International Journal of Computer Science & Applications © 2006 Technomathematics Research Foundation
Vol. 3 Issue 3, pp 52 - 64

Implementations of Web-based Recommender Systems Using Hybrid Methods

Janusz Sobecki

Institute of Applied Informatics
Wroclaw University of Techmology, 50-370 Wroclaw,
ul. Wyb. Wyspianskiego 27, Poland
e-mail: sobecki@pwr.wroc.pl

Abstract

Application of hybrid recommendation enables overcoming disadvantages of the basic recommendation methods: demographic, collaborative and content-based. In this paper the two consensus-based hybrid recommendation methods are presented. Then some examples of their implementations to different web-based systems are shown.

1 Introduction

With growing popularity of the web-based systems that are applied in many different areas, they tend to deliver customized information for their users by means of utilization of recommendation methods. We can distinguish three basic recommendation approaches: demographic, content-based and collaborative. The demographic approach uses stereotype reasoning, which is mainly classification problem, in its recommendations [6] and is based on the information stored in the user profile that contains mainly different demographic features to generate initial predictions about the user [7]. Content-based recommendation takes descriptions of the content of the previously evaluated items to learn the relationship between a single user and the description of the new items [7]. In this method a user is supposed to like a new item if the item is similar to other items that are liked by the user [1]. The collaborative recommendations are able to deliver recommendations based on the relevance feedback from other similar users. Its main advantages over the content-based architecture are the following [7]: the community of users can deliver subjective data about items together with their ratings and it is able to offer completely new items to the particular user.

All the above mentioned methods have some disadvantages. The demographic recommendations have two basic disadvantages [7]: they may be too general and they do not provide any adaptation to user interests changing over time. Content-based method tends to overspecialize recommendations and it is based only on the particular user relevance. Collaborative recommended agents have also some disadvantages: they offer poor prediction when the number of similar users is small and there is a lack of the transparency in the predictions.

The disadvantages of all the above mentioned recommendation methods may be overcome by application of the hybrid solution. For example, the lack of adaptation in the stereotype may be easily overcome by application of collaborative or content based recommendation. On the other hand, disadvantages of the collaborative approach of the insufficient number of the similar users at the early stages of the system operation may be overcome by application of the demographic stereotype reasoning [14]. Finally some disadvantages of content-based recommendation, such as overspecialization, may be overcome by application of collaborative approach [14].

In the following section the user model in the hybrid recommender system is defined. In the section three two hybrid recommendation methods are presented. The section four contains description of different implementations of these two hybrid methods applied for different web-based systems and finally, in the summary the efficiency of the hybrid approach and future work is discussed.

2 User model in the web-based hybrid recommendation

A user model contains knowledge about the individual preferences which determine his or her behavior within the system [11]. The main problems with the user modeling encompass the user model representation and acquisition. Here the following user model elements will be presented: user profile representation, user profile initialization, distance and similarity functions among user profiles and user profile clustering methods.

2.1 User profile representation

User profile should contain all necessary data to model the user in recommender system. The user profile in the recommender systems could be represented by many different forms: binary vectors, feature vectors, trees, decision trees, semantic networks, Bayesian networks, etc. The feature vector is the most popular among many recommender systems [7]. In this paper, the user profile is represented by a tuple that have similar descriptive strength as the feature vector but it is used in many previous works on information systems and consensus methods that is applied for the recommendation in the systems presented in this paper. The tuple is a function $p: A \rightarrow V$, where A is a set of attributes and V is a set of their elementary values and $(\forall a \in A)(p(a) \in V_a)$.

The user profile usually contains two types of data: user data and usage data [6]. The user data may contain different user characteristics: demographic data (name, gender, address, etc.), user knowledge, user skills and user interests. The usage data are observed directly from the user's interaction with web-based system. The usage data concerning web-based system user interface recommendation may contain information on the interface layout, the information content and its structure. The usage data may also contain usability measure value of the particular interface settings [4].

However the quality of the system interface may be difficult to measure the domain of the HCI has worked out several methods to evaluate the user interface for example [8]: heuristics, questionnaires and user tests.

