Data association rules mining method based on genetic optimization algorithm

Binghui Xu¹*, Sizhe Ding² and You Li³

¹ Taizhou Vocational & Technical College, Taizhou, Zhejiang, 318000, China
² School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei, 430072, China
³ Donald Bren School of Information Sciences, University of California, Irvine, Irvine, CA, 92697, USA

*Corresponding author’s e-mail: 0708chengji@163.com

Abstract. Traditional data mining methods usually need to visit the database repeatedly to determine the frequent item set, which makes the data server burden heavier and reduces the efficiency of data mining. To solve this problem, this paper combines the immune mechanism and genetic algorithm dynamically to improve the traditional genetic algorithm (GA), and proposes a data association rule mining method based on improved immune genetic algorithm (IIGA), which realize the effective analysis of big data. The experimental results show that the algorithm we proposed is better than immune genetic algorithm and Apriori algorithm in data mining time and association rules mining accuracy, which can be better applied to data analysis. The research results have positive reference significance for the field of data mining.

1. Introduction

The existing association rule algorithms in the field of data mining are mainly Apriori algorithm and optimized Apriori algorithm. This kind of method is based on the search algorithm to access the database effectively. By setting the support threshold, the frequent item set can be processed effectively, and the corresponding association rules can be obtained. Because of the need of continuous access operation in the process of determining frequent item set, it will greatly increase the running burden of data server and affect the efficiency of data mining. Therefore, it is urgent to introduce new methods to solve such problems [1-2].

GA is a practical intelligent algorithm. Traditional GA has some problems, which need to be improved. Some scholars have combined immune algorithm with GA to get immune genetic algorithm (IGA). Since IGA algorithm can effectively prevent the local optimal problems, ensure the diversity of population, and effectively improve the accuracy of convergence results, this paper introduces it into association rule calculation and improves it to a certain extent to achieve efficient and accurate association rule mining of data.

2. Algorithm optimization

2.1 IGA analysis

GA is essentially a strategy of natural selection.[4]. The algorithm can code and retrieve database information according to certain principles, but sometimes there are still problems of low accuracy and...
high redundancy. Therefore, some scholars have developed IGA by integrating memory function into GA. Compared with GA, IGA has the following outstanding characteristics: 1) Fitness function can be used as antigen, but also as constraint condition, and the objective function is solution algorithm. 2) The antibody is filed and the antibody group contains a group of antibodies. Like GA, IGA still contains binary and decimal systems. 3) The repulsion between antibodies represents their force and binding force. 4) The affinity of antigen antibody and antigen induction can obviously indicate the matching degree of both in IGA to a certain extent. 5) Memory unit is the antibody group in IGA, which can ensure the fusion speed, quality as well as population diversity. 6) Similar to biology, vaccines can prevent pathogens in advance by analyzing the pathogenic mechanism. IGA operation requires knowledge of evolution as well as estimation of the best individual gene.

2.2 Algorithm improvement

In GA, due to the selection of crossover and mutation operators, some excellent genes are lost prematurely. For the time being, the scope of search becomes narrower, which obviously making it difficult to search the optimal solution of global condition, and also inhibiting the search efficiency in the later stage of evolution. In practical applications, if the population is small, we will find that the iterative error is almost inevitable. This defect will lead to the result of GA deviate from the expectation. Some researchers have found that the adaptive GA can make up for the shortcomings of GA. But IGA also has the problems of low convergence efficiency and long calculation time. Therefore, by adjusting the probability of crossover and mutation adhere to the value of fitness function in IGA algorithm, we can keep the excellent individuals as much as possible and speed up the convergence speed, so that the algorithm can show more excellent calculation efficiency in the face of massive water resources data [5-6]. The probability of crossover and mutation of IGA algorithm are

\[
P_c = \begin{cases} 
(P_1 - P_2) \cdot (F - F_{avg})/(F_{max} - F_{avg}), & F \geq F_{avg} \\
(P_1, F \leq F_{avg}) 
\end{cases}
\]

\[
P_m = \begin{cases} 
(P_{m1} - P_{m2}) \cdot (F - F_{avg})/(F_{max} - F_{avg}), & F \geq F_{avg} \\
(P_{m1}, F \leq F_{avg}) 
\end{cases}
\]

