Basket Analysis for Graph Structured Data

Akihiro Inokuchi, Takashi Washio and Hiroshi Motoda[1]
Kouhei Kumasawa and Naohide Arai[2]

[1] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, Japan
{inokuchi,washio,motoda}@ar.sanken.osaka-u.ac.jp
[2] Recruit Co., Ltd
8-4-14 Ginza Chuo-ku, Tokyo, 104-8001, Japan

Abstract. The Basket Analysis derives frequent itemsets and association rules having support and confidence levels greater than their thresholds from massive transaction data. Though some recent research tries to discover wider classes of knowledge on the regularities contained in the data, the regularities in form of the graph structure has not been explored in the field of the Basket Analysis. The work reported in this paper proposes a new method to mine frequent graph structure appearing in the massive amount of transactions. A specific procedure to preprocess graph structured transactions is introduced to enable the application of the Basket Analysis to extract frequently appearing graph patterns. The basic performance of our proposing approach has been evaluated by a set of graph structured transactions generated by an artificial simulation. Moreover, its practicality has been confirmed through the appliaction to discover popular browsing patterns of clients in WWW URL network.

1 Introduction

The Basket Analysis derives frequent itemsets and association rules having support and confidence levels greater than their thresholds from massive transaction data [1],[2]. Some recent research of the Basket Analysis tries to discover wider classes of knowledge on the regularities contained in the data. One representative work is to introduce taxonomy of items and Boolean constraints among items under the taxonomy [3]. The association rules among items satisfying the specified constraints are efficiently derived from massive data in their approach. Another extension of the Basket Analysis on the class of the knowledge discovery is to derive association rules among continuously ordered items, i.e., sequential item patterns [4]. The taxonomy and Boolean constraints are one of the most commonly used constraints in various data analyses. The sequential data of items are also frequently observed in practical applications.

Another familiar structure and constraint of data which have not been explored in the field of the Basket Analysis are the graph structure, i.e., constraints on the uni-directional and/or bi-directional relations among nodes. The data having the graph structure are widely seen in various problem domains such as the network flow phenomena in information stream of internet, that of car traffic stream in urban areas, the parallel process streams in computer operating systems, the structure of URLs and their links in the WWW service and causality among physical states. The discovery of frequently observed graph structure from a set of given data has been researched in the machine learning area, and the most representative approach is called "GBI (Graph Based Induction)" [5],[6]. Given a set of transactions where each transaction represents a graph consisting of some nodes and links, GBI searches typical graph structures observed more than a threshold frequency in the transaction data. Another version of GBI program can discover some specific graph structures which characterize the features of nodes and/or links contained in the graph. Though GBI provides a powerful measure to figure out important graph structures from a set of given data, its basic algorithm requires a thorough search in the data to find links contained in the objective structures. Accordingly, the state of the art to mine the graph structures is not satisfactory for the really massive data.

The work reported in this paper proposes a new method to mine frequent graph structures appearing in the massive transactions. A specific procedure to preprocess graph structured transactions is introduced to enable the application of the Basket Analysis to extract frequently appearing graph patterns.

2 Transaction having Graph Structure

The transaction analyzed by the conventional Basket Analysis is a set of items. For example, the transactions for customers buying items in a grocery store are represented as follows.

$$customer_1 : \{milk, bread, butter, ---\},$$
$$-----$$
$$customer_n : \{milk, bread, apple, ---\}.$$

On the other hand, a transaction of graph structure contains nodes and links as depicted in Figure 1. Given the graph structured massive transactions, if a subgraph pattern of $\{A \rightarrow B \rightarrow C \rightarrow A\}$ appeares more than a certain frequency level, then this subgraph can be called a "*frequent subgraph*" similar to the "*frequent itemset*". Furthermore, if the transactions containing the subgraph of $\{A \rightarrow B\}$ also contains $\{A \rightarrow B \rightarrow C \rightarrow A\}$ more than a certain fraction of the transactions, then an "*association rule among subgraph structures*" of $\{A \rightarrow B\} \Rightarrow \{A \rightarrow B \rightarrow C \rightarrow A\}$ is mined from the transactions. Though the basic content of this problem is similar to the data mining of association rules among items, the conventional Basket Analysis can only discover the frequent itemsets and the association rules among the itemsets but not those of graph structures. To apply the Basket Analysis to the graph structured data, the data representation of each graph structured transaction is transformed into the form of the itemset transactions in our approch. Thus, the derivation of association rules among graph structured transactions is enable by this transformation within the framework of the conventional Basket Analysis.

