An Adaptive Web Site for the UM2001 Conference

Eric Schwarzkopf
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
Eric.Schwarzkopf@dfki.de

Abstract. We describe the user adaptive features of the UM2001 conference web site as well as their realization. The site provides for each user a personalized portal page, offering a collection of relevant links and news items, based on the user’s interest profile. The interest profile is computed automatically on the basis of the user’s browsing behavior by extracting evidences from the access log of the site, which are then processed using Bayesian networks.

1 Introduction

A visitor should have easy access to the relevant content of a web site. This does not pose a problem as long as the space needed to present the information is small, but as soon as content is distributed over several pages and different subsets of these are relevant to different people, offering easy and fast access to relevant information for everyone becomes increasingly difficult.

One way to solve this problem is to provide for each interest group, or even each visitor, specially tailored portal pages and navigation aids. But then you have a different problem: How to determine the interests of a user? A common method, utilized, for example, by the web sites of CNN and My Yahoo! 1, 2, is to offer to each user the possibility to create a personal portal page on his or her own, but, according to the experiences of Manber et al. [2], “most users take what is given to them and never customize”. Another way is to examine the behavior of a user on a visit to the web site and use this knowledge to create an interest profile. This profile can then be used to adapt the site on the user’s later visits automatically.

Since this latter method is obviously not as exact as the first one, care must be taken to insure that usability of the site is not lowered by the adaptations. We think a web site that provides automatic adaptations should

- allow the user to deactivate adaptive features if these constitute more than a simple addition to the general site

1 http://cnn.com
2 http://my.yahoo.com
allow the user to take a look at his or her interest profile and correct it if necessary.

We used the chance of having to create the web site of the 8th International User Modeling Conference \(^3\) to implement and test these ideas in real life. Although the site is relatively small, it should be interesting to see how people use the adaptive features and what they think about them.

The remainder of this paper describes the features of the UM2001 web site and their realizations. In section 2, we present the adaptive features of the conference site as well as the rationales behind them. Section 3 explains the techniques used to create the interest profiles, section 4 gives some details about the implementation. Sections 5 and 6 present related work and conclude the paper, respectively.

2 Adaptive Features

A web site of an upcoming conference is mainly used to decide whether to participate in the conference and if so, to stay up to date with the latest announcements. To make this tasks as easy and fast as possible, we offer a personalized portal page to each user (see figure 1). This page is dynamically build according to the current user’s interest profile (on the first visit, when no profile is available, a short description of the conference and of the features of the web site is displayed). It offers

- a list of relevant announcements the user has not yet seen (like updates to pages of interest)
- a list of shortcuts to pages not accessible via the top level navigation menu that were visited by the user and in which he or she seems to be interested
- a list of recommended links to pages the user has not yet seen, but which are likely to be of interest according to the user’s profile
- reminders of upcoming events important to the user (like submission deadlines)

Since the portal page should not be too crammed with information, shortcuts and recommendations are ranked and only the top three above an interest threshold are listed. The ranking is done by comparing the interest levels stored in the user’s profile.

Announcements have a priority: They can be rated as not important, normal or important. No announcements that are not important are displayed on the portal page, but all important ones the user has not yet seen are, no matter how many. If announcements are rated as normal, they are ranked according to the highest interest in their subjects (subjects are provided to the system), and the three highest ranked new items, with a rank higher than a set minimum, are displayed. An exception in the normal case are announcements with a very

\[^3\text{http://www.dfki.de/um2001} \]
Fig. 1. The portal page of the UM2001 site offers reminders (not shown), relevant news, shortcuts and page recommendations. The box on the top contains this information in condensed form and is displayed on every page.

High ranking, which means we are almost certain they are of interest to the user. These are listed even if the total number of items will be more than three.

Reminders are tagged with a date and, like announcements, with a subject. If we are in a certain time interval before the date tagged to the reminder, the reminder is ranked and if its rank is above a certain threshold, it is displayed.

Now, it would be cumbersome for the user if it is necessary to return to the portal page each time he or she wants to follow one of the listed links (and “shortcuts” would be quite a misnomer), so the news, shortcuts and recommendations are placed in a condensed form in a message box on top of every page. If the user does not want to see this message box (perhaps screen space is scarce), it can be hidden via a simple mouse click.

