Using site semantics to analyze, visualize, and support navigation

Bettina Berendt
Humboldt University Berlin, Institute of Pedagogy and Informatics, Geschwister-Scholl-Str. 7, D-10099 Berlin, Germany, berendt@educat.hu-berlin.de

Abstract. To satisfy potential customers of a Web site and to lead them to the goods offered by the site, one should support them in the course of navigation they have embarked on. This paper presents the tool STRATDYN, developed as an add-on module to the Web Usage Miner WUM. WUM not only discovers frequent sequences, but it also allows the inspection of the different paths through the site. STRATDYN extends these capabilities: It tests differences between navigation patterns, described by a number of measures of success and strategy, for statistical significance. This can help to single out the relevant differences between users' behaviors, and it can determine whether a change in the site's design has had the desired effect. STRATDYN also exploits the site's semantics in the classification of navigation behavior and in the visualization of results, displaying navigation patterns as alternative paths through a strategy space. This helps to understand the Web logs, and to communicate analysis results to non-experts. Two case studies investigate search in an online catalog and interaction with an electronic shopping agent in an online store. They show how the results of analysis can lead to proposals for improving a Web site. These highlight the importance of investigating measures not only of eventual success, but also of process, to help users navigate towards the site's offers.

Keywords: Web usage mining, sequence mining, visualization, statistical techniques

1. Introduction

The way users navigate a Web site can be used to learn about their preferences and offer them a better interface. These preferences will often depend on the task and behavior at hand—each individual user should be offered the best interface for the course of navigation he has just embarked on.

Sequence mining is the branch of data mining investigating the temporal characteristics of Web usage (Agrawal and Srikant, 1995; Baumgarten et al., 2000; Borges and Levene, 2000; Mannila and Toivonen, 1996; Srikant and Agrawal, 1996; Wang, 1997). Going beyond the discovery of frequent sequences, the Web Usage Miner WUM (Spiliopoulou, 1999; Spiliopoulou and Faulstich, 1999) also allows the analyst to investigate the details of the user paths that constitute a result pattern. This often leads to the discovery of possibly infrequent, but interesting behavior, allowing one, for example, to track what happened to those users who did *not* eventually buy anything, and to find out possible causes for their leaving the site (Spiliopoulou et al., 2000).

However, WUM is limited in its ability to support statistical tests for significance, and it is—by design—independent of the site's semantics. The tool STRATDYN (STRATegy DYNamics), which can be used as an add-on module to WUM, was designed to overcome these two limitations. It has a more restrictive method of aggregating sequences, which allows χ^2 tests for the comparison of confidence values to be carried out. A consideration of the site's semantics helps in the automatic generation of queries for these tests. Knowledge about site semantics is usually present in an ongoing analysis of a site. This is also the basis for the generation of easy-to-understand graphics describing essential features of the way users use information.

The paper is organized as follows: Section 2 discusses pattern discovery and pattern comparison as two goals of Web usage mining that may require different analysis techniques. Section 3 introduces STRATDYN as a tool for pattern comparison. This section also describes the use of site semantics and visualization, and shows the relations to pattern discovery techniques. Section 4 presents two examples. It first gives an overview of a case study in which WUM and STRATDYN were used to analyze usage data of a large non-commercial site. This study and the associated modeling of online catalogs as a type of site are also used as a running example to illustrate the concepts introduced. The second case study investigates interaction with an electronic shopping agent in an online store. Section 5 outlines extensions of the approach to other types of sites. Section 6 concludes the paper and discusses further work.

2. Pattern discovery

Pattern discovery as a core element of Web usage mining consists of finding patterns in data. *Patterns* are instances of pattern templates, expressions in the knowledge representation scheme used for mining. For a pattern to be identified in the data, a corresponding set of sequences must be found in the data. A *sequence* is any subsequence (in the technical sense of a subset maintaining order) of a coherent subset of requests from the log file. These coherent subsets are usually individual (server) sessions (also known as visits), or multiple sessions of one user (Baumgarten et al., 2000). Each sequence belonging to the set identified must match the grammatical and/or numerical constraints in the pattern template, and the set as a whole must satisfy the numerical constraints of the specified pattern.

Numerical constraints on the set as a whole usually concern *support* (frequency of the association in the whole dataset) and *confidence* (frequency of requests for B conditional on requests for A). Sophisticated mining formalisms provide more complex pattern templates, including regular expressions to constrain node identities and path lengths between these nodes, as well as negation and temporal constraints (Baumgarten et al., 2000). The sequence miner WUM's pattern templates and visualization served as the basis for developing STRATDYN. WUM will therefore be described in more detail, and used as a reference in the introduction of formal concepts.

2.1. WUM: SEQUENCE MINING AND INTERNAL PROCESS

Knowledge discovery in WUM (http://wum.wiwi.hu-berlin.de, for details, see (Spiliopoulou, 1999; Spiliopoulou and Faulstich, 1999)) rests on three concepts. (1) The basic unit of analysis is a *sequence* as defined above, with each URL annotated by its occurrence number in the session. I.e., if one URL is requested repeatedly during a session, it is assigned occurrence numbers 1, 2, ... So a sequence is an ordered list of page occurrences: [[URL,occurrence]]. (2) A *generalized sequence (g-sequence)* consists of page occurrences and constraints $\star$ on the lengths of the path between any two subsequent page occurrences (minimum and maximum number of nodes): an ordered list of [[URL,occurrence],$\star$]. (3) All the sequences that match a g-sequence are aggregated into a tree, the *navigation pattern* (cf. Fig. 1).

A WUM query specifies constraints on a set of generalized sequences.

Example. As an example, consider the following Query 1:

```
select t
from node a b, template a * b
where a.url startswith ''SEITE1-''
and a.occurrence = 1
and b.url contains ''1SCHULE''
and b.occurrence = 1
```

This query returns navigation patterns that are constrained by the names of the start and goal pages ({a|b}.url), and the length of the path between them (here: arbitrary length *). The expression {a|b}.occurrence=1 restricts the patterns to those whose start node is the *first* request, within the session, for the URL of node a (b).[1]

The constraints on the URLs are only partial (the URL's name startswith or contains a certain string, as opposed to being equal to that string, =). This means that the query can return several navigation patterns, one for each pair of URLs that can be bound to a and b (and their occurrence numbers, which are both constant in this example). In general, this returns a *class* of navigation patterns (here: search for a school (...1SCHULE...) from the beginning of a list of schools (SEITE1...)).

Figure 1. Part of a navigation pattern.

Figure 1 shows part of a navigation pattern produced by this mining query (Spiliopoulou and Berendt, 2001). The root node is the URL bound to `a` (here: SEITE1-LALI-D). The branches are different paths to the URL bound to `b` (here: 1SCHULE-D); that URL is also the leaf of each branch. Each URL on each branch is annotated by its occurrence and by the number of users that have reached this URL along this branch. For example, the second row of Fig. 1 shows that of the 629 users who requested SEITE1-LALI-D for the first time in this session, 106 requested the page SEITE1-LASALI-D next, and for the first time in this session. This row also shows that 19 of these 106 went directly to the goal node. The fourth row shows that a further 16 of these 106 requested another page of the same kind (SEITE1-LASALI-D with occurrence = 2). The second window at the bottom left of the figure shows the aggregate statistics of the pattern.

Each of the different paths to reach the goal page, and in particular those on which many users abandoned the search, may have its own special characteristics and require a particular kind of navigation support. In the analysis of such alternatives, problems, and ideas for page re-design, Web usage mining and Web design must work hand in hand, and the visualization of the internal details of a navigation pattern is a valuable help in this cooperation.

