Comparative Study and Results Analysis of Feature Vector and Adjacency Matrix for Representing FSM for Data Reduction

Tawfiq Abdulkhaleq Abbas, Ph.D.
Collage of Information Technology, Babylon University, Iraq

Abbood Kirebut Jassim
Collage of Science, Babylon University, Iraq

ABSTRACT

A finite state automaton is conceptual graphs which are considered to be an important type of graph method. As a result of the expansion of using the graphs in the process of data mining, the use of FMS is still limited because of the difficulty in processing databases, therefore this paper is to find an approach that make it easier to deal with large groups of machines as a database is encourage to use of this type of representation in mining techniques. This paper gives a approach for finding a match between machines, which appear frequently in a single environment or similar environments, the approach consist of two methods one for machines matching as adjacency matrices and another method for matching machines as vectors of features ,hence prove that second method more efficient to control the match processing

Keywords Features, Finite State Automata, Data Reduction, Feature Extraction, Vector Feature.

1. INTRODUCTION

The field of graph mining has seen raped explosion in the data mining.

Graph mining denotes a set of algorithms for mining the relational aspects of data represented as a graph [1].

The graph representation that collection of nodes and links between nodes accept process of mining algorithm Graph-based data mining has two main approaches: frequent subgraph mining and graph relational knowledge [2].The core of graph mining is to extract valuable knowledge from graph represented data by via techniques from fields such as data mining, machine learning, statistics, pattern recognition and graph theory[1], pattern recognition and graph theory.

The graph is a data structure [3] (which implements the algorithms on which several types of graph, including both direct and indirect, labeled or not and weighted or not.[4]

Finite-state machines provide a simple computational model with many applications [5]. Wikipedia web site has a good definition of the FSM that described it as follows, finite-state machine (FSM) or finite-state automata, or simply a state machine, can be defined as a mathematical model of computation; the main purpose is for designing of both computer programs and sequential logic circuits. It is considered 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 be changed from one state to another when initiated by any event or condition that may trigger which is called a transition. The FSM can be defined by a list of its states along with the triggering condition that label the each transition.

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

2. MACHINES MATCHING MINING APPROACH

Database Selected is composed of machines that represent data in an environment or other similar 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 adjacency matrices and vectors of features that are extracted from environment or environments of similar features to make the database composed of adjacency matrices and vectors representing machines .

The second step is the process of detection of matching e between arrays or vectors by comparing each array with other arrays and vector with other vectors, where there is a similarity between them, the number of original machine placement to the similar machine and so on and hence there are two types of transactions in the database.

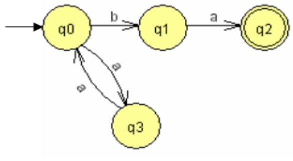
The first type are essential records and the second type are the similar records,(see in the table 1).

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 continue until customize all machines and vectors to appropriate essential machines and vectors of features.

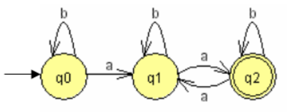
To illustrate the approach twelve machines are used as example and converted in adjacency matrices and vectors of features(chosen features set is{a,b,ab,ba,ai,bai,abi}) to be dataset ready for mining task as shown in figure(1) in steps.



	q₀	q₁	q₂	q₃
q₀	-	b	-	a
q₁	-	-	a	-
q₂	-	-	-	-
q₃	a	-	-	-

1	1	1	1	0	0	0

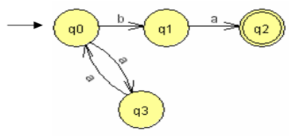
Fig (1.1) FSM1 with their Matrix and vector



	q_0	q_1	q_2
q_0	b	a	-
q_1	-	b	a
q_2	-	a	b

1	1	1	1	0	0	0

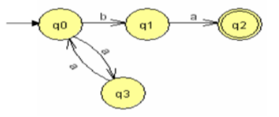
Fig (1.2) FSM2 with their Matrix and vector



	q_0	q_1	q_2	q_3
q_0	-	b	-	a
q_1	-	-	a	-
q_2	-	-	-	-
q_3	a	-	-	-

1	1	1	1	0	0	0

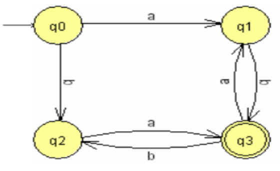
Fig (1.3) FSM3 with their Mtrix and vector



	q_0	q_1	q_2	q_3
q_0	-	b	-	a
q_1	-	-	a	-
q_2	-	-	-	-
q_3	a	-	-	-

1	1	1	1	0	0	0

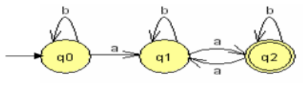
Fig(1.4) FSM4 with their Matrix and vector



	q_0	q_1	q_2	q_3
q_0	-	a	b	-
q_1	-	-	-	b
q_2	-	-	-	b
q_3	-	a	b	-

1	1	1	1	0	0	0

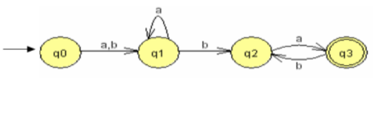
Fig(1.5) FSM5 with their Mtrix and vector



