Mining Association Rules in Hypertext Databases

José Borges and Mark Levene
Department of Computer Science
University College London
Gower Street
London WC1E 6BT, U.K.
Email:{j.borges, mlevene}@cs.ucl.ac.uk

RN/98/24

June 7, 1999

Abstract

Association rule techniques traditionally aim to mine information from databases consisting of a set of flat transaction records. In this work we propose a generalisation of the notion of association rule in the context of flat transactions to that of a composite association rule in the context of a structured directed graph, such as the world-wide-web. The techniques proposed aim at finding patterns in the user behaviour when traversing such a hypertext system. We redefine the concepts of confidence and support for composite association rules, which are trails of links representing a user's navigation session; the actual data may be obtained from log files.

Two algorithms to mine composite association rules are exhibited: one is a modification of the directed graph Depth-First-Search algorithm, and the other uses an incremental approach to build the set of composite rules of size $n+1$ from the set of composite rules of size n. Extensive experiments with random data were conducted in order to analyse the comparative performance of the two algorithms and to measure the sensitivity of the results to variation of confidence and support thresholds. The experiments show that, in spite of the worst-case complexity analysis which indicates exponential behaviour, in practice the algorithms complexity, measured in the number of iterations performed, is linear in the number of nodes traversed.

Keywords. Data mining, association rules, hypertext systems, trails

1 Introduction

Data Mining is an active research field whose significance has recently been increasing on account of its role as a tool to cope with the explosive growth in the amount of data being stored [CY96, FPSS96, Man97, FPSM91]. In fact, the capacity to store data has been increasing at a faster pace than the ability of the tools available to inspect and extract meaningful information from such large amounts of data. Researchers in the field of data mining and Knowledge Discovery are aiming to restrain this trend.

Knowledge Discovery (KD) can be defined as the nontrivial extraction of implicit, previously unknown, and potentially useful information from data [FPSM91]. On the other hand data mining is defined in [FPSS96] as the step in the KD process responsible for the techniques and algorithms that produce knowledge. Several data mining techniques have been being studied and among them is the problem of mining association rules in large datasets of flat transactions. The problem of mining association rules was first introduced by Agrawal *et al.* [AIS93], and has been being intensively studied since then, see for example [AMS$^+$96, MTV94, PCY95, SON95, Toi96, HF95, SA96, FMMT96, CA97]. The standard example consists of a large dataset of supermarket sales transactions. Each transaction reveals the items purchased by a particular client at the supermarket. The challenge consists in finding the items that tend to appear together in a significant number of transactions. In this context an association rule is a statement such as "90% of the customers that purchase milk also buy bread".

The explosive growth and widespread use of the world-wide-web suggests the potential of adapting the concept of association rule to the context of internet commerce, such as online bookstores or supermarkets. The success of such web-sites relies heavily on the quality of the service provided, especially since competitors on the web are just a mouse-click away, making it easy for a user to move to an alternative site if he does not find, quickly and easily, what he is looking for. Therefore, a method which analyses the user behaviour when browsing online services can be very useful both to the service provider and to the user. Knowing the user's patterns of behaviour can help the service provider to improve the design of the underlying hypertext system, providing links to the pages with the desired content.

The idea of applying association rules techniques to this context was first suggested by Chen *et al.* in [CPY96]. Therein a method was proposed to mine access patterns in an information providing environment where related documents or objects are connected by oriented links. In their approach Chen *et al.* propose the conversion of server log data into a set of *maximal forward references* and two algorithms are devised to find the set of *large reference sequences*. The difference between this problem and that of finding large itemsets is that a large reference sequence has to consist of consecutive references in a maximal forward reference, as opposed to a large itemset that is just a combination of items in a transaction.

