AN IMPROVED GRAPH BASED METHOD FOR EXTRACTING ASSOCIATION RULES

Wael AlZoubi

Ajloun University College, Balqa Applied University
PO Box: Al-Salt 19117, Jordan

ABSTRACT

This paper proposes an improved approach to mine strong association rules from an association graph, called graph based association rule mining (GBAR) method, where the association for each frequent itemset is represented by a sub-graph, then all sub-graphs are merged to determine association rules with high confidence and eliminate weak rules, the proposed graph based technique is self-motivated since it builds the association graph in a successive manner. These rules achieve the scalability and reduce the time needed to extract them. GBAR has been compared with three of the main graph based rule mining algorithms; they are, FP-Growth Graph algorithm, generalized association pattern generation (RIOMining) and multilevel association pattern generation (GRG). All of these algorithms depend on the construction of association graph to generate the desired association rules. On the other hand, this chapter expresses the observation results from the implementation of GBAR method recorded through the experiment. The detailed results are shown by different case studies in different minimum support thresholds values ranging from 90% down to 10% and minimum confidence values range from 55% to 95%. Generally, the observations focused on the execution time, the dimensionality of rules and the number of rules generated, because the performance of the association rule mining process affected directly of these criteria. Generally, the GBAR method has successfully reduced the execution time required to generate desired association rules in almost all of the dataset.

KEYWORDS:

Graph, association rules, transaction data

1. INTRODUCTION

There are several graph based algorithms for mining association rules from transaction datasets, some of these algorithms are: FP-Growth-Graph algorithm (Tiwari et al. 2010), graph-based rule-chain incremental online mining algorithm (RIOMining) (Ning et al. 2009), and graph based algorithm for association rules generation (GRG) (Li et al. 2003). In this paper, an improved method for association rules extraction is proposed, which will be called (GBAR), GBAR stands for Graph Based Association Rule mining.

A dataset of transactions must be divided into disjoint groups or clusters before applying the proposed algorithm. The datasets that are used in this paper are included to evaluate the strength of the proposed technique in a broader boundary. Ten datasets have been used as case studies in the experiments, namely Chess and Mushroom. The selected datasets are commonly used in data mining research (Orlando et al. 2003, Cule & Goethals 2010, Margahny&Mitwaly 2005). Some

DOI : 10.5121/ijsea.2015.6301

of these datasets have been cleaned on receipt and others will be in their original form.
It is so difficult to construct an association graph for the frequent itemsets of different lengths, i.e. from the whole clusters of transactions; this is the third challenge in the research. So, a graph based association rules mining (GBAR) has been proposed to construct an association graph for each cluster, GBAR is considered as a solution to this challenge, and this facilitates the traversing of graph, and accelerates the association rules generation.

The evaluation measurements in the graph based phase are focused on four different measurements, i.e. the rule confidence, the number of rules generated, the dimensionality of the rules, i.e. the average number of items per rule, and the time required to mine the desired association rules from the association graph.

2. A GRAPH-BASED FRAMEWORK FOR TRANSACTION DATA MINING

In general, GBAR is used mainly for visualization of the results and for extraction of confident association rules among the set of frequent itemsets in a systematic way. The main advantage of using graph data structure in this research is its ability to solve the space complexity problem because the graph uses an item as a node exactly once rather than two or more times as was done in the previous works (Vivek et al. 2010).

As discussed earlier in figure 3.2 that illustrates the research framework, there are three main steps to construct an association graph for frequent itemsets that are generated and reduced in previous phases, i.e. the preprocessing phase, the clustering phase, and dataset reduction based on clustering phase; these three steps are:

(i) Sorting frequent itemsets in each cluster in an ascending order, i.e. if the set of frequent itemsets in the first cluster – cluster of transactions that contain only one item – are {{1}, {5}, {3}, {29}, {7}}. The frequent 1-itemsets should be reordered to facilitate the construction of the association graph; they should be as following {{1}, {3}, {5}, {7}, {29}}.

(ii) The second step; which is the main step in this phase; is graph construction step. The association graph used in the research is directed, if the result after performing logical and operation (^) between the bit vector of item i (BVi) and the bit vector of item j (BVj) exceeds the minimum support threshold value assigned by the user, formally ($BVi \wedge BVj$) $\geq$ min-support, then a directed edge is drawn from item i to item j, where both i and j are frequent 1-itemsets and they are in ascending order ($i < j$).

