Automatic Cognitive Style Identification of Digital Library Users for Personalization

Enrique Frias-Martinez, Sherry Y. Chen and Xiaohui Liu
Department of Information Systems & Computing, Brunel University, Uxbridge, UB8 3PH, United Kingdom. E-mail: {enrique.frias-martinez, sherry.chen, xiaohui.liu}@brunel.ac.uk.

Digital Libraries have become one of the most important web services for information seeking. One of their main drawbacks is their global approach: in general, there is just one interface for all users. One of the key elements in improving user satisfaction in digital libraries is personalization. When considering personalizing factors, cognitive styles have been proved to be one of the relevant parameters that affect the way in which a user interacts with an interface. This justifies the introduction of cognitive style as one of the parameters of a web personalized service. Nevertheless, this approach has one major drawback: each user has to run a time-consuming test that determines his/her cognitive style. In this paper we present a study of how different classification systems can be used to automatically identify the cognitive style of a user using the set of interactions with a digital library. These classification systems can be used to automatically personalize, from a cognitive-style point of view, the interaction of the digital library and each one of its users.

Introduction

Digital Libraries (DL) are collections of information that have associated services delivered to user communities using a variety of technologies (Callan et al., 2003). The collections of information can be scientific, business or personal data and can be represented as a digital text, image, audio, video or other media. Due to the amount and great variety of information stored, DL has become, with search engines in general, one of the major web services (Liaw & Huang, 2003). Typically, DL have a global approach in which all users are presented with the same interface regardless the diversity of users in terms of preferences or skills. Nevertheless, different studies in information seeking have shown that matching the interface with users' preferences can help them to achieve their tasks in a satisfactory way (Marchionini, Plaisant & Komlodi, 1998; Bladfor, Stelmaszewska & Bryan-Kinns, 2001). From this perspective personalization is a key tool to increase user satisfaction.

Personalization is defined as the ways in which information and services can be tailored to match the unique and specific needs of an individual or a community (Callan et al., 2003). The key element of a personalized environment is the user model. A user model is a data structure that represents user behavior and captures human factors. The more information a user model has, the better the content and presentation will be tailored for each individual user. A user model is created through a user modeling process in which unobservable information about a user is inferred from observable information from that user; for example, using the interactions with the system (Zukerman, Albrecht, & Nicholson, 1999). User models can be created using a user-guided approach, in which the models are directly created using the information provided by each user, or an automatic approach, in which the process of creating a user model is hidden from the user. The personalization elements constructed using a user-guided approach are usually called adaptable (Fink, Kobsa, & Nill, 1997), while the ones produced using an automatic approach are usually called adaptive (Fink, Kobsa, & Nill, 1997; Brusilovsky & Schwarz, 1997).

When considering human factors for personalization, cognitive style, which influences an individual's preferences for organizing and processing information, has been largely ignored. Nevertheless, in recent years, different studies (Graff, 2003; Chen & Macredie, 2002) have found that users' cognitive styles significantly influence their reaction to the application interface in terms of user control, multiple tools and nonlinear interaction. Some of these studies have focused on information searching interfaces (Moos & Hale, 1999). This relevance implies that cognitive style can be a very relevant factor to include in a user model for personalization purposes, especially for information searching purposes (Moos & Hale, 1999). The main inconvenience of including cognitive style in a user model is that, in order to identify the cognitive style of a user, each user will have to run a questionnaire. This process is a time-consuming activity that not all users of a personalized system would be willing to undertake.

While there are different applications, mainly learning environments, that personalize presentation and content using a cognitive approach (Triantafillou et al., 2002, Papanikolau et al., 2003; Bajraktarevic, Hall & Fullick, 2003), all of them assume that the cognitive style of each user of the system is known in advance, which implies that all users of that system have run a cognitive style test. While this assumption can be valid for testing environments (like the applications commented) or systems with a reduced number of users, the approach is not valid for large-scale personalized systems. In this context, the idea of automatically create user models that identify users' cognitive styles is essential, because it saves the user time consuming activities and directly allows the

personalized system to present a cognitive personalized interface.

This paper presents, within the context of an information seeking environment known as the Brunel Library Catalogue, an automatic approach to identify users' cognitive styles to create adaptive user models. The user models created, which will contain the cognitive styles, can be used to personalize the digital library interface according to each user's cognitive style.

The rest of the paper is organized as follows: first we introduce the concept and architecture of personalized DL. Second we present the relationship between cognitive styles and user interface design. The third section of the paper presents the design of the experiments run for Brunel Library catalogue. The interactions captured in these experiments will be used to characterize and identify each user's cognitive style. The fourth section presents the design of different cognitive classification systems that automatically identify each user's cognitive style. We conclude the paper in section 5.

Personalized Digital Libraries

In general, DLs are made up of four components (Theng, Duncker, & Mohd-Nasir, 1999): (1) information; (2) structure, describing the syntactic and semantic characteristics of the information; (3) interaction elements, referring to the searching interface, screen design, etc.; and (4) properties, referring to security, copyright issues, etc., of the information available in the DL. The services provided by DL through their interaction elements (interface) can be classified into three groups:

• Mechanisms for content selection. These mechanisms make it possible for each user to create a personal DL that contains only the information that is interesting and relevant to that user
• Mechanisms to help in the process of navigation. These services present each user with an environment that better suit the way in which that user interacts with the DL.
• Information filtering (IF) and information retrieval (IR) mechanisms. These services provide ways to find and filter the vast amount of information that a user accesses and receives.

Although these three basic types of services provide the basic functionality needed by a DL user, they can be improved by the introduction of personalization. Personalization will create more tailored services that help and simplify the process of finding relevant information by using the content of each user model. Formally, a user model is as a set of information structures designed to represent one or more of the following elements (Kobsa, 2001): (1) representation of assumptions about the knowledge, goals, plans preferences, tasks and/or abilities about one or more types of users; (2) representation of relevant common characteristics of users pertaining to specific user subgroups (stereotypes); (3) the classification of a user in one or more of these subgroups; (4) the recording of user behavior; (5) the formation of assumptions about the user based on the interaction history and/or (6) the generalization of the interaction histories of many users into stereotypes. In the
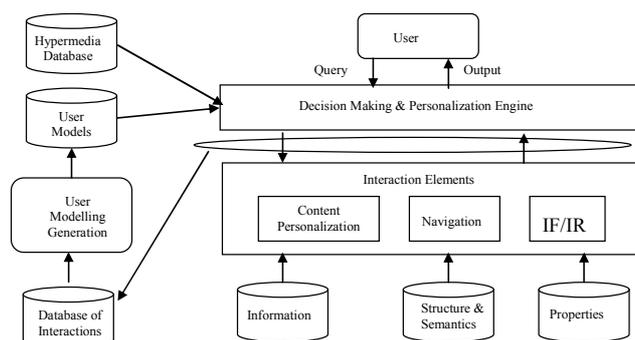

FIG. 1. Generic Architecture of an Adaptive DL.

context of user modeling a stereotype is defined as a cluster of users that share a common behavior.

Typically, personalization in DL has been user-driven. In this approach the user specifies his/her preferences directly to the DL, from the color background of the page, to the layout of the components or to the content of the information presented. The main inconveniences that this approach has are: (1) the concept of personalization cannot be necessarily understood by all the users of a DL, (2) users are not usually willing to give feedback to the system, even if it is for receiving a better service, and (3) users do not necessarily know what their interests are and how they change over time, and cannot provide information to the system. In constructing a user model that contains the cognitive style, there is an added inconvenience: users need to run a time-consuming test to identify the cognitive styles.

