Personalization on the Web

Alexander Pretschner

Institut für Informatik
Technische Universität München
Arcisstraße 21, 80290 München
Germany

www4.in.tum.de/~pretschn
pretschn@in.tum.de

Susan Gauch

Department of EECS
The University of Kansas
233 Snow Hall, Lawrence, KS 66045
USA

www.ittc.ukans.edu/~sgauch
sgauch@ittc.ukans.edu

December 2nd, 1999

Technical Report ITTC-FY2000-TR-13591-01
Information and Telecommunication Technology Center
Department of Electrical Engineering and Computer Science
The University of Kansas

Abstract

As of October, 1999, about 200 million people regularly access the Internet. However, this access is still more or less standardized in that almost everyone uses the same means of information retrieval. It is unlikely that 200 million people are so similar in their interests that one standardized way of retrieving information fits all needs. This paper takes a look at about 50 available personalization systems, proposes a classification scheme and discusses the systems w.r.t. to this classification.

Acknowledgements

The research presented in this Technical Report was partially supported by the National Science Foundation CAREER Award 97-03307.

The first author was in part supported by the German-American Fulbright Program.

Contents

1 Introduction

Soon after the WWW emerged, work on personalizing the access to or views of the Web began. This chapter gives an overview over existing *systems* and *approaches* to personalization developed over the last several years. Due to the large number of systems, this chapter is necessarily incomplete, but well-known representative systems are described.

In order to structure the wealth of approaches to personalization, the discussion will be organized according to the following orthogonal dimensions:

- **application**:

  For what are the user profiles used? Application fields can broadly be divided in *personalized access* to certain resources (personalized "portals" to the Web, file systems) and *filtering/ranking* issues: electronic newspapers, Usenet news, e-mails, recommendation services (browsing, navigation), tutoring systems, and search.

- **creation and representation of the user profiles**:

  What data is used to build the profiles? How are they built, i.e., what is the learning mechanism (if any)? How are they stored - structured or unstructured?

  - **data source for user profiles**:

    What is learned, i.e. how is the user profile obtained? More precisely, does the system learn *implicitly* by observing the user's behavior, or does it learn *explicitly* by requiring the user to enter her interests [45]?

  - **learning algorithms**

    Once information on a user is gathered, *how* is it used to build the profile? Is the system adaptive in that the profile changes over time, hopefully adjusting to a user's actual interests? Examples for learning algorithms are probabilistic algorithms, genetic algorithms [29], and algorithms working in the vector space model [50]. [31] contains a detailed bibliography for all of these approaches in the context of text learning.

  - **representation of user profiles**:

    How are the interests of a user stored? Common representations include Boolean or weighted keyword vectors, semantic nets, n-grams, and keyword vectors for a small number of categories.

- **rating and filtering algorithms**:

  Which algorithms are used to decide whether or not a user is interested in a particular item? In other words, how is the matching of a document with a user profile done?

- **collaborative vs. individual filtering**:

  Does the personalization and/or filtering process focus on *one* user, or is it also concerned with a community of users (*collaborative* filtering)?

- **architecture**:

  For collaborative and search issues, does the user profile reside on the *server* side, or is it local to the *user's* machine? A possible partitioning of this dimension is "agent" and "non-agent" systems, but there is no agreement on how to use this word, so this distinction is not considered in this survey.

The following sections present the systems grouped together by their application. Each of the above dimensions will be discussed. Discriminating features concerning the other dimensions will be presented together with their discussion in these brief presentations.

If the description of a system does not include a discussion on one of the dimensions, the description is necessarily omitted in this survey.

Section 3 contains a summary in tabular form.

Whenever possible, references to comparisons of products in one of the categories are given. [39] is a compilation of freely available information filtering systems (some of which will not be discussed here), and [20] is an early approach to categorizing Usenet news filtering systems. [30] and [31] contain a more recent discussion on some of the systems, and [9] gives an overview on other "intelligent" information brokering systems. Finally, [40]

contains a thorough discussion of trends in text filtering, in particular with respect to personalized approaches.

The word "agent" is rarely found in this overview, even though the ubiquitous use of this word would allow to classify all the presented systems as such. This is a deliberate choice. [16] is an extensive online bibliography on agents.

This overview is a revised version of the one contained in [43].

2 Applications of Personalization

Applications are coarsely divided into two areas: personalized access to some resources, and approaches involving filtering.

2.1 Personalized Access

As the Web continues to gain popularity, it is not surprising there do exist commercial providers for personalized information systems. Examples include **Entrypoint**[1] and **InfoQuest**[1]. These programs provide the user with a desktop containing links to different sources of information (news, weather, stock market, television programs and the like), and they allow for (explicitly) specifying topics of interest to the user.[2] The profiles here consist of a simple list of words or subjects. A very similar approach is implemented in the personal MyYahoo section of the popular search engine **Yahoo**[1]. Recently, this kind of service has been dubbed "portal".

Popular Internet browsers such as Microsoft's Internet Explorer or the Netscape Navigator allow for organizing bookmarks in a personalized manner. An early approach to personalization in this direction is the **PAINT**

[1]URLs: *Entrypoint*, formerly Pointcast: www.pointcast.com, *InfoQuest:* www.inforian.com/quest, *Yahoo:* www.yahoo.com

[2]The techniques used for filtering are very simple versions of the ones used by the systems described in section 2.2

system [41] which views the Internet as a file system and helps in personalizing views on it. PAINT considers the navigation problem a name space management problem and therefore aims at personalizing this name space. **BASAR** [56] assists users in managing their personal information spaces by updating links (in the bookmarks) and deleting links that are seldom or never used.

Finally, personalization is very common in the area of e-commerce, where a user explicitly wants the site to store information on her, such as credit card numbers and/or addresses (e.g., **Amazon.com**[3] which also regularly sends information on new books of interest (based on a list of categories a user enters) or **eBay**[3]) as well as user preferences such as "no frames" or "text only" (in the **Personal Wall Street Journal**[3]). This kind of information is typically stored in form of cookies [27]. Amazon and e-Bay are collaborative systems in the following sense: They allow for assessing books or vendors, respectively, and this collected information is visible to every user. **Firefly**[3]is a provider for personalized information systems, featuring customizable versions of MyYahoo, personalized movie recommendations, finding people with similar interests, or applications in e-commerce, e.g. **Barnes and Noble**[3]. User interests are determined by keywords, and later on, by reviews they write. This allows for personalized delivery of book recommendations and other information services.

