Designing a New Fuzzy Genetic Gravity Algorithm for Data Mining

Mostafa Moradi
Department of Computer Engineering, Islamic Azad University (IAU), UAE branch
Dubai, UAE

Mos_moradi1359@yahoo.com

ABSTRACT

Nowadays, due to the high volume of data and their complexity and the human needs to the hidden knowledge in them, using an efficient method is necessary. In this study, our purpose is to present an efficient method of data mining in order to fetch knowledge of input data set. The given proposed method – with the help of fuzzy systems based on rules which are a set of if-then rules – fetch the needed knowledge and classify it. Fuzzy rules are paid attention because they can be interpreted by a human expert. In fact, the our purposed knowledge can be considered as a fuzzy database which improved during the data mining process and with the help of optimum algorithm according to the criteria such as accuracy and ability to interpretation. In order to optimize the obtained fuzzy rules set, a combination of genetic algorithm and the assimilated cooling heuristic is used. The assimilated cooling which is based on statistical mechanics, according to the criteria of accuracy and ability to interpret, try to find a set of fuzzy if-then rules in the state space related to set of rules which can have the best performance and that can escape from the local optimal solutions with the help of genetic algorithm mechanisms. Finally, the proposed method has been implemented in software and applied on a set of UCI data set. The obtained results were compared with the results of the famous methods in this field such as *Support vector, N-bayes, KNN , D-tree, GBML and* they have shown good accuracy and efficiency.

KEYWORDS

Fuzzy If-then rules, gravitational rule, genetic algorithm

1- INTRODUCTION

Nowadays, given the growing volume of data, the need to convert data into a useful method and the extraction importance of this method, data mining has become one of the most important discussions in most fields. Market analysis, medicine, customer retention, science production, exploration and even security issues are some applications of data mining. Data mining is divided into two tasks (processes): clustering and classification [1].

The basic operation in data mining is classification. It is a process that involves detecting a model in which data classes are determined to use this model for data classification. The model obtained for data is usually established based on analyzing a collection of training data in which data are divided into pre-defined and known classes [2, 3].

Today, fuzzy systems which operate based on If-then fuzzy rules are successfully used in great fields, because accuracy, interpretability and applicability in their manipulation, depending on different needs, have made the use of them easy and funny. Recently, fuzzy rule-base systems are widely used for classification questions, while non-fuzzy input vectors are used for one of classified classes. The point is that fuzzy If-then rules are

analyzed and interpreted instructively by experts. Recently several methods have been experienced for the automatic production and calibration of If-then fuzzy rules without need for supporting from human experts [4-6]. But a challenge in establishing fuzzy systems is to ensure that whether we can extract optimized classified rules from training data and that the extracted data must be interpretable and accurate for human. Gravitational algorithm is an optimized search algorithm operating based on the gravitational rule between different objects; thus, when different masses are located in a hypothetical gravitational coordinates, they tend to reach a gravitational balance after a while; so that all the forces acting on objects will tend to zero [8]. Recently. the gravitational algorithm is used foı problems to search the optimal solution among existing solutions [9]. As a production and optimization tool, the genetic algorithm is used in designing fuzzy rule-base systems [10]. GA-base studies conducted on the design of fuzzy Rule-base systems are usually relied on Genetic-base machine learning methods.

In the genetic algorithm, exchange and mutation are commonly used for producing new generations. The Pittsburgh method is similar to the Genetic Algorithm, where each sample is a collection of rules representing a complete solution of the learning problem, while the Michigan method uses a little different solution from learning processes. Each sample represents a single rule and leads to a solution through communication with other rules in the rules population. In this paper, we generate a Rule-base system based on if-then fuzzy rules which will be based on a combination of genetic algorithm and gravitational rule to improve two important issues in data mining: interpretability and accuracy of rules in fuzzy systems. The proposed method is tested using the machine learning data obtained from UCI and are compared with several famous classification methods [11]. Section 3 describes fuzzy-genetic-gravitational method for classification. Section 4 presents experimental results and comparing the proposed method, and the last section is the conclusion.