We think that a better approach to provide personalized DL services will be based on using user models that are automatically constructed using machine learning techniques, i.e. adaptive user models, because the application of these techniques will remove the limitations that a user-driven approach entails. Although machine learning techniques have been extensively used in e-commerce sites (mainly for recommendation purposes), their implementation in DL has been very limited up to now.

Figure 1 presents the architecture of an adaptive DL. As can be seen, the interaction between the User and the DL is handled by a Decision Making & Personalization Engine, which takes into account the user models and the interaction, personalizes the interface and the results to each user. The adaptive characteristic is given by the "User Modeling Generation" module, which has as input a database containing the interactions between the set of users and the library, and automatically produces the set of user models. This automatic approach allows to observe users in an unobtrusively way and solves the problem that the user-guided approach has.

In this paper we are going to focus on the two modules responsible for automatically creating user models: The "User Modeling Generation" module and the "Database of Interaction" module. First we will design a set of experiments aimed at capturing user interactions with Brunel Library Catalogue, the output of which will be stored in the "Database Interaction Module". Then, using this information we are going to design a "User Modeling Generation module" that

automatically constructs a set of user models that describe the cognitive style of each user. This set of user models makes it possible to implement a personalized DL from a cognitive style point of view while at the same time avoids the user-driven approach which will imply that each user of the library would need to run a test to identify his/her cognitive style. The following section highlights why cognitive style is a very relevant parameter for personalization.

Relevance of Cognitive Styles for Personalization

Although there are a lot of different definitions of cognitive style in the literature, it can be defined as an individual's preferred and habitual approach to organizing and representing information (Riding and Rainer, 1998). Cognitive style is a personality dimension, which influences the way individuals collect, analyze, evaluate, and interpret information (Harrison and Rainer, 1992). There are a variety of dimensions of cognitive styles, but among these dimensions, Field Dependence versus Field Independence has significant impacts on users' information processing, because it reflects how well an individual is able to restructure information based on the use of salient cues and field arrangement (Weller et al., 1994). Their different characteristics are:

Field Dependence(FD): Field Dependence describes the degree to which a user's perception or comprehension of information is affected by the surrounding perceptual or contextual field, that is, "the extent to which the organization of the prevailing field dominates perception of any of its parts" (Witkin, et al., 1971). Field Dependent individuals typically see the global picture, ignore the details, and approach a task more holistically. Field Dependent individuals are considered to have a more social orientation than Field Independent persons since they are more likely to make use of externally developed social frameworks. They tend to seek out external referents for processing and structuring their information. They are more readily influenced by the opinions of others, and are affected by the approval or disapproval of authority figures (Witkin et al., 1977).

Field Independence(FI): Field Independent individuals tend to discern figures as being discrete from their background, to focus on details, and to be more serialistic in their approach to learning. These individuals tend to exhibit more individualistic behaviors since they are not in need of external referents to aide in the processing of information. They are better at processing impersonal abstract material, are not easily influenced by others, and are not overly affected by the approval or disapproval of superiors (Witkin et al., 1977).

Recent studies have found that users' cognitive styles significantly influence their reaction to the user interface in terms of user control, multiple tools, and non-linear interaction. With respect to user control, several studies have suggested (Chuang, 1999, Chanlin, 1998) that FI individuals could particularly get benefit from the control of media choice.

Other studies (Marrison and Frick, 1994) have suggested that FD users prefer to have auditory cues in the systems. Regarding multiple tools, Ford and Chen (2000) showed that FD individuals tend to build a global picture with the hierarchical map when interacting with web services, while Palmquist and Kim (2000) found that FD novices tend to follow links prescribed by a web page. Regarding non-linear interaction, Dufresne and Turcotte (1997) investigated the effect of cognitive style within a searching information environment. They found that FD students who used the system with non-linear structure spent more time completing the test than those who used the system with linear structure. FI individuals consulted the user guide for a longer period than FD individuals in the linear version, while FD individuals consulted it for longer in the non-linear version.

Results from these studies suggest that different cognitive style groups prefer different interface functionalities and structures provided by web-based applications. These results indicate the relevance of cognitive styles for personalization. In this paper we present a system that automatically identifies the cognitive style of a library catalogue user for personalization purposes. This approach is even more relevant because different studies have shown that FD/FI is consistent across domains and stable over time (Goodenough, 1976; Messick, 1976; Witkin, et al, 1977; Witkin and Goodenough, 1981). This basically implies, that once a user has been assigned a cognitive style type, that type is constant in that environment, Brunel Library Catalogue in our case, and not only that, but that it is constant in other domains, i.e. if we identify the cognitive style of a user for Brunel Library catalogue, the identification of his/her cognitive style may be able to be used for other applications.

Experiment Design

This section describes the different characteristics of the experiments that were designed to capture interaction data to automatically create cognitive user models. The following subsections present the characteristics of the participants, the research instruments used, including the DL in which this study focuses, and the tasks designed and data collection techniques used.
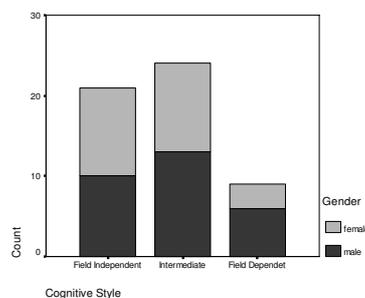

FIG. 2. Number of participants in each cognitive style.

Participants

The study was conducted at Brunel University's Department of Information Systems and Computing. A total of 54 students participated in this study. All participants had the basic computing and Internet skills necessary to use Brunel Library Catalogue. Figure 2 illustrates the number of participants in each cognitive style.

Research Instruments

The research instruments used include: (1) Cognitive Style Analysis (CSA) to measure participants' cognitive styles, (2) a digital library catalogue, Brunel Library catalogue, which is the focus of the study, and (3) a tool for capturing user interaction and storing a user questionnaire, WebQuilt.

Cognitive Style Analysis

A number of techniques have been developed to measure Field Dependence/Field Independence, and among those we have chosen the Cognitive Styles Analysis (CSA) by Riding (1991). The CSA test includes two sub-tests: (1) the first presents items containing pairs of complex geometrical figures that the individual is required to judge as either the same or different and (2) the second sub-test presents several items each comprising a simple geometrical shape, such as a square or a triangle, and a complex geometrical figure and the individual is asked to indicate whether or not the simple shape is contained in a complex one by pressing one of two marked response keys (Riding and Grimley, 1999). These two sub-tests have different purposes. The first sub-test is a task requiring *Field Dependent* capacity, while the second sub-test requires the disembedding capacity associated with *Field Independence.* This provides a big advantage with other methods that only measure one of the factors.

