In 6th European Conference on Case Based Reasoning, ECCBR2002, pp 395-405, Springer Verlag Lecture Notes Series, September 2002, Aberdeen, Scotland.

Category-Based Filtering and User Stereotype Cases to Reduce the Latency Problem in Recommender Systems

Mikael Sollenborn and Peter Funk

Mälardalen University
Department of Computer Science and Engineering
Västerås, Sweden
mikael.sollenborn@idt.mdh.se, peter.funk@idt.mdh.se

Abstract. Collaborative filtering is an often successful method for personalized item selection in Recommender systems. However, in domains where items are frequently added, collaborative filtering encounters the *latency problem*. Characterized by the system's inability to select recently added items, the latency problem appears because new items in a collaborative filtering system must be reviewed before they can be recommended. Content-based filtering may help to counteract this problem, but runs the risk of only recommending items almost identical to the ones the user has appreciated before. In this paper, a combination of category-based filtering and user stereotype cases is proposed as a novel approach to reduce the latency problem. Category-based filtering puts emphasis on categories as meta-data to enable quicker personalization. User stereotype cases, identified by clustering similar users, are utilized to decrease response times and improve the accuracy of recommendations when user information is incomplete.

1 Introduction

Personalization on the Internet is today a growing research area, as the information overload problem has created an emerging need for individualized user treatment. By focusing on each visitor's requirements, the user's effort in navigating vast amounts of information can be made more focused, efficient and manageable. The underlying idea of personalization is the assumption that individualized content will satisfy users and increase revenue directly or indirectly, e.g. attract new users and make them more willing to revisit a web site and buy more services and products [1].

For personalization of web pages, Recommender systems are currently the most common approach. Based on the information filtering technique known as automated collaborative filtering (ACF) [2, 3, 4], standard Recommender systems essentially function on a peer review basis. When making recommendations, users with similar preferences are identified, and their item ratings are used to propose items to one another. Implementation of an ACF Recommender system can be divided into three steps [5]:

1. Record the behavior of a large number of people, e.g. their interest in selected items such as adverts, news, books, etc.
2. Select a number of users who's past behavior is similar to the current user.
3. Recommend personalized items based on preferences of the selected users.

In addition to collaborative filtering, personalized selections based on matching the current user's previous selections with individual items - known as content-based filtering - is also very common [6]. In short, where filtering with ACF involves comparing a user with other users, content-based filtering is performed by comparing the user's preferences with the available information about items, e.g. meta-data or content keywords.

One potential problem with standard Recommender systems is that all reasoning is done online. With impatient users waiting for quick responses, the search for similar users must be very time-efficient. This time restriction also results in fewer possibilities when trying to improve or extend the content filtering strategies. In order to improve both speed and recommendation effectiveness, current approaches to building Recommender systems often try to perform some of the reasoning offline using clustering techniques [7, 8].

Traditional Recommender systems also encounter the *latency* problem [9], i.e. new items incorporated into a Recommender system cannot be used in collaborative recommendations before a substantial amount of users have evaluated it, as the recommendations rely on other users opinions. This problem is especially apparent in domains where new items are often added and old items quickly get out of date. Content-based filtering may be a solution, but runs the risk of only recommending items almost identical to the ones the user has appreciated before [9]. As noted in [10], the most obvious solution to the latency problem is to categorize the items in the system. In this paper we go one step further and assume that for some applications domains, Recommender systems solely based on categories provide sufficient personalization.

Our proposed approach for reducing the latency problem in highly dynamic domains is called category-based filtering. In a category-based filtering system, user preferences reflect attitudes not towards single items, but categories of similar items, both on a collective and an individual level. At the collective level, off-line clustering is used to find user stereotype cases, thus employing a Case-Based Reasoning view of information filtering. Clustered user data enables quicker response times and makes collaborative reasoning possible for meta-data in the form of categories.

