

Comprehensive Exam

Topic: Recommendation Systems

Mihai Rotaru
Computer Science Department
University of Pittsburgh

1. Introduction

With the advent of Internet trading and the exponential increase of the number of items available, users are constantly confronted with situations in which they have more options than they can explore. In an ideal world, users are exposed only to the information of interest. Because in the real world information is not annotated with respect to user interests (which itself is a very complex and time consuming process), special systems need to be developed that sit between the vast amount of information and users. These systems are called Recommendation Systems (**RS**) and their main task is personalized information delivery and access.

The “information of interest” criterion is the one that separates Recommendation Systems from Information Retrieval systems. In the information retrieval approach, users know their interests and *how to translate* these interests into the query language supported by the information retrieval engine (typically, natural language keywords). From this point of view, RS are an extension of information retrieval systems: their task is to *learn* users’ interests and to use this information to *locate and recommend* items of interest.

The main task of a RS is to predict user’s ratings. Given certain background information (usually in form of previous users’ ratings), a RS needs to predict the rating that a certain user (*the active user*) will give to a certain item (*the active item*). Once a prediction is available for the active item, the rating is either presented to the active user, or more items are rated and a list of items ordered based on the predicted ratings is presented.

We distinguish between several approaches to RS based what type of information is being used to generate recommendations: information about the items, information about the users or information about both items and users. While other classification schemes are possible (e.g. type of interaction), we adopt this classification because it tells us which RS approach is appropriate for a given domain. In item-centered approaches, regularities between items of interest for a user are used to make predictions. In contrast, in user centered approaches, regularities between users’ properties are exploited to generate recommendations. Hybrid approaches try to exploit the complementarity between item-centered and user-centered approaches to improve prediction performance.

We conclude by identifying a number of issues regarding RS. We focus on issues that should be taken into consideration when designing and implementing a RS (cold start problems, evaluation metrics, interaction with the user, etc.).

2. Item-centered RS

Item-centered RS exploit properties of the items already rated by the active user to predict a rating for the active item. We begin by enumerating types of item properties used by previous work and then we focus on how this information was combined to produce a prediction.

The item properties exploited range from attribute values or description words to other users' rating of the item. For domains in which the rating is highly dependent on the item decomposition in attributes (e.g. cars, houses), the attribute values were used to derive predictions using Multi-Attribute Utility Theory (Jameson, Schafer, Simons, & Weis, 1995). In some cases, the attribute values were processed through a set of constraint functions and then combined to provide a rating (Linden, Hanks, & Lesh, 1997). For domains in which the connection between the rating and the item decomposition in attributes is not that clear (e.g. news stories), item description encoded as word occurrence vectors is usually used (Sarwar et al., 1998). To account for importance of words, Term Frequency – Inverse Document Frequency (TF/IDF) representation was also proposed (Billsus & Pazzani, 2000). Finally, for domains in which the connection between the rating and the item decomposition in attributes is very weak (e.g. movies), other users' rating of the items are used (Sarwar, Karypis, Konstan, & Riedl, 2001; Shardanand & Maes, 1995).

Based on how the item information is processed, we classify item-centered RS into two categories: *memory-based* and *model-based*. The memory-based approach corresponds to lazy learning algorithm in machine learning (case-based learning, memory-based learning (Rotaru & Litman, 2003)): in this approach, no explicit model is built out of the properties of the items already rated by the active user. The item properties and the item ratings are consulted and combined to produce a rating for the active item. In contrast, in the model-based approach (which corresponds to abstraction based learning in machine learning (Rotaru & Litman, 2003)), the item properties and item ratings are compiled in a model. Active item properties are processed through the learned model to produce a rating prediction.

Memory-based item-centered RS

In *memory-based* RS, the process of predicting the rating for the active item is performed in two steps: neighborhood formation and the neighborhood ratings combination. Neighborhood formation requires the retrieval of a number of items already rated by the active user that are similar to the active item. A similarity function is used to determine the degree of similarity between two items. Various similarity functions have been proposed in the literature. The cosine similarity (Billsus & Pazzani, 2000; Sarwar et al., 2001) measures the similarity between two items through the cosine of the angle between the vector representation of the two items. In (Sarwar et al., 2001) the item vector contains the other users' rating of the item. Because users use their own scale when rating items, (Sarwar et al., 2001) proposes an extension of the cosine similarity called adjusted cosine similarity: all vector are normalized with respect to the mean value of their components. In their experiments, the adjusted cosine similarity performed better than the regular cosine similarity suggesting that this metric should be used whenever the vector representation of items is not normalized. Another set of metrics commonly used are the correlation based similarities. Various correlation coefficients are used. The Pearson correlation coefficient was used in (Sarwar et al., 2001) and was found to perform similar to the cosine based similarity.

