Adaptive Personalization for Mobile Content Delivery

Daniel Billsus

FX Palo Alto Laboratory
3400 Hillview Avenue, Bldg. 4
Palo Alto, CA 94304
billsus@fxpal.com

Raymond Klefstad, Michael J. Pazzani

Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697
{klefstad, pazzani}@uci.edu

Abstract

While personalization has proved to be an important supplement to web applications, the constraints of mobile information access make personalization essential to producing usable applications. Mobile devices, such as cell phones or personal digital assistants, have much smaller screens, more limited input capabilities, slower and less reliable network connections, less memory and less processing power than desktop computers. We discuss an adaptive personalization technology that automatically delivers personalized information available to the mobile user via wireless or wired synchronization on platforms such as AvantGo or Qualcomm's BREW™. Both of these platforms have capabilities not available in most browsers for wireless devices. In particular, they allow for local storage of some content on the mobile device that may be accessed without wireless connectivity. Our applications attempt to optimize the batch download of information to wireless devices so that the delays and costs associated with interactive browsing are reduced. We present evidence that the personalization algorithm increases the usage of mobile content applications by displaying personally relevant information to individual users.

1 Introduction

Although browsing and searching are acceptable methods of locating information on the wired web, these operations soon become cumbersome and inefficient in the wireless setting. In this paper, we discuss a news client for mobile phones and PDAs. The news client displays section names (e.g., Sports), headlines, and articles. It is intended to be used by a single news site to deliver its content to readers of that site, rather than aggregating news across multiple sites. Personalization reorders sections of the news site so that the most frequently accessed sections may be accessed without scrolling, reorders headlines within a section so that the most personally relevant items are displayed toward the top, selects headlines for display on the front page, and optionally determines which articles are downloaded and stored on the phone. Downloaded content can then be browsed at any time during the day without any additional wireless charges or delay. The application provides value to consumers by allowing them to use off-peak minutes or synchronization when connected to a desktop computer to download content. We discuss two implementations of this general approach on two different device architectures.

Our first application is a browser implemented on Qualcomm's BREW™ platform, a programming environment that allows small client applications to be installed on the phone and interact with servers over a wireless network. The client on the mobile device is responsible for providing a unique but anonymous identifier for each user, requesting content from the server, displaying content, keeping track of the content that a user selects and ignores, and optionally waking up at a user-defined time to request the batch download of content.

Our second application provides access to personalized content on any mobile device that supports the AvantGo® platform. This platform consists of a client-server framework that enables content synchronization for offline reading, as well as interactive wireless access. It has some basic capabilities for saving information on the mobile device, such as form submissions, that are transmitted to the server on the next synchronization.

In each case, the server is responsible for automatically learning the preferences of the user and locating content that matches these preferences. The client is responsible for displaying content to the user, keeping track of what the user has selected, and communicating this information to the server. In combination, the client and server automatically personalize content and increase the usage and acceptance of mobile information access through content that is easily accessible and personally relevant.

2 The BREW™ Environment

While many wireless devices have a browser for a simplified mark-up language such as WML or XHTML, BREW™ is a general purpose programming environment with capabilities not present in browsers, such as the ability to set alarms and wake up a program at a specific time, or the ability to write to a local file system. BREW™ provides an API for user interface, file system and network functions that free the programmer from thinking about the capabilities of different phones from different manufacturers.

Nonetheless, BREW™ applications must be carefully designed to live within the constraints of a mobile device. To give an idea of the constraints a mobile application faces, an early entry-level phone had 80K of RAM, a stack of 500 bytes, 1Mb available for storing programs and temporary files, and a monochrome display of 90 by 100 pixels. Although network throughput is typically less than 20kbs, network latency often presents more of a problem for user interfaces than bandwidth. As a consequence, storing some articles on the phone dramatically improves the usability of the news client by eliminating network delays.

The goals of BREW™ are similar to Sun's Java-based J2ME environment. Although our solution is implemented in BREW™, it would be straightforward to port it to J2ME or any wireless application environment.

