

Mining association rules in big data with NGEF

Yunliang Chen · Fangyuan Li · Junqing Fan

Received: 28 September 2014 / Revised: 6 December 2014 / Accepted: 28 December 2014 / Published online: 9 January 2015
© Springer Science+Business Media New York 2015

Abstract Analyses and applications of big data require special technologies to efficiently process large number of data. Mining association rules focus on obtaining relations between data. When mining association rules in big data, conventional methods encounter severe problems incurred by the tremendous cost of computing and inefficiency to achieve the goal. This study proposes an evolutionary algorithm to address these problems, namely Niche-Aided Gene Expression Programming (NGEP). The NGEF algorithm (1) divides individuals to several niches to evolve separately and fuses selected niches according to the similarities of the best individuals to ensure the dispersibility of chromosomes, and (2) adjusts the fitness function to adapt to the needs of the underlying applications. A number of experiments have been performed to compare NGEF with the FP-Growth and Apriori algorithms to evaluate the NGEF's performance in mining association rules with a dataset of measurement for environment pressure (Iris dataset) and an Artificial Simulation Database (ASD). Experimental results indicate that NGEF can efficiently achieve more association rules (36 vs. 33 vs. 25 in Iris dataset experiments and 57 vs. 44 vs. 44 in ASD experiments) with a higher accuracy rate (74.8 vs. 53.2 vs. 50.6 % in Iris dataset experiments and 95.8 vs. 77.4 vs. 80.3 % in ASD experiments) and the time of computing is also much less than the other two methods.

Keywords Big data · Association rules · Gene expression programming · Niche

Y. Chen (✉) · F. Li · J. Fan
School of Computer Science, China University of Geosciences
(Wuhan), Wuhan, Hubei, China
e-mail: cyl_king@hotmail.com

J. Fan
e-mail: fanjunqing@sina.com

1 Introduction

Big data is an all-encompassing term for any collection of data sets which are very large and complex [1]. The trend to larger data sets is due to the additional information derivable from analysis of a large set of related data, allowing correlations to be found to “spot business trends, prevent diseases, combat crime and so on [2–5].

Data obtained by a variety of complex data system is often a series of isolated data [6,7]. Big data system is constructed by a number of inter-related data [8], therefore, correlations among big data should be observed for further analysis and applications. Correlation analysis should be certainly an important foundation for big data science [9,10].

Association rules mining have been extensively studied relations between variables in large databases [4,11]. It is intended to identify strong rules discovered in databases using different measures of interestingness [12]. Based on the concept of strong rules, Rakesh Agrawal et al. introduced association rules [13] for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection, Continuous production, and bioinformatics [14]. In contrast with sequence mining [15], association rule learning typically does not consider the order of items either within a transaction or across transactions.

Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Association rule generation is usually split up into two separate steps: First, minimum

support is applied to find all frequent item sets in a database. Second, these frequent item sets and the minimum confidence constraint are used to form rules.

Some well known algorithms are Apriori, Eclat and FP-Growth. Apriori [16–18] is the best-known algorithm to mine association rules. It uses a breadth-first search strategy to count the support of item sets and uses a candidate generation function which exploits the downward closure property of support. Eclat [19] (alt. ECLAT, stands for Equivalence Class Transformation) is a depth-first search algorithm using set intersection. FP stands for frequent pattern, in the first pass, the FP-Growth [20] algorithm counts occurrence of items (attribute-value pairs) in the dataset, and stores them to “header table”. There are also others algorithms based on constraints such as Context Based Association Rule Mining Algorithm and Node-set-based algorithms. CBPNARM [21] is the newly developed algorithm developed in 2013 to mine association rules on the basis of context. It uses context variable on the basis of which the support of an item set is changed and the rules are finally populated to the rule set. FIN [22], PrePost [23] and PPV [24] are three algorithms based on node sets. They use nodes in a coding FP-tree to represent item sets, and employ a depth-first search strategy to discovery frequent item sets using “intersection” of node sets. Another type of methods applied for association rules mining are based on evolution algorithms, for instance, Genetic Algorithm (GA) was applied to explore the rules in a certain database and was shown its advantages [25].

Big data requires exceptional technologies to efficiently cope with large number of data within tolerable elapsed times [5, 26, 27]. Although these methods have achieved a great deal in obtaining association rules, there are some drawbacks appearing with the increasing of the data [20–23]. The first one is the computing cost. While these methods are applied in a very large dataset, it needs to search databases much more times to form frequent item sets. The second one is the efficiency of the exploring of the rules. With the data growing up, some rules will be missed and the accuracy of the results obtained will be decreased. The motivation of this research is to explore an association rules mining algorithm which can address the defects mentioned above in big data.

Gene Expression Programming (GEP) is a salient approach in creating computer programs denoting the learned models and/or discovered knowledge [28, 29]. It differs from these evolutionary approaches mainly in chromosome encoding. GEP encodes individuals as chromosomes and implement them as linear strings with fixed lengths [30]. GEP algorithm begins with randomly generating linear fixed chromosomes for individuals within the initial population. Each individual is judged by a fitness function for each evolution generation. The individuals are then reserved by fitness values to reproduce the modification. The new individuals are subjected to the same process. The evolution process will

continue till it reaches a pre-specified number of generations or a solution is found.