2.2 Filtering and Rating

Filtering and rating seem to be the main focus of research in personalization. This section presents personalized newspapers, Usenet news filtering systems, recommendation systems for browsing and navigation, and search.

2.2.1 Newspapers

The electronic version of the **Personal Wall Street Journal**[3] allows for personalization in a similar way as do Yahoo or Pointcast.

[3] *Amazon:* www.amazon.com, *eBay:* www.ebay.com, *Wall Street Journal:* www.wsj.com, *Firefly:* www.firefly.net, *Barnes and Noble:* www.barnesandnoble.com

Information sources: The user interests have to be provided explicitly (by clicking on categories of interest), or they are inferred from a user's stock portfolio - the Wallstreet Journals proposes links or articles to follow which are related to the shares contained in the user's portfolio(s),[4] a somewhat more "intelligent" approach to personalization. Indeed, this seems to be a clever approach since the shares of a user will naturally reflect her interests!

Learning algorithm, profile representation: The underlying technology is not documented, but it is reasonable to assume that the companys' names are labels of classes, and these classes are presumably defined (trained) by words occurring in articles concerning a particular company. Concerning the change of user profiles over time, there seems to be no learning process (except when portfolios are updated).

Architecture: No explicit information is disclosed, but a list of cookies used by this site does not exhibit any cookie containing portfolio information. It seems reasonable to assume that this information is stored on the server's side.

Another electronic newspaper, **Fishwrap** [11] (the technology is used within the electronic version of the San Francisco Chronicle[6]), is quite similar in that it allows for choosing topics of interests and customizing the layout of the personalized news page.

A somewhat different approach was chosen for **Krakatoa** [18] and its successor, **Anatagonomy** [48], in that these products infer the user profiles from the user's behavior. The presentation of the articles "can be personalized in terms of contents, layout, media ..., advertisement, and so on"[18].

Information sources: An initial profile may be provided in form of a list of keywords. The user behavior is tracked while he reads: activities like scrolling, peeking at, maximizing, opening articles in new windows, or saving them to a scrapbook probably mean a user is interested in that article. Explicit feedback is also supported, and it turns out that, not surprisingly, explicit feedback yields better results than implicit feedback, and a combination of both clearly yields the best results w.r.t. recall and precision[7].

[4] Yahoo's *investment challenge* offers a similar service.

[6] http://www.sfgate.com

[7] Performance evaluation in terms of recall and precision is undertaken in ProFilter [8], too.

Profile representation: No explicit information is given, but the article suggests the profile is a list of weighted keywords (this is indicated by the fact that an initial profile can be given in form of a list, and that the user profiles and documents can be compared easily).

Collaborative vs. individual: Anatagonomy supports both collaborative as well as individual filtering.

Architecture: The user profiles are stored at the server's side (without this, collaborative rating would be much more difficult, but there are, however, privacy concerns).

[49] pushes the personalization a little further in that it assumes a user visits the newspaper several times a day and would probably like newer ("fresher") information to be visibly distinguished from known articles. This is done by putting "fresh" articles at exposed positions such as the top of the article list.

The **SmartPush** System [22] is used for information delivery of economic data as provided by a major Finnish newspaper.

Profile representation: An initial profile can be provided by ranking sample documents, giving a list of keywords or choosing among a set of default profiles. The profile is stored in the form of a concept hierarchy, or rather ontology.

Rating and filtering: Documents are augmented with ontologies similar to the user profiles, and they are created by hand by the document's author. These metadata describe the *content* of a document and are attached to the latter. According to the authors, the ontologies will eventually reach a size of 600 nodes (40 as of March 1999). The matching process then becomes slightly more complicated, since in this case distances between weighted hierarchies have to be calculated rather than distances between vectors ([51] proposes an asymmetric distance measure).

Learning algorithm: [22] emphasizes the adaptive nature of SmartPush, but no information is given on how implicit and explicit feedback actually is provided.

Collaborative vs. individual: At present, SmartPush is an individual system, but future versions are envisioned to support collaboration.

2.2.2 Usenet News

Taking into account the number of Usenet news filtering systems, this kind of information system clearly exhibits a need for personalization. This conforms to daily experience: With a hundred or even more articles per group, it is impossible to read them all.

The following briefly presents the systems NewT, SIFT, PHOAKS, PSUN, and GroupLens (as these seem to be good representatives for the different classes of news filtering software). There are many other news filtering systems (e.g., **NewsWeeder, Browse, NewsClip, Lurker, Smart, Borges, InfoScan, RAMA, Pefna**, and **InfoScope**) which can be found in [20] or [39].

NewT's [52, 53] personal profiles are initially provided by the user in form of a list of keywords. Whenever presenting a filtered article to a user, the latter decides by explicit relevance feedback if he liked or disliked this article. This feedback is then used to modify the profile.

Source of information, profile representation: Profiles are stored as vectors of weighted keywords (and so are the documents). Explicit user feedback is the input to the learning algorithm.

Learning algorithm: A user's interests are learned by means of a genetic algorithm. Several instances of the user profile (called agents) compete with each other, and an agent is rewarded when the user liked a suggested document (and punished when the user disliked it). The common techniques of crossover and mutation then yield a generation of agents that eventually represent a user's interests suitably.

Rating and filtering: Documents are compared with the user profiles using the cosine measure in the vector space model.

Collaborative vs. individual: Clearly, NewT is an system focusing on one individual.

In **SIFT** [59], filtering is done by comparing articles to an individual user's static profile. SIFT is the representative of the earliest class of news filtering programs which use the same profile representation and exhibit no adaptivity or learning component.