2.2 User profiles initialization

The initial profile may be empty, especially in case of content-based recommendations. In other approaches the initial profile is quite often created from the questionnaire that is filled in by the user. The questionnaire usually contains information on user data that contains different information, i.e. demographic data containing: record data (name, address, phone number, e-mail), geographic data (city, region, country and zip-code), user's characteristics (sex, education, occupation), and some other customer qualifying data. The user data may also contain information on users' knowledge, their skills, interests and preferences and also their plans and goals.

The usage data, the second element of the user model, is observed and recorded during the whole process of user's interactions with web-based systems. It may concern selective operations that express users' interests, unfamiliarity or preferences, temporal viewing behavior, as well as ratings concerning the relevance of these elements that are expressed by different events, such as: opening a page, purchasing a product, sending feedback information to the system are stored. There are of course more sophisticated and general methods for gathering such data, as for example DoubleClick mechanisms. DoubleClick enables tracking the user web activities and serving personalized advertisements [17] by using cookies entries with unique identifiers and placing web-bug image on every page that is to be tracked.

The initial profile may be also modified according to the whole user population behavior, which is the case of collaborative recommendation. However, this brings some problems with finding similar users that could be solved by clustering methods [5].

In all implementations presented in this paper, the user profile is created from the questionnaire that is filled in by each user during the registration process that is obligatory for each user. This is necessary because we use demographic recommendation. The user profile is also modified during working with the system by each particular user. These data is then used in the collaborative and content based recommendation.

2.3 Distance and similarity between user profiles

The distance function between values of each attribute of the user profiles is defined as a function δ^{at}: $V_a \times V_a \rightarrow [0,1]$ for all $a \in A$. This function should be given by the system interface designer and fulfill all the distance function conditions, but not especially all the metrics conditions, and should be determined for each attribute and its every pair of atom values. The distance function values could be enumerated or given in any procedural form.

The distance between user profiles could be defined in many different ways. First, the distance between tuples *i* and *j* could be defined as a sum of distances between values of each attribute:

$$d(r_i,r_j) = \sum_{a\in A} d^{at}(r_i(a),r_j(a)).$$

Second, the root of the sum of squares of these distances, or finally third, we also can indicate the importance of each attribute *a* by multiplying the distance by appropriate factor defined as a function c: $A \rightarrow [0,1]$:

$$d(r_i,r_j) = \sum_{a\in A} [c(a) * d^{at}(r_i(a),r_j(a))] .$$

However, it is also possible to use some similarity functions instead of distance functions. Very popular among them is the cosine similarity function [13]:

$$s(u_j,u_k) = \frac{\sum_{i=1}^{m} u_{k,i} * u_{j,i}}{\sqrt{\sum_{i=1}^{m} (u_{j,i})^2 * \sum_{i=1}^{m} (u_{k,i})^2}} ,$$

where u_j and u_k are vectors of real values of *m* attributes describing users *j* and *k* respectively.

We can find quite many different similarity functions for example in the work [13], the function used in the Dattola clustering algorithm [2] is shown below:

$$s(u_j, u_k) = \sum_{i=1}^{m} \min(u_{k,i}, u_{j,i}) .$$

In all implementations presented in this paper, we applied the third distance, the one with a factor that defines importance of each attribute. This distance is rather easy to implement, however quite expressive. For finding the similarity between the user profile and items description we used the cosine similarity function that is the most popular one especially in the area of Information Retrieval (IR).

2.4 User Profile Clustering

The clustering problem is defined as partitioning the given set of users U={u$_1$,…,u$_n$} into subsets of set U according to some optimization criterion. We can distinguish three major types of clustering algorithms [3]: hierarchical, Euclidean or similar metric space and similarity matrix. The most popular are those algorithms that belong to the family of Euclidean or similar metric space clustering.

Then the clustering optimization criteria could be described as finding such partition of the set U into p disjoint subsets C_i $i=1,..,p$ of users such that the distance among all the members of each class - $d(C_i)$ is minimal:

$$d(C_i) = \sum_{j=1}^{r} \sum_{k=1}^{r} d(u_j, u_k),$$

where r= Card(C_i), or the distance between all p groups as stated below is maximal:

$$d(U^p) = \sum_{i=1}^{p} \sum_{j=1}^{i-1} d(C_i, C_j),$$

$$where$$

$$d(C_i, C_j) = \sum_{l=1}^{r} \sum_{k=1}^{q} d(u_l, u_k)$$

and r=Card(C_i), q=Card(C_l).