In GEP, keeping the diversity of chromosome plays an important role in the evolution process. A novel algorithm, Niche-Aided Gene Expression Programming (referred to as NGEPE) is introduced in this study. Niche technology [31] performs efficiently while solving multi-objective optimization problems. According to the properties of Niche technology, NGEPE will divide the entire population into several niches and apply the evolution operators on each niche separately which would enhance the diversity of the population, and also extend the scope of search.

A dataset of measurement for environment pressure (MEP) and An Artificial Simulation Database (ASD) have been tested to check whether a NGEPE algorithm is capable of solving association rule mining problems. In experiments on the Iris data set from UCI Machine Learning Repository, an optimal parameter setting has been giving and a performance comparison has been made between NGEPE, FP-Growth and Apriori. In the ASD test, three methods are evaluated for association rule mining problems with the data increasing in a large datasets. Results show the proposed algorithm is efficient in addressing these problems.

The remainder of this paper is organized as follows: Sect. 2 recaps some relevant concepts about the association rules and a brief introduction on GEP. Section 3 presents the NGEPE algorithm for association rule mining problems. In Sect. 4, we present the experiments and results of using NGEPE to dealing with association rules problems. We concluded the paper and present the future work in Sect. 5.

2 Fundamentals of association rules and GEP algorithm

Big data requires the relations of data from the large and complex dataset [32]. Association rules mining is applied to explore the interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used [18].

2.1 Association rules

Let D_m be a set of transactions, that is, $D_m = \{d_1, d_2, \dots, d_m\}$; all the transactions are comprised of an item set A_n , (Item set A_n is a set of abstract symbols, $A_n = \{a_1, a_2, \dots, a_n\}$). The association rule over D_m is an expression of the form: $X \Rightarrow Y$, $X \in A_n$, $Y \in A_n$, and $X \cap Y = \phi$. Each association rule has two main attributes: support ($spt(X \Rightarrow Y, D_m)$) and confidence ($cnf(X \Rightarrow Y, D_m)$) to reflect its importance to D_m . An association rule is strong if its support and confidence are not less than minimum support value (\min_spt) and minimum confidence value (\min_cnf) respectively.

Table 1 A supermarket database

Transaction ID	Bread	Beer	Butter	Milk
1	1	1	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	1	1
5	1	0	0	0

To illustrate Association rules, Table 1 shows a small example from the supermarket domain. The set of items is $A_n = \{bread, beer, butter, milk\}$ and a small database containing the items (1 codes presence and 0 absence of an item in a transaction) has 5 transactions. An association rule for the supermarket could be $\{butter, bread\} \Rightarrow \{milk\}$ meaning that if butter and bread are bought, customers also buy milk.

In big data applications, a rule needs a support of thousands or millions transactions before it can be considered statistically significant, and datasets often contain billions of transactions.

2.2 GEP algorithm

Gene Expression Programming (GEP) is a powerful evolutionary method which overcome the common drawbacks of GA and GP [28]. Similar to GA and GP, GEP follows the Darwinian principle of the survival of the fittest and uses populations of candidate solutions to a given problem in order to evolve new ones [33]. The difference amongst GEP, GA and GP is the way in which individuals of a population of solutions are represented [31]. Although GEP has a simple and linear form, it is the most flexible and powerful method in solving complex problems.

In GEP, an individual (chromosome) is represented by a genotype, constituted by one or more genes. A chromosome is a linear and compact entity, which can be easily manipulated with genetic operators such as mutation, crossover, and transposition.

When using GEP to solve a problem, there are five components should be specified: the function set, the terminal set, the fitness function, GEP control parameters, and the stop condition:

Generation of the initial population of solutions is the first step. This can be done by using a random process. The individuals are then expressed as expression trees (ETs, an example is given in Fig. 1), which can be evaluated according to a fitness function that determines how good a solution is in the problem domain. According to the value of each chromosome evaluated by the fitness function, the operator on the selected chromosomes will be applied such as crossover, mutation and rotation. If a solution of satisfactory quality is

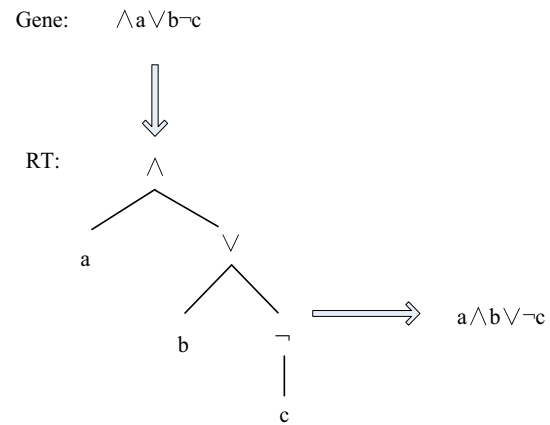


Fig. 1 chromosome (gene) and its RTs

found, or a predetermined number of generations are reached, the evolution stops and the “best-so-far” solution is returned.