In section 2 the category-based filtering approach is explained. Section 3 gives a more detailed exploration of classification, clustering and item selection. The research prototype, a personalization system based on category-based filtering and user clustering, is briefly described in section 4, and the following section gives a conclusion.

2 Category-based filtering approach

In this section we explore how category-based filtering is used in a Recommender system and how it is integrated with clustering and user modeling.

2.1 Rating technique

Typically, rating methods are divided into invasive and non-invasive techniques. An invasive rating method requires explicit user feedback. A commonly used approach is to let users mark their appreciation of items viewed or purchased on a scale. In contrast, non-invasive methods observes the user's behavior, requiring no more input than the user's normal interaction with the system. As a result, non-invasive methods generate noisier data, but have the benefit of being invisible to the user.

For our purposes, i.e. dynamic domains where data changes frequently, invasive techniques put too much burden on the users. Instead, a simple non-invasive technique was chosen. The system selects a set of items to show, and observes user reactions to these items. The system notes whether the user responds positively, by clicking on one of the currently shown items, or negatively, by ignoring them. We do not consider viewing time following a click, because the number of responses to a category of items will be many times as high as that for single items in a representationless ACF system, making the consequences of a single click less relevant.

2.2 Category-based filtering

We refer to the personalization approach proposed in this paper as *category-based filtering*. Its main characteristic is that selection of information is based on category ratings instead of item ratings, in contrast to other content-filtering strategies in general, and representationless collaborative filtering in particular. To function, category-based filtering requires categorization of every item, either manually or by an automated process.

In our implementation of category-based filtering, the selection of items is based partly on individual user models, and partly on collective user stereotypes cases. A *user model* represents the current knowledge about a user's reaction towards shown categories of items. A *user stereotype case*, in contrast, consists of collective information about a group of users.

User stereotypes, as introduced by Rich in [11], require two types of information. The system must know what properties capture a stereotype, and what events or behavior that implies a particular stereotype. On the Internet, this information is highly dynamic and in our domain dependent on both content of categories and population of users. A clustering approach, as described below, is therefore preferable to static stereotypes, since it is able to automatically identify related categories and adapt to a changing population of users and their preferences.

By representing a solution to the problem of supplying a 'typical' kind of user with appropriate information, it is natural to see user stereotype cases as part of a Case-Based Reasoning process. When the information in a user model is insufficient for deciding which items to select, the user stereotype case most closely resembling the user is consulted to make assumptions about the user's expected behavior (Retrieve, Reuse). The case is Revised when the user evaluates the recommended items, and Retained when the user stereotype cases are updated.

The system could also be seen as a hybrid of collaborative and content-based filtering, with strong emphasis on categories as item meta-data. Unlike other such hybrid systems [12, 4], the collaborative selection is also based on meta-data, as the peer reviewing process deals with categories instead of items.

The focus on categories reduces the latency problem, as new items can be recommended as soon as the system knows the user's attitude towards the item's corresponding category. Because of this, selecting items based on category ratings instead of ratings of individual items is especially suited for domains where there is a constant flow of new information (e.g. news and adverts), provided that effective categorization is possible.

As category-based filtering could possibly be seen mainly as an extension to existing filtering strategies, one might feel inclined to propose the terms category-based collaborative filtering and category-based content-based filtering instead. However, apart from the clumsiness of these expressions, the term category-based collaborative filtering has been used for other purposes [13].

The user stereotype cases needed for collaborative selection of items are created offline using clustering methods described in section 3. Each cluster represents a part of the entire user population. Probabilistic nets within the cases, formed from the cluster data, represent collective attitudes towards categories of items.

With the clusters identified, a new user can be assigned a user stereotype case after a short period of initial observation. As the user model matures, the case assignment may change to point out the characteristics of the user in a more precise way.

The frequency of generating clusters and updating the user stereotypes cases depends on the application domain, e.g. the number of visits to the web site it's being used on. All individual user information is always preserved, enabling the system to perform a re-clustering at any time.