3 Mobile News Delivery for BREW™: Solutions Considered

One alternative considered for delivering news to a BREW™ enabled device was to create a general HTML browser that simply starts at a particular news site. This has the advantage of leveraging existing infrastructure, but it does not take into account the constraints of the mobile device. For example, due to the memory, latency, and screen size constraints, transmitting a full-length article including images at once isn't practical. We decided to send only the text of articles and text associated with headlines and section names. Since the full power of HTML wasn't needed, we designed a news-specific mark-up language with tags for headlines, sections, and article bodies. The body of an article is broken up into multiple pages to reduce the latency of displaying the initial section of an article and the RAM required by the application. The control flow, e.g., moving between pages of an article, from an article to a section of headlines or the front page is built into the application, eliminating the need to transmit an encoding of this functionality in the mark-up language associated with each screen.

Since so little content fits on one screen, personalization is important in the mobile environment. Many wireless carriers offer a limited degree of personalization where users customize their menus by visiting a web site and selecting the order in which items appear or indicating topics of particular interest. However, most carriers report that only 2-5% of users customize their experience. Several factors contribute to the limited usage of such approaches. First, it is fairly complicated to create an account on the web and associate that account with a wireless device, limiting the audience to the technically sophisticated. Second, the categories tend to be very coarse-grained, allowing users to select from general categories such as "International News" rather than specific categories such as "telecom industry in Korea." Third, topic profiles require maintenance to be useful. Many people don't regularly follow sports, but gain interest in championship games. Others closely follow particular teams but change their reading habits during the Olympics. Few users are willing to continually maintain their customized profile on a regular web site for an optimal wireless web experience. Finally, such approaches may prevent wireless users from gaining access to information outside their profile without going back to a wired web site and adjusting the settings. This clearly is in conflict with the goal of being able to access any information at any time from any location.

Explicit customization approaches, such as web-based questionnaires, place the burden of personalization on the user. In contrast, we implemented an adaptive personalization solution: an automated approach that uses artificial intelligence and statistical techniques to construct a model of each user's interests. It is important that adaptive systems learn from a few examples and adapt quickly to changing user interests. To be truly useful in a mobile context, users' interests must be inferred implicitly from actions (Claypool et al., 2001) and not obtained exclusively from explicit content ratings provided by the user. For example, the more a user reads of news story, the higher their level of interest. Obtaining explicit feedback about user interest requires a user interface consuming screen real estate and additional data to be transmitted. We also argue that the adaptive interface should not limit the user's choices by filtering information, but rather order it so that the most relevant information is at the top of each screen.

4 Adaptive Personalization for News

We believe that for adaptive personalization to meet the tough test of user acceptance, it must be designed with certain constraints in mind:

- Provide a good initial experience and learn quickly for new users. The first use should provide an acceptable, non-personalized experience. The benefits of adaptive personalization must be apparent within the first few uses if users are to return. Furthermore, the transition from a non-personalized to a personalized experience must be smooth without going through a state of presenting random items to the user. Most of the results in information categorization (e.g., Pazzani & Billsus, 1997; Yang, 1999) report on asymptotic performance. However, in an adaptive interface, the performance on a handful of examples is very important: to be useful, the service must adapt quickly to changing interests. For example, users frequently change their reading habits due to external events. A user who skips 160 stories on a baseball team during the regular season and reads about its victory in 4 league championship games is probably more than 4/164ths interested in the World Series.

- Avoid tunnel vision. Personalization should not get in the way of finding important novel information or breaking news stories.

- Do not require hand tagging of content with category labels. A process that requires the content provider to do extra work by hand, such as adding meta-tags with category labels is not feasible with thousands of new items being added daily.

- Avoid brittleness. A single action, such as selecting something accidentally or skipping over an article on a topic (perhaps because one heard about the details on the radio or the wireless connection is dropped) should not have a drastic and unrecoverable effect on the presentation.

We have found that a combination of similarity-based methods (e.g., Cover & Hart, 1967), Bayesian methods (e.g., Duda & Hart, 1973) and collaborative methods (e.g., Billsus & Pazzani 1998; Pazzani, 1999) meet the above requirements by achieving the right balance of learning and adapting quickly to changing interests while avoiding brittleness. However, we take editorial input into account in ordering content for delivery by boosting the priority of lead stories. The effect of this boosting is that first-time users of the wireless news site see articles in the same order as on the wired web site and that all users always see the lead story in each section. This also allows the adaptive personalization engine to learn more about each user. Finally, the similarity-based methods also are used to make sure that a variety of news articles are presented on each screen in much the same way that a newspaper does not fill the front page with articles on the same topic.