(iii) The third and final step is the association pattern generation step, in this step, the last item of the k-frequent itemset is used to extend the frequent k- itemset into k+1 itemsets, where $k \geq 2$.
The general framework for the graph-based approach to analyze a large amount of transaction data consists of five phases (Yen & Chen 2001).

(i) Numbering phase: In this phase, all different items are assigned unique integer numbers, since dealing with numbers is much easier than dealing with all other data types for calculation of support and confidence purposes.

(ii) Large item generation phase: This phase generates large (frequent) items and records related information. A large item is an item whose support is no less than a user specified minimum support.

(iii) Association graph construction phase: This phase constructs an association graph to indicate the associations between large items.
(iv) Association pattern generation phase: This phase generates all association patterns by traversing the constructed association graph, which is much simpler and less in size than the original dataset of items.

(v) Association rule generation phase: The association rules can be generated directly according to the corresponding association patterns.

In this research, the number of phases or steps reduced from five to three as mentioned later in the previous discussion, and this in turn reduce significantly the time needed to build the graph and to extract desired association rules.

The concentration of the following example will be restricted to study the graph technique as the data items are clustered, the frequent itemsets are already generated and the dataset is minimized.
Example 1: Consider the database in Table 1. Each record is a <TID, Itemset> pair, where TID is the identifier of the corresponding transaction, and itemset records the items purchased in that transaction. Assume that the minimum support is 50 % (i.e., 2 transactions in this example).

TID	Items
10	C A D
20	E C B
30	A B C E
40	E B

(a)

TID	Items
10	A C D
20	B C E
30	A B C E
40	B E

(b)

Cluster	TID
2	40
3	10, 20
4	30

(c)

Item	Number Associated to Each item	Bit Vector
A	1	1010
B	2	0111
C	3	1110
D	4	1000
E	5	0111

(d)

Figure 1(a)Database of Transactions (b) The Database after Sorting of Items (c) Cluster Data (d) GBAR representation of Data

GBAR receives the transaction dataset in form of clusters, as shown in Figure 1 c. After the numbering phase, the numbers assigned to the items A, B, C, D, and E are 1, 2, 3, 4, and 5, respectively as shown in Figure 1 d. The frequent items found in the database are items 1, 2, 3, and 5, the item D is infrequent because its support (0.25) is less the user defined minimum support, so it is removed from further processing. From now on, the number of an item will be used to represent this item. The support for the itemset $\{i_1, i_2, \ldots, i_k\}$ is the number of 1s in BV i_1 ^ BV i_2 ^ … ^ BV i_k, where the notation ^ is the logical AND operation.

In the association graph construction phase, the Association Graph Construction algorithm (AGC) (Yen & Chen 2001) is applied to construct an association graph. The AGC algorithm is described as follows: For every two frequent items i and j. such that i < j, if the number of 1s in BV_i ^ BV_j achieves the user-specified

minimum support, a directed edge from item i to item j is created. Also, itemset (i, j) is a frequent 2-itemset. The association graph for this example is shown in Figure 2.

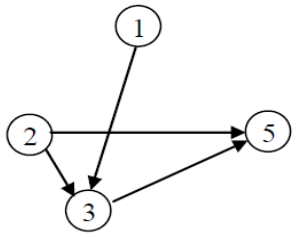

Figure 2: The association graph for example 1

The frequent 2-itemsets are generated directly after the association graph construction phase. The fourth phase, which is the association pattern generation phase, an extension should be done for large 2-itemsets generated to generate large k-itemsets (k > 2), such that the last item of the k-itemset is used to extend the large itemset into k+1 itemsets, i.e. in the aforementioned example, {{1, 3}, {2, 3}, {2, 5} and {3, 5}} are the set of frequent 2-itemsets, and they will be used as a base to generate frequent k- itemsets, where k ≥ 3.

As a rule, for a frequent itemset $\{i_1; i_2; \ldots; i_k\}$, if there is no directed edge from item i_k to an item v, then itemset $\{i_1; i_2; \ldots; i_k; v\}$ cannot be a frequent itemset. According to this rule and depending on the graph in figure 2, there is an edge from 1 to 3 and another edge from 3 to 5; So, the logical AND operation should be carried out on the bit vectors for these itemsets to check if it frequent or not, 1010 ^ 1110 ^ 0111 = 0010, as the number of 1's in the result is less than 2, there will be no frequent itemsets and the association rule generation phase no longer be required since the generation of the itemsets is terminated.