Profile representation, Rating and filtering: The profile is represented by

a Boolean or weighted vector of keywords. Matching of the profile and an article occurs w.r.t the cosine similarity of the vector space model.

Collaborative vs. individual: SIFT's focus is on individual users.

PHOAKS[8] (People Helping One Another Know Stuff) is not concerned with user profiles but rather with collective assessments of newsgroups. It allows for getting a summary of the group or statistics on top posters or freshness of messages. By making people vote for Web resources (not only newsgroups) and using the number of votes for quality assessments, PHOAKS clearly is a collaborative system.

PSUN [54] differs from the previous systems in the representation of the profiles and the learning technique.

Profile representation: Profiles are provided initially by presenting the system with some articles a user finds interesting. Recurring words in these are stored by means of *n-grams* (n words found to occur after each other a significantly high number of times and thus providing some context), and the n-grams are stored in a network of mutually attracting or repelling words, the degree of attraction being determined by the degree of co-occurrences/ Different user profiles are then stored in a way similar to Minsky's K-lines [28], connecting n-grams of different weights. Each user has multiple profiles that compete via a genetic algorithm.

Source of information: Explicit feedback is needed for the learning algorithm.

Learning algorithm: The user profiles consisting of K-lines-like connections of weighted n-grams compete with each other. The usual operations in genetic algorithms then eventually lead to a generation of profiles that represent the user's interests accurately. (Since this is a particularly original approach, it is regrettable there is no evaluation).

Collaborative vs. individual: PSUN aims to support single users.

GroupLens[9] [21] is different from the previous approaches in that it allows for implicit rating and is an exclusively collaborative filtering system.

[8] www.phoaks.com

[9] www.cs.umn.edu/Research/GroupLens

Information sources: Quality assessments of articles are based on explicit feedback and the time a user spent on a page (an approach also investigated in [36, 38, 40] and, with some modifications, is also implemented in [43] where it is discussed in some depth.

Collaborative vs. individual: GroupLens is not suited for individual personalization (and can therefore be seen as a recommendation service as discussed below).

A more recent system is **Alipes** [58] which allows for explicitly modeling *dis*interest in a particular field, and can be used for both searching and news filtering tasks.

2.2.3 Recommendation Services

Recommendation services usually suggest that a user follow a link on a page he is currently visiting (or suggesting that he *not* follow it). This recommendation is based on the user's interests or on his immediate browsing history within one site (Web usage patterns, [32]).

In this section, the following systems are presented: Alexa, Amalthaea, ifWeb, FAB, Letizia, SiteIF, Siteseer, Syskill and Webert, WebMate, WebWatcher, Personal WebWatcher, WebSift, and WebACE.

Alexa[10] uses collective usage patterns as a source of quality assessments of sites and as a basis for determining related links[10].

Profile representation and Information sources: A user's surfing history is sent regularly to the Alexa server. No information about the profile representation is disclosed (Alexa is a commercial product). The company claims no relationship between surfing history and user identity is stored.

Rating and filtering: Links are considered to be related if many users of the Alexa community show a similar usage pattern w.r.t. these links. Clustering/Data mining and text analysis of web sites are other components of Alexa's technology.

Collaborative vs. individual: Alexa's concern is collaborative use.

[10] www.alexa.com and www.alexa.com/support/technology.html

Amalthaea [37] is exploring personalized data discovery and information filtering. The Web is searched for documents that might be of interest for a user, and the user profiles are also used for news filtering.

Profile representation and Information sources: Initially, the user provides Amalthaea with a list of keywords reflecting her interests. The profile is stored in form of weighted keyword vectors. Explicit feedback is given by the user to decide if she liked or disliked the documents presented by Amalthaea.

Learning algorithm: Learning is done by means of genetic algorithms which compete in representing a user's interest most accurately. Eventually, the fittest class of algorithms will suitably represent the user's interests. Amalthaea can be bootstrapped with new genetic algorithms (profiles, list of keywords) that are explicitly provided by the user.

Rating and filtering: The quality of a document in terms of the user's interest is assessed by calculation of cosine similarities in the vector space model.

ifWeb [2] supports two modes: navigation support and support in document search.

Profile representation and Information sources: User profiles are stored in the form of "weighted semantic networks". These semantic networks differ from those in the knowledge representation domain, since they represent terms and their context by linking nodes (words) with arcs which represent co-occurrences in some documents. The authors claim that ifWeb supports implicit feedback, but their description lacks any mention thereof. The author was unable to verify this. In addition to the unconventional method of representing profiles, ifWeb is, however, interesting for two other reasons: It takes into account not only interests, but also explicit *dis*interest, and therefore presumably reflects a user's idiosyncrasies more accurately. Secondly, it incorporates a mechanism for temporal decay, i.e., ages the interests as expressed by the user.

Rating and filtering: No details are disclosed, but evaluations of the personalized orderings of some search results by means of the *ndpm* comparison [60] exhibit a good performance of the system.

Collaborative vs. individual: ifWeb focuses on individual users.

FAB [3] is a collaborative recommendation service and succeeds the **LIRA** system [4].

Profile representation and Information sources: User profiles are stored in form of weighted keyword vectors and updated on the basis of explicit relevance feedback. Documents and user profiles are matched according to the cosine similarity in the vector space model.

Learning algorithm: As stated above, profiles are updated w.r.t. explicit user feedback. FAB implements a (temporal) aging function for a user's interests.

Collaborative vs. individual: As already stated, FAB's focus is on collaborative filtering. A central repository of recommended documents is (automatically) updated with documents that are recommended by a user who exhibits interest in the document (whose interest profile matches the document). User profiles are compared on the basis of the cosine similarity, too.

Architecture: User profiles seem to be stored at the server's side (which seems inevitable in a collaborative system).

Letizia [25, 24] assists a user when browsing by suggesting links that might be of interest and are related to the page the user currently visits.

Profile Representation: No explicit information is available, but since the documents to be matched with a profile are stored as a weighted keyword vector, it is reasonable to assume that the user profile is a weighted keyword vector as well.