2.2.1 User models

Based on the dimensions identified in [11], a user model in the proposed system has the following properties: each user has a separate user model, the model is built and refined non-invasively by the system on each site visit, and the model contains both specific, short-term information and (limited) long-term information.

A user model is represented by a matrix of choices and preferences For each category, the number of times the user has been approached with items belonging to it is stored, as well as the number of positive responses. Table 1 shows an example preference matrix with a simplified history ("Last ten"-column) that would capture sudden changes of user preferences. In this example, only two clicks the last ten times the user was approached with hunting items may indicate a decline of interest for such adverts.

Table 1. Example preference matrix

	Shown	Clicked	Last ten
Hunting	24	11	2
Fishing	18	4	3
Cosmetics	12	1	0

When the preferences of a user are to be ranked, the value for each category may be reduced, e.g. to one of four levels: positive, neutral, negative, or unknown. The unknown attitude is reserved for categories that have not yet been evaluated.

2.2.2 User stereotype cases and appreciation nets

User stereotypes cases are representations of common attitudes among a group of similar users. The chosen method of capturing collective interests is to utilize what will be referred to as appreciation nets. Appreciation nets are graphs with nodes and directed edges, where edges represent a probabilistic relationship. If every node in the net has an edge going into every other node, the appreciation net is said to be complete, with n(n-1) node connections. In Figure 1 an example of an appreciation net is given for four item categories. In this example population the likelihood that a person who likes hunting is also interested in motor sports is 60% (indicated by 0,6 at the edge from hunting to motor sports). In the opposite direction, a person that appreciates motor sports also enjoys hunting with a probability of 30%. Of all the persons belonging to this population, 50% enjoy motor sports, but only 20% appreciates hunting, as indicated in the category nodes.

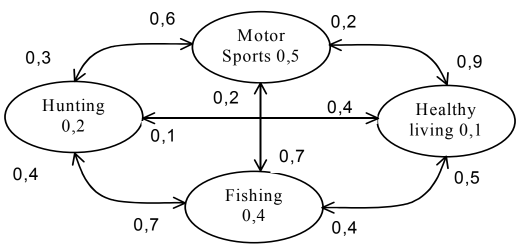

Fig. 1. An appreciation net with four item categories

2.3 System architecture

Figure 2 shows a schematic view of a system using category-based filtering. Each user visiting the web site is assigned an individual user agent. The agent's task is to handle web page modifications and interaction with the user involving personalized items. The Reasoner uses category-based filtering to select a set of items assumed to be of interest for the current user, based on the user model (the user's preference matrix) and the closest user stereotype case. The user agent tracks all user responses and stores them in the user model. The cases are updated offline (as indicated by the dotted line) by clustering similar user models.

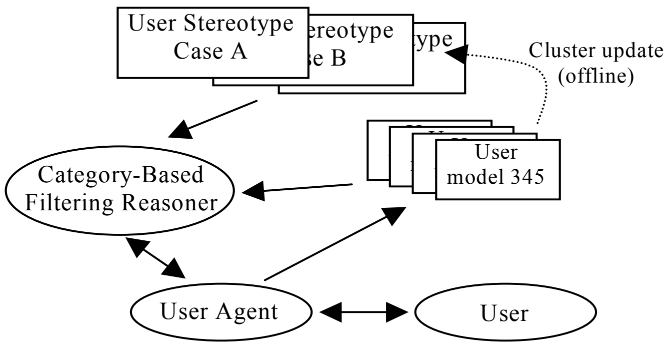

Fig. 2. Example of a personalization system using category-based filtering

3 Clustering and selection

This section takes a closer look at techniques and algorithms used for personalization in a category-based system.