2.3 Chromosome encoding for association rules

In GEP, chromosomes can be expressed as ETs, which can be also mapped into Rule Trees (RTs) [34]. Each chromosome is a character string in fixed-length, which can be composed of any element from the function set or the terminal set. Each gene has a head and a tail. The size of the head (h) is defined by the user, but the size of the tail (t) is obtained as a function of h and a parameter n (the number of elements of the function set). The tail size can be calculated by the following equation:

$$t = h * (n - 1) + 1 \tag{1}$$

In Table 1, RTs can be composed of any element from the function set or the terminal set. The former consists of 3 logical symbols (that is, “ \wedge for and”, “ \vee for or” and “ \neg for not”), while the latter, comes from the items attribute set $A_n = \{a(bread), b(beer), c(butter), d(milk)\}$.

The RT shown in Fig. 1 corresponds to a sample chromosome, and can be interpreted in an association rule. The chromosome is constructed by six elements, i.e., “ $\wedge a \vee b \neg c$ ”.

The initial population is composed of several individuals that related to all kinds of RTs generated in real-world applications. After being applied with generation operators, the individuals will survive or die according to their fitness values.

GEP uses genetic operators, i.e., mutation, transposition, and crossover, to create variations for evolution. A mutation operator introduces a random change into symbols at any position in a chromosome [28]. A transposition operator transports the sequence elements of gene to another place. The crossover operator chooses and pairs two chromosomes to exchange some elements between them [35].

How to efficiently create variation depends on the nature of the complex problem under investigation.

Generally, after applying genetic operators to create variation in each generation, GEP selects some individuals and copies those into the next generation based on their fitness, such as simple elitism [36] and cloning of the best individual. Typically, roulette-wheel method [28] is used in many GA [37] and GP algorithms [38].

References [30,39] take the following formula to evaluate a chromosome:

$$fitness(X \Rightarrow Y) = \partial_s \times (spt(X \Rightarrow Y, D_m) - \min_spt) + \partial_c \times (cnf(X \Rightarrow Y, D_m) - \min_cfd) \tag{2}$$

where ∂_s, ∂_c are the castigation coefficients of support and confidence respectively; however, the above fitness function may easily lead to “false rules”, that is:

$$spt(X \Rightarrow Y, D_m) - \min_spt < 0 \text{ and } cnf(X \Rightarrow Y, D_m) - \min_cfd \geq spt(X \Rightarrow Y, D_m) - \min_spt \tag{3}$$

or

$$cnf(X \Rightarrow Y, D_m) - \min_cfd < 0 \text{ and } spt(X \Rightarrow Y, D_m) - \min_spt \geq cnf(X \Rightarrow Y, D_m) - \min_cfd \tag{4}$$

If either of above condition is satisfied, we can obtain that $fitness(X \Rightarrow Y) \geq 0$. Nevertheless, such kind of individual cannot satisfy the strong rule. To address this problem, a modified fitness function is proposed here. First of all, we define an operation “Or-multiple, \otimes ”:

Definition 1 Suppose two real numbers $x, y \in [-1, 1]$. Both of them can be denoted as $x = a \times 10^m, y = b \times 10^n$, where $a, b \notin (-1, 1), m, n \in (-\infty, 0]$. Then we have:

$$x \otimes y = \begin{cases} a + b > 0, & \text{if } a > 0, b > 0 \\ 0 & \text{else} \end{cases} \tag{5}$$

Then a new fitness function ($fitness^*(X \Rightarrow Y)$) which improves the formulary 2 can be obtained as follows:

$$fitness^*(X \Rightarrow Y) = [\partial_s \times (spt(X \Rightarrow Y, D_m) - \min_spt)] \otimes [\partial_c \times (cnf(X \Rightarrow Y, D_m) - \min_cfd)] \tag{6}$$

3 NGEF for association rules mining

The fundamental of niche technology is to divide the entire population into several niches, which have the same size under the initial condition; however, as the evolutionary process goes forward, the size will adjust adaptively in a

dynamic way according to the mean fitness of each niche. Moreover, each niche may control its population by setting the maximum size MAX and minimum size MIN. And various kinds of genetic operations towards chromosome are only performed in each niche itself. Such technology enhances the diversity of the population to a large extent, and also makes the scope of search wider and more extensive [30,40].

3.1 Flow of NGEF algorithm

A NGEF process can be separated into several parts including: (1) population initialization; (2) genetic operation, selection and reserving; (3) revealing the global solution. The NGEF algorithm flow can be illustrated in Fig. 2:

