Improving the Effectiveness of a Web Site with Web Usage Mining

Myra Spiliopoulou¶ Carsten Pohle|| Lukas C. Faulstich* **

Abstract

Effective web presence is for many companies indispensable for their success to the global market. In recent years, several methods have been developed for measuring and improving the effectiveness of commercial sites. However, they mostly concentrate on web page design and on access analysis. In this study, we propose a methodology of assessing the quality of a web site in turning its users into customers. Our methodology is based on the discovery and comparison of navigation patterns of customers and non-customers. This comparison leads to rules on how the site's topology should be improved. We further propose a technique for dynamically adapting the site according to those rules.

1 Introduction

Effective presence in the web is for many companies the key to success in the global market. This key has two dimensions: content and form. The evaluation of a web site in promoting its owner company should take both aspects into account.

Methods of measuring the quality of web page design (the *form*) have been proposed by [4, 5, 6] and others. However, the quality of a web *site* is not only determined by exciting web pages. A web site is a complicated network of pages, in which users should be assisted in finding information or products in a way *intuitive to them.*

Sequence mining [1, 9, 12] is the appropriate paradigm for analyzing the behaviour of users in a site. However, the original paradigm for discovering frequent sequences is not adequate, because it only extracts frequently visited nodes but no patterns on how users navigate. In [11, 10], we have proposed the miner `WUM`[1] that shifts the focus from frequent sequences to navigation patterns that are interesting in terms of their statistical and/or structural properties.

In this study, we propose the comparative analysis of navigation patterns of customers and non-customers, in order to improve the site's efficiency in turning non-customers into customers. To this purpose, we analyze the differences between their navigation patterns, and find the pages and paths that must be redesigned in order to conform to the perception of potential customers.

In the next section we describe how navigation patterns are discovered by our web utilization miner `WUM`. In section 3, we describe and extend a model measuring the efficiency of a web site in turning visitors to loyal customers into a model for comparing the efficiency of multiple typical navigation patterns. We then describe how navigation patterns of customers and non-customers are compared and how improvement hints can be drawn from the comparison. Section 4 describes how the results of the analysis can be put to praxis by processing the discovered patterns and building links according to them. Section 5 reports on the current state of our work and on future activities.

¶Institute of Information Systems, Humboldt-University Berlin `myra@wiwi.hu-berlin.de`

||Institute of Information Systems, Humboldt-University Berlin `cpohle@snafu.de`

*Institute of Computer Science, Free University Berlin `faulstic@inf.fu-berlin.de`

**Supported by the German Research Society, Berlin-Brandenburg Graduate School on Distributed Information Systems (DFG grant no. GRK 316)

[1] `http://wum.wiwi.hu-berlin.de`

2 Discovering navigational patterns with WUM

A sequence is a vector of events. In web usage analysis, those events are accesses to web pages. In sequence mining, the events constituting a *frequent sequence* need not be adjacent [1]. For instance, *ad* is a frequent sequence over a transaction log comprised of *abcd, baed, abcefd*, because it is contained in all sequences, although *d* never follows *a* directly. When we study the navigation behaviour of web users, this may lead to misinterpretations of the results.

In [10], we propose the notion of "g-sequences" as a vector composed of events and wildcards. In this terminology, a frequent sequence is conceptually identical to a g-sequence with wildcars between its elements. In the above example, the g-sequence *a*d* states that any number of events can occur after *a* and before *d*. Wildcards can also contain structural constraints. For example, the g-sequence *a*[1;2]d* matches all sequences with one or two events between *a* and *d*, only.

A g-sequence stands for a uniquely defined group of sequences that constitutes a "*navigation pattern*". The navigation pattern of *b*[1;5]d* is shown in Fig. 1 as a tree; this representation has no information loss. Each node is annotated by the number of subsequences up to this node.

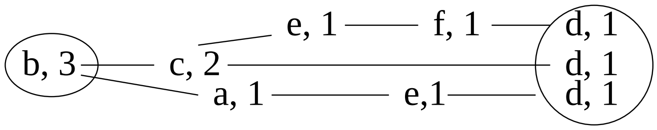

Figure 1: The navigation pattern for the g-sequence *b*[1;5]d*

A conventional sequence miner returns all frequent g-sequences of arbitrary length. `WUM` rather discovers all navigation patterns that satisfy a template with a fixed number of variables, subject to more arbitrary constraints on the variable's statistics and the template's structure. For example:

```
SELECT t FROM NODE AS x y, TEMPLATE x *[1;5] y AS t
WHERE x.support >= 3 AND ( y.support / x.support ) >= 0.5
```

This mining query, expressed in the language MINT [11], asks for all g-sequences, such that (i) the event bound to x appears in at least 3 sequences and (ii) the event bound to y appears in at least 50% of *those sequences* (i.e. after the event bound to x). The g-sequence *b*[1;5]d* satisfies template t. `WUM` returns as result *b*[1;5]d and* the navigation pattern of Fig. 1.