The CSA measures what the authors refer to as a *Wholist/Analytic* (WA) dimension, noting that this is equivalent to Field Dependence (Riding & Rayner, 1998). WA dimension is a real number between the values of 0.6 and 3.0 that indicates the degree of field dependence. Riding's (1991) recommendations are that scores below 1.03 denote Field Dependent individuals; scores of 1.36 and above denote Field Independent individuals; and scores between 1.03 and 1.35 are
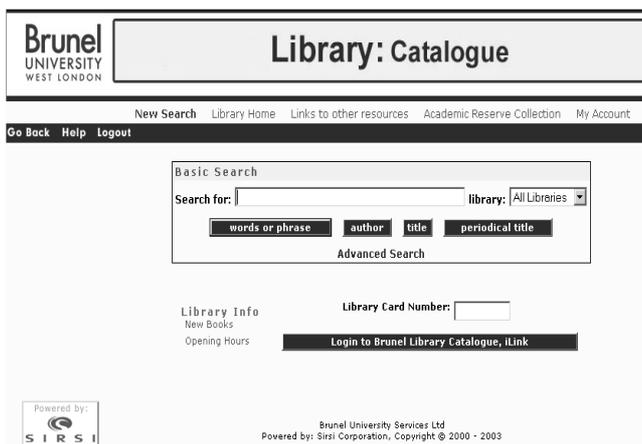

FIG. 3. Basic Search Interface of BLC.

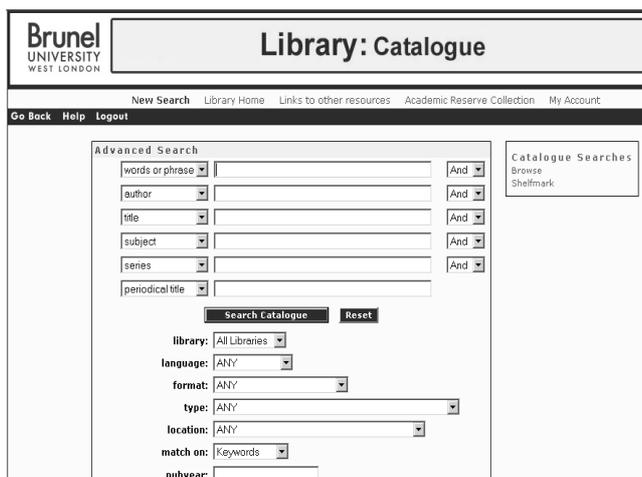FIG. 5. Multiple Results Interface of BLC.

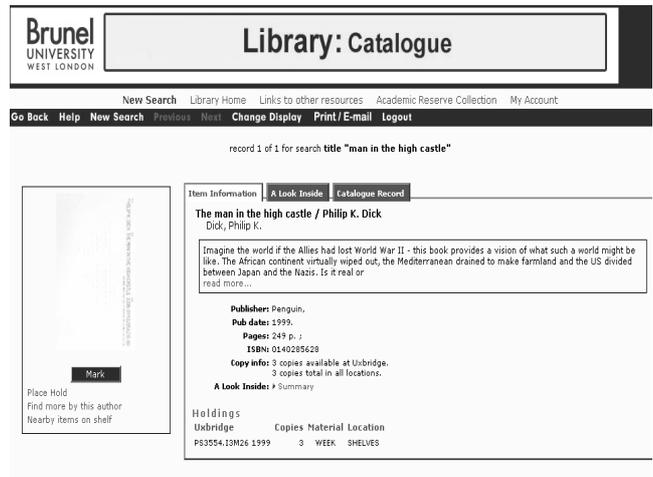FIG. 4. Advanced Search Interface of BLC.

FIG. 6. Results Interface of BLC.

TABLE 1. Set of tasks designed for the experiment and their type.

	Task	Type
1	Find the Call Number of the book "The Man in the High Castle" by Philip Kendred Dick.	Search
2	Find the title of any book related with applications of fuzzy logic.	Browse
3	Find the number of books written by Aldous Huxley that are part of TWICKENHAM Library	Search-Browse
4	Find a book about how to implement data mining with Java.	Browse
5	Find a Java book written by Hugh Vincent.	Search
6	Find a book about 20th century American Drama in TWICKENHAM campus.	Browse
7	Please find an IEEE journal on consumer electronics.	Search

classified as Intermediate. In this study, categorizations were based on these recommendations.

Brunel Library Catalogue

Brunel Library Catalogue (BLC) is a typical digital library to access the bibliographical resources of Brunel University. BLC has two main mechanisms that provide different strategies for finding information: (1) Basic Search (Figure 3), which is the one presented by default by the system, and (2) Advanced Search (Figure 4) which is accessed through the corresponding link presented in Figure 3. Basic Search allows to run a quick search of the library catalogue using a set of keywords and one of the following commands: "*word or phrase*", "*author*" "*title*" or "*periodical title*". Also the user can choose in which library he/she wants to search that information. The help link describes briefly what each link is supposed to do. Advanced Search, as presented in Figure 4, presents the user with a much broader way of searching information. The user can give value to each field (a generic work, author, title, subject etc.), and combine these words using and/or Boolean operators. The system also allows users to select other information like the library, the language, the publication year, etc.

Once a user submits a query to the system using the Basic Search or the Advance Search, the system responds with the items found in the database. An example of the interface presented is given in Figure 5. The system presents a set of buttons in the top part: "*Go Back*", "*Limit Search*", "*New Search*", "*Backward*", "*Forward*", "*Prefs*" and "*Exit*". The "*Limit Search*" option is a link to the bottom of the page where the search mechanism used (Basic Search or Advanced Search) is presented with the terms used and a set of options for Search Limits (language, publication year, etc.). The limit search is obtained adding more words to the set of terms already introduced. The "*New Search*" option presents again the interface of Figure 3. The "*Backward/Forward*" button allows to move up and down the items found. Once a user selects one item the information and interface given is presented in Figure 6.

WebQuilt & Exit Questionnaire

WebQuilt Proxy Server (Hong et al., 2000) (http://guir.berkeley.edu/projects/webquilt) is a proxy system that unobtrusively gathers click stream data as users complete specified tasks. It is designed to conduct remote usability testing on a variety of Internet-enabled devices and provide a

way to identify potential usability problems when the tester cannot be present to observe and record user actions. WebQuilt utilizes Java Servlet and JSP technology to track users' interaction and then store that data by (1) creating a log file of each user's web use and (2) additionally caching the pages a user accesses for later viewing. Figure 7 shows the basic architecture of WebQuilt Proxy. Once the proxy server is running, each user connects to any web page through the web server. The proxy server stores any interaction between the user and any web pages and a snapshot of each page visited by each user. These snapshots are given a number that is the same one used to describe the sequence of pages visited by the user. The Web proxy server has the possibility of adding a task box that can be used to indicate when a task has been finished. Once a user finishes each task and uses the task box links to finish it, Web Quilt allows to present to each user a set of questions regarding the task.

All the information captured is stored in the proxy server creating in each case a file using and id for each user. This allows to centralize all the information in the same place and at the same time being able to access the information of each user independently. The use of a proxy server architecture allows us to easily capture all the interaction between users and BLC, which otherwise would be far more difficult due to the changes needed to be implemented in BLC.

Task Design

The purpose of this experiment is to collect enough data to create an automatic classification system capable of identifying each user's cognitive style. The main behaviors that a user that access a web library catalogue has are two: browsing and searching (Bryan-Kinns, 2000). In this context browsing is defined as the search of ill-defined information while searching is defined as the localization of specific well-defined
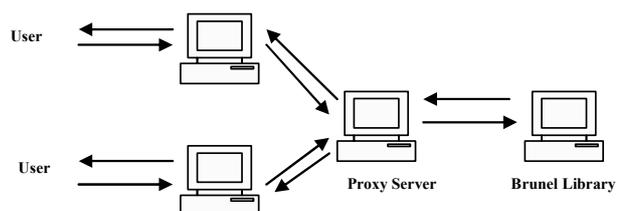

FIG. 7.. Generic Architecture of an Adaptive DL.