The analogous criteria for the similarity functions may be also shown. Computation complexity of the problem of partition set U with n users into two clusters is exponential with respect to n. So, practically other sub-optimal, however more effective, algorithms should be used. They usually are based on the selection of some initial partition as for example in the Dattola method [2], presented below:

Start

In the beginning we divide the set U={u_1,…,u_n} of n users into k initial classes: $C_{1,1}$, $C_{1,2}$,.., $C_{1,k}$. For all of the classes we calculate the centroid. The centroid of the class C_j is denoted as O_j=($o_{j,1}$,$o_{j,2}$,.., $o_{j,m}$} where,

$$o_{j,s} = \begin{cases} 0 \quad if \quad f_{j,s} = 0 \\ b - r_{j,s} \quad otherwise \end{cases}$$

where b is a priori assumed constant

$$f_{j,s} = \sum_{u_i \in C_j} u_{i,s}$$

$$r_{j,s} = 1 + \max\left\{f_{j,t} : t = 1,2,..,m\right\} - f_{j,s}$$

Then in t-1 iteration we have the partition $C_{t-1,1}$, $C_{t-1,2}$,.., $C_{t-1,k}$ with adequate centroides $O_{t-1,1}$, $O_{t-1,2}$,.., $O_{t-1,k}$. Let T be the threshold used to construct new classes $C_{t,j}$ that is determined in the following way:

$$C_{t,j} = \left\{ \begin{matrix} a_i : s(a_i, O_{t-1,j}) \geq T \; and \\ s(a_i, O_{t-1,j}) = \max\{s(a_i, O_{t-1,l}) \; for \; l = 1,2,.., k\} \end{matrix} \right\}$$

All the user profiles that where not included into any class should be inserted into set L_t of isolated objects. New centroides are determined as described above if the following condition is fulfilled:

$$\sum_{a_i \in C_{t,j}} s(a_i, O_{t,j}) > \sum_{a_i \in C_{t,j}} s(a_i, O_{t-1,j}),$$

otherwise $O_{t,j}$:= $O_{t-1,j}$. Users from the set of isolated objects L_t . The iteration ends when for particular t and all $j=1,2,..,k$ occurs $O_{t,j}$= $O_{t-1,j}$. Then the objects from L_t are treated as a separate class or joined to those classes to which they are the most similar.

End.

The other way to solve the clustering problem, quite similar to Dattola algorithm is so-called Lloyd's algorithm [5] and is one of the most popular solutions to the k-means problem. The Lloyds's algorithm has the following steps. First, select randomly k elements as the starting centers of the clusters (centroides). Second, assign each element of the set to a cluster according to the smallest distance to its centroid. Third, recompute the centroid of each cluster, for example the average of the cluster's elements. Finally, repeat steps 2 and 3 until some convergence conditions have not been met (for example centroids do not change).

This algorithm is rather simple and has an ability to reach the end when using the above mentioned convergence condition and for configurations without equidistant elements to more than one centroid, it takes a long time to run. First, the step 2 that has to be performed in each iteration costs $O(kdN)$, where d is the dimension of each element and N is the number of elements. Second, algorithm usually needs many iterations to terminate.

There are however quite many modification of this algorithm that run faster, for example bisecting k-means, that begins with single cluster containing all the elements, then splits it in two clusters and replaces it by split clusters.

Both presented here algorithms were applied in recommender systems, which implementations will be described in the section 4.

3 Consensus-based user interface recommendation

Consensus theory has its general origins in the social sciences and in the theory of choice in particular [9]. The social choice theory considers problems of analyzing a decision between a collection of alternatives made by a collection of different voters with separate opinions and the selected choice should reflect the desires of all the individual voters to the possible extent [16]. The main difference between consensus theory and the choice theory is that the former one does not necessitates the solution belonging to the set of opinions under consideration. In this section the model of consensus and hybrid recommendation is presented.

3.1 Model of consensus

In the consensus model we distinguish the following elements: conflict system, conflict profiles and conflict determination. Within this model it is assumed that a real world domain is described by means of a finite set *A* of attributes and a set *V* of attribute *elementary values* [9]. Furthermore let $B \subseteq A$ and a tuple of type *B* is a function r_B: $B \rightarrow \Pi(V_B)$ where $(\forall b \in B)(r_b \subseteq V_b)$, and the set of all tuples of type *B* is denoted by *TYPE(B)*.