Inspecting the pattern in Fig. 1, we found that the most frequent next step is a 'refinement' of the search, a search query with an additional search criterion specified. This is indicated in the URL name: ...LASA... instead of ...LA... This led us to also inspect the other patterns returned by the query for refinements. This interpretation of query results depends on an understanding of the pages' semantics. The visualization module aids the interpretation (and generation of new hypotheses leading to new queries) by the human analyst.

This example shows that the search for all sequences matching a given pattern template is well-suited for answering questions of the kind: What behavior occurred, in which different forms, and how often? I.e., the focus is on the *discovery* of patterns. This may imply that *comparison* of such patterns is not straightforward. For example, one may obtain the result that $a\%$ of first-time visitors of a start page A reached a goal page C, while only $b\%, b < a$, of first-time visitors of a start page B reached C. Is $b\%$ 'really' less than $a\%$, i.e., is the difference statistically significant? Such differences in proportions are commonly measured using Chi-square tests. However, the above results cannot be tested in this way. This is because *all* subsequences of a session are sequences. Different sets of sequences, e.g., those defining the different navigation patterns returned by a query, may not be disjoint. Disjointness is a necessary prerequisite for the application of Chi-Square tests. For example, a session that contains A, then B, and then C (all first-time visits) is counted towards two sets of sequences: those with start page A and goal page C, and those with start page B and goal page C. A related problem are possible overlaps. For example, $[A_1,C,A_2,D,B_1]$ and $[A_2,D,B_1,E,B_2]$, both matching $[a * b]$, may be disjoint, but they may also be overlapping parts of one session.

So while all paths from one instantiation of the start page to one instantiation of the goal page are found, and these are unique and therefore maximal relative to these instantiations, they may not be maximal relative to the pattern template, and therefore not disjoint. This maximality can be enforced by post-processing (e.g., (Baumgarten et al., 2000)). This will filter out sequences that are contained in other sequences, and therefore also overlapping sequences: $[B,...,C]$ is contained in $[A,...,B,...,C]$, and both $[A_1,C,A_2,D,B_1]$ and $[A_2,D,B_1,E,B_2]$ are, if they overlap, contained in their common supersequence $[A_1,C,A_2,D,B_1,E,B_2]$. However, such maximal sequences may not always be what the analyst is looking for. As shown in the example above, one will often be interested in paths between specific occurrences of pages, e.g., the path to the *first* occurrence of a goal page. So how can these be discovered *and* guaranteed to be disjoint?

Patterns can be guaranteed to be disjoint if the set of *sessions* is partitioned. Partitioning may be based on sequence-independent criteria, (e.g., the distinction of customers sessions containing the "payment" page from non-customer sessions not containing that page (Spiliopoulou et al., 2000)), or it may rely on sequential structure. Since this method generally filters out of some instances of the pattern in the data, it can be regarded as complementary to the methods of pattern discovery described above.

3. Pattern discovery and comparison in STRATDYN

The tool STRATDYN proposes a sequence/session classification algorithm that partitions the set of sessions. It improves on this generally applicable method, and offers new ways of visualization, by drawing on the site's semantics. These construct compact graphics, supporting the human analyst and Web designer (or another non-expert) in the understanding and comparison of navigation patterns.

A surrounding platform is needed to provide data acquisition, warehousing, and pre-processing facilities (cf. (Cooley et al., 1999; Spiliopoulou, 1999). At present, STRATDYN operates on a file of sessions generated by WUM's data preparation module.

3.1. Type hierarchies

STRATDYN uses *sessions* as the basic unit of analysis. It classifies each session as belonging to exactly one *type* from a given set of types, and returns the number of sessions belonging to each type. For example, a session containing A, then B, and then C, is only counted towards the set of sessions with start page A and goal page C.

Queries are *type templates* that identify *types* from requests for pages from a set $\mathcal{A}$, grouped by session and ordered by timestamp in a log $\mathcal{L}$.

Definition 1. A *type template* of length D is a regular expression $TT^D = [a_1, ps_1, a_2, ps_2, \ldots, ps_{D-1} a_D]$, with *nodes* a_i and *path specifications* ps_i. ps_i specifies the minimum and maximum number of arbitrary nodes that may lie between a_i and a_{i+1}.

A type template defines a set of instantiations: the *types.* They are denoted by $\tau_j^D = [A_{j1}, ps_1, A_{j2}, ps_2, \ldots, ps_{D-1}, A_{jD}]$, with one type for each distinct combination of instantiations A_i of the template's a_i.

A type τ^d defines d supertypes, with $supertype(\tau^{d'}, \tau^d)$ if and only if $\tau^{d'}$ is of length $d' < d$ and contains the first d' nodes and path specifications of τ^d. The *parent*(τ^d) of τ^d is τ^{d-1} such that $supertype(\tau^{d-1}, \tau^d)$. *subtype* and *child* are defined analogously. The root node τ^0 is the generic (super)type "any session" that encompasses all sessions.

Definition 2. A *type hierarchy* of depth D is the set of types τ^D defined by a type template TT^D that each have at least one instantiation in a log $\mathcal{L}$, joined together at their common prefixes as a tree and extended as follows:

For each type $\tau_j^D = [A_{j1}, ps_1, A_{j2}, ps_2, \ldots, ps_{D-1}, A_{jD}], d = 1, ..., D$, the tree contains exactly one path $[root, A_{j1}, A_{j2}, \ldots, A_{jd}]$ from the root node. All A_{jD} are leaf nodes.

All $A_{jd}, d = 0, ..., D-1$, have an additional child node "end" if there are sessions in $\mathcal{L}$ which (a) match the corresponding τ^d, but none of its subtypes defined by TT^D, and (b) whose last node is A_{jd}. They have an additional child node "other" if there are sessions in $\mathcal{L}$ which (a) match the corresponding τ^d, but none of its subtypes defined by TT^D, and (b) extend beyond A_{jd}. All "other" and "end" nodes are leaf nodes.

At each node $A_{jd}, d = 0, ..., D$, and at each "other" and "end" node, there is a counter giving the frequency of the corresponding type τ^d, i.e., the number of sessions in $\mathcal{L}$ that match this type.

Definition 3. A *p-type template* is a type template whose nodes are variables that match one or several pages from $\mathcal{A}$. So *p-types* and *p-type hierarchies* consist of nodes that are pages.

A p-type template corresponds to the template specified in a WUM query, except for a lack of occurrence numbers (implicitly, only the first occurrences of a node following its predecessor nodes are considered). A p-type corresponds to a g-sequence, and frequencies are the hits of a g-sequence. A p-type hierarchy consists of the g-sequences that are bindings of a template. It can be regarded as both extending and restricting the idea of a navigation pattern: (1) A type hierarchy extends the partitioning given by a navigation pattern's branches to the root node: every node is partitioned by its children. (2) Paths with "other" or "end" leaves are added. (3) Two subsequent nodes need not be subsequent in the session, but can be separated by nodes as specified in the path specification. So details of the sessions instantiating the type may be lost. A central idea of WUM is to show these details to help the analyst detect patterns that were unexpected and therefore not formulated in the query. A central idea of STRATDYN is to support the analyst by providing methods of generating type templates that formulate unexpected patterns in the query.

For example, Query 1 would produce the types *[any], [any,SEITE1-LALI-D], [any,SEITE1-*]* (for each other possible binding of `a`), *[any, other], [any,SEITE1-LALI-D,1SCHULE-D], [any,SEITE1-LALI-D, other], [any, SEITE1-*,1SCHULE-D], [any,SEITE1-*,other]*, (and analogously for each other possible binding of `b`, "end" branches omitted).