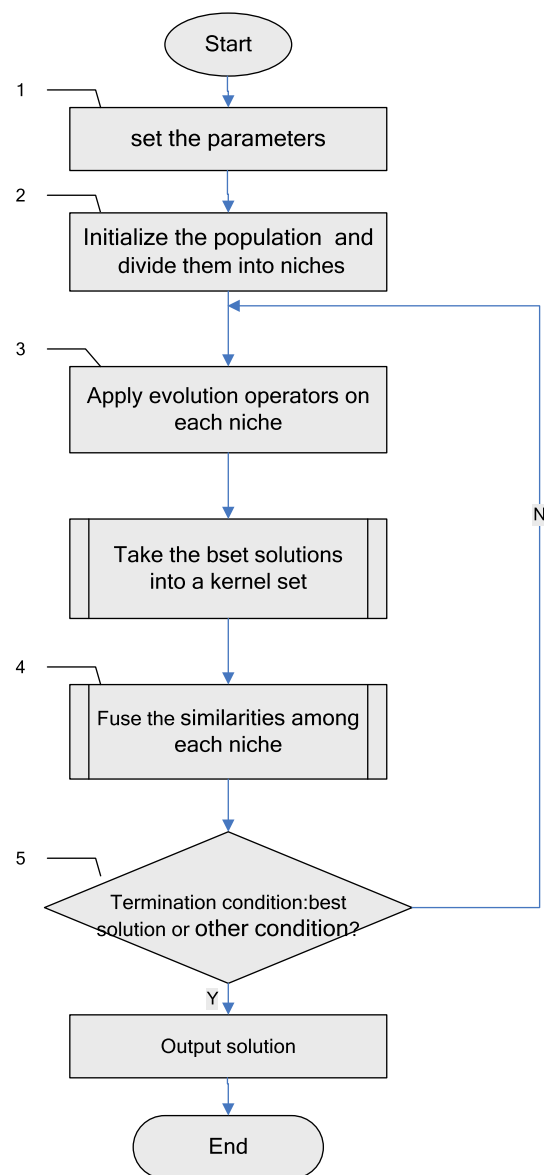


Fig. 2 NGEF algorithm flow

Step 1 Set the parameters: chromosome's head length, the operation rate of each operator, the function set and the terminal set;

Step 2 Initialize the population P, set the number of niches as well as MAX and MIN, and then divides the chromosomes into these niches equally;

Step 3 Use evolution operators on each niche (see Algorithm 2), and take the solution set($OP(P(t+1), \infty)$) into a kernel set of niches as the computation result in this generation;

Step 4 Find the similarities among the best individuals from each niche; fusion will be operated according to Algorithm 1;

Step 5 If the global optimal solution is found or the preset maximum number of generations is reached, end the evolution process. Otherwise, go to Step 3.

3.2 Niche fusion

Considering the isomorphism of best individuals (that is, Hamming Distance [30] between them is less than a threshold; in this paper we define these two individuals as isomorphic chromosomes) may be generated in some niches during the evolution, the fusion-algorithm is required to be done on the niches so as to maintain the diversity of population and avoid the emergence of redundancy rules. Two niches can be fused by algorithm 1.

Algorithm 1 Niche Fusion

- ① Merge all individuals of two niches (which are supposed as nich₁ and nich₂, and before fusion, their sizes are s_1 and s_2 respectively) to be fused into nich₁; go to ②;
- ② Examine the similarity of nich₁ so as to exclude isomorphic chromosomes for fusion, and then obtain the size s'_1 of the modified nich₁; go to ③;
- ③ If s'_1 is bigger than MAX, redundant individuals will be selected out by Roulette Wheel; then adjust s'_1 and go to ④; else go to ⑤;
- ④ If s'_1 is smaller than MIN, the new individual will be introduced randomly until the smallest size is satisfied; then adjust s'_1 and go to ⑤;
- ⑤ Construct nich₂ randomly, and make the equation satisfy $s'_2 = s_1 + s_2 - s'_1$.

3.3 Niche evolution

The algorithm listed below is shown how to obtain the optimal rule set. Let $S_i, i = 1, 2, \dots, m$, (where m is the number of niches) be the next generation evolved from current generation $P(t)_i$, and then we have:

Algorithm 2 Niche Evolution

- ① Genetic operations such as crossover and mutation are executed in the internal of all niches $P(t)_i$ so as to obtain child generation S_i , and then merge the next generation S into the parent niche populations to gain $P(t+1)_i = P(t)_i \cup S_i$; go to ②;
- ② Select the top 10% individuals depending on the fitness from the set $P(t+1)_i$; and then make a Cartesian product over them to obtain the solution set($OP(P(t+1), \infty)$) which is regarded as the outputs of current niches; go to ③;
- ③ Apply Elitist Strategy to select a pre-given of best individuals from the child generation to substitute for worst ones from father generation in the same amount.

4 Tests and evaluation

A series of experiments has been performed to evaluate the performance of NGEF for association rules mining. We first assess the performance of NGEF algorithm through a database of measurement for environmental pressure. After that, we compared the performance of NGEF with FP-Growth and Apriori on an Artificial Simulation Database (ASD). All experiments were executed over a desktop computer with configurations: CPU (Intel Core i5-540M, 2.53 GHz); RAM (8 GB), Operating System (Windows 7 Professional).

The parameters presented in Table 2 were specified with empirical values as suggested in [28, 41, 42].

All experiments are repeated independently for 100 trials. Besides, we introduce a quantitative analytical function to evaluate the diversity of rule set. Population (rule set) diversity can be defined as follows:

$$DIV_{pop} = \frac{\sum_{m=1}^N \sum_{n=1}^N d_{m,n}}{N^2} \quad (7)$$

Where N is population size, and $d_{m,n}$ represents the Hamming distance between two certain Individuals m and n .