In the scope of the consensus-based hybrid recommendation of a web-based system user interface, as the source of opinions we may assume each client in the distributed system, that could be also called agents, or different events of the specified user client. The subjects of agents' interest consist of events occurring in the world, i.e. mainly observing user behavior, interface settings and its usability values. These observations are called events and are stored as attribute values in a tuple of some type. The conflict system definition is a modification of the one presented in the work [10]. Here the notion of the category is introduced for differentiation of the recommendation type: demographic, collaborative and content-based. The second difference relies on the identification of the information sources. In the conflict system defined in [9] agents are the source of different (conflict) information. The conflict system defined here as the source of conflict information identifies not only separate agents but also depending on the category also different events from the given agent. We can distinguish some subset $T \subseteq A$ that contains attributes for event identification.

<u>Definition 1</u>

A conflict system of some category c is a quadruple: $S^c = (A, X, P, Z)$*, where:*

A – is a finite set of attributes (as defined above), including attributes that identify each event;

X – is a finite set of conflict carriers, X ={P(V_a*): a Î A};*

P – is a finite set of relations on carriers from X, each relation is of some type L (for L Í A and L contains attribute or attributes that enable to identify the observation from set T Í L);

Z – is a finite set of logic formulas for which the model is a relation system (X,P).

Relations from the set *P* are classified in such a way that each of them includes relations representing similar events. Here these observations and events concern the different aspects of the user model. A conflict situation for a given category *c* of the conflict system S^c contains information about a concrete conflict as defined below.

Definition 2

A conflict situation of a given category c cs^c *is a pair <P, Y® B>, where Y is a set of attributes that have influence on the interface settings: Y Í L\T and B Í L\T and YÇ B=Æ and* r_Y*¹ q for every tuples r Î P.*

A conflict situation consists of event identifiers (conflict body) which appear in relations *P* (conflict content) representing the observed (or induced) knowledge of referring to subjects represented by set *B* of attributes, in this case interface settings. Expression $Y \rightarrow B$ means that in the observed events there are differences referring to combinations of values of attributes from Y with values of attributes from B, and the purpose of the consensus choice is that for a tuple type Y at most one tuple of type B should be assigned.

For a given situation cs^c, we determine the set of events which take part in the conflict as the projection of the set of relations P to the set of attributes K, Event(cs^c) = ?^{K}(P), where $K \subseteq A$, and K is a key of relation P. The set of subject elements (or subjects for short) is defined as the projection of the set of relations P to the set of attributes Y: Subject(cs^c) = ?^{Y}(P) where $Y \subseteq L \backslash T$. Then for each subject $e \in$ Subject(cs^c) let us determine set with repetitions Profile(e) which include knowledge from events on subject $e \in Subject(cs^c)$, as the set of relations that identify the given subject e reduced to the set of attributes $B \cup K$ of for and they are included Profile(e)=$\{r_{B \cup K}$: $(r \in P) \wedge (e \prec r_A)\}$.

The definition of consensus is based on the definition given in [9].

Definition 3

*Consensus on subject e*Î*Subject(*cs^c*) of situation* cs^c*=<P,Y®B> is a tuple (C(*cs^c*,e)) where C(*cs^c*,e)*Î*TYPE(Y*È*B) that fulfils logic formulas from set Z and one of the following postulates are fulfilled: knowledge closure and consistency, Condorcet consistency condition for choice and convergence condition.*

The following theorem should enable to determine a consensus that satisfies all the postulates from the above definition. The proof of this theorem is some modification of the one presented in the work [9].

Theorem 1

If there is a defined distance function d between tuples of TYPE(B), then for a given subject e of situation cs^c*=<P,Y®B> tuple C(*cs^c*,e) which satisfies conditions of Definition 3 and minimize the expression* $\sum_{r \in \mathrm{P}rofile(e)} d(r_B, C(cs^c, e)_B)$ *should create a consensus satisfying all postulates from the definition 3.*

When all the attributes from the user profile are independent then the consensus determination in the Profile(e) is reduced to the determination of consensus for each attribute in the tuple of *TYPE(B)*. Then depending on the microstructure of attribute values such as 1-element sets or sets of values, and macrostructure of their universe (distance function definition) different algorithms for consensus determination could be distinguished.

