Understanding Web usage at different levels of abstraction: coarsening and visualising sequences*

Bettina Berendt

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

Abstract

As Web sites begin to realise the advantages of engaging users in more extended interactions involving a semi-structured combination of information and communication behaviour, the log files recording Web usage become more complex. While Web usage mining provides for the syntactic specification of structured patterns like association rules or (generalised) sequences, it is less clear how to analyse and visualise usage data involving longer patterns with little expected structure, without losing an overview of the whole of all paths. The method proposed in this paper employs two ways of analysing complex Web usage data. Concept hierarchies are used as a basic method of aggregating Web pages. *Interval-based coarsening* is then proposed as a method for representing sequences at coarser levels of abstraction. Relations to the measures support, confidence, and ways of analysing generalised sequences are shown. The tool STRATDYN that implements interval-based coarsening uses χ^2 testing for analysing differences in support values, and *coarsened stratograms* to visualise Web usage, and to provide coarsening and zooming. A case study of agent-supported shopping in an E-commerce site illustrates the formalism.

Keywords: Web usage mining, sequence mining, visualisation, statistical methods, abstraction, agent communication

The way users navigate a Web site can be used to learn about their preferences and offer them a better adapted interface, from improving site design [SP01] to offering dynamic personalisation [MCS00].

However, behaviour is complex and can exhibit more 'local' and more 'global' regularities. This becomes particularly important in a site where meaningful behavioural patterns, i.e. episodes [W3C99], may extend over longer periods of time. Episodes are becoming longer as Web sites go from offering information, online catalogues, and purchasing options to utilising the full power of interactivity and communication. For example, E-commerce sites start to employ agents that offer users support along their way through the site and engage them in a sales dialogue. This kind of dialogue, along with the option to abandon it and/or restart it at any time, provides a rich, semi-structured interface, leading to more extended user interaction, and therefore more navigational information waiting to be discovered.

Much of the information contained in an interaction process is sequential. Sequence mining investigates the temporal characteristics of Web usage (e.g., [BBA+00, BL00, MT96, SA96, Wan97]). Queries and result representation focus on statistical measures like the frequency of sequences, and in addition may allow the specification and visual inspection of alternative paths taken through a site to reach a given goal page from a given start page [Spi99].

The powerful techniques available for the identification of patterns often lead to huge result sets. The mining challenge is to combine openness and little specification to be able to find unexpected patterns with enough structure to easily find meaningful results, i.e. interesting patterns.

One approach is to *select* patterns, e.g. by filtering based on numerical criteria like support thresholds or more sophisticated mechanisms [Coo00], or query languages to constrain patterns syntactically, e.g., [BBA+00, Spi99].

Another approach is to *abstract* from details by classifying accessed pages or paths. Concept hierarchies treat a number of Web pages as instances of a higher-level concept, based on page content (as in market basket analysis [HK01]), or by the kind of service requested, for example, the query options that a user employs to search a database [BS00].

*I thank the IWA team for supplying a highly useful data set, and my reviewers for helpful comments and suggestions.

Generalised sequences [Spi99] are used to define pattern templates that summarise a number of different sequences of requests. For example, $[A_1, [0; 5], A_2]$ matches all sequences found in user paths that start with a node A_1 and end with a node A_2, with up to 5 further arbitrary nodes in between. A generalised sequence thus abstracts sequences by declaring parts of user paths to be of secondary interest. One problem of this kind of aggregation is that the paths in between the specified nodes are either lost in the representation of results (if only the support and confidence of a pattern are given), or are represented in very fine detail (e.g., as the disjunction of all the different paths actually taken in [Spi99]). The latter then necessitates the visual inspection and formulation of new, abstract or specific, hypotheses to be tested. If these in-between paths are of interest, and there is not enough information or prior expectation to specify them syntactically, a mechanism appears desirable that presents a large number of paths in a compact way.

Visualisations can be regarded as supporting abstraction *per se*, because they utilise the human capabilities of quickly recognising patterns that may not stand out in a non-pictorial representation of the data. Different kinds of visualisations emphasise different aspects of Web usage. A popular class (e.g. [CPP00, CS99]) is related to visualisations of Web sites: Each page is assigned one place in 2D or 3D space, using graph layout algorithms to maximise the clarity of distribution across space. Transitions are shown as arrows between pages. Line thickness may be used to encode frequencies of transitions [HL01]. However, to identify the temporal progress of a user path, these arrows have to be traced individually, which makes it difficult to perceive a sequential pattern. Progress can be traced in detail in the tree representations of [Spi99], which involve multiple representations of pages that were visited several times. In both these kinds of approaches, the analyst can usually simplify the visualisations by selection, filtering out nodes and/or links. However, the position of a page in space bears no meaning, implying that positions and directions of visual elements cannot be directly interpreted.

To visualise single user paths, pages are often plotted against time (e.g., [JB95]). This employs "alignment" [CMS99], associating one page with one y coordinate. This makes position interpretable. In the "proofograms" of [OCM+96], the y values are arranged in a meaningful order, so relative position or directions of visual elements such as "lines from top left to bottom right" can be interpreted. However, the focus on single user paths makes these approaches inadequate for mining large amounts of data.

The present paper proposes *stratograms* as a way of combining these visualisation approaches. It extends this basic idea by introducing *coarsening* as an abstraction along the temporal dimension, as measured by the order of requests. The proposed method, *interval-based coarsening*, deals with binary as well as n-ary sequences, and it can be used to analyse generalised sequences. Relations to the standard Web usage mining measures support and confidence will be shown. *Coarsened stratograms* are presented as powerful visualisations that allow the results to be easily communicated to non-experts. This may allow more local *and* more global patterns to be detected. A coarse first overview of the data can also help the analyst to quickly concentrate on areas of the data that contain interesting patterns.

The paper first describes stratogram visualisations and coarsening. Throughout, a case study will be used to illustrate the formal argument. Section 4 then presents the statistical background, pattern discovery and comparison using type hierarchies. Algorithms to compute stratograms based on type hierarchies are then described, and extensions discussed. The methods are implemented in the tool STRATDYN, which was first presented in [Ber00].

1 Example: agent-supported shopping in an online store

The Web site used in the case study is an online store developed at the Institute of Information Systems 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 parallelised according to sales strategy, see [ASS01]), the shopping interface is identical for different products.

Here as in many other analyses of Web usage, the initial analysis questions were "What did users do in this site? When? And how often?". A first, incomplete conceptual sketch of possible activities in the site is shown in Fig. 1 (a).