4.1 Problem of Iris data set

In this subsection, experiments for association rules mining on an Iris data set are carried out to test the effectiveness of NGEF Algorithm. The Iris data set from UCI Machine Learning Repository can be downloaded from "<http://www.sgi.com>". The data set contains 4 attribute items including "sepal-length", "sepal-width", "petal-length" and "petal-width" and owns 150 samples. We select the first 100 records as a training set, hoping to find out the association rules

Table 2 Parameters for NGEP

Parameter	Description	Parameter	Description
Function set	F = \wedge, \vee, \neg stands for three basic operations such as and (\wedge), or (\vee) and not (\neg) respectively		
Terminal set	Item attribute set (A_n)		
Population size	100	Constants set	Randomization
Head length	5	Crossover rate	0.33
Number of genes	3	Mutation rate	0.01
Linking function	\wedge	Selection	Roulette wheel
IS transposition rate	0.1	Generations	1,000
RIS transposition rate	0.1	Number of Niches	5
Gene transposition rate	0.1	Gene recombination rate	0.1
Maximum size of Niche	60	Minimum size of Niche	10

Table 3 Result of ASD

Algorithm	Time (s)	Number of rules	Accuracy rate (%)	Prediction rate (%)
NGEP	12.3	36	74.8	67.3
FP-Growth	18.6	33	53.2	51.1
Apriori	21.4	25	50.6	48.4

between the above items. The rest 50 records are used for testing. First of all, the data set is preprocessed; and each above attributes are giving by two values (i.e. “low, short” for “sepal-length” and “petal-length”; “wide, narrow” for “sepal-width” and “petal-width”). NGEP, Apriori and FP-Growth algorithms are applied to the experiments of association rules mining on the Iris data set.

Table 2 summarizes the experimental results, where Accuracy Rate refers to the average of the accuracy for each run of the experiments and Prediction Rate for Testing Set refers to the average of the prediction accuracy for each run of the experiments on the testing data set. As seen from Table 3, for the average values over the 100 runs, NGEP has consistently achieved a higher accuracy value and a low time cost, besides, NGEP can get more rules from the data set.

Figure 3 shows the curve of diversity value of NGEP. From generation 0 to generation 1,000, the value of NGEP is 22 or so. Even in the 700th generations the value of DIV of NGEP is still high. As the solution is undergoing, the diversity decreases in general (which is reasonable), but it still maintains relatively high, which makes it have the capability of obtaining new solutions and not easy to converge to the local optimization. Table 4 shows some rules obtained by NGEP.

4.2 Artificial simulation database (ASD)

An artificial simulation database was build to evaluate the performance of NGEP, FP-Growth and Apriori with the data

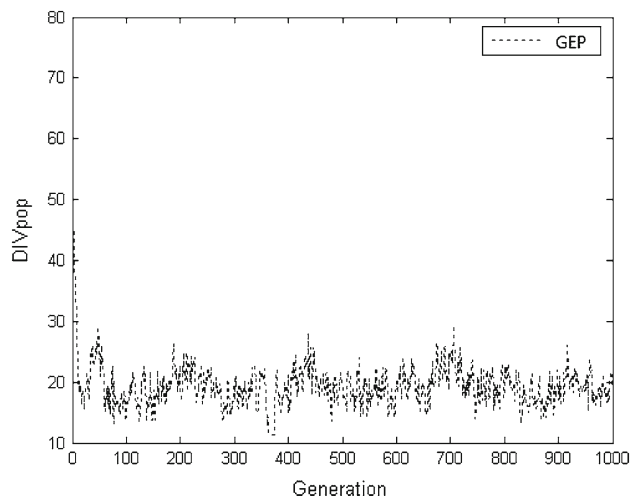


Fig. 3 Diversity of NGEP

Table 4 Rules of Iris by NGEP

Chromosome	Connected association rule
Rule1 $\wedge \vee \wedge \neg \text{heehfea}$ $e \neg \vee \vee \vee \text{faabbc}$ $\neg f \vee \wedge \wedge \text{bddfce}$	If petal-length is short, then Iris is Setosa
Rule2 $\vee \vee \text{cd} \neg \text{dadfbec}$ $\wedge \neg \neg \text{ggbefdgc}$ $\wedge \vee \text{hb} \neg \text{bcaefhh}$	If petal-width is wide, then Iris is Virginica
Rule3 $\wedge \wedge \wedge \text{g} \neg \text{gghahe}$ $\neg \neg \vee \neg \text{eebfcdg}$ $\wedge \text{aegeaacgbh}$	If sepal-length is short, petal-length is short and petal-width is narrow, then Iris is Setosa
Rule4 $\wedge \wedge \wedge \text{e} \neg \text{gafcdh}$ $\wedge \wedge \text{c} \neg \text{abedhga}$ $\neg \neg \vee \neg \text{ffbfhg}$	If sepal-width is narrow, petal-length is long and petal-width is narrow, then Iris is Virginica
Rule5 $\text{g} \wedge \vee \neg \text{bdcaecae}$ $\wedge \text{c} \vee \text{d} \neg \text{dfebae}$ $\wedge \neg \wedge \text{efebahcb}$	If sepal-length is short, sepal-width is narrow, petal-length is short and petal-width is narrow, then Iris is Setosa
Rule6 $\neg \wedge \wedge \neg \text{bhagcdd}$ $\wedge \wedge \wedge \text{bdddecab}$ $\wedge \neg \wedge \text{e} \neg \text{cfhehc}$	If sepal-length is long, sepal-width is wide, petal-length is long and petal-width is wide, then Iris is Virginica