In case of the simplest microstructure when an attribute $a \in A$ is represented by 1-element sets of values from some set V_a, the consensus determination in the profile is based on the consensus choice function from the *Theorem 1*. In case of other microstructures such as number intervals, rankings and sets the algorithms for consensus are more complicated and they can be found in work [10].

Some examples of the conflict systems, of Event(cs^c), Subject(cs^c) and Profile(e) for each category, as well as consensus determination within the profile will be shown in the following sections.

3.2 Consensus-based recommendation methods

The first consensus-based recommendation method was presented in previous works [10] and [14]. This method of the user interface recommendation was based on the hybrid approach that was the mixture of the demographic and the collaborative recommendation with some components of the content -based approach.

The system adaptation (see figure 1) starts with registering each new user. The registration data are stored in the user profile. Then according to the user profile the user is assigned to the appropriate group. With each group of users there is associated a corresponding interface profile. The user registration is not obligatory but in this case the default interface profile is delivered.

According to the interface profile the actual user interface content, layout and structure is generated. The user may start to work with the system and if he or she wishes also modify the interface settings. Finally, these settings together with usability evaluation given by user are stored in the interface profile.

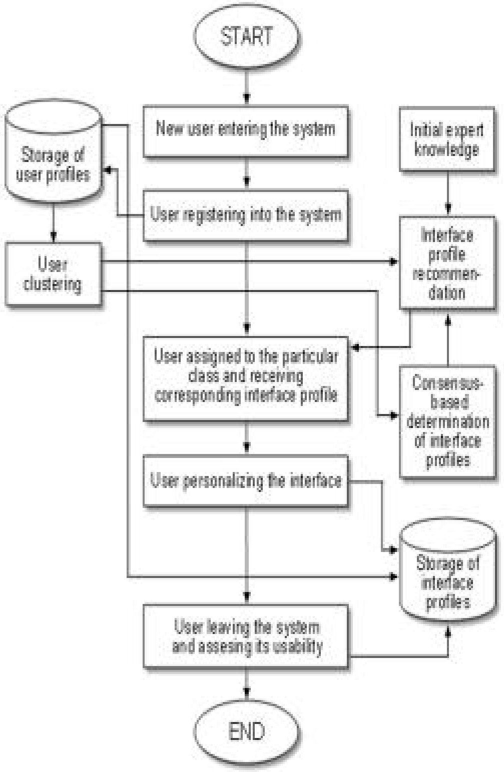
START
New user entering the system
Storage of user profiles
Initial expert knowledge
User registering into the system
Interface profile recommen-dation
User clustering
User assigned to the particular class and receiving corresponding interface profile
Consensus-based determination of interface profiles
User personalizing the interface
Storage of interface profiles
User leaving the system and assesing its usability
END

Figure 1. Architecture of the consensus-based user interface adaptation [14]

When the system registers required number of users, first users are clustered using Dattola algorithm and then according to these clusters using consensus methods new interface profiles for recommendation are distinguished. These procedures may be repeated from time to time in the following cases: many new users registering to the system or interface recommendations become poor usability ratings.

The second method of the consensus-based hybrid recommendation using demographic, collaborative and content-based approaches applied to different components of the user model. In this paper we will only give a short description of the three types of recommendation, the more precise one may be found in [15].

In the user profile represented by the set of attributes A we can distinguish the following subsets: the demographic attributes set D, the demographic attributes of the centroid classes of users set N ($N \subseteq D$), the set of recommended interface settings I, the set of the interface setting attributes made directly by the user J, the set of the actual interface setting attributes together with usability valuation values F, the set C of attributes associated with the content (for example visited pages, purchased or ordered items, retrieved elements) and finally, we can distinguish some attributes used for identification and authorization purposes T. So the set of attributes equals the sum of its elements: $A=D \cup J \cup I \cup C \cup T$.

Each of the determined centroids has the corresponding interface settings, which is assigned by the expert and is recommended to the user after registering to the system by delivering demographic information (entering values of attributes from the set N).

In the implementation described in work [14] we considered only single expert opinion, we can assume however that more than one expert opinion is allowed and then determine consensus among all the opinions.