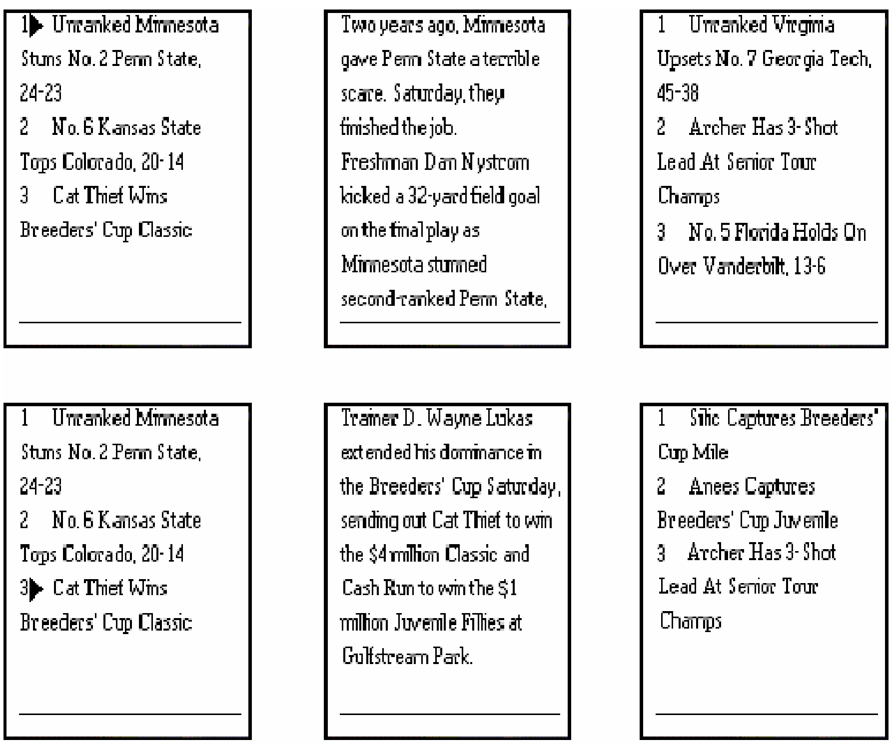

Figure 1: Two different user interactions illustrating the effects of personalization

Figure 1 illustrates how the system adapts to two different users. Both are shown the same three headlines initially. On the top row, a user reads a college football story and when the next page of headlines is requested, additional college football stories are shown. In the bottom row, a user instead reads a horse-racing story and is shown additional stories on this topic. In each case, a golf story is included on the next page, both to allow some diversity in the stories and to present the system with the opportunity to learn more about the user.

5 Batch Personalization for AvantGo® and BREW™

On both the AvantGo® and Brew™ versions of the application, our clients have the capability of downloading a group of news headlines and articles at once. The Brew™ application may be configured so that it wakes up at a set time each day (e.g., early in the morning at off-peak rates) and wirelessly downloads content. The AvantGo® platform does this whenever the user synchronizes content by connecting the device to a desktop or laptop computer. To personalize content, the client must keep track of the user's content access history. For example, for each news headline presented to the user, our clients keep track of the number of corresponding story pages accessed by the user. This counter serves as an indicator of the user's level of interest. Specialized logic was written in our Brew™ application for this purpose. On the AvantGo platform, similar functionality can be easily realized by storing information about button clicks (e.g. implemented as form submissions) on the mobile device, and transmitting the stored information to the server during the next synchronization.

Conceptually, our server supports both batch download clients in the exact same way. A request for new content to store on the mobile device is initiated by first transmitting the user's most recent content access history to the server. The server then uses this information to update its model of the user's interests, as described in Section 4. The modified user model is then used to rank the set of available stories to determine a personalized subset, which is transmitted back to the client, where it is stored to allow offline access.

To preserve memory on the mobile device, content sent to the client may include headlines without stories or only the first page of an article. If the user wants to read content that is not present on the device, the client on each platform retrieves it wirelessly.

6 Evaluation

Our personalization server reorders news stories with respect to users' individual interests. The main intuition is that such a modified order helps users access relevant content. However, information is rarely presented in random order. For example, editors prioritize news stories based on human judgement, which means that, in this case, users access content in an order deemed appropriate by human professionals. While such an order is static in the sense that it is the same for every user, it is possible that it is sufficient for most users to easily access relevant content. In this section, we present results from two studies that compare personalized information access provided by our personalization server to static information access. These results show that the system's user modeling algorithm generates an adaptive order that has two closely related effects: it simplifies locating relevant content and leads to an overall increase in accessed information. The main idea underlying both experiments is to present items either in static or adaptive order so that resulting differences in users' selection and browsing behavior can be quantified. Comparing static and adaptive news access requires disabling all personalization capabilities when the system is in static mode, which clearly reduces the system's utility. Therefore, the experiments were conducted over a short period of time in order to reduce potential user dissatisfaction: two weeks for experiment #1 and four weeks for experiment #2.