Information sources: Letizia relies on implicit feedback: Links followed from the currently visited page are assumed to reveal interest in the document containing the link. Bookmarking a page also means this page is interesting. Furthermore, as (Western) users tend to read from the top left corner to the right bottom corner, links that are omitted during the reading process might express disinterest in the referenced document.

Rating and filtering: There is no ordinal scale for the importance of suggested links but rather a (cardinal) preference ordering. It is reasonable to assume the filtering mechanism involves cosine similarities in the vector space model since the documents to matched are stored as weighted keyword

vectors.

Collaborative vs. individual: Letizia is for individual use. **Let's Browse** [23] extends Letizia for collaborative use.

SiteIF [55] strongly resembles ifWeb, except that explicit user interaction is avoided. The cited paper does not contain information on actual deployment of the system, so its current objective seems to be gathering and maintenance of user profiles.

Profile representation: As in ifWeb, profiles and documents are stored as semantic networks (terms and correlated terms, their context). SiteIf also involves a decay function for aging user interests.

Information sources: The profile is built in terms of the links followed by the user.

Architecture: Since users must enter a login name and a password is required, the author assumes that profiles are centrally stored.

Collaborative vs. individual: Like ifWeb, SiteIF is concerned with individual users.

SiteSeer [46] is another collaborative web page recommendation system.

Information source, Profile representation: User profiles are extracted from their bookmark files, taking into account the *content* of the referenced documents, and the *structure* of the bookmark file. The folders in the bookmark files are used to identify the user's categories of interest. Even though there is no technical information on how the representation is done, it is said that the system does not derive any semantic value from the content of the stored URLs. The profiles thus seem to consist of a list of URLs together with their structure.

Collaborative vs. individual: SiteSeer is a purely collaborative system. Recommendations occur when the profiles (as derived from the bookmarks) of two users match in terms of the URLs contained therein (and thus measuring the overlap), by giving additional weight to URLs that do not occur frequently, an approach similar to the *tf*idf* approach for content determination of documents [50].

Architecture: Being a collaborative system, the user profiles are most likely stored on a central server in order to allow for matching users.

Syskill and Webert [42] allows for both personalized search and recommendation (navigation). Search results, returned by Lycos, are annotated with symbols reflecting the assumed interest (good, okay, don't know, poor). In the recommendation mode, the system suggests links to follow on "index pages" which contain many links related to a given topic. Recommendation is done as in the search mode by graphically annotating the links on the index page. Examples for index pages are the pages contained in Yahoo's Browsing hierarchy or many overview pages in the WWW Virtual Library.[11]

Information sources: Syskill and Webert relies on explicit user feedback on a three point scale.

Profile representation: A user's interests are divided into classes (which simply coexist; there is no hierarchical relationship between classes). These interest classes describe the content of the index page, the links contained in which will be annotated later. Within each class, the profiles consist of boolean keyword vectors.

Learning algorithm: A thorough investigation[12] of which learning algorithm to choose resulted in choosing a naïve Bayes Classifier. Classification is done w.r.t. to the different categories of a user's interests. It is found that for good classification results, it is not necessary to characterize whole documents, but that the first 96 words of a document are sufficient[13]. Interestingly, this yields better results than working with entire documents.

Rating and filtering: Rating is done by classifying the documents w.r.t. to the user profile and determining the degree of membership.

Collaborative vs. individual: Syskill and Webert is an individual system.

WebMate [10] spiders a URL the user wants to be monitored, typically pages that contain many news headlines such as the homepage of NewsLinx[14]. The articles associated with headlines are fetched and compared to the user's

[11] http://vlib.org, a good example for an index page is the complete index for medicine related issues: http://www.ohsu.edu/clinicweb/wwwvl/all.html

[12] The discussion in [42] compares Bayesian classifiers, Nearest Neighbors, PEBLS, Decision Trees, *tf*idf*, and Neural Nets. [6] focuses on probabilistic user models. Text Learning techniques in this context are discussed in [31, 62].

[13] A similar observation is made in ProFilter [8].

[14] www.newslinx.com

profile, resulting in a personalized presentation of news. It is thus a recommendation service.

Profile representation: The system stores documents as weighted keyword vectors and clusters them. These clusters are then automatically labeled with the "most important" word and are assumed to represent one domain of a user's interests. The profile consists thus of the cluster centers together with their associated documents the number of which is bounded to save space.

Learning algorithm, information sources: There is some evidence the system relies on explicit feedback. The learning is done by adjusting the cluster centers as new documents are stored.

Individual vs. collaborative: WebMate is an individual system.

WebMate is an integrated tool which also provides assistance in searching by expanding queries (unpersonalized).

WebWatcher [1, 17] is a popular browsing assistant. For a particular site, WebWatcher takes the role of a museum guide, pointing the visitor to interesting documents.

Profile representation: The interests of a user are given at the beginning of the tour in form of a list of keywords (and therefore represent rather a "goal" than an "interest").

Information sources, Learning algorithm: Individually, the interests of a user are given at the beginning of the tour. No further learning takes place w.r.t. to a user's profile. In terms of collaboration, all hyperlinks are annotated with the profile (the goal in form of keywords) of the user who followed them. WebWatcher uses reinforcement learning to associate links with the content of the underlying documents. This aims at finding "paths through the Web which maximize the amount of relevant information encountered" [17].

Collaborative vs. individual: WebWatcher combines collaborative and individual aspects. Whenever a user selects a link, his interests (in form of some keywords) are annotated with that link. This information is subsequently used in the recommendation process by matching the annotation with the interests as expressed by the user. Personal WebWatcher is an adaptive version of WebWatcher.

Architecture: User profiles are rather a goal for one browsing session, and

this goal is stored at the server's side. The same is true for the collaborative implicit feedback (which links were chosen).

Personal WebWatcher [30] augments WebWatcher with adaptive behavior towards one user. It is thus a recommendation service, too. The suggestions are restricted to links that already exist on a page, and if the system considers them interesting, these links are highlighted.

Information source: To build and update profiles, Personal WebWatcher uses the content of links that have been followed as examples for interesting pages, and links that have not are considered boring.

