Optimizing hierarchical menus: a usage-based approach
Hollink, V.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 2
The role of information gain in menu optimization

In this chapter we explore methods to minimize the number of steps users need to make to reach their target information. We identify a serious shortcoming of existing methods that under certain circumstances prevents them from minimizing the number of steps. At each step these methods focus on maximizing the probability of guiding a user to a target directly. We show why this sometimes leads to suboptimal results and present an alternative method based on information theory. We report on three studies that explore possibilities to apply this method in various settings.

The first study is based on a paper authored by M. W. van Someren, V. Hollink and S. ten Hagen, published in Web Mining: From Web to Semantic Web, Proceedings of the First European Web Mining Forum (Van Someren et al., 2004). The second study, which is co-authored by M. W. van Someren, S. ten Hagen and B. J. Wielinga, is based on papers presented at the Workshop on Intelligent Techniques for Web Personalization (Hollink et al., 2005b) and the Sixth Dutch-Belgian Information Retrieval Workshop (Hollink and Van Someren, 2006). The third study is co-authored by M. W. van Someren, S. ten Hagen, M. C. Hilgersom and T. J. M. Rövekamp and was presented at the Workshop on Intelligent Techniques for Web Personalization (Hollink et al., 2007a).

2.1 Introduction

The World Wide Web has made large amounts of information publically available. However, most of this information is not relevant for most users. As a result, users experience more and more difficulties to find the information they need among the overwhelming quantities of uninteresting information. Adaptive web sites aim to solve these problems by pointing users to the information that is interesting for them. In contrast to search engines, these systems are part of a web site and help a user to find information within this site.
Chapter 2: The role of information gain in menu optimization

In this chapter we address adaptive web sites that provide online navigation assistance by dynamically adding hyperlinks to the static menus of the web pages that a user is visiting. The links shown in a static menu are created by the developers of the site and look the same each time a page is visited. When a user requests a page, the menu adaptation system adds some dynamically created links to the static menu links that are already present on the page. In the next step, the user chooses one of the links (a static or dynamic link) and the system again adds some links to the requested page. When a site does not provide a static menu, a menu adaptation system determines completely which menu links are available to the users. In this case, menu adaptation reduces to menu generation.

When a person searches a web site to find specific information, there is a set of pages which together provide the best answer to his or her information needs. We refer to the pages in this set as the user's target pages. The goal of a menu adaptation system is to help the user to reach his target pages as fast as possible. In this Chapter, we assume that navigation time is determined only by the number of clicks the user needs to make. In other words, the adaptation component needs to minimize the number of clicks between the user's entry point and his target pages. Once the menu adaptation system knows a user's goal, it can show a direct link to the target information allowing the user to reach his goal in one click. The challenge for these systems is thus to find out as fast as possible what the user's goal is.

In theory, a menu adaptation system can make all pages of a site available in one step by showing links to all pages on the site's entry page. However, this solution does not help the users very much as presenting too many links on a page increases the effort needed to select the best link. Therefore, we fix the number of links that are added to each page as is done by most systems (e.g. Symeonidis et al., 2006; Pazzani and Billsus, 2002; Zhang and Iyengar, 2002).

Menu adaptation systems estimate the probability that a user is interested in each page of a site using, for instance, the pages that the user has visited previously or the content of the pages. On the basis of this information the system selects links that are added to the page that the user has requested. Most systems maximize the probability of leading the user to a target page directly by always adding the links with the highest probability of being the user's target (e.g. Balabanović, 1997; Burke, 2002; Lekakos and Giaglis, 2007). Throughout this paper this strategy will be referred to as the greedy strategy.

Although the greedy strategy does maximize the probability of showing a link to a target page at each step in the navigation process, it does not necessarily minimize the length of the path to the target pages. A better strategy is to actively try to learn the user's interests, i.e. show those links that provide most information. Compare this to binary search: if we want to determine a number between 1 and 100, the optimal strategy is not to start guessing 'Is it 37?', but to cut the range of possible numbers in two by asking 'Is it higher than 50?'.

A menu adaptation system cannot ask a user questions directly, but it can use the added links to gain information about the user's targets. When a user opens a link, this selection provides information about the user's targets. In particular, it tells the adaptation system that with high probability the user perceived the selected link
2.2. Related work

As stated in the introduction, the majority of the menu adaptation systems always adds links to the pages that they believe to be most interesting for the user. These systems include, for instance, Lieberman (1995), Balabanović (1997), Burke (2002), Zhang and Iyengar (2002), Smyth and Cotter (2003), Adda et al. (2005), Symeondis et al. (2006) and Lekakos and Giaglis (2007). A few non-greedy selection strategies have been proposed, most of which are based on the idea that a set of links that are added to a menu must not contain too similar items. In this section we give an overview of these strategies. In addition, we discuss several methods for tasks that are closely related to menu adaptation.

Various non-greedy methods have been proposed in the context of recommender systems. Information theory prescribes that the most informative question is the one that divides the set of possible target items into sets with equal probability mass. For instance, in the number example, the probability of the set of numbers higher than 50 is equal to the probability of the set of numbers lower than 50 (assuming that the numbers have uniform a priori probabilities). Therefore, ‘Is it higher than 50?’ is a maximally informative question. On average maximally informative questions will lead you to the target number as fast as possible. In the same vein, a maximally informative set of links is a set that divides the set of pages of a site into parts with equal probability of containing the user’s targets. Showing these links to the user will on average result in minimal path lengths.

In this chapter we report on three studies that explore possibilities to divide page sets by adding links to menus. We experiment with three methods that rely on various premises. The first method assumes that the pages of a site can be scaled along one dimension in such a way that users prefer pages that are on the scale closer to their targets over pages at larger distance. This method uses the scale to determine which links will provide most information. The second method does not require such a rigid structure, but assumes the pages are labeled with keywords. The third method computes distances between pages and adds links that are at large distance of each other. In the three studies the methods are evaluated and the advantages and limitations of the methods are assessed. We compare each of the methods to the greedy approach and discuss the added value of the methods in theory and in practice.

The rest of this chapter is organized as follows. Section 2.2 discusses related work. In Section 2.3 we describe the problem setting. The three experimental studies are presented in Sections 2.4, 2.5 and 2.6. The last section summarizes the lessons we have learned and discusses implications for further research.
systems. Recommender systems are very similar to the menu adaptation systems discussed in this chapter as both types of systems provide users with a number of links with the aim to improve navigation efficiency. Smyth and McClave (2001) argue that diversity is an important property of a recommendation set. They provide a metric to compute diversity and a number of selection strategies that enhance diversity. In Bradley and Smyth (2001) these strategies are refined. They evaluate the effects of the selection strategies on the diversity of recommendations and the computational costs of the selection. The effects on user navigation are not assessed. Ziegler et al. (2005) provide another diversity measure based on the distance between items in a taxonomy. A linear combination of page probability and diversity is used to select recommendations. The method is evaluated in a survey among users of an online book site. This survey shows that users like the lists of recommendations that are selected in this way better than the lists that are selected on the basis of page probability alone. Again, the evaluation does not address the effects of diversity on navigation. Bala-banović (1998) proposes to recommend pages of which the interest of the user is least certain. Simulation experiments show that this strategy can help a recommender to learn the users' interests faster, especially when users have complex interest patterns.

The benefits of diverse page sets is also researched in the context of critiquing. Instead of just selecting a link to an item, with critiquing users provide feedback in the form of statements like 'I want something like this item, but the value of attribute X must be more Y'. McGinty and Smyth (2003) show with simulation experiments that in this setting the diversity enhancing strategy from Smyth and McClave (2001) can lead to shorter navigation paths than a strategy that always selects the most probable links. However, in user experiments increased diversity made navigation paths longer (McCarthy et al., 2005). The ExpertClerk critiquing system (Shimazu, 2002) ensures diversity by showing links to items with various attributes. No experiments were done regarding the efficiency of this diversity enhancing strategy.