increasing sharply. The database D consists of 10,000 transactions with the item attribute set A_n containing ten attributes $\{a_1, a_2, a_3, \dots, a_{10}\}$, among which $\{a_9, a_{10}\}$ exists as the disturbance attribute. Let $\min_spt = 0.1$ and $\min_cnf = 0.5$. The database is established by the three following rules:

$$(a_1 \wedge a_2) \vee (\neg a_3) \rightarrow a_6, (a_2 \vee a_3) \wedge a_4 \rightarrow a_7,$$

$$\text{And } (\neg a_3) \wedge (a_4 \vee a_5) \rightarrow a_8;$$

To evaluate the performance of three methods while dealing with big data problems, experiments were carried out separately with three number of transaction: 3,000, 6,000 and 10,000.

As shown in Fig. 4, NGEP performs better than FP-Growth and Apriori do in association rules mining problems. With the increasing number of transactions, the differences amongst NGEP, FP-Growth and Apriori increase significantly. NGEP maintains a rule number higher than 46 while that using FP-Growth gently increase from 33 to 44 and that using Apriori is from 35 to 44. The success rates also indicate that NGEP can ensure an accurate rules mining.

It also can be indicated that NGEP executes extremely faster than FP-Growth and Apriori do with the number of transactions increasing when achieving the above successes. When the transactions grows up from 3,000 to 1,000, the execute time of NGEP increase from 36.3 to 107.6 s, while that of FP-Growth is from 43.2 to 485.4 s and for Apriori is from 58.6 to 574.3 s.

5 Conclusions and future work

Analyses of big data request new methods to meet the demands of applications. To overcome the drawbacks of traditional methods, we examined the feasibility and effectiveness of a Niche-aided GEP approach to solving association rules mining problems in big data. The niche method combined with GEP is designed to explore association rules in large quantities of data set. The procedure of NGEP begins with the people initialization, and then divides individuals into each niche. Evolution operators are applied on each niche and then some of the sub-niches are fused according to the similarity of best individuals. After that, Cartesian product operation is applied in the kernel set of niches to generate better outputs.

A series of experiments have been carried out to have an in-depth investigation on the performance of the proposed NGEP. For the iris data set, ten sets of experiments have first been performed to assess the efficiency of NGEP. The results show that the number of rules obtained using NGEP can reach a high value and the execution time is less than the other methods. We then applied the NGEP algorithm on

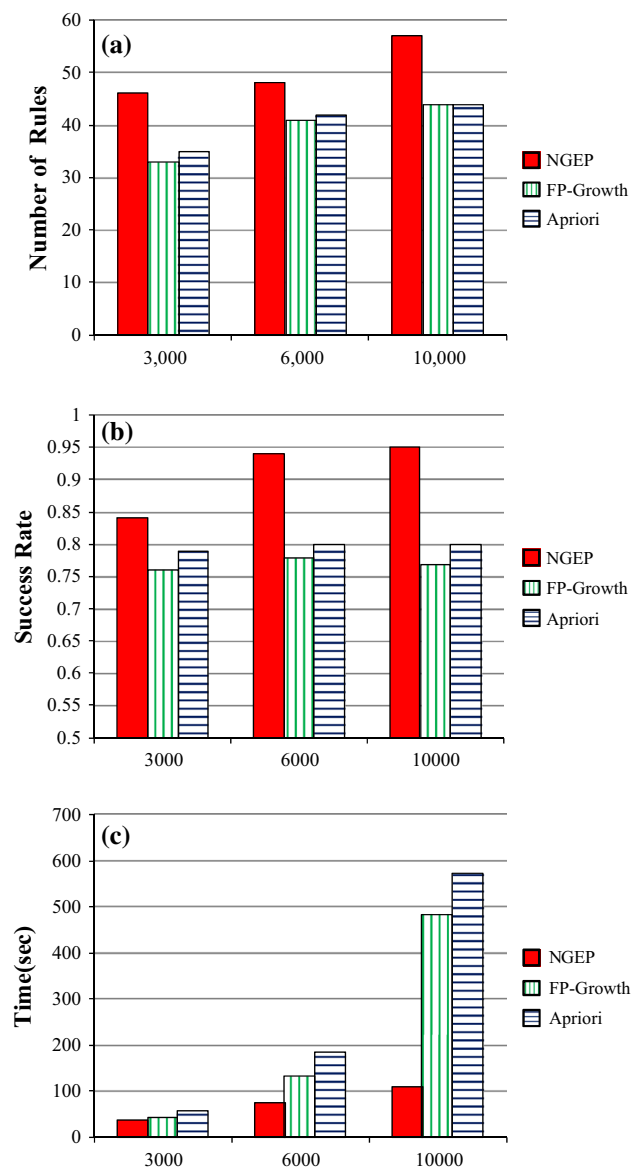


Fig. 4 Experimental results with NGEP, FP-Growth and Apriori. **a** Number of Rules: with the number of transactions increasing, rules obtained from NGEP increases from 46 to 48 and then to 57, while that of FP-Growth grows up from 33 to 41 and then to 44, and that of Apriori raises from 35 to 42 and then to 44. **b** Success rate: the success rate of NGEP remains higher than other two methods. While in number 3,000, the success rate of NGEP is 0.843, while that of FP-Growth is 0.763, and Apriori is 0.792. While in number 6,000, the results are (0.946, 0.788 and 0.806). While in number 10,000, the results are (0.958, 0.774 and 0.803). **c** Time: with the number = 3,000, the time results for NGEP, FP-Growth and Apriori are (36.3, 43.2 and 58.6 s). With the number = 6,000, the results are (74.2, 132.6 and 184.7 s). With the number = 10,000, the results are (107.6, 485.4 and 574.3 s)

