Evolutionary Granular Computational Model and Applications

Jake Zhang
张江
Beijing Jiaotong University
Problems

- New Problems in Information age
  - Database → Data Tomb
  - Information Garbage
  - Data Mining & KDD are very important

- Complexity science
  - Emergence
  - Evolution
  - How to use self-organizing to discover knowledge in database.
Background

- **Evolutionary computation**
  - Basic building blocks (Nontrivial) as the population
  - Stochastic combinations of building blocks
  - Mutations with small possibility
  - Pressure of selection gives the evaluation feedback to the population.
  - After many iterations $\rightarrow$ Emergent behaviors or properties will be observed

- **How to find building blocks in data mining?**
  - Granular computation
  - Relational database $\rightarrow$ Information Granules
  - Granular computation: computing with word
Basic Idea

A relational database can be regarded as an information table

![Table 1: Information table](image)

According to [1], this Information table implies many basic logical formulas, i.e., a pair of attribute and value \((a, v)\), such as:

\[ a_1 = v_1, a_3 = v_3, \text{ example: color=red, size=big} \]

\[ a_1 = v_1 \land a_2 = v_2, \text{ example: color=red} \land \text{ size=big} \]

\[ a_1 = v_1 \lor a_2 = v_2, \text{ example: color=red} \lor \text{ size=big} \]
Building blocks

- **Formula**
  - One formula is a classifier
    - $a_i = v \colon \{ u \mid a_i(u) = v \}$, Such kind of formulas are called atom
    - $a_i = v \land a_j = v_1 \colon \{ u \mid a_i(u) = v \text{ and } a_j(u) = v_1 \}$, they are combinations of atoms
  - The meaning of the formula
    If $\Phi$ is a formula, then we define the set:
    $$m(\phi) = \{ x \in \mathcal{U} \mid x \models \phi \}$$
    as the meaning of the formula.

- **Granule**
  - The pair $\langle \Phi, m(\Phi) \rangle$ is called a granule. Here $\Phi$ is a formula and $m(\Phi)$ is its meaning. $m(\Phi)$ is called the formula part of the granule, and as the set part. If $\Phi$ is an atom, then $\Phi$ is an atomic granule.
Combination of building blocks

- New granules can be obtained by basic operators.
- “\” and “\” operators can be used to create more specific/general granules.
- All of the operators are applied on both two parts (formula and its meaning) of the granule.
- We can define various operators on granules except for “\” and “\” as long as they can be used both on formula and set.
Compatibility of granules

- For any two atomic formulas $\alpha$ and $\beta$ ($\alpha : a_1 = v_1$, $\beta : a_2 = v_2$), if $a_1 = a_2$ but $x_1 \neq x_2$, then they are incompatible and can be denoted as $\alpha \downarrow \beta$. If $a_1 \neq a_2$ or $(a_1 = a_2$ and $x_1 = x_2)$ then they are compatible, denoted as $\alpha \uparrow \beta$.

- The atom is compatible with an “And”/”Or” combination of formulas if the atom is compatible with every/at least one atom in the formula.

- If the formula parts of two granules are compatible, then these granules are compatible.
Collections of Granules

- A record in information table is a vector of formulas, such as \( a_1 = v_1, a_2 = v_2, \ldots, a_m = v_m \)

- If a granule’s formula is compatible with all of these formulas, then this granule is activated by the input record

- Consider a collection of granules in a system. Some incompatible granules may co-exist in the system.

- An input record is a stimulus that can activate many granules in the collection. All of active granules form a new collection of active elements.
Granule Network

- When the attributes of the information table can be divided into the conditional attributes and decisional attributes, we can define granule network.
- If one granule $\Phi$ in conditional attribute is active always followed by an activation granule $\Gamma$ in decisional attribute, then we denote $\Phi \rightarrow \Gamma$
- All of $\Phi \rightarrow \Gamma$ granule pairs can compose a granule network
Evolutionary granular computational model

- The system contains:
  - A collection of granules: *Elements*
  - A granule network
  - An information table which is the environment of the system

- The dynamic of the system:
  - Scan the information table one record by one record
  - Forming some basic granules in *Elements*
  - Active some granules in elements when input new record
  - Changing the topology of the network
  - The system will adapt to its environment
  - The collection of granules and the granule network are the outputs of the system
Adaptation to the environment

- Database is the environment of the system
- Using fitness function to evaluate every granule and connections between granules
- Keeping a fixed number of population of granules, killing low fitness valued granules and connections
- The power of creativity and the selective pressure can make granules and network evolve to adapt the environment
Pressure of selection