Other application areas in which users' clicks are said to be minimized are the automatic construction of web directories and the automatic clustering of web search results. In both areas large sets of web pages are clustered to allow users to browse through the information more efficiently. Web directories are static hierarchies of clusters of a selected set of web documents. Web search result clustering happens online after a search engine has retrieved a set of documents matching a user's query (e.g. Zamir and Etzioni, 1999; Osdin et al., 2002; Hearst and Pedersen, 1996). In these areas the clusters are formed in such a way that the documents in a cluster are closely related in terms of content or usage. To our knowledge no attempts have been made to optimize the clusters from an information theoretic perspective.

Witten et al. (1984) automatically create hierarchical menus that function as indexes for digital phonebooks. The menus contain links that refer to segments of the alphabet (e.g. 'Adda-Bradley', 'Burke-Cramer'). They use the entropy of the access probabilities of the names to select the segments that minimize the lengths of the user's paths. A limitation of their method is that it can only be used in domains in which users know the names of the searched items in advance so that they can choose the appropriate segments. This excludes situations in which users only know the topics of their search and not the exact titles of their target pages.
2.3. Problem setting

Golovchinsky (1997) presents a method to add in-text links to documents retrieved by a search engine. When the links are clicked the words around the anchor term are used to expand the search query and retrieve a new set of documents. The inverse document frequency (idf) of terms is used to find terms that ‘discriminate well among documents in a collection’ (Golovchinsky, 1997, p. 70). Terms with a high idf score occur in very few documents. As a consequence, when the links that are created in this way are clicked, they provide much information about the user’s targets. However, these links also have a low probability of being clicked so that on average they do not lead to a high information gain. Another difference between Golovchinsky’s approach and the ones presented in this chapter, is that Golovchinsky chooses the links independently. In other words, he chooses the set of best scoring links instead of the best scoring set of links. This can lead to redundant links when multiple links point to (almost) the same set of documents.

Dasgupta et al. (2002) discuss the problem of selecting a set of items for which a user will be asked to provide a rating. The ratings are used to find a user profile that matches the interests of the current user. They give an optimal worst-case upper bound for the number of ratings needed. An algorithm is presented that minimizes the number of ratings needed to find a matching profile by selecting items that discriminate well between user groups. The item selection task is related to menu adaptation as in both cases one has to select the items that provide most information about the users’ interests. However, item selection is a simpler task as it does not require that the selected items are also interesting for the user.

2.3 Problem setting

In the next sections we present three studies in which we investigate possibilities to optimize menu adaptation in various settings. The goal of the first study is to demonstrate the potential of page set division strategies. Simulation experiments are performed in a highly restricted setting. In addition, in this setting we investigate the effects of incorrect assumptions about the probability distribution of the target pages. In the second study we move to a more realistic setting in which some of the restrictions of the first study are relaxed. In a user experiment we demonstrate the effects of page set division on the navigation of human users. Moreover, we study the effects of the amount of navigation mistakes that users make by means of simulations experiments. In the last study we apply page set division to a real web site in actual use. This study examines the benefits of page set division for real users in a situation in which the menu adaptation system needs to compete with static navigation means.

In this section we discuss the various dimensions of the experimental setting. The features of the settings that are used in the three studies are summarized in Table 2.1. Below we discuss each feature in detail.

The first dimension concerns the targets of the users. Menu adaptation methods can assume that users search for exactly one page or that a user’s target information can be spread over multiple pages. In the first study, we use a simple method that does not accommodate for multiple targets. This means that with this method users have
Chapter 2: The role of information gain in menu optimization

The second and third studies use more advanced methods that take the users’ previous targets into account when selecting links so that later targets can be found more efficiently.

The second dimension are the navigation mistakes of the users. The first method assumes that users never make choices that do not match their targets. The other two methods allow for the possibility that users sometimes make navigation mistakes.

Third, in a natural setting menu adaptation systems function besides other navigation means such as static menus and site search engines. In this case the quality of the menu adaptation system depends not only on the quality of the links it adds, but also on the novelty of these links compared to the links offered by the other navigation means. Moreover, in this setting users can choose to ignore the added links and use other navigation means so that the ‘questions’ posed by the added links remain unanswered. In the first two studies we aim to measure the quality of the added links independently of the other navigation means. Therefore, we use a setting in which the added links form the only way a user can navigate through a site. In other words, the links are added to an empty menu. In the last experiment we study menu adaptation systems in a more realistic setting.

In all studies we assume that the a priori probability distribution over the target pages is known in advance. For each page of the site, the adaptation system knows how likely it is that the user is looking for the page when no information about the user has been collected yet. Usually, a priori probabilities can be estimated on the basis of the access frequencies of the pages that are recorded in the server logs of web sites. In these cases, the a priori probabilities reflect the popularity of the pages. When no log files are available, a uniform distribution can be used. In the first study, we examine the effects of incorrect assumptions about the probability distribution on the results of menu adaptation.

To be able to make inferences about the users’ preferences beyond the pages that the user has visited, the methods exploit structures in the set of pages of the site. For instance, the third method uses distances between the pages that reflect the probability that a page will be a target provided that another page is clicked. The type of structure that is used forms a key element of the methods and determines how page set division is performed. The structures will be discussed in detail in the following sections.
2.4 Study 1

In this study we demonstrate the working of page set division in a very restricted setting in which the use of information theory is very natural. In any step in the interaction, the menu adaptation system selects exactly two links that are shown to the user. The user has to select one of these links because no other links are available. The system adds two new links to the requested page and shows the page with the links to the user. The interaction continues until the user reaches his target page.

The method presented in this section relies on page structures called preference scales. It assumes that the pages of the site can be scaled onto one dimension, in such a way that a user prefers pages that are on the scale closer to his target over pages at larger distances. This situation is comparable to a game in which one person has a number between 1 and 100 in mind and another person needs to determine the target number by asking questions like ‘Is the target closer to a or to b?’. The first person can answer these questions because the numbers are naturally ordered on a scale from 1 to 100.

One-dimensional scales can be constructed from a set of preference statements. Suppose that we extract from log data a number of preference indications of a user, such as that he prefers page a over page b and page c over page d. From these data we can construct a preference ordering over the pages of the site. When such orderings are constructed for a number of users, it may be possible to unite the orderings into one one-dimensional scale.

For example, suppose a site about holiday destinations has five pages: France, Spain, Morocco, Denmark and Norway. Two users have indicated their preferences, which are shown in Table 2.2. Even though the users have very different preferences, both sets of preferences comply with the scale that is shown in Figure 2.1. The target of user 1 is Denmark. The target of user 2 is Spain. The scale is such that both users prefer pages closer to their targets over pages further away.

Various methods have been developed to construct scales from preferences. For an overview see Coombs (1964). It is not always possible to construct a perfect scale that reflects the preferences of all users. Most methods introduce a ‘stress’-factor that

<table>
<thead>
<tr>
<th>User 1</th>
<th>User 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denmark &gt; France</td>
<td>Spain &gt; Morocco</td>
</tr>
<tr>
<td>Denmark &gt; Norway</td>
<td>Morocco &gt; Denmark</td>
</tr>
<tr>
<td>France &gt; Spain</td>
<td>Spain &gt; France</td>
</tr>
<tr>
<td>Norway &gt; Morocco</td>
<td>France &gt; Norway</td>
</tr>
</tbody>
</table>

Table 2.2: Example preferences of two users. $x > y$ stands for ‘prefers $x$ over $y$’.

| Norway | Denmark | France | Spain | Morocco |

Figure 2.1: Example of a one-dimensional page scale.
indicates the proportion of the preferences that are not consistent with the scale. In
the rest of this section we will assume the scales are stress-free.

2.4.1 Method

The scale makes it possible to draw conclusions from links that are clicked by a user.
When a user chooses link $a$ from the available links $a$ and $b$, the system infers that the
user’s target is closer to $a$ than to $b$. Therefore, all pages that are closer to $b$ than to $a$
are eliminated as possible targets, i.e. their probabilities are reduced to 0.

The greedy method always shows the two links with the highest probability of
being the user’s target. This means that the system has the largest probability of
leading the user to his target directly. However, when the target is not among the
added links, the system can sometimes eliminate only a small portion of the pages.
For instance, suppose that a greedy system playing the number game suspects that the
user has a preference for higher numbers. In the first step it shows links to numbers 99
and 100. When the user selects 99 (and 99 is not his target), the only numbers that can
be eliminated as potential targets are 99 and 100. In the next step the system shows
98 and 97, etc. In the worst case, the greedy system needs $k/2$ steps to determine the
user’s target, where $k$ is the number of pages on the site.