TABLE 2. Table of variables that compose a user behaviors' vector.

	Variable Name	Information
1	BS	Number of times that the user used the Basic Search functionality to solve the seven tasks
2	AS	Number of times that the user used the Advance Search functionality to solve the seven tasks.
3	SE	Number of times that the user used the "Word or Phrase" option from the Basic Search Interface to solve the seven tasks.
4	ATS	Number of times that the user used the "author", "title" and "periodical" options from the Basic Search interface to solve the seven tasks.
5	NS	Number of times that the user pressed the New Search functionality to solve the seven tasks.
6	GB	Number of times that the user pressed the Go Back button to solve the seven tasks.
7	CS	User cognitive style obtained using CSA test (Field Dependent, Intermediate of Field Independent)
8	WA	WA ratio of the user provided by the CSA test.

information. Theng et al. (1999) presents an example of this approach, with two experiments designed to capture user interaction with a digital library in order to evaluate the degree of satisfaction of the users with the library interface.

In order to identify users' real perceptions, participants were asked to perform a set of seven practical tasks (Table 1). The design of the task was interface dependent: the set of tasks was designed to involve all the functionalities that BLC provides to each user and the different behaviors (searching and browsing) that a user can show.

The first question captures a searching behavior, as it has a clear well-defined answer contained in the library catalogue. It is also designed to capture if the user uses the "*Search Everything*", "*Author*" or "*Title*" options (which are different ways of approaching the problem) or if a Power Search is used. In case the power search is used it will be interesting to capture which elements are used (title, author, year), and if any search limit is introduced. The second task is a browsing question designed to test if the user uses the "*Subject*" option or prefers an approach using "*Title*" or "*Search Everything*". The third question is a combination of searching-browsing task designed especially to see if users use the option of selecting a specific campus or do it manually. It is also interesting to see how the user approaches the problem (searching by author or by general search). The rest of the tasks are designed to replicate some of the functionalities and/or behaviors, in order to have more data to effectively construct a cognitive classification system.

Experimental Procedures

The experiment was conducted using Brunel Library catalogue. The experiment comprised three different steps:

(1) Participants were given a task sheet, which described the task activities that they needed to complete with BLC. One participant carried out the experiment at a time.
(2) The CSA was used to classify participants' cognitive styles into Field Independent, Intermediate, or Field Dependent.
(3) Participants were observed while they were carrying out the tasks, and clarifications were given when requested.

(4) At the end of each task the user answer the following questions: (1) was the user able to solve the task?, (2) was the task difficult?, and (3) what is the answer found?.

Data Collection & Summarization

The data collected from each user was centrally stored in the proxy server. For each user, the interaction data captured for solving the seven given tasks was summarized into six dependant variables, variables 1 to 6 in Table 2, that formed one vector containing the elements describe in Table 2. Each variable was then normalized "to one task" by dividing each value by seven. After normalization, each vector captured the way in which each user interacts with BLC to solve one generic task. Each vector had also two independent variables: users' cognitive styles and WA ratio (variables 7 and 8 in Table 2). The final database contains one vector for each user describing his/her interactions with BLC and his/her cognitive style and WA ratio. This provides an environment in which machine learning theories can be easily applied.

Cognitive Automatic Classification System

In this section we detail the construction of an efficient cognitive identification system. We start with a traditional machine learning approach using classification with neural networks and decision trees. After studying why the results are not satisfactory, we propose the construction of classification systems based on regression.

Decision Trees and Neural Networks

Two traditional approaches for constructing classification systems are decision trees and neural networks. Both theses approaches can also be used for regression (function approximation).

Decision tree learning (Mitchell, 1997; Winston, 1992) is a method for approximating discrete-valued functions with disjunctive expressions. There is a great variety of different decision tree algorithms in the literature: Classification &

TABLE 3. Classification and Regression results.

	3- Cross Validation	66% Split
Classification		
C4.5	45.8 %	52.9 %
MLP	60.4 %	41.1 %
Regression		
C&RT	70.2%	71.4%
MLP	68.2% (5 Neurons) 56.5 % (10 Neurons)	66% (5 Neurons) 58% (10 Neurons)

Regression Trees (C&RT) (Breiman et al, 1984), Chi-squared Automatic Interaction Detection (CHAID) (Kass, 1980), C4.5 (Quinlan, 1993), or ID3 (Quinlan, 1986). In the context of user modeling, decision trees can be used to classify users, according to the level of expertise, or the interests, or the cognitive style, etc., in order to use this information for personalization purposes (Tsukada et. al, 2002; Beck et al., 2003; Paliouras et al, 1999; Zhu et al., 2003; and Webb et al., 1997). Classification rules are an alternative representation of the knowledge obtained from classification trees. Algorithms such as CART and C4.5 include methods to generalize rules associated with a tree.

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems process information (Fausett, 1994; Haykin, 1999). NNs are able to derive meaning from complicated and/or imprecise data. Also, NN does not require the definition of any metric which makes them completely application independent. No initial knowledge about the problem that is going to be solved is needed. These characteristics make NNs a powerful method to model human behaviour and an ideal technique to create user models for personalization (Bidel et al., 2003; Sas et al., 2003; Beck et al., 2003; and Sheperd et al., 2002). Depending on the architecture of the network, different NN can be defined. One of the most typical architectures is Multi-Layer Perceptrons (MLP), which are fully connected feed-forward nets with one or more layers of nodes between the input and the output nodes (typically there are three layers, called the input, hidden and output layer) and where each layer is composed of one or more artificial neurons in parallel that use the backpropagation training algorithm.

Result Analysis using Classification: C4.5 and MLP

Two cognitive classification systems were constructed using: (1) C4.5 as an example of classification trees and (2) MLP as an example of neural networks. Both systems were constructed using Weka Data Mining Software (Witten & Frank, 1999), http://www.cs.waikato.ac.nz/~ml/weka/. MLP was designed with a three-layer network with a sigmoidal transfer functions and was run for 500 epochs with a backpropagation algorithm. The training vectors consisted of six dependent variables (variables 1-6 in Table 2) and one independent variable, the cognitive style (variable 7 in Table

2). The output in each case was a classification system that identifies the cognitive style of a BLC user (FD, Intermediate or FI) by considering the set of interactions of that user. In order to test the classification system, two testing techniques were applied: (1) splitting and (2) cross-validation. Splitting divided the file into 66% for training and 33% for testing and 3 cross-validation, was applied. Table 3 present the cognitive classification results.

Classification results are not satisfactory: basically only one out of two users is assigned their correct cognitive style. In our opinions, the possible reasons are: (1) although it may seem that the problem we are dealing is a traditional classification problem in which all the instances (users) have assigned a class (cognitive style), this is not entirely true, because each user, originally, is assigned a number (WA ratio) which is then translated into a class, (2) the definition of cognitive style is something fuzzy which is not completely understood how it translates into the context of a library catalogue, (3) the behaviour of users within a class (cognitive style) is not necessarily constant: two users with the same cognitive style can have very different behaviours if for example one of them has a WA ratio near a border with another cognitive style, with which will share some behaviour patterns, and the other has a WA value not near any border, showing a pure behaviour of that cognitive style. This characteristic makes it harder to construct an efficient cognitive classification system.