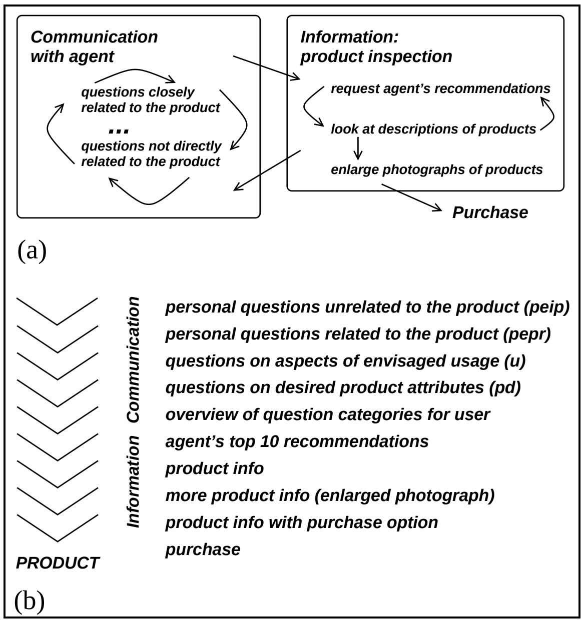

Figure 1: Activities/requests in the example site: (a) related to one another in an initial sketch, (b) ordered by increasing closeness to product.

The analyst's first task is to design a scheme which allows the classification of each URL and which distinguishes relevant activities. The result is shown in Fig. 1 (b).

The URLs of the main shopping phase were generated dynamically. URLs were classified as follows: "Q(uestion) categories" is an overview 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. 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 [ASS01]. 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 to bottom.

The present analysis used the navigation data of participants of an experiment with a choice of two product categories, compact cameras and winter jackets. Buying decisions were binding for participants (for details, see [SGB01] and http://iwa.-wiwi.hu-berlin.de).

Figure 2 shows stratograms aggregating the paths taken by 152 camera shoppers and 50 jacket shoppers through the store. The analysis focused on behaviour after the (highly structured) introductory phase. So all requests prior to a user's first request for the question categories page are not shown. In the phase shown, users were free to explore the site.

Each segment along the x axis denotes one step in the original logs. The two numbers at the right hand side of the figure both denote the maximal number of steps considered. Each line in the diagram connects a point (t, v) with another point $(t+1, v')$, see for example the zigzagging series of lines at the bottom left of each stratogram in Fig. 2 that connect "top 10" and "product info". Some lines are thicker, e.g., those at the bottom right between "top 10" and "product info" that are so close as to generate a visual 'block'. In contrast, connections between questions of different kinds in the top middle of the lower diagram are thinner. The thicker the line, the higher the proportion of shoppers that went from a URL plotted at v to another plotted at v' between steps t and $t+1$.

To find interesting patterns, pages have been abstracted using a concept hierarchy. To also find unexpected patterns, all paths through the site are investigated in their total length.

The figures show the unexpected result that two phases emerged in user behaviour: a 'communication phase' (top left) and an 'information phase', in which products were inspected (bottom right), and that their distinctness changes with product category. Commonalities and differences in behaviour are easily seen: First, most users have answered most of the questions, regardless of legitimacy / relevance, in the order suggested by the site. This is shown by the relatively few, thick lines at the top left. However, camera shoppers followed the sequence of questions even more closely before entering the information phase. In contrast, jackets were inspected already during the communication phase (see bottom left), and answers corrected, resulting in a longer communication phase. Also, in

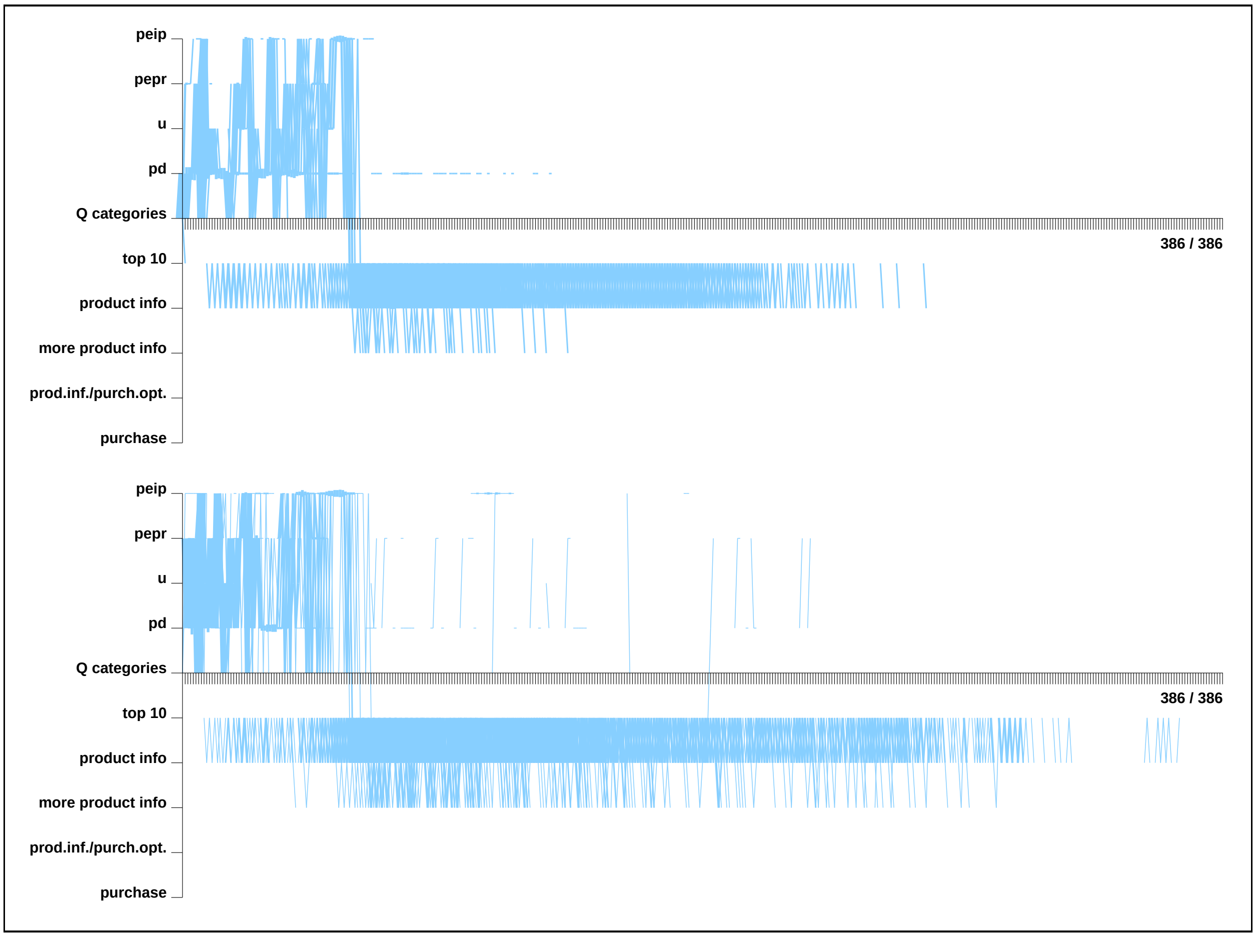

Figure 2: Basic stratograms of camera shoppers (top) and jacket shoppers (bottom)

