for training and 60% for testing.

Parameters used for proposed genetic algorithm approach are:

Population Size: 100

No of Generations: 500

Probability of Crossover: 0.75

Probability of Mutation of flag bit per bit: 0.02

Probability of Mutation of attribute bit per bit: 0.08

Overall Accuracy Calculation

Class Distribution (number of instances per class)

Class	N	N[%]
not_recommended	4320	(33.33 %)
recommend	2	(0.015 %)
very_recommended	328	(2.53 %)
priority	4266	(32.92 %)
special_priority	4044	(31.20 %)

Total Instances = 12960

Overall_Accuracy= $\sum_{i=1}^{5} Confidence\_Average\_of\_Class\_'i' * Fraction\_of\_Class\_'i'\_tuple\_in\_dataset$

'Confidence_Average_i' is the average of confidence of rules class-i. Refer Table VIII for Confidence and Completeness values.

Confidence_Average_of_Class_1= 1*1+1*1+1*1) / (1+1+1) =1.

Confidence_Average_of_Class_2=(1*0.5+1*0.5+0.67* 1) / (0.5+0.5+1) = 0.83.

Similarly 'Confidence_Average_of_class' is computed for other class values as shown in Table VII.

TABLE VII : AVERAGE CONFIDENCE FOR EACH CLASS

Class_Label	Confidence_Average_of_class_'i'
Not_recommend	1
Recommend	0.83
Very_recommend	0.66
Priority	0.62
Spec_priority	0.77

Overall_Accuracy=0.3333*1+0.00015*0.83+0.0253*0. 66+0.3292*0.62+0.3120*0.77 = **0.7965 Or 79.65%.**

TABLE VIII : FITNESS VALUES FOR TEST DATASET

Class Labels	**Mined Rules**	**Confidence**	**Completeness**	**Fitness**
not recommend	1) If(Health=not_recom) Then (class = not_recom)	1	1	1
	2) If(Health=not_recom) Then (class = not_recom)	1	1	1
	3) If(Health=not_recom) Then (class = not_recom)	1	1	1
recommend	1) If(parent=usual) AND (Has_nurs=proper) AND (Form=complete) AND (Children=1) AND (Housing=convenient) AND (Finance=convenient) AND (Social=slightly_prob) AND (Health=recommended) Then (class = recommend)	1	0.5	0.5
	2) If(parent=usual) AND (Has_nurs=proper) AND (Form=complete) AND (Children=1) AND (Housing=convenient) AND (Finance=convenient) AND (Social=slightly_prob) AND (Health=recommended) Then (class = recommend)	1	0.5	0.5
	3) If(parent=usual) AND (Has_nurs=proper) AND (Form=complete) AND (Children=1) AND (Housing=convenient) AND (Finance=convenient) AND (Health=recommended) Then (class = recommend)	0. 67	1	0. 67

very recommend	1) If(Has_nurs=less_proper) AND (Housing=convenient) AND (Finance=convenient) AND (Social=slightly_prob) AND (Health=recommended) Then (class = very_recom) 2) If(Has_nurs=proper) AND (Housing=convenient) AND (Finance=convenient) AND (Social=slightly_prob) AND (Health=recommended) Then (class = very_recom)	0. 67 0.06	0. 09 0.94	0.06 0.06
priority	1) If(Health=recommended) Then (class = priority) 2) If(parent=usual)AND (Health=priority) Then (class = priority) 3) If(parent=pretentious) AND (Has_nurs=less_proper) AND (Health=priority) Then (class = priority)	0.56 0.67 1	0.56 0.23 0.07	0.32 0.15 0.07
special priority	1) If(Has_nurs=very_crit) Then (class = spec_prior) 2) If(Has_nurs=very_crit) AND (Health=priority) Then (class = spec_prior) 3) If(parent=great_pret) AND(Health=priority) Then (class = spec_prior)	0.59 0.86 0.99	0.37 0.30 0.21	0.22 0.26 0.21

TABLE IX : COMPARISION OF NAÏVE BAYES V/S GENETIC ALGORITHM

Sl. No.	Description	Records/ Intances	Percentage of Accuracy
1	Naïve Bayes Classifier	12960	49.32
2	Genetic Algorithm Classifier	12960	79.65

The results of GA have been compared with the Naïve Bayes classifier over here [19]. We got good results using GA due to the fact that there is very much attribute dependency in the problem which is analogous to many real world scenarios.