A more efficient way to determine a user’s target, is to actively maximize the por-
tion of the links that can be eliminated at each step by using a form of binary search.
At each step we select two links that divide the scale into two parts, in such a way
that both sides have equal probability of containing the user’s target. To find two links
that divide the scale in this way, we first determine the center of probability mass: the
point on the scale such that the sums of the probabilities of the pages on each site of
the point are equal. Then, we find two links on the scale that are on either side and
at equal distance from the center of probability mass. When these links are shown to
the user, we can be sure that in each step we eliminate fifty percent of the probabil-
ity mass. In the number example, the center of probability mass is initially located
at 50.5 (when probabilities are uniform). The scale dividing system shows two num-
bers at equal distance from 50.5, such as 20 and 81. When the user selects 20, all
numbers above 50.5 are eliminated. In the next step the center lies at 25.5 and the
system shows, for instance, 10 and 41. With uniform page probabilities, this method
determines a user’s target in no more than $\log_2(k)$ steps.

There can be many pairs of links that are at equal distance from the center of prob-
ability mass. To increase the probability of a direct hit, we choose the pair containing
the link with the highest probability of being the user’s target. Alternatively, one could
choose links at large distance from each other so that users can easily discriminate
between them.

Complexity

Computational complexity is a major issue for menu adaptation methods because the
links must be selected while the user is waiting for his page. In theory, all possible
interaction sequences can be computed in advance, but in most applications this is
not tractable as the number of possible sequences is exponential in the length of the sequences.

Updating page probabilities requires the methods to determine for each page whether it is closer to the page that the user has selected or to the page that the user has not selected. By making use of the page scale this can be done in at most $O(2k)$ steps, where $k$ is the number of pages on the site. The greedy method can select the pages with the highest probability by going through the page probabilities once. As a result, the total time complexity of the greedy method is $O(3k)$. The scale division method can find the center of probability mass by adding up the probabilities of the pages on the scale until they add up to 0.5. In the worst case this takes $O(k)$ time. Finding the page with the highest probability and its counterpart takes another two passes through the page scale. In total, the time complexity is $O(4k)$. Both methods keep the current page probabilities of each user in memory so that the space complexity is $O(uk)$, where $u$ is the maximum number of users that visit the site simultaneously. In conclusion, the time and space complexity of both methods scale linear with the number of pages of the site, which makes them scalable and efficient.

2.4.2 Experiments

We demonstrate the effectiveness of scale division in a series of simulation experiments. The scale division method is compared to the greedy method using various probability distributions of targets. In addition, we study the effects of incorrect assumptions about the probability distributions.

We created an artificial site with 32 pages. Four probability distributions over the pages are considered: decreasing, triangular, uniform and peaked. The distributions are shown in Figure 2.2

In the first set of experiments, we draw 5000 pages from each of the probability distributions. For each page we simulated a user session in which the page was the user’s target. Each session was repeated two times: once with a greedy system and once with a scale dividing system. Both systems knew the probability distribution that was used. In all sessions the preference scale was perfect and simulated users did not make navigation mistakes. This means that the users always selected the links that were closest to their targets.

We counted the number of navigation steps of the simulated users. The results of the experiments are shown in the top four rows of Table 2.3. With the decreasing, triangular and uniform distributions the scale division strategy was on average more efficient than the greedy strategy. Only when the distribution was extremely peaked, the greedy method was faster. With all distributions the scale dividing system had a much better worst case performance. This shows that scale division provides better assistance to users with uncommon targets.

In the second set of experiments, the menu adaptation systems had incorrect assumptions about the distribution of the targets. For instance, in one experiment the systems worked with a decreasing distribution (Figure 2.2(a)), while the actual distribution of the targets was peaked (Figure 2.2(d)). Again, we simulated 5000 users per configuration and measured the numbers of steps needed by the greedy and the scale
Chapter 2: The role of information gain in menu optimization

Figure 2.2: Probability distributions over the 32 pages of the artificial site. The vertical axes represents the probability of being a user’s target. The horizontal axes shows the 32 pages ordered according to the preference scale.

dividing systems. As can be seen in Table 2.3, for all assumed and real distributions the scale dividing system needed on average less steps than the greedy system. The scale dividing system also had the best worst case performance. These findings show that the scale division method is less sensitive to inaccuracies in the assumptions about the probability distribution.

2.4.3 Lessons learned

In our experiments page set division considerably reduced the number of navigation steps needed to determine a user’s target compared to the greedy strategy. Moreover, page set division proved more robust against incorrect assumptions about the page probabilities. These findings show that adding links to the pages with the highest probability of being the users target is not always optimal. This is an important result as the greedy strategy is a very common strategy among systems that adapt link structures.
Table 2.3: The numbers of steps that simulated users needed to reach their target pages with various assumed and real target page distributions. Best scores are shown in bold.

<table>
<thead>
<tr>
<th>Probability distribution</th>
<th>Greedy Average</th>
<th>Greedy Worst case</th>
<th>Scale division Average</th>
<th>Scale division Worst case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>Assumed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decreasing</td>
<td>Decreasing</td>
<td>4.07</td>
<td>17</td>
<td>2.73</td>
</tr>
<tr>
<td>Triangular</td>
<td>Triangular</td>
<td>2.74</td>
<td>10</td>
<td>2.65</td>
</tr>
<tr>
<td>Uniform</td>
<td>Uniform</td>
<td>4.79</td>
<td>17</td>
<td>4.44</td>
</tr>
<tr>
<td>Peaked</td>
<td>Peaked</td>
<td>1.60</td>
<td>17</td>
<td>1.78</td>
</tr>
<tr>
<td>Decreasing</td>
<td>Triangular</td>
<td>3.58</td>
<td>10</td>
<td>2.94</td>
</tr>
<tr>
<td>Decreasing</td>
<td>Uniform</td>
<td>2.97</td>
<td>17</td>
<td>2.77</td>
</tr>
<tr>
<td>Decreasing</td>
<td>Peaked</td>
<td>2.89</td>
<td>17</td>
<td>2.77</td>
</tr>
<tr>
<td>Triangular</td>
<td>Decreasing</td>
<td>5.00</td>
<td>17</td>
<td>2.73</td>
</tr>
<tr>
<td>Triangular</td>
<td>Uniform</td>
<td>2.90</td>
<td>17</td>
<td>2.68</td>
</tr>
<tr>
<td>Triangular</td>
<td>Peaked</td>
<td>2.84</td>
<td>17</td>
<td>2.77</td>
</tr>
<tr>
<td>Uniform</td>
<td>Decreasing</td>
<td>9.50</td>
<td>17</td>
<td>4.91</td>
</tr>
<tr>
<td>Uniform</td>
<td>Triangular</td>
<td>6.00</td>
<td>10</td>
<td>4.97</td>
</tr>
<tr>
<td>Uniform</td>
<td>Peaked</td>
<td>4.69</td>
<td>17</td>
<td>4.59</td>
</tr>
<tr>
<td>Peaked</td>
<td>Decreasing</td>
<td>3.00</td>
<td>17</td>
<td>1.87</td>
</tr>
<tr>
<td>Peaked</td>
<td>Triangular</td>
<td>2.20</td>
<td>10</td>
<td>1.79</td>
</tr>
<tr>
<td>Peaked</td>
<td>Uniform</td>
<td>1.91</td>
<td>17</td>
<td>1.88</td>
</tr>
</tbody>
</table>

In the limited setting that was used in our experiments the scale division method worked very well, but loosening the restrictions of the setting will cause several problems. First, if the scale does not perfectly reflect the preferences of all users, the system will sometimes draw incorrect conclusions about a user's target. The system has no means to recover from these mistakes as pages are eliminated as possible targets for the rest of the session. As a result, users may never reach their targets. A similar problem occurs if users make navigation mistakes (not always click the links that are closest to their targets). Like imperfect scales, these mistakes can cause the system to eliminate target pages. In real applications, these problems are likely to occur as user preferences and navigation are seldom completely regular.