In IGA, the probability of crossover and mutation are

\[
P_c = \begin{cases} 
(P_1(F_{avg} - F') + P_2(F' - F_{min}))/F_{avg} - F_{min}, & F \geq F_{avg} \\
(P_2(F_{avg} - F') + P_3(F' - F_{min}))/F_{max} - F_{avg}, & F \leq F_{avg} 
\end{cases}
\]

\[
P_m = \begin{cases} 
(P_{m1}(F_{avg} - F') + P_{m2}(F' - F_{min}))/F_{avg} - F_{min}, & F \geq F_{avg} \\
(P_{m2}(F_{avg} - F') + P_{m3}(F' - F_{min}))/F_{max} - F_{avg}, & F \leq F_{avg} 
\end{cases}
\]

\(F_{max}\) is the maximum individual adaptability of population, \(F_{avg}\) is the average individual adaptability of population, \(F_{min}\) is the minimum individual adaptability of population, \(F'\) is the maximum adaptability of population, \(F\) is the adaptability of variant individuals. The improved \(P_c\) and \(P_m\) are non-zero and change automatically with individual fitness, which will be beneficial to increase the genetic function of excellent individuals and avoid evolution stagnation or convergence to local optimum. The core of the algorithm improvement is to keep high-quality individuals in the evolution process of the population by determining the average individual adaptation of the current population, and constantly modify the probability calculation method to improve the convergence speed of the algorithm. Thus, iiga can effectively prevent the local optimal trap and reduce the calculation time while avoiding premature convergence.
2.3 Algorithm design

Design of gene coding. Binary coding can effectively improve the speed of algorithm, and real coding can maximize the effect of data mining [7].

\[ F(x) = W_S \cdot \frac{S(x)}{S_{min}} + W_C \cdot \frac{C(x)}{C_{min}} \]  

(5)

Where \( W_S + W_C = 1 \), \( W_S \) and \( W_C \) are constants, and \( S_{min} \) is the preset support threshold. \( C_{min} \) is the preset confidence threshold. \( S(x) \) is the support value and \( C(x) \) is the confidence value.

Design of immune memory function. The core idea of IGA is similar to the working principle of immune system. The corresponding database is built in the memory unit to store the memory of pathogens. When pathogens invade again, the database is searched quickly to find the saved data, so as to effectively improve the speed of iterative computing.

Design of antibody improvement and blocking function. IGA introduces immune algorithm into GA, which makes full use of the advantages of immune algorithm and improves the fitness function in combination with practical problems. The adaptive cross mutation probability is also adjusted continuously to improve the flexibility of individual iteration. The implementation method is as follows:

Antibody can be expressed by fitness function \( F(x) \), the problem with solution is antigen, and the non empty antigen set is expressed by \( x \). The matching relationship between antibody and antigen set \( x \) was as follows.

\[ \sum_{j=1}^{N} \left| F(x_i) - F(x_j) \right| \]  

(6)

The antibody concentration is

\[ D(x_i) = \frac{1}{\rho(x_i)} = \frac{1}{\sum_{j=1}^{N} \left| F(x_i) - F(x_j) \right|} \]  

(7)

The selection operation based on antibody concentration is as follows.

\[ P_{S}(x_i) = \frac{\rho(x_i)}{\sum_{i=1}^{N} \rho(x_i)} = \frac{\sum_{j=1}^{N} \left| F(x_i) - F(x_j) \right|}{\sum_{i=1}^{N} \sum_{j=1}^{N} \left| F(x_i) - F(x_j) \right|} \]  

(8)

The mixed selection probability based on antibody concentration and fitness is

\[ P(x_i) = \theta \cdot P_{S}(x_i) + (1 - \theta) \cdot P_f = \theta \cdot \frac{\sum_{j=1}^{N} \left| F(x_i) - F(x_j) \right|}{\sum_{i=1}^{N} \sum_{j=1}^{N} \left| F(x_i) - F(x_j) \right|} + (1 - \theta) \frac{F(x_i)}{\sum_{i=1}^{N} F(x_i)} \]  

(9)

Where \( P_f \) is the probability of selecting individuals based on fitness; \( \theta \) is the attenuation coefficient of antibody concentration, and the value range of the coefficient is \( 0 < \theta < 1 \).

Using the mixed selection probability calculation method to determine the selection probability can maintain the diversity of population and improve the convergence rate. If the fitness function value is larger, the larger the selection probability is, the population size will be maintained to a certain extent; correspondingly, the higher the antibody concentration is, the smaller the probability of selection is, and the population size will be rapidly reduced.