In case of the multiple expert opinions we can distinguish two situations. In the first one, all the experts share the same centroid attribute values (concerning demographic attributes) but give different opinions on interface settings. In he second situation, all the experts give opinions on centroids settings and corresponding user interface attributes values.

In both cases, however, we try to find consensus for each distinct centroid. Concerning the second case, the number of centroid may increase significantly. When they outnumber the desired limit we can group them, find new centroids and then find the consensus within these groups.

The application of collaborative recommendation is possible when significant number of users have been registered, used the system, personalized the interface and delivered ranked the interface usability. More precisely we must have the group of similar users G concerning values of demographic attributes from the set D.

The user groups are identified by the centroids, determined by the user clustering described above. For each group corresponding interface settings are entered into the same consensus profile and then the consensus is determined by finding the values of the interface settings for which:

$$\sum_{r \in \mathrm{Profile}(e)} r_{usability} * d(r_B, C(cs^c, e)_B),$$

is minimal, where attribute set $B=J \setminus \{usability\}$. We should notice that comparing to the expression from the *Theorem 1* the distance is multiplied by the value of the usability of each particular interface.

To deliver content-based recommendation for a particular user we must have sufficient usage data of that user and appropriate inductive rules that will transform this data into the user interface settings.

The rules for efficient content-based recommendations strongly depend on the goals of the web based system. For example for web-based information retrieval systems we can consider the previous relevant items as a basis for recommendation of further retrievals. In this case many different methods can be used: fuzzy retrieval, Bayesian networks or other intelligent information retrieval method. For quite many systems however, the logic used for the retrieval systems, does not hold. So, for each recommended item, no matter if an element of interface settings or a content item, we shall define precise relationship between the user profile (or also other users profiles) and this particular item, which may be implemented for example as ruled based system, Bayesian or neural networks. However the usage data may lead to many different recommendations so there is a place for consensus determination.

Beside above mentioned recommendations: demographic, collaborative and content-based, we should also mention other ones, such as: environment, situation or emotion based. This kind of recommendations may be dealt in two different ways. The first one is based on the expansion of the subject's attribute set with the attribute concerning platform, situation or emotions in standard collaborative or content consensus-based recommendation. The second method treats these recommendations as separate ones.

The consensus system for that kind of recommendation is similar to the collaborative one. However despite grouping users according to the demographic features we can group them according other attributes describing computer environments, situation of use, etc.

The result of consensus determination in all the categories: demographic, collaborative, content-based, environment or situation based is a recommendation of the user interface for the particular user. Obviously there can be significant differences in the recommendation of each user interface attributes. So the question arises, which type of recommendation should be preferred?

In the cases when only one type of recommendation is available and the others do not deliver any (*null* value of the recommendation) then it is obvious that we should use the one that is available. However usually three or more types of recommendation may deliver different settings. In such cases specific selection rules should be applied. Some examples that describe several implementations of web-based systems user interface and content recommendations, will be given in the following section.

4 Implementations of web-based hybrid recommender systems

In this section we will present six selected web-based recommender systems, first in the subsection 4.1 three systems that are based on the combination of demographic and collaborative filtering and then in the subsection 4.2 the other three systems that are based on the complete hybrid recommendation.

The first car information system [14] was implemented by M. Weihberg within his master thesis work supervised by the author of this paper, who also carried out series of tests that were necessary to verify recommendation method. The other systems were implemented by students of the course "Interactive web-based information systems design" curried out in the academic years 2003/2004 and 2004/2005 supervised by the author of this paper. The application domains of recommender systems were following: windsurfing, digital cameras (two systems), notebooks, computer news, movies (two systems), cooking, motor-bikes, mobile phones (two systems) and CD's with music (two systems). Here we will present five of them applied in the following areas: hair-dressing, mobile-phones, cooking, movies and computer news.

4.1 Implementations using demographic and collaborative filtering methods

In the car model information system that was started in 2003 we have chosen Renault Megane II (c) the model of the car that was awarded with Car of The Year 2003. That choice was made in order to attract as many users as possible and also because of the ease of acquiring all the necessary materials to build the system. This system had no commercial application and was used only in our laboratory. The car model presentation system is quite simple and so is the user profile. The data stored in the user profile are entered by the users during the registration process. The user data are reduced to only few attributes of demographic information and one that characterizes user's interests. The interface profile attributes that may be personalized by the user or recommended by the system, contain information concerning interface layout, music and sounds, information content and usability factor. The attributes values of the initial centroids for stereotype reasoning were selected by experts so that none of the stereotype had all the extreme (maximal or minimal) values and the distance between consecutive centroids was similar.