2.5 Study 2

In this section we present a method to divide page sets that is less sensitive to irregularities in user navigation than the scale division method. Moreover, this method exploits information about a user's previous targets to help him find future targets more efficiently. We call this method the information gain method as it directly maximizes the information that is gained in each step of the navigation process.

Instead of depending on a one-dimensional preference scale, the information gain method makes use of keywords that connect sets of pages. The keywords are used to 'question' the user about the topic of his search. For example, the information gain method can show a link named 'Pages related to dizziness'. When the user selects this link, the system learns that the user's target is most likely related to the keyword
Chapter 2: The role of information gain in menu optimization

'dizziness'.

The experimental setting that is used in this section is such that users view at each step either a navigation page or a content page. Navigation pages contain a menu with a fixed number of links and no other links or content. It is the task of the menu adaptation system to choose which links are included in the menu. A menu link is either a link to a content page or a link to another navigation page. Links to content pages have the name of the page as anchor. Links to navigation pages have anchors of the form 'Pages related to keyword', where keyword is some keyword. Because the system can show only a limited number of links, it can happen that none of the presented links is related to the user's targets. Therefore, the last menu link is always named 'None of the above'. This link points to a new navigation page. Content pages contain only content and two links: 'I am done' and 'I want to search more information'. When a user clicks 'I want to search more information', he is taken to a navigation page. If he clicks 'I am done' the interaction ends.

To interpret a click on 'Pages related to keyword' the menu adaptation system needs to know which pages are related to the keyword. We assume that the pages are annotated with keyword meta tags which describe their contents. The keywords might be added by hand by a webmaster or extracted automatically. In Section 2.5.2 is explained how we annotated the pages that we used for evaluation.

2.5.1 Method

The greedy method is employed in the same way as in Section 2.4. In the first step of the interaction, it shows the n links with the highest a priori probability of being a target, where n is the number of links that can be added in each step. If the user clicks 'None of the above', the greedy method shows the next most probable links. The greedy method never shows keyword links as keyword links lead to navigation pages and, therefore, have zero probability of leading the user directly to a target.

At a high level the information gain method resembles the scale division method. It uses page probabilities to find the most informative sets of links. When a user chooses a link, the probabilities of the pages are updated. However, the use of keywords instead of a preference scale changes the way in which these two steps are performed. Below, we first describe how the information gain method updates page probabilities and subsequently how it selects links.

Updating page probabilities

Clicks on keywords are used to update the page probabilities. For instance, if the user clicks a link named 'Pages related to dizziness' the system increases the probability of all pages annotated with the term 'dizziness' and decreases the probability of the other pages. The information gain method thus decreases the probabilities of pages which are not annotated with the chosen keyword, but does not set their probabilities to zero. In other words, pages are never completely eliminated as potential targets. This makes the method robust against inaccuracies in the keyword annotations and
navigation mistakes of the users. To compute how much the probabilities must be adjusted when a user clicks a link, we make the following definitions:

- $c_w$ is the fact that the user clicks on the link ‘Pages related to $w$’.
- $a_w$ is the set of pages annotated with keyword $w$.
- $\overline{a}_w$ is the set of pages not annotated with keyword $w$.
- $P_i(d)$ is the estimation made in interaction step $i$ of the probability that page $d$ is a target for the user.

The probability that one of the user’s target pages is annotated with keyword $w$, $P_i(a_w)$, is given by:

$$P_i(a_w) = \sum_{\{d \in a_w\}} P_i(d)$$

When a user clicks on a keyword $w$, the probability of the pages annotated with $w$ must be increased such that:

$$P_{i+1}(a_w) = P_i(a_w | c_w)$$

Because it is hard to estimate the value of $P_i(a_w | c_w)$ directly, we make use of Bayes’ rule:

$$P_i(a_w | c_w) = \frac{P_i(c_w | a_w)P_i(a_w)}{P_i(c_w | a_w)P_i(a_w) + P_i(c_w | \overline{a}_w)P_i(\overline{a}_w)}$$

We do not have information about the quality of the individual keywords. Therefore, we assume that the probability that a user clicks on a keyword provided that one of his targets is annotated with the keyword is the same for all keywords and in all navigation steps. The probability of clicking keywords that do not appear in the annotation of the target pages (‘incorrect’ keywords) is also assumed to be constant. We represent these probabilities by constants that we name respectively $p_{i,a}$ and $p_{i,\overline{a}}$. In Section 2.5.2 it is explained how we estimate the values of these constants.

With these definitions we can compute how much the probabilities of the pages annotated with keyword $w$ must be changed when $w$ is clicked:

$$P_{i+1}(d) = \frac{P_i(d)}{P_i(a_w)} \cdot \frac{p_{i,a} P_i(a_w)}{p_{i,a} P_i(a_w) + p_{i,\overline{a}} P_i(\overline{a}_w)}$$

if $d \in a_w$

The page probabilities of the other pages are decreased to make sure that the new page probabilities add up to one.

After the update, it holds that $P_{i+1}(a_w) = P_i(a_w | c_w)$. Moreover, we can prove that the probabilities of the pages annotated with $w$ are always increased and the probabilities of the other pages are always decreased:

$$\forall d \in a_w : P_{i+1}(d) > P_i(d) \quad \text{iff} \quad p_{i,a} > p_{i,\overline{a}} \quad (2.1)$$

$$\forall d \in \overline{a}_w : P_{i+1}(d) < P_i(d) \quad \text{iff} \quad p_{i,a} > p_{i,\overline{a}} \quad (2.2)$$

$$\forall d \in \overline{a}_w : P_{i+1}(d) > 0 \quad \text{iff} \quad p_{i,\overline{a}} > 0 \quad (2.3)$$
The conditions in inequalities 2.1 and 2.2 require that the probability that a user clicks a keyword is larger when one of his targets is annotated with the keyword than when none of his targets is annotated with the keyword. It is natural to assume that this is indeed the case. The last inequality states that the page probabilities of pages that are not annotated with the clicked keyword are never set to zero, except when we assume that users do never click incorrect keywords.

A similar update is performed when a user clicks ‘None of the above’. In this case we increase the probability of all pages that are not annotated with any of the presented keywords and we decrease the probability of the other pages. We make use of two constants: \( p_0 \) and \( p_0^* \). \( p_0 \) represents the probability that a user clicks ‘None of the above’ when indeed none of the keyword that were shown appears in the annotation of his target pages. \( p_0^* \) is the probability that a user clicks ‘None of the above’ while in fact some of the keywords that were shown appear in his targets’ annotations.

Until now we assumed that the same target page probability distribution was used for each user and each target. When a target had been found and the user indicated that he wanted to search more information, all target probabilities were reset and a new search was started. We will compare this method to a method that uses information about a user’s previous targets to personalize the probability distribution. This form of personalization draws on the premise that the targets in one user session are often related. When a user has reached a target, the a priori probabilities of pages that are similar to the target are increased and the a priori probabilities of dissimilar pages are decreased. The personalized probability distribution is used as starting point for the search for the user’s next target. We call the information gain method which uses this form of personalization the personalized information gain method. The same form of personalization can be incorporated in the greedy method, resulting in the personalized greedy method. For details on the personalization procedure, see Hollink et al. (2005b).

Selecting informative link sets

At each step the information gain method chooses \( n \) links from the available keywords and links to content pages. During the selection it treats links to content pages the same as keywords: it considers content links as keywords that are associated with exactly one page.

The information gain measure (Quinlan, 1986) is used to estimate the information that is gained by showing a set of links. This approach is similar to the one followed in Witten et al. (1984) (see Section 2.2). However, the use of keywords instead of segments of the alphabet makes our approach suitable for users who do not know the exact titles of their target pages.

The information gain of a question is the difference between the number of bits of information needed to determine the target before and after asking the question. The expected information gain, \( IG \), of a set of added links \( L \) is given by:

\[
IG(L) = H(P) - \Sigma_{l \in L} \left( P(l) \times H(P|l) \right)
\]
Here $P$ is the current probability distribution over the set of pages $D$ and $H(P)$ gives the entropy of $P$. $H(P|l)$ is the entropy of the probability distribution after link $l$ has been chosen. $H(P)$ is given by:

$$H(P) = -\sum_{d \in D} (P(d) \log(P(d)))$$

Ideally, the information gain method would always select the set of links with the highest expected information gain. Unfortunately, computing the information gain of all possible link sets is not always tractable. If $k$ is the number of pages on the site, $w$ is the number of available keywords and $n$ is the number of links that are added in each step, then the number of possible link sets is $\binom{k+w}{n}$. Since the computation must be done online while the user is waiting for his page, heuristics are needed to reduce the number of link sets which are considered.