- Fitness of the granule
  \[ f(\text{act}, \text{age}, \text{len}, \text{likeness}) \]
  this function has 4 variables, they are
  - the activated times
  - the age of existing in the system
  - the number of atoms containing
  - the similarity degree to the input data

- Strength of connection
  - Strength of each connection: \[ f(\text{right}, \text{wrong}) = \omega_1 \text{right} + \omega_2 \text{wrong} \]
  - Variables are right times and wrong times.
  - Consider a connection \( a \rightarrow b \), if \( a \) is active and \( b \) is active in the same time, the connection is doing right, so right++, otherwise wrong++
Learning algorithm of the system

1. Read the new record as \textit{InputRecord};
2. Loop for all elements in \textit{Elements}, activate the granules that are compatible with \textit{InputRecord};
   2.1 Loop for all active granules in \textit{activeElements}, strengthen or weaken the connections between granules.
   2.2 Create new connections among granules randomly.
3. Separate the formulas group \textit{InputRecord}, add new granules into \textit{Elements}.
4. Create new granules with possibility $P_{create}$ to add into \textit{Elements}.
5. If the total amount of granules in the \textit{Elements} exceeds \textit{MaxElementNum}, then delete several granules with the smaller fitness degree.
Problem solving phase

- Input an incomplete record to the system, it can activate some competing granules and network connections to reason out the empty attribute’s value.
Case study

Original database (In [4] on page 117)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case study (collection of granules)

Extracted Granules

<table>
<thead>
<tr>
<th>Granule</th>
<th>Fitness</th>
<th>Granule</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>e=3</td>
<td>78.81</td>
<td>d=1</td>
<td>33.91</td>
</tr>
<tr>
<td>b=2</td>
<td>26.20</td>
<td>b=1</td>
<td>23.10</td>
</tr>
<tr>
<td>c=1</td>
<td>23.10</td>
<td>d=2</td>
<td>21.60</td>
</tr>
<tr>
<td>e=2</td>
<td>21.32</td>
<td>d=1∧b=2</td>
<td>18.51</td>
</tr>
<tr>
<td>e=1</td>
<td>16.31</td>
<td>c=1∧d=2</td>
<td>13.32</td>
</tr>
</tbody>
</table>
Case study (connections of the granule network)

Granule network’s edges

Table 3  Granule connections and their strength.

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>d=1-&gt;e=3</td>
<td>47</td>
</tr>
<tr>
<td>b=1Λc=2Λd=2-&gt;e=1</td>
<td>28</td>
</tr>
<tr>
<td>d=1Λa=2-&gt;e=3</td>
<td>23</td>
</tr>
<tr>
<td>a=1Λc=1Λd=2-&gt;e=2</td>
<td>22</td>
</tr>
<tr>
<td>a=3Λb=1Λc=1-&gt;e=3</td>
<td>22</td>
</tr>
<tr>
<td>d=1Λb=2-&gt;e=3</td>
<td>39</td>
</tr>
<tr>
<td>a=1Λd=1-&gt;e=3</td>
<td>25</td>
</tr>
<tr>
<td>c=1Λd=2-&gt;e=2</td>
<td>23</td>
</tr>
<tr>
<td>a=1Λb=1Λd=1 -&gt;e=3</td>
<td>22</td>
</tr>
<tr>
<td>a=3Λc=2Λd=1-&gt;e=3</td>
<td>22</td>
</tr>
<tr>
<td>d=1Λb=2-&gt;e=3</td>
<td>39</td>
</tr>
</tbody>
</table>
Simple problem solving

Input attributes and values with some attributes unknown, the system can reason out the absent attributes values

<table>
<thead>
<tr>
<th>Input</th>
<th>System’s answers</th>
<th>Right answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(a=1, b=1, c=2, d=2)$</td>
<td>$e=1$</td>
<td>$e=1$</td>
</tr>
<tr>
<td>$(a=2, b=1, c=1, d=2)$</td>
<td>$e=2$</td>
<td>$e=2$</td>
</tr>
<tr>
<td>$(a=2, b=2, c=2, d=2)$</td>
<td>$e=3$</td>
<td>$e=3$</td>
</tr>
<tr>
<td>$(a=3, b=1, c=2, d=2)$</td>
<td>$e=3$</td>
<td>Null</td>
</tr>
<tr>
<td>$(c=1, d=2)$</td>
<td>$e=2$</td>
<td>Null</td>
</tr>
</tbody>
</table>
Features of evolutionary granular computational model

- The results from the EGCM are imperfect as precisely compared with the traditional approaches of data mining
- It is an open system that can adapt to the database environment forever
- Flexibility and adaptability are the advantages of this approach
Thank you~

Jake
Reference


