Using Vectors of Features for Finite State Automata Dataset Reduction

TAWFIQ A. AL-ASSADI and ABOOD KIREBUT JASSIM

1Collage of Information Technology, Babylon University, Iraq.

(Received: November 06, 2014; Accepted: December 01, 2014)

ABSTRACT

A finite state automata is the most important type of graphs, which is called conceptual graphs, while the expansion of using the graphs in the process of data mining, the use of FSM is still limited because of the difficulty in processing in databases, therefore in order to find methods that make it easier to deal with large groups of machines, as a database, is encourage to use of this type of representation in this paper of graph mining. This paper presents a method using vectors of features for find machines matching, which is one task of mining graph data, which are frequently found in a single environment or similar environments, thereby reducing the number of records and increase efficiency of mining tasks.

Key words: Data reduction, Essential machines, FSM, Graph mining, Machine matching vectors of features.

INTRODUCTION

The graph mining has been raped sudden increase in the data mining. Graph mining denotes a group of algorithms for mining the relational aspects of data represented as a graph.[1]

The graph depiction that set of nodes and links between nodes allow practice of mining algorithm. Graph-based data mining has two major approaches: frequent subgraph mining and graph relational data [2]. The center of graph mining is to extract important knowledge from graph represented data by techniques from fields such as data mining, machine learning, pattern recognition, statistics and graph theory.[1]. The graph is a data construction[3](which implements the algorithms on which a number of types of graph, including together direct and indirect, labeled or not and weighted or not.[4]

Finite-state machines offer a simple computational form with many applications [5], finite-state machine (FSM) or finite-state automaton (plural: automata), or simply a state machine, is an arithmetical model of computation used to propose both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of states. The machine is in only one state at a time; the state it is in at any given time is called the current state. It can transform from one state to another when initiated by a triggering occurrence or
condition; this is called a transition. A particular FSM is defined by a set of its states, and the triggering condition for each transition.

FSM are important techniques of graphs [6] and formulas that define the conceptual graphs and consists of states, transitions, and be in two basic formats [7].

**Machines Matching Method**

The selected database is composed of machines that represent data in an environment or similar other environments to enable the mining algorithms to find the best results, the following steps depict the method used as.

First, all FSM are converted into vectors of features that are extracted from environment or r environments of similar features to make the database composed of only vectors representing machines.

Second step is the process of detecting matches between vectors by comparing each one with the other vectors features in the dataset and when there is a similarity between them, the number of the original vector placement to the similar vector is given and so on and hence two types of transactions in the database.

The first type are essential records and the second type are the similar records, (see in the table 1).

<table>
<thead>
<tr>
<th>Table. 1: Frequent FSM in data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>F1</td>
</tr>
<tr>
<td>F2</td>
</tr>
<tr>
<td>F3</td>
</tr>
<tr>
<td>F4</td>
</tr>
<tr>
<td>F5</td>
</tr>
<tr>
<td>F6</td>
</tr>
<tr>
<td>F7</td>
</tr>
<tr>
<td>F8</td>
</tr>
<tr>
<td>F9</td>
</tr>
<tr>
<td>F10</td>
</tr>
<tr>
<td>F11</td>
</tr>
<tr>
<td>F12</td>
</tr>
</tbody>
</table>

The data mining tasks will deal with the first type, while the second type ignored.

The calculation of the matching process need efficient function to get the job done accurately and therefore be choose the equality function as in the proposed algorithm, see in fig(2).

The process continues until all vectors are allocated correctly.

The dataset are then considered to be unique machines, see table (2).

<table>
<thead>
<tr>
<th>Table. 2: essential type of machine (first type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
</tr>
<tr>
<td>F2</td>
</tr>
</tbody>
</table>

**Fig. 1: FSM1 with their vector**

| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

**Fig. 2: FSM2 with their vector**

| 1 | 0 | 0 | 0 | 0 | / | 1 |

**Fig. 3: FSM3 with their vector**

| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

Fig. 1: FSM1 with their vector

Fig. 2: FSM2 with their vector

Fig. 3: FSM3 with their vector
Fig. 4: FSM4 with their vector

Fig. 5: FSM5 with their vector

Fig. 6: FSM6 with their vector

Fig. 7: FSM7 with their vector

Fig. 8: FSM8 with their vector

Fig. 9: FSM9 with their vector

Fig. 10: FSM10 with their vector

Fig. 11: FSM11 with their vector
To illustrate the method the twelve machines are used as an example to be converted in the vectors of features (1 for feature existence, 0 for not) with set of feature {a, b, ab, ba, ai, bai, abi} to be the dataset for mining task as shown in fig(1).

1. Input set of machines
2. Output essential of machines data set
3. for each machine
4. Extract features vector
5. End
6. Set sim[0], flag=0 // sim is Similarity machine vector
7. For k = 1 to n // n is a number of vectors
8. For i = 1 to n
9. For j = 1 to m // m is a length of features vector
10. begin
11. if f(k, m) = f(j, m) flag = 1
12. end j
13. if flag = 0
14. sim(j) = k
15. if flag = 1 then flag = 0
16. end i
17. end k
18. for i = 1 to n
19. for j = i+1 to n
20. if sim(i) = -1
21. if sim(j) = sim(i)
22. sim(j) = -1
23. end for i
24. end for j
25. for l = 1 to n
26. if sim(l) = -1
27. output f(i)
28. end for i
29. end for

Figure (2) finite state machine matching algorithm

RESULTS AND DISCUSSION

After implementing the algorithm the placement for each finite state machine will be as seen in table (1).

By ignored the second type of machines the essential machines will be present, shown in table(2).

CONCLUSION

1. Data generates in one environment or in similar environments will generate many machines similar.
2. The tasks of data mining is the most efficient and fastest method with Trimming effect.

REFERENCES

2. Yan, Xifeng, X. Zhou, and Jiawei Han, “frequent subgraph mining and graph relational knowledge Mining closed relational graphs with connectivity constraints.” Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, ACM, 2005.