Taking into account the previous conclusions we think that a regression approach (function approximation), in which we construct a system that predicts the WA ratio of a user and from that his/her cognitive style, instead of obtaining his/her cognitive style directly, would produce better results because the system would avoid the ill-definition of the concept that is trying to classify.

Result Analysis using Regression: C&RT and MLP

Two regression systems were constructed using: (1) C&RT as an example of classification trees used for regression and (2) MLP as an example of neural networks architecture using for regression. MLP was designed with a two-layer network with backpropagation learning, with a tan-sigmoid transfer function in the hidden layer and a linear transfer function in the output layer, which is a typical structure for regression problems (Demuth & Beale, 1998), and was trained until the root mean square (RMS) error was smaller than 0.01. Both systems were designed with MATLAB (www.mathworks.com/products/matlab), using the NN Toolbox (www.mathworks.com/products/neuralnet) for MLP and the Computational Statistics Toolbox (Martinez & Martinez, 2001) for C&RT. The training vectors consisted of six dependent variables (variables 1-6 in Table 2) and one independent variable, the WA ratio (variable 8 in Table 2). The output in each case was a regression system that obtained and approximated value for the WA ratio of each user considering the set of interactions of that user. Again, in order to test the classification system two testing techniques were applied: (1) splitting and (2) 3-cross validation. Table 3

presents the cognitive classification results for C&RT and MLP, considering for MLP two different architectures with 5 and 10 neurons in the hidden layer. In order to obtain the correct classification rate when using the regression approach, first, the WA ratio predicted by the system was transformed into the cognitive style, and after that, the comparison with the correct cognitive style was made.

In general, we can see an increment in the correct classification rate, for both MLP and decision trees. This demonstrates that the regression approach outperforms the classification approach, which may be because of the ability of regression to manage the ill-definition of the concept being classified. This result is in accordance with what Peterson, Deary & Austin (2003) stated, that the use of category information on the CSA is considerably less reliable than the use of the WA ratios. When using MLP it seems that an increment in the number of hidden neurons reduces the correct classification rate. The best results are obtained using the decision tree approach with C&RT, which achieves a 70% correct classification rate, 5% higher than using MLP. We consider that one of the limitations of using a decision tree approach is that we lose the ability to capture the inherent uncertainty that modelling human behaviour has. In this context, a combination of decision trees with a soft computing technique (like NN for example) could increase the correct classification rate. Considering that decision trees can also be expressed in the form of classification rules, there are in the literature a variety of algorithms that combine classification rules with neural networks, generally called neuro-fuzzy systems. Neuro-fuzzy systems provide an excellent framework to automatically create rules that learn from examples using neural networks and that are able to handle the uncertainty and fuzzyness of the concepts and data being used.

Neuro-Fuzzy System

Neuro-Fuzzy systems combine a knowledge representation framework, fuzzy logic, with the learning capabilities of neural networks (Jang & Sun, 1995; Jang and Sun, 1999).
Fuzzy Logic defines a framework in which the inherent uncertainty of real information can be captured, modeled and used to reason with uncertainty (Klir, 1995; Yang et al. 1994). Fuzzy Inference Systems (FIS) are constructed using a set of membership functions for each input (also called linguistic labels) and Fuzzy Inference rules. Fuzzy Inference Rules take the form ``IF x is a, THEN y is b'', where x and y are inputs of the system and a and b are membership functions defined in x and y respectively. Under classical logic, the THEN implication is true if the antecedent is evaluated to be true. For fuzzy rules, the implication is set to be true to the same degree as the antecedent. A traditional FIS is divided into three steps: (1) fuzzification; (2) fuzzy inference; and (3) defuzzification.

The basic idea of combining fuzzy systems and neural networks is to design an architecture that uses a FIS to represent knowledge and the learning ability of a neural network to optimize its parameters. Neuro-Fuzzy systems (NFS) use NNs to learn and fine tune rules and/or membership functions from input-output data. NFS automate the process of transferring expert or domain knowledge into fuzzy rules. One of the most important NFS algorithms is ANFIS, Adaptive-Network-based Fuzzy Inference Systems, (Jang, 1993), which has been used in a wide range of applications (Bonisone, Badami & Chiang, 1995). The combination of NN and fuzzy sets offers a powerful method to model human behavior which allows NFS to be used for a variety of user modeling tasks (Lee, 2002; Stathacopoulou Grigoriadou & Magoulas, 2003; Drigas et al. 2004; Magoulas, Papanikolau & Grigoradou, 2001).

One of the main limitations of NFS is the training time needed, which is exponential with the dimension of the input space. This dimensionality problem also appears with the rules, the number and size of rules exponentially increments with the dimensionality of the input space (with a factor given by the number of membership functions). This complexity in rules will also affect the execution time of NFS, i.e. the time needed to identify the cognitive style of a user. In our case this is of critical importance because we want to implement a system that is able to identify the cognitive style of a user in real-time in order to present a cognitive personalized interface. These problems imply that, in order to efficiently implement a NFS approach, we first need to reduce the input space of the system.

Feature Selection by Information Gaining

The objective of this section is to identify the subset from the six original dependent variables that better characterize each cognitive style within the context of a neuro-fuzzy classifier in order to avoid the dimensionality problem. To identify which variables are more relevant we have selected all subsets of one, two, three and four variables from the original six, and for each combination a neuro-fuzzy system has been trained using data splitting, with 66% for training and 33% for testing (NFS of higher dimensionality were not able to be trained due to the dimensionality problem). All neuro-fuzzy systems were implemented using MATLAB's Fuzzy Logic Toolbox (www.mathworks.com/products/fuzzylogic). The training process was run for one epoch, the original fuzzy logic knowledge base was automatically generated using grid partition and each input was assigned two labels. For each system we collected its Root Mean Square (RMS) training error (the error when the testing file is the same 66% used for training) and its RMS testing error (the error when the testing items used were the 33% of the file not used for training).

Figure 8 presents the training and testing error for each subset of one variable (6 subsets) ordered using the training error. Figure 9 to Figure 11 present the same results for each subset of two variables (15 subsets), three variables (20 subsets) and four variables (showing the 15 subsets with smaller training error). From Figure 8, we can obtain that IN2 (AS in Table 2) provides the better training error, 0.4448, and IN1 (BS in Table 2) the best testing error, 0.3499. Figure 9 corroborates the importance of IN2 for training error, because the smallest training errors are always obtained using IN2 as one of its inputs. Also, again, the smallest testing error is obtained in combination with IN1, by the pair of inputs IN1-IN6 (GB in table 2). Figure 10 and Figure 11 show that,
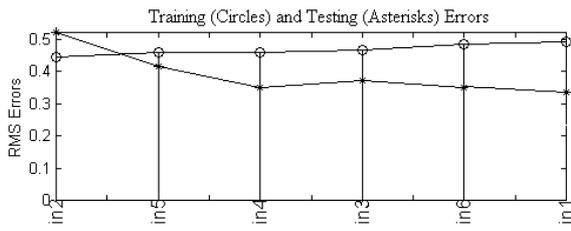
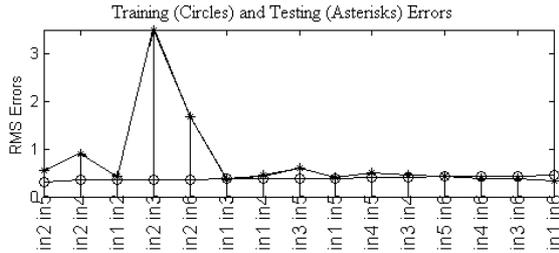FIG. 8. RMS Error for one-dimensional systems.