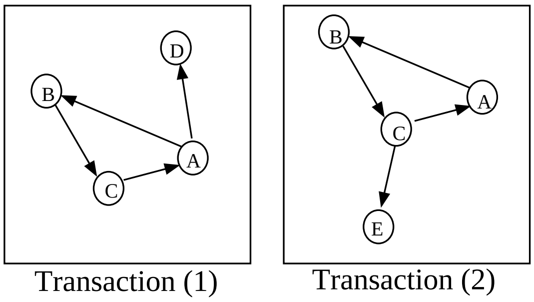

Fig. 1. Transactions containing graph structures.

The basic principle of the transformation of the graph structured transactions into the itemset transactions is as follows. Given a set of all nodes $V_{all} = \{v_1, v_2, ..., v_p\}$ and a set of all links among them $L_{all} = \{v_i \rightarrow v_j | v_i, v_j \in V_{all}\}$, then the transaction T_k having a graph structure is represented as a subset of L_{all}, i.e., $T_k \subseteq L_{all}$. As a transaction to be analyzed in the conventional Basket Analysis is a subset of all items in the data excluding the null set, the transactions having graph structures can also be analyzed in the same framework by handling each link $v_i \rightarrow v_j$ in T_k as an item.

For example, the two graph structured transactions depicted in Figure 1 can be represented as the itemset transactions as

$$\{A \rightarrow B, B \rightarrow C, C \rightarrow A, A \rightarrow D\}, \quad (1)$$

$$\{A \rightarrow B, B \rightarrow C, C \rightarrow A, C \rightarrow E\}. \quad (2)$$

By regarding each different link as a corresponding different item, the standard Basket Analysis is applicable to mine frequent subgraph structures and the association rules among the structures.

3 Implementation of Basket Analysis

3.1 Preprocessing graph structured transactions

The transactions containing graph structured data are given in various forms in practical fields. For example, the data of the network flow phenomena are usually represented as a collection of the series of nodes and links where the objects such as information packets and cars go through. In case of parallel process streams in computer operating systems, the streams of the processes and the data exchanged among the processes are represented in form of the list of the names of the processes and data together with the list of the pointers connecting these processes and the data. However, any graph data contained in a transaction can be generally transformed without much computational effort into the form of an adjaency matrix which is a very well known representation of a graph in the mathematical graph theory [7]. Each row and column of the matrix correspond to a node that appears in the graph respectively, and if a link from the i-th node to the j-th node appears in the transaction, the value "1" is assigned to the ij-element of the matrix, otherwise the value "0" is assigned.

For the first example in Figgre 1, the adjaency matrix becomes as follows. For the second, it is similarly represented.

$$\begin{array}{cc} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad \text{for the transaction (1)}$$

Once the adjaency matrix of each transaction is derived, the transaction in form of an itemset representing the graph structure is obtained by choosing each pair of nodes (i, j) having the value "1" in the ij-element and adding the arrow "$\rightarrow$" from the node i to the node j in each pair. When the transaction data contain only non-directed graph structure where the direction of each link between any two nodes is not specified, the adjaency matrix of each transaction becomes diagonally symmetric. In this case, the bar "$-$" is added between the nodes i and j, and $i - j$ and $j - i$ become an identical link.

After the data transform of the transactions have been conducted, all links appearing in all transaction are sorted and numbered in a lexicographical order for the efficient item processing similarly to the conventional Basket Analysis. In the example of Figure 1, all links are numbered as follows.

$$A \rightarrow B \equiv 1, \quad B \rightarrow C \equiv 2, \quad C \rightarrow A \equiv 3, \quad A \rightarrow D \equiv 4, \quad C \rightarrow E \equiv 5.$$

Then, the expression (1) is rewritten as

$$\{1, 2, 3, 4\}, \tag{3}$$

The frequent subgraphs and the association rules among the subgraph structures directly obtained by the Basket Analysis are also represented by the number labels of the links. For the comprehensiveness of the results, their representations are transformed back to the original links at the final stage of the analysis.