3. THE PROPOSED GBAR STEPS

Figure 3 presents the steps of the proposed graph based association rule mining (GBAR) algorithm. The GBAR algorithm takes as input the local frequent itemsets generated by the cluster based frequent itemset generation (CBFIG) algorithm. After that, a directed association sub-graph is constructed between local frequent itemsets, such that a directed edge is drawn from frequent itemsets i to frequent itemset j, where i < j, each edge in the sub-graph represents an association rule and its confidence is directly computed according to equation x.

Due to the existence of large number of rules among the local frequent itemsets, the implementation of the GBAR algorithm shows only the confident rules, i.e. those that have confidence not less than the user defined minimum confidence threshold, this simplifies counting the association rules and calculating the rules dimensionality. Then the sub-graphs merged

together, the confidence is updated for the similar edges and they are represented as one edge, to eliminate any redundancy may occur. The last step is the evaluation of the association rules generated with respect to the following measurements:

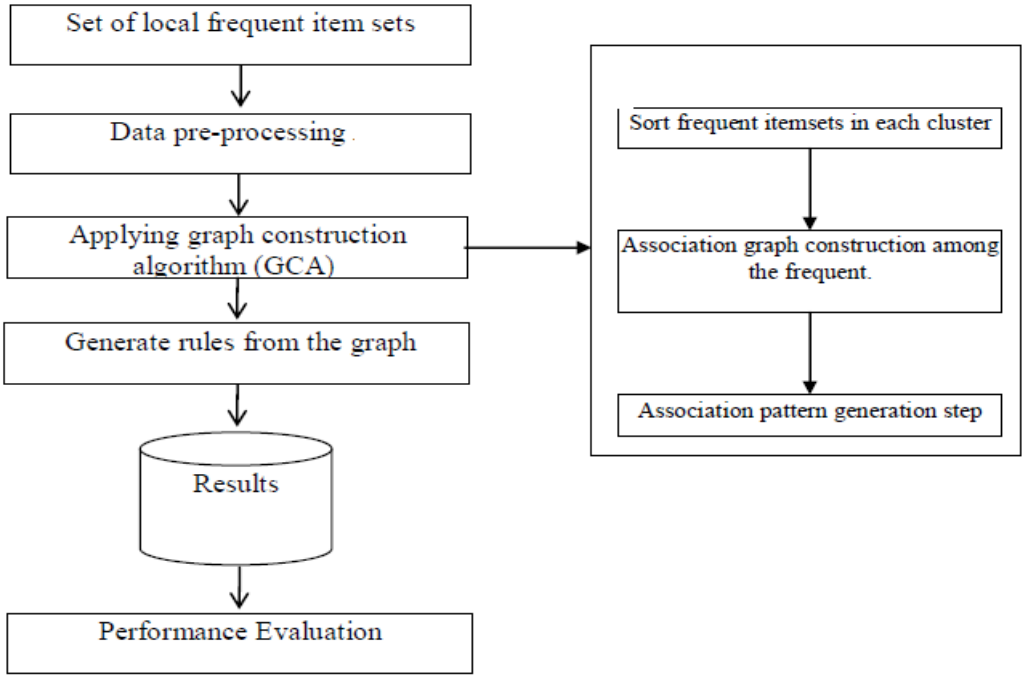

Figure 3 The proposed graph construction steps

4. COMPARISON BETWEEN THE PROPOSED GRAPHS BASED ALGORITHM AND OTHER METHODS

The good performance of GBAR over GRG algorithm comes from the fact that GRG algorithm passes over the database of transactions twice to represent each item as a bit vector while GBAR requires traverses the association graph only once to generate the association rules as mentioned earlier in this chapter.

The good performance of GBAR over RIOMining comes from the following reasons:

(i) Number of edges in the association graph constructed by RIOMining can be very large.
(ii) Frequent itemset generation by direct extension takes much more time.
The good performance of GBAR over FP-Growth graph algorithm comes from the following reasons:
(i) The number of nodes in the FP–graph equals number of distinct items in the database D. So, GBAR is better than FP-Growth graph algorithm as it deals with little number of nodes to mine the required association rules.