The introduction of adaptive cross mutation makes the algorithm adapt to its mutation rate effectively and improve the convergence of the algorithm rapidly. When using IGA, the crossover probability changes with the fitness value. The calculation formula of the cross probability \( P_c \) is shown in formula (3). The general values of \( P_{c1} \) and \( P_{c2} \) are 0.9 and 0.6 respectively.
When using classical GA, mutation can be defined as a supplementary search operation, and the mutation probability usually can be defined as a fixed value, which is helpful to keep the individual differences of population. However, in the IGA algorithm, the variation probability $P_m$ varies with the fitness value, and the calculation formula is shown in formula (4). Generally, the value of $P_{m1}$ is 0.1, while that of $P_{m2}$ is 0.001.

2.4 Algorithm flow
IGA introduced above uses the immune algorithm to construct the fitness function, at the same time we adjust the formula of crossover and mutation probability adhere to the actual fitness value.

The IGA is implemented as follows:
1. Algorithm start, set algorithm parameters
2. Perform real coding operation to generate initial population;
3. Perform a full retrieval operation on the database, calculate $F(x), S(x)$ and $C(x)$ of each person in storage database I;
4. Adding an overrun individual to the rule table; otherwise, perform the following steps;
5. Perform update as well as block operations in the antibody library;;
6. Perform adaptive crossover and mutation operations;
7. If the number of iterations exceeds the set value, the algorithm stops; otherwise, go to step (3);
8. Finally, we get the rule table and the rule result through the output.

3. Experimental verification
Programming is done on the MATLAB 2018. The experimental data set we use is open source UCI Ecoli data set. The algorithm parameters we used above are: $P_c$=0.95, $P_m$=0.01, $S(x)$=0.3, $C(x)$= 100, data amount is 100, iteration times are 300; population iteration times are 40; population genetic algebra is 40. The experimental results are shown in the Figure 1.

![Figure 1. Experimental results](image)

Compared with other two algorithms, IGA has a shorter running time, which is due to the fact that vector distance is mainly considered in concentration selection. In the process of algorithm implementation, the crossover and mutation operations can constantly adjust the selection probability, which makes IGA more accurate and efficient in the implementation of data mining.
4. Conclusion
We combine the immune algorithm with the traditional GAm, redesign the corresponding fitness function, and adjust the crossover and mutation probability during the algorithm execution, so as to get the improved IGA. The principle and process of the new algorithm are introduced in detail. Through the comparison of IGA, IgA and Apriori algorithm, we can see that the data mining efficiency of IGA is high, and the accuracy is also excellent. The results provide a reference for the direction of data association rule mining.

Acknowledgments
This work was supported by Major and Special Project of Taizhou Vocational & Technical College.

References
[1] Hills, J., Bagnall, A., Iglesia, B., et al. (2013) BruteSuppression: a size reduction method for Apriori rule sets. Journal of Intelligent Information Systems, 40(3):431-454.
[2] Lazcorreta, E., Botella, F., Fernández-Caballero, A. (2008) Towards personalized recommendation by two-step modified Apriori data mining algorithm. Expert Systems with Applications, 35(3):1422-1429.
[3] Gong, C., Xu, C.H., Wang, J. (2018) An Efficient Adaptive Real Coded Genetic Algorithm to Solve the Portfolio Choice Problem Under Cumulative Prospect Theory[J]. Computational Economics, 52(21):1-26.
[4] Sevinç, E., Coşar, A. (2018) An Evolutionary Genetic Algorithm for Optimization of Distributed Database Queries. Computer Journal, 2018, 54(5):717-725.
[5] Soroudi, A., Ehsan, M. Caire, R., et al. (2011) Hybrid Immune-Genetic Algorithm Method for Benefit Maximization of DNOs and DG Owners in a Deregulated Environment[J]. Iet Generation Transmission & Distribution, 5(9):961-972.
[6] Zhou, J., Zu, Y.X. (2010) A parallel immune genetic algorithm in adaptive resource allocation for cognitive radio network. Acta Physica Sinica, 59(10):7508-7515.
[7] Sellami, K., Ahmed-Nacer, M., Tiako, P.F., et al. (2013) Immune genetic algorithm for scheduling service workflows with QoS constraints in cloud computing[J]. South African Journal of Industrial Engineering, 24(3):68-82.