Once a number of items similar to the active item have been selected, the active user's ratings of these items need to be combined to produce a rating for the active item. In general this is done by taking a weighted sum of the ratings. The similarity between the active item and items in the neighborhood is usually used as weights (Billsus & Pazzani, 2000; Sarwar et al., 2001). Because the correlation and cosine based similarities can produce highly similarity values for vectors that are not normalized (e.g. one vector is equal to the other vector times a scalar), (Sarwar et al., 2001) proposes the use of linear regression to adapt the similarity weights. They find that the regular weighted sum performs better when the number of neighbors is small (less than 20) while the regression weighted sum performs better when a larger number of neighbors is used.

The memory-based approach has been extensively used in Collaborative Filtering RS. For this reason, the literature in Collaborative Filtering RS contains a larger number of similarity metrics and rating combination techniques. These techniques will be described later in Section 3. Nonetheless, they can be adapted easily to the item-centered memory-based RS.

Model-based item-centered RS

The memory-based approach has a big disadvantage: the similarity between the active item and the items previously rated by the user needs to be computed for each active item. If a large number of items needs to be rated, the computational cost of this approach is very high. In the model-based approach, researchers have tried to learn a model out of the items previously rated by the active user and use the model to predict the rating for the active item. The simplest approach is the mean vector approach. In this approach, a mean vector is computed for each user without looking at the active item. The dot product between the mean vector and the vector representation of the active item is used to compute the rating. (Pazzani, 1999) proposes using Winnow algorithm for computing the mean vector in a binary (like/dislike) prediction task (restaurant recommendation). (Good et al., 1999) uses the mean vector approach to create a set of FilterBots for each user (called Doppelganger bots). They use various parts of the movie information (cast, description) as item properties and represent the items using word occurrence vectors. The mean vector is the average of the vectors representing previously rated movies by the active user.

Other model-based approaches have cast the rating prediction task as machine learning problem and then applied standard machine learning algorithms. In this approach, the item properties are the features, the ratings are the classes and the training data is made of all previously rated items and their ratings. For example, (Billsus & Pazzani, 2000) applies Naïve Bayes in the news domain, while (Good et al., 1999) uses a well known learner, Ripper, in a movie RS. A large number of features is major problem for machine learning algorithms and may result in overfitting. Some learners have built in feature selection. Since (Billsus & Pazzani, 2000) Naïve Bayes implementation does not have any feature selection, they perform the feature selection before applying the learner: a set of relevant words for each news topic is identified and only those words are used when encoding new stories using TF/IDF vectors. Similarly, (Good et al., 1999) uses only the 200 most frequent words in the movie descriptions.

Several model-based approaches that focus on the user navigation of the domain have been proposed: utility-based RS and knowledge-based RS. Multi-Attribute Utility Theory (MAUT) approach is somewhat similar to mean vector approach (Faltings, Pu, Torrens, & Viappiani, 2004; Jameson et al., 1995; Linden et al., 1997). In this approach, each item is decomposed into a set of attributes. For each attribute there is a utility function that associates for each value a number representing its utility for the user. Each attribute also has a weight associated with it that measures the importance of that attribute for the user. The rating of an item is determined as the weighted sum of the utility of its attribute values. The main problem with this approach is that the utility function needs to be specified beforehand. (Jameson et al., 1995) uses Bayesian Belief Networks to reason about the importance of each attribute for the user. In their systems, the weights are adapted as interaction progresses based on the user feedback. An interesting application of MAUT for RS is presented in (Faltings et al., 2004; Linden et al., 1997). Instead of using item attributes, they use a set of constraint functions obtained from the user's critiques with respect to the item(s) displayed. Each constraint has a weight associated with it that encodes the likelihood that the user will accept violations of that constraint. The main advantage of MAUT approaches is that they can model specific biases for attribute values. For example, in the travel domain, if all attributes are the same, a flight with a smaller price is preferred over a flight with a higher price. These biases can not be handled in general by similarity based approaches.