The user has access to an abstract view of the interest profile (see figure 2) that was computed based on his or her browsing history on the web site. If this profile is modified by the user, the contents of the portal page and the message box are recompiled to reflect the changes, so that the user gets feedback immediately (although this immediate update is only a simplified version of the standard one, as explained below).

The interest profile is updated only between two user sessions, where a session is delimited by an access pause of about 30 minutes. This guarantees that the information on the portal page does not change while the user is browsing, which could be irritating (imagine you want to check out all recommendations, but as
Fig. 2. The site offers a high level view of the user’s interest profile (the doctoral consortium, for example, has sub-interests like the interest in the dc dates page and the dc committee page, but these are not part of the shown profile), which can be modified by the user.

soon as you visit the first one, others are replaced or rearranged because your profile changed due to this visit). Another advantage of this scheme is that updating the profile does not need to be very fast, because nobody is waiting for the results (if a user revisits the site while an update is in progress, the update is temporarily stopped and the old profile is used to adapt the site). As stated above, an exception to this rule is the immediate reaction to a user’s modification of the profile.

3 Techniques

The main source of information used to create and update the interest profiles is the browsing behavior of visitors. Specifically, what we are processing are the pages that were visited and the links that were followed during a session. Another source of data about interests is the direct feedback from the user, but since we do not want to force the user to provide his or her interests explicitly, the computation of the profile must in general be performed without this precious information.

To allow us to recommend pages the user has not yet visited, we group them according to their subjects (one page may belong to several groups). By estimating the interest in a group based on the interests in its elements, we are able to provide (hopefully) useful recommendations. This grouping leads to an hierarchical structure of the interest profile: Some groups have the same higher level subject, so we can group them again.

To estimate the level of interest, we need a method that
– can easily update an existing profile when new data is available
– can create a useful profile even if there is only very few data available (that is, it should be able to utilize prior knowledge about the domain)
– can use incomplete data to update a profile (a user may visit any subset of pages in a session)

Based on these requirements, we decided to use Bayesian networks.

3.1 Bayesian Networks

Here, we provide only a very short and informal introduction to Bayesian networks. You can find further information in Russel et al. [7] and Pearl [3], among others.

Bayesian networks (also called *belief* or *causal networks*) are directed, acyclic graphs used to economically represent joint probability distributions. The essential idea is to use knowledge about the independence of events to minimize the amount of data needed to store a distribution. Each node in a Bayesian network represents a probability variable, and an edge from node X to node Y can be interpreted as representing a direct influence of X on Y. A node is tagged with a *conditional probability table* (CPT) that provides the distribution of the represented variable given instantiations of the its parent nodes.

Bayesian networks are usually constructed on the basis of a causal model (depicting causal interrelationships between events) of the domain that is to be represented. As a concrete example, take a look at figure 3, (a). This net models the following thesis: Interest in *Paper Submission* causes interest in the Call for Papers and in the Submission Page.

![Bayesian Network Diagram](image)

*Fig. 3. In (a), the Bayesian network used to estimate interest levels in the paper submission subjects is shown. The numbers on the right of each node represent probability distributions of the variables corresponding to the nodes (the probability of interested/not interested given certain evidences). In (b), some dummy nodes were added that inject new evidential support extracted from the access log into the core network.*

If we observe the state of some variables (the *evidence*), we can use a Bayesian network to compute the probability distributions of another variable given that
evidence. To this end, we need to know the level of influence parent nodes have on the probability distributions of each of their children, that is, we need to know all conditional probabilities $P(x|H)$, where $x$ is a state of the child and $H$ a state of the set of its parents. If a node has no parents, an a-priori probability distribution of the represented variable must be known instead. Having this data and some evidence, the Bayesian rule is used to update the probability distributions throughout the network. The Bayesian rule states that $P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$.