Learning algorithm, Profile representation: Learning is done by a naïve Bayes classifier where the documents are represented as weighted keyword vectors, and the classes are "interesting" and "not interesting". The profile is then described by these two classes with the associated sets of documents (their vector representation).

Rating and Filtering: Bayesian Classification is used to distinguish between interesting and uninteresting pages.

Collaborative vs. individual: Personal WebWatcher is an individual system related to the collaborative WebWatcher.

WebACE [33] is a browsing assistant that is based on usage patterns. A user's browsing history within one particular site is monitored and used to determine the "best" links to follow which is done by comparison with other users who previously accessed that site.

WebSIFT[15] [13] is concerned with Web usage mining. The idea is to use (potentially global) access patterns of Web usage to recommend links at a particular site by comparing a (probably short) user's browsing history (within that site) with other users' browsing histories. The result of this comparison is then used to point users to interesting links, where interesting links are determined as an extrapolation of an individual user's surfing history. Individual user's profiles are deleted when this user leaves a website.

[15] www-users.cs.umn.edu/ cooley/websift/

References to other collaborative browsing assistants such as **Firefly**,[16] **Webhound** and **Ariadne** can be found in [23]. Since they are similar to other approaches, their description is omitted here.

2.2.4 Search assistance

ProFusion Personal Assistant [8] is a filtering tool for results returned by the meta search engine ProFusion [14]. It decides which results to present to the user and which to discard. This judgement is done for the results of queries that are *re*submitted regularly.

Information source: Explicit relevance feedback is used to determine the areas of interest.

Profile representation, Learning algorithm: User profiles are stored as sets of two classes of documents: interesting and rather boring ones. Documents are stored as weighted keyword vectors, and for both classes, every term is assigned a weight representing its membership to "its" class. Explicit feedback updates the two classes by simply adding the document to its class and possibly modifying the weights of the occuring terms in both classes.

Rating and filtering: For each term in the retrieved documents (or rather their summaries), its weight in the irrelevant set and in the relevant set are used to assess how interesting the document is. This is done by calculating similarities with the two classes in the vector space model.

Architecture: The user profile is stored on the server's side.

PEA [35] is similar to Syskill and Webert in that search results are augmented with icons indicating a possible interest of the user. PEA is intended to work on top or together with other personalization services.

Profile representation, Information sources: Profiles are essentially bookmark files, similar to Siteseer. Different folders represent different classes of interest. Documents contained in these classes are stored as weighted keyword vectors. PEA also allows for adding interesting search results to an index which initially contains the bookmarks.

The system described in [**27**] re-ranks search results rather than filtering

[16] http://www.agentsinc.com

them.

Profile representation, Information Sources: Profiles are stored as weighted keyword vectors. These vectors contain the frequencies of all the words occuring in a user's entire file system, and therefore, no explicit feedback is required. No adaptation takes place.

Rating and filtering: For all documents (URLs) that were returned by a search engine, every word contained in them is looked up in the profile. If it exists in the user profile, its weight in the retrieved document is added to the URLs score. This yields a new personalized ranking.

Architecture: Profiles are stored on the client machine.

The system described in [**44**] aims at improving search results by re-ranking and filtering them.

Profile representation, Information Sources: Profiles are created as a function of the web surfing history of an individual user. Surfed pages are characterized (i.e., their content or descriptive categories are determined) w.r.t. to a concept hierarchy comprised of 4,300 nodes (using the vector space model). This hierarchy also serves as the profile template: the result of the characterization process (i.e., the top nodes) yields the nodes of the profile hierarchy that are updated in terms of the time spent on a given page and the amount to which they describe a page.

Learning algorithm: See above. The weights in the profile nodes are updated constantly, thus allowing for the detection of shifting user interests.

Rating and Filtering: Pages that are returned by some search engine are categorized w.r.t. the aforementioned hierarchy (multiple cosine similarity in the vector space model). The system is unique in that different interests are kept different; there is no average as in other approaches that use the vector space model. The match with the user's profile is then used to re-rank search results.

Collaborative vs. Individual: The current system is individual. However, as the profiles are not built for some specific reason, other areas of application are obvious.

Architecture: User profiles are stored on the user's machine.

As **ifWeb** [2] and **Syskill and Webert** [42] are both recommendation ser-

vices and personalized search engines, their characteristics were discussed on pages 10 and 13, respectively.

Rating and Filtering: Concerning ifWeb, it is not clear if the personalization process is done by filtering or re-ranking of the returned results. Syskill and Webert annotates search results graphically, in a way similar to PEA.

2.3 Other

This section presents systems that do not fit in one of the other categories: expertise location, e-mail filtering, tutoring systems, and machine-dependent link annotation.

The system described in [**57**] exhibits a quite different form of personalization. Its aim is to find experts in a given field, e.g., the JAVA programming language.

Profile Representation and Information Source: Profiles are built by scanning a user's JAVA source code and storing the classes and/or constructs he uses in form of weighted keyword vectors.

Rating: Users in need of an expert submit their query which is then matched with all user profiles. The person with the presumably best knowledge to answer this question is determined by calculating a cosine similarity between the query and all user profiles.

Other similar projects use papers written, e-mails and citations to determine the field of expertise of a particular user [19].

Information Lens [26] is a tool for filtering and ranking e-mails.

Profile representation: Profiles are stored as rules on structured lists of keywords, where the structure is determined by the components of mails: sender, subject, etc.

Information source, Learning algorithm: Rules have to be built by hand.

There is a wealth of e-mail filtering systems. Examples include [7, 12, 47, 34].

CIRCSIM-Tutor[17] [61] is an automatic tutoring system. The number and

[17] www.csam.iit.edu/~circsim

sequence of correct and incorrect answers to a set of questions is used to dtermine the "best" next question, according to the performance of the student.

WBI [5] provides the user with rudimentary browsing assistance by recording his entire surfing history (as done in the most recent versions of Netscape Navigator and Microsoft Internet Explorer) and thus allowing for shortcuts. An interesting feature is the annotation of links with "their" network speed or download time.

