Navigation Pattern Discovery from Internet Data*

A.G. Büchner[a], M. Baumgarten[a], S.S. Anand[ß], M.D. Mulvenna[?] and J.G. Hughes[a]

[a] Northern Ireland Knowledge Engineering Laboratory, University of Ulster
[ß] School of Information and Software Engineering, University of Ulster
[?] MINE*it* Software Ltd, Faculty of Informatics, University of Ulster
email: {ag.buchner, m.baumgarten, ss.anand, md.mulvenna, jg.hughes}@ulst.ac.uk

Abstract

Electronic commerce sites need to learn as much as possible about their customers and those browsing their virtual premises, in order to maximize their marketing effort. The discovery of marketing related navigation patterns requires the development of data mining algorithms capable of discovering sequential access patterns from web logs. This paper introduces a new algorithm called M*i*DAS that extends traditional sequence discovery with a wide range of web-specific features. Domain knowledge is described as flexible navigation templates that can specify navigational behavior, as network structures for the capture of web site topologies, in addition to concept hierarchies and syntactic constraints. Unlike existing approaches, field dependency has been implemented, which allows the detection of sequences across monitored attributes, such as URLs and http referrers. Three different types of contained-in relationships are supported, which express different types of browsing behavior. The carried out experimental evaluation have shown promising results in terms of functionality as well as scalability.

1 Introduction

Direct marketing is the process of identifying likely buyers of products or services and promoting them accordingly (Ling & Li, 1998). The difference between traditional and electronic commerce marketing is the availability of more detailed data, the necessity of the incorporation of web marketing-specific domain knowledge, the potential application of more sophisticated direct marketing strategies, and, thus, the requirement of more assorted data mining goals (Mulvenna *et al.*, 1998). A key in discovering marketing intelligence in electronic businesses is that of finding navigational patterns, which can be used for online promotion and personalization activities. The objective of this paper is to propose a novel method for discovering marketing-driven navigation patterns from Internet log files.

The outline of the paper is as following. In Section 2, the structure and content of web log files is described, which is accomplished by web-specific domain knowledge, namely navigation templates, topology networks, as well known concept hierarchies and syntactic constraints. In Section 3, the algorithmic navigation pattern discovery, which has been termed M*i*DAS (Mining Internet Data for Associative Sequences) is described. In Section 4, the experimental evaluation of M*i*DAS is provided, which includes complexity measurements, as well as performance results. In Section 5, related work is evaluated, before Section 6 concludes with a summary of contributions and the outline of further work.

2 Web Data and Domain Knowledge

2.1 Web Log Files

The data available in web environments is three-fold and includes server data in form of log files, web meta data representing the structure of a web site, and marketing information, which depends on the products and services provided. For the purpose of this paper it is assumed that goal-orientated materialized views have been created a priori, for instance, as part of a web log data warehouse (Büchner & Mulvenna, 1998). Thus, this paper fully concentrates on the core activity of discovering navigational patterns in form of web-specific sequences from pre-processed log files.

The data input for M*i*DAS is a sorted set of navigations, which contains a primary key (for instance, customer id, cookie id, etc.), a secondary key (date and time related information, such as login time), and a sequence of hits, which holds the actual data values (for example, URLs).

Definition 1. A log file L is defined as a sequence of navigations $L = <N_1, N_2, \ldots, N_{|L|}>$. Each N_i is of the form (a, b, H), a representing the primary key, b the secondary key, and H a non-empty sequence of hits $H_i = <h_1, h_2, \ldots, h_{|H|}>$, where each h_i is an item which represents a single value. ♦

* This research has partly been funded by the ESPRIT project Nº 26749 (MIMIC — Mining the Internet for Marketing IntelligenCe).