A simple example should give you an idea about how this works: Let us say we have two probability variables $X$ and $Y$, which can take on the values true and false. $Y$ has a direct influence on $X$ and we have observed $X$ to be currently true. Assuming we know the a-priori distribution of $Y$ and the conditional probabilities $P(X|Y)$, we can compute $P(Y = \text{false}|X = \text{true})$ and $P(Y = \text{true}|X = \text{true})$ in the following way:

$$P(Y = \text{false}|X = \text{true}) = \frac{P(Y = \text{false}|X = \text{true}, Y = \text{false}) P(Y = \text{false})}{P(X = \text{true})}$$

$$P(Y = \text{true}|X = \text{true}) = \frac{P(Y = \text{true}|X = \text{true}, Y = \text{true}) P(Y = \text{true})}{P(X = \text{true})}$$

$P(X = \text{true})$ is not given, but we can compute it:

$$P(X = \text{true}) = P(X = \text{true}|Y = \text{false}) \cdot P(Y = \text{false}) + P(X = \text{true}|Y = \text{true}) \cdot P(Y = \text{true})$$

Hence, we get the updated distribution of $Y$ in respect to $X = \text{true}$.

As you can see, Bayesian networks fulfill all our requirements.

### 3.2 Semantics

Our Bayesian networks reflect the hierarchical structure of the interest profiles (this structure is interpreted as a causal model: interest in a subject group causes interest in the contained subjects, which in turn causes interest in the respective pages). We use probability variables with only two values, not interested and interested, associated with 0 and 1, respectively. We interpret the expectation of such a variable as the level of interest in the respective subject. This means we map an interest to a value in $[0, 1]$, since the expectation simplifies to the probability of interested. This makes comparing interest levels straightforward, which simplifies the ranking process as well as the estimation of probability values (see below). Since we do not know the interest levels of a new user, we initially use uniform probability distributions (that is, all interests have a value of 0.5 in the beginning).

To determine the conditional probabilities, you would normally analyze a relevant amount of empirical data, but since there was no data available, we had to come up with some useful values on our own. As the interest levels are used to rank recommendations, only the relative values of these interest levels are relevant to our system. Therefore, we first rated correlations between variables
qualitatively ("very strong", "strong", "average", "low", "very low", as well as "positive" and "negative"), mapped them to values within intervals determined by this rating (see Table 1) and made some trial runs with the resulting networks.

<table>
<thead>
<tr>
<th></th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>very low</td>
<td>(0.5, 0.6)</td>
<td>(0.4, 0.5)</td>
</tr>
<tr>
<td>low</td>
<td>(0.6, 0.7)</td>
<td>(0.3, 0.4)</td>
</tr>
<tr>
<td>average</td>
<td>(0.7, 0.8)</td>
<td>(0.2, 0.3)</td>
</tr>
<tr>
<td>strong</td>
<td>(0.8, 0.9)</td>
<td>(0.1, 0.2)</td>
</tr>
<tr>
<td>very strong</td>
<td>(0.9, 1.0)</td>
<td>(0.0, 0.1)</td>
</tr>
</tbody>
</table>

Table 1. Conditional probabilities for use in the Bayesian networks are estimated by mapping the qualitative correlations between variables to values in the respective interval.

Although this constitutes quite an ad hoc approach, the system displayed the intended behavior after only a very small number of changes to the initial conditional probability estimates. The small networks used are certainly the main reason for this, so this simple approach will not scale to larger systems.

3.3 Updating the interest estimates

To update the probability distributions in the networks after each session, the influences of relevant user actions from the access logs are quantified similar to the conditional probabilities as explained above. This new information is fused with all previous evidence by "injecting" it into the network via a dummy node ([3], page 170). The one and only job of a dummy node is to cause an update of the network, so we remove it as soon as the update is completed. In effect, we have a core network that is used to store the interest levels and to update these interests according to past and current evidences, and we have dynamic dummy nodes that inject evidences at the appropriate places (see figure 3, (b)).

If the user changes his or her profile via the forms on the web site, we update the probability distributions of the corresponding variables, but since propagating changes through the Bayesian networks takes some time, the actual propagation is done not immediately, but after the user has finished the session. To provide instant feedback to the user, so that he or she can see what the settings will do, we simulate an instant profile update by temporarily setting the interests according to simple rules of thumb. As an example, if a user lowers the level of interest in Paper Submission, the levels of interests in the Call for Papers and the Submission Page are lowered accordingly (ignoring any former evidence).