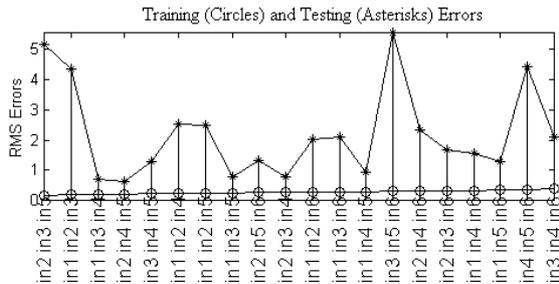FIG. 9. RMS error for two-dimensional systems.
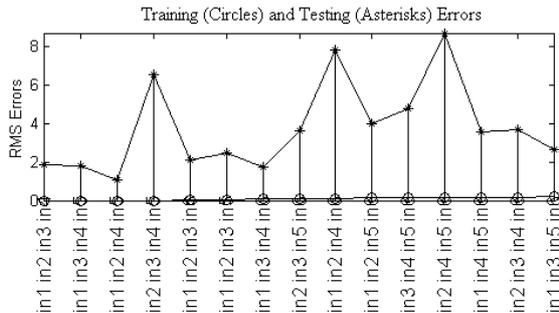

FIG. 10. RMS error for three-dimensional systems.

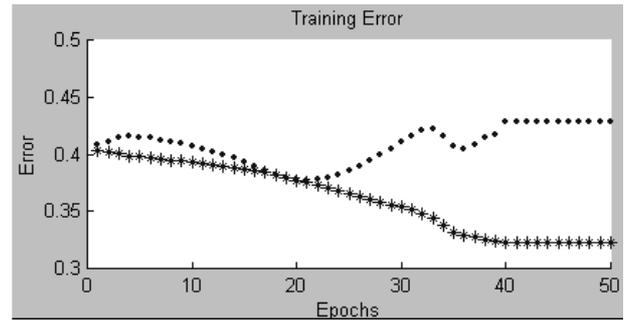FIG. 11. RMS error for four-dimensional systems.

FIG. 12. Training error of the neuro-fuzzy system.
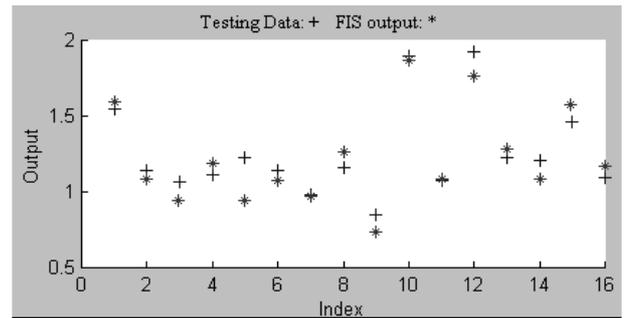
FIG. 13. Comparison between the testing WA ratios (+ signs) and the predicted WA ratios (* signs).

although the incorporation of a third and fourth variables can produce subsets of inputs with similar training errors, testing errors are much higher. Typically the set of variables would be chosen according to the smallest testing error, which produces IN1-IN6 as input variables (testing error of 0.3442). Choosing the set of variables with smallest testing error does not imply that these variables will produce the best solution, because each neuro-fuzzy system was run for just one epoch, and there is no indication about how both testing and training error will evolve. Also, considering that in this case IN1-IN6 has the highest training error for the set of two variables, we also decided to choose a set of variables that, while having a

testing error similar to IN1-IN6, also had one of the smallest training errors, in other words, the combination of variables with the smallest testing error among the ones with the smallest training error. Considering this approach we selected the pair IN1-IN2, which has a testing error of 0.3597 but a smaller training error than IN1-IN6. Also, this combination of variables is very promising because it combines the variable that minimizes the training error (IN2), with the variable that minimizes testing error (IN1).

We reach the conclusion that an optimum system in the sense of: (1) size of the fuzzy knowledge base, (2) training and testing time and (3) efficiency of the classification system, would be achieved by a two dimensional system with a possible combination of BS and AS or BS and GP as inputs. The following sections checks which one of these combinations produces better results.

Result Analysis: Neuro-Fuzzy Cognitive Classification

Two neuro-fuzzy systems were constructed using the Fuzzy Logic Toolbox implementation of ANFIS as learning algorithm. The training vectors were composed of two variables: BS and AS (variables 1 and 2 of Table 2) in the first case, and BS and GP (variables 1 and 8 of Table 2) for the second case. The independent variable was in both cases the WA ratio (variable 8 of Table 2). Both neuro-fuzzy systems were designed not to directly give the cognitive style, but to obtain the WA ratio. In order to test each neuro-fuzzy system two testing techniques were applied: (1) 66%-33% splitting and (2) 3-cross validation. In both cases, the original fuzzy system was automatically generated using grid partition, and learning run for 50 epochs. The training algorithm selected the fuzzy inference system that minimized the testing error within

the first 50 epochs. Figure 12, where dots represent testing error and asterisks the training error, shows the learning process and how the minimum testing error is obtained in epoch 21 and has a value of 0.38 when using BS-AS as inputs. Both fuzzy system are quite simple with two inputs (BS-AS or BS-GP), one output (WA value), three membership functions per input, three output singletons and only three rules.

Figure 13 shows the output of the testing file when using 66%-33% splitting for the system constructed with BS-AS. As can be seen, although the exact WA ratio is not predicted, the system is able to give very approximate values that actually classify the user, in general, in the correct cognitive style. Nevertheless, users that have a WA ratio near cognitive style borders have a higher probability of being incorrectly classified. Using splitting the correct classification rate obtained is 82% with BS-AS and 75% with BS-GP while 3-cross validation provides a 76% correct classification rate with BS-AS and 73% with BS-GP. The results show, that for our problem, BS-AS captures better that BS-GP the characteristics of each cognitive style. These results are better than the ones provided using just a rule-based approach, as done by C&ART, which implies that the soft computing approach captures, to some extent, the uncertainty of modelling cognitive styles. Also the reduction of the input space allows to implement an ANFIS system with a real-time response, which is specially critical for personalization of applications.

Cognitive Classification from a Personalized Application Perspective

As we said previously, the concept of cognitive style (Field Dependent / Intermediate / Field Independent) is actually constructed using the concept of WA ratio (a real number in the range of 0.6-3.0), in which WA scores below 1.03 denote Field Dependent individuals; scores of 1.36 and above denote Field Independent individuals; and scores between 1.03 and 1.35 are classified as Intermediate. Such classification is given by Riding (1991), but other values for classification of cognitive styles are also possible, i.e. the borders between cognitive styles are fuzzy. Taking that idea into account and also considering that our cognitive identification system will be part of a personalized environment for BLC, it can be possible that some of the users that have been assigned to an incorrect cognitive style can find the interface assigned to them by personalization useful. Users with WA ratio near cognitive borders have a higher probability of being incorrectly classified; nevertheless it also implies that these users, to some extent, share the behaviours of its neighbours, so they can also find useful their personalized interface.