3.2. Classification and χ^2 tests

Given a type template as query, the following algorithm operates on the session log file. It classifies each session at most D times, and constructs the type hierarchy. In the pseudocode notation, `+` denotes concatenation, and `TT.d` the d^{th} node of the type template TT.

```
classify_log

Input: a log of sessions, a type template
Output: a type hierarchy

{for each session in log do
   classify(any,-,0)}

classify

Input: a current node n, a parent node n', a level d
Output: a type
Access to global variables type template, type hierarchy

{if (n does not exist){create n as child of n',freq.=0}
   frequency(n)++;              // type frequency counter
   if ( (d>D) or (n==``other'') or (n==``end'') )
     {a=[];}                              // end recursion
   else {min = min(d); max = max(d);  // path specific.
       a=0;
       for (i=0;i=min-1;i++)
         {read request;
          if (request==EOS)                // session end
            {a=[other]+[classify(other,n,d+1)];break;}}
       i=min-1;
       while (a=0) { // session can still be classified
         if (i>max) {            // path length exceeded
               a=[other]+[classify(other,n,d+1)];}
         else {read request;
          if (request==EOS) {
                a=[end]+[classify(end,n,d+1)];break;}
          else {for each instantiation m of TT.(d+1) do
                 {if (request matches pages(m))
                  {a=[m]+[classify(m,n,d+1)];
                   break;}}}} // session classified ...
         i++; }}     // ... or max. path length reached
   return a; }
```

Theorem 1. The algorithm `classify_log` is linear in the size of the log $\mathcal{L}$ for a given type template. More specifically, its worst-case time complexity is $O(L \times T)$, where L is the number of requests in $\mathcal{L}$, and T is the maximum number of different instantiations of a node of the underlying type template.

Proof: The algorithm reads each request at most once, see lines (8) and (15). If a session has been classified, but the last request used for classification is not the last request of the session, the remaining requests are read to reach the beginning of the next session, or skipped if the beginning of the next session can be reached directly. Requests to be ignored because of the path specification are only tested once for end-of-session (line (9)). For each request to be analyzed, one test is performed for "end" (line (16)), one for "other" continuations, and at most one for each instantiation of the $(d+1)^{st}$ node in the type template. Their number is bounded above by T. □

Theorem 2. The worst-case space complexity of `classify_log` is $O(min(T^D, D \times \text{no-of-sessions}))$.

Proof: Every type template node can have at most any of the T instantiations. In addition, there can be "other" and "end" nodes ending each type. This gives rise to at most $(T+2)^d$ nodes at every level $d = 0, \ldots, D$, which sums to $((T+2)^{D+1} - 1)/(T+2-1) \approx T^D$. However, each session can only be classified at most D times. If every session is different, this gives rise to $(D \times \text{no-of-sessions})$ different types. The maximal number of different types is the minimum of these. □

In general, $(D \times \text{no-of-sessions})$ is smaller than T^D. Therefore, `classify` constructs the type hierarchy dynamically. (Generated before parsing the log, it would always require T^D memory.) This also increases the flexibility of the tool in the discovery of patterns.

Limiting the size of T will limit time and space requirements. This can be achieved by specifying restrictive type templates, and by abstracting from page detail by using conceptual hierarchies to pre-process the log data.

The interaction of type hierarchy and log data may further reduce execution time. A request in a session will only be tested if it is still possible to classify that session, reducing the number of actually performed tests. Uniformity of behavior may also reduce the number of types identified.

Figure 2 shows results from performance tests that varied L, T, and D. The left hand side shows results for data from the site described in section 4.1, which is characterized by many short sessions. The data were the first 50 000, 100 000, and 500 000 requests from a log spanning two weeks during a normal time of operation. To create a conservative test, all D nodes of the type template were allowed to match any of the requests. Requests had been classified using conceptual hierarchies of different degrees of abstraction. This yielded, for the 500K log, the numbers of different pages given as T. All path specifications were

[0,0]. Runtime was confirmed to be proportional to the number of tests performed. So the figure shows the numbers of tests performed as proportions of their upper bound $L \times T$ (top half). The lower half of the figure shows the number of types identified as the proportion of their upper bound. Thicker lines are results for $D = 10$, and dashed lines indicate results for the 50K log. This shows that both time and space requirements are below their upper bounds, and that they increase with D. The time increase is a result of the [0,0] path specifications used in the tests, which imply that at most D requests are tested in one session. The figure also shows that relative time requirements decrease with T. In another test (not shown), more restrictive type templates were used, with T smaller than the number of distinct requests in the log. This led to similar space requirements and considerably lower time requirements (≈ 0.2).

The right hand side of Figure 2 shows results for data from the site described in section 4.2, which is characterized by fewer, but longer sessions. For $L = 35586$ and $T = 10$ as specified by the conceptual hierarchy used, D was varied. The figure shows that users' relative uniformity of behavior further limited time and memory requirements.

Figure 2. Performance test results: time (no. of tests performed / $(L \times T)$), space (no. of types identified / (min (T^D,($D\times$ *no-of-sessions*)).

The purpose of the classification of sessions is to find the values of *measures* for types. A measure describing a property of a type τ^d is the *confidence* of a pattern: the frequency of a subtype $\tau^{d'}$ of τ^d, divided by the frequency of τ^d. Typical measures are the *popularity* within a parent type, i.e., the frequency of all subtypes divided by the frequency of their parent type, and various *conversion rates*, i.e., confidences of patterns given by the number of sessions visiting the goal page divided by the number of sessions that earlier visited the start page (Berthon et al., 1996; Spiliopoulou and Berendt, 2001).

The frequencies of types to be analysed further define a table with one row per analysed type $\tau^i, i = 1, \ldots, I$, and one column for each subtype $j = 1, \ldots, J$. Each cell (i, j) contains the absolute frequency of subtype ij. Cell $(i, (J+1))$ contains the absolute frequency of type i. Cell $((I+1), j)$ contains the absolute frequency of subtype j. Cell $((I+1), (J+1))$ contains the absolute frequency of the parent type of types $i = 1, \ldots, I$.

These frequencies are tested against the null hypothesis of equal distribution (if $J = 1$), or independence of the distributions of the types (rows) and subtypes (columns). The latter means that expected frequencies in each cell are defined by the product of the marginal frequencies. To investigate more specific hypotheses, the table is partitioned in the required ways (Bresnahan and Shapiro, 1966) to compare single cells (e.g., is the conversion rate of page A higher than that of page B?, comparing types ij and kj), or groups of cells (e.g., compare the conversion rates of pages $A_1, A_2, \ldots$ with those of pages $B_1, B_2, \ldots$). For each comparison, a χ^2 and a p are computed to determine whether or not the difference is statistically significant.

In a typical data mining situation without prior hypotheses, these individual comparisons will typically be true post hoc analyses investigating possible differences between consecutive categories of types. This means that alpha error corrections, e.g., Bonferoni corrections, should be performed (Bortz, 1993). In subsequent analyses, designed to test hypotheses gleaned from datasets used in a first analysis, the paired comparisons may test specific a priori hypotheses, which can obviate the need for alpha error corrections (Bortz, 1993).

3.3. Site semantics: strategy spaces

Site semantics denotes the formal description of the 'meaning' of a site's different Web pages. Classifying a site's pages allows the analyst to identify a larger number of user sessions or episodes as instances of one general pattern, as opposed to the very specific paths individual users take through the site. This is useful in a wide range of analysis tasks: identifying association rules between purchases of goods, determining differences between site designers' goals and users' actual behavior, identifying semantically meaningful navigation episodes, improving the interface, and characterising the workload of a site (cf. (Cooley, 2000) for an overview and a number of case studies).