3.2 Deriving representative association rules

The standard Basket Analysis derives all frequent itemsets and all association rules having support and confidence levels greater than their thresholds, and filters out trivial rules by applying statistical heuristics [1],[2]. However, some researches have pointed out that this standard approach can not get rid of redundant rules and also misses some essential rules because of the incompleteness of the statistical heuristics used for the rule filtering [8], [9]. To alleviate this difficulty, the authors have proposed a complete logical rule filter which can retain only "*representative association rules*" [8]. The identical idea has also been provided by the other researcher [9]. The representative association rule has the characteristic to derive maximal consequences from minimal facts while maintaining their support and confidence greater than or equal to the given threshold levels. We apply this principle to derive "*association rules among subgraph structures*"

The principle to derive the representative association rules is briefly explained in this subsection. An association rule has the following form where "*Body*": B stands for an itemset and "*Head*": H another itemset (a superset of B). [1]

$$B \Rightarrow H, \text{ where } B \subset H.$$

The "*support*" values of B and H, i.e., $sup(B)$ and $sup(H)$, are ratios of the number of transactions including each set to the total number of transactions respectively. The "*confidence*" value, $conf(B \Rightarrow H)$, stands for the credibility of the rule, and is defined as a ratio of the number of transactions including H to the number of transactions including B. The Basket Analysis generates all frequent itemset having its support value greater than a threshold $l - sup$, and derives all association rules where its head is a frequent itemset and its confidence value is greater than another threshold value $s - conf$.

The association rules derived by these procedures have many redundancies that derive identical consequences from identical given facts. These redundancies reduce the comprehensiveness of the regularities discovered in the data and the efficiency of the use of those rules for some specific purposes. Instead of using statistical heuristics to remove the redundancies, we apply the following criteria.

Support threshold: The head of every association rule must have the support greater than a threshold "*lowest support*": $l - sup$.

Uniform confidence: Every association rule must have a confidence close to but not less than a level "*specified confidence*": $s - conf$.

Maximal consequence: Every association rule must derive a maximally specific consequence from a minimal fact.

The following definitions are introduced to implement these criteria.

Minimal bodyset: For a specified confidence $s - conf$, if a rule $B \Rightarrow H$ satisfies the following condition, B is said to be a "*minimal bodyset*" of $Head : H$ under $s - conf$.

$$conf(B \Rightarrow H) \geq s - conf \text{ and } conf(B' \Rightarrow H) < s - conf \quad \forall B' \subset B$$

Maximal headset: For a specified confidence $s - conf$, if a rule $B \Rightarrow H$ satisfies the following condition, H is said to be a "*maximal headset*" of $Body : B$ under $s - conf$.

$$conf(B \Rightarrow H) \geq s - conf \text{ and } conf(B \Rightarrow H') < s - conf \quad \forall H' \supset H$$

Representative association rule: For a specified confidence $s - conf$ and a lowest support $l - sup$, if $Body : B$ and $Head : H$ of a rule $B \Rightarrow H$ are the minimal bodyset and the maximal headset respectively, $B \Rightarrow H$ is said to be a "*representative association rule*".

This rule satisfies the aforementioned three criteria.

The representative association rules still contain some redundancy in terms of the inference ability. We apply a logical "*rule-filter*" where the rule $AB \Rightarrow ABR$ is removed, when two maximal estimation rules

$$AB \Rightarrow ABR \text{ and } B \Rightarrow BCR$$

---

[1] This representation of association rules is different from the standard notion $B \Rightarrow R$ where $R = H - B$. We use H instead of R for ease of our explanation.

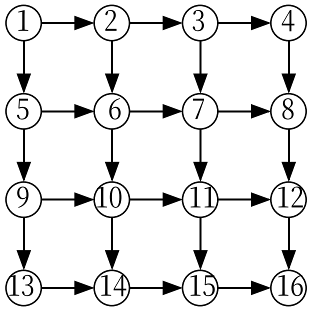

Fig. 2. A 4×4 path array.

are obtained. Here, every intersection among A, B, C, R is empty, and $AB = A + B, ABR = A + B + R$ and $BCR = B + C + R$. This rule-filtering does not violate the aforementioned criteria.