the information phase, "more product info" was requested more often (see bottom right), and the information phase lasted longer. Statistical analysis showed that conversion efficiency, the ratio of the number of buyers to the number of all shoppers, was higher for cameras (55%) than for jackets (24%) ($\chi^2_1 = 14.75, p < 0.01$). In particular, conversion efficiency over short paths was higher (35% vs. 10%, $\chi^2_1 = 11.37, p < 0.01$). Paths were classified as "short" if they were shorter than half the maximal length of purchasing sessions. These results suggest that the design of the online store and its shopping agent may be more suited to selling search goods like cameras, i.e. products that may be judged by examining a range of technical details. Online selling of experience goods like jackets, on the other hand, may require further interface developments, offering better substitutes for the 'experience' of fabric and fit typically required to judge these products.

As this example has shown, stratograms address all three of the initial analysis questions. The ordering of pages along the y axis makes the nature of sequences of activities visible, e.g., "remaining within communication", "engaging in prolonged information-seeking behaviour", or "changing / alternating between communication and information". This addresses the question "What did users do in the site?". The ordering of requests along the x axis makes the temporal characteristics of sequences of activities visible, e.g., the division into a communication and an information phase. This addresses the question "When did users do something in the site?". The distribution of transitions along the x axis, together with the relative thickness of visual elements, addresses the question of "how often" certain activities were pursued.

2 Basic stratograms

A basic stratogram thus rests on the relative frequencies of transitions, i.e., binary sequences. For each session s from a total of S sessions, all requests after an offset are considered. The offset may be the first request in a session, or it may be the (first) request for a certain page, such as the first request for the question categories page in the example. $s.(o_s + t)$ denotes the t^{th} request, or step, in session s after the offset o_s. The *normalised frequency* of the transition from node A_1 to node A_2 at the t^{th} step after the respective session offsets o_s is[1]

$$f(A_1, A_2, t) = \frac{|\{s \mid s.(o_s+t)=A_1 \wedge s.(o_s+t+1)=A_2\}|}{S}. \quad (1)$$

Since the number of all transitions between t and $t+1$ is at most S (it may be less because some sessions may end earlier than $t+1$ steps after their respective offsets), each normalised frequency is at most 1, and their sum is at most 1.

In addition to frequencies, a stratogram requires a function v that maps the visited pages from the set *pages* to numerical values N according to some interpretation of the site's structure and content.[2] This may be a 'degree of specificity' in a search, or some other scale along which pages may be ordered for the analysis question at hand [Ber00]. In the running example of agent-supported online shopping, integers reflect the ordering of pages by closeness to the product. To be able to identify a transition's frequency with that of its associated numerical values, it is assumed for simplicity that the function v is bijective, i.e. that pages are not further summarised by v. Each session is augmented by a request for the special page "end" after its last request.

Definition 1 *A* basic stratogram strat *is defined as*

$$\begin{aligned} strat = \langle pages, st, v, tr, \theta_1, \theta_2 \rangle \quad with \\ st = \{0, \ldots, max_s(|s| - 2)\}, \\ v : pages \mapsto N, \\ tr = \{f(A_1, A_2, t) \mid \\ A_1 \in pages, A_2 \in pages \cup \{end\}, t \in st\} \end{aligned}$$

where the θ are support thresholds.

[1] The concepts and measures used in this paper are relative to a log and a page classification. To simplify notation, both will be assumed given and not included as extra arguments.

[2] More complex stratograms that make v depend on the page and the history before it was requested are discussed in [Ber00].

A basic stratogram visualization *consists of (1) for each t, A_1, A_2 s.t. $A_2 = end$ and $f(A_1, A_2, t) \geq \theta_1$: a circle with center $(t, v(A_1))$ and radius increasing with $f(A_1, A_2, t)$, and (2) for each other t, A_1, A_2 s.t. $A_2 \neq end$ and $f(A_1, A_2, t) \geq \theta_2$: a line from $(t, v(A_1))$ to $(t+1, v(A_2))$, with thickness increasing with $f(A_1, A_2, t)$.*

In the following, "stratogram" and "stratogram visualisation" will be used interchangeably when clarified by the context.

The number of steps is bounded above by the number of nodes in the longest session minus 1, so t ranges from 0 to $max_s(|s| - 2)$.

The stratogram is normalised by support levels either found in the data or imposed by the analyst, i.e. there are minimal support levels, or support thresholds, $sup_{min1|2} = \theta_{1|2}$. (In Figures 2 to 4, $\theta_1 = \theta_2 = 0.05$.)

3 Interval-based coarsening

The visualisation of longer episodes in basic stratograms harbours the danger that one may 'not see the wood for the trees' in the fine resolution of single-step actions. Also, actions that do occur frequently around, but not exactly at the same step, will not stand out as frequent.

3.1 Coarsened frequency tables and coarsened stratograms

Interval-based coarsening summarises transitions in consecutive, disjoint intervals of a size $g \geq 1$, starting from the respective offset. The *normalised frequency* of the transition from node A_1 to node A_2 in the t^{th} interval after the respective session offsets o_s is

$$f_g(A_1, A_2, t) = \sum_{x:t \times g}^{(t+1) \times g - 1} f(A_1, A_2, x). \quad (2)$$

This measure may count a given session several times if it contains more than one transition between A_1 and A_2 between steps tg and $(t+1)g$. However, each binary transition in the log is still counted exactly once.