An ontology describing the site may depend on the domain and typical page content, as in Craven et al.'s (Craven et al., 2000) example of computer science departments: these have faculty, staff, courses, etc. Classification by page content is also typical in market basket analysis (Berry and Linoff, 1997). The ontology may depend on page structure and function, as in the (Cooley et al., 1999) distinction between "head" pages serving as entry points for a site, "navigation" pages containing many links and little information, "content" pages containing a small number of links and designed to be visited for their content, "look-up" pages with many incoming links and few outgoing ones, usually providing a definition or acronym expansion, and "personal" pages with very diverse characteristics and no significant traffic. An ontology based on a domain event model is proposed by (Menascé et al., 1999). They distinguish typical activities performed in an e-commerce site: browse, search, select, add to cart, and pay. Given an ontology, page classification may be done (semi-)automatically, employing machine learning algorithms that operate on properties of the HTML code and its vocabulary. Here, Web usage mining is joined by Web structure and Web content mining.

The *service-based conceptual hierarchies* proposed in (Berendt and Spiliopoulou, 2000) classify dynamically generated pages based on user queries in a form-based site. In Web usage mining, as in data mining in general (Han and Kamber, 2001), conceptual hierarchies are often indispensable for abstraction, in particular if database accesses generate a large number of different Web pages (see also (Baumgarten et al., 2000)).

Example (continued). Typical online catalogs, e.g., online bookstores, have pages that can be described along 3 dimensions: (1) *entity class* (e.g., books, CDs, etc.), (2) *level of detail*, typically *top-level* (main page or starting page for a specific entity class), *list (of individuals)*, and *individual*, and (3) *specifiable property.* The specifiable properties are usually search criteria. They can be described by their values (e.g., "author", "title", "author+title"). A more abstract description only distinguishes *how many* properties are specified. When long results are split, *first pages* can be distinguished from *further pages.*

We have implemented a data pre-processing module that classifies Web log data according to these conceptual hierarchies. The task can be semi-automated by parsing URLs for specific parameters as search criteria or page number specifications, for file system path names indicating entity classes, etc. Thus, we were able to use concepts of different granularity in the regular expressions provided as templates (cf. the example query in section 2.1—the "URL"s used there are in fact names of higher-level concepts).

A user's request for a page can be viewed as a *strategy* taken to reach or come closer to the goal of navigation.[2] The page delivered by the Web server will influence the subsequent strategy: pages are catalysts for *strategy modification.* For example, a user may refine a database query, or select an alternative of a chosen product. In general, different pairs of strategies may represent the same modification. For example, "author+title" is a refinement of both "author" and "title", and "author+language" is another refinement of "author".

Example (continued). The structure of common search interfaces can be described as follows: From a search initiated by the specification of n search criteria to obtain page_1, the *first page* of a *list* of individuals of some entity class, a user can take one of 7 kinds of next steps in requesting page_2. Each of these operations can be detected by parsing the two pages. In the first five of the following cases, entity class(page_2)=entity class(page_1); in cases 2–5, the level of detail(page_2)=*list* :

1. *goal*: reach an individual page. This is the shortest search. level of detail(page_2)= *individual*.
2. *continuation*: request a *further page* of the same list. specifiable properties(page_2)= specifiable properties(page_1), position(page_2)= *further page*.
3. *repetition*: initiate a new search with the same search strategy, but probably different parameter values. page_2= page_1.
4. *refinement*: add further search criteria to narrow the search. specifiable properties(page_2)=specifiable properties(page_1) plus further ones, position(page_2) = *first page*.
5. *strategy change*: initiate a new search with a different search strategy. position(page_2)= *first page*, but neither of type "repetition" nor "refinement".
6. *end* leave the site. page_1 is the last node of the session.
7. *other* do something else. This covers all remaining sessions.

For a given level of detail in the conceptual hierarchies, the Web site has a set of pages $\mathcal{A}$ and a set of modifications $\mathcal{M}$ that are binary relations on $\mathcal{A}$. A strategy space with k dimensions consists of points $Points = \{P_1, P_2, P_3, \ldots\} \subseteq R^k$, and a set of binary relations Rel on $Points$, where R is the set of reals. The site's semantics consists of a strategy space, together with one-to-one mappings from $\mathcal{A}$ to $Points$, and from $\mathcal{M}$ to Rel.

Let $P[l]$ be P's l^{th} component. An example of a relation on P is *is-more-specific-than*$(P_1, P_2) \leftrightarrow ((P_1[l] > P_2[l]) \wedge (\forall l' \neq l : P_1[l'] = P_2[l']))$, with l identifying the dimension "number of specified properties". The relation "refinement" on pages is mapped to *is-more-specific-than* on points. In other words, regardless of the (common) entity type of two pages, a move along the dimension "specifiable properties" from a set of properties to a larger set of properties can be regarded as a refinement.

3.4. Strategy dynamics

How can site semantics be used to generate type hierarchies? Modifications can be expressed as types. Given a query to investigate searches, i.e., types starting with a first list page, STRATDYN can (1) identify the distinct first list pages by matching to create the nodes at the first level, and then (2) generate the second level (the 7 next steps, or some grouping of them, for each node at level 1 define the types at level 2). With I list types or *search strategies* and $J \leq 7$ *follow-up strategies*, this gives rise to an $I \times J$ table of frequencies.

However, a *course* of navigation may not be exhaustively described by the first follow-up step. For example, there may be a group of users who *continue* looking at the same list for many steps, while others immediately (and repeatedly) *refine.*

To capture such behavior, it is useful to extend the definition of type templates. *Strategy modification sequences* that describe a course of navigation are sequences like *[refinement, new search, goal]*, or *[refinement, continuation, end]*. This gives rise to the following

Definition 5. An *m-type template* is a type template whose nodes are variables that match one or several page variables or one or several modification variables, and whose path specifications are all [0,0].

m-types and *m-type hierarchies* consist of nodes that are pages from $\mathcal{A}$ or modifications from $\mathcal{M}$.

An m-type can be rewritten in terms of pages: The strategy modification sequence $[M_1, M_2, \ldots, M_D]$ can be rewritten as a number of p-types: Each $[A_1, A_2, \ldots, A_D]$ such that $M_d(A_{d-1}, A_d)$ for all $d = 1, \ldots, D$. If a strategy modification sequence contains a strategy A_i on its own, this is treated like a node in a p-type.

Rewriting is implemented by extending step (19) of the algorithm to a match with the `pages(m,request')`, with `request'` being the previous request.

How does this affect the algorithm's performance? Given a previous request, each possible strategy can only count towards at most one modification. So the number of tests performed on one request is bounded above by the number of distinguished strategies, which is an upper bound on T in the original version of the algorithm.

STRATDYN takes D, the number of navigation steps to be examined, as input, and generates the m-type template $[s_1, m_1, m_2, \ldots, m_D]$, with s_1 an arbitrary initial strategy, and each m_i instantiable by any of the considered strategy modifications. The input log is then parsed, and the m-type hierarchy generated. This defines $I \times J^{d-1}$ types at level d of the hierarchy, $d \geq 1$. For these, an ever-increasing number of χ^2

tests can be performed. Alternatively, they can be further abstracted and visualized in a low-dimensional state space to aid the process of interpretation.

3.5. Stratograms: visualizing strategy dynamics

Stratograms are diagrams that plot the requested pages against time, with connecting lines showing the transitions, vary the lines' thickness to indicate frequencies, and use a *meaningful* ordering to arrange the pages along the y axis. This allows the analyst to interpret the position and/or direction of points and lines, an idea inspired by the "proofograms" of (Oberlander et al., 1996)).

Stratograms are based on type hierarchies, which provide the temporal order of steps as well as the frequencies of types. They augment this information by employing a linear order on pages or requests and assigning a numerical *strategy value* to them.