We propose a different approach based on the generalisation of the concept of association rule rather than pre-processing the server data into a form which is then amenable to analysis by the standard association rule algorithms as in [CPY96]. In our approach the semantics of the user's navigation are maintained thus alleviating the danger for over evaluating references when the input is converted to standard association rule form. We formalise the notion of a composite association rule in the context of a directed graph such as the world-wide-web, or more generally a hypertext system [Nie90]. The weight of an arc represents the number of times the corresponding link was traversed; see the example in Figure 1. We propose to model the user's navigation session from log data converting the data into composite rules according to specified thresholds of support and confidence. While confidence corresponds to the probability of each link in a page be chosen, support takes in account the overall number of times each link was traversed. In the example of Figure 1, $H \rightarrow A \rightarrow B \rightarrow C \rightarrow D$ is a composite association rule with $confidence = 0.56$ and $support = 1.13$; and the trail $F \rightarrow G \rightarrow E \rightarrow D$ is a composite rule with $confidence = 0.5$ and $support = 1.28$. While confidence is a normalized value, in $[0, 1]$, the support is a value distributed around the average value, 1, which means that rules with support greater than 1 include links that were traversed more times than the average.

Two algorithms to mine composite association rules are proposed: one is a modification of the directed graph Depth-First-Search algorithm, and the other uses an incremental approach to build the set of composite rules of size $n + 1$ from the set of composite rules with size n. Extensive experiments with random data were carried out in order to analyse the comparative performance of the two algorithms and to measure the sensitivity of the results to variation of confidence and support thresholds. The experiments show that in spite of the complexity analysis which indicate exponential behaviour, in practice the algorithms complexity, measured in the number of iterations performed, is linear in the number of nodes traversed. Moreover, the performance of the algorithms in terms of CPU time is still exponential in the number of nodes traversed, since iterations become more complex for larger graphs.

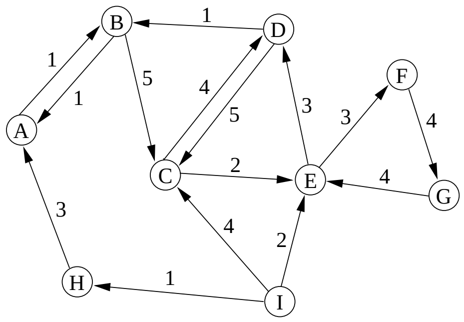


Figure 1: A weighted directed graph.

The rest of the paper is organised as follows. Section 2 provides a description of the problem, and the formal model of the problem is presented in Section 3. Section 4 then presents two algorithms for mining user patterns. Section 5 presents the experiments results, and related work is discussed in Section 6. Finally, Section 7 presents our concluding remarks.

2 Problem Description

Log files contain information about user accesses to the server, including information such as the user identification (or his IP address), the URL of the requested page and the date and time of access. This information allows us to reconstruct user navigation sessions. Having characterised user sessions we build a model that summarises all the available information and describes the user's *view* of the hypertext system. A view of the hypertext system corresponds to the subgraph of the hypertext system that was traversed by the user, therefore delineates the domain for mining user patterns.

We must keep in mind, however, that log files do not always reflect the exact user behaviour. This occurs due to the widespread use of various levels of cache, and also to the difficulty in identifying individual users' accesses as opposed to identifying the individual machines that access the server.

Definition 1 A *hypertext system* is a database of pages connected by oriented links that provide a non-sequential method to access information.

We are assuming that our hypertext system only contains elementary pages, where an *elementary page* is a page that introduces a single concept or idea, such that we cannot decompose its content without loss of meaning. We also assume that: (i) every link which refers to a page refers to the page as a whole and not to a subset of its content; (ii) the hypertext system does not contain loops, i.e., links in which the source page coincides with the target page; (iii) there is at most one link between any source and target page.

We model the hypertext system as a weighted directed graph constructed from the user's sessions log data, where nodes correspond to the visited pages and arcs to traversed links. The weight of an arc corresponds to the number of times the correspondent link was traversed. When summarising the information this way we lose the concept of individual session and the model represents a summary of a set of sessions. Therefore this model is indicated to represent the traversal information of a single user or a group of users with similar interests and navigating with the same goal.