2.2 Domain Knowledge Specification

In order to discover web-specific sequential patterns, domain knowledge is being incorporated, with the objective to constrain the search space of the learning algorithm, and to reduce the quantity of patterns discovered. For the purpose of discovering marketing intelligence from Internet log files, four web-specific types of domain knowledge are supported, namely navigation templates, topology networks, concept hierarchies, and syntactic constraints. Latter is expressed as the threshold sextuple $t = (s, d, ?^-, ?^+, ?^-, ?^+)$, where $s \in [0,1]$ representing the minimum support, $d \in [0,1]$ the minimum confidence, $?^-$ and $?^+$ the minimum and maximum length of a sequence, respectively, and $?^-$ and $?^+$ the minimum and maximum time gap between two hits.

2.2.1 Navigation Templates

In order to perform goal-driven navigation pattern discovery it is almost always necessary that a virtual shopper has passed through a particular page or a set of pages. This can include start, end, as well as middle pages. A typical start locator is the home page, a middle page of a site, a URL providing information about a special marketing campaign, and a regularly specified end page, where a purchase can be finalized. For simplification, all three constructs are accumulated to *navigation templates*, where a template consists of constants, wildcards, and predicates restricting the permissible values of the wildcards.

Definition 2. A navigation template t is defined as a non-empty sequence of hits, where each item h_t is either a hit or a placeholder taken from {*, ?, n}. The set of navigation templates $T = \{t_1, t_2, \ldots, t_{|T|}\}$. ♦

An example shall illustrate the specified additions to standard sequences, in order to specify regular expression constraints in the form of navigation templates. Imagine the analysis of a marketing campaign within an online bookstore, introducing reduced gifts (line 1, item 3). Only customers are of interest, which have gone through the site's home page (line 1, item 1), and only transactions that have led to purchases are to be considered (line1, item 5) at a different visit. Furthermore, the standard special offers are to be excluded from the analysis (lines 2-4). The specification is shown in Figure 1 below.

```
[
(1) <index.htm | * | /offers/gifts.htm ; * ; purchase.htm | ?>
(2) ^<* ; offers/reduced.htm ; *>
(3) ^<* ; offers/junk.htm ; *>        | same visit    ; across visits
(4) ^<* ; offers/2ndhand.htm ; *>     * wildcard      ? place holder
]                                     ^ negation
```

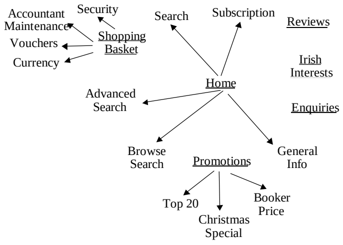
Figure 1. Example Navigation Template

2.2.2 Network Topologies

The second type of domain knowledge is that of *network* structures, which is useful when the topology of web site or only a sub-network of a large site has to be represented.

Definition 3. A network $W = (N, E)$ is a directed, cyclic graph, where $N = \{n_1, n_2, \ldots, n_{|N|}\}$ and $E = \{e_1, e_2, \ldots, e_{|E|}\}$, each n_i representing a node in W. Each e_i has the form $e_i = \{n_i, n_j\}$; $n_i, n_j \in N$. ♦

A network can theoretically be replaced by a set of navigation templates, however, navigation templates a more of a dynamic nature, whereas networks remain static over a longer period of time. An example network of a bookstore is shown graphically in Figure 2(a), where an underlined word describes a page that can be reached from any other page on the site. The textual counterpart in depicted Figure 2(b), where an asterisk connotes the set of all pages.



{*,Home}
{*,Shopping Basket}
{*,Irish Interests}
(*,Enquiries} {*,Reviews}
{Shopping Basket,Security}
{Shopping Basket,Account}
{Shopping Basket,Vouchers}
{Shopping Basket,Currency}
{Home,Search}
{Home,Subscription}
{Home,General Info}
{Home,Advanced Search}
{Home,Browse Search}
{Promotions,Top 20}
{Promotions,Christmas}
{Promotions,Booker Price}

(a) Graphical and (b) Textual Representation

Figure 2. Example Network Topology

2.2.3 Concept Hierarchies

The third type of domain knowledge that is supported are well known *concept hierarchies* (Han & Fu, 1994). In addition to marketing-related hierarchies, such as product categorizations or customer locations, a typical application is the topological organization of Internet domain levels (Büchner & Mulvenna, 1998). An example concept hierarchy *c* is depicted in Figure 3.