Knowledge-based RS combine similarity retrieval with a set of navigation tweaks (Burke, 2001, 2002). These tweaks allow the user to navigate from item to item using higher level concepts (e.g. cheaper, better mileage, etc). Certain tweaks can be used to change more than one attribute in the same time (e.g. sportier

in the car domain). User actions (in terms of which navigation tweak was selected) can be recorded and standard Collaborative Filtering techniques can be applied to improve the navigation once enough users have used the system (Burke, 2002). The main disadvantage of knowledge based techniques is that the knowledge needs to be encoded in the system manually.

Except for utility-based and knowledge-based RS, all item-centered RS are also known as *content-based* RS.

3. User-centered RS

There are two types of user-centered RS: collaborative filtering RS and demographic RS. Collaborative filtering is by far the most frequently used approach and the most successful user-centered approach. It exploits regularities between users by examining their previous ratings. Demographic RS are less popular due to their relatively low performance. They exploit demographic information about the users. Nonetheless they can be used as a good starting point to augment existing RS if demographic information is available.

The term “*Collaborative Filtering*” (CF) was coined by (Goldberg, Nichols, Oki, & Terry, 1992). They were the first to acknowledge the importance of other users’ opinions for the process of information filtering. Their system, TAPESTY, was designed to assist users in finding relevant newsgroup articles. Up to that point, the paradigm for information filtering was that of content filtering (item-centered RS): users had to manually write rules that were exploiting the similarities between items they liked or disliked. But, just like in real life where people rely on other people for recommendations (e.g. movies, books, newspaper articles, research papers), (Goldberg et al., 1992) proposed using other users’ ratings (like or dislike) to improve information filtering. In their systems, users had access to the ratings of other users. The collaborative process had two stages. First, each user was supposed to find a “clique” of users with similar interest (e.g. colleagues that worked on similar tasks, etc). This is similar to neighborhood formation step in memory-based item-centered RS. In the second stage, users used the ratings of other users from that “clique” to improve the filtering process. This is similar to neighborhood ratings combination step in memory-based item-centered RS. Both stages required the user to supply the necessary information. Since then, research in CF RS has attempted to automate the two stages: how to automatically identify the “clique” of users with similar interests and how to automatically combine the ratings of similar users.

In a CF RS, the input is a collection of users’ ratings for a set of items. This information is represented as a matrix of ratings (Herlocker, Konstan, Borchers, & Riedl, 1999): rows correspond to user ratings, columns to ratings given to an item by users.

As with item-centered RS, we further classify CF approaches into two categories: memory-based and model-based. In the memory-based approach, the RS follows the two stages proposed in (Goldberg et al., 1992). In model-based, the two stages are implicitly performed by the machine learning algorithm used to learn the model.

Memory-based Collaborative Filtering

Memory-based CF is similar to the memory-based item-centered approach. Instead of computing the similarity between items based on their properties, memory-based CF computes the similarity between users based on their previous ratings. A large number of similarity metrics have been proposed (some of them were already discussed in Section 2). The most commonly used metrics are the cosine similarity and the Pearson correlation similarity (Breese, Heckerman, & Kadie, 1998; Herlocker et al., 1999). (Shardanand & Maes, 1995) proposes the mean squared difference of vectors as similarity metric and a variation of the Pearson correlation, the constrained Pearson correlation (takes positivity and negativity of the rating into account). The Spearman rank coefficient was proposed in (Herlocker et al., 1999) to

address the linear relationship assumption behind the Pearson coefficient. They show that a CF that uses the Spearman correlation performs similar to a CF that uses the Pearson correlation; they also suggest that the Spearman coefficient should be used for a small rating scale while the Pearson coefficient should be used on larger rating scales or continuous scales. The entropy was also proposed as a similarity metric in (Herlocker et al., 1999). In general, the cosine similarity and the Pearson coefficient similarity outperform other similarity metrics (Breese et al., 1998; Herlocker et al., 1999). Which one of these two yields the best performance seems to be dependent on the domain being used (Breese et al., 1998).

(Breese et al., 1998) proposes two tweaks that can be applied to any similarity metric. One, called “default voting” acts as a smoothing technique. Because the number of items that have rating from two users is relatively small, correlation coefficients can have large variance. The other, called “inverse user frequency”, puts less weight on popular items (items that are liked by many users). They show that both tweaks lead to improvement. Surprisingly, (Herlocker et al., 1999) reports that a Pearson coefficient modified to take into account the inverse user frequency, called the Pearson with variance weighting coefficient, performs worse than the regular Pearson coefficient. They claim that this might be due to the fact that the Pearson with variance weighting coefficient is not taking into account user disagreement with majority (which is frequent in movie ratings).