Finally, document filtering systems of various kinds can be perceived as personalizing systems. So-called **cybersitters**[18] allow parents to enumerate categories or rather keywords that should not be contained in documents their kids retrieve. SurfWatch[19] allows companies to restrict the Internet access of their employees. As in the case of the cybersitters, the goal of this product is to "block objectionable sites".

3 Summary

This chapter summarizes the described systems in tabular form. The first column contains the **name of the system**, if available, and the second column briefly describes its **application, or purpose**. The third column gives a brief description of some **technical internals**, such as how the profile is built, and what data it is built on. Whether a system is **adaptive or static**, i.e. if a profile changes over time or not, is indicated in the fourth column. The fifth column indicates how the **matching process** of a profile with a document is done, and finally, the sixth column indicates if a system is a **collaborative or an individual** one.

[18] e.g. www.solidoak.com
[19] www1.surfwatch.com

System	Application	profile: what+how?	ad./ st.	rating model	coll./ ind.
Alexa	recomm.	collective browsing patterns	ad.	-	coll.
Alipes	news and search	cat. of interest, keyword vectors, expl. feedback	ad.	based on cosine sim.	ind.
Amalthaea	data disc. + news	keywords, expl. feedback, genetic alg.	ad.	cosine sim.	ind.
Amazon	e-comm.	cr.card#, book rev.+ ass., stored as keywords	st.		both
Anatagonomy	newspaper	browsing beh.	ad.		both
Ariadne	browsing				coll.
eBay	e-comm.	vendor ass., stored as keywords	st.		both
BASAR	bookmarks	URLs+their usage	ad.		ind.
Borges	news retrieval	keywords, based on SMART	st.	cosine sim.	ind.
Browse	news	articles read or not read, neural network	ad.	sim. of pairs of words	ind.
CIRCSIM-Tutor	tutoring	answer history		dyn.	ind.
FAB	recomm.	pages, weighted keywords, explicit feedback	st.	cosine sim.	coll.
FireFly	multiple		both		both
FishWrap	newspaper	keywords	st.		ind.
GroupLens	news	pages + time spent, expl. feedback	ad.		coll.
ifWeb	nav. + search	pages, expl. feedback?, sem. networks	ad.		ind.
InfoQuest	pers.acc.	interests stored as keywords	st.		ind.

System	Application	profile: what+how?	ad./ st.	rating model	coll./ ind.
Information Lens	e-mail filtering	mail components, hand built rules to connect them	st.		ind.
InfoScope	usenet news				
Krakatoa	newspaper	browsing beh.	ad.		ind.
Letizia	recomm.	pages, links followed, keyword vectors?	ad.	cosine sim.?	ind.
LIRA	recomm.	keywords	st.	cosine sim.	ind.
Lurker	news	rules (boolean conn. of keywords)	st.		ind.
MS Internet Explorer	bookm.+hist., portal	URLs, interests as keywords	st.		ind.
Netscape Navigator	bookm.+hist., portal	URLs, interests as keywords	st.		ind.
NewsWeeder	news	expl. feedback on articles, stored as keywords	ad.	cosine sim.	ind.
NewT	news	keywords, expl. feedback, gen. alg.	st.	cosine sim.	ind.
PAINT	bookmarks org.	URLs	st.		ind.
PEA	search	bookmarks + their structure, stored as keywords	ad.	cosine sim.?	ind.
Pefna	news	explicit feedback on articles in different categories	ad.	cosine sim.	ind.
Personal WebWatcher	browsing ass.	links followed, Bayes class.	ad.	Bayes class.	both
Pointcast	pers. access	interests as keywords	st.		ind.
PHOAKS	News	-	ad.	user votes	coll.

System	Application	profile: what+how?	ad./ st.	rating model	coll./ ind.
ProFusion Pers. Ass.	Search	search results, expl. feedback, stored as keyword vectors in 2 classes	ad.	based on cosine sim.	ind.
PSUN	news	few art. of interests, stored as K-Lines, expl. feedback	ad.	based on n-grams	ind.
SIFT	news	keywords	st.	cosine sim.	ind
SiteIF	recomm.	pages, profiles stored as sem. networks	ad.		ind.
SiteSeer	recomm.	bookmarks+their structure	st.	bookmark overlap	coll
SmartPush	econ. newspaper	interest cat. stored as ontology	st.		ind.
Syskill + Webert	search + recomm.	pages, explicit feedback, 1 prof. per user interest stored as weighted keywords, prob. learning and others	ad.	cosine sim.	ind.
Wall Str. J.	newspaper	portfolio, interest cat., stored as keywords	st.		ind.
WBI	browsing ass.	visited URLs	ad.		ind.
WebACE	browsing	usage patterns	ad.		coll.
Webhound	browsing				coll.
WebMate	searching, browsing, newspaper	interest categories learned automatically, explicit feedback?	ad.	cosine sim. with multiple categories	ind
WebSIFT	browsing ass.	surfing history	ad.	surfed pages	both

System	Application	profile: what+how?	ad./ st.	rating model	coll./ ind.
WebWatcher	browsing ass.	keywords repr. interests/goals, links annotated with prof., reinforcem. learning	st.		coll.
Yahoo	pers.access, portal	keywords	st.		ind.
[**27**]	search	local file system, stored as keywords	st.	based on freq.	ind.
[**44**]	search	surfing behavior, ontology of 4,300 nodes	ad.	structured cosine sim.	ind.
[**57**]	expertise location	JAVA source codes	st.		coll.

Table 1: Systems with personalization services

4 Discussion

Personalization is a very active and broad area of research with many applications. The main applications are

- customizing access to information sources such as articles in newspapers or products,
- filtering news or e-mails
- recommendation services for the browsing process,
- tutoring systems, and
- search.

This chapter introduced a classification methodology and briefly described roughly fifty personalized information systems. Different models of profile representation and learning algorithms were discussed and put in context with their respective application, mainly *rating*, *ranking*, or *filtering*.