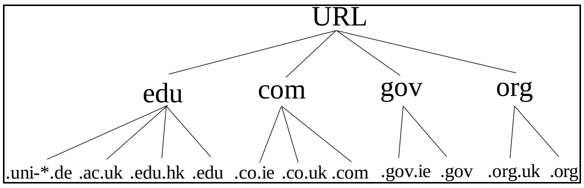


Figure 3. Example Concept Hierarchy

3 Navigation Pattern Discovery

3.1 Problem Statement and Notation

Given a log file that represents customer interactions on a web site, it is the objective to discover navigation patterns in the form of sequences, based on user-driven domain knowledge.

The log file L (as specified in Definition 1) is partitioned into P_1, P_2, …, $P_{|L|}$, where each partition can be uniquely identified by the primary key. A partition is converted into a sequence S.

M*i*DAS ← Sorted log file L, thresholds t(s, d, ?, ?$^+$, ?, ?$^+$), domain knowledge K(T, W, c)

A priori	(01)	M = {all 1-sequences}	// Input data preparation
	(02)	Map M onto c	// Concept Hierarchies
	(03)	L_T = {transformed log file}	// Data transformation
Discovery	(04)	**foreach** set S_{LT} **in** L_T **do**	// Build pattern tree P for each 1-sequence
	(05)	**foreach** *1-sequence* **in** M **do**	
	(06)	$P_{1\text{-}sequence}$ **:=** Update(S_{LT}, $P_{1\text{-}sequence}$)	// Increase frequency counter if S_{DT} exists, add S_{DT} to P otherwise
	(07)	**end**	
	(08)	**end**	
	(09)	**foreach** P_I	
	(10)	Read all *n-sequences*, with s from P_i and append them to U	
	(11)	**end**	
A posteriori	(12)	Filter out all sequences in U where $U \notin T$ and $U \notin$ W)	// Navigation Templates & Network Topology
	(13)	Delete all sequences in U which are not maximal and satisfy d, ?, ?$^+$, ?, and ?$^+$	// Pruning

M*i*DAS → U

Algorithm 1. The M*i*DAS Algorithm

A navigational pattern represents a special kind of sequence. Considering this, a navigational pattern is treated synonymously to a sequence, where a sequence is defined as follows.

Definition 4. A sequence $S = \langle s_1, s_2, \ldots, s_{|S|}\rangle$, where each s_i represents a non-empty sub-sequence $\langle h_1, h_2, ..., h_{|s_i|}\rangle$, each h being an hit. ♦

Definition 5: A sequence of length n is called an *n-sequence*, where n is the number of hits. ♦

3.2 The M*i*DAS Algorithm

The M*i*DAS algorithm consists of three major phases, which are described in the following sub-sections.

3.2.1 A Priori Phase

The first step of the a priori phase is the input data preparation, which consists of data reduction and data type substitution. Former counts the number of all item occurrences in L and excludes the hits, which have a support less than s. Latter replaces all hits in L with a hit identifier h and stores the result sequences in M. Each h_i Î M represents a unique hit and its frequency. M also represents the set of all 1-sequences, which is the basis for the pattern tree construction phase, discussed later.

The second step is to map M onto the concept hierarchy c. This further reduces the number of unique hits (see Han & Fu, 1994 for generation and refinement of concept hierarchies).

In the data transformation phase a new database L_T is created that includes only the hit identifier of attributes that are included in M. The database includes the primary and secondary key, as provided by the original log file. The transformed hits in L_T are no longer field-oriented; this information is represented through the hit identifier. However, the field independence remains, since only items of the same attribute are being allotted the same identifier (that is IP = index.html and HTTP_REFERRER = index.html have two distinct hit identifiers).

3.2.2 Discovery Phase

The pattern tree is the core element of M*i*DAS in order to discover sequences of hits. Simplified, the pattern tree is a directed, acyclic graph, where a node contains the properties of a hit and the arcs represent the relationship between two nodes. The depth *n* of a node also represents the position of a hit in an *n*-sequence. There exist two different link types for describing the relationships between two nodes. *Sequence arcs* connect two nodes that go across sub-sequences (multiple visits on a web site), and *tuple arcs*, which connect two nodes that are in the same sub-sequence (same visit). An example abstract pattern tree is shown in Figure 4.

