A Discretization Algorithm Based on Cultural Immune Algorithm

Guangjun Yang

Mechanical Electronic Engineering Department, Dezhou University, Dezhou 253023, China, dzuygj@163.com

Abstract

Now more and more researchers pay attention to discretization of the continuous attributes. A hybrid evolutionary system based on immune algorithm and cultural algorithm is proposed for discretization which combines the strengths of evolutionary computing, social computing in this paper. It uses the cultural algorithm framework in which the immune algorithm embedded. Immune algorithm discretizes the continuous data and the cultural algorithm constitutes the commonly accepted beliefs to guide and speed up the search. In addition, a new diversity operator is put forward in order to maintain the diversity of population. Finally, experiments are conducted for extensive performance testing of the new method on several publicly available data sets. The results show that the algorithm has higher convergence speed and less classification errors than many previously known discretization methods.

Keywords: Discretization, Continuous Attribute, Cultural Algorithm, Immune Algorithm

1. Introduction

Many real-world databases are composed of continuous attributes whose values are integers or real numbers. Before mining data from continuous attributes, continuous attributes are frequently converted into symbolic ones during a special process, called discretization[1]. There are some classical discretization algorithms, for example, ChiMerge[2], Entropy-MDLC[3], and so on. However, the discretization process generally leads to a loss of information. A good discretization algorithm is to minimize such information loss. Many researchers are now increasingly focused on intelligent algorithms for discretization [4-6]. The artificial immune algorithm is a kind of evolutionary computational models for solving some optimization problems[7-8]. In fact, the discretization process is to search the optimal cut-points sets of the continuous attributes. So the immune algorithm can be used for discretization. But in the evolutionary process the blindness and randomness of crossover and mutation lead to slower convergence. Cultural algorithm [9-11] can fuse any population-based computation algorithms into the framework of the culture, which makes population evolution in a certain speed and rapidly adapt to the environment.

In this paper, a new algorithm is proposed for discretization. It uses the cultural algorithm framework in which the immune algorithm embedded. In this algorithm, immune algorithm is used to discretize the continuous data and the cultural algorithm is used to constitute the commonly accepted beliefs to guide and speed up the search. In addition, a new diversity operator is put forward in order to maintain the diversity of population. The results of experiments show that the algorithm has higher convergence speed and less classification errors than many previously known discretization methods.

This paper is organized as follows. In section 2, the related works of discretization, immune algorithm and cultural algorithm are reviewed in brief. The details of our new algorithm which is based on cultural immune algorithm for discretization are presented in section 3. Then, the experiments for testing the efficiency and effectiveness of our algorithm are proposed in section 4. Finally, the conclusion is given in section 5.

2. Related works

2.1. Discretization

The discretization is the process of partitioning the range of a continuous attribute into intervals. And then the values of the attribute fall in each interval are represented as different symbols or integers. Some definitions are described as follows.
Set decision table be \( DT = (U, A, V, f) \), where \( U = \{x_1, x_2, \ldots, x_n\} \) is the finite universe, and \( A = C \cup D \) is the set of attributes where \( C \) is the set of the condition attributes and \( D \) is the set of the decision attributes. Set \( a \in C \) be any one of continuous attributes. The domain of \( a \) is represented as \( V_a = [l_a, r_a] \). Set \( \{P_1^a, P_2^a, \ldots, P_k^a\} \) be the cut-point set of \( a \), where 

\[
I_a = P_0^a < P_1^a < P_2^a < \ldots < P_k^a < P_{k+1}^a = r_a \\
V_a = [l_a, P_0^a] \cup [P_1^a, P_2^a] \cup \ldots \cup [P_k^a, P_{k+1}^a] = r_a
\]

So \( P = \bigcup_{a \in C} P^a \) defines a new decision table \( DT^P = \{U, A^P, V^P, f^P\} \), where \( A^P = \{a^P : a \in A\} \). Through the above discretization, the values of the continuous attributes become the integers.