3.1 Clustering users

For the creation of user stereotype cases, an agglomerative, hierarchical clustering method was chosen, avoiding partitioning since the number of appropriate clusters will be difficult to guess in advance. The variables determining cluster membership are as many as there are categories in the system, but categories that have not yet been evaluated by a user are not included when comparing him/her to other users. Different values are assigned to the category attitudes negative, neutral, and positive. To measure distance between clusters, the city-block (Manhattan distance) metric is used. Similar clusters are merged using the unweighted pair-group average method [14].

After clustering, the (highly subjective) optimal number of clusters must be determined. Currently, the chosen method is to pick a maximum number of clusters M based on the number of users N registered on the web site. The minimum amount of users U belonging to a cluster is then calculated as U = N/M, meaning we treat a group of U or more similar individuals as "statistically significant" to form a collective model. Now, traversing the cluster tree selecting clusters of size U or bigger results in a number of clusters from 1 to M. Although this method produces acceptable results for the limited test domain, more formal approaches for determining the number of clusters, such as LMM or BIC, are considered for the final implementation.

As noted earlier, a user stereotype case contains an appreciation net, with all nodes connected to every other node in both directions. When forming such a net, a joint distribution is made from the ranked category preferences of every user belonging to the group, resulting in a two-dimensional matrix exposing collective preferences. Building the user stereotype case is primarily a question of how much information from the joint distribution to include in the appreciation net.

For each category C the system stores the probability of a positive evaluation by any user belonging to the group, as node values in the appreciation net. This information is very important because for categories where P(C) is high, quick tests can be made to determine whether new users conform to a specific cluster.

Secondly, couplings between pairs of categories are examined. The probability of a user appreciating C in case the user likes D, P(C|D), is preserved for each category-to-category connection, stored as binary relationships between category nodes in the appreciation net.

The dominating preference information can be captured in binary relations between category nodes. Some preference information may also be captured in probability values involving more than two category nodes (e.g. if users are interested in category D and E the likelihood for interest in C is P(C|D∧E)). A calculation of probability values among all possible n-tuples of relations may have a too high computational price and make the resulting model unnecessarily complex. Considering the possibly increased inference ability gained from probability relations involving three of four categories, such probability values may be worth preserving in the appreciation net if the values are distinctive enough (i.e. exceptionally low or high).

3.2 Classification of users

Automatic classification is attempted by targeting the user with information corresponding to differential probability values in the user stereotype cases appreciation nets. The goal is to determine which case resembles the new user the most.

Beginning with the biggest cluster is an adequate starting point, because it's where the user most likely belongs. What is sought for is a number of category nodes in the appreciation net with high appreciation probabilities, with these values being as unique as possible compared to the equivalent category values in other cluster models. Categories with unique but low appreciation probabilities are not as interesting, as a positive response can be considered a lot more informational than a negative, i.e. a user showing interest probably *is* interested, but a user that doesn't may simply be temporary ignorant or short of time. The appropriateness of being chosen is calculated for every category node *C* by comparing it to the corresponding category node C_i in every other case appreciation net, using

$$F = P(C) * \sum_i \left|P(C) - P(Ci)\right| \ . \qquad \textbf{(1)}$$

The three categories with highest F are chosen for testing, meaning information belonging to these categories will be shown to the user, with five items per category. This may or may not take several visits to the site, depending on the type of site and how much information can be shown normally during a visit. When all items have been shown, an initial user stereotype case membership determination is performed. The formula resembles a Naive Bayesian Classifier, but sums probabilities instead of multiplying them to avoid having occasional conditional probabilities close to zero produce an unsuitably low total similarity value. The categories involved in the test are compared to each user stereotype case, putting emphasis on similar categories with high probability values (again because it's more important what is appreciated than what is not), and limiting the comparison to categories already tested. The similarity S is calculated for every user stereotype case, where Ci is category i in the tested user's preference matrix, $Csui$ the corresponding category i for member u in user stereotype s, and M an empirically chosen modification rate, using

$$S = \sum_i P(P(Csui) = (P(Ci) \pm 0.1) \mid s) * (P(Csi) * M + (1 - M)) \, . \qquad \textbf{(2)}$$