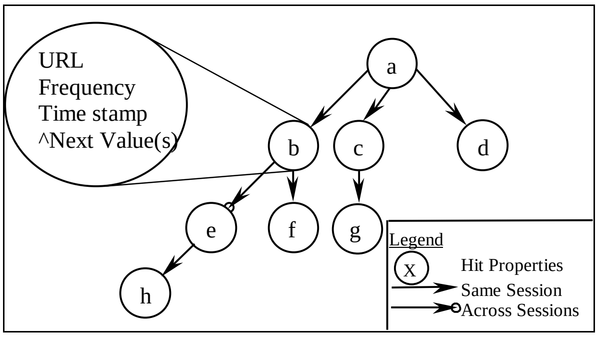


Figure 4. Abstract Pattern Tree

Finally, a set of sequences *U* is created that satisfies the confidence threshold *d*.

3.2.3 A Posteriori Phase

The first step in the a posteriori phase is to filter out all sequences that do not fulfil the criteria laid out in the specified navigation templates *T* and the topology network *W*.

The pruning phase is the last stage of the M*i*DAS algorithm. It removes all sequences that satisfy *d*, *?*, $?^+$, *?*, and $?^+$ and that are not *maximal*.

Definition 6. In a set of sequences *Q*, a sequence Q_i is *maximal* if Q_i is not contained in any other sequence Q_j, that is Q_i %**p** Q_j. ♦

M*i*DAS provides three different methods to decide when a sequence Q_i is contained in another sequence Q_j. M*i*DAS produces different kinds of result sequences, which can be *associative*, *partial* or *full*.

To describe the different pruning methods, a set of input sequences $U = \{Q_1, Q_2, Q_3, \ldots\}$ is defined. Each $Q_i = \langle H_1, H_2, H_3, \ldots \rangle$, where each H_i is a sub-sequence and declared as $H = \langle h_1, h_2, h_3, \ldots \rangle$, each h_i being a hit. If a sequence is contained in another sequence, then it is not maximal and will be removed.

Associative sequences represent patterns, which have maximal length, independent of its order. These represent visited page sequences of customers during relatively long stays. Similar methods have been applied in the context of web data in order to discover path traversal patterns (Chen *et al.*, 1996).

Definition 7. A sequence Q_i is associatively contained in another sequence Q_k (Q_i $\mathbf{p}_a$ Q_j) if each $a_j \in T_i$ (where $T_i \in Q_i$) is also element of a sub-sequence $T_j \in Q_j$. ♦

For example, ⟨(d)(a)⟩ $\mathbf{p}_a$ ⟨(a)(c d)(h f i)(b)⟩, since (d) ⊆ (c d) and (a) ⊆ (a). However, ⟨(e)(d)⟩ %$\mathbf{p}_a$ ⟨(a)(c d)(h f i)(b)⟩ because (e) ⊄ (a), (e) ⊄ (c d), (e) ⊈ (h f i), and (e) ⊈ (b).

Partial sequences are similar to their associative counterparts, but have a certain order. These patterns can be interpreted as navigational browsing behavior. The “partially contained in” relationship is identical to the “contained in” relationship proposed by Agrawal & Srikant (1995).

Definition 8. A sequence Q_i is partially contained in another sequence Q_j, (Q_i $\mathbf{p}_p$ Q_j) iff $\exists q_{2_x} \subseteq q_{1_1} \wedge \exists q_{2_x} \subseteq q_{1_2} \wedge \mathbf{K} \wedge \exists q_{2_x} \subseteq q_{1_{|Q_1|}}$. ♦

For instance, ⟨(a)(h i)(b)⟩ $\mathbf{p}_{p}$ ⟨(a)(c d)(h f i)(b)⟩, since (a) ⊆ (a), (h i) ⊆ (h f i) and (b) ⊆ (b). However, ⟨(a h)(i)(b)⟩ %$\mathbf{p}_{p}$ ⟨(a)(c d)(h f i)(b)⟩ because (a h) $\underline{\not\subset}$ (a), (a h) $\underline{\not\subset}$ (c d), (a h) $\underline{\not\subset}$ (h f i) and (a h) $\underline{\not\subset}$ (b). Generally, this means that the sequence ⟨(x)(y)⟩ %$\mathbf{p}_{p}$ ⟨(x y)⟩ and vice versa.