3 Comparing the navigational behaviour of user groups

Frequent navigation patterns give indications on how the site is being used. However, this information is still not sufficient for building a successful web site. To improve a site's quality, we need to compare the behaviour of users exploring the site as expected to those that do not.

3.1 Types of site users

Berthon et al classify the users of a site into "short time visitors", "active investigators" and "customers" [3]. A short time visitor is a user staying in the site for only a short while. An active investigator stays longer and explores the site. Some active investigators become customers, in the sense of buying products or exploiting the offered services. The ratio of active investigators to all users measures the site's "contact efficiency"; the ratio of customers to active investigators measures its "conversion efficiency" [3]. Our goal is to maximize these ratios.

We use a variation of this classification, by specifying the types of users more explicitly. A "short time visitor" is a user that visits only one page of the site, the page by which she reached the site. All other users are "active investigators". A "customer" is an active investigator that has purchased something, otherwise she is a "non-customer". Since we allow for multiple groups of customers, manifesting themselves by different navigation patterns, we take account for the fact that experienced customers behave differently than ones unfamiliar with the site.

3.2 Patterns Reflecting Contact Deficiencies

To compute the contact efficiency of a site, we just enumerate its users and extract its active investigators [3]. To *improve* the contact efficiency, we must compare the behaviour of the active investigators to the behaviour of short time users. For this, we discover all patterns that consist only of the first page being accessed and all those containing at least one more page. In WUM, we issue two queries [11]:

```
SELECT t FROM NODE AS x, TEMPLATE ^ x AS t
SELECT t FROM NODE AS x y, TEMPLATE ^ x y AS t
```

where the ^ symbol denotes that the variable x should be bound to the first page of a navigation path. The application expert may add constraints and restrict the discovery to pages that have been accessed at least a minimum number of times.

The first query returns all first pages accessed by users. The second query returns those of them, from which one further access has taken place. Pages appearing only in the first query should be subjected to careful *page* redesign.

Similarly, pages bound to x in both queries but rarely leading to a further access are candidates for redesign. WUM supports the analyst in inspecting these pages by its visualization module. From the graphical representation of the result, the analyst can check whether the number of accesses in the first page is equal or close to the total number of accesses among the paths emanating from this page. Pages with a large gap between those two numbers have a potential for increasing the contact efficiency.

3.3 Patterns Revealing Conversion Deficiencies

To compute the conversion efficiency of the site, we simply divide its customers by its active investigators [3]. To improve this value, we must compare the navigation patterns of customers to those of non-customers. The members of each of the two groups form subgroups reflecting differences in navigation style, explorative impetus and need for assistance. We can assume that members of each subgroup are present in each of the main groups. Hence, by discovering each typical navigation pattern of customers and finding its differences to the most similar ones of non-customers, we can extract concrete hints on how the web site should be improved to make the behaviour of the latter similar to the behaviour of the former.

To this task, we first filter the web server log to remove the accesses of short time visitors. We then generalize from the contents of the individual web pages to more general concepts. These can be product types as in market basket analysis, document descriptors, or query parameters for automatically generated web pages. This issue is discussed in [2].

We split the preprocessed log into the log of customers and the log of non-customers. Pattern discovery is performed in each log separately and according to the same statistical and structural constraints. Then, the navigation patterns discovered in the two logs must be compared.

A mining result is a g-sequence and its navigation pattern, both annotated with statistical data. This makes the comparison of mining results difficult: For example, the sequence *abc* contains *ab* but the support of *ab* is higher than that of *abc*, because all accesses to c via *ab* contribute to the support of *ab*. To simplify the comparison process, we use the following rules.

(R_1) *Two patterns are comparable, if their g-sequences have at least the first n values in common.* The analyst may set this n to a small number, like 1 or 2, if she suspects that customers and non-customers access completely different pages. Otherwise, she may set n to the length of each g-sequence, so that only patterns of g-sequences containing the same URLs are comparable. (R_2) *Within each pattern, only statistically dominant path fragments are considered for comparisons, while rare fragments are removed.* An intuitive lowerbound would be the support threshold used to discover frequent patterns in the whole customer, resp. non-customer log. This lowerbound applies to fragments *in the context of the pattern*, in order to remove rarely visited pathways.

With these rules in mind, we establish the collections of patterns to be compared as follows:

1. The miner discovers the g-sequences $g_1, \ldots, g_k$ in the *customer log* and the g-sequences $ng_1, \ldots, ng_m$ in the *non-customer log*.
2. For each g-sequence g_i, a MINT query describing the structure and URLs of its first n variables is issued against the non-customer log. The result is a g-sequence ng_i with the same URLs as g_i but with different statistics.