Unfortunately, only a few systems evaluate and discuss their results scientifically - as [15] puts it, "... we laud with our hearts, not with our heads." This is in part due to the fact that it actually is hard to determine how well a personalization systems works, as this involves purely subjective assessments.

However, some approaches are discussed. These discussions then include comparisons of different learning algorithms, of personalized orderings vs. non-personalized ones, and discussions of well known measures from IR, recall and precision.

Due to a lack of data, a comparison of the systems with respect to performance is currently impossible.

References

[1] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. WebWatcher: A Learning Apprentice for the World Wide Web. In *Proc. AAAI Spring Symp. on Information Gathering from Heterogeneous, Distributed Environments*, March 1995.

[2] F. Asnicar and C. Tasso. ifWeb: a prototype of user model-based intelligent agent for documentation filtering and navigation in the World Wide Web. In *Proc. 6th Intl. Conf. on User Modeling*, Chia Laguna, Sardinia, June 1997.

[3] M. Balabanović. An adaptive web page recommendation service. In *Proc. 1st Intl. Conf. on Autonomous Agents*, Marina del Rey, CA, USA, February 1997.

[4] M. Balabanović and Y. Shoham. Learning information retrieval agents: Experiments with automated web browsing. In *Proc. 1995 AAAI Spring Symposium on Information Gathering from Hetereogeneous, Distributed Resources*, 1995.

[5] R. Barrett, P. Maglio, and D. Kellem. How to personalize the Web. In *Proc. ACM CHI'97*, Atlanta, USA, 1997.

[6] D. Billsus and M. Pazzani. Learning probabilistic user models. In *Proc. 6th Intl. Conf. on User Modeling, Workshop on Machine Learning for User Modeling*, Chia Laguna, Sardinia, June 1997.

[7] G. Boone. Concept features in re: Agent, an intelligent email agent. In *Proc. 2nd Intl. Conf. on Autonomous Agents*, St. Paul, MN, USA, 1998.

[8] E. Casasola. ProFusion PersonalAssistant: an agent for personalized information filtering on the WWW. Master's thesis, The University of Kansas, Lawrence, KS, 1998.

[9] E. Casasola and S. Gauch. Intelligent information agents for the World Wide Web. Technical Report ITTC-FY97-11100-1, Information and Telecommunication Technology Center, The University of Kansas, 1997.

[10] L. Chen and K. Sycara. A personal agent for browsing and searching. In *Proc. 2nd Intl. Conf. on Autonomous Agents*, pages 132–139, St. Paul, MN, USA, 1998.

[11] P. Chesnais, M. Mucklo, and J. Sheena. The fishwrap personalized news system. In *Proc. IEEE 2nd Intl. Workshop on Community Networking Integrating Multimedia Services to the Home*, Princeton, New Jersey, USA, June 1995. http://fishwrap.mit.edu.

[12] W. Cohen. Learning rules that classify e-mail. In *Proc. 1996 AAAI spring symposium on machine learning in information access*, 1996.

[13] R. Cooley, P.-N. Tan, and Jaideep Srivastava. WebSIFT: The Web site information filter system. In *Proc. Web Usage Analysis and User Profiling Workshop (WEBKDD'99)*, August 1999.

[14] S. Gauch, G. Wang, and M. Gomez. ProFusion: Intelligent fusion from multiple distributed search engines. *J. of Universal Computing*, 2(9), September 1996.

[15] R. Glass. Inspections - some surprising findings. *Communications of the ACM*, 42(4):17–19, April 1999.

[16] H. Helin. Bibliography on software agents. http://www.cs.helsinki.fi/~hhelin/agents/agent-bib.html, 1999. Department of Computer Science, University of Helsinki, Finland.

[17] T. Joachims, D. Freitag, and T. Mitchell. WebWatcher: A Tour Guide for the World Wide Web. In *Proc. IJCAI'97*, August 1997.

[18] T. Kamba, K. Bharat, and M. Albers. The Krakatoa Chronicle - an interactive, personalized newspaper on the Web. In *Proc. 4th Intl. WWW Conf.*, pages 159–170, 1995.

[19] H. Kautz, B. Selman, and A. Milewski. Agent amplified communication. In *Proc. 8th Annual Conference on Innovative Applications of AI*, Portland, Oregon, USA, August 1996. As cited in [57].

[20] F. Kilander. A brief comparison of news filtering software, 1996. http://www.dsv.su.se/~fk.

[21] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, March 1997.

[22] T. Kurki, S. Jokela, R. Sulonen, and M. Turpeinen. Agents in delivering personalized content based on semantic metadata. In *Proc. 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, pages 84–93, Stanford, USA, 1999.

[23] H. Lieberman, N. van Dyke, and A. Vivacqua. Let's browse: a collaborative Web browsing agent. In *Proc. Intl. Conf. on Intelligent User Interfaces*, January 1999.

[24] H. Liebermann. Letizia: An agent that assists Web browsing. In *Proc. Intl. Conf. on AI*, Montréal, Canada, August 1995.

[25] H. Liebermann. Autonomous interface agents. In *Proc. ACM Conf. on Computers and Human Interaction (CHI'97)*, Atlanta, USA, May 1997.

[26] T. Malone, K. Grant, F. Turbak, S. Brobst, and M. Cohen. Intelligent information sharing systems. *Communications of the ACM*, (30):390–402, May 1987.

[27] X. Meng and Z. Chen. Improve Web search accuracy using personalized profiles, January 1999. http://www.cs.panam.edu/~meng/unix-home/Research/DataMine/Writing/spects99.ps.

[28] M. Minsky. *The Society of Mind.* Simon and Schuster, New York, 1987.

[29] M. Mitchell. *An Introduction to Genetic Algorithms.* "A Bradford Book". The MIT Press, 1996. ISBN 0-262-63185-7.

[30] D. Mladenić. Personal WebWatcher: design and implementation. Technical Report IJS-DP-7472, J. Stefan Institute, Department for Intelligent Systems, Ljubljana, 1998.

[31] D. Mladenić. Text-learning and intelligent agents. Technical Report IJS-DP-7948, J. Stefan Institute, Department for Intelligent Systems, Ljubljana, 1998.