The difference being partial and *full sequences* is that in the latter gaps (pages which have not been visited, hence skipped) are considered as valid result.

Definition 9. The "fully contained in" relationship $\mathbf{p}_{f}$ is defined as a match of sequence Q_i in another sequence Q_k, where the location of Q_i in Q_k is irrelevant. ♦

For example, ⟨(a)(h f)(b)⟩ $\mathbf{p}_{f}$⟨(a)(c d)(h f i)(b)⟩, since (a) ⊆ (a), (h f) ⊆ (h f i) and (b) ⊆ (b). However, ⟨(a)(h i)(b)⟩ %$\mathbf{p}_{f}$ ⟨(a)(c d)(h f i)(b)⟩ because (h i) $\underline{\not\subset}$ (a), (h i) $\underline{\not\subset}$ (c d), (h i) $\underline{\not\subset}$ (h f i) and (h i) $\underline{\not\subset}$ (b).

4 Experimental Evaluation

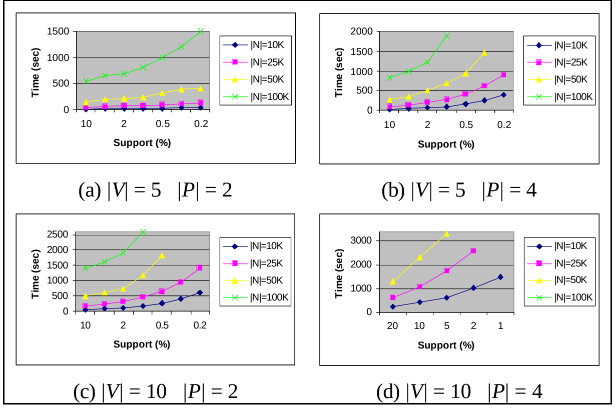
To assess the relative performance of M*i*DAS and study its scale-up properties a host of synthetic data sets has been created. The parameters of the data are shown in Table 1.

Parameter	Description
$\|N\|$	Number of visitors
$\|V\|$	Average number of visits per visitor
$\|P\|$	Average number of pages per visit

Table 1. Parameters for Synthetic Data Generation

A permutation of the three parameters has been performed such that $|N|$ has taken on the values 10K, 25K, 50K and 100K, $|V|$ 5 and 10, and $|P|$ 2 and 4, which has led to 16 data sets. The number of visits is picked from a Poisson distribution with mean μ = $|V|$, and the number of pages is picked from a Poisson distribution with mean μ = $|P|$. The number of pages per site has been set to 100, the average length of potentially large navigations to 5.

Figure 5 shows the execution times of the M*i*DAS algorithm for the generated datasets as the minimum support is decreased from 10% down to 0.2%, except for Figure 5(d) where the minimum support is decreased from 20% down to 1%. As expected, the performance decreases with lower minimum support. The increasing number of visitors shows the scale-up properties of M*i*DAS, which have shown similar behavior to existing sequential data mining algorithms without web-specific functionality.



(a) $|V|$ = 5 $|P|$ = 2

(b) $|V|$ = 5 $|P|$ = 4

(c) $|V|$ = 10 $|P|$ = 2

(d) $|V|$ = 10 $|P|$ = 4

Figure 5. M*i*DAS Execution times

The qualitative evaluation has shown that the three types of different pruning methods provide a valuable filtering mechanism in different web mining exercises. The size of the result space can be controlled through the pruning type, where associative pruning produces the least and full pruning the most navigational patterns.

5 Related Work

For the purpose of this related work section, only approaches that discover sequential patterns from web log files are considered. Detailed evaluation of generic sequence algorithms and web discovery mechanisms can be found elsewhere.