3 Formal Model

We have a directed graph $G = (N, E)$ that consists of a set of *nodes*, N and a set of *arcs*, E. A node, $A \in N$, is the entity that corresponds to a page. An arc is a pair $(A, B) \in E$, which is the entity that corresponds to a link; for simplicity we have assumed there is at most one link between any two pages. Given link (A, B) we call A the *source* and B the *target* node. We do not assume that the graph is connected. In addition the graph is weighted and the weight of an arc expresses the number of times the user traversed the corresponding link.

The basic interaction a user can have with a hypertext system is the choice of an outgoing-link among all those available in a given page. To capture the user's preferences among the available links in a page we introduce the concept of *single association rule*. A single association rule is a statement such as "when a user reaches node A he will next choose link $A \rightarrow B$ with a certain probability". The probability of the link being chosen is defined as the *confidence* of the rule (see Definition 3).

Definition 2 A *single association rule* is an expression of the type $(A \rightarrow B)$, where $A, B \in N$ and $(A, B) \in E$, meaning that when the user reaches node A he will next choose the link (A, B), with certain confidence.

We say that a single association rule $r = (A \rightarrow B)$ holds with confidence $\mathcal{C}_r$, where $\mathcal{C}_r$ is the proportion of the traversals with A as a source node that had B as the target node.

Definition 3 $|(A, B)|$ represents the number of times link (A, B) was traversed and $\sum_{\{x|(A,x)\in E\}} |(A, x)|$ the total number of times the user traversed all the links that have A as their source node. A single association rule $r = (A \rightarrow B)$ holds with *confidence* $\mathcal{C}_r$, where $\mathcal{C}_r$ is defined as:

$$\mathcal{C}_r = \frac{|(A, B)|}{\sum\limits_{\{x|(A,x)\in E\}} |(A, x)|}$$

In order to detect the more relevant rules to the user we define the concept of *support*. In fact, a rule that holds with full confidence is relevant only if the corresponding link was traversed a significant number of times, relative to the average.

Definition 4 A single association rule $r = (A \rightarrow B)$ holds with *support* $\mathcal{S}_r$, where $\mathcal{S}_r$ is defined as the number of times the link (A, B) was traversed over the average of times all the links in the graph were traversed:

$$\mathcal{S}_r = \frac{|(A, B)|}{\frac{\sum\limits_{\{i|(x_i,x_{i+1})\in E\}} |(x_i,x_{i+1})|}{|E|}}$$

This definition leads to values of support distributed around the value 1, meaning that rules with support greater than 1 consist of links traversed more than the average. Given the probability distribution of the weights we can test the probability of having a certain level of support.

In the example of Figure 1 he have $\mathcal{C}_{(C\rightarrow E)} = \frac{2}{2+4} = 0.33$ and $\mathcal{S}_{(C\rightarrow E)} = 2/(\frac{43}{15}) = 0.70$.

We will now extend the preceding concepts to the general case, in which the rules are composed by any number of links. As we will next see, the concept of trail plays an important role in our search for patterns on the user behaviour. We now give the definition of trail according to [BH90].

Definition 5 A *trail* $(v_1, e_1, v_2, e_2, \ldots, e_{n-1}, v_n)$ is an alternative sequence of nodes and arcs such that $v_i \in N, 1 \leq i \leq n$ and every $e_i = (v_{i-1}, v_i) \in E, 2 \leq i \leq n$ are distinct.

Observation 1 As was stated before we are assuming that the graph does not contain any two links with the same source node and target node. Thus, a trail can be characterised solely by its nodes. From now on a trail $(v_1, e_1, v_2, e_2, \ldots, e_{n-1}, v_n)$ will be referred to as $(v_1, v_2, \ldots, v_n)$.

We now generalise the concept of a rule to *composite association rule*, which is a statement such as "when a user browses the hypertext system he will traverse the trail defined by the nodes $A, B, \ldots, Y$ with a certain probability".