The case that most closely resembles the initial behavior of the user (highest S) is now chosen for a second pick of categories using (1), separate from the ones chosen before. A new test is done, followed by another comparison using (2). This process continues every time the user visits the site. Eventually, the system will mix the testing data with items selected by assuming that the user does in fact belong to the cluster the user currently resembles the most, as well as re-evaluating categories that were positively responded to before. As more and more categories are evaluated, the amount of testing data ceases gradually. Cluster membership may still change, either because the number of clusters or the user's behavior has changed, but no longer as a result of evaluating "classification-aggressive" testing data.

3.3 Selection of personalized items

Once a user is assigned to a user stereotype case, the Reasoner (figure 2) is able to make qualified guesses about what a semi-new user might and might not appreciate. Whenever there is insufficient information about a user as an individual during decision-making, the case connected to the user will be examined to find out how similar users have behaved.

Asked to select personalized information for a specific user, the system initially decides whether or not it knows enough about the user's behavioral patterns to determine cluster membership. If not, the system will try to classify the user as described above. If cluster affiliation can be guessed but not completely determined, the system may alternately pick items it assumes the user will appreciate, while at the same time trying to further strengthen the belief that the user belongs to a specific cluster.

The Reasoner selects two types of information, appreciation-known and appreciation-assumed. A new user is confronted with a lot of appreciation-assumed information, but as the user provides more information, the appreciation-known information gradually replaces it.

Appreciation-known items are chosen only from categories the user has 'sufficiently evaluated'. A sufficiently evaluated category simply means a category that the user has evaluated enough times to be reasonably sure about the individual's attitude towards it. The number labeled sufficient varies however, as the system gradually tries to follow a new or semi-new user when providing more information. The system may also decide that a category needs re-evaluation if the user's last ten responses to it has been significantly different to the corresponding long-term behavior.

Appreciation-known information is selected by ranking the user preferences, picking items from categories that have been positively evaluated. In this process, the system tries to balance the number of shown items among the positive preferences, as well as sometimes picking sufficiently evaluated categories with a less positive ranking to allow for re-evaluation.

When appreciation-assumed items are to be selected, the system chooses a category node starting point in the appreciation net among the users positively ranked preferences. With this node as base, the system examines all connected category nodes. The category to select information from is chosen randomly from a dynamically generated pie chart, where each category not among the user's positive preferences gets a slice size (choice probability) calculated using equation 3. W is the connection weight, C is the number of clicks done on items belonging to this category, S the number of times shown to the user, and L how many of the last H items in the category that has been clicked by the user. H is domain dependent; in our test evaluation the history length is ten items, as shown in Table 1.

$$P = W * ((C + 1)/(S + 1) + (L+1)/H) \quad \textbf{(3)}$$

Another form of appreciation-assumed item selection, used in parallel with the method above, essentially uses the same method as the automatic classification process: picking items from categories in the appreciation net where the probability of a positive response is high. This item selection method is used only if there are still categories with high appreciation probabilities that the user has not yet evaluated.

The items selected by using each of these techniques are finally merged, and presented to the current user.

4 Implementation

The research prototype is currently being implemented, and has so far been tested on a small set of users in a limited surrounding. News and adverts were chosen as item types, as they both represent dynamic domains were items generally change often.

In the testing environment, users are shown a selected number of news summaries, containing approximately 200 letters. These news items are categorized manually in advance. The user is able to receive the full article by clicking on a news item. This information is used to build a preference matrix for the current user to aid in the category-based filtering approach. Adverts are handled similarly. A selected number of adverts are shown to the user and when clicked, additional product information is displayed.

To keep the user models reasonably small, item selections in categories are not given any time stamps, only the number of positive responses for the last ten times the item was shown (se example in Table 1).