The system's ultimate goal is to simplify access to interesting content. A simple and informative measure that quantifies progress towards this goal is the average display rank of selected stories. If the system successfully learned to order items with respect to users' individual interests, this would, on average, result in interesting stories moving toward the top of users' personalized lists of items. Therefore, the average display rank quantifies the system's ability to recommend interesting items. Since this measure does not depend on a predicted numeric score or classifying news stories based on predicted interest levels, it is possible to apply it to static information access allowing for a comparison of both strategies.

The "alternating sessions experiment" quantifies the difference between static and adaptive information access by randomly determining whether a user receives content in static or adaptive order. During a period of two weeks, our personalization server used its user modeling approach for approximately half of the users, while the other half received news stories in static order determined by an editor at the news source. On odd days, users with odd account registration numbers received news in personalized order and even users received a static order. On even days, this policy was reversed. To quantify the difference between the two approaches, we measure the mean rank of all selected stories for the personalized and static operating modes. Since a difference between static and adaptive access can only be determined for users that previously retrieved several stories, we restrict the analysis to users with a minimum of five selected stories. Comparing both access modes for this subset of users revealed a significant difference. The average display rank of selected stories was 6.7 in the static mode and 4.2 in the adaptive mode (based on 50 users that selected 340 stories out of 1882 headlines). The practical implications of this difference become apparent by analyzing the distribution of selected stories over separate headline screens (every screen contains 4 stories). Figure 2 illustrates these two distributions visually. In the static mode, 68.7% of the selected stories were on the top two headline screens, while this was true for 86.7% of the stories in the personalized mode. It is reasonable to argue that this makes a noticeable difference when working with handheld devices, and we interpret this result as promising evidence for the utility of adaptive content access. In addition, this result suggests that effective personalization can be achieved without requiring any extra effort from the user.

While the reported results are promising, the described experiment has several shortcomings. First, the results are based on a small data set, consisting of only 50 users who selected 340 stories (since users can likely perceive the effect of disabling personalization, data collection was limited to two weeks). Second, due to the high cost of information access on wireless devices (e.g. many cell phone plans treat data access just like regular voice minutes) users typically only select a small number of headline screens in each session. It is likely that users select from these few screens the stories that interest them most, and that this is true for both the static and adaptive access modes. Therefore, a drawback of the "alternating sessions experiment" is that users might not see stories they would have seen in the adaptive mode. Likewise, in the adaptive mode, users might not see stories they would have seen in static mode. The following experiment addresses this problem by displaying both adaptive and static stories on the same screen.