Now there are more and more algorithms for discretization. Generally, the idea of these algorithms is not to change the compatibility of decision table when discretization, that is, discretization should select the least cut-points in the conditions that keep the decision table compatibility. The more influential method is the rough set and boolean reasoning approaches presented by Skowron et al[1]. This method had completeness, that is, in theory it could find all possible combinations of cut-points, but the computational complexity was exponential. Therefore, some greedy algorithms are presented in [12-14] based on [1]. These algorithms were based on the discriminability of cut-points and instances, which were local optimization algorithms. In [15], MODL is proposed by Boulle. This discretization method built an optimal criterion based on a Bayesian model. A dynamic programming approach and a greedy heuristic approach were developed to find the optimal criteria. In [5-6], they used genetic algorithm for discretization, which belonged to global searching algorithms.

Optimal discretization has been proved to be NP problem[1], while the immune algorithm has good optimization. In this paper, immune algorithm is used to search the cut-points of the continuous attributes.

2.2 Artificial immune algorithm

Artificial immune system (AIS) is a computational intelligence inspired by the biological immune system, which typically exploit the immune system’s characteristics of learning and memory. Now AIS performs a wide range of tasks in various application areas, such as data mining. In AIS antigen and antibody are in action. The antigen is the problem to solve, and the antibody is the generated solution. It computes the affinity between the antigens and antibodies, and the antibodies which have higher affinity can survive. The evolution process also includes cell proliferation, immune selection, crossover, mutation to generate new population. The classical immune algorithm can be presented as follows.

Step1: initialize the antibody population \( A_0 \); 
Step2: Compute the affinity of antibodies; 
Step3: select a number of higher affinity antibodies; 
Step4: Perform crossover, mutation by low probability. Then a new population \( A_{k+1} \) is generated; 
Step5: \( k = k + 1 \), repeat steps 2-5 until a termination criterion is met.

2.3 Cultural algorithm

Reynolds et al[9] firstly developed cultural algorithm (CA) in 1994. CA is a class of evolutionary social system which was inspired by the cultural evolution process which is used in solving optimization problems in various domains. It has three major components: a population space, a belief space, and a protocol that describes how knowledge is exchanged between the first two components. The basic pseudo-code description of cultural algorithms is described as follows:

Input: data source, parameters 
Output: optimized set with user specified properties
A Discretization Algorithm Based on Cultural Immune Algorithm
Guangjun Yang

3. The design of DCIA

In this paper, a new algorithm for discretization based on cultural immune algorithm (DCIA) is presented. This algorithm combines the immune algorithm and cultural algorithm, that is, immune algorithm is embedded in population space of the cultural algorithm. It establishes the dual evolution to makes the global convergence more stable and the convergence speed faster. The details of the algorithm are as follows.

3.1. Population space

In population space, immune algorithm is used to search the cut-points of the continuous attributes. The design of the algorithm is as follows.

(1) Encoding scheme

In the proposed algorithm, the antigens and antibodies are encoded by binary code. All candidate cut-points of all attributes are encoded in binary. In detail, each gene bit is corresponding to each cut-point. 1 indicates to select that cut-point, whereas 0 indicates not to select the cut-point. The total length of the chromosome is the sum of the cut-points.

For example, the set of the condition attributes is \( C = \{a_1, a_2\} \). The values of \( a_1, a_2 \) are respectively \( \{P_1^{a1}, P_2^{a1}\}, \{P_1^{a2}, P_2^{a2}, P_3^{a2}\} \). Then the chromosome is decoded into 01110, which indicates that the attributes \( a_1, a_2 \) respectively select the cut-points \( \{P_1^{a1}\}, \{P_1^{a2}, P_2^{a2}\} \).

(2) Calculation of affinity

The function of affinity is an evaluation between antigen and antibody. Generally, a good discretization method depends on two aspects[2]: less inconsistency of information system and less number of the cut-points. Minimizing the inconsistency indicates the indistinguishable instances are less. Minimizing the number of intervals can reduce the penalty of the model number of classification errors.

Set the decision table \( DT = (U, A, V, f) \) be described as section 2. After a discretization, the
The original decision table becomes \( DT^P = \{U, A^p, V^p, f^p\} \). The inconsistency of two instances in the decision table is that the values of the condition attribute are the same, but the values of the decision attribute are not equal. The ratio of the inconsistency is defined as:

\[
R_{inc}(S) = \frac{|Inc(S)|}{|U|}
\]

where \(|Inc(S)|\) is the number of the inconsistent instances, and \(|U|\) is the total number of the instances.