A number of user stereotype cases have been initiated in advance, and off-line update of clustering is performed frequently in the form of refinement (a new cluster is generated from the same set of users). The re-clustering algorithm for grouping similar users into new clusters, as discussed in section 3.2, has been implemented but remains to be thoroughly tested and integrated. The appreciation nets currently only capture binary relations between item categories.

The prototype has been evaluated by a number of hypothetical users. The testing of the prototype has shown ability to quickly adapt to users preferences for a small number of categorized news and adverts (50 news and 70 adverts, 5 different hypothetical users with their interest profile predetermined and with a consistent behavior and low amounts of noise).

5 Conclusions

In this paper we have presented an approach to Recommender systems for application domains where items are frequently added. Provided that sufficient categorization is possible, we have shown that category-based filtering enables handling the latency problem.

In the proposed approach, users are represented partly by individual user models, and partly by user stereotypes cases. The cases, which are created offline through clustering, are used when the knowledge about an individual user is too limited to draw the needed conclusions for recommending items. The system will automatically attempt classification of new users by comparing the user's behavior with the user stereotype cases, selecting the most similar one.

Personalized information is divided into two categories: appreciation-known and appreciation-assumed. While the former represents item selections based on a user's known previous behavior, appreciation-assumed items are chosen because of high appreciation probabilities among other users belonging to the same user stereotype case as the current user.

So far, category-based filtering has been tested on hypothetical users in a limited surrounding, where the approach has shown the ability to adapt to user's needs.

Large-scale tests to further confirm the usability of category-based filtering for practical domains are currently being prepared.

References

1. Reichheld, F.F., Sasser, W.E.: Zero defections:quality comes to service. Harvard Business Review, 68:105-7, September-October (1990)
2. Shardanad, U., Maes, P.: Social Information Filtering: Algorithms for Automating 'Word of Mouth'. Proceedings of the ACM Conference on Human Factors in Computing Systems, ACM Press (1995) 210-217
3. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordan, L.R., Riedl, J.: GroupLens: Applying collaborative filtering to usenet news. Communications of the ACM 40 (1997) 3:77-87
4. Cotter, P., Smyth, B.: WAPing the Web: Content Personalisation for WAP-Enabled Devices. Proceedings of the 1st International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Trento, Italy (2000) 98-108
5. Mukherjee, R., Dutta, S.: MOVIES2GO – a new approach to online movie recommendations. In workshop Intelligent Techniques for Web Personalization, IJCAI-2001, Seattle Washington (2001) 26-34
6. Rafter, R., Smyth, B.: Passive Profiling from Server Logs in an Online Recruitment Environment. In workshop Intelligent Techniques for Web Personalization, IJCAI-2001, Seattle Washington (2001) 35-41
7. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data. In workshop Intelligent Techniques for Web Personalization, IJCAI-2001, Seattle Washington (2001) 53-60
8. Ungar, L.H., Foster, D.P.: Clustering Methods for Collaborative Filtering. Proceedings of the Workshop on Recommendation Systems. AAAI Press, Menlo Park California (1998)
9. Funakoshi, K., Ohguro, T.: A content-based collaborative recommender system with detailed use of evaluations. In Proceedings of 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, Volume 1 (2000) 253-256
10. Hayes, C., Cunningham, P., Smyth, B.: A Case-Based Reasoning View of Automated Collaborative Filtering. In Proceedings of 4th International Conference on Case-Based Reasoning, ICCBR2001 (2001) 243-248
11. Rich, E.: User Modeling via Stereotypes. Cognitive Science 3 (1979) 329-354
12. Balabanovic, M., Shoham, Y.: Combining Content-Based and Collaborative Recommendation. Communications of the ACM 40 (1997) 3:66-72
13. Kohrs, A., Merialdo, B.: Using category-based collaborative filtering in the Active WebMuseum. Proceedings of IEEE International Conference on Multimedia and Expo, New York (2000) 1:351-354
14. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall (1988)