4 Performance Evaluation

4.1 Validation through simulation data

The basic performance of our proposed method to discover frequent graph structures and the association rules among the structures has been validated through the graph structured transaction data having clear characteristics. The data have been artificially generated through a Monte-Carlo simulation on a path array shown in Figure 2. This is a 4×4 link array, and vehicles starts from the node 1 to arrive at the node 16 by following directed (one way) links. Each vehicle chooses one of the links to proceed with an equivalent probability (50% each) at every binary branch of the links. A run of a vehicle from the start node 1 to the final goal node 16 corresponds to a transaction consisting of the intermediate links along the path that the vehicle passes through. The total number of the links between the adjacent nodes in this path array is 24, and the number of the items (links) involved in each transaction is 6. Furthermore, the total number of path sequences from the node 1 to 16 is 20, and the total number of the possible frequent itemsets in this example are theoretically known to be 847. The probability of each path that vehicles go through can also be theoretically evaluated. Totally, 10000 transactions, i.e., the 10000 history simulations of vehicle operations, are generated to ensure sufficient statistical accuracy of the validation. Multiple combinations of support and confidence thresholds of the Basket Analysis are applied in the validation analysis. The algorithm of a priori [1] has been used to derive frequent subgraphs. The effect of the support threshold $l - sup$ has been assessed in the range of $[0\%, 35\%]$, and that of the confidence threshold $s - conf$ has been changed in the range of $[30\%, 90\%]$.

Three examples of the frequent subgraphs discovered by the analysis are shown bellow.

$$\{1 \rightarrow 2, 2 \rightarrow 3\} \qquad\qquad (support = 25.3\%),$$

$$\{1 \rightarrow 2, 8 \rightarrow 12, 12 \rightarrow 16\} \qquad (support = 25.4\%),$$
$$\{1 \rightarrow 5, 5 \rightarrow 9, 9 \rightarrow 10, 10 \rightarrow 11, 11 \rightarrow 15, 15 \rightarrow 16, \} \ (support = 3.1\%).$$

The first example is a trivial case that its theoretical support value is easily known to be 25% because of the twice branching at the nodes 1 and 2. The second example contains two separated subgraphs of $1 \rightarrow 2$ and $8 \rightarrow 12 \rightarrow 16$. This is because many path ways from the node 2 to the node 8 exists, and the frequencies that each of the intermediate paths between the nodes 2 and 8 appear in the data are less than the support threshold $l - sup = 25\%$ in this case. The expected probability to go from the node 2 to the node 8 is 50%, and thus the total expected probability to go though these three paths are 25% which is consistent with the support value obtained in the simulation. The last example contains a full path ways from node 1 to node 16. The vehicle chooses one of the binary branching paths at the node $1, 5, 9, 10$ and 11 with equivalent probability. Thus, the expected probability to occur this path way is $(1/2)^5 = 1/32$ which is also consistent with its support value.

The followings are two examples of the association rules among subgraph structures.

$$\{1 \rightarrow 2\} \Rightarrow \{1 \rightarrow 2, 2 \rightarrow 3\} \ (support = 25.3\%, confidence = 50.9\%),$$
$$\{2 \rightarrow 3\} \Rightarrow \{1 \rightarrow 2, 2 \rightarrow 3\} \ (support = 25.3\%, confidence = 100.0\%).$$

Both of them represent the fact that a vehicle goes through the path way $1 \rightarrow 2 \rightarrow 3$ with the support value of about 25%. However, the difference of their confidence reflects the geometrical configuration of the paths $1 \rightarrow 2$ and $2 \rightarrow 3$. When a vehicle goes to the node 2 from the node 1, there are two choices to go forward at the node 2. In contrast, in the case that a vehicle goes through the path $2 \rightarrow 3$, it always should have passed the path $1 \rightarrow 2$. More complex examples reflecting the geometry of the paths are shown below where the confidence threshold $s - conf$ is set at 30%.

$$\{2 \rightarrow 6, 11 \rightarrow 15\} \Rightarrow \{1 \rightarrow 2, 2 \rightarrow 6, 6 \rightarrow 7, 7 \rightarrow 11, 11 \rightarrow 15, 15 \rightarrow 16\}$$
$$(suppoprt = 2.9\%, confidence = 50.6\%),$$
$$\{2 \rightarrow 6, 11 \rightarrow 15\} \Rightarrow \{1 \rightarrow 2, 2 \rightarrow 6, 6 \rightarrow 10, 10 \rightarrow 11, 11 \rightarrow 15, 15 \rightarrow 16\}$$
$$(suppoprt = 2.9\%, confidence = 49.4\%).$$