(ii) The FP-Growth-Graph algorithm gets bad results when the minimum support threshold is assigned small value since the data structure size rapidly increase which leads to increase the used memory space.

5. IMPLEMENTATION OF GBAR METHOD

The GBAR algorithm improves the graph based transaction rule mining algorithms as discussed in the previous section, it builds an association graph for each local frequent itemset generated by applying the proposed frequent itemset generation based on clustering (CBFIG) method, GBAR uses the data that have been gotten from the cluster matrix and frequent item matrix to draw the relationships between the frequent items.

The graphs are constructed in advance, and so GBAR is called successive graph based method for mining association rules, starting from frequent 2-itemsets in the 2-itemset transaction cluster, since the first cluster contains only 1–itemset transactions, and grows to cover all subsequent clusters. Figure 4 shows a sample association graph, which will be illustrated later, in this section. The graphs are constructed using MATLAB programming language.

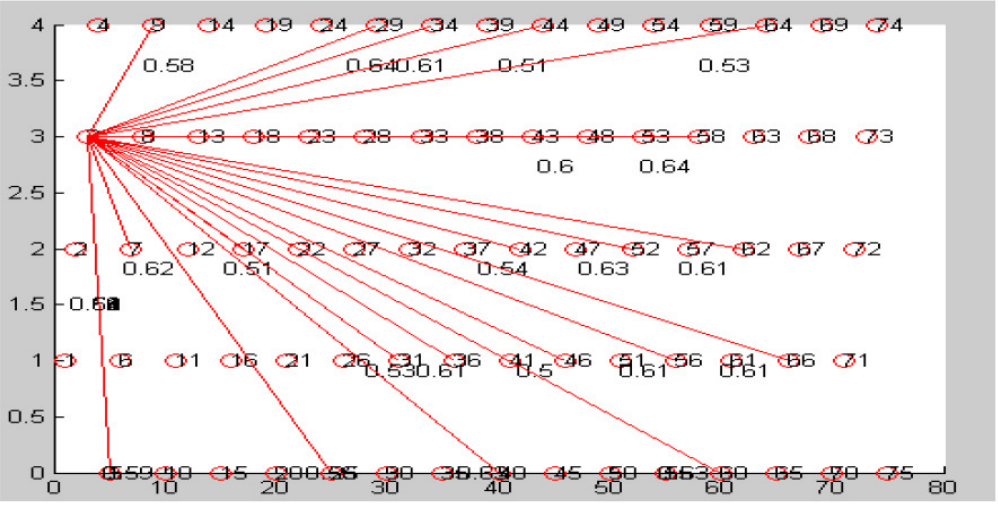

Figure 4 Association graph represents the rules associated with frequent itemset {3}

Each frequent item is represented by a node which is a circle on the graph, the graph construction is based on the matrices concept, and it looks like a 2 dimensional matrix of rows and columns, where the X-axis represents the frequent item number while the Y-axis doesn't represent any meaningful value, it is just used to separate the nodes in order to make them clear to recognize. Each edge represents an association rule where the labels of the edges are their confidence value. The directions were removed from the association graph to clarify the relations since all edges originate from the lowest item number.

According to Figure 4, the frequent itemset {3} has 19 different relations with other frequent itemsets whose positions are beyond {3}, i.e. {4} and above, these relations will be referred to as association rules, therefore, there are 19 association rules originates from the item {3}, some of these rules are:

If 3 then 7, with 0.62 confidence value
If 3 then 5, with 0.60 confidence value

It is not only impossible to take each graph as a unit and write each edge in it as a separate rule but also this operation is not rational. Thus all the graphs should be taken together to clarify all relations among different itemsets of different lengths, the goal behind drawing a graph for each frequent itemset is to achieve simplicity and avoid any overlapping or misunderstanding may occur due to the huge amount of relations.

6. THE GRAPH PRUNING IN GBAR METHOD

Constructing the transactions graph is combined with the generation of 2-itemsets and hence, it does not introduce any additional computational overhead to the graph construction process. Support pruning is next applied to the transactions graph to remove links with support below minimum confidence threshold and divide the original graph into a set of sub-graphs.