Taking into account the above observation, we can define the concept of "being a user near a cognitive border" as a user whose WA ratio is within (-0.1, +0.1) of the border of a cognitive style. Considering the previous assumption, those users would find useful the two possible personalized interfaces associated with each one of their two valid cognitive styles. For example, a user with a WA ratio of 1.01 could be classified as Field Dependent, or considering that 1.01 is included in [1.03±0.1], with 1.03 the border value between FD

TABLE 4. Classification and Regression results from an application perspective.

	3- Cross Validation	66% Split
Regression		
C&RT	77.4%	79.2%
MLP	73.5% (5 Neurons)	71.4% (5 Neurons)
NFS		
ANFIS – (BS,AS)	91.5%	100%

and Intermediate, it can also be classified as Intermediate. It would be interesting to find out the correct classification rate of the proposed regression systems taking into account this definition. Table 4 presents the correct classification rates using MLP, C&RT, ANFIS with BS-AS (which provided better results) and the new concept of classification rate. Note that when using ANFIS we achieve, in the worst case, a 91% correct classification rate. These results, which arouse from an application perspective, show that the automatic identification of a user cognitive style within BLC is feasible, and opens the door to automatically personalize BLC to each one of its users from a cognitive perspective without the need to run a cognitive style test.

Conclusions

The cognitive style of a user is a very relevant factor to determine the way in which a user interacts with a web-based service. This importance implies that cognitive personalized services can be very useful to tackle the different problems that users have when interacting with Internet, especially in environments as relevant as digital libraries. The main drawback of considering a cognitive personalized interface is that in order to assign a cognitive style to a user, each user needs to take a cognitive style test. This process is time consuming and some users would not be willing to take it.

In this paper we have proposed an approach to overcome the inconvenience of implementing a cognitive interface by automatically identifying each user's cognitive style. We have reached two main conclusions: (1) In order to better identify the CS of a user, due to the fuzziness of the definition of cognitive style, a regression approach, in which we obtain the WA ratio, outperforms a classification approach, in which the cognitive style is directly identified, and (2) in general, and considering that we are modeling human behavior, the use of a soft computing approach improves the classification rate. Also we have proposed a correct classification rate definition from an application perspective, in which users near cognitive borders can have two correct cognitive styles. We have focused our study in what we consider very relevant Internet tools such as digital libraries, using Brunel Library Catalogue as a testing environment. The results obtained by using a neuro-fuzzy approach in this context have shown that the system can be applied to automatically generate user models

for cognitive personalization, thus avoiding the main inconvenience of constructing cognitive personalized interfaces. We consider that these improvements open the door to actually implement high-scale personalized DL based on cognitive styles.

We plan to study how this methodology applies to, first, other digital libraries/search interfaces and, second, to other applications/interfaces. Initial studies have shown that FD/FI is consistent across domains and stable over time (Witkin, et al, 1977; Witkin and Goodenough, 1981). This implies that once we have constructed a cognitive user model, the same model can be applied to any application and it will not change over time. We plan to check these theoretical results from an application perspective to see if our approach can actually be applied to a variety of applications.

Acknowledgments

References

Bajraktarevic, N., Hall, W., & Fullick, P. (2003). Incorporating Learning Styles in Hypermedia Environment: Empirical Evaluation. In The Fourth Conference on Hypertext and Hypermedia, Workshop on Adaptive Hypermedia and Adaptive Web-based systems AH2003.

Beck, J., Jia, P., Sison, J., & Mostow, J. (2003). Predicting Student Help-Request Behavior in an Intelligent Tutor for Reading. In Proc. of the 9th Int. Conf. on User Modeling, LNAI 2702, 303-312

Bidel, S, Lemoine, L., & Piat, F. (2003). Statistical machine learning for tracking hypermedia user behavior. In 2nd Workshop on Machine Learning, Information Retrieval and User Modeling, 9th Int. Conf. in UM.

Blandford, A., Stelmaszewska, H., & Bryan-Kinns, N (2001). Use of multiple digital libraries: a case study. In Proceedings of the JCDL'01, ACM Press.

Bonissone,P., Badami, C., & Chiang, X. (1995). Industrial Applications of Fuzzy Logic at General Electric. In Proceedings of the IEEE, 83(3), 450-465.

Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J. (1984). Classification and Regression Trees. Wadsworth Int. Group/Probability Series, Belmont, California, USA.

Brusilovsky, P., & Schwarz, E. (1997). User as Student: Towards an Adaptive Interface for Advanced Web-Based Applications. In A. Jamesson, C. Paris and C. Tasso (Eds.), User Modeling: Proceedings of the Sixth International Conference, UM97, 177-188.

Bryan-Kinns, N., Blandford, A., & Thimbleby, H. (2000). Interaction Modelling for Digital Libraries. In Workshop on Evaluation of Information Management Systems.

Callan, J., Smeaton, A., Beaulieu, M., Borlund, P., Brusilovsky, P., Chalmers, M., Lynch, C., Riedl, J., Smyth, B., Straccia, U., & Toms E. (2003). Personalization and Recommender Systems in Digital Libraries, Joint NSF-EU DELOS Working Group Report , URL http://www.ercim.org/publication/ws-proceedings/Delos-NSF/Personalisation.pdf.

ChanLin, L. (1998). Students' cognitive styles and the need for visual control in animation. Journal of Educational Computing Research, 19(4), 351-363.

Chen, S. Y., & Ford, N.J. (1997). Towards adaptive information systems: individual differences and hypermedia. Information Research, Vol.3 (2).

Chen, S.Y., & Macredie, R. (2002). Cognitive Styles and Hypermedia Navigation: Development of a Learning Model. Journal of the American Society for Information Science and Technology, 53(1), 3-15.

Chuang, Y-R. (1999). Teaching in a Multimedia Computer Environment: A study of effects of learning style, gender, and math achievement. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 1(1) 1999. Available online: http://imej.wfu.edu/articles/1999/1/10/.

Demuth, H., & Beale, M. (1998). Neural Network Toolbox User's Guide. Ver. 3.0, The Mathworks, Inc.

Drigas, A., Kouremenos, S., Vrettos, S., Vrettaros & J., Kouremenos, D. (2004). An expert system for job matching of the unemployed. Expert Systems with Applications 26, 217-224.

Dufresne, A. & Turcotte, S. (1997). Cognitive style and its implications for navigation strategies. In B. Boulay and R. Mizoguchi (eds.) Artificial Intelligence in Education Knowledge and Media Learning System. IOS Press, Amsterdam, 287-293.

Fausett L., Fundamentals of Neural Networks, Prentice-Hall, 1994

Fink, J., Kobsa, A., & Nill, A. (1997). Adaptable and Adaptive Information Access for All Users, Including the Disabled and the Elderly. A. Jamesson, C. Paris and C. Tasso (Eds.), User Modeling: Proceedings of the Sixth International Conference, UM97, 171-173.

Ford, N. & Chen, S. Y. (2000). Individual differences, hypermedia navigation and learning: an empirical study. Journal of Educational Multimedia and Hypermedia, 9(4), 281-312.

Goodenough, D. (1976). The role of individual differences in field dependence as a factor in learning and memory. Psychological Bulletin, 83, 675-694.