The frequencies as defined in equation (2) can be tabulated, e.g. in a table with one row per transition (A_1, A_2), or transitions differentiated by product, and one column per interval t. The resulting table represents a coarsening of the table corresponding to equation (1). Frequency tables aggregated in

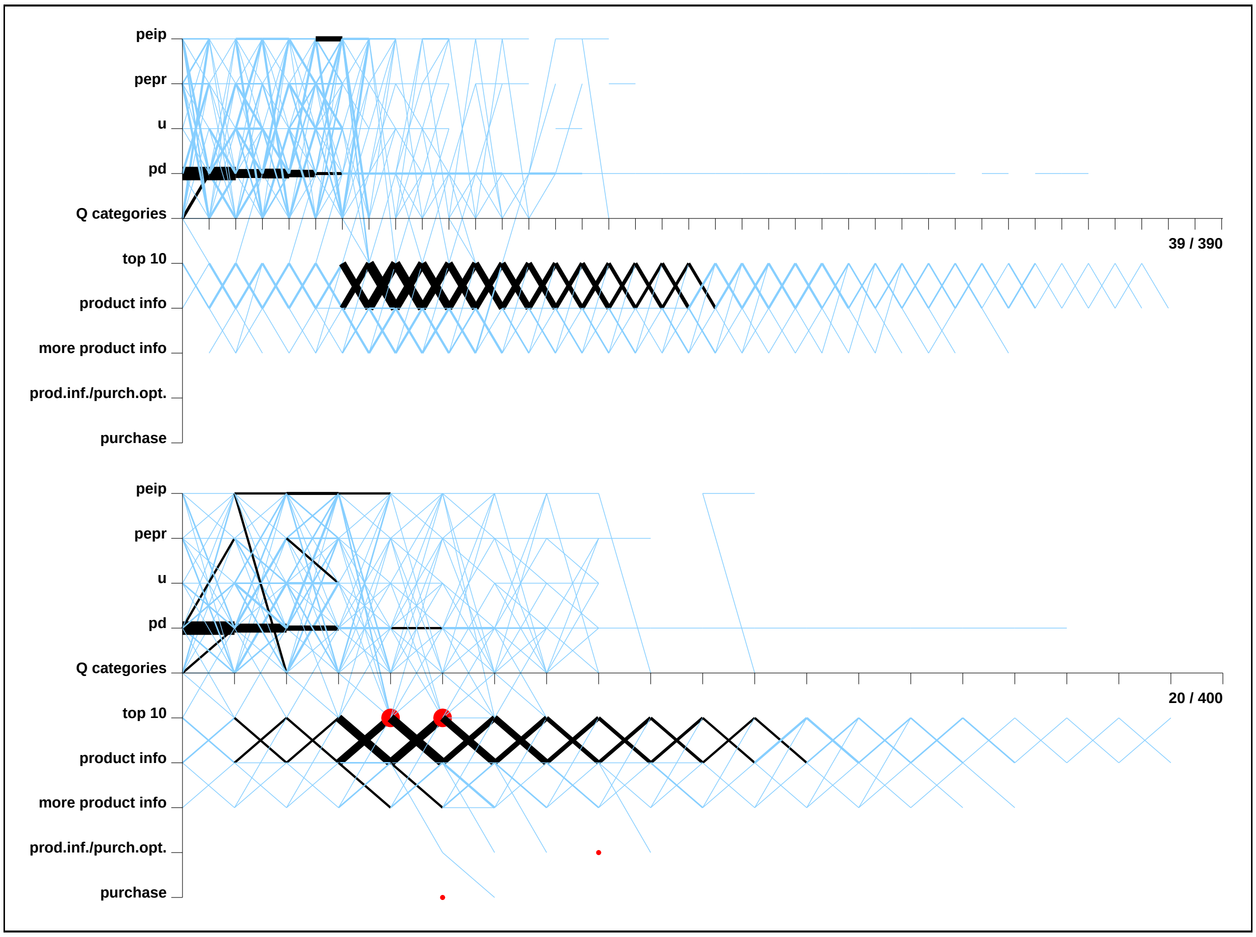

Figure 3: Cameras, $g = 10$ (top), $g = 20$ (bottom)

this way can be tested for statistically significant differences between single cells or groups of cells using χ^2 tests ([Ber00], see [OHR94] for generalisations to higher-order frequency tables).

Note that each cell can be interpreted as the *support* of that sequence in that interval. Adding cells, $\sum_{A_2} f(A_1, A_2, t)$ gives the support of node A_1 in that interval, allowing the *confidence* of each sequence to be calculated.

The visual equivalent of coarsened frequency tables is given by

Definition 2 *A* coarsened stratogram strat_g *with degree of coarsening g is defined as*

$$strat_g = \langle pages, st, v, tr, \theta_1, \theta_2, g \rangle \quad with$$
$$st = \{0, \ldots, int(\frac{max_s(|s| - 2)}{g})\},$$
$$v : pages \mapsto N,$$
$$tr = \{f_g(A_1, A_2, t) \mid$$
$$A_1 \in pages, A_2 \in pages \cup \{end\}, t \in st\}$$

where the θ are support thresholds.

A coarsened stratogram visualization *is defined analogously to a basic stratogram visualization.*

In a coarsened stratogram, the set of all transitions between t and $t+1$ includes g steps, so their number is at most $g \times S$. Therefore, a normalised frequency may be larger than 1. This can only be the case if in at least one session, the transition under consideration has occurred more than once. So this transition may be considered as 'more characteristic' of this interval than others. Therefore, these transitions are displayed not only as thicker to indicate their higher frequency, but also in a darker colour (black vs. grey) to indicate this qualitative