As easily understood by viewing Figure 2, when a vehicle goes though the paths $2 \rightarrow 6$ and $11 \rightarrow 15$, it necessarily goes though $1 \rightarrow 2$ and $15 \rightarrow 16$, and there are only the two choices $6 \rightarrow 7 \rightarrow 11$ and $6 \rightarrow 10 \rightarrow 11$ to go from the node 6 to the node 11. Accordingly, the confidence of each rule becomes around 50%. These observations indicate that the Basket Analysis for graph structured transactions in our proposed framework can properly derive the frequent subgraphs and the association rules among subgraph structures in a quantitative sense. Table 1 shows the effect of the condition of support and confidence thresholds on the computational complexity required. The level of the support threshold has a significant influence to the number of the frequent subgraphs. The computation time of "*Apriori*" is the time required to derive all frequent subgraphs, and that of "*Rulegen*" is the time to derive all representative association rules from the

Table 1. Computational complexity for support and confidence thresholds.

l-sup	s-conf	Num. of freq. subgraphs	Max. size of freq. subgraphs	Num. of rules	Comp. time[sec]	
					Apriori	Rulegen
35	90	5	2	0	0.07	0.06
	70			0		0.05
	50			2		0.05
	30			4		0.06
25	90	18	3	4	0.13	0.06
	70			6		0.05
	50			12		0.05
	30			12		0.05
15	90	54	4	12	0.24	0.06
	70			17		0.07
	50			22		0.07
	30			22		0.07
5	90	523	6	68	0.55	0.42
	70			86		0.42
	50			120		0.44
	30			130		0.45
0	90	847	6	134	0.58	0.89
	70			152		0.88
	50			179		0.89
	30			226		0.97

frequent subgraphs. The task to derive all frequent subgraphs faces the combinatorial explosion of the items for a low support threshold, while the "*Apriori*" maintains its significant efficiency due to its well organized algorithm. The computation time required by "*Rulegen*" also does not show very drastic increase. These observations are consistent with the complexity analysis for the conventional Basket Analysis [1], [8]. The increase of the maximum size of the frequent subgraphs saturates under the condition of $l - sup$ less than 15%, where the length of the full path way from the node 1 to the node 16 is 6. This is because the support of some full path ways such as $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 12 \rightarrow 16$ has the maximum value of $(1/2)^3 = 1/8$ which is slightly less than 15%.

4.2 Application to WWW browsing histories

The practical performance of the proposed method has been examined through a real scale application. The data analyzed is the log file of the commercial WWW server of Recruit Co., Ltd. in Japan. The URLs on WWW form a huge graph, where URLs are nodes mutually connected by many links. When a client visits the commercial WWW site, he or she browses only a small part of the huge graph at an access session as depicted in Figure 4, and the browsing history of the session becomes a graph structured transaction. The total number of the URLs involved in this commercial WWW site is more than 100000, and it is one

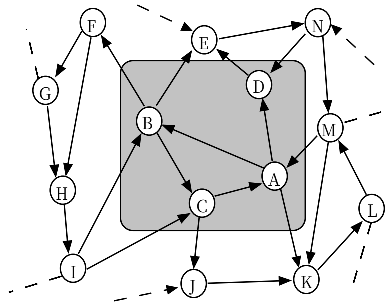

Fig. 3. A subgraph transaction in a huge URL graph.

of the largest site in Japan. Its total number of hit by the nation wide internet users always remains within the third place from the top in every month in Japanese internet record, and the typical size of the log file of the WWW server for a day is over $400MB$.

The basic format of an access record to a URL by a client in the log file is indicated in Figure 4. As the log file consists of the sequence of the access records, they are initially sorted by the IP addresses, and each subsequence having an identical IP address corresponding to the browsing access history in a session of an individual client is extracted. Then, those subsequences are transformed into adjaency matrix, and each graph structured transaction for a session of the individual IP client address are generated as explained in the earlier section.