2- FUZZY CLASSIFY

Fuzzy systems classification Suppose in a n-dimensional pattern space the pattern classification problem is a c-class problem with continuous attributes. Also suppose m real patterns are as follows $X_p = (X_{p1} \ldots X_{pn})$, p=1,2…,m .

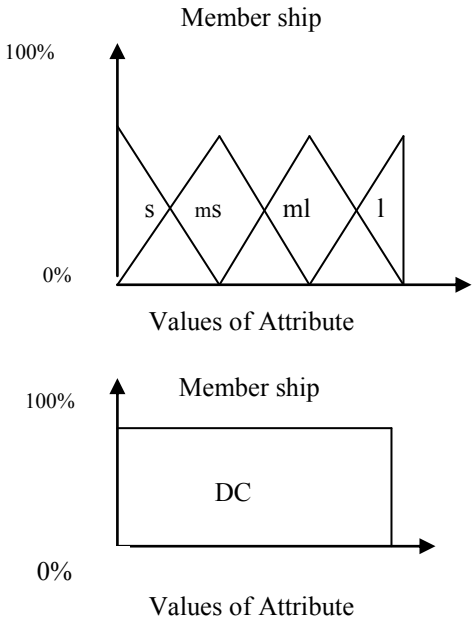

Figure 1: The use of fuzzy prior set in this article: small, medium small, -medium large, large, don't care[7].

For example, there are training samples in c-class. Since the pattern space is $[0,1]^n$, each sample's value is $X_{pi} \in [0,1]^n$ for I =1, ..., n, p = 1,2, ..., n. in the proposed fuzzy classification system. The fuzzy rule is used as follows:

Rule R_j: if x_1 is A_{j1} and x_n is A_{jn} Then Class C_j j=1…n (1)

where A_{Ij} is the j^{th} label of the fuzzy rule, A_{j1} ... A_{jn} is the equivalent fuzzy set on

the range [0,1], c_j is the result class and fuzzy rule, and $CFACT_j$ is the certainty degree of the R_j fuzzy rule. In computer simulation, we use the linguistic values in Figure 1 as fuzzy resultant sets. The number of fuzzy rules is 5^n *if-then*, which is a large number and it is impossible that we use all 5^n if-then fuzzy rules in a fuzzy Rule-base systems for large values of n. The proposed procedure seeks to produce a collection of fuzzy rules with very high classification accuracy. The result class and certainty degree of each fuzzy rule can be obtained according to the following procedure [4].

Stage 1: Calculating the compatibility of each training pattern, $X_p = (X_{p1} \ldots X_{pn})$, with the R_j rule with the operator production

$$\mu_j(x) = \mu_{j1}(x_1) \times \ldots \times \mu_{jn}(x_n) \quad (2)$$

P=1,2,3, ... , m where $(XP_i)\mu_{j\,i}$ is the membership function of $A_{j\,i}$.

Stage 2*:* Calculating the relative sum of compatibility degrees for experienced patterns with R_j.

$$B_{class\,h} = \sum_{xp \in classh} \mu_{Rj} \quad (Xp) \quad h = 1 \ldots c \quad (3)$$

where $\beta_{class\ h}$ is the addition of compatibility degrees for training samples in Class H according to the rule R_j.

Stage 3: Obtaining the class that has the highest value of: $B_{Class\,h}(R_j)$

$$B_{Class\,c\,j}(R_j) = \text{Maximum}\{\beta_{Class\,1}(R_1), \ldots, \beta_{Class\,c}(R_j)\}. (4)$$

If more than one class has the highest value, the C_j result class is not obtained for the R_j rule uniquely. In this study, we assume that C_j is equal to φ.

Stage 4: If the result class is φ, suppose that the $CFACT_j$ certainty degree of the R_j rule is zero. Otherwise $CFACT_j$ is obtained from the following formula.

$$CFACT_j = \frac{\beta\ class_c\ (Rj) - \bar{\beta}}{\sum_{h=1}^{c} \beta\ classh(Rj)} \quad (5)$$

$$\bar{\beta} = \frac{\sum_{h=cj} \beta_{Class\ h}\ (R\,j)}{C-1} \quad (6)$$

Once a rule set is given, an input sample is classified using the single winner rule of R_W in the S rule set which is obtained from the following relation:

$$\mu_w(x_p).CFACT_j = \text{maximum}\{\mu_j(x_p).CFACT_j \mid R_j \in S) \quad (7)$$

So that it has the maxim winner rule, compatibility and degree. The classification will not be accepted if no if-then fuzzy rule compatible with the x_p input sample is obtained.