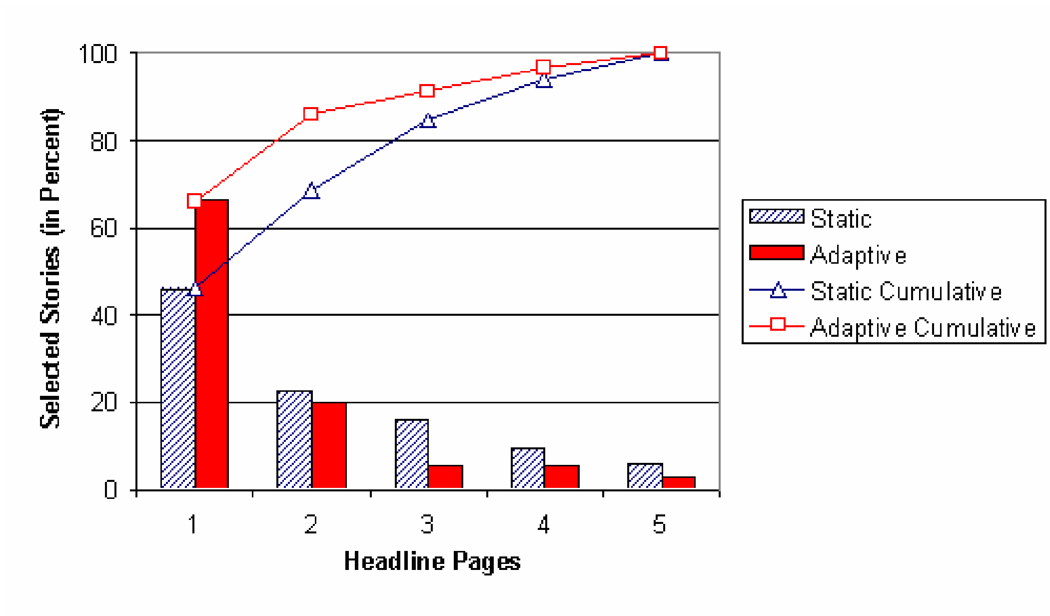

Figure 2: Distribution of selected stories (alternating sessions experiment)

The "alternating stories" experiment is similar in principle to the "alternating sessions" experiment, i.e. it is designed to quantify the difference between static and adaptive information access. However, the "alternating stories experiment" displays stories selected with respect to both the adaptive and static strategies on the same screen. During the experiment the client was configured to display four stories on each screen, with every screen containing two adaptive stories and two static stories. The server determines randomly if the first displayed story is a static or adaptive story, and the remaining stories are selected by alternating between the two strategies. The "alternating stories" methodology has two advantages. First, the system still adapts to the users' interests, because every screen contains two stories that were selected adaptively. This results in a change of system behavior that is much more subtle from a user perspective than the resulting change of the "alternating sessions" experiment. Therefore, it is possible to run the experiment over a longer period of time, because all users still receive a useful service. Second, users see the current top-ranked adaptive and static stories on the same screen, allowing for a direct comparison between the two selection strategies. If the system learns to adjust to users' individual interests, users can be expected to select more adaptive stories when presented with a choice between adaptive and static content.

Our personalization server used the "alternating stories" methodology over a period of four weeks to collect access data for 5000 adaptive stories and 5000 static stories that were shown to users who had previously selected a minimum of 5 stories. Using these criteria, data obtained from 222 different users were included in the experiment. Similar to the "alternating sessions" experiment, the average display rank can be used to quantify the difference between the two display strategies. However, using the "alternating stories" methodology, the difference between the two average display ranks was not as pronounced as in the "alternating sessions" experiment: 5.8 for the static mode vs. 5.27 for the adaptive mode. Likewise, the distributions of selected stories over consecutive headline pages revealed only a small difference between the two display modes: for the static mode, 75.57% of the selected stories were on the top two headline screens, while this was true for 80.44% of the stories in the adaptive mode. We attribute the smaller difference between the two modes mainly to the presence of adaptive stories on every page. As a result, the user's information need might be satisfied after seeing only a small number of headline pages. If users do not have to request multiple screens to find relevant information, the observable difference in display ranks is reduced. However, this explanation only holds if users indeed select more adaptive stories than static stories. The percentage of selected stories for the two display modes clearly indicates that users are more likely to select adaptive stories than static stories. In particular, users selected 13.26% of all displayed static stories (663 stories), vs. 19.02% (951 stories) of all displayed adaptive stories, which amounts to a 43.44% increase in selected content. Figure 3 shows how this difference is distributed over consecutive news headline screens. For each headline screen, this plot compares the probability that a selected story was an adaptive story to the probability that the story was presented in static order. More formally, the plot compares the conditional probabilities p(adaptive | selected) and p(static | selected) for separate headline screens.

Figure 3 also shows that the difference in selection probabilities is particularly noticeable on the first headline screen and then decreases gradually from page to page. On the first headline screen, p(static | selected) is 0.33 vs. 0.66 for p(adaptive | selected). This difference indicates that the adaptive display strategy indeed helps users locate relevant content, as users prefer adaptive stories over static stories on average.