Table 2 summarizes the statistical result of the analysis of this data by our approach varying the support threshold $l - sup$ and the confidence threshold $s - conf$. This table also shows the similar tendencies on the number of frequent subgraphs and the computation time with Table 1 that their increases are observed when $l - sup$ is decreased. In contrast, the number of the derived association rules are decreased, when $s - conf$ is decreased for low $l - sup$. This tendency is contradictory to the case of Table 1. This tendency is attributed to the feature of the WWW accesses that only the limited number of the URL access patterns are commonly shared among many clients while the access patterns of an individual client are slightly different from those common patterns. In other words, the transaction data involve many common subgraphs and the associations among them in the WWW case. For example, if the following two association rules involving common subgraphs of $A \to B$ on the lhs and $E \to F$ on the rhs are derived under a high confidence threshold $s - conf_H$,

$$\{A \to B, B \to C\} \Rightarrow \{A \to B, B \to C, E \to F\},$$
$$\{A \to B, B \to D\} \Rightarrow \{A \to B, B \to D, E \to F\},$$

then these two rules are subsumed into the following rule under a lower confidence threshold $s - conf_L$, and the above two rules are filtered out by the

IP address of a client	Δ	Time stamp of the access	Δ	URL address

Δ:space character

Fig. 4. Basic format of an access record.

Table 2. Statistics of analysis on WWW access transactions.

l-sup [%]	s-conf [%]	Num. of freq. subgraphs	Num. of rules	Comp. time[sec] Apriori	Comp. time[sec] Rulegen
0.6	90.0	132	5	151	1
	70.0		8		2
	50.0		18		2
	30.0		30		2
0.4	90.0	625	251	392	24
	70.0		186		25
	50.0		216		24
	30.0		241		25
0.3	90.0	4,568	2,292	629	486
	70.0		773		441
	50.0		107		419
	30.0		101		420

aforementioned maximal consequence principle.

$$\{A \rightarrow B\} \Rightarrow \{A \rightarrow B, E \rightarrow F\}.$$

On the other hand, the transactions generated in the path array example do not share very much common association patterns among subgraphs, since the motion of the vehicles along the directed paths are randomly determined. This feature induces the monotonic increase of the rules under the decease of $s-conf$.

Finally, we show two examples of the association rules among subgraph structures obtained in this application. Figure 5 depicts a rule representing that more than 50% of the clients who pass the link from the URL titled as "Sports" to another "Ball Game" also pass the link from the "Ball Game" to that of "Baseball". Another example shown in Figure 6 says that nearly 60% of the clients who go though the link from "Travel" to "Restaurant" also go through the path of "Restaurant" → "Hobby" → "Arts" → "Society and Culture" → "Entertainment" → "News" → "Sports". Though the lhs and the rhs of these example rules just represent the node sequences, the association rules among various types of subgraphs including branching and cyclic structures are derived. Figure 7 shows such an example including loops in the pattern. This type of knowledge derived by the proposed approach can be used to investigate the associations among the interest topics of clients of the WWW site which provides important insights for marketing on necessary services.

5 Related Work and Discussion

R. Feldman et al. applied the conventional Basket Analysis to mine associations rules among keyword subsets involved in text files such as document files and HTML files, and they proposed a method to generate keyword graphs from the association rules [10]. The graphs are generated by merging the pairwise associations among keywords involved in the rules. Though this approach can represent

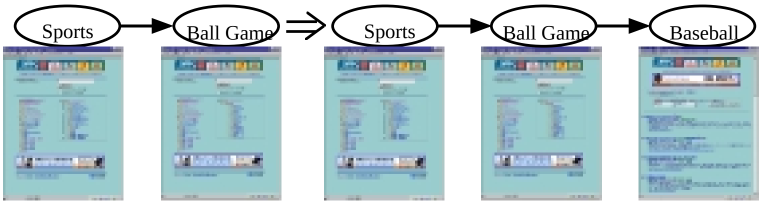

Fig. 5. Rule example (1) ($support = 0.4\%, confidence = 52.1\%$)