As a first filter we throw out keywords with a very small probability of being chosen. If a keyword is associated with only one page it is obviously better to provide a direct link to the page than to show the keyword. Therefore, we compute for each keyword the probability that a target page is annotated with the keyword and throw out all keywords with a probability smaller than the average page probability. Furthermore, if it is almost certain that the target is related to some keyword, then clicking this keyword does not provide much new information. For this reason, keywords with very large probabilities are also filtered out.

In a pilot study we compared two heuristics for finding the best link set among the links that remain after filtering (Hollink et al., 2005b). The heuristic that proved most effective uses a form of hill climbing. First, it computes the information gain of all individual links. The $n$ links with the highest information gain are used as start set. Then, one link from the start set is exchanged for another link. If this results in a set with a higher information gain the change is pertained; otherwise it is undone. This exchange process is repeated until no more changes can be tried or until a maximum number of steps is reached. The resulting set of links is presented to the user. Like all hill-climbing methods, this heuristic can converge to a local maximum, but experiments show that in practice it finds good sets of links.

Complexity

Updating the page probabilities consists of two steps: computing $a_w$ for the clicked keyword $w$ (or $a_a$) and computing the new page probabilities. $a_w$ is computed as the sum of the probabilities of the pages annotated with keyword $a$. This takes $O(k)$ time, where $k$ is the number of pages on the site. Computing the new page probabilities requires another $O(k)$ time.

For the computation of the information gain of one link set $L$, the algorithm first computes the probability of each link in $L$. It adds up the probabilities of the pages annotated with the link’s keyword, which can be done in $O(k)$ time. Then, for each link $l$ in $L$, it computes $H(P|l)$, which takes another pass through all page probabilities. $H(P)$ is not computed as this term is equal for all link sets and does not influence the relative ordering of the link sets. In total, computing the information gain of a link set $L$ takes $O(2k|L|)$ time.
Chapter 2: The role of information gain in menu optimization

The heuristic prescribes that we compute the information gain of all individual keywords and page links: $O(2(w + k)k)$, where $w$ is the number of available keywords. The link exchange process (part two of the heuristic) requires the computation of the information gain of link sets of size $n$ for a maximum of $i$ steps, where $n$ is the number of added links per page and $i$ is a parameter that defines the maximum number of steps. This takes $O(2ikn)$ time.

In total, the time complexity of the information gain method is:

$$O(2k + 2kw + 2k^2 + 2ikn)$$

Thus, computation time is linear in the number of keywords and the number of added links per page and quadratic in the number of pages of the site. This means that the method is efficient for moderately sized sites, but can become intractable for very large sites. The parameter that determines the maximum number of exchange steps ($i$) can be used to reduce computation time to a certain extent.

Memory requirements are small. The algorithm stores the associations between pages and keywords and for each visitor the current page probabilities. As a result, space requirements are $O(kw + ku)$, where $u$ is the maximum number of users who visit the site simultaneously.

2.5.2 Experiments

The information gain method was evaluated in three experiments. The first experiment was a simulation experiment. This experiment compared the greedy and the information gain methods when simulated users performed a number of search tasks. In addition, we compared the information gain method with another menu adaptation method that also made use of keyword links, but not information gain. In the second experiment we studied the effects of navigation mistakes on the efficiency of the various methods. In the third experiment we evaluated the real world value of the methods by asking human users to perform the search tasks.

Experimental set-up

We evaluated the menu adaptation strategies on the combined set of pages of two Dutch web sites for elderly people: the SeniorGezond site (SeniorGezond, 2007) and the Reumanet site (Reumanet, 2007). Both sites were developed by The Netherlands Organization for Applied Scientific Research (TNO) in cooperation with domain specialists from the Leiden University Medical Center. SeniorGezond contains information about the prevention of falling accidents. Reumanet contains information about rheumatism. The sites are mainly focused on elderly and volunteers in the care for elderly. The sites have very similar structures: they consist of a set of short texts describing a particular problem or product and a hierarchically structured navigation menu. The menu provides information about the relations between the pages, but each text is written in such a way that it can also be understood in isolation.

From all pages of the two sites we removed the navigation menu and all in-text links. Fifteen texts that were in almost the same form present on both sites were
2.5. Study 2

mapped onto one page. After this mapping 221 unique pages remained, each consisting of a title and some flat text.

Server logs of five months were used to estimate the a priori page probabilities. First, the log data were preprocessed. The sessions of individual users were restored with the method described in Cooley et al. (1999). All requests coming from the same IP address and the same browser were attributed to one user. When a user was inactive for more than 30 minutes, a new session was started. In addition, the agent fields were used to remove sessions of bots. The access frequencies of the pages in the cleaned logs were used as basis for the page probabilities.

We annotated the pages with a number of keywords by means of a hand made domain-specific ontology consisting of 800 terms or phrases and a broader term - narrower term hierarchical relation. We counted for each text and each term in the ontology the evidence that the term was a keyword for the text: the number of times the term or one of its descendants appeared in the text. We annotated pages with all terms with an evidence of at least 2. The domain-specific ontology was created by hand because there was no ontology available for the domain and many of the domain-specific keywords were not in the Dutch version of WordNet (Vossen, 1998). On average each page was associated with 7.7 keywords, with the minimum number of keywords being 1 and the maximum 30.

The quality of the keywords was evaluated in a survey. We had 10 participants read 12 texts and answer 85 questions about these texts. In each question the participants had to choose the word that fitted the text best among 4 keywords or answer that none of the words was appropriate for the text. We found that on average in 60% of the 85 questions the keyword from the page's annotation (the 'correct keyword') was chosen. In 36% of the questions the participants chose 'None of the above' and in 4% of the questions the participants chose a keyword that was not in the annotation of the page (an 'incorrect keyword').

We used the figures found in the survey to set the parameters of the information gain method (see Section 2.5.1). The probability that a user clicks on a correct keyword when there is one, \( p_{1w} \), was set equal to the fraction of the questions in which the correct keyword was chosen (0.60). We approximated the probability that an incorrect keyword is clicked as \( 1 - 0.60 = 0.40 \) because there were 3 incorrect keywords in our multiple choice questions. Assuming this probability is independent of the presence of a correct keyword, \( p_{1w} \) was also set to 0.40. In the survey, the probability of choosing 'None of the above' when there was a correct keyword, \( p_{0\text{c}} \), was 0.36. The probability of clicking 'None of the above' when there is no correct keyword, \( p_{0\text{nc}} \), is set to \( 1 - (0.60 \times 0.36) = 0.95 \) because this is the same as not clicking one of the four incorrect keywords. Table 2.4 summarizes the probabilities.

We defined 12 search tasks. Each task consisted of a short description of a specific problem of an elderly person. The users had to search all pages related to the problem. The topics of the tasks were chosen after consultation of the creators of the sites. We tried to choose problems that were realistic in the domain to get a realistic simulation of the sites' users. For the simulation we defined by hand which pages were in the target sets for the tasks. The tasks had between 2 and 12 target pages with an average of 6.1 target pages. An example of a task description is shown in Figure 2.3.
Chapter 2: The role of information gain in menu optimization

The information gain method and the greedy method were implemented in menu adaptation systems that provided access to the web pages described above. For both methods a personalized and a non-personalized version were created (for details see Hollink et al. (2005b)). This resulted in four menu adaptation systems: a greedy system, a personalized greedy system, an information gain system and a personalized information gain system. The systems provided links while the (simulated) users performed the search tasks. In each navigation step the systems provided four links, in addition to the 'None of the above' link.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{1</td>
<td>a}$</td>
</tr>
<tr>
<td>$p_{1</td>
<td>a}$</td>
</tr>
<tr>
<td>$p_{0</td>
<td>0}$</td>
</tr>
<tr>
<td>$p_{0</td>
<td>1}$</td>
</tr>
</tbody>
</table>

Table 2.4: Parameter settings used for updating the page probabilities.