3 - DESIGNING THE FUZZY-GENETIC- GRAVITATIONAL CLASSIFIER

For our designer, we propose a fuzzy-genetic-gravitational classifier during the following steps:

Stage1: Establishment of the initial set of if-then fuzzy rules and setting the gravitational acceleration to an initial value.

Stage2: Valuation of the current set of if-then fuzzy rules using an evaluation function (EFACT).

Stage3: Establishment of a new set of initial rules using genetic operations such as exchange and mutation.

Stage 4: If $EFACT_{new} < EFACT_{cur}$, then the new set of rules is accepted. Otherwise, the acceptance of new rules is calculated by the following turbulence function:

$$e^{\frac{-(EFACT\ n - EFACTC)}{a}} \quad (8)$$

where **a** is the initial acceleration.

Stage5: Repeating steps2 through 4 for *k* in each acceleration.

Stage6: Deceleration using the deceleration parameter.

Stage7: End of the algorithm in case of reaching the end of conditions. Otherwise, returning to step 2.

Next, each of the above steps will be explained in detail.

4 - INITIALIZATION

To implement the proposed model, we need to start with a high initial acceleration, but if this beginning value a

$_{init}$ is too high, it will lead to overhead and processing time difference. Thus, the initial acceleration value should be set according to the size and number of masses so that all good acceleration changes or displacement states are acceptable or testable.

Suppose that the number of if-then fuzzy rules in each generation is N_{POP}. To produce an initial generation N_{POP}, if-then fuzzy rules are determined by a random pattern in the test data set. Note that, as mentioned in the previous section, we will suppose gravitational genetic fuzzy systems for each classified class problems separately. So the produced random pattern is determined by training data patterns and their type and class in each set on which it is working. In addition, our random pattern has the most frequent combination compatible with result fuzzy systems and is determined using 5 linguistic values. We can determine the compatibility of fuzzy systems with the randomized pattern by (2). After producing each if-then fuzzy rule, the result class of this rule is determined based on heuristic methods described in the previous stage. The production of each fuzzy rule is accepted if and only if the consequent class is the same as its belonging randomized pattern class. Otherwise, the produced fuzzy rule is rejected. This process is repeated until the class of each produced pattern is determined. Once N_{POP} of the fuzzy if-then rule is established, the fitness value of each rule is determined by the classification of all training patterns and by the if-then fuzzy rule in the current population. The fitness value for each rule is determined using the following formula:

$$Fit(R_j)=NCCP(R_j) \quad (9)$$

where NCCP (R_j) is the number of training patterns correctly classified using the if-then rules for R_j.

4.1 Valuation

The fuzzy if-then rule set for our model must have a high accuracy and interpretability according to its fuzzy nature. So we use the following function for the valuation of each rule.

$$NNCP(S) = m - \sum_{Rj \in s} NCCP(R_j) \quad (10)$$

where m is the number of all patterns in the training data, N is the number of rules in the rule set, EFACT(S) is the number of inaccurate training patterns classified by S.

4.2 Producing Samples

To produce a new if-then fuzzy rule for the next generation, we use a combination of the current generation's if-then rule pairs. Each fuzzy rule in the current generation is selected by the following selection function.

$$P(R) = \frac{fit(R) - fit_{minimum}(s)}{\sum R_k \in S[fit(Rk - fitminimum(s)]} \quad (11)$$

where $fit_{minimum}$ is the minimum fit value of the fuzzy if-then rule in each generation. This procedure is repeated until a predetermined fuzzy rule pair value is selected (note that our selection procedure is the Roulette wheel or proportionate selection). For random selection of pairs of fuzzy rules, we use the Crossover operator for a predetermined number. Notice that the samples selected for the Crossover operation should be different. We use the One-Paint-cross in our simulation. After Crossover, the result classes of the produced samples are estimated. If the class of each sample is different from that of its parents, it will be repeated. For the mutation operator, we use the predetermined number. Each fuzzy result set randomly uses the rule set with a different fuzzy set after the exchange operation for displacement. After mutation, the result class of mutated samples is estimated. If we see the same class as result before mutation, the sample is accepted. Otherwise, the mutation operation is repeated for a predetermined number. After the selection operation, the