Definition 6 A *composite association rule* is an expression of the form $[(A_1 \rightarrow A_2) \wedge \ldots \wedge (A_{n-1} \rightarrow A_n)]$ where $A_1, A_2, \ldots, A_{n-1}, A_n \in N$ and $\{(A_i, A_{i+1})|i = 1, \ldots, n-1\} \in E$, meaning that when a user traverses the hypertext system there is a certain confidence that he will follow the trail $(A_1, A_2, \ldots, A_{n-1}, A_n)$.

Definition 7 For a composite association rule $r = [(A_1 \rightarrow A_2) \wedge \ldots \wedge (A_{n-1} \rightarrow A_n)]$ *confidence* $\mathcal{C}_r$ is defined as the product of the confidences of all the corresponding single rules:

$$\mathcal{C}_r = \prod_{i=1}^{n-1} \mathcal{C}_{(A_i \rightarrow A_{i+1})}$$

Definition 8 A composite association rule $r = [(A_1 \rightarrow A_2) \wedge \ldots \wedge (A_{n-1} \rightarrow A_n)]$ holds with *support* $\mathcal{S}_r$, where $\mathcal{S}_r$ represents the average of times the links of the rule were traversed over the average of times all the links of the hypertext system were traversed.

$$\mathcal{S}_r = \frac{\frac{\sum_{i=1}^{n-1} |(A_i, A_{i+1})|}{n-1}}{\frac{\sum_{i|\{(x_i,x_{i+1})\} \in E} |(x_i, x_{i+1})|}{|E|}}$$

Observation 2 As can be easily verified when a composite association rule has only one link (i.e., it is a single association rule) Definition 4 is a special case of Definition 8.

In example of Figure 1 $\mathcal{C}_{(B \rightarrow C \rightarrow D)} = \frac{5}{5+1} \times \frac{4}{2+4} = 0.55$ and $\mathcal{S}_{(B \rightarrow C \rightarrow D)} = (\frac{5+4}{2})/(\frac{43}{15}) = 1.57$.

We now define some useful concepts which clarify description of our techniques. Our approach aims at finding all the trails in the graph with support and confidence above specified thresholds, $\mathcal{C}$ and $\mathcal{S}$ respectively. Having a trail in hand, we expand it until the values of its characteristics drop to values bellow the respective thresholds. A trail can only be expanded with neighbour links.

Definition 9 Given a trail $t = (v_1, v_2, \ldots, v_n)$, a *neighbour link* is a link that can be appended to the trail t, guaranteeing that the resulting trail t' complies with Definition 5.

Note that a trail can may cycles but every arc in the trail is distinct. Definition 9 can be specialised to two types of neighbour links as follows.

Definition 10 Given a trail $t = (v_1, v_2, \ldots, v_n)$, a *backward neighbour* is a link (x, v_1), where $x \in N$, $x \neq v_n$ and $(x, v_1) \in E$, with $(x, v_1) \notin t$.

Definition 11 Given a trail $t = (v_1, v_2, \ldots, v_n)$, a *forward neighbour* is a link (v_n, y), where $y \in N$ and $(v_n, y) \in E$, with $(x, v_1) \notin t$.

Definition 12 The *monotonic support*, $\mathcal{S}_M$, of a composite association rule is defined as the minimum value of support among all the single rules that constitute the composite rule.

Definition 12 has the advantage of potentially improving algorithm performance, since for the same support threshold value the rules will tend to be shorter with monotonic support. On the other hand, Definition 8 has the advantage of capturing rules that are good as a whole but have some of its links with support bellow the threshold.

In the example of Figure 1 trail $H \rightarrow A \rightarrow B \rightarrow C \rightarrow D$ has $\mathcal{S}_r = 1.13$ and $\mathcal{S}_{M\,r} = 0.35$. In this case if we consider a support threshold $\mathcal{S} = 1$ this trail is a rule only with the non-monotonic definition of support.

We note that the definition of confidence is itself monotonic, so when considering the monotonic definition of support every trail contained in a composite association rule is also a rule with at least the same confidence and support.