Figure 5 shows a simple illustration of the graph construction process. The shown graph is a representation of the following set of transactions {ACF, CF, ABDH, ABEI, DH, DGJ, DGJH, BEI}. The result of the previous graph construction process is a set of sub graphs each representing a subset of the transactions database. Next, the subset of the transactions database corresponding to each sub-graph is identified and the graph construction algorithm (GCA) is applied on each of these subsets individually.

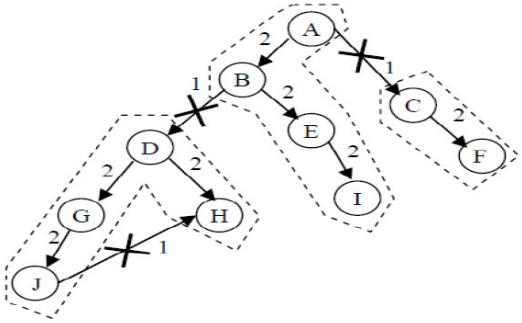

Figure 5: Transaction Graph Division (Karam et al. 2004)

GBAR starts its work after grouping the transactions into clusters, and processing each cluster to generate local frequent itemsets then global frequent itemsets going through dataset reduction, makes the proposed graph method highly efficient and scalable for almost all transaction datasets regardless to their size, this is due to the nature of market basket data where users are free to select any items together. Therefore, constructing a graph directly for market basket data will lead to an almost fully connected graph with very high degree of complexity. Pruning a fully connected graph will result in a single large sub-graph which will reduce the overhead incurred in the frequent itemsets generation step, but also, it will add an extra overhead to the process of executing the transactions graph division algorithm.

7. THE RESULTS

The results of the experiments have been recorded to be compared against the selected comparison graph based rule mining methods reported in the literature review chapter. This section reports the observations from the experiments on all ten datasets according to case studies. The description starts from the small size datasets to the medium size datasets and ends with the large size datasets. These all datasets with different sizes are included to evaluate the performance of the proposed graph based rule mining method (GBAR).

The scalability of the proposed GBAR method was evaluated by using different datasets with various sizes at the same values of minimum supports, the values of minimum support chosen in the experiments ranges from ten percent to ninety percent with 10% step, whereas the values of minimum confidence threshold ranges from 95% down to 55%, the datasets of specific size are grouped together and the average is computed and recorded in terms of all measurements used. There are two datasets used in the experiments, i.e. Chess, and Mushroom datasets. In general, the performance of GBAR is rather high comparing with the FP-Growth graph algorithm, RIOMining, and GRG algorithms at different values of minimum support and minimum confidence thresholds due to its efficiency in traversing the association graph and merging any redundant edges. However, among the comparison algorithms, it has shown better results than FP-Growth graph algorithm (Tiwari et al. 2010), RIOMining algorithm (Ning et al. 2009) and GRG algorithm (Li et al. 2003) in these datasets. FP-Growth graph algorithm remains the best among the comparison algorithms for small size datasets for the reasons listed earlier in section 3.

Table 1 shows the execution time in seconds of the proposed graph based technique (GBAR) against FP-Growth graph, RIOMining, and GRG algorithms.

Table 1: The Execution Time of GBAR Method, GRG, RioMining, and FP-Growth graph

Support	Confidence	FP-Growth graph	RIOMining	GRG	GBAR
10 %	95%	12.45	14.24	18.68	11.75
20 %	90%	13.30	12.50	16.55	10.35
30%	85%	12.1	12.45	15.93	10.18
40%	80%	11.5	12.80	15.62	9.94
50%	75%	11.25	12.00	14.86	9.32
60%	70%	10.95	11.75	14.50	9.10
70%	65%	10.72	11.65	14.32	8.84
80%	60%	9.55	11.30	13.80	7.66
90%	55%	8.23	10.90	13.15	7.02

From the observations recorded in Table 1, the execution time improved while moving from GRG to FP-Growth Graph algorithm, but the proposed GBAR algorithm still gives best results among all these techniques since it reduces the steps of mining association rules from the constructed association graph from five to three as mentioned earlier in section 3 and it doesn't take the whole items in consideration during the process of mining association rules. As noticed

from the table above, the improvement on the execution time ranges from 18.9% in case of FP-Growth Graph to 63.3% in case of GRG. On the other hand, the time decreases while the minimum support threshold increases and minimum confidence decreases due to the fact that the number of frequent itemsets is going down if the minimum support value going up and the number of rules is being lesser and lesser by increasing the value of minimum confidence threshold.