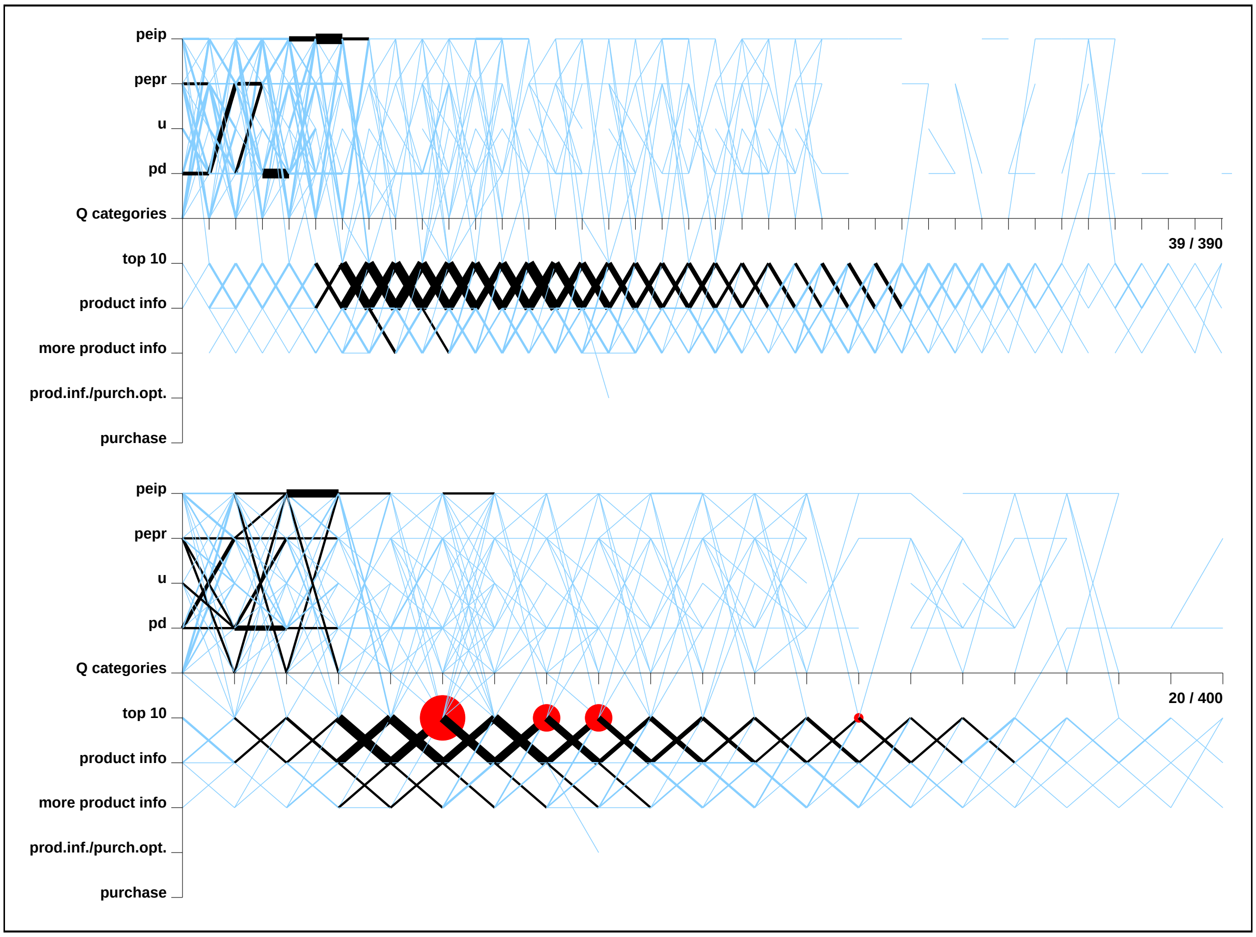

Figure 4: Jackets, $g = 10$ (top), $g = 20$ (bottom)

difference. Since each user leaves exactly once, the cumulation of frequencies does not apply for the circles denoting exits, so there is only one kind (colour) of circles. Figures 3 and 4 show examples of coarsened stratograms.

A value t along the x axis should be read as the $(tg)^{th}$ step in the original log, as shown by the two numbers at the right hand side of the figures.

It is straightforward to see that basic stratograms are one limiting case of coarsened stratograms ($g = 1$). In the opposite limiting case, $g \to \infty$, there is only one interval $[t, t+1] = [0, 1]$ to consider, which comprises all transitions anywhere between the first step $(0 \times g)$ and the last step of each session after its respective offset $(1 \times g)$. Frequencies of a binary transition are summed across all occurrences of that transition in all sessions, showing the support of that sequence over the whole log. An example is shown in Figure 5.

3.2 Visual operations and newly emerging patterns

One advantage of coarsened stratograms is that they summarise behaviour that may occur in roughly the same shape, but starting at different offsets after the initial offset o_s. This is illustrated in Figures 2 to 4. They also allow the analyst to first gain a summary view of the data and then 'zoom in' by decreasing the value of g.

Figure 6 illustrates the use of another zoom / unzoom operation: increasing the support thresholds reduces the number of transitions shown, increasing visual clarity. The figure shows that behaviour in the communication phase was more homogeneous in the first of the four distinct 'spiky' parts than in the rest.

Another advantage is that new regularities become visible. For example, Figures 3 and 4 show

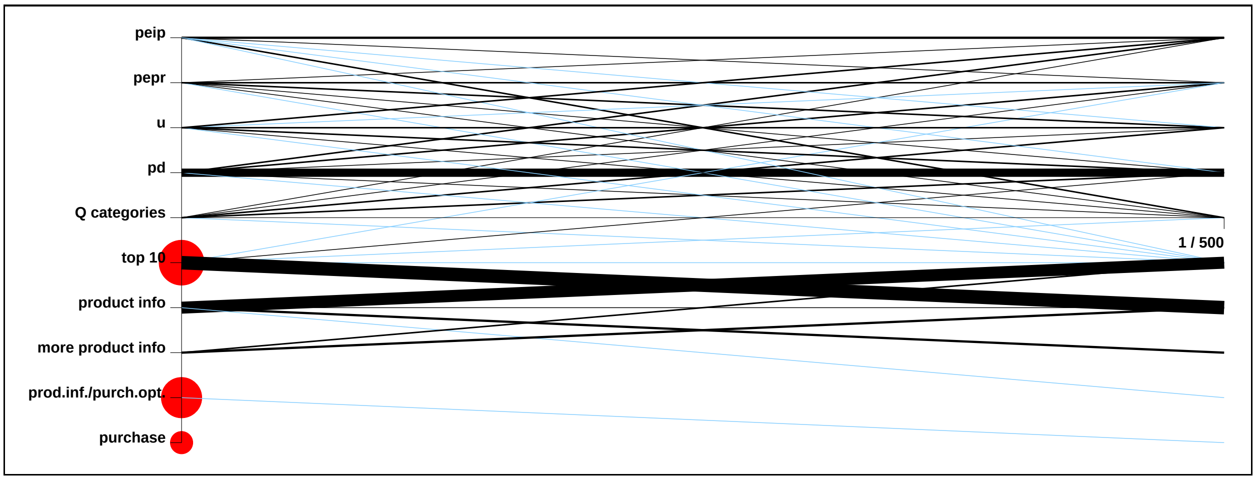

Figure 5: Camera shoppers, $g = 500$

that camera shoppers more often went from 'innocuous' pd questions to other pd questions, while jacket shoppers were more at risk of not only answering *one* personal peip question, but proceed directly to the next one, and that these patterns occurred at different stages of the whole navigation history. This is not visible in the basic stratograms in Figure 2, and the information would be lost in a standard analysis considering support across the whole session (cf. also Figure 5). Statistical analysis comparing the frequencies of these two transitions with those of other question transitions confirmed the visual impression ($\chi^2_2 = 422.71, p < 0.001$).[3]