V. CONCLUSION

Generally, Naïve Bayes always gives better results in comparison with genetic algorithm when there is no feature dependency in the problem domain which may not be true in the real world problems. Thus, we apply both the algorithms for nursery dataset.

The key factor of GA is its fitness function, the convergence of search space is directly proposal to the effectiveness of fitness function in other words if fitness function is good then better the convergence of GA for a given problem. Also genetic operator refinement for a problem plays a vital role for the convergence of search space to an optimal solution.

Finally, proper mutation probability distribution for an attributes with different range of attribute value and its representation in the form of chromosomes to avoid redundancy and inconsistency. We can further improve upon these factors of GA in future.

VI. ACKNOWLEDGEMENT:

This research work was partially funded by IBM Centre for Advanced Studies Projects (CAS), currently going on at IIT Roorkee.

REFERENCES

[1] Vipin kumar, Michael steinbach and Pang-Ning Tan, “Introduction to Data mining ,” Third Impression, Pearson Education, 2009,ISBN:978-81-317-1472-0.

[2] Langley P., Iba, W. and Thomas K. “An analysis of Bayesian classifiers,” Proceedings of the Tenth National Conference of Artificial Intelligence, AAAI Press, Stanford, 1992, pp. 223–228.

[3] Kohavi R., ”Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid,” Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI Press, Stanford 1996, pp. 202–207.

[4] Frank E., Hall M., and Pfahringer B.,” Locally Weighted Naive

[5] Bayes," Proceedings of the Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, Seattle 2003, pp. 249–256.

[6] Langley P. and Sage, S., "Induction of selective Bayesian classifiers," Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, 1994, pp. 339-406.

[7] Jiang L., Zhang H., Cai Z and Su J. "Evolutional Naive Bayes," Proceedings of the First International Symposium on Intelligent Computation and its Applications, ISICA, China University of Geosciences Press, 2005, pp.344–350.

[8] Kohavi R. and John G., "Wrappers for Feature Subset Selection," Artificial Intelligence journal, special issue on relevance, 97(1-2),1997, pp. 273–324.

[9] Liangxiao Jiang, Dianhong Wang, Zhihua Cai, and Xuesong Yan, "Survey of Improving Naive Bayes for Classification," Proceedings of ADMA International conference,Springer, 2007 pp. 134–145.

[10] W. Banzhaf P., Nordin R., Keller and F. Francone, Genetic Programming: An Introduction. Morgan Kaufmann, 1998.

[11] John J . Grefenstette, "Credit assignment in rule based systems based on genetic algorithms," Machine learning, vol. 3, 1988, pp. 225-245.

[12] L.B. Brooker , D.E. Goldberg and J.H. Holland, "Classifier systems and genetic algorithms," Arificial intelligence, vol. 40, 1989, pp.235-282.

[13] Alex A. Freitas, "A survey of evalutionary algorithms for data mining and knowledge discovery,",Springer-verlag, Newyork, ISBN:3-540-43330-9, 2003.

[14] J. Eggermont, J. Kok, and W. Kosters, "Genetic programming for data classification: Refining the search space," Proceedings of the 15th Belgium/Netherlands Conference on Artificial Intelligence (BNAIC'03), 2003, pp. 123–130.

[15] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm,"Proceding of the 13th International conference on Machine Learning, 1996, pp. 148–146. Morgan Kaufmann.

[16] L. Hyafil and R. Rivest, "Constructing optimal binary decision trees is NP-complete," Information Processing Letters, vol. 5 No.1, 1976, pp.15–17.

[17] M. Keijzer, J. J. Merelo, G. Romero and M. Schoenauer, "Evolving objects: A general purpose evolutionary computation library," Proceedings of Evolution Artificielle'01, Springer Verlag ,vol. 2310 2001, pp. 231–244.

[18] W. B. Langdon and S. M. Gustafson, "Genetic Programming and Evolvable Machines," ten years of reviews, journal of Genet Program Evolvable Machines, vol 11, 2010, pp.321–338.

[19] Keshavamurthy B.N., Mitesh Sharma and Durga Toshniwal, "Privacy Preservation Naïve Bayes Classification for a Vertically Distribution Scenario using Trusted Third Party," ARTCom-2010, IEEE Intl. conference,India, 2010, pp.404-407.

[20] Asuncion, A. and Newman, D.J.. UCI Machine Learning Repository.[http://www.ics.uci.edu/~mlearn/ML Repository.html].Irvine, CA: University of California, School of Information and Computer Science,2007.

◈◈◈