4 Algorithms for Mining Composite Association Rules

We now propose two algorithms to mine composite association rules. The first is a variation of a directed graph Depth-First-Search (DFS) and the second uses an incremental approach to recursively build the set of composite rules with n links from the set of composite rules with $(n-1)$ links. The algorithms presented implement the non-monotonic definition of support, however the extension to the monotonic definition is straightforward. The intermediate output from each of the algorithms is the set of all trails with confidence above the threshold, i.e., the candidate rules. The candidate rules that have support above the threshold are the rules.

We now explain our notation used in the algorithm description. Given a trail x, $forw\_neigh(x)$ refers to its next non-visited forward neighbour and $last\_link(x)$ to the last link in x. Given a trail x and a link e, $x+e$ denotes the concatenation of x and e. Finally $x[i]$ refers to the i^{th} link of trail x.

4.1 A Modified Depth-First Search Algorithm

This algorithm is a special case of a directed graph DFS. From each link in the graph a depth first search tree is built in which each branch corresponds to a candidate composite association rule. Each DFS tree identifies all the candidate composite association rules that have the root link of the tree as their first link. The exploration of a branch ends when it finds a node such that all its outgoing-links have already been marked as visited, or when the confidence of the corresponding trail drops to a value bellow the specified threshold. In addition each branch exploration has is own independent list of visited links, since it corresponds to a new trail.

Algorithm 1.1 (Modified_DFS(G,$\mathcal{C}$))

begin
    for each $\{e \in E : C_e \geq \mathcal{C}\}$
        Explore(e, C_e);
    end for
end.

Algorithm 1.2 (Explore($trail$, C_{trail}))

begin
    $Cand\_Rules := Cand\_Rules \cup trail$;
    for each ($fn := forw\_neigh(trail)$)
        if ($C_{fn} \times C_{trail} \geq \mathcal{C}$) **then** Explore($trail + fn$,$C_{trail}$);
    end for
end.

4.2 An Incremental Step Algorithm

This algorithm takes advantage of the principle that every subtrail contained in a trail with confidence above the threshold has itself the confidence above the threshold. This means that a rule with n links can be obtained by combining the rules with $n-1$ links. The algorithm starts with the set of trails with $n=1$ links that have confidence above the threshold $\mathcal{C}$, $Cand\_Rules_1$, and recursively builds the set of trails with $n+1$ links, $Cand\_Rules_{n+1}$, from the set of trails with n links, $Cand\_Rules_n$.

Algorithm 2 (IncrementalStep(G, $\mathcal{C}$))
begin
  for each $e \in E$
    if $C_e \geq \mathcal{C}$ **then** $Cand\_Rules_1 := Cand\_Rules_1 \cup e$;
  end for
  $i = 1$;
  repeat
    for each $r \in Cand\_Rules_i$
      for each $\{x \in Cand\_Rules_i : x \neq r \wedge (x[j] = r[j+1] : 1 \leq j \leq i)\}$
        if $C_r \times C_{last\_link(x)} \geq \mathcal{C}$ **then** $Cand\_Rules_{i+1} := Cand\_Rules_{i+1} \cup (t + last\_link(x))$;
      end for
    end for
    $i = i + 1$;
  until $(Cand\_Rules_{i+1} = \emptyset)$
end.

5 Experiments

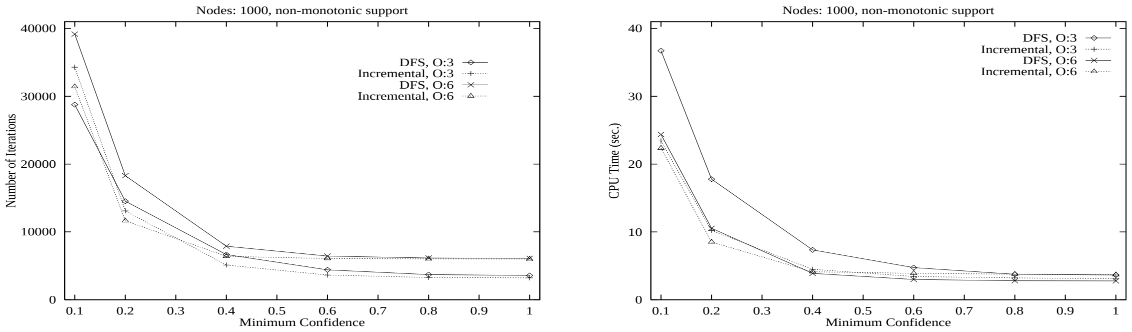
To assess the performance of the algorithms we conducted an extensive set of experiments with random data. Random data has the advantage of allowing tests with many different configurations of the model. The process of generating the random data consisted in two steps: the first was the characterisation of the hypertext system, and the second the characterisation of the user interaction with it.