The function of affinity is defined as:

\[
f(X) = \frac{1}{N_s \ast R_{inc}(X)}
\]

where \(X\) is a antibody, \(N_s\) is the selected cut-points of \(X\), and \(R_{inc}(X)\) is the inconsistency ratio based on the selected cut-points of \(X\). \(f(X)\) shows that only the antibodies which have less inconsistency and less number of the cut-points can survive.

(3)Antibody’s diversity operator
The diversity of antibodies is maintained by production and the antibody’s self-regulation[7] so as to discover global solutions not local solutions. An antibody’s diversity operator that is denoted as \(D(X_i)\) is presented in this paper in order to maintain the diversity of population. It depends on the antibody’s concentration that is denoted as \(C(X_i)\) which is the proportion of similar antibodies in the population. \(C(X_i)\) is defined as follows.

\[
C(X_i) = \frac{\sum_{j=1}^{N} S(X_i, X_j)}{N}
\]

Where \(N\) is the population size, \(S(X_i, X_j)\) is the similarity of any two individuals \(X_i\) and \(X_j\), that is,

\[
S(X_i, X_j) = \frac{\sum_{k=1}^{M} X_{ik} \cap X_{jk}}{M}
\]

where \(M\) is the length of the antibody, \(X_{ik} \cap X_{jk}\) denotes that the corresponding bits of \(X_i\) and \(X_j\) carry on “and” operation. \(S(X_i, X_j) \in [0,1]\), if \(S(X_i, X_j)\) equals to 1, it is indicated that the two antibodies are the same. While \(S(X_i, X_j)\) is equal to 0, it is indicated the two antibodies are entirely different.

Depending on the antibody’s concentration \(C(X_i)\), the diversity operator \(D(X_i)\) is defined as:

\[
D(X_i) = \begin{cases} 
\text{increase } X_i p1\% \text{ more, if } C(X_i) < \epsilon 1 \\
\text{reduce } X_i p2\% \text{ less, if } C(X_i) > \epsilon 2
\end{cases}
\]

where \(p1, p2\) are the proportions of increasing or reducing, and \(\epsilon 1, \epsilon 2\) are the similarity constants, generally \(0 < p1 < 100\), \(0 < p2 < 100\) and \(0 < \epsilon 1 < \epsilon 2 < 1\). We take \(p1 = p2 = 30\), \(0 < \epsilon 1 < 0.1\) and \(0.8 < \epsilon 2 < 1\) in this paper. If the antibody’s concentration is lower than \(\epsilon 1\), we increase this antibody more. And if the antibody’s concentration is higher than \(\epsilon 2\), we reduce this antibody less. According to this mechanism, we produce antibodies with low concentration and suppress the antibodies with high concentration so as to ensure the diversity of population and avoid premature.
3.2. Belief space

In the cultural algorithm, five basic categories of cultural knowledge have been identified [10]: situational knowledge, domain knowledge, normative knowledge, topographical knowledge, and history knowledge. The five kinds of knowledge recorded different information are suitable for different situations. In this paper, the belief space of the new algorithm uses the situational knowledge to save excellent individuals and normative knowledge to provide guidelines within which individual adjustments can be made. That is, the structure of the belief space is $<E, N>$, where $E$ represents situational knowledge and $N$ represents normative knowledge.

(1) The situational knowledge

The situational knowledge is used to represent an excellent individual generated from population space. In the immune algorithm the excellent individuals are stored in the memory cells bank. So the situational knowledge structure is $E = <X_1, X_2, \ldots, X_n>$, where $n$ is the capacity of situational knowledge, and $X_i$ is an excellent individual in the memory cells bank.

It is simple to update the situational knowledge. That is, it can add the population’s best individual to the situational knowledge if it outperforms the current best, as shown below.

$$E_{t+1} = \begin{cases} X_{best}^t & \text{if } f(X_{best}^t) < f(E^t) \\ E^t & \text{otherwise} \end{cases}$$

where $X_{best}^t$ is the best individual in the population at time $t$.