	q_0	q_1	q_2
q_0	b	a	-
q_1	-	b	a
q_2	-	a	b

1	1	1	1	0	0	0

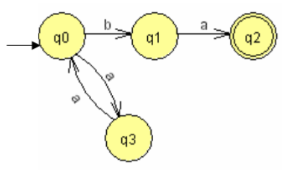
Fig(1.6) FSM6 with their Mtrix and vector



	q_0	q_1	q_2	q_3
q_0	-	a,b	-	-
q_1	-	a	b	-
q_2	-	-	-	a
q_3	-	-	b	-

1	1	1	1	0	0	0

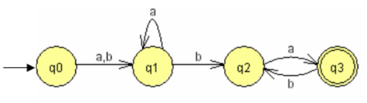
Fig (1.7) FSM7 with their Mtrix and vector



	q_0	q_1	q_2	q_3
q_0	-	b	-	a
q_1	-	-	a	-
q_2	-	-	-	-
q_3	a	-	-	-

1	1	1	1	0	0	0

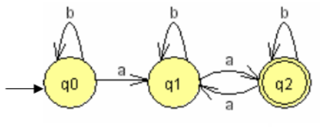
Fig(1.8) FSM8 with their Mtrix and vector



	q_0	q_1	q_2	q_3
q_0	-	a, b	-	-
q_1	-	a	b	-
q_2	-	-	-	a
q_3	-	-	b	-

1	1	1	1	0	0	0

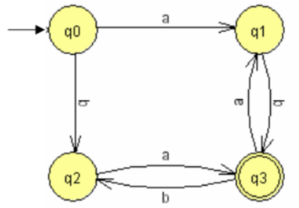
Fig(1.9) FSM9 with their Matrix and vector



	q_0	q_1	q_2
q_0	b	a	-
q_1	-	b	a
q_2	-	a	b

1	1	1	1	0	0	0

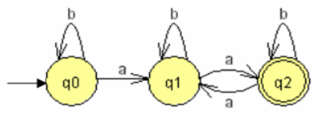
Fig(1.10) FSM110 with their Matrix and vector



	q_0	q_1	q_2	q_3
q_0	-	a	b	-
q_1	-	-	-	b
q_2	-	-	-	b
q_3	-	a	b	-

1	1	1	1	0	0	0

Fig(1.11) FSM11 with their Matrix and vector



	q_0	q_1	q_2
q_0	b	a	-
q_1	-	b	a
q_2	-	a	b

1	1	1	1	0	0	0

Fig(1.12) FSM12 with their Mtrix and vector

However, the algorithm for machine matching can be summarized in the following figure.

```
1- Input set of machines
2- Output essential of machines set by using adjacency matrix represented and essential machines by using vectors of features represented
3- For each machine
4-    Extract vector of features
5-    Convert machine to adjacency matrix
6- End
7- For each matrix and vector
8-    Begin
9-       Compare matrix with other matrices
10-      Compare vector with other vectors
11-          If matrix or vector not placement
```

```
        Begin
            If match    then placement matrix to original matrix number
            If match    then placement vector to original vector  number
    End if
End for
```

Fig (2) Algorithm for machines matching

3. RESULTS AND DISCUSSIONS

The results can be classified into the following categories:

A- Result Using Adjacency Matrices Dataset

The result of apply algorithm give the placement similar machines to original machines as show below in table (1)

Table (1). table of machines placement

machine	replacement
F1	0
F2	0
F3	1
F4	1
F5	0
F6	2
F7	0
F8	1
F9	7
F10	2

The result represents essential machine and number of frequent in dataset as (see in table (2)).

Table (2) of essential machines

F1	4
F2	3
F5	1
F7	2

B- Results Using Vector of Features Data Set

After algorithm done, the placement for each vector of features (see in table (3)).

Table (3) placement vectors in the dataset

Machine	Replacement
F1	1
F2	2
F3	1
F4	1
F5	2
F6	1
F7	1
F8	1
F9	1
F10	2
F11	1
F12	2

Where ignored the second type of machines the essential type will rest as shown in table (4)

Table (2) essential type of machine (first type)

F1
F2

4. CONCLUSIONS

1- That generates data in one environment or in similar environments will generate many machines Similar.

2- The tasks of data mining is the most efficient and fastest data with Trimmed.

3- The vectors of features method more controlled for matching processing.

5. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

6. REFERENCES

[1] Daine J. Cook, Lawernce B. Holder, "mining graph data" ,willy,Washington 2007.

[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.

[3] Wiskott, Laurenz, et al. "Face recognition by elastic bunch graph matching."Pattern Analysis and Machine Intelligence", IEEE Transactions on 19.7 (1997): 775-779.).

[4] Raymond, John W., Eleanor J. Gardiner, and Peter Willett. "Heuristics for similarity searching of chemical graphs using a maximum common edge subgraph algorithm." Journal of chemical information and computer sciences42.2 (2002): 305-316.)

[5] Casey, Mike. "The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction." Neural computation 8.6 (1996): 1135-1178.

[6] Yen, Ti-Yen, and Wayne Wolf. "An efficient graph algorithm for FSM scheduling." Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 4.1 (1996): 98-112.)

[7] Yang, Shih-Yao, and Von-Wun Soo, "Extract conceptual graphs from plain texts in patent claims." Engineering Applications of Artificial Intelligence 25.4 (2012): 874-887.