A stratogram displays a transition from a page or request A to B at a given distance d from the start node as a line between their strategy values. This transition may have been made in sessions of different types, i.e., on different branches of the type hierarchy. To show the total frequency of this transition, these frequencies have to be added. Stratograms concentrate on transitions contained in the type template and do not display "other" continuations. They display "end"s in special ways. This gives rise to the following

Definition 6. A *stratogram* for a type hierarchy with depth D and frequencies $frequency(\tau)$ is defined as

$$
\begin{aligned}
& strat = \langle v, tr, \theta_1, \theta_2 \rangle \quad \text{with} \\
& \quad v : \{\tau\} \mapsto R \\
& \quad\quad \text{s.t. } v(\tau^d) = \infty \text{ for } \tau^d.d = \text{“end”}, v(\tau^d) = -\infty \text{ for } \tau^d.d = \text{“other”}, \\
& \quad tr = \{\langle d, v_1, v_2, pr(v_1, v_2, d)\rangle \mid d = 0, \ldots, D-1, \\
& \quad\quad pr(v_1, v_2, d) = \sum_{\{\tau^{d+1} \mid v(parent(\tau^{d+1})) = v_1 \wedge v(\tau^{d+1}) = v_2\}} \frac{frequency(\tau^{d+1})}{frequency(\tau^0)},
\end{aligned}
$$

where the θ are support thresholds.

A *stratogram visualization* consists of (1) for each d, v_1, v_2 s.t. $v_2 = \infty$ and $pr(v_1, v_2, d) \geq \theta_1$: a circle with center (d, v_1) and radius increasing with $pr(v_1, v_2, d)$, and (2) for each other d, v_1, v_2 s.t. $v_2 \neq -\infty$ and $pr(v_1, v_2, d) \geq \theta_2$: a line from (d, v_1) to $(d+1, v_2)$, with thickness increasing with $pr(v_1, v_2, d)$.

The strategy value of a page A can be the value of the k^{th} component of the point in strategy space that A is mapped to.

For example, in the analysis of search dynamics, pages are ordered by *information concreteness*. This measure is derived from the site's semantics in the following way: A list page with n specified parameters is assigned a value of n, and a page describing an individual is assigned a value of the maximum number of specifiable parameters plus 1.

In this case, $v(\tau^d)$ reduces to $v(\tau^d.d)$, a function mapping pages to numerical values. ($\tau^d.s$ is the s^{th} node of τ^d.) Stratograms can also take the temporal order of requests into account. In the following example, information concreteness is combined with temporal order as $v(\tau^d) = v(parent(\tau^d)) + f(\tau^d.(d-1), \tau^d.d)$. f is a function of the kind of strategy modification performed, and $v(\tau^0)$ is fixed.

Figure 3. Stratograms showing search dynamics in an online catalog.

Example (continued). Figure 3 shows typical stratograms for individual strategies (maximal number of specified parameters 3; 5 strategy modifications plotted). Continuations and repetitions are shown as lines with a smaller slope: 0.1 upwards (repetitions), and 0.1 downwards. (To avoid misinterpretations, the parameters must be chosen small enough to ensure that values will remain close to the strategies' start values.) The figure shows two different types: 'horizontal' stratograms (a) and (b), and 'vertical' ones (c). Horizontal stratograms are characterized by a long process of remaining at the same level of concreteness, i.e., strategy modifications that are continuations or repetitions. In vertical stratograms, the degree of concreteness increases quickly, usually refining or even reaching the goal in the first step. The strategy modification "new strategy" can lead to movements in arbitrary directions.

This can be interpreted as follows: users choosing strategies that produce horizontal stratograms (1) would rather reach their goal quickly, but are not aware of the existence of more restrictive search options, or (2) enjoy getting an overview of the range of offers of the site. Users choosing strategies that produce vertical stratograms, on the other hand, appear to be oriented mainly towards getting a specific information on an individual entity. This suggests offering support to users: For example, upon invocation of a horizontal strategy modification, additional help messages or links may be offered to explain the search interface, and to give an overview of the range of the site's offers.

Stratograms describe Web navigation in cognitive dimensions, and they complement WUM's visualizations by respecting two basic principles of the human reading of diagrams. First, magnitudes that are ordered in the diagram tend to be interpreted as ordered in reality (Berendt et al., 1998). In stratograms, both x and y axis have meaning, unlike in Fig. 1, where the y axis has no meaning. Second, an 'identity of place' is useful to interpret a diagram. However, if pages are mapped to only one place (e.g., (Chi et al., 2000; Cugini and Scholtz, 1999)), the 'progress' in a path can only be seen indirectly by tracing a sequence of arrows. Stratograms rely on the alignment technique (Card et al., 1999) that associates one page with one y coordinate or neighborhood of y coordinates.

4. Case studies

4.1. Search behavior in an online catalog

SchulWeb ("School Web", http://www.schulweb.de) is a form-based site offering, among other services, a database of more than 8000 schools searchable with a number of search criteria and input modes. The site offers typical online catalog functionality: browsing and searching for information on a selection of real-world entities. These collections of information are the goods offered by the site, and the site's goal is to lead users quickly and efficiently to all the information they need. The data and their preparation are described in (Berendt and Spiliopoulou, 2000; Spiliopoulou and Berendt, 2001).

The site's semantics were analyzed as described in section 3.3. In this site, the specifiable properties for the entity class "school" are *country*, (federal) *state* within that country, *school type*, and *textual property*, a typed-in search string to be matched against school name, city name, or webmaster's name. We concentrated on the entity class "school" and searches for schools in the country Germany, because these constitute the main use of the site. We analyzed the follow-up steps, the *conversion efficiency over all paths* (how many of the sessions of a type reached a school URL eventually), and the *conversion efficiency over minimal paths* (how many of the sessions of a type reached a school URL in the next step) (Spiliopoulou and Berendt, 2001).

The popularity measure showed large differences within the 1449 school search sessions (out of a total of 6059 in the whole log), see Table I. No state+string searches were found. The null hypothesis of equal distribution could be rejected ($\chi^2_5 = 667.17, p < 0.01$).[3]

Conversion efficiency over all paths was between 51% (state searches) and 58% (type searches), with no significant differences between strategies. However, the conversion efficiency over minimal paths differed, and also the general distribution of next steps. Table I shows the percentages. (By definition, refinement is impossible in state+type+string searches.) The null hypothesis of independent distributions could be rejected ($\chi^2_{30} = 402.35$, $p < 0.01$). Partitioning the table and computing χ^2 for each column separately showed that each measure was distributed non-equally. These results were significant at the 0.01 α level.

Table I. Search strategies: Popularity and distribution of next steps (percent): Sch(ool), Rep(etition), Cont(inuation), Ref(inement), Strat(egy) ch(ange). Note: state = LALI in Fig.1,state+type = LASALI in Fig.1.

Strategy	Popularity	Sch.	Rep.	Cont.	Ref.	Strat.ch.
state	37	10	7	22	26	0
type	7	15	8	25	25	2
string	7	29	21	3	9	7
state+type	27	29	9	28	5	4
type+string	9	27	21	2	2	8
state+type +string	11	36	25	1	0	11

Further analysis of groupings showed that the input mode for the search criteria may account for these differences. Strategies which required typing a search string ("specification-based" strategies: string, type+string, state+type+string) exhibited significantly lower proportions of continuations and of refinements than those which required choosing from a menu or clickable map ("choice-based" strategies: state, type, state+type). The conversion efficiency over minimal paths was also higher for specification-based strategies. All results were significant at the 0.01 α level. Specification-based strategies had vertical stratograms (e.g., Fig. 3 (a): string search), while choice-based strategies had horizontal stratograms (e.g., Fig. 3 (b), (c): state and state+ type searches). Differences for stratograms were significant except for "end of session" (significant at $\alpha = 0.05$ for "3-parameter search", and at $\alpha = 0.01$ for all other next steps). In each stratogram, intended usage according to the 3-click rule would be a line connecting the starting point along the y axis with a school reached along the minimal path, i.e., point (1,4). This is actually the most common path taken in the searches shown in Fig. 3 (b) and (c), but not in the search shown in Fig. 3 (a).