To characterise the graph of the hypertext system we have to specify two parameters: the number of nodes and the average number of outgoing-links per node. The graph is randomly generated from these two parameters. In order to characterise the user interaction with the hypertext system we need to determine the weight of each arc. The weight of an arc is an integer number that represents the number of times the corresponding link was traversed. To generate these values we considered two probability distributions that are known to characterise a great variety of problems, the Uniform distribution and Poisson distribution. Several tests were performed for different values of the parameters that characterise the probability distributions and the graph.

With the DFS algorithm an iteration corresponds to the process of checking if a forward neighbour of a trail can be augmented to the current trail. With the IncrementalStep algorithm an iteration corresponds to the evaluation of a trail in the $Cand\_Rules_x$ set against another trail in the same set to check if they can form a trail for the $Cand\_Rules_{x+1}$ set.

5.1 Performance Analysis

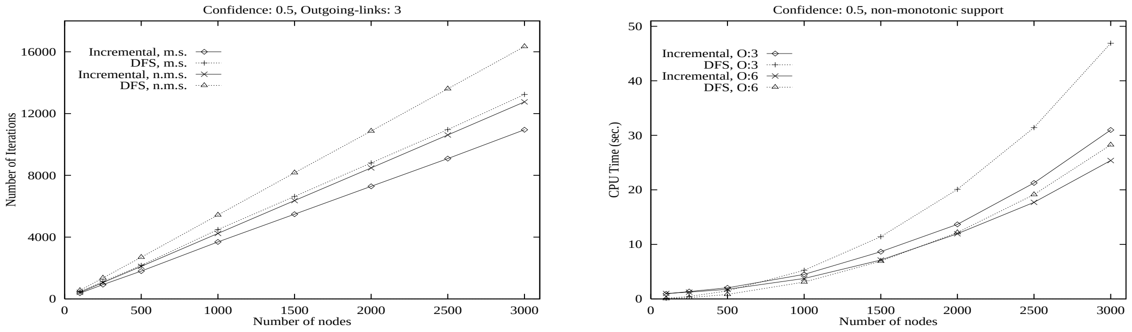
Our experiments show that the chosen probability distribution and respective parameters do not have significant influence on the performance of the algorithms. Figure 2 represents the number of iterations (2a) and execution time (2b) of both algorithms for a directed graph with 1000 nodes and two different values for the average number of outgoing-links per node (O:3 and O:6). We can see how the performance is significantly affected by the confidence threshold, especially for values below 0.4, due to a much larger number of rules and to a larger average size of the rules.



a b

Figure 2: Performance analysis for 1000 nodes and varying confidence.

Figure 3a shows that despite of the exponential worst case time complexity of the algorithms, in practice both algorithms complexity, measured in the number of iterations performed, is linear in the number of nodes. Moreover we can see that there is not a direct correspondence between the number of iterations and CPU time, since the same iteration takes longer for larger graphs due to the fact that it has to manipulate more complex data strucutures.



a b

Figure 3: Performance analysis for confidence = 0.5 and varying number of nodes.

We can also see that the monotonic support definition leads to a lower number of iterations with both algorithms. In addition we can also say that for a fixed number of nodes when the graph density increases the performance (in terms of CPU time) improves although the number of iterations also increases. This fact can be explained since the higher the graph density the more rules we have, but these have shorter length, and iterations that deal with small rules are significantly faster.