The information gain method led to significantly lower numbers of clicks than the greedy methods.

Simulated search without navigation mistakes

In the first experiment we evaluated the menu adaptation methods on simulated user behavior. Each simulated user had a set of pages which were his target pages. The target sets corresponded to the targets of each of the 12 search tasks. The simulated users never went to content pages which were not in their target set and when a link to a target page was available they always went there directly. When no links to target pages were available and a keyword from the target pages' annotations was shown, they clicked on the keyword. When also no relevant keywords were shown, they clicked 'None of the above'.

All search tasks were performed 25 times. Table 2.5 gives the average number of clicks the users needed to reach their targets. The table shows that the information gain methods led to significantly lower numbers of clicks than the greedy methods.

1In the simulation experiments significance is computed with a one-tailed paired t-test with a confidence level of 0.95.
2.5. Study 2

This holds for both the personalized and the non-personalized versions of the methods. Personalization significantly improved the performance of the greedy and the information gain method. With personalization the difference between the two methods is smaller, but the information gain method is still 49% faster.

We compared the information gain method to a third menu adaptation method to determine whether the advantage of the information gain method resulted from the use of information gain or from the use of keywords. This clustering method made use of keyword links, but did not select them on the basis of information gain. Instead, it used a conventional clustering algorithm based on page distances to group the potential targets under a common link (for details see Hollink and Van Someren (2006)). No personalized version of the clustering method was created because the clustering method does not use the target page probabilities. The clustering method needed significantly more steps than the information gain method as can be seen in Table 2.5. This shows that the good performance of the information gain method can at least in part be attributed to the use of information gain for selecting keywords.

<table>
<thead>
<tr>
<th>Method</th>
<th>No. steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy</td>
<td>27.7</td>
</tr>
<tr>
<td>Personalized greedy</td>
<td>9.0</td>
</tr>
<tr>
<td>Information gain</td>
<td>8.2</td>
</tr>
<tr>
<td>Personalized information gain</td>
<td>4.6</td>
</tr>
<tr>
<td>Clustering</td>
<td>15.1</td>
</tr>
</tbody>
</table>

Table 2.5: The average number of steps of simulated users without navigation mistakes.

Navigation mistakes

Miller and Remington (2004) showed that clicks on links that do not lead to a user’s target can have a strong negative influence on navigation times. In the next experiment we evaluated the effects of navigation mistakes on the efficiency of the various menu adaptation methods.

We compared simulated ‘perfect’ users that always chose the correct categories with ‘imperfect’ users who sometimes made navigation mistakes. In this setting navigation mistakes are clicks on keywords that were not in the annotation of the users’ target pages. These mistakes occur when users disagree with the developer of the web site about the relevance of keywords.

Navigation mistakes were simulated by adding random mistakes to the behavior of the simulated users. Presented keyword that were not in the target pages’ annotations, had a probability of $\epsilon_w$ of being clicked. When there was a keyword from the pages’ annotations there was a probability of $\epsilon_0$ that the user clicked ‘None of the above’.

We varied the amount of navigation mistakes and measured the effect on the performance of the various menu adaptation methods. The results are presented in Figure 2.4. In this figure mistake level 1 correspond to the values found in the keyword evaluation survey: $\epsilon_w = 0.013$ and $\epsilon_0 = 0.36$. For the other mistake levels these
values were multiplied by the mistake level. The efficiency of the greedy methods is not shown. These methods do not use keywords and therefore are insensitive to the types of mistakes that were added. Figure 2.4 shows that the efficiency of the clustering and the information gain methods decreased rapidly when the users made more mistakes. At the highest mistake levels the path lengths became even longer than the path lengths of the greedy method. At all mistake levels the information gain method outperformed the clustering method. However, the influence of the type of the menu adaptation method appeared to be smaller than the influence of the amount of incorrect choices.

### Human search

We compared the information gain method and the greedy method in a user experiment. Thirteen participants were asked to perform the 12 search tasks. The participants got only the topics of the tasks and not the sets of target pages. Every participant was assisted by two menu adaptation systems, by one during the first 6 tasks and by another during the next 6 tasks. The order of the tasks and the methods was varied over the participants. We measured the number of clicks the participants needed to find the targets and the number of relevant pages that they found.

Table 2.6 shows the results of the user experiment. Users assisted by the information gain methods needed significantly\(^2\) less steps to reach the targets than users assisted by greedy methods. Personalization significantly reduced the number of steps of both the greedy and the information gain method.

The participants did not always understand the keywords that were presented by the information gain methods. 17% of the keyword links that were clicked were not in the target pages' annotations. In 7% of the cases in which 'None of the above' was clicked, there actually was a relevant link among the presented links. Because of these

\(^2\)In the user experiment significance is computed with a one tailed t-test with a confidence level of 0.95.


<table>
<thead>
<tr>
<th>Method</th>
<th>No. steps</th>
<th>No. targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy</td>
<td>17.8</td>
<td>0.9</td>
</tr>
<tr>
<td>Personalized Greedy</td>
<td>9.8</td>
<td>1.7</td>
</tr>
<tr>
<td>Information gain</td>
<td>11.1</td>
<td>1.5</td>
</tr>
<tr>
<td>Personalized information gain</td>
<td>6.9</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Table 2.6: The average number of steps that human users needed to reach a target and the average number of targets that they found per task.

‘mistakes’ the paths of the participants were longer than the path lengths measured in the simulation experiments (Table 2.5). However, our results show that even with these large amounts of mistakes maximizing information gain can be effective.

We did not find large differences between the numbers of target pages that were found. The only significant result was that users found more targets when assisted by the personalized greedy method than when assisted by the greedy method. Probably users of the greedy system were tempted to give up when they saw that they would have to go through the same lists of links again. With all methods the users found very few targets: on average only 26% of the targets was found. Most likely this is a consequence of the limited interface. Many participants reported that they had trouble judging how many relevant pages the site contained because the interface did not provide an overview of the contents of the site. This problem is less likely to occur on real sites, where besides links added by the menu adaptation system also other navigation means are present.

2.5.3 Lessons learned

The experiments show that adding links that maximize information gain can significantly reduce the length of the paths to the users’ target pages compared to a greedy approach. In contrast to the scale division method, the information gain method is not only effective in noise free domains, but also when used by real users who make considerable amounts of navigation mistakes. The information gain and greedy methods were tested on two probability distributions: a personalized and a non-personalized version. On both distributions the information gain method outperformed the greedy method. This confirms the conclusion from Section 2.5 that the advantage of page set division is independent of the probability distribution.

Simulation experiments demonstrated that the effect of navigation mistakes was much larger than the method that was used to select links. Navigation mistakes occur when users cannot predict which pages are connected to which keywords. Consequently, the success of the information gain method strongly depends on the quality and the availability of keywords. This suggests that information gain is a useful criterion to choose between links with equally good keywords, but that the highest priority must be to given to finding links with high quality keywords.
2.6 Study 3

In this section we study the effects of page set division on navigation of real web users. A menu adaptation system based on page set division is incorporated in an existing web site. Compared to the previous two studies, this setting poses several challenges. First, the web site offers several other navigation means, including static menus, in-text links and a site search engine. Consequently, users can choose to ignore the links added by the menu adaptation system and navigate via other navigation means. This means that the added links must not only be informative when clicked, but also invite users to choose them. Moreover, the adaptation system must be careful not to show too many links that are not close to the users' targets. Too many irrelevant links can cause users to feel that the system is not useful and discard the added links for the rest of the session (Cramer et al., 2006). The second challenge is that the web site does not allow keyword links, which excludes the information gain method proposed in Section 2.5. Finally, the web site requires that the menu adaptation system generates links within 0.2 seconds. If computation takes longer, the page is shown to the user without added links. Thus, in this setting it is even more important that the page set division method is computationally efficient.

When keyword links are not possible and pages cannot be scaled reliably onto one-dimension, another structure is needed to make inferences about the users' targets. The method presented in this section uses distances between pages. The distances reflect the similarities and dissimilarities between pages. Closely related pages are at short distance of each other, while very different pages are at large distance of each other. Henceforth this method will be referred to as the distance-based method.