Graff, M (2003). Cognitive Style and Attitudes Towards Using Online Learning and Assesment Methods. Electronic Journal of e-learning, Vol. 1, Issue 1, pp. 21-28

Harrison, A. W. & Rainer, R. K. (1992). The influence of individual differences on skill in end-user computing. Journal of Management Information Systems, 9(2), 93-111.

Haykin S., Neural Networks, 2nd Edition, Prentice Hall, 1999

Hong, J., Heer, J., Waterson, S., & Landay, J.A. (2001). WebQuilt: A Proxy-based Approach to Remote Web Usability Testing. ACM Transactions on Information Systems, Vol. 19 (3), 263-385.

Jang, J.S. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference Systems. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23(3), 665-685.

Jang, J.S.R. & Sun, C.T. (1995): Neurofuzzy modelling and control. Proceedings IEEE (3,) Vol. 83, 378-406.

Jang, J.S.R., Sun, C.T. & Mizutani E. (1997). Neurofuzzy and Soft Computing: a Computational Approach to Learning and Machine Intelligence. Prentice Hal.

Kass, G.V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. Applied Statistics 29: 119-127.

Klir, J. & Yuan,B. (1995). Fuzzy Sets and Fuzzy Logic. Theory and Applications. Prentice Hall.

Kobsa, A. (2001). Generic User Modeling Systems. User Modeling and User-Adapted Interaction 11, 49-63.

Lee, R.S.T. (2002). iJADE IWShopper: A New Age of Intelligent Web Shopping System Based on Fuzzy-Neuro Agent Technology. Web Intelligence: Research and Development, Zhong, N., Yao, Y., Liu, J., Ohsuga, S. Editors, Lecture Notes in Artificial Intelligence, vol 2198, 403-412.

Liaw, S. & Huang, H. (2003). An Investigation of User Attitude toward search engines as an information retrieval tool. Computers in Human Behavior, Volume 19, Issue 6, November 2003, 751-765

Magoulas, G.D., Papanikolau, K.A. & Grigoriadou, M. (2001). Neuro-fuzzy synergism for planning the content in a web-based course. Informatica Vol. 25(1), 39-48.

Marchionini, G., Plaisant, C. & Komlodi, A. (1998) "Interfaces and tools for the Library of Congress National Digital Library Program". Information Processing & Management, 34(5), 535-555

Marrison, D. L. & Frick, M J. (1994). The Effect of Agricultural Students' Learning Styles on Academic Achievement and their Perceptions of Two Methods of Instruction. Journal of Agricultural Education, 35(1), 26-30.

Martinez, L. W., & Martinez, A.R. (2001). Computational Statistics Handbook with MATLAB. Chapman & Hall / CRC.

Messick S. (1976). Individuality in learning. San Francisco: Jossey-Bass.

Mitchell, T. (1997). Decision Tree Learning. In Machine Learning, McGraw-Hill, Inc., 52-78.

Moss, N, & Hale, G. (1999). Cognitive Style and its Effect on Internet Searching: A Quantitative Ingestigation. In European Conference on Educational Research, Lahti, Findland 22-25.

Paliouras, G., Karkaletsis, V., Papathedorou, C., & Spyropoulos, C. (1999). Exploiting Learning Techniques for the Acquisition of User Stereotypes and Communities. In Proceedings of the International Conference on User Modelling (UM '99).

Palmquist, R. A. & Kim, K.-S. (2000). Cognitive style and on-line database search experience as predictors of Web search performance. Journal of the American Society for Information Science, 51(6), 558-66.

Papanikolau, K.A., Grigoradiou, M., Kornilakis, H., & Magoulas, G.D. (2003). Personalizing the Interaction in a wed-based Educational Hypermedia System: the case of INSPIRE. User Modelling and User-Adapted Interaction 13: 213-267.

Peterson, E. R., Deary, I. J., and Austin, E. J. (2003) The reliability of Riding's Cognitive Style Analysis test. Personality and Individual Differences, 34(5), 881-891.

Quinlan, J.R. (1986). Induction of Decision Trees. Machine Learning (1), 81-106.

Quinlan, J.R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann.

Riding R. J. & Grimley, M. (1999). Cognitive style, gender, and learning from multimedia materials in 11 year-old children. British Journal of Education Technology, 30(1), 43-56.

Riding, R. & Rayner, S. G. (1998). Cognitive Styles and Learning Strategies. David Fulton Publisher, London.

Riding, R. J. (1991). Cognitive Styles Analysis. Learning and Training Technology, Birmingham.

Sas, C., Reilly, R., & O'Hare, G. (2003). A Connectionist Model of Spatial Knowledge Acquisition in a Virtual Environment. In 2nd Workshop on Machine Learning, Information Retrieval and User Modeling.

Sheperd, A., Watters, C., Marath, A.T. (2002). Adaptive User Modeling for Filtering Electronic News. In Proc. of the 35th Annual Hawaii Intl. Conf. on System Sciences (HICSS-02), Vol. 4.

Stathacopoulou, R., Grigoriadou, M., & Magoulas, G.D. (2003). A Neuro-fuzzy Approach in Student Modeling. Proceeding of the 9th Int. Conf. on User Modeling, UM2003, Lecture Notes in Artificial Intelligence, vol. 2702, 337-342.

Theng, Y.L., Duncker, E., & Mohd-Nasir, N. (1999). Design Guidelines and User-Centred Digital Libraries. Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries, LNAI 1696, 167-183.

Triantafillou, E., Pomportsis, A., & Georgiadou, E. (2003). AEC-CS: Adaptive Educational System based on Cognitive Styles. In WASWBE'02, Workshop on Adaptive Systems for Web-based Education, 2nd Intl. Conf. on Adaptive Hypermedia and Adaptive Web Based Systems.

Tsukada, M., Washio, T., & Motoda, H. (2001). Automatic Web-Page Classification by Using Machine Learning Methods. In Web Intelligence: Research and Development, LNAI 2198, 303-313.

Webb, G. I., B. C. Chiu, & M. Kuzmycz (1997). Comparative Evaluation of Alternative Induction Engines for Feature Based Modelling. International Journal of Artificial Intelligence in Education, 8:97-115.

Weller, H. G., Repman, J., & Rooze, G. E. (1994). The relationship of learning, behavior, and cognitive styles in hypermedia-based instruction: Implications for design of HBI. Computers in the Schools, 10(3/4), 401-420.

Winston, P. (1992). Learning by Building Identification Trees. Artificial Intelligence. Addison-Wesley Publishing Company, 423-442

Witkin, H. A. & Goodenough, D. R. (1981). Cognitive styles: Essence and origins: Field dependence and field independence. New York, NY: International Universities Press

Witkin, H. A., Moore, C. A., Goodenough, D. R., & Cox, P. (1977). Field-dependent and field independent cognitive styles and their educational implications. Review of Educational Research, 47(1), 1-64.

Witten, I.H., & Frank E. (1999). Data Mining. Practical Machine Learning Tools and Techniques with JAVA Implementations. Morgan Kaufman Publishers.

Yan, J., Ryan, M. & Power, J.: (1994).Using Fuzzy Logic. Prentice Hall.

Zhu, T., Greiner, R., & Haubl, G. (2003). Learning a Model of a Web User's Interests. In Proceedings of the 9th International Conference on User Modeling, LNAI 2702, 11-21.

Zukerman, I., Albrecht, D.W., & Nicholson, A.E. (1999). Predicting Users Request on the WWW. In Proceedings of the 7th International Conference on User Modeling, UM99, 275-284.