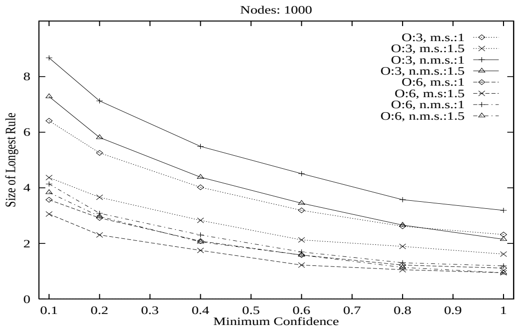


Figure 4: Maximum size of the rules for varying confidence.

In Figure 4 we show how the size of the largest rule increases when the confidence threshold decreases. We can also see how the graph density and monotonic support definition affect the rules size.

6 Related Work

Techniques for mining association rules between items in large data sets have been an active topic of research since [AIS93], Chen *et al.* [CPY96] were the first to study the application of these techniques to the problem of mining user access patterns in a web environment. In their approach the log data is converted into a set of maximal forward references, and two algorithms are proposed to determine the frequent traversal patterns from the set maximal forward references previously obtained. We think, however, that this innovative approach has two drawbacks. The first is the way backward references are discarded with the justification that they occur for user's ease of traveling. As we can see in a study which is focused on user behaviour when browsing the web [Cat95], the users of a web browser use the back command intensively, which in the study constituted 40.6% of the interactions. Moreover, log data does not provide enough information to distinguish between the use of the back command from the selection of an identical link. The second point we think deserves to be mentioned is the procedure used to build the maximal forward references. With the proposed method links closer to the root in the resulting traversal tree tend to be over-evaluated in the set of maximal forward references. For example, a link from the root to one of its child nodes will appear in all forward references passing through that child, enlarging its support, while this link may have only been traversed once.

The work by Yan *et al.* in [YJGMD96] proposes a clustering technique to automatically classify visitors of a web site according to their access patterns, and a method for dynamic link suggestion based on the obtained categories. In [SZAS97, ZAS$^+$97] the authors propose a set of tools and techniques to detect users navigation paths to the advantage of the web-site owner. A profiler responsible for keeping records of clients' selected links, accurate page viewing time and cache references, is described in [SZAS97]. In addition, [ZAS$^+$97] proposes an algorithm that clusters the users that navigate with similar interests into classes, and checks the probability of a new user becoming a member of each class. They claim that these methods can be used in capturing the characteristics of potential clients as they navigate through the hypertext system by relating the obtained classes with the answers of target questions.

Our work is also related with another emerging research field, that is the study of techniques to create adaptive web sites, or more generally, adaptive hypermedia systems [Bru96, PE97]. In [Bru96] an adaptive hypermedia system is described as a system that builds a model of each user's goals and knowledge of the system in order to adapt the page contents to the individual user characteristics.

7 Concluding Remarks

We have proposed a generalisation of the association rule concept to the context of a hypertext system, such as the world-wide-web. Our solution provides a model which captures user patterns when accessing on-line services, such as an electronic bookstore. We have formalised the concept of composite association rule, and redefined support and confidence in this context. Moreover we devised two algorithms to mine composite association rules. Experiments with random data revealed that in practice the algorithms complexity, measured in the number of iterations performed, is linear in the number of nodes of the hypertext system. The explosive growth of the world-wide-web suggests the potential of these concepts.

As future work, we are planning to conduct simulations with real data and to develop a more general model to capture access patterns between different groups of users, not necessarily with the same goal in hand.

References

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proc. of the ACM SIGMOD International Conference on Management of Data*, volume 22, pages 207–216, Washington, May 1993. ACM press.

[AMS+96] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.

[BH90] Fred Buckley and Frank Harary. *Distance in Graphs.* Addison-Wesley, 1990.

[Bru96] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction*, 6(2-3):87–129, 1996.