ASD to evaluate the ability of association rules mining with the data increasing sharply. In comparison with FP-Growth and Apriori, the number of rules obtained using NGEP are always higher than the other methods. Especially the convergence speeds of NGEP are higher than those of FP-Growth

and Apriori (e.g., with transactions is 10,000, the execution time are 107.6 s (NGEP) vs. 485.4 s (FP-Growth) vs. 574.3 s (Apriori)).

For future work, we will consider other measures of interestingness for rules such as Collective strength and Conviction [12]. Another interesting work is to use NGEP on fuzzy association rules and the categorical attributes [43,44].

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (Nos. 61272314, 61361120098, 61440018), the China Postdoctoral Science Foundation (2014M552112), the Hubei Natural Science Foundation (No. 2014CF-B904).

References

- Lizhe, W., Ke, L., Peng, L., et al.: IK-SVD: dictionary learning for spatial big data via incremental atom update. *Comput. Sci. Eng.* **16**(4), 41–52 (2014)
- Barnes, J.: Data, data, everywhere. *ITS Int.* **20**(1), 44–49 (2014)
- Deng, Z., Wu, X., Wang, L., et al.: Parallel processing of dynamic continuous queries over streaming data flows. *IEEE Trans. Parallel Distrib. Syst.* (2014). doi:10.1109/TPDS.2014.2311811
- Chen, D., Wang, L., Wu, X., et al.: Hybrid modeling and simulation of huge crowd over a hierarchical grid architecture. *Future Gener. Comput. Syst.* **29**(5), 1309–1317 (2013)
- Chen, D., Wang, L., Zomaya, A., et al.: Parallel simulation of complex evacuation scenarios with adaptive agent models. *IEEE Trans. Parallel Distrib. Syst.* (2014). doi:10.1109/TPDS.2014.2311805
- Xue, W., Yang, C., Fu, H. et al.: Enabling and scaling a global shallow-water atmospheric model on Tianhe-2. In: *Proceedings of the 28th International Parallel and Distributed Processing Symposium* (2014). IEEE
- Zhao, J., Wang, L., Tao, J., et al.: A security framework in G-Hadoop for big data computing across distributed cloud data centres. *J. Comput. Syst. Sci.* **80**(5), 994–1007 (2014)
- Chen, D., Turner, S.J., Cai, W., et al.: Synchronization in federation community networks. *J. Parallel Distrib. Comput.* **70**(2), 144–159 (2010)
- Ma, Y., Wang, L., Liu, D., et al.: Distributed data structure templates for data-intensive remote sensing applications. *Concurr. Comput. Prac. Exper.* **25**(12), 1784–1797 (2013)
- Ma, Y., Wang, L., Zomaya, A., et al.: Task-tree based large-scale Mosaicking for remote sensed imageries with dynamic DAG scheduling. *IEEE Trans. Parallel Distrib. Syst.* **25**(8), 2126–2137 (2013)
- Wang, L., von Laszewski, G., Younge, A., et al.: Cloud computing: a perspective study. *New Gener. Comput.* **28**(2), 137–146 (2010)
- Piatetsky-Shapiro, G.: Discovery, analysis and presentation of strong rules. In: Piatetsky-Shapiro, G., Frawley, W.J. (eds.) *Knowledge Discovery in Databases*, pp. 229–248. AAAI Press (1991)
- Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: *Proceedings of the ACM SIGMOD Record* (1993)
- Li, L., Xue, W., Ranjan, R., et al.: A scalable Helmholtz solver in GRAPES over large-scale multicore cluster. *Concurr. Comput. Prac. Exper.* **25**(12), 1722–1737 (2013)
- Chen, D., Li, X., Cui, D., Wang, L., Lu, D.: Global synchronization measurement of multivariate neural signals with massively parallel nonlinear interdependence analysis. *IEEE Trans. Neural Syst. Rehabil. Eng.* **22**(1), 33–43 (2014)
- Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB* (1994)
- Duru, N.: An application of apriori algorithm on a diabetic database. In *Knowledge-Based Intelligent Information and Engineering Systems*, pp. 398–404. Springer, Berlin (2005)
- Aflori, C., Craus, M.: Grid implementation of the Apriori algorithm. *Adv. Eng. Softw.* **38**(5), 295–300 (2007)
- Zaki, M.J.: Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.* **12**(3), 372–390 (2000)
- Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Burlington (2005)
- Shaheen, M., Shahbaz, M., Guergachi, A.: Context based positive and negative spatio-temporal association rule mining. *Knowledge-Based Syst.* **37**, 261–273 (2013)
- Deng, Z.-H., Lv, S.-L.: Fast mining frequent itemsets using Node-sets. *Exper. Syst. Appl.* **41**(10), 4505–4512 (2014)
- Deng, Z., Wang, Z., Jiang, J.: A new algorithm for fast mining frequent itemsets using N-lists. *Sci. China Inform. Sci.* **55**(9), 2008–2030 (2012)
- Deng, Z., Wang, Z.: A new fast vertical method for mining frequent patterns. *Int. J. Comput. Intell. Syst.* **3**(6), 733–744 (2010)
- Romão, W., Freitas, A.A., Gimenes, I.M.D.S.: Discovering interesting knowledge from a science and technology database with a genetic algorithm. *Appl. Soft Comput.* **4**(2), 121–137 (2004)
- Kołodziej, J., González-Vélez, H., Wang, L.: Advances in data-intensive modelling and simulation. *Future Gener. Comput. Syst.* **37**, 282–283 (2014)
- Chen, D., Li, D., Xiong, M., et al.: GPGPU-aided ensemble empirical-mode decomposition for EEG analysis during anesthesia. *IEEE Trans. Inform. Technol. Biomed.* **14**(6), 1417–1427 (2010)
- Ferreira, C.: Gene expression programming: a new adaptive algorithm for solving problems. *arXiv:cs/0102027* (2001)
- Chen, Y., Chen, D., Khan, S.U., et al.: Solving symbolic regression problems with uniform design-aided gene expression programming. *J. Supercomput.* **66**(3), 1553–1575 (2013)
- Wei, W., Wang, Q., Wang, H., et al.: The feature extraction of nonparametric curves based on niche genetic algorithms and multi-population competition. *Pattern Recognit. Lett.* **26**(10), 1483–1497 (2005)
- Ferreira, C.: Mutation, transposition, and recombination: an analysis of the evolutionary dynamics. In: *4th International Workshop on Frontiers in Evolutionary Algorithms* (2002)
- Wang, L., Chen, D., Hu, Y., et al.: Towards enabling cyberinfrastructure as a service in clouds. *Comput. Electr. Eng.* **39**(1), 3–14 (2013)
- Freitas, A.A.: A survey of evolutionary algorithms for data mining and knowledge discovery. In *Advances in Evolutionary Computing*, pp. 819–845. Springer, Berlin (2003)
- Noda, E., Freitas, A.A., Lopes, H.S.: Discovering interesting prediction rules with a genetic algorithm. In: *Proceedings of the 1999 Congress on Evolutionary Computation* (1999)
- Lopes, H.S., Weinert, W.R.: EGIPSYS: an enhanced gene expression programming approach for symbolic regression problems. *Int. J. Appl. Math. Comput. Sci.* **14**(3), 375–384 (2004)
- Ferreira, C.: Function finding and the creation of numerical constants in gene expression programming. In *Advances in Soft Computing: Engineering Design and Manufacturing*, p. 265 (2003)
- Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-wesley, Boston (1989)
- Koza, J.R.: *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge (1994)
- Zhang, J., Huang, D.-S., Lok, T.-M., et al.: A novel adaptive sequential niche technique for multimodal function optimization. *Neurocomputing* **69**(16), 2396–2401 (2006)