The two search options state and state+type together account for far over half of all searches ($\chi_1^2 = 113.2; p < 0.01$). This highly uneven distribution of popularity led us to conclude that a search by location is needed by many users. However, the search option *state* is not very restrictive, leading to long lists and long search episodes. Also, we know from e-mail contacts that many of the site's users are interested in information on schools in their present or future neighborhood, i.e., usually within a specific *city*. Using this combination of Web mining results and background knowledge, we proposed to better support search by location. Design was changed to highlight the possibility to search by typing in a city name (plus optionally choosing the school type).

The statistical tests employed to test difference within one log can also be applied in comparisons between different logs. This can be used to assess the effect of a change in design. Evaluating the effect of these changes made to the SchulWeb site, (Spiliopoulou and Pohle, 2001) found a marked increase in the use of the more efficient specification-based strategies.

Further analysis showed that there was a change in the relative popularity of the combination strategies "type+string" and "state+type+string". While their combined popularity remained constant, the popularity of "type+ string" rose, while that of "state+type+string" fell ($\chi_2^2 = 10.88, p < 0.01$). This can be interpreted as follows: It has become more obvious to users that to search by location, they can directly specify a location as a "city" by typing, instead of via choosing a state. So a number of searches that involved the specification of three search criteria could be 'replaced' by searches that involved only the specification of two search criteria, making the interface easier to use. Refining the concept "string" in the conceptual hierarchy, we found that over 80% of all requests for pages with string specifications used the search parameter "city". The proportion was particularly high for type+string searches, confirming the initial interpretation of the substitution of "city" for "state". A qualitative analysis of a sample showed that the large majority also intended to search by city. Opportunities for further improvement were signaled by the observation that on average, around 30% of parameter values did not fit the chosen parameter, e.g., a school name was entered for the parameter "city", or vice versa. This problem is currently addressed in the context of a site relaunch.

4.2. Shopping in an online store with a shopping agent

The Web site used in the second case study is an online store developed at the Institute of Information System of Humboldt University Berlin, in cooperation with a German retail chain. After selecting a product category and going through an introductory phase, users are encouraged to answer up to 56 questions related to the product they intend to purchase. This communication is initiated by an anthropomorphic shopping agent. At any time, the agent can be asked to determine the current top 10 products out of the shop's offers, based on the user's preferences as stated in the answers given so far. From the top 10 page, information on each product can be obtained. From there, further information is available, and the product can be placed into the shopping cart and purchased. From the top 10 page, users can also go back to continue answering questions, or to revise given answers. Exit without purchasing is possible at any time. Apart from the questions tailored to the product category and the products on offer (but parallelized according to sales strategy, see (Annacker et al., 2001)), the shopping interface is identical for different products.

The present analysis used the navigation data of participants of an experiment in which one of two product categories could be chosen, compact cameras and winter jackets. Buying decisions were binding for participants (for details, see (Spiekermann et al., 2001) and http://iwa.-wiwi.hu-berlin.de).

The site's pages were classified[4] as follows: "Q categories" is a survey page containing seven groups of questions of different content, visible to the user. For the analysis, the questions were classified into four categories ordered by decreasing relatedness to the product and judged by independent raters as decreasingly legitimate and relevant in the shopping context: desired product attributes (pd), aspects of envisaged usage (u), personal questions related to the product (pepr), and personal questions unrelated to the product (peip). This manipulation was an intentional part of the shopping process designed to find out in how far shoppers let themselves be involved in a communication that elicits privacy-relevant information (Annacker et al., 2001). The remaining pages were ordered by increasing closeness to a decision for a product: "top 10", "product info", "more product info", "product info with purchase option", and "purchase". This gives rise to an order on pages defined by *closeness to the product*, increasing from top (maximum values) to bottom (minimum values). Movements made by less than 10% of users, and exits made by less than 1%, were filtered out to improve visual clarity. A maximum number of 347 steps was investigated.

Figure 4. Shopping for a search good (top) and an experience good (bottom) (347: max. no. of steps displayed).

Figure 4 shows the unexpected result that two phases emerged in user behavior: a communication phase (top left) and an information phase characterized by product inspection (bottom right). Instead of using the freedom afforded by the online shopping environment, users closely followed the order of questions suggested, letting themselves be drawn into an increasingly personal dialogue. This is shown by the relatively few, thick lines at the top left.

The figure also shows that these phases' distinctness changes with product category: The upper part of Fig. 4 shows navigation of users in the "camera" section of the store (152 sessions). The lower part of the figure shows navigation of users in the "jackets" section (50 sessions).

A camera is a typical *search good*: a decision for a particular camera is highly dependent on the careful examination of a comparatively large number of technical details. A jacket is a typical *experience good*: a jacket must be worn by the customer, and ideally, it should be 'experienced' before it is bought. Search goods and experience goods are associated with different kinds of risk. While financial and functional risk are generally perceived as higher for search goods, socio-psychological risk is generally perceived as higher for experience goods. Questionnaire data confirmed that this was also the case for the camera and jacket shoppers in this experiment. An underlying aim of site design had been to create a shopping agent that not only addressed, as most Web sites do, financial and functional risk, but also socio-psychological risk (Spiekermann and Parachiv, 2000). For example, the agent not only asked whether shoppers wanted to restrict attention to jackets in a particular price range, and what jacket size they needed, but also whether they liked to attract attention to themselves. (Similar questions were designed for cameras.) Since no brand information was available to reduce perceived risk, and the price ranges of the two products were matched, a question of the experiment had been how the different kinds of perceived risk would influence navigation behavior, and in how far the agent could help in addressing these risks.

The navigation paths taken suggest that camera shoppers tried to counter their perceived financial and functional risk by closely following the agent's suggestions (highly uniform paths at the top left), and that this information was considered to be sufficient for a buying decision (comparatively short information phase, bottom right). Statistical analysis showed that conversion efficiency was higher for cameras (55%) than for jackets (24%) ($\chi^2_1 = 14.745, p < 0.01$). In particular, conversion efficiency over short paths was higher (35% vs. 10%, $\chi^2_1 = 11.368, p < 0.01$). Paths were classified as "short" if they were shorter than half the maximal length of purchasing sessions.

Jacket shoppers, on the other hand, relied more on their own information gathering: Jackets were inspected already during the communication phase (see bottom left), and answers corrected, resulting in a longer communication phase. Also, in the information phase, "more product info" was requested more often (see bottom right), and the information phase lasted longer. More exits occurred, in particular before purchase was completed (see the circles at y values other than "purchase"). Statistical analysis confirmed that jacket shoppers were more likely to spend a 'long' time in the information phase than camera shoppers (types distinguished by number of strategy modifications remaining within the information section, classified as 'long' if over 100, $\chi^2_2 = 12.906, p < 0.01$).

These results suggest that the online store investigated may be more suited to selling search goods than experience goods. It appears that the agent achieved a satisfactory communication about financial and functional risks, but did not manage to address socio-psychological risk adequately. Users relied more on inspecting product description and photographs, possibly as a surrogate for real experience like touching the fabric, examining the fit, etc. However, as the long history of mail-order clothing sales shows, many customers are willing to enter the risk of buying an experience good before experiencing it, and the anonymous setting may bear distinct advantages. Corresponding marketing instruments may contribute to the success of online sales. Also, site design may still have been too static and too much focused on an itemized description of product characteristics. Lastly, dynamic adaptations of the site may help customers to not get lost in long information phases.