   The same is done for each ng_j against the customer log.
3. All pairs of g_i, ng_i g-sequences whose ratio of support is below a noise threshold, are removed.
4. A set of potentially comparable results is built: $\mathcal{R} = \{((g_i, CP_i), (ng_i, nCP_i))\}$ where g_i and ng_i are the retained g-sequences and CP_i, resp. nCP_i are their navigation patterns.
5. For each navigation pattern in $\mathcal{R}$, we remove all path fragments that are rare *within the pattern* (cf. rule (R_2)).
6. From each pair $((g_u, CP_u), (ng_u, nCP_u)) \in \mathcal{R}$, we extract all pairs of individual paths (x, y), forming a set $\mathcal{R}'$.

To compare the individual paths, we introduce the statistic measure of "successor-confidence" as the ratio of accesses between each node a in a path and its predecessor a': $SC(a, a') \equiv SC$. Then, for a non-customer path y, the following cases may occur:

1. There is a customer path $x \cdot x'$, where x contains the same URLs as y but has lower support.

   We compare the successor-confidence for each pair of consecutive pages in y and in $x \cdot x'$, until we find the first page in y that shows a lower SC value towards its predecessor than the respective pair of pages in $x \cdot x'$. This predecessor page should be redesigned.
2. There is a customer path $x \cdot x'$, where x contains the same URLs as y but has higher support.

   We find the non-customer paths of the form $y \cdot y'$ and we compare each y' to x' in terms of content. Of interest is each common page a with different followup pages $a_{y'}$ in y' and $a_{x'}$ in x', such that the SC value of $a_{y'}$ is larger than that of $a_{x'}$. For each such page, its connection to $a_{x'}$ appearing in the customer pattern should be reconsidered, since non-customers apparently prefer another page to it.
3. There is a customer path $x_1 \cdot x \cdot x_2$, where x contains the same URLs as y.

   We find the non-customer paths of the form $y_1 \cdot y \cdot y_2$, such that y_1 has no counterpart among the customer paths. Each such path fragment y_1 indicates a route that needs improvement. For instance, users who enter the site at a page other than the official welcome-page might experience difficulties in getting oriented.

Finally, non-customer paths with no comparable customer paths can be observed as reflections of users that perceive the site in a completely different way than assumed by the site designer. If there are many such users, special adjustments of the site for them might be considered.

4 Improving a Web Site by Dynamic Rule-Based Links

The results of our comparative study of navigation patterns can lead to extensive re-design of the site. In many cases, though, only some pages need to be modified or linked. Such changes should not be performed statically, because they reflect the navigation behaviour and needs of only one group of users. We rather propose the use of a proxy server, which establishes an adaptive web site, according to the mining results.

4.1 Derivation of annotation rules

We propose the establishment of a base of "annotation rules" having the form $g, a \rightarrow b$, where g is a g-sequence and a, b are URLs. $g \cdot a$ is typical for non-customers, while $g \cdot b$ is a frequent customer g-sequence. To guide a user who has followed g and now requests a, a link to page b should be added to page a. Note that in static redesign, the last page of g would be modified to stress the link to b. In contrast, dynamic adjustment avoids links that might confuse the group of users that would choose b anyway. Instead, a link is created on a to stress the existence of b as an alternative route.

These annotation rules are derived from the comparative analysis described in the previous section, after the inspection of the analyst. The derivation itself cannot be automated, since the designer must decide which patterns would lead to permanent changes of the site, and when dynamic adjustment should be preferred.

4.2 Applying annotation rules with the HyperView system

We adapt the Web site linkage to the annotation rules by coupling the original web site with a proxy server that applies those rules. As proxy server we use the HyperView system [7, 8] coupled with a metadata server that maintains the annotation rules.

This architecture is depicted in Fig. 2. We can see that the original site is left unchanged. WUM is used to discover the navigation patterns reflecting the behaviour for customers and non-customers in the site. The patterns are compared as described in section 3. The derived annotation rules are stored in the "metadata server" as RDF [2] documents. The HyperView proxy intercepts a HTTP request for a page of the original Web site, retrieves this page from the original Web site, adapts the requested page on the fly according to the user's browsing history and information from the metadata server. It then forwards the adapted page to the user.