ISBN: 978-0-9891305-5-4 ©2014 SDIWC

number of fitness for each produced rule is determined using (8) by two genetic crossover and mutation operators. Finally, a predetermined number of fuzzy rules in the current generation are replaced with the number of new produced rules which is specified by P_{rep}.

4.3 Acceptance of Rules

If the EFACT value for new rules, S_{new}, is less than the S_{cur} value, then the new rule set is accepted and we assume $S_{cur} = S_{new}$. If the value of the evaluation function for the new rules is less than the best rules S_{best} so far, we replace S_{best} with S_{new}. If the EFACT value in the new rule set is greater than the current set, then the proposed procedure will accept the new rule set.

A randomized number is accepted in the range 0 and 1. If the random number is less than the value witch given in our formula, then the new rule set is accepted [7].

4.4 Repeating deceleration and ending the algorithm

In each acceleration, the proposed procedure loops for a specified number. The deceleration parameter is used to update acceleration. Obviously, acceleration should decrease slowly until it ends with zero so that the algorithm can fully explore the state space for each rule set. When acceleration reaches its final value, the algorithm ends.

In computer simulations: K = 80, α = 0.8, fit = 0.01.

5 SIMULATION RESULT

use the values of the UCI machine training sets to test our procedure. Table I shows important features of this set.

T ABEL I .BASIC FEATURES OF DATASETS

Name	i	Number of Attribute	Number of Class
WINE	178	13	3
IRIS	150	4	3
CREDIT APPROVAL	690	15	2
LABOR	57	16	2
PIMA	768	8	2

In computer simulation for all pages of the training set, values between 0 and 1 are obtained and calculated using the following iteration:

$$\mathbf{X}_N = \frac{X - X_{minimum}}{X_{maximum} - X_{minimum}} \qquad (12)$$

So it seems each data set as a pattern classification problem in a N- dimensional space with values in $[0, 1]^n$, where n is the number of attributes. We use the 10- CV technique for each training set using different samples of the training set for ten times. In the 10- CV technique, the training set is distributed into ten subsets with the same size. 9 subsets are used as training patterns and another subset is used as the test pattern.

TABEL II. PARAMETERS SPECIFICATION IN COMPUTER SIMULATIONS FOR PROPSED APPROACH

Parameter	Value
Initial accuracy(a_0)	100
Final accuracy(a_f)	0
Accuracy rate(α)	0.8
#iteration at each gravity proc (K)	80
Population size (N_{pop})	100
probability of Crossover(P_c)	0.9
Probability of Mutation(P_m)	0.1
Percentage of Replacement(P_{rep})	20

Table II shows the specifications of the parameters used in our computer simulation. The classification performance of our proposed procedure is measured and compared with that of well-known algorithms such as XCS, SVM, K-NN, Naïve baye, and C4.5 [12]. In KNN, the k parameter is initialized to 3. XCS is the most famous Michigan system from GBML. It uses the tournament selection method instead of the conventional method based on fitness. The used NB version estimates the real values for attributes using a kernel density estimator. Table 3 shows the results for different algorithms on the wine set.

TABEL III. RESULAT OBTAINED FORM WINE DATA SET

Method	Accuracy percent of training set	Accuracy percent of experimental set
GAssist	98.66	96.33
C4.5	98.86	92.24
K-NN	96.66	96.61
NB	98.67	97.20
LIBSVM	99.33	98.10
XCS(Michigan)	99.97	95.60
Proposed(GSG)	98.20	95.81

According to the table III, we see that the accuracy obtained on the training set in the proposed method is higher than K-NN and less than other methods. About the test set, the proposed method has a higher accuracy than other methods and is ranked after GA, K-NN, LIBSVM, and NB.