A second relevant finding of this experiment is a null result: Although most participants had professed to be concerned about online privacy in a questionnaire, and had been warned of a subsequent use of their navigation data, they all disclosed an alarming amount of personal information. There were hints of specific strategy modifications: the (more or less frequent) transition from any communication page to the top 10 page. This user-initiated interruption of communication to request and explore the current top 10 recommendations signaled caution. However, the differences in the frequencies of these navigation types between user groups with different privacy concerns were not significant. These results call for an 'education in privacy consciousness' involving the analysis of navigation paths, and recommendations on how to avoid losing control in the flow of online communication.

5. Modeling other types of sites

The type hierarchy classification as such is general-purpose and can be used for any type of site (Srivastava et al., 2000). Stratograms can in principle also be used with any ordering of pages, and thus any site (cf. the visualization of individual navigation histories in (Jones and Berger, 1995)). However, the modeling of sites in terms of strategy spaces is a core element in the automated generation of type hierarchies as well as in the interpretation of stratograms.

A number of other classifications of pages also lend themselves to this approach. For example, "head", "navigation" and "content" pages (Cooley et al., 1999) can be modeled and ordered like the "top-level", "list", and "individual" pages investigated in the present paper.

The approach can be generalized to sites with different ontologies. Partial orders can be used, as long as a sufficient number of entities are ordered. For example, the pages in the e-commerce site event model in (Menascé et al., 1999) could be ordered in a similar way as in the second case study above, since there is a canonical direction towards the (site owner's) goal: browse/search, select, add to cart, and pay. (Cf. the process of book purchase illustrated by icons arranged from left to right in sites like amazon.com.) "Browse" and "search" could be aggregated, or placed in an arbitrary order. However, a larger number of unordered entities may pose problems. For example, moves between pages on different types of staff in (Craven et al., 2000) are difficult to interpret spatially. Here, common techniques for dealing with nominally scaled variables should be employed.

Other models of navigation behavior that imply total or partial orders on pages are based on information retrieval, search, or information foraging theory. Search engines rank pages based on their relevance to a user query. Google (Page et al., 1998), for example, combines an ordering by hyperlink citation frequency that is independent of the specific query with an ordering based on keywords matching the user query. Ordering by hyperlink frequency is related to the categorization by page structure as described above, and it could be investigated as an alternative way of defining the usage-independent part of ordering. Moreover, strategy values could be defined as depending also on the *parameter values* of user queries, extending their current definition that only considers the *parameters* of user queries. The ensuing stratograms would show an order of pages based also on their value in terms of the initial query's content.

While these approaches derive an order from the site or from a query, the reverse direction of inference may also be taken. At each stage of a search, there are an (information) goal and a number of cues that allow the user to evaluate the next options reachable via hyperlinks. Chi, Pirolli, and Pitkow (2000) call this "information scent": the "imperfect, subjective, perception of the value, cost, or access path of information sources obtained from proximal cues, such as Web links, or icons representing the content sources" (p. 162). So as in search engines, content and structure must be examined in conjunction to understand and improve search. Assuming that users proceed towards their goals, Chi et al. trace clustered user paths in the reverse direction to infer what information scent was followed. In a last stage, this information is presented to a human analyst, who can identify and name the information goal. To better understand usage and its relation to designer intentions, data-driven method like this could be used as complementary to the hypothesis-driven method presented here.

6. Conclusions and outlook

The two tools described in this paper, STRATDYN and WUM, discover knowledge that not only concerns aggregate statistics of various patterns of Web navigation, but also the paths taken through the site. This allows more insights into the process of navigation, helping site designers to better support individual users by creating pages that adapt to users' chosen strategies. The usefulness of integrating the site's semantics in the classification of navigation behavior and in the visualization of results has been emphasized. Adding these techniques to WUM has allowed us to evaluate them in the context of a Web mining testbed. It has also allowed insights into the complementarity of pattern discovery and comparison. Further work will extend the techniques presented in this paper.

Given a session beginning with types $\tau^1, \tau^2, \ldots$, the conversion rates obtained from past data could be used to predict likely continuations. This could be used to, e.g., generate and offer personalized Web pages (cf. (Mobasher et al., 2000)). For analysis, prediction, and user modeling, the information gained from an analysis of usage paths can be related to user-specified data such as demographic attributes (cf. the case study in section 4.2). It can also be related to variables like cognitive and learning style (Berendt and Brenstein, 2001).

Chi-square tests are applicable across a wide range of data mining tasks (cf. (Brin et al., 1997), who have extended association rule mining in market basket analysis with Chi-square tests). Extending the two-dimensional nature of χ^2 contingency tables, log linear modeling allows the formulation of higher-dimensional contingency tables. This allows the investigation of higher-order interactions such as changes in the distribution of follow-up strategies over time, cf. (Olson et al., 1994). (Borges and Levene, 2000) model behavior in terms of N-grams of different lengths and use χ^2 tests to identify changes in model accuracy. Translating this to the current context, one could determine the value of D that gives the best tradeoff between model size and accuracy.

Another current research direction aims at extending the generality of patterns and pattern matching. The current focus on the *first* possible rewrite could be augmented by the identification of the *best* possible rewrite, and type hierarchies could be extended to implement context-free expressions (Fu, 2001). Also, the current classification method that rests on regular expressions will be combined with other statistical methods to improve the analysis of numerical information like the frequency of a certain event within one session.

Page abstraction using conceptual hierarchies has been proposed as the major method of dealing with the influence of the number of distinguished pages on algorithmic complexity. Further methods will be investigated to also allow more fine-grained analysis and representation, allowing higher numbers of distinguishable pages.

Lastly, we are working on developing stratograms further. One research direction concerns extensions of their way of representing complex navigation histories, including graphical abstraction. Another direction concerns functionality: In their current form, stratograms emphasize the presentation of data in a concise format to allow perception 'at one glance', as one 'image' unit (Green, 1998). Animation and interactivity will be explored to extend their expressive power.

Acknowledgements

I wish to thank Myra Spiliopoulou, Sarah Spiekermann, Elke Brenstein, and the SchulWeb team for discussions and other work that have made this research possible, and my reviewers for valuable comments and questions. I wish to thank Humboldt University Berlin for funding my research, and DFN-Verein and FWU for funding SchulWeb.

Notes

[1] WUM queries can also constrain node support and pattern confidence, but this is not relevant for the present argument and therefore omitted.

[2] To what degree this appears to be 'strategic', rational, purposeful behavior will depend on user and analyst. But even apparently aimless surfing through a Web site can be seen as a strategy with the goal of entertainment.

[3] Statistical significance is reported with respect to the common α levels of 0.05 and 0.01, although most differences had much lower p values. All reported multiple comparisons used Bonferoni corrections.

[4] The automated classification used the URL stems of the site's dynamic pages.

References

Agrawal, R. and Srikant, R. 1995. Mining sequential patterns. Proceedings of the International Conference on Data Engineering. Taipei, Taiwan, March, pp. 3–14.

Annacker, D., Spiekermann, S., and Strobel, M. 2001. Private consumer information: a new search cost dimension in online environments. Proceedings of the 14^{th} Bled Conference. Bled, Slovenia, June 2001.

Baumgarten, M., Büchner, A.G., Anand, S.S., Mulvenna, M.D., and Hughes, J.G. 2000. User-driven navigation pattern discovery from Internet data. In (Spiliopoulou and Masand, 2000), pp. 74–91.