(2) The normative knowledge

The normative knowledge is represented as a set of intervals characterizing the range of what is believed to be a good solution for each parameter. These ranges provide guidelines within which individual adjustments can be made. It is useful for tracking the optimum in some types of dynamic behaviors where the change occurs within the parameter ranges. The normative knowledge data structure is $N = <V_1, V_2, \ldots, V_n>$, where $n$ is the number of variables. For each variable $V_i$, the data structure contains the upper and the lower bounds $u_i, l_i$. So the normative knowledge is expressed as follows:

$$\begin{array}{cccccccc}
  & l_1 & l_2 & \cdots & l_i & \cdots & l_{n-1} & l_n \\
  & u_1 & u_2 & \cdots & u_i & \cdots & u_{n-1} & u_n \\
\end{array}$$

In the database containing continuous attributes, each continuous attribute has an upper bound and a lower bound. So in the paper the normative knowledge keeps the upper bounds and the lower bounds of the attributes computing from individuals in situational knowledge.

3.3. The acceptance operation

The belief space will be updated by accepting the excellent individuals from the population space. It is often determined as a percentage of the number of excellent individuals ranging between 1% and 100% of the population size. In this paper, a dynamic acceptance function can be used by adjusting the number of accepted individuals over time, using the following function.

$$accept = p\% + \frac{p\%}{k}$$

where $p$ represents the initial given percentage of the population space that will affect the belief space; $k$ is the number of time steps or generations in the current environment, and is reset to one with every environmental change. As $k$ increases, the number of accepted individual’s decreases. For example, if $p$ in the acceptance function above is set to 20%, then the number of the accepted individuals in the first generation (where $k$ equals 1) will be 40% of the population space. In the second generation (when $k$ equals 2), the number of accepted individuals is 30% of the population space.
This acceptance function is dynamic, that is, in the later evolution process the accepting percentage of excellent individuals is smaller, which can avoid randomness. This strategy can improve the convergent speed and the search accuracy in the later evolution stage.

3.4. The influence operation

The situational knowledge represents excellent individuals in the process of evolution. The normative knowledge represents the parameter range of the best solutions, and it will be used to influence the direction of the search efforts within the promising ranges. In this paper we use normative and situational knowledge to influence the population space. The mutation step size is relative to the distance between the upper and the lower limit of each attribute. The influence rule can be represented as follows:

\[
x'_{i,j} = \begin{cases} 
x_{i,j} + (u_j - l_j) \cdot N(0,1) & \text{if } x_{i,j} < E_j \\
x_{i,j} - (u_j - l_j) \cdot N(0,1) & \text{if } x_{i,j} > E_j \\
x_{i,j} & \text{otherwise}
\end{cases}
\]

(8)

where \(x_{i,j}\) is the \(j\)th variable of the \(i\)th individual, \(x'_{i,j}\) is variable after mutation. \(E_j\) is the best exemplar parameter value in the situational knowledge for variable \(j\) in belief space, \(N(0,1)\) is the random variable subject to standard normal distribution.

This approach uses normative knowledge to influence the change in mutation step size relative to the distance between the upper limit and the lower limit, and situational knowledge to influence the mutation direction toward the current best.

3.5 The steps of DCIA

The pseudo-code description of discretization based on cultural immune algorithm (DCIA) proposed in this paper are as follows.

Input: dataset, parameters
Output: the set of cut-points

Begin
k=0;
Initialize the population space and belief space;
While termination condition is not true do
   In the population space, calculate affinity and carry on immune operator;
   Update the belief space by accepting individuals using the acceptance function;
   Influence the individuals in the population space according to influence function;
   k = k + 1;
End while
Stop

4. Experiments and analysis

To test the efficiency and effectiveness of the DCIA proposed in this paper, some experiments have been carried out. The experiment datasets are from UCI machine learning repository. Five datasets were selected, because each of them only includes continuous attributes and has no inconsistent objects. Table 1 gives the summary of these data sets.
Table 1. Experimental datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>No. of instances</th>
<th>No. of continuous attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>iris</td>
<td>150</td>
<td>4</td>
</tr>
<tr>
<td>heart</td>
<td>270</td>
<td>13</td>
</tr>
<tr>
<td>ecoli</td>
<td>336</td>
<td>7</td>
</tr>
<tr>
<td>austra</td>
<td>690</td>
<td>14</td>
</tr>
<tr>
<td>pima</td>
<td>768</td>
<td>7</td>
</tr>
<tr>
<td>wine quality</td>
<td>4898</td>
<td>11</td>
</tr>
</tbody>
</table>