By allowing the user to modify the interest profile, it is possible that the changes made contradict the model used to construct the networks. To illustrate this, imagine that there are two interests X and Y whose level can be set by the user. The network containing the nodes representing X and Y was
constructed under the assumption that $X$ has a positive influence on $Y$ (i.e., $P(Y = \text{high} | X = \text{high}) > 0.5$). Now, the user sets the interest in $X$ to a high level. The interest in $Y$ will be decreased while the interest in $X$ will be increased due to the positive correlation. In the next session, the user corrects the levels of interest, but the "wrong" network will again cause the values to deviate. This is definitely annoying. To prevent this situation, we implemented a rudimentary network trainer. This trainer looks for feedback data that contradicts the network assumptions and modifies the conditional probabilities.

The trainer compares the difference of the levels of interest in two correlated variables $X$ and $Y$ before the change with the difference after the change. If the latter is bigger, this is interpreted as evidence for the correlation in the network being too strong, so this correlation is reduce by a predetermined step width. If, on the other hand, the difference after the change is smaller, the correlation will be increased. We use relatively large step-widths (there are five steps from the minimum to the maximum correlation, corresponding to the "quality classes" of table 1), so that the network is adapted very quickly.

4 Implementation Details

Figure 4 shows the high level architecture of the system. The profiler and the network trainer are implemented in Java, since we use Fabio G. Cozman’s JavaBayes 4 to handle the Bayesian networks. The other components are implemented in C++. Most of the work is done in plug-ins that can be added to the content supplier component dynamically (the implementation is based on [6]). The plug-ins are called via special tags in the HTML pages. These tags are replaced in the content supplier by the output of the specified plug-ins, so the clients only receive standard HTML documents. Since every page request is processed by the content supplier component, we decided to use FastCGI 5 to speed up response time: As a FastCGI program, the content supplier awaits and processes requests instead of being restarted with every access.

5 Related Work

Perkowitz and Etzioni [4] formulated the challenge to the AI community to address the problem of web site design by creating adaptive web sites: "web sites that automatically improve their organization and presentation by learning from visitor access patterns." In contrast to adapting the site to the needs of individual users, their approach is to optimize the structure of the site based on interactions with all visitors [5].

4 http://www.cs.cmu.edu/~jawabayes
5 http://www.fastcgi.com
Fig. 4. *High level architecture of the UM2001 web site system.*

SETA \(^6\) is a shell which supports the creation of adaptive web stores. Web pages presenting products are dynamically generated based on the current visitor’s user model, which is far more elaborate than the one used in the UM2001 system. The user model is initialized by utilizing a stereotype knowledge-base together with data that the system explicitly elicits from the user. Then, the model is updated automatically on the basis of the user’s behavior and product selections \([1]\).

An example for the use of Bayesian networks in user modeling is the project READY \(^7\) by Jamson et al. In READY, it is investigated how an assistance system can recognize the user’s current time and working memory limitations on the basis of the user’s behavior and adapt its own behavior appropriately \([8]\).

6 Conclusion

In this paper, we described the ideas and techniques behind the realization of the (mildly) user adaptive UM2001 web site. This site offers to each visitor a personalized portal page and personalized navigation aids, which are created based on

\(^6\) http://www.di.unito.it/~seta
\(^7\) http://www.coli.uni-sb.de/sfb378/projects/READY-en.html
the visitor's interest profile. Bayesian networks are used to automatically create the interest profiles from evidences collected from the access log of the site, so a user does not need to fill out any forms to benefit from the adaptive features, although he or she is still able to modify the profile by hand.

We hope the personalized features add value to the web site, and indeed, first user reactions are quite positive. A survey form will be offered on the site when the users have had some experience with the system working on the complete set of conference data (data about paper and poster sessions was not yet available at the time of writing). It will be interesting to see what people liked or did not like about the web site, and how the adaptations worked for them.

Acknowledgements

I thank Mathias Bauer and Dietmar Dengler who supervised this work. It was supported by the Federal Ministry of Education and Research (BMBF) under grant 01 IW 001 as part of the MIAU project.

References