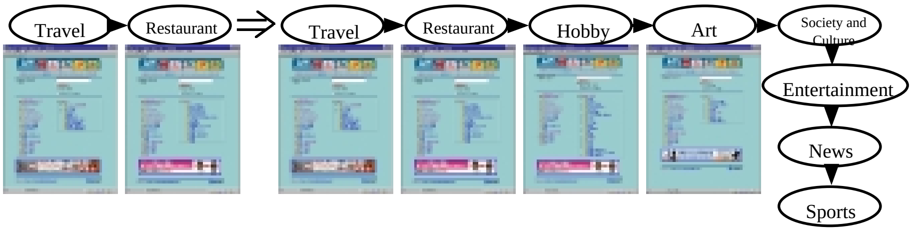

Fig. 6. Rule example (2) ($support = 0.4\%, confidence = 59.7\%$)

the associations among multiple nodes in form of graphs, it is to derive associations among sets of items and not for the applications where the transactions contain graph structured data. On the other hand, Chen et al. proposed to derive the longest access sequence patterns among URLs. Their work is close to our approch. However, knowledge representation discoverd by their framwork is limited to the access sequence patterns, whereas our apporch can discover graph structured patterns which cover wider classes of knowledge.

As shown in the previous section, the ability of our method to mine frequent subgraph structures and the associations among them are valid, and it is efficient for some practical and large scale applications. However, one weakness of our current approach is the requirement that all nodes must be mutually distinct in the object which produces the transactions. For instance, a common graph structure such as memory cell circuits contained in an LSI chip can not be discovered from the transactions representing fragments of the chip, because all nodes (devices) must be labeled by mutually different numbers, and the memory cells having an identical structure are represented by the transactions containing different links in this situation. To overcome this limitation, our approach must be extended to handle the types(colors) or attributes of nodes and links in graphs.

Fig. 7. Rule example (3) ($support = 0.3\%, confidence = 47.7\%$)

6 Conclusion

The work reported in this paper proposed an approach to mine frequent graph structures and the association rules among them embedded in massive transaction data. The approach consists of a preprocessing stage of the transaction data and the Basket Analysis. The validity of the principle we proposed has been confirmed by adopting to an artificially simulated graph structured data, and its practicality has been also demonstrated through a large scale real world problem to mine frequent browsing patterns of URLs and the associations among those patterns.

References

1. R.Agrwal and R.Srikant.: Fast algorithms for mining association rules. In *Proc. of the 20th VLDB Conference,* (1994) 487–499
2. G. Piatestsky-Shapiro.: Discovery, analysis, and presentation of strong rules. Knowledge Discovery in Databases. AAAI/MIT Press (1991)
3. R.Srikant, Q.Vu and R.Agrwal.: Mining Association Rules with Item Constraints. In *Proc. of 3rd Conference on Knowledge Discovery and Data Mining,* (1997) 67–73
4. T.Shintani and M.Kitsuregawa.: Mining Algorithms for Sequential Patters in Parallel: Hash Based Approach. In *Research and Development in Knowledge Discovery and Data Mining (Proc. of 3rd Conference on Knowledge Discovery and Data Mining),* (1998) 283–294
5. K.Yoshida and H.Motoda.: Clip: Concept Learning from Inference Pattern. In *Artificial Intelligence,* Vol.75, No.1 (1995) 63–92
6. H.Motoda and K.Yoshida.: Machine Learning Techniques to Make Computers Easier to Use. in *Proc. IJCAI97: Fifteenth International Joint Conference on Artificial Intelligence,* Vol.2 (1997) 1622–1631
7. N.Biggs.: Algebraic Graph Theory. Cambridge Univ. Press. (1973)
8. T.Washio and H.Motoda.: Mining Association Rules for Estimation and Prediction. in *Proc. PAKDD98: Research and Development in Knowledge Discovery and Data Mining,* (1998) 417–419
9. M.Kryszkiewicz.: Representative Association Rules. in *Proc. PAKDD98: Research and Development in Knowledge Discovery and Data Mining,* (1998) 198–209
10. R.Feldman, W.Klosgen and A.Zilberstein.: Visualization Techniques to Explore Data Mining Results for Document Collections. in *Proc. KDD97: Third International Conference on Knowledge Discovery and Data Mining,* (1997) 16–23
11. M.-S. Chen, J.S. Park and P.S Yu.: Efficient Data Mining for Path Traversal Patterns. In *IEEE Transaction on Knowledge and Data Engineering,* Vol.10, No.2 (1998) 209–221