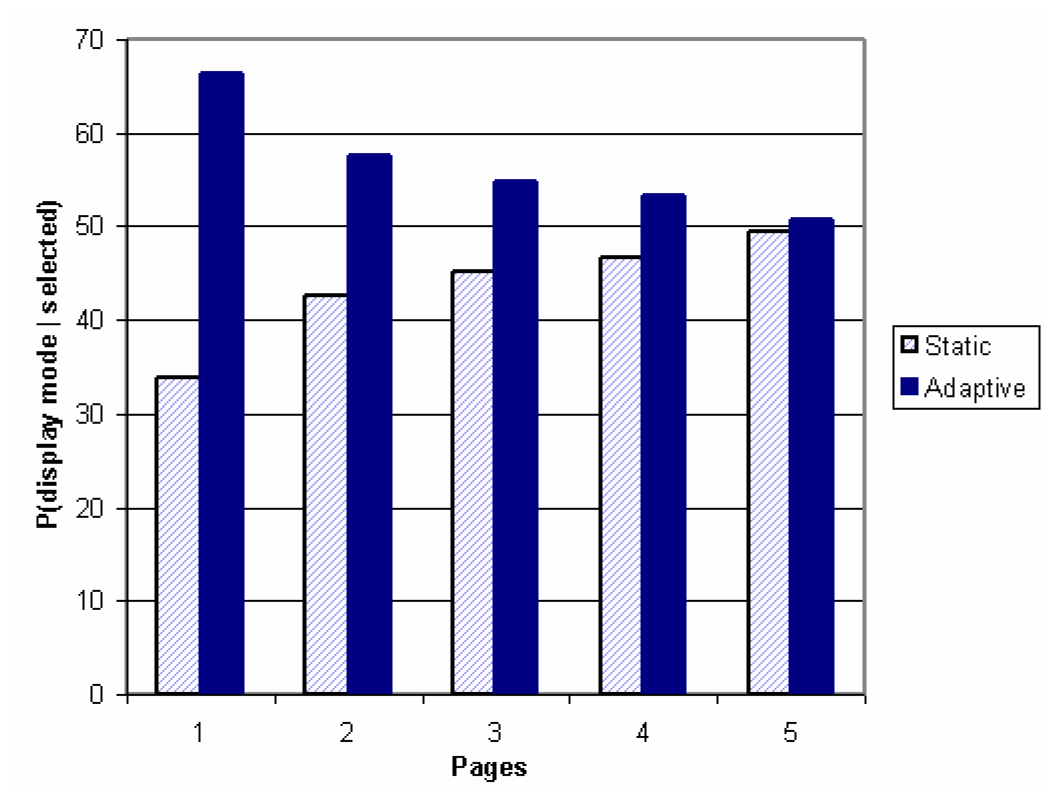

Figure 3: Static vs. adaptive selection probabilities

In summary, the “alternating sessions” and “alternating stories” experiments both show that adaptive information access is superior to static access. The “alternating sessions” experiment demonstrated that the adaptive order helps to move interesting items towards the beginning of personalized item lists, simplifying access to relevant content. The “alternating stories” experiment showed that the system is capable of ordering content in a way such that the top-ranked stories have a significantly higher chance of being selected than the top-ranked stories obtained from a static order.

7 Conclusion

A literal translation from the web to wireless is inadequate. Merely squeezing data onto small screens detracts from the user experience on mobile devices. By adaptively learning users’ preferences, all users can have easy access to a vast amount of information at any time.

8 References

Billsus, D., & Pazzani, M. (1998). Learning collaborative information filters. Proceedings of the Fifteenth International Conference on Machine Learning (pp. 46-54). Madison, WI: Morgan Kaufmann.

Billsus, D., & Pazzani, M. (2000). User Modeling for Adaptive News Access. *User Modeling and User-Adapted Interaction*, 10(2/3): 147-180.

Buchanan, G., Farrant, S., Jones, M., Marsden, G. Pazzani, M &, Thimbleby. H. (2001). Improving Mobile Internet Usability. In Proceedings of the Tenth International World Wide Web Conference, Hong Kong, pp. 673-680.

Cheverst, K., Mitchell, K., and Davies, N. (2002). The role of adaptive hypermedia in a context-aware tourist GUIDE. *Communications of the ACM*, 45, 5.

Claypool, M., Le, P., Wased, M. and Brown, D. (2001). Implicit interest indicators. Proceedings of the International Conference on Intelligent User Interfaces, Santa Fe, 33-40.

Cover, T. Hart, P. (1967). Nearest Neighbor pattern classification, *IEEE Transactions on Information Theory*, 13, pp. 21-27.

Duda, R. & Hart, P. (1973). Pattern Classification and Scene Analysis. New York, NY: Wiley and Sons.

Kobsa, A. (2002). Personalized hypermedia and privacy. *Communications of the ACM*, 45, 5.

Pazzani, M. (1999). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, 13(5-6): 393-408

Pazzani M., & Billsus, D. (1997). Learning and Revising User Profiles: The identification of interesting web sites. *Machine Learning*, 27, 313-331.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, Vol 1, No. 1/2, pp. 67—88.