While 'directly repetitive patterns' thus show up as thick horizontal lines, a new kind of regularity also becomes visible: cyclic behaviour. This is shown by thick X-shaped crossings between one step and the next. To understand why, consider the meaning of the two legs of an X: one marks a frequent transition, in the respective interval, from a node A_1 to a node A_2, while the other marks a frequent transition, in the same interval, from A_2 to A_1. The advantage of coarsening is that this kind of cyclic behaviour is not restricted to $[A_1, A_2, A_1, A_2, ...]$ sequences, but may involve in-between visit to other nodes. Figures 3 and 4 show clearly that there was a marked tendency for all shoppers to cycle between top 10 and product info pages, although this occurred earlier for camera shoppers than for jacket shoppers. The figures also show that cycling between product info and photo enlargement pages was much less pronounced. Both cycles went on for a much larger number of steps for jacket shoppers than for camera shoppers. Investigating the distributions of (top 10, info) transitions as a characteristic part of this pattern, it was found that their occurrence in the first 60 steps, the 200 steps after that, and the rest were highly different between products ($\chi^2_2 = 49.75, p < 0.001$).

Moreover, patterns of leaving the site become clearer. In the example, a clearer pattern emerges in the $g = 20$ stratograms concerning where and when a majority of shoppers leave the site. Statistical analysis showed that for cameras, more exits occurred in the first 100 steps than after that, and vice versa for jackets ($\chi^2_1 = 7, p < 0.05$).

In general, coarsening causes all lines and circles to become thicker, and some grey lines to become black. Additional elements may appear. This is because the summation in equation (2) makes the frequencies at each step t increase monotonically with g. Also, series of visual patterns are reduced to fewer items. For example, every two consecutive X-shaped crosses between "top 10" and "product info" in Fig. 3 (top) are reduced to one cross in Fig. 3 (bottom) because $g_2 = 2 \times g_1$. However, coarsening is usually combined with an increase in support thresholds. This has the reverse effect on the shape of the graph: Lines become thinner, change from black to grey, or disappear altogether. Circles are affected analo-

[3] The post hoc analysis should include α error corrections. However, in contrast to the shorter episodes analysed in [Ber00], the fine-grained analysis presented here allows for, and the visualisations encourage, a very large number of post hoc tests. Therefore, testing the hypotheses with a different dataset than the one used for exploratory analysis is advisable, and will be the subject of future work.

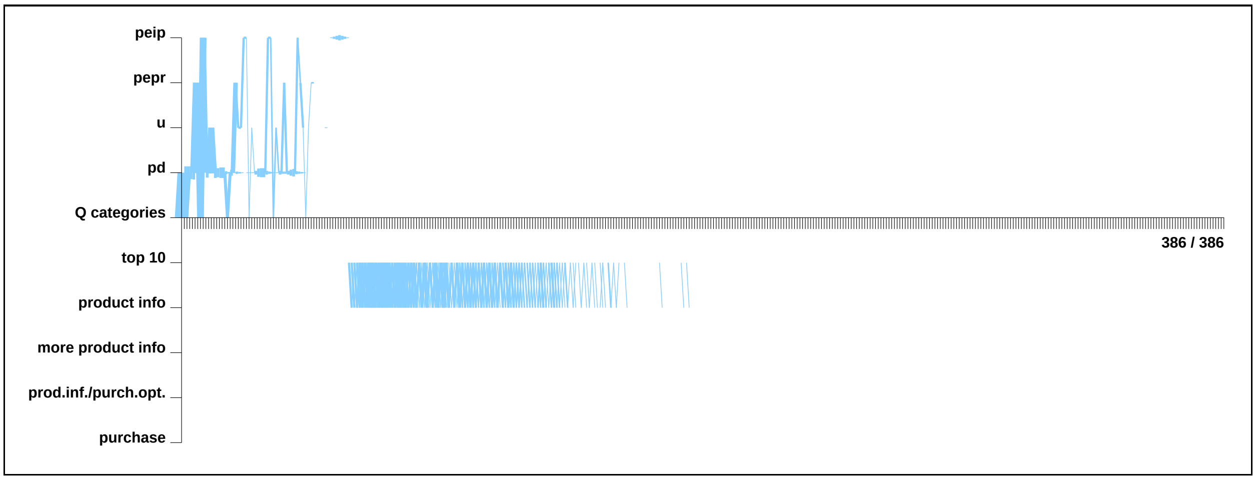

Figure 6: Basic stratograms of camera shoppers, $\theta_1 = \theta_2 = 0.15$

gously. The exact changes between one stratogram and another, coarsened one will therefore depend on the interaction of data, degree of coarsening, and adaptation of the support thresholds.

4 Pattern representation and discovery

4.1 Type hierarchies

Stratograms rest on *type hierarchies* that allow the discovery and statistical comparison of patterns independently of visualisation. Type hierarchies have been introduced in [Ber00] and will only be described shortly here. A type $[A_1, p_1, A_2, p_2, \ldots, A_n]$ matches sequences in user paths that contain A_1, then A_2, $\ldots$, and then A_n, with the paths in between constrained by the path specifications p_i specifying a minimum and a maximum number of requests that may lie between A_i and A_{i+1}. So types correspond to generalised sequences.[4] Several types can be instances of one type template $[a_1, p_1, a_2, p_2, \ldots, a_n]$. The types corresponding to one type template are joined together at their common prefixes to form a tree. Each path of length n from the root node of that tree to a leaf node is a type as specified by the type template, and each path from the root node to a non-leaf node is a supertype of at least one of these types, and itself a type $[A_1, p_1, \ldots, A_k]$ with $k < n$.

"end" and "other" leaf nodes are added to account for sessions that match up until an A_k, $k < n$, and then "end" or continue in ways "other" than the allowed types. To decrease memory usage, types and their frequency counters are only constructed by the algorithm when an instance is found in the data. So each session is classified as at most one type matching the type template. ("At most" because the session may not match any of the types.)

In the example, 11 types of length 2 starting at the offset question categories can be distinguished: *[Q categories, peip], [Q categories, pepr], . . . , [Q categories, purchase], [Q categories, end]*. The path specifications are all $[0, 0]$, i.e., there may be no further requests in between the nodes specified.

Since each session is counted only once, and classified entities are thus disjoint, the frequency counters allow one to use χ^2 tests to determine, for example, whether there were significantly more sessions that went from the initial question categories page to a *pd* page than to a *peip* page. All 11 session types of length 2 are related to the frequency of their common supertype *[Q categories]*.