When a site has been online for some time, distances between pages can be computed from the frequencies of the pages in the site’s log files. The distance between two pages is the inverse of their conditional probability:

\[ \text{Distance}(p, q) = \frac{|\text{Sessions}(p)|}{|\text{Sessions}(p) \cap \text{Sessions}(q)|} \]

Here \( p \) and \( q \) are pages and \( \text{Distance}(p, q) \) is the distance from \( p \) to \( q \). \( \text{Sessions}(p) \) is the set of user sessions in which \( p \) occurs. When two pages have never occurred together in a session, their distance is set on a value that is larger than any distance between two pages. Note that this measure is not symmetrical so that it is not a distance measure in the mathematical sense: the distance from \( p \) to \( q \) is not necessarily equal to the distance from \( q \) to \( p \). We do not require symmetry because interest in page \( p \) can be a good indication for interest in page \( q \), while the reverse is not true. When no log files are available content-based distances can be used, for example, based on the overlap between the terms occurring on the pages.

2.6.1 Method

Updating page probabilities

When a user selects a link (either static or added by the adaptation system), the distance-based method infers that the user is probably more interested in the selected
link than in the (other) added links. This knowledge is incorporated in the target page probabilities. For each page \( p \), the method looks up the distance from the selected page to page \( p \) and the distance from the added links to page \( p \). Probabilities of pages that are closer to the selected page than to any of the added links are increased. Probabilities of pages closer to an added link that was not selected are decreased. This process is illustrated in Figure 2.5. The probability of the selected page itself is decreased because the user has already seen this page. The probabilities of the pages that were added but not selected are decreased because the user has seen these links and decided not to click them.

Like the information gain method, the distance-based method decreases probabilities, but does not entirely eliminate pages as possible targets. If, in navigation step \( i \), \( P_i(p) \) is the probability that page \( p \) is the user’s target, then the probability in step \( i+1 \) becomes:

\[
P_{i+1}(p) = \begin{cases} 
P_i(p) + \delta_{\text{other}}, & \text{if } p \neq \text{selected and } p \notin L \\
& \text{and } \forall l \in L : \text{Distance}(l, p) > \text{Distance}(\text{selected}, p) \\
&P_i(p) - \delta_{\text{other}}, & \text{if } p \neq \text{selected and } p \notin L \\
& \text{and } \exists l \in L : \text{Distance}(l, p) \leq \text{Distance}(\text{selected}, p) \\
&P_i(p) - \delta_{\text{shown}}, & \text{if } p \neq \text{selected and } p \in L \\
&P_i(p) - \delta_{\text{selected}}, & \text{if } p = \text{selected} \end{cases}
\]

Here \( L \) is the set of links that were added to the menu in the previous step and \( \text{selected} \) is the page that the user has requested. \( \delta_{\text{other}}, \delta_{\text{shown}} \) and \( \delta_{\text{selected}} \) are parameters. After all probabilities have been updated, the probabilities are normalized by adding or subtracting the same amount to all probabilities. This mechanism does not prevent the probabilities from exceeding 1 or becoming lower than 0. Therefore, these values are not really probabilities, but more ’interest values’. However, to keep terminology consistent with the first two studies, we will still refer to these values as probabilities.

Table 2.7 shows the parameter values that are used in the experiments in Section 2.6.2. The values are chosen in such a way that more certain knowledge leads to larger changes in page probabilities. We are certain that the user has already seen the
selected page and thus that this is not a very interesting link anymore. Therefore, the value of the selected page is reduced most. The (other) pages that have been shown to the user are not clicked. This is a fairly strong indication that the user found the selected page more interesting than the (other) added pages. However, the user may find the other added pages interesting as well. Therefore, their probabilities are reduced less strongly than the value of the selected page. The remaining pages are not shown and their update is based on the distances between the pages. As this evidence is weak, their probabilities are changed only a small amount. The absolute values of the parameters correspond to the speed of the adaptation. When high values are used, the system adapts quickly to the behavior of the user. As a result, after only a few navigation steps the system bases its links mainly on the pages that the user has visited. Low parameter values make the adaptation slower so that the influence of the initial probability values lasts longer. The parameter values in Table 2.7 are based on the above considerations and some small offline experiments. The performance of the method can probably be improved by using parameter values that are optimized for a site’s user population.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta_{\text{other}}$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\delta_{\text{shown}}$</td>
<td>2.0</td>
</tr>
<tr>
<td>$\delta_{\text{selected}}$</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Table 2.7: Parameter values used in the SeniorGezond experiment.

Selecting diverse link sets

When the page probabilities have been updated, the distance-based method chooses the links that are added to the page that the user has selected. The user already has access to the links in the static menu of the page that he is visiting so that there is no reason to add these links once more. Therefore, as a first filter the method removes links that are already visible.

Next, the distance-based method selects pages with a reasonably high probability. This decreases the probability that the user finds the links totally irrelevant and loses his interest in the dynamically added links. All pages with a probability above a threshold are marked as candidate pages. As threshold we can use a fixed value or the average of the current page probabilities, but these thresholds often select too few or too many pages. Instead, we chose to use a threshold that is based on the median page probability. This threshold causes the method to select more pages when there are many pages with high probabilities and less pages when there are less pages with high probabilities.

To get maximal information gain, the set of links that are shown to the user should be spread as much as possible over the information space. Finding the set of pages that are at maximal distance of each other is computationally very expensive. Instead, we use an incremental method that is much more efficient (see the discussion on
complexity below). The first link that is added is a link to the candidate page with
the highest probability. The second link points to the candidate page at the largest
distance from the first page. The third link points to the candidate at the largest total
distance to the first two pages, etc.

Complexity

For the update of the probability of a page $p$, the system needs to look up the distance
between $p$ and the page that the user has selected and the distance between $p$ and
each of the added links. When all values are updated, the system goes through the
probabilities once more for normalization. Therefore, the time complexity of the up-
date is $O(k(n + 1) + k)$, where $k$ is the number of pages of the site and $n$ is the number
of links that are added to each page.

For the computation of the threshold value used to select candidates, the system
has to look at all page probabilities once, which takes $O(k)$ time. Then, candidates
are selected by comparing the probabilities of all pages to the threshold, which again
takes $O(k)$ time. The first link that is added can be found by scanning the values of
all candidates: $O(c)$, where $c$ is the number of candidate pages. In the worst case all
pages are selected as candidates so that $c = k$. To select the other links, we need to
look up the distance between the candidates and the links selected so far. The time
needed for the selection is $O(\sum_{i=1}^{c-1} ik)$, which is equal to $O(0.5n^2k + 0.5nk)$.

The time complexity of the total menu adaptation process is:

$$O(0.5n^2k + 0.5nk + 5k)$$

Thus, time is linear in the size of the site ($k$), which means that the distance-based
method scales very well to larger sites. The time is quadratic in the number of links
that is added to each page ($n$). For all practical applications this is no problem as
this number is usually quite small (typically between 1 and 20). In the experiments
described below, the system almost always generated links within 0.2 seconds.

The memory requirements of the distance-based method are also moderate. At
time it needs to store the distances between the pages and the initial probability
values. In addition, it stores the current probabilities of ongoing sessions. As a result,
the space complexity is no more than $O(k^2 + k + uk)$, where $u$ is the maximum number
of users that visit the site simultaneously.

2.6.2 Experiment

The distance-based menu adaptation method is tested online on the SeniorGezond site
(SeniorGezond, 2007), which was described in Section 2.5.2. When the web site was
launched it included a manually created menu adaptation system. In our experiment
this system served as a baseline to which we compare the distance-based method. The
manually created system showed its links in a small box at the top of the static menu.
The box could contain maximally three links at the time. The links were presented
in the form of recommendations, suggestions for pages that might interest the user.
Chapter 2: The role of information gain in menu optimization

A screenshot of the SeniorGezond site is shown in Figure 2.6. Figure 2.7 shows an enlargement of the recommendation box.