In order to test the validity of the DCIA, we use other three methods to evaluate the performance. The first is based on greedy algorithm[12], which is denoted as A1. The second is based on the importance of cut-points[14], which is denoted as A2. The third is based on genetic algorithm[5], which is denoted as A3. And our algorithm is denoted as A4. The results are listed in Table 2 and Table 3.

Table 2. Discrete results

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Consistency</th>
<th>No. of cut-points</th>
<th>Execution time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A1</td>
<td>A2</td>
<td>A3</td>
</tr>
<tr>
<td>iris</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>heart</td>
<td>0.981</td>
<td>0.982</td>
<td>0.981</td>
</tr>
<tr>
<td>ecoli</td>
<td>0.963</td>
<td>0.970</td>
<td>0.969</td>
</tr>
<tr>
<td>austra</td>
<td>0.977</td>
<td>0.974</td>
<td>0.982</td>
</tr>
<tr>
<td>pima</td>
<td>0.920</td>
<td>0.914</td>
<td>0.916</td>
</tr>
<tr>
<td>wine quality</td>
<td>0.659</td>
<td>0.785</td>
<td>0.678</td>
</tr>
</tbody>
</table>

From Table 2, it can be seen that the consistent degrees of the five sets are changed small after the discretization by our algorithm DCIA, while degraded much by other algorithms. The number of cut-points of each dataset is the least by our algorithm. Especially in larger dataset, this superiority is obvious. This is because the knowledge of culture algorithm guides the evolution of immune algorithm and weakens the blindness and randomness of the immune algorithm so as to speed up the search.

To evaluate classification accuracies of our algorithm, we use ID3 algorithm[16], which is a well-known method and worked well for many decision-making problems. In this experiment, we firstly use the above four algorithms to discretize the continuous attributes, and then use ID3 to extract the classification rules. The results of classification accuracies are shown in table 3.

Table 3. The accuracy of classification by ID3

<table>
<thead>
<tr>
<th>Datasets</th>
<th>ID3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A1</td>
</tr>
<tr>
<td>iris</td>
<td>98.8%</td>
</tr>
<tr>
<td>heart</td>
<td>98.1%</td>
</tr>
<tr>
<td>ecoli</td>
<td>95.2%</td>
</tr>
<tr>
<td>austra</td>
<td>91.6%</td>
</tr>
<tr>
<td>pima</td>
<td>90.1%</td>
</tr>
<tr>
<td>wine quality</td>
<td>64.6%</td>
</tr>
</tbody>
</table>

From Table 3, it can be concluded that the performance of DCIA is more or less enhanced than that of other algorithms. This is because that the affinity in DCIA depends on two aspects: inconsistency of information system and number of the cut-points. The individuals which have less inconsistency and less number of cut-points can be retained. In addition, a new diversity operator is put forward to maintain the diversity of population in order to avoid premature.
From the experimental results, the discretization based on cultural immune algorithm has fast convergence, with good global and local search capabilities.

5. Conclusion

It is popular to discretize continuous attributes now. A novel algorithm based on culture algorithm and immune algorithm is presented in this paper. There are several features of the algorithm.

1. In the population space of culture algorithm, immune algorithm is used to discretize the continuous data rapidly. In the belief space of the culture algorithm, the situational knowledge and normative knowledge are redefined to guide immune algorithm and speed up the search.

2. In immune algorithm, the affinity is defined based on less inconsistency of information system and less number of the cut-points so as to improve the quality of discretization. In addition, a new diversity operator is put forward in order to maintain the diversity of population.

3. The results show that the algorithm has higher convergence speed and less classification errors than many previously known discretization methods. The method maintains the original indistinguishable relationships of the information systems as possible as it can.

6. References