[32] B. Mobasher, R. Cooley, and J. Srivastave. Automatic personalization based on Web usage mining. Technical Report TR99-010, Department of Computer Science, DePaul University, 1999. http://maya.cs.depaul.edu/~mobasher/personalization/index.html.

[33] B. Mobasher, E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and J. Moore. WebACE: A web agent for document categorization and exploration. In *Proc. 2nd Intl. Conf. on Autonomous Agents*, St. Paul, MN, USA, 1998.

[34] K. Mock. Dynamic email organization via relevance categories. In *Proc. 11th Intl. Conf. on Tools with Artificial Intelligence*, pages 399–405, Chicago, IL, USA, November 1999.

[35] M. Montebello, W. Gray, and S. Hurley. A personable evolvable advisor for WWW knowledge-based systems. In *Proc. 1998 Intl. Database Engineering and Application Symposium (IDEAS'98)*, pages 224–233, Cardiff, Wales, UK, July 1998.

[36] M. Morita and Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proc. 17th Annual Intl. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, pages 272–281, 1994.

[37] A. Moukas. Amalthaea: Information discovery and filtering using a multiagent evolving ecosystem. In *Proc. 1st Intl. Conf. on the Practical Application of Intelligent Agents and Multi Agent Technology*, London, 1996.

[38] D. Nichols. Implicit rating and filtering. In *Proc. 5th DELOS Workshop on Filtering and Collaborative Filtering*, Budapest, Hungary, November 1997. ISBN 2-912335-04-3.

[39] D. Oard and J. Kim. Freely available information filtering systems, 1998. http://www.clis.umd.edu/dlrg/filter/software.html.

[40] D. Oard and G. Marchionini. A conceptual framework for text filtering. Technical Report EE-TR-96-25 CAR-TR-830 CLIS-TR-9602 CS-TR-3643, University of Maryland, May 1996.

[41] K. Oostendorp, W. Punch, and R. Wiggins. A tool for individualizing the Web. In *Proc. 2nd WWW Conference: Mosaic and the Web*, 1994.

[42] M. Pazzani, J. Muramatsu, and D. Billsus. Syskill&Webert: identifying interesting web sites. In *Proc. 13th Natl. Conf. on Artificial Intelligence*, 1996.

[43] A. Pretschner. Ontology Based Personalized Search. Master's thesis, The University of Kansas, Lawrence, KS, 1999. www4.in.tum.de/~pretschn/papers/kuthesis.ps.gz.

[44] A. Pretschner and S. Gauch. Ontology Based Personalized Search. In *Proc. 11th Intl. Conf. on Tools with Artificial Intelligence*, pages 391–398, Chicago, IL, USA, November 1999.

[45] E. Rich. User modeling via stereotypes. *Cognitive Science*, 3:329–354, 1979. As cited in [40].

[46] J. Rucker and M.J. Polanco. Siteseer: Personalized navigation for the web. *Communications of the ACM*, 40(3):73–75, 1997.

[47] M. Sahami, , S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In *Proc. AAAI workshop on learning for text categorization*, Madison, WI, July 1998.

[48] H. Sakagami and T. Kamba. Learning personal preferences on online newspaper articles from user behaviors. In *Proc. 6th Intl. World Wide Web Conf.*, pages 291–300, 1997.

[49] H. Sakagami, T. Kamba, A. Sugiura, and Y. Koseki. Effective personalization on push-type systems - visualizing information freshness. In *Proc. 7th Intl. WWW Conf.*, Brisbane, Australia, April 1998.

[50] G. Salton. *Automatic Text Processing.* Addison-Wesley, 1989. ISBN 0-201-12227-8.

[51] E. Savia, T. Kurki, and S. Jokela. Metadata based matching of documents and user profiles. In *Proc. 8th Finnish Artificial Intelligence Conference, Human and Artificial Information Processing*, pages 61–69, 1998. As cited by [22].

[52] B. Sheth. A learning approach to personalized information filtering. Master's thesis, Massachusetts Institute of Technology, February 1994.

[53] B. Sheth and P. Maes. Evolving agents for personalized information filtering. In *Proc. IEEE Conf. on AI for applications*, 1993.

[54] H. Sorensen and M. McElligott. PSUN: a profiling system for Usenet news. In *Proc. CIKM'95 workshop on Intelligent Information Agents Workshop*, Baltimore, USA, December 1995.

[55] A. Stefani and C. Strappavara. Personalizing access to web sites: The SiteIF project. In *Proc. 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT'98*, Pittsburgh, USA, June 1998.

[56] C. Thomas and G. Fischer. Using agents to personalize the web. In *Proc. ACM IUI'97*, pages 53–60, Orlando, Florida, USA, 1997.

[57] A. Vivacqua. Agents for expertise location. In *Proc. 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, pages 9–13, Stanford, USA, 1999.

[58] D. Widyantoro, J. Yin, M. El Nasr, L. Yang, A. Zacchi, and J. Yen. Alipes: A swift messenger in cyberspace. In *Proc. 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, pages 62–67, Stanford, USA, 1999.

[59] T. Yan and H. Garcia-Molina. SIFT - a tool for wide-area information dissemination. In *Proc. 1995 USENIX Technical Conf.*, pages 177–186, 1995.

[60] Y. Yao. Measuring retrieval effectiveness based on user preference of documents. *J. of the American Society for Information Science*, 46(2):133–145, 1995.

[61] Y. Zhou and M. Evens. A practical student model in an intelligent tutoring system. In *Proc. 11th Intl. Conf. on Tools with Artificial Intelligence*, pages 13–18, Chicago, IL, USA, November 1999.

[62] X. Zhu, S. Gauch, L. Gerhard, N. Kral, and A. Pretschner. Ontology based web site mapping for information exploration. In

Proc. 8th Intl. Conf. on Information and Knowledge Management (CIKM'99), pages 188–194, Kansas City, MO, USA, November 1999. http://www.ittc.ukans.edu/obiwan/publications/papers/ CIKM.html.