In general, type hierarchies allow the specification of types of arbitrary length. In the current example, for a given depth n of the hierarchy, in principle all 10^n combinations of n nodes are conceivable types, plus those types with "end" occurring before length n. A stratogram frequency $f(A_1, A_2, t)$ is then the sum of the frequencies of all types τ of length $(t+1)$ such that $\tau.t = A_1$ and $\tau.(t+1) = A_2$. This draws on the classification of sessions.

[4] The difference is that in [Spi99], occurrence numbers of pages in the request sequence can also be specified in a generalised sequence; here, only the *first* occurrences are found.

However, the essential feature of type hierarchies is not that they classify each session only once, but that they partition the log or a subset of it. This means that a session may also be classified more than once, *as long as the classified sequences are disjoint.* So a set of stratograms frequencies $f(A_1, A_2, t)$ for $t = 0, \ldots, t_{max} = int(\frac{max_s(|s|-2)}{g})$ can also be regarded as the result of a repeated classification of sessions into types of length 2 (specifically, the type $[A_1, A_2]$). This idea is the basis for the first algorithm presented in the next section.

4.2 Two algorithms to produce coarsened stratograms

Coarsened stratograms can be derived from type hierarchies. An algorithm to classify sessions into a type hierarchy is given in [Ber00]: `classify(a,d)` recursively classifies a session starting from node `a` at level `d` of the hierarchy, starting with the root node at level 0. The algorithm was shown to be linear in $(L \times T)$, with L the size of the log and T the number of children of a node in the type hierarchy.

The general and highly localised type hierarchy employed in the coarsened stratograms shown so far considers binary sequences. It considers the $|pages|^2$ possible types for the given number of pages (here: $|pages| = T = 10$).

The difference to normal session classification are the repeated classification and the segmentation into intervals of size g. To ensure that the last node of a classified binary transition becomes the first node of the next, `classify` is adjusted such that it increments a line counter `i` by 1 with each read line. Moreover, it is adjusted to return a pair consisting of the last line read and counted towards the finished pattern, and a running number of that line. The type hierarchy contains one counter per interval `t`, and each sequence is counted towards the interval in which it started. I.e., the argument `t` of `classify` is unchanged in the recursive calls of `classify`.

```
(1) determine_frequencies_1 (g)
(2)   for each session do
(3)      repeat
(4)         read(z[0]);
(5)      until ((is_offset(z[0]))
(6)          or (end-of-session));
(7)      t := 0; i := 1;
(8)      while (not end-of-session) do
(9)         (z[0],i) := classify(z[0],i,t);
(10)        t := div(i,g);
```

Each request in each session (i.e., each request in the whole log) is read exactly once. Step (9) reads each request after the first, regards it as the second node of the binary transition, and therefore has to test it against T possibilities. So again, the complexity is $O(L \times T)$, and since T is constant for a given analysis, this means that the algorithm is linear in the size of the log. As discussed above, it is useful to classify pages using concept hierarchies, such that T will typically be small compared to L.

Alternatively, coarsened stratograms can be computed incrementally from their corresponding basic stratogram. As can be seen from equation (2), only the type hierarchy, and not the whole log, needs to be parsed:

```
determine_frequencies_2 (g)
  initialise all f_g(A1,A2,t):=0;
  for each A1 do
     for each A2 do
        x:=0; t:=0;
        while (x <= t_max) do
           f_g(A1,A2,t) += f(A1,A2,x)
           x++;
           if (x >= (t+1)*g) then
              t++;
```

This involves reading each of the nfc original frequency counters once, with $nfc \leq (T^2 \times t_{max})$, and summing them into at most $(T^2 \times \frac{t_{max}}{g})$ new frequency counters. Since the maximum number of different binary transitions in a log is limited by the total number of transitions in that log, we obtain $nfc \leq (L - 1)$. This shows that `determine_frequencies_2` requires less time than `determine_frequencies_1`. Repeating this procedure, e.g., by computing the $f_4(\ldots)$ from the $f_2(\ldots)$, will further reduce the number of steps needed.

4.3 Extensions: n-ary sequences and generalised sequences

A generalisation to n-ary types is desirable when contiguous non-atomic actions of interest involve more than 2 steps. First, consider n-ary types with empty path specifications, i.e., sequences of length n. The generalisation is straightforward: An n-ary sequence, like a binary sequence, is counted only once, in the interval containing its first node. The procedure developed in the previous paragraphs requires the following adaptations:

1. The frequency definition in expression (2) is extended to produce $f_g(A_1, A_2, \ldots, A_n, t)$. The changes to the right hand side of the definition, as well as to definition 2, are straightforward.

2. Within each interval $[t, t+1]$, $n-1$ subintervals are marked by the stratogram drawing routine, for example by vertical grid lines.

3. To ensure that lines do not obscure one another, a data structure is added that maintains a vertical offset for each grid point (i.e. each pair of a vertical grid line and a v value). Whenever one n-ary pattern has been drawn that traverses a grid point, the offset is incremented, such that the next line traversing the point will be drawn a little higher. It must be ensured that the vertical distance between different values of v is sufficiently large compared to the number of lines that can traverse a grid point.

Generalised sequences have a fixed number of nodes, for example, the generalised sequence *[A * B * C]* has three nodes, and path specifications in between that allow an arbitrary number of nodes. The basic algorithm of `classify` is already suited to identifying generalised sequences in a session. If the algorithm is started again after it has identified an instance of a sequence in a session, it will classify this session twice or several times, but again, it will only classify each binary transition at most once. Therefore, a generalised sequence with n nodes can be treated like an n-ary sequence by the algorithm and visualisation.

Setting $g = \infty$ allows one to derive an overall *support* and *confidence* value for a generalised sequence. For example, for an association rule defined as a generalised sequence with 2 fixed nodes $[A_1 * A_2]$, the frequency of non-overlapping occurrences of paths $[A_1, ..., A_2]$ in the whole log is given by $f_\infty(A_1, A_2, 0) = sup(A_1, A_2)$. The confidence of that generalised sequence can be computed analogously (cf. section 3.1).

5 Conclusions and outlook

The current paper has presented interval-based coarsening, and its inverse zooming, as a technique to mine Web usage at different levels of abstraction. Basic and coarsened stratograms have been proposed to visualise Web usage at different degrees of detail. Using a case study of online shopping with an anthropomorphic agent, we have demonstrated that this kind of abstraction offers new possibilities of understanding complex paths through a semi-structured, interaction-rich environment.