The manually created system based its recommendations on three information sources. First, static relations between the contents of pages were established by experts. When a user viewed a page, related pages were marked as possible recommendations. The second source was the information provided by a questionnaire. The questionnaire asked users questions about their personal circumstances and determined which pages were relevant on the basis of the user’s answers. The relevant pages were possible recommendations in later navigation steps. When the user entered a query in the site search engine, this information was used as the third information source. When the user opened one of the search results, the other search results were no longer visible. As these results could still be interesting, they were included in the list of links that could be shown in the recommendation box. A relevance score was assigned to the possible recommendations and the most relevant recommenda-

Figure 2.6: A screenshot of the SeniorGezond site. The dynamically added links are shown in the recommendation box in the upper left corner of the page (see Figure 2.7).
2.6. Study 3

Figure 2.7: The recommendation box of the SeniorGezond web site. ‘Tip! kijk ook bij’ is Dutch for ‘Recommendation! Look also at’.

...tions were shown to the user. In this sense, the manually created system implemented a greedy method.

Server logs from August 2004 to September 2006 were used to compute page distances and a priori page probabilities needed for the distance-based method. The distance-based method was implemented in a system that could show links in the recommendation box.

For two and a half months the baseline system and the distance-based system ran simultaneously on the SeniorGezond site. Half of the visitors received links from the distance-based system and the other visitors received links from the baseline system. Users were randomly assigned to one of the systems and kept the same system for the duration of the session. The links of both systems looked the same and were placed in the same recommendation box. As a result, users did not know how their links were generated. In fact, they didn’t even know there was more than one system.

Results

As a first indication of the users’ interest in the added links, we look at the number of times the users clicked on links added by either system. The two systems were assigned to almost equal numbers of users. The distance-based system was used to generate links in 8278 sessions, while the baseline system generated links in 8444 sessions (see Table 2.8). However, if we look at the number of times a user clicked a link added by each of the systems, we see a large difference. The distance-based links were clicked 220 times in 115 sessions. The baseline links were clicked only 25 times in 20 sessions. These differences are significant3. Moreover, users who clicked a distance-based link were more inclined to visit another recommended link in the same session.

The percentage of clicks on recommended links is small for both systems. Most likely, the reason for this is that the site offers many other navigation means, including an extensive static menu. With these navigation means most users can find the information they need quite easily (Alpay et al., 2007).

In the following, we assess the influence of the systems on the time that users needed to find their target information. Determining the exact times is difficult be-

3Significance is tested with a two-tailed z-test at a significance level of 0.98.
Chapter 2: The role of information gain in menu optimization

<table>
<thead>
<tr>
<th>Measure</th>
<th>Baseline</th>
<th>Distance-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. sessions</td>
<td>8444</td>
<td>8278</td>
</tr>
<tr>
<td>No. clicks on DALs</td>
<td>25</td>
<td>220</td>
</tr>
<tr>
<td>No. sessions with clicks on DALs</td>
<td>20</td>
<td>115</td>
</tr>
<tr>
<td>Average no. clicks per session</td>
<td>21.6</td>
<td>13.9</td>
</tr>
<tr>
<td>Click on DAL is last click</td>
<td>25%</td>
<td>33%</td>
</tr>
<tr>
<td>Average reading time DALs (seconds)</td>
<td>156.6</td>
<td>24.2</td>
</tr>
<tr>
<td>Average reading time non-DALs (seconds)</td>
<td>73.3</td>
<td>37.9</td>
</tr>
</tbody>
</table>

Table 2.8: Comparison of the results of the distance-based and the baseline system. 'DAL(s)' is short for 'dynamically added link(s)'.

cause we do not know whether users found the information they were searching for. However, the session statistics give a general impression of the efficiency of the navigation. One indication of efficiency is the number of clicks in the sessions. As visible in Table 2.8, sessions in which users clicked on a distance-based link consisted on average of 13.9 clicks. Sessions in which the baseline links were clicked had on average 21.6 clicks. This difference is significant and suggests that the distance-based system is more effective in reducing navigation time.

In various studies the assumption is made that a user stops searching when his information needs are answered so that the last page of a session is the user's target page (e.g. Anderson et al., 2001). If we look at the position in the sessions of the clicks on added links, we see that 33% of the clicks on distance-based links were the last clicks of a session. When the clicks on added links would have occurred at random positions in the sessions, only 16% would have ended a session. This difference is significant. The clicks on baseline links were in 25% of the cases the last clicks, which is not significantly less than the distance-based system, but also not significantly more than random. These results suggest that both systems often showed links to target pages and thus helped users to reach their targets faster. However, an alternative explanation can be that people who had already finished their search decided to take a look at the recommended links.

Another indication of the users’ interest in the recommended links is the time they spent reading the pages that they reached through these links. In Table 2.8 we see that users spent on average 24.2 seconds on a distance-based link and 156.6 seconds on a baseline link, which is a significant difference. This suggests that many of the distance-based links were not as interesting as they seemed so that users left the pages early. However, an alternative explanation for the short reading times arises from the comparison between the sessions of users who clicked distance-based links and the sessions of users who clicked baseline links. This analysis shows that users who clicked baseline links spent on average 73.3 seconds on a page that was not reached via a recommended link. Users who clicked distance-based links spent only 37.9 seconds on a not recommended page. Thus, users who clicked a distance-based link tended to read all pages shortly. This suggest that these users were engaged in a more informal

---

4Significance is tested with a two-tailed t-test at a significance level of 0.98.
2.7. Conclusions and discussion

search (browsing), in which pages were scanned quickly to see whether they contained anything of interest. Nevertheless, even for these users the reading times of the pages reached via the recommended links are short. Note that these results do not mean that the two systems are assigned to different user groups as this analysis involves only sessions of users who actually chose to click a recommended link.

2.6.3 Lessons learned

The SeniorGezond experiment indicates that page set division can help web navigation in practice. The links that were added by the distance-based system were used more often than the links that were added by the greedy baseline system. Moreover, distance-based links were associated with shorter sessions and a higher probability that a recommended link was the last page of the session. This suggests that after visiting a distance-based link more users had found what they were looking for and left the site.

In theory, page set division ensures that the system gains much information about a user’s target. However, this information is only gained when the user clicks one of the added links. This appears to be a major drawback of applying these types of methods in practice. Our experiments show that even though the added links can make navigation more efficient, not many users are inclined to follow these links.

2.7 Conclusions and discussion

Menu adaptation systems often add links to menus of web sites with the aim to minimize the number of clicks that users need to make to find their target information. However, most of these systems in fact do not directly minimize the number of clicks, but maximize the probability of showing a link to a target page at each step. In other words, they follow a greedy strategy.

In this chapter we presented three menu adaptation methods that actively minimize the length of the user sessions. At each step these methods divide the set of pages of a site into parts with more or less equal probability of containing the user’s target. In this way, they balance the costs of collecting more information about the user against the expected gain of the extra knowledge. Evaluation with artificial and experimental data shows that these methods can effectively reduce the users’ numbers of clicks compared to the greedy strategy.

Moving from clean experimental settings to a real web site revealed an important drawback of the presented methods. When selecting the links, the page set division methods do not take the static links of the site into account. As a result, they add links that are optimal when the added links are the only means of navigation, but suboptimal in combination with the site’s static links. We found that when users could choose to use other navigation means, they did not click the added links very often. Consequently, in most of the navigation steps less information was gained than expected. This suggests that menu adaptation can be improved by taking the whole link structure of a site into account. This requires a richer model of user navigation, that tells
Chapter 2: The role of information gain in menu optimization

us how users navigate through a link structure and how the addition of certain links will influence their navigation. This idea will be explored further in the next chapter.

Another issue that is not yet addressed is the optimization of the number of links that are shown on a page. In this chapter we assumed that a user’s navigation time was determined only by the number of clicks he had to make. We did not take into account that it takes time to choose a link from the available links on a page. To fully optimize navigation time a menu adaptation system should find the right balance between the number of links that are shown on a page and the number of pages that the user has to visit before reaching his targets. The models presented in next chapter will allow us to optimize this trade-off.

Another finding in our experiments was that clicks on links that do not lead to the users’ targets cause a dramatic increase in navigation time. Moreover, the influence of these navigation mistakes appeared to be larger than the influence of the type of the menu adaptation method. This finding supports the conclusion of Miller and Remington (2004) that finding links for which descriptive anchors can be found is at least as important as finding an optimal link structure. Improving link anchors is the topic of Chapter 5.