The effectiveness of the recommendation was tested in controlled conditions by 75 users that were students of masters' degree and postgraduate studies. The users were of different age, gender, education, musical and graphical taste, preferences concerning the system layout and information content. The tests were conducted in three steps. In the first step a group of the users registered themselves and they were assigned to the appropriate group according to the smallest distance to the centroids. According to these assignments corresponding interface profiles were delivered to them. In this step users were not allowed to personalize their interfaces and at the end the users were asked to fill-in the questionnaire concerning seven usability aspects, i.e. information content, visual content, interaction etc. In the second step a new group of users was asked to register themselves and personalize their user interface according to their preferences starting from the settings assigned by stereotype reasoning. Then they were asked to assess the general usability with four grades scale. At this point recommendation procedures were carried out. In the third step the last group of users, as in the first step, was assigned to the appropriate groups according to their distance to the centroids. Then according to these assignments, corresponding interface profiles ware delivered to the users. Again after working with the system and obtaining all desired information on the car model users were asked to fill-in the usability questionnaire. Comparison of the all the user interface usability aspects between results obtained in the first step and the third step showed that the interface adaptation results in receiving higher scores. The raise of marks was rather small (0.3 in average in the 10 point scale) but was encountered in all aspects, so user interface recommendations delivered by adaptive procedures were better than those delivered by experts, but both were rated very high - 8.42 before and 8.72 after the adaptation.

The second system that was also implemented using the first recommendation method was "Have-a-Look!" - hair-dress information system, by K. Górka and E. Wasilewska. The system recommends the following interface settings: layout, font size and type, text and background color, hints, sound track and loudness. The user profile attributes concern some demographic features such as: age, profession, interests; and some specific system domain features such as: reason for using the system and how often the user changes hair-dress style. The recommended user interface usability of this system was tested using two methods questionnaire and user tests that showed that the interface is easy to use even for not very experienced users and any major usability problems were encountered.

The third system by M. Ciesla and A. Siodlak is presenting selected mobile phone model. The system recommends the following interface settings: layout, colors, music and music loudness. The user profile attributes concern only some demographic features such as: age, gender, education and life-style. The system was also tested using questionnaire and user tests methods. Questionnaires showed that implemented interface recommendation delivers better settings than the ones delivered by experts however user tests showed some usability problems that should be considered in a real system.

4.2 Implementations using hybrid recommendation methods

The following three implementations used the second hybrid recommendation method that takes into account also content-based approach. In "The cooking assistant" by M. Podyma & M. Swieczak recommender systems, different types of recommendation were used: basic and hybrid ones. The demographic, that basing on the age and gender of the user offers different interface settings such as: music track, volume, font size and hints; and content settings: additional information and wine selection. These settings may be changed by the users so it is possible to find consensus among these settings and offer them as a new recommendation for similar users. This implementation also delivers situation recommendation that offers receipts for breakfast, lunch or dinner according to the daytime. Finally also content-based recommendation is applied according to the preferences explicitly stated by the users concerning preferred cuisine and specified food products. The system usability was tested with 12 users using the questionnaire method and the classical usability tests [12]. The test proved that the system usability is satisfactory.

In "The Movies" by F. Luczys & M. Filarska recommender system also different types of recommendation were used. The system delivers information about movies stored in the system "Stopklatka" and current cinema repertoire from that information system. The user model is initialized during registration process, where each user is asked to give ratings of selected films (as in MovieLens system). Then according these ratings each user is grouped according to the stereotype reasoning specific interface settings concerning layout, types of icons, color and background are recommended. The user profile settings concerning different movie attributes such as: genre, director, writing, music, cast and cast are changed according to the explicit user ratings of films or specified attributes and implicit user searching and browsing of movies.

The system delivers reach collaborative recommendation that concerns: interface settings according to the changes delivered by the similar users, sorting the current cinema repertoire for unregistered users according to the preferences of all the system users and optionally also for registered users according to the preferences of all similar users, three extra recommended movies according to the highest ranks given by the similar users. The content based recommendation for registered user that sorts the cinema repertoire according to the preferences reflected in the user profile. By default the hybrid recommendation that is mixture of collaborative and content-based repertoire sorting is applied. The conducted usability tests containing the questionnaire method and the classical usability tests showed some minor problems with interaction but most of the users appreciated implemented recommendation mechanisms.