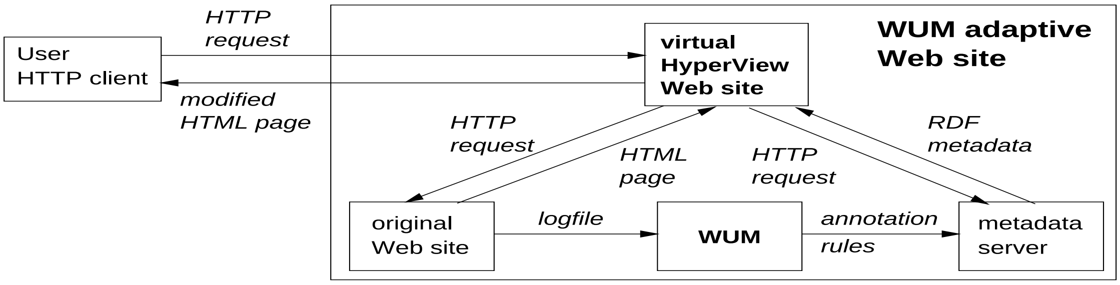

Figure 2: Building an adaptive web site with WUM and HyperView

To adapt the web pages for each user, the HyperView server builds and maintains the users' sessions, using the same heuristics as the data preparation module of WUM [11]. For each page request, the HyperView server retrieves the annotation rules whose LHS matches the pages requested thus far, and applies them to the requested page before transmitting it to the user.

Applying an annotation rule $g, a \rightarrow b$ implies modifying the requested page a to contain a link to b. A simple method would be to create a HTML frame containing a and the desired link. However, the effectiveness of this method may be low since the original page is not changed to reflect the importance of the new link. A better method would be to automatically embed the link in the original page, probably labeling and surrounding it with further material from b. HyperView provides a rule-based language, HVQL, with which the dynamically formed page can be expressed as a view combining data from both pages a and c.

A third solution is possible if the original Web site is itself a dynamic Web site based on HyperView. It then suffices to modify the views defining the Web site in order to take the annotation rules into account. Thus, the original Web site itself is turned into an adaptive Web site.

[2] The *Resource Description Framework* (see <http://www.w3.org/RDF>).

5 Current Status and Future Work

We propose the comparative study of the navigation behaviour of customers and non-customers in order to assess and improve the quality of a commercial web site. We increase the efficiency of the site in motivating site users to become customers, by identifying comparable navigation patterns of customers and non-customers. We detect the pages at which they deviate from each other and adjust them dynamically, according to the behaviour pattern of each user.

We currently apply our method in the analysis of SchulWeb (`http://www.schulweb.de`), a site accommodating the largest database of German high schools in the web. SchulWeb is organized like an online catalog with fill-in forms that generate pages of results. It does not have "customers" per se, so we did not partition the log. We rather identified patterns corresponding to the expectations of the designer and compared them to the frequent patterns discovered in the log. Our preliminary results lead to the comparison of an expected pattern *abc* with a pattern *a*[1;12]bc* having higher support. The inspection of *a* revealed indeed a shortcoming in the description leading to *b* [2].

Our future work includes optimizations of the algorithm performing the comparisons, by employing heuristics to reduce the paths being tested. Moreover, we anticipate that users can also be grouped by interests and demographics. We intend to combine such user clusters with groups formed by people having similar navigation behaviour and compare their activities in the site.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *ICDE*, Taipei, Taiwan, Mar. 1995.

[2] Bettina Berendt and Myra Spiliopoulou. Analysing navigation behaviour in web sites integrating multiple information systems. submitted, June 1999.

[3] Pierre Berthon, Leyland F. Pitt, and Richard T. Watson. The world wide web as an advertising medium. *Journal of Advertising Research*, 36(1):43–54, 1996.

[4] Xavier Drèze and Fred Zufryden. Testing web site design and promotional content. *Journal of Advertising Research*, 37(2):77–91, 1997.

[5] Xavier Drèze and Fred Zufryden. Is internet advertising ready for prime time? *Journal of Advertising Research*, 38(3):7–18, 1998.

[6] John Eighmey. Profiling user responses to commercial web sites. *Journal of Advertising Research*, 37(2):59–66, May-June 1997.

[7] Lukas C. Faulstich. Building HyperView web sites. Technical Report B 99-09, Inst. of Computer Science, Free University Berlin, 1999.

[8] Lukas C. Faulstich and Myra Spiliopoulou. Building HyperView wrappers for publisher websites. In *2nd European Conf. on Digital Libraries (ECDL'98)*, LNCS 1513, Heraclion, Greece, Sept. 1998. Springer Verlag. Extended version to appear in the IJODL Special Issue "In the Tradition of Alexandrian Scholars".

[9] Heikki Mannila and Hannu Toivonen. Discovering generalized episodes using minimal occurences. In *KDD'96*, pages 146–151, 1996.

[10] Myra Spiliopoulou. The laborious way from data mining to web mining. *Int. Journal of Comp. Sys., Sci. & Eng., Special Issue on "Semantics of the Web"*, Mar. 1999.

[11] Myra Spiliopoulou and Lukas C. Faulstich. WUM: A Tool for Web Utilization Analysis. In *extended version of Proc. EDBT Workshop WebDB'98*, LNCS 1590. Springer Verlag, 1999.

[12] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *EDBT*, Avignon, France, Mar. 1996.