Berendt, B. and Brenstein, E. 2001. Visualizing individual differences in web navigation: STRATDYN, a tool for analyzing navigation patterns. Behavior Research Methods, Instruments, & Computers.

Berendt, B., Rauh, R., and Barkowsky, T. 1998. Spatial thinking with geographic maps: an empirical study. H. Czap, H.-P. Ohly, and S. Pribbenow (Eds.), Herausforderungen an die Wissensorganisation: Visualisierung, multimediale Dokumente, Internetstrukturen (Proceedings ISKO'97). Würzburg, Germany: ERGON-Verlag, pp. 63–74.

Berendt, B. and Spiliopoulou, M. 2000. Analysis of navigation behaviour in web sites integrating multiple information systems. The VLDB Journal, 9: 56–75.

Berry, M.J.A. and Linoff, G. 1997. Data Mining Techniques: For Marketing, Sales and Customer Support. New York: John Wiley & Sons, Inc.

Berthon, P., Pitt, L.F., and Watson, R.T. 1996. The World Wide Web as an advertising medium. Journal of Advertising Research, 36: 43–54.

Borges, J. and Levene, M. 2000. Data mining of user navigation patterns. In (Spiliopoulou and Masand, 2000), pp. 92–111.

Bortz, J. 1993. Statistik für Sozialwissenschaftler. 4^{th} rev. ed.. Berlin: Springer.

Bresnahan, J.L. and Shapiro, M.M. 1966. A general equation and technique for the exact partitioning of chi-square contingency tables. Psychological Bulletin, 66: 252–262.

Brin, S., Motwani, R., and Silverstein, C. Beyond market baskets: generalizing association rules to correlations. ACM SIGMOD International Conference on Management of Data, pp. 265–276.

Card, S.K., Mackinlay, J.D., and Shneiderman, B. 1999. Information visualization. S.K. Card, J.D. Mackinlay, and B. Shneiderman (Eds.), Readings in Information Visualization: Using Vision to Think. San Francisco, CA: Morgan Kaufmann, pp. 1–34.

Chi, E.H., Pirolli, P., and Pitkow, J. 2000. The scent of a site: a system for analzing and predicting information scent, usage, and usability of a web site. Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems. Amsterdam: ACM Press, pp. 161–168.

Cooley, R. 2000. Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data. University of Minnesota, Faculty of the Graduate School: Ph.D. dissertation. http://www.cs.umn.edu/research/websift/papers/rwc_thesis.ps

Cooley, R., Mobasher, B., and Srivastava, J. 1999. Data preparation for mining world wide web browsing patterns. Journal of Knowledge and Information Systems, 1: 5–32.

Cooley, R., Tan, P.-N., and Srivastava, J. 2000. Discovery of interesting usage patterns from web data. In (Spiliopoulou and Masand, 2000), pp. 163–182.

Craven, M., DiPasquo, D., Freitag, D., and McCallum, A. 2000. Learning to construct knowledge bases from the World Wide Web. Artificial Intelligence, 118: 69–113.

Cugini, J. and Scholtz, J. (1999). VISVIP: 3D Visualization of Paths through Web Sites. Proceedings of the International Workshop on Web-Based Information Visualization (WebVis'99). Florence, Italy: IEEE Computer Society, pp. 259-263.

Fu, W.-T. 2001. ACT-PRO Action Protocol Analyzer: a tool for analyzing discrete action protocols. Behavior Research Methods, Instruments, & Computers.

Green, M. 1998. Toward a Perceptual Science of Multidimensional Data Visualization: Bertin and Beyond. http://www.ergogero.com/dataviz/dviz0.html.

Han, J. and Kamber, M. 2001. Data Mining: Concepts and Techniques. San Francisco, LA: Morgan Kaufmann.

Jones, T. and Berger, C. (1995). Students' use of multimedia science instruction: Designing for the MTV generation? Journal of Educational Multimedia and Hyermedia, 4: 305–320.

Mannila, H. and Toivonen, H. 1996. Discovering generalized episodes using minimal occurrences. Proceedings of the 2^{nd} International Conference KDD'96, pp. 146–151.

Menascé, D.A., Almeida, V., Fonseca, R., and Mendes, M.A. 1999. A methodology for workload characterization of E-commerce sites. Proceedings of the ACM Conference on Electronic Commerce, Denver, CO, November 1999.

Mobasher, B., Cooley, R., and Srivastava, J. (2000). Automatic personalization based on Web usage mining. Communications of the ACM, 43: 142–151.

Oberlander, J., Cox, R., Monaghan, P., Stenning, K., and Tobin, R. 1996. Individual differences in proof structures following multimodal logic teaching. Proceedings COGSCI'96. 201–206.

Olson, G.M., Herbsleb, J.D., and Rueter, H. 1994. Characterizing the sequential structure of interactive behaviors through statistical and grammatical techniques. Human-Computer Interaction, 9: 427–472.

Page, L., Brin, S., Motwani, R., and Winograd, T. 1998. The PageRank Citation Ranking: Bringing Order to the Web. http://www-db.stanford.edu/~backrub/pageranksub.ps.

Spiekermann, S., Grossklags, J., and Berendt, B. 2001. Stated privacy preferences versus actual behavior in EC environments: a reality check. Proceedings der 5. Internationalen Tagung Wirtschaftsinformatik. Augsburg, Germany, September 2001.

Spiekermann, S. and Parachiv, C. 2000. Motivating human-agent interaction: transferring insights from behavioral marketing to agent design. Proceedings of the 3^{rd} International Conference on Telecommunications and Electronic Commerce ICTEC3. November, pp. 387–402.

Spiliopoulou, M. 1999. The laborious way from data mining to web mining. International Journal of Computer Systems, Science, and Engineering, 14: 113–126.

Spiliopoulou, M. and Berendt, B. 2001. Kontrolle der Präsentation und Vermarktung von Gütern im WWW anhand von Data-Mining-Techniken. H. Hippner, U. Küsters and M. Meyer (Eds.), Handbuch Data Mining im Marketing. Wiesbaden, Germany: Vieweg, pp. 855–873.

Spiliopoulou, M. and Faulstich, L.C. 1999. WUM: A tool for web utilization analysis. Extended version of Proceedings EDBT Workshop WebDB'98. Berlin: Springer, pp. 184–203.

Spiliopoulou, M. and Masand, B. (Eds.). Advances in Web Usage Analysis and User Profiling. Berlin: Springer.

Spiliopoulou, M. and Pohle, C. 2001. Data mining for measuring and improving the success of web sites. Journal of Data Mining and Knowledge Discovery, 5: 85–14.

Spiliopoulou, M., Pohle, C., and Faulstich, L.C. 2000. Improving the effectiveness of a web site with web usage mining. In (Spiliopoulou and Masand, 2000), pp. 142–162.

Srikant, R. and Agrawal, R. 1996. Mining sequential patterns: generalizations and performance improvements. EDBT. Avignon, France, March, pp. 3–17.

Srivastava, J. Cooley, R., Deshpande, M., and Tan P.-N. 2000. Web usage mining: discovery and application of usage patterns from web data. SIGKDD Explorations, 1: 12–23.

Wang, K. 1997. Discovering patterns from large and dynamic sequential data. Intelligent Information Systems, 9: 8–33.

Bettina Berendt holds Master's Degrees in Economics, Business Economics, and Artificial Intelligence from Berlin, Cambridge, and Edinburgh University, and a PhD in Computer Science / Cognitive Science from Hamburg University. She is currently a research fellow at the Institute of Pedagogy and Informatics at Humboldt University Berlin. Her research integrates methods from Computer Science, Economics, and Psychology to analyze World Wide Web usage and interactions with computer visualizations.