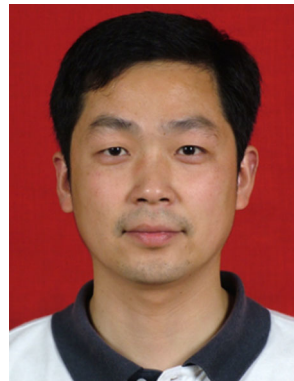
40. Ferreira, C.: Genetic representation and genetic neutrality in gene expression programming. *Adv. Complex Syst.* **5**(04), 389–408 (2002)
41. Siwei, J., Zhihua, C., Dang, Z.: Parallel gene expression programming algorithm based on simulated annealing method. *ACTA Electr. Sinica* **33**, 2017–2021 (2005)
42. Zuo, J., Tang, C., Zhang, T.: Mining predicate association rule by gene expression programming. In *Advances in Web-Age Information Management*, pp. 281–294. Springer, Berlin (2002)
43. Kuok, C.M., Fu, A., Wong, M.H.: Mining fuzzy association rules in databases. *ACM Sigmod Rec.* **27**(1), 41–46 (1998)
44. Chen, D., Li, X., Wang, L., Khan, S., Wang, J., Zeng, K., Cai, C.: Fast and scalable multi-way analysis of massive neural data. *IEEE Trans. Comput.* (2014). doi:[10.1109/TC.2013.2295806](https://doi.org/10.1109/TC.2013.2295806)



Fangyuan Li received the B.Sc. from China University of Geosciences. Now, she is a graduate student with the school of Computer Science, China University of Geosciences, Wuhan, China.



Yunliang Chen received the B.Sc. and M.Eng. degree from China University of Geosciences, and the Ph.D. degree from Huazhong University of Science and Technology, China. He is currently an Associate Professor with the School of Computer Science, China University of Geosciences, Wuhan, China.



Junqing Fan received the B.Sc. degree from Nanjing University of Science & Technology and M.Eng. degree from Huazhong University of Science and Technology, China, and the Ph.D. degree from China University of Geosciences. He is currently a Lecturer with the School of Computer Science, China University of Geosciences, Wuhan, China.