TABEL IV. RESULAT OBTAINED FORM BSWD DATA SET

Method	Accuracy percent of training set	Accuracy percent of experimental set
GAssist	92.14	89.62
C4.5	89.93	77.66
K-NN	90.52	86.09
NB	91.92	91.43
LIBSVM	91.01	90.90
XCS(Michigan)	95.19	81.1
Proposed(GSG)	93.41	90.69

Table IV shows the results for the mentioned algorithms on the labor training sets. According to the table, for the training data set, the proposed method is above the C4.5 method and below other methods, but in the classification of test sets, it is above the C4.5 and XCS methods and below other methods.

TABEL V. RESULAT OBTAINED FORM PIMA DATA SET

Method	Accuracy percent of training set	Accuracy percent of experimental set
GAssist	83.11	74.46
C4.5	84.43	75.44
K-NN	85.67	74.52
NB	77.07	75.30
LIBSVM	78.27	77.32
XCS(Michigan)	98.90	72.40
Proposed(GSG)	84.89	75.16

Table V shows the accuracy of algorithms on the Pima training set. The accuracy of the GSG method on the training set is ranked after the XCS method. For the test set, the accuracy of the GSG method is ranked after the SVM, K-NN, C4.5 and NB methods.

TABEL VI. RESULAT OBTAINED FORM IRIS DATA SET

method	Accuracy percent of training set	Accuracy percent of experimental set
GAssist	99.47	94.49
C4.5	98.00	94.22
K-NN	96.59	94.89
NB	96.67	96.22
LIBSVM	97.11	96.22
XCS(Michigan)	99.10	94.70
Proposed(GSG)	99.71	96.66

Table VI shows the results for algorithms on the NS training set. The results obtained for the proposed method in the training is ranked after the GA and XCS methods, and for the test set, it is the lowest between all methods.

ISBN: 978-0-9891305-5-4 ©2014 SDIWC

TABEL VII. RESULAT OBTAINED FORM ION DATA SET

Method	Accuracy percent of training set	Accuracy percent of experimental set
GAssist	98.49	90.43
C4.5	98.68	88.97
K-NN	90.94	85.66
NB	93.00	91.50
LIBSVM	94.19	92.14
XCS(Michigan)	99.86	90.10
Proposed(GSG)	99.66	91.02

As shown in Table VII, the results from the training data set for the proposed method is above C4.5 and below other methods, but in the classification of test sets, it is above C4.5 and XCS and below other methods.

TABEL VIII. RESUALT OBTINED FROM CERDIT APPROVALL DATA SET

method	Accuracy percent of training set	Accuracy percent of experimental set
GAssist	90.61	85.18
C4.5	90.31	85.55
K-NN	91.05	84.73
NB	82.58	81.07
LIBSVM	55.52	55.51
XCS(Michigan)	98.90	85.60
Proposed(GSG)	90.61	85.18

According to table VIII, we see that the training set accuracy of the GSG method is ranked after the similar Michigan method which is based on GA. The test set accuracy of our method based on GSG is below the XCS, C4.5, k-NN and GA methods, and above other methods.

As experimental results show, our proposed procedure generates good and comparable results in terms of accuracy for the training set compared with several famous classification algorithms. Because of accurate initialization, the use of genetic operators and proper evaluation functions, our proposed method searches the state space efficiently and attempts to push us from local optimal solutions to general optimal solutions.

6- CONCLUSION

In this paper, we introduce and propose a fuzzy-genetic-gravitational classification method. The basis of the procedure efficiency is combination of genetic and gravitational algorithms with fuzzy if-then rules witch is properly used for producing estimation samples and exploring the state space classification. In computer application, the proposed method was applied to Wine, Iris, Waveform, Balance Scale WD, Credit Approval, JH Univ. Ionosphere, Labor Negotiation, and Pima data sets. The results indicated that the accuracy of our proposed method in classification of the training data sets is higher than the accuracy obtained from the k-NN, C4.5, NB, LIBSVM methods and other genetic based algorithms such as GAssist and is equal to the accuracy resulted from the GA-based XCS method which follows the patterns of the Michigan method. In addition, the results suggest that the accuracy obtained for the test sets using the proposed method is higher than the accuracy resulted from the methods which are based on GA, k-NN, C4.5, and LIB algorithms and sometimes it is the best.