Further work will include extensions of the expressive power of the pattern representation language, and the associated abstraction possibilities. Two extensions are of particular interest. First, timestamp information will be investigated as a richer source of temporal information than the order information used here. Second, more complex grammatical forms of pattern notation will be investigated. Proceeding from the regular expressions that are generally employed in Web usage mining to context-free expressions [OHR94, Fu01] can enhance the power of abstraction considerably.

One of the main aims will be to find further ways of abstraction. An important factor is the number of pages visited, and the number of pages distinguished T. In the present paper, an aggregation of pages by concept hierarchies has been employed. This can also be regarded as a clustering of requests, or pages, along the stratograms' y axis: a user navigates from one cluster (e.g., a question page) to another cluster (e.g., a top 10 page). Interactive enhancements of stratograms could allow the analyst to delve into this cluster and distinguish which individual URLs were visited at this step by individual users. Requests / pages could also be clustered along the temporal dimension, i.e., along the x axis. This would show navigation between clusters, e.g., from questions to top 10 pages, without internal differentiation regarding how many question pages were visited. This abstraction requires an extended notion of generalised sequences: For example, navigation from the question cluster to the top 10 page would be a sequence $[question, question*, top10]$, with $*$ denoting an arbitrary number of pages of the given category. This requires an extension of the path specification concept to allow an arbitrary number of pages of a given category. The corresponding extensions of the query language will be a topic of further work.

This will be combined with a semantic analysis of different types of sites to derive further proposals of integrating background knowledge into mining.

References

[ASS01] Annacker, D., Spiekermann, S., & Strobel, M. *Private consumer information: A new search cost dimension in online environments.* To appear in *Proceedings of* 14^{th} *Bled Electronic Commerce Conference.* Bled, Slovenia, June 2001.

[BBA+00] Baumgarten, M., Büchner, A.G., Anand, S.S., Mulvenna, M.D.& Hughes, J.G. (2000). User-driven navigation pattern discovery from internet data. In M. Spiliopoulou & B. Masand (Eds.) *Advances in Web Usage Analysis and User Profiling.* Berlin etc.: Springer.

[Ber00] Berendt, B. (2000). Web usage mining, site semantics, and the support of navigation. In R. Kohavi, M. Spiliopoulou, J. Srivastava, & B. Masand (Eds.), *Working Notes of the Workshop "Web Mining for E-Commerce – Challenges and Opportunities." 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* (pp. 83–93). August 2000. Boston, MA.

[BS00] Berendt, B. & Spiliopoulou, M. (2000). Analysis of navigation behaviour in web sites integrating multiple information systems. *The VLDB Journal, 9,* 56–75.

[BL00] Borges, J. & Levene, M. (2000). Data mining of user navigation patterns. In M. Spiliopoulou & B. Masand (Eds.) *Advances in Web Usage Analysis and User Profiling.* Berlin etc.: Springer.

[BMS97] Brin, S., Motwani, R., & Silverstein, C. Beyond market baskets: Generalizing association rules to correlations. In *ACM SIGMOD International Conference on Management of Data* (pp. 265–276).

[CMS99] Card, S.K., Mackinlay, J.D., & Shneiderman, B. (1999). Information visualization. In S.K. Card, J.D. Mackinlay, & B. Shneiderman (Eds.), *Readings in Information Visualization: Using Vision to Think.* San Francisco, CA: Morgan Kaufmann, pp. 1–34.

[CPP00] Chi, E.H., Pirolli, P., & Pitkow, J. 2000. The scent of a site: a system for analzing and predicting information scent, usage, and usability of a web site. In *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems* (pp. 161–168). Amsterdam: ACM Press.

[Coo00] 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

[CS99] Cugini, J., & Scholtz, J. (1999). VISVIP: 3D Visualization of Paths through Web Sites. In *Proceedings of the International Workshop on Web-Based Information Visualization (WebVis'99)* (pp. 259–263). Florence, Italy: IEEE Computer Society.

[Fu01] Fu, W.-T. ACT-PRO Action Protocol Analyzer: a tool for analyzing discrete action protocols. To appear in *Behavior Research Methods, Instruments, and Computers.*

[HK01] Han, J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques.* San Francisco, LA: Morgan Kaufmann.

[HL01] Hong, J., & Landay, J.A. WebQuilt: A Framework for Capturing and Visualizing the Web Experience. In *Proceedings of The Tenth International World Wide Web Conference,* Hong Kong, May 2001.

[JB95] Jones, T. & Berger, C. (1995). Students' use of multimedia science instruction: Designing for the MTV generation? *Journal of Educational Multimedia and Hyermedia, 4,* 305–320.

[MT96] Mannila, H. & Toivonen, H. (1996). Discovering generalized episodes using minimal occurences. In *Proc. of 2nd Int. Conf. KDD'96* (pp. 146–151).

[MCS00] Mobasher, B., Cooley, R., & Srivastava, J. (2000). Automatic personalization based on web usage mining. *Communications of the ACM, 43(8),* 142–151.

[OCM+96] Oberlander, J., Cox, R., Monaghan, P., Stenning, K., and Tobin, R. 1996. Individual differences in proof structures following multimodal logic teaching. In *Proceedings COGSCI'96* (pp. 201–206).

[OHR94] Olson, G.M., Herbsleb, J.D., & Rueter, H. (1994). Characterizing the sequential structure of interactive behaviors through statistical and grammatical techniques. *Human-Computer Interaction, 9,* 427–472.

[SGB01] Spiekermann, S., Grossklags, J., & Berendt, B. Stated privacy preferences versus actual behaviour in EC environments: a reality check. To appear in *Proceedings der 5. Internationalen Tagung Wirtschaftsinformatik 2001.* Augsburg, Germany, 19-21 September 2001.

[Spi99] Spiliopoulou, M. (1999). The laborious way from data mining to web mining. *Int. Journal of Comp. Sys., Sci. & Eng., 14,* 113–126.

[SP01] Spiliopoulou, M. & Pohle, C. (2001). Data Mining for Measuring and Improving the Success of Web Sites. *Journal of Data Mining and Knowledge Discovery, Special Issue on E-commerce, 5*, 85–14.

[SA96] Srikant, R. & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *EDBT* (pp. 3–17). Avignon, France, March 1996.

[Wan97] Wang, K. (1997). Discovering patterns from large and dynamic sequential data. *Intelligent Information Systems, 9,* 8–33.

[W3C99] World Wide Web Committee Web Usage Characterization Activity. *W3C Working Draft: Web Characterization Terminology & Definitions Sheet.* www.w3.org/1999/05/WCA-terms/