[CA97] Keith C.C. Chan and Wai-Ho Au. Mining fuzzy association rules. In Forouzan Golshani and Kia Makki, editors, *Proc. of the Sixth International Conference on Information and Knowledge Management (CIKM'97)*, pages 209–215, Las Vegas, November 1997. ACM Press.

[Cat95] Lara D. Catledge. Characterizing browsing strategies in the world wide web. In *Proc. Third International World Wide Web Conference*, Darmstadt, Germany, April 1995.

[CPY96] Ming-Syan Chen, Jong Soo Park, and Philip S. Yu. Data mining for path traversal patterns in a web environment. In *Proc. of the 16th Conference on Distributed Computing Systems*, pages 385–392, May 1996.

[CY96] Ming-Syan Chen and Philip S. Yu. Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):866–883, December 1996.

[FMMT96] Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Mining optimized association rules for numeric attributes. In *Proc. of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1996)*, pages 182–191, Montreal, Canada, June 1996. ACM Press.

[FPSM91] William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus. Knowledge discovery in databases: An overview. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, *Knowledge Discovery in Databases*, pages 1–27. AAAI/MIT Press, 1991.

[FPSS96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: An overview. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI/MIT Press, 1996.

[HF95] Jiawei Han and Yongjian Fu. Discovery of multiple-level association rules from large databases. In Umeshwar Dayal, Peter M.D. Gray, and Shojiro Nishio, editors, *Proc. of the 21st International Conference on Very Large Data Bases (VLDB'95)*, pages 420–431, Zurich, Switzerland, September 1995.

[Man97] Heikki Mannila. Methods and problems in data mining. In F. Afrati and P. Kolaitis, editors, *Proc. International Conference on Database Theory (ICDT'97)*, pages 41–55, Delphi, Greece, January 1997.

[MTV94] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Efficient algorithms for discovering association rules. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *Proc. of the AAAI Workshop on Knowledge Discovery in Databases*, pages 181–192, Seattle, Washington, July 1994.

[Nie90] Jakob Nielsen. *Hypertext & Hypermedia*. Academic Press, 1990.

[PCY95] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu. An effective hash-based algorithm for mining association rules. In *Proc. of the ACM SIGMOD*, pages 175–186, San Jose, USA, May 1995.

[PE97] Mike Perkowitz and Oren Etzioni. Adaptive web sites: an AI challenge. In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI'97)*, Nagoya, Japan, 1997.

[SA96] Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In *Proc. ACM SIGMOD Conference on Management of Data*, pages 1–12, Montreal, Canada, June 1996.

[SON95] Ashok Savasere, Edward Omiecinski, and Shamkant Navathe. An efficient algorithm for mining association rules in large databases. In Umeshwar Dayal, Peter M. D. Gray, and Shojiro Nishio, editors, *Proc. of the 21st International Conference on Very Large Data Bases (VLDB'95)*, pages 432–444, Zurich, Swizerland, September 1995.

[SZAS97] Cyrus Shahabi, Amir M. Zarkesh, Jafar Adibi, and Vishal Shah. Knowledge discovery from users web-page navigation. In *Proc. of the Seventh International Workshop on Research Issues in Data Engineering*, pages 20–29. IEEE Compt. Soc. Press, 1997.

[Toi96] Hannu Toivonen. Sampling large databases for association rules. In *Proc. of the 22nd International Conference on Very Large Data Bases (VLDB'96)*, pages 134–145, Mumbay, India, September 1996. Morgan Kaufman.

[YJGMD96] Tak Woon Yan, Matthew Jacobsen, Hector Garcia-Molina, and Umeshwar Dayal. From user access patterns to dynamic hypertext linking. In *Proc. of the fifth International World Wide Web Conference*, pages 1007–1014, Paris, France, May 1996.

[ZAS+97] Amir M. Zarkesh, Jafar Adibi, Cyrus Shahabi, Reza Sadri, and Vishal Shah. Analysis and design of server informative www-sites. In Forouzan Golshani and Kia Makki, editors, *Proc. of the Sixth International Conference on Information and Knowledge Management (CIKM'97)*, pages 254–261, Las Vegas, November 1997. ACM Press.