Zaïane *et al.* (1998) and Cooley *et al.* (1997/9) have created web log architectures, which support data cleansing, OLAP-like organization of data, and the application of various traditional data mining techniques, including sequences. The main obstacle of both endeavors is the non-incorporation of explicit web-specific domain knowledge. Spiliopoulou (1999) has developed a sequence discoverer for web data, which is similar to our M*i*DAS algorithm. Their GSM algorithm uses aggregated trees, which are generated from log files, in order to discover user-driven navigation patterns. The mechanism has been incorporated in a SQL-like query language (called MINT), which together form the key components of their Web Utilization Analysis platform (Spiliopoulou, *et al*., 1999).

6 Conclusions and Future Work

A new algorithm for discovering sequential patterns from web log files has been proposed that provides behavioral marketing intelligence for electronic commerce scenarios. New domain knowledge types in the form of navigational templates and web topologies have been incorporated, as well as syntactic constraints and concept hierarchies. The concept of field dependency has been introduced, which allows to represent hits from multiple attributes. Three different types of sequences are supported, which leave room for typical navigational browsing behavior on the Internet, such as skipping pages or bookmarking pages for later usage. Also, hit duplicates can be handled, as they reflect browser refresh/reload operations. Finally, all newly proposed mechanisms have been applied in a large-scale electronic commerce data mining project (Anand *et al*., 1999) and performance tests on synthetically generated data have shown promising results.

Further work in the area of discovering marketing-driven navigation patterns is twofold. First concentrates on practical issues, which include horizontal and vertical diversification of digital behavioral data (such as Web TV, Internet channels, or wireless mobile devices) and a smoother interface to a web-enabled data warehouse. Second is concerned with the improvement of the algorithmic part. In order to truly leverage the knowledge in concept hierarchies and navigation templates for providing business intelligence, domain knowledge will be incorporated into the discovery phase.

7 References

Agrawal, R. & Srikant, R. (1995) Mining Sequential Patterns, *Proc. Int'l Conf. on Data Engineering*, pp. 3-14.

Anand, S.S., Büchner, A.G., Mulvenna, M.D. & Hughes, J.G. (1999) Discovering Internet Marketing Intelligence through Web Log Mining, *Unicom'99.*

Büchner, A.G. & Mulvenna, M.D. (1998) Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining, *ACM SIGMOD Record*, **27**(4):54-61.

Chen, M.S., Park, J.S. & Yu, P.S. (1996) Data Mining for Path Traversal Patterns in a Web Environment, *Proc. 16th Int'l Conf. on Distributed Computing Systems*, pp. 385-392.

Cooley, R., Mobasher, R. & Srivastava, J. (1997) Web Mining: Information and Pattern Discovery on the World Wide Web, *Proc. 9th IEEE Int'l Conf. on Tools with Artificial Intelligence*.

Cooley, R., Mobasher, R. & Srivastava, J. (1999) Data Preparation for Mining World Wide Web Browsing Patterns, *Knowledge and Information Systems*, **1**(1).

Han, J. & Fu, Y. (1994) Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases, *Proc. KDD'94,* pp. 157-168.

Ling, C.X. & Li, C. (1998) Data Mining for Direct Marketing: Problems and Solutions, *Proc. KDD'99*, pp. 73-79.

Mulvenna, M.D., Norwood, M.T. & Büchner, A.G. (1998) Data-driven Marketing, *The Int'l Journal of Electronic Commerce and Business Media*, **8**(3):32-35.

Spiliopoulou, M. (1999) The laborious way from data mining to web mining, *Int'l Journal of Computing Systems, Science & Engineering*, March.

Spiliopoulou, M., Faulstich, L.C. & Winkler, K.A. (1999) A Data Miner analyzing the Navigational Behaviour of Web Users. *Proc. ACAI'99 Workshop on Machine Learning in User Modelling.*

Zaïane, O.R, Xin, M. & Han, J. (1998) Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs, *Proc. Advances in Digital Libraries Conf.*, pp. 19-29.