Table 2 explains the number of association rules generated and the dimension of the generated associated rules using the previous two datasets.

Table 2: The Number of Generated Rules and the Dimensionality of Rules using Chess and Mushroom datasets

Min Confidence	Min Support	# of rules generated				Dim of rules			
		FP-Growth Graph	RIOMining	GRG	GBAR	FP-Growth Graph	RIOMining	GRG	GBAR
95%	10 %	58	54	51	50	10	12	14	8
90%	20 %	57	55	50	49	11	10	13	8
85%	30 %	54	52	48	47	9	10	11	7
80%	40 %	50	49	45	44	8	11	10	7
75%	50 %	48	48	44	44	8	10	10	7
70%	60 %	45	44	42	44	7	9	8	6
65%	70 %	39	43	41	42	8	8	9	5
60%	80 %	42	41	40	39	6	8	7	5
55%	90 %	40	38	40	35	6	7	6	4

From Table 2, the proposed GBAR technique is the best among all other algorithms in terms of the number of generated rules and the average number of items per rule (dimensionality of the rules). The increase in number of generated rules in case of GRG is due to the big size of the association graph and to the huge amount of edges in it, the average number of confident association rules in case of GRG is 45 rules. FP-Growth Graph gives better results than RIOMining for the following reasons: (i) the graph size is increased because of using concept hierarchy in numbering of items in the database, and (ii) all ancestors of each item in a transaction are added to the transaction and then the FP-Growth Graph algorithm is applied on the extended transactions. These reasons lead to an increase of the number of edges in the association graph. FP-Growth Graph outperforms GRG since the numbering of items in GRG occurs level by level (from the highest concept level to the lowest concept level) and then each concept level is processed separately to generate large itemsets. The number of confident association rules increases as the minimum support threshold value decreases and the minimum confidence threshold increases. On the other hand, GBAR is the best comparing with all these comparable algorithms especially in the cases of high support values and small confidence values due the simplicity of the association graph construction in the case of primitive rules.

With respect to the dimensionality of the confident rules generated measurement, GBAR is the best as it simplifies the rules as possible by reduction of the items participating in them by – in average – 43.9% comparing with the others due to absence of the concept hierarchy in case of GBAR.

REFERENCES

[1] Tiwari, V., Gupta, S. & Tiwari, R. 2010. Association rule mining: A graph based approach for mining frequent itemsets. International Conference on Networking and Information Technology (ICNIT), 2010, pp. 309 – 313.

[2] Ning, H. 2009. Rule-chain incremental mining algorithm based on directed graph. Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09, pp. 118 – 122.

[3] Li, L. 2003. GRG: an efficient method for association rules mining on frequent closed itemsets. IEEE International Symposium on Intelligent Control, pp 854 – 859.

[4] Orlando, S., Palmerini, P., Perego, R., Lucchese, C. & Silvestri, F. 2003. kDCI: a multi-strategy algorithm for mining frequent sets. In Proc. of the Int. Workshop on Frequent Itemset Mining Implementations in conjunction with ICDM'03, pp 1 – 10.

[5] Cule, B. & Goethals, B. 2010. Mining association rules in long sequences. advances in knowledge discovery and data mining. Vol. 6118 of Lecture Notes in Computer Science. Springer, pp 300 – 309.

[6] Margahny, M. H. & Mitwaly, A. A. 2005. Fast algorithm for mining association rules. AIML 05 Conference, 19-21 December 2005, CICC, Cairo, Egypt, pp 1 – 5.

[7] Yen, S.-J. & Chen, A. L. P. 2001. A graph-based approach for discovering various types of association rules. IEEE transactions on knowledge and data engineering, vol. 13, no. 5, September/October 2001, pp. 839 – 845.

[8] Vivek T., Vipin T., Shailendra G., & Renu T. 2010. Association rule mining: A graph based approach for mining frequent itemsets. Networking and Information Technology (ICNIT), 2010 International Conference, pp. 309 – 313.

[9] Karam, O. H., Hamad, A. & Riad, W. 2004. Fast Efficient Association Rule Mining From Web Data. Fourth International ICSC Symposium on Engineering of Intelligent Systems (EIS 2004), Portugal February 29 – March 2, 2004, pp.764–770.