7- REFERENCES

1. Krishnapuram ,R .IBM India Res. Lab., Indian Inst. Of Technol., New Delhi, India; Joshi, A. Nasraoui, O Yi, L., "low-Complexity Fuzzy Relational Clustering Algorithms For Web Mining ," Published in: Fuzzy System . Ieee Transactions on (Volume:9,Issue:4),Date of Publication Aug2001 Page-595-607 ,Issn:1063-6706 INSPEC Accession Number :7022482 ,Digital Object Identifier 10.1109/91.940971.

2. Tzung-Pei Hong ,Kuei-Ying Lin, Fuzzy Data Mining For Interesting Generalized Association rules."Fuzzy Sets And system Volume 138.Issue 2,1 September 2003,Page 255-269.

3. Rubing Duan ; Inst. of Comput. Sci., Innsberuck Univ.; Prodan ,R.; Fahringer .t., "Short Paper: Data Mining-based Fault Prediction An Grid," Published in: high Performance Distributed Comuting,2006 15th Ieee International Symposium on Date of Conference :page305-308 ISSn:1082-8907.Print

ISBN:1-4244-0307-0307-3 .INSPEC Accession Number:9018044,Paris,2006.

4. Hisao Ishibuchi ,Takashi Yamamoto ,"Fuzzy Rule Selection by Multi-Objective Genetic Local Search Algorithms And Rule Evaluation Measures in Data Mining," Department Of Industrial Engineering ,Osaka Prefecture university Gakun-cho,Sakai,Osaka599-8531,Japan.

5. Tparpinelli, R.S. : Coordenacao De Pos-Graduacao Em Engenria Electerica E Informatica Ind., Centro FED.de Educacao Tecnologia Do Parana , Curitiba ,brazil ; Lopes ,H.s, Freitas, A.A, "Data Mining With An Ant Colony Optimization Data mining With An Ant Colony Optimization Algorithm, Published in : Evolutionary Computation, IEEE Transactions on (Volume:6,Issue:4) Date Of Publication :Aug 2002 Pages:321-322 Issn:1089-778X, INSPEC Accession Number:7376421.

6. Published In :Evolutionary Computation ,IEEE Transactions on (Volume:6,Issue:4),Date of Publication Aug2002 Page-321-332 ,Issn:1089-778X INSPEC Accession Number :7376421,"Fuzzy Clustering Methods In data mining :A Comparative case Analysis," Ublished In: Advanced Computer Theory And Engineering,2008 ICACTE 08.International Conference On Date Of Conference 20-22 DEC.2008 Pages:489-493.

7. Chun-Hao Chen1, Tzung-Pei Hong2;3; Vincent s.Tseng2 , Lien-Chin Chen, "MULTI-OBJECTIVE GENETIC-FUZZY DATA MINING," International Journal of Innovative Computing, Information and Control ,Volume 8, Number 10(A), October 2012 pp. 6551{6568}

8. Esmat Rashedi, Hossein Nezamabadi-pour *, Saeid Saryazdi, GSA: A Gravitational Search Algorithm.", Information Sciences 179 (2009) 2232–2248 .

9. J. Silk, Duman, S. ; Dept. of Electr. Educ., Duzce Univ., Duzce, Turkey ; Sonmez, Y. ; Guvenc, U. ; Yorukeren, N. "Application of gravitational search algorithm for optimal reactive power dispatch problem ublished in: Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on Date of Conference:15-18 June 2011 Page(s):519 - 523 Print ISBN:978-1-61284-919-5 INSPEC Accession Number:12109140Conference Location :Istanbul Digital Object Identifier :10.1109/INISTA.2011.5946133 .

.

10. Tzung -Pei Hong, Chun-Hao Chen, Yeong-Chyi Lee, and Yu-Lung Wu, "Genetic-Fuzzy Data Mining With Divide-and-Conquer Strategy, " IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 12, NO. 2, APRIL 2008

11. UCI Machine Learning Repository : http://www.ics.uci.edu/mlearn /Databases.