Finally, the simplest system "Comp News" by L. Sliwko & K. Jankowski recommends computer news. At the beginning users delivers some demographic data, preferred interface style and their interests on four topics: purchase optimization, over-clocking, hardware modding and new products, in form of a feature vector. Then news is sorted according to the similarity with the interest vector, which is subject to constant changes according to reading consequent news. The user may also select an average interest vector determined for the group of similar users. Even that simple approach was assessed by the experimental users as being pretty useful.

5 Summary

In this paper only six hybrid recommendations implementations were shown. However, many other web-based recommender systems were implemented by students of the course "Interactive web-based systems design" and masters works in recent two years, proving that it is possible to implement hybrid recommendation in many different areas. In these implementations the consensus methods were usually used in the collaborative recommendation, but it is also possible to apply the consensus methods for demographic, content-based and also combined hybrid recommendations.

It is quite difficult to test recommender systems, especially in controlled condition, because many different users are necessary to show how collaborative method operates, as well as each user needs rather long time of working with the system to show how content-based method operates. However all of the systems were tested with tens of users (one almost 80) and some methods were tested with most exhaustive user tests method, which methodology suggest to test only five users. These tests showed the increase of some usability factors after application of system recommendation.

References

[1] Dastani M, Jacobs N, Jonker CM, Treur J, Modelling User Preferences and Mediating Agents in Electronic Commerce. LNCS 1991, 2001, pp. 163-193.

[2] Dattola RT, A fast algorithm for automatic classification. Report ISR-14 to the National Science Foundation, Section V, Cornell University, Department of Computer Science, 1968.

[3] Estivil-Castro V, Yang J, Categorizing visitors dynamically by fast and robust clustering of access logs. LNAI 2198, 2001, pp. 498-507.

[4] International Standard ISO 9241-11, Ergonomic requirements for office work with visual display terminals (VDT's) – Part 11: Guidance on Usability, 1997.

[5] Kanungo T, Mount DM, Netanyaho NS, Piatko C, Silverman R, Wu AY, An Efficient k-means clustering algorithm: analysis and implementation. *IEEE Trans. On Pat. Analysis And Mach. Intell.*, 24 (7), 2002, pp. 881-892.

[6] Kobsa A, Koenemann J, Pohl W, Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships, *The Knowledge Eng. Review*, 16(2), 2001, pp. 111-155.

[7] Montaner M,. Lopez B, de la Rosa JL, "A Taxonomy for Recommender Agents on the Internet", *Artificial Intelligence Review*, 19, 2003, pp. 285-330.

[8] Newman WM, Lamming MG, Interactive system design, Addison-Wesley, Harlow, 1996.

[9] Nguyen NT, Consensus System for Solving Conflicts in Distributed Systems, *Journal of Information Sciences*, 147, 2002, pp. 91-122.

[10] Nguyen NT, Sobecki J, Using Consensus Methods to Construct Adaptive Interfaces in Multimodal Web-based Systems, *J. of UAIS.*, 2(4), 2003, pp. 342-358.

[11] Papatheodorou C, Machine Learning in User Modeling, Machine Learning and Its Applications, 2001, pp. 286-294.

[12] Pearrow M, Web Site Usability Handbook, Charles River Media, 2000.

[13] Rijsbergen C, Information Retrieval. Butterworths, London, 2nd edition, 1979.

[14] Sobecki J, Weihberg M, Consensus-based Adaptive User Interface Implementation in the Product Promotion, in Keates S (et al.), Design for a more inclusive world, Springer-Verlag, London, 2004, pp. 111-121.

[15] Sobecki J, Consensus-Based Hybrid Adaptation of Web Systems User Interfaces, *Journal of UCS*, 11(2), 2005, pp. 250-270.

[16] Social Choice Theory, http://mathworld.wolfram.com/SocialChoiceTheory.html, online Oct. 2004.

[17] Whalen D, The Unofficial Cookie FAQ, Version 2.54 Contributed to Cookie Central by David Whalen, http://www.cookiecentral.com/faq/#2.7, online 2002.