A question unaddressed in the item-centered memory-based description is how many similar users should be included in the neighborhood. (Herlocker et al., 1999) shows that including too many neighbors can decrease the quality of ratings in a CF RS. Correlation thresholding proposed by (Shardanand & Maes, 1995) uses an absolute threshold to decide which users are used in the similarity neighborhood. They show that there seems to be a tradeoff between the coverage (number of items for which the system can make prediction) and the accuracy based on the specific threshold. (Herlocker et al., 1999) shows that an n-best approach performs better in terms of both coverage and accuracy.

With regard to how to combine the ratings of the users from the neighborhood, the significance weighted sum with user’s rating normalization seems to perform the best (Herlocker et al., 1999). Significance weighting is used to measure the confidence in the correlation between two users: the larger the number of items the two users rated in common, the higher the confidence. User normalization is used to account for users using different rating scale. Their combination function performs better than other sum-based functions that did not have similarity weighting, significance weighting or user rating normalization. In addition, (Breese et al., 1998) proposes a method called case amplification in which the weights used in the sum are modified so that weights close to one are emphasized and weights close to zero are penalized. Case amplification further improves a RS. Moreover, they show that the improvements obtained from inverse user frequency and case amplification are additive.

Model-based Collaborative Filtering

The computational and the space requirements for memory-based CF can be very large especially if the number of users or items is large (which is the case in common e-commerce applications). Model-based CF has been proposed to address this issue. The model learned by a model-based CF will require less memory space than the rating matrix. Moreover, while the time needed to train the model might be long (but this can be done offline), the time needed to produce a prediction is relatively small.

A number of model-based CF approaches have been proposed. (Breese et al., 1998) proposes using Bayesian Belief Networks (**BBN**) to model the probability that a user will make a certain rating given the ratings he has made so far. The BBN nodes are the items and its structure and parameters are learned from data. They also propose clustering users using a Naïve Bayes with hidden class approach. In their experiments, BBN and Pearson correlation similarity are the only robust performers across various data sets and experimental procedures. The cosine similarity and clustering seem to work well when limited information (in terms of rating) is available.

A number of model-based CF have been designed to address the data sparsity problem of CF. In many domains, the number of items for which two random users provide ratings is relatively small. This limits the number of similar users the CF can exploit, potentially affecting the quality of the similarity metric. Several methods have been proposed to address this issue. Clustering of both users and items is proposed in (Ungar & Foster, 1998). Their results are encouraging. They also claim that their model can be easily extended to handle a common case in e-commerce: different users using the same account. (Goldberg, Roeder, Gupta, & Perkins, 2001) apply Principal Component Analysis on a set of items that are rated by all users (gauge items). Their experiments indicate that a performance similar to standard CF methods can be achieved in this way. In addition, their algorithm is two orders of magnitudes faster than standard CF when making predictions.

(Billsus & Pazzani, 1998) shows how a CF problem can be cast as a machine learning problem. Once the rating prediction task has been reduced to a standard machine learning problem, standard machine learning algorithms can be applied easily. The data sparsity problem in this case becomes a feature selection/reduction problem and existing feature reduction algorithms can be applied. They show that by applying the Singular Value Decomposition based feature selection and a neural networks learning algorithm they improve performance over standard CF.

Demographic user-centered RS

While CF RS exploit similarities between users in terms of their ratings, demographic user-centered RS exploit the similarity between demographic attributes of users. (Krulwich, 1997) proposes using a set of 62 demographic clusters discovered from a large scale demographic database (PRIZM). Classifying a new user into a demographic cluster is based on asking demographic-related questions. This process can be very time consuming and tedious. Once users are assigned to a cluster, recommendation is based on information about other users in this cluster. As expected, the accuracy is relatively low. Nonetheless, they can be used to develop broad user profiles which, in turn, can be used in a CF RS or to improve an existing RS (Pazzani, 1999). Interestingly, in (Pazzani, 1999) the “demographic” information is the content of user’s web page. The Winnow algorithm is used to learn the characteristics of the web pages associated with users that like a particular item. While their demographic based RS performance is lower than the CF or content-based RS used in their study, using the demographic-based RS in an ensemble combination of RS increases the overall performance.

4. Hybrid RS

The CF approach and the content-based approach have many complementary properties. CF RS base their recommendation solely on the community of users without taking into account item properties. For this reason, CF RS can handle complex ratings that are based on a complex combination of item properties (e.g. the writing style, the entertaining quality of a movie) (Herlocker et al., 1999; Shardanand & Maes, 1995). In contrast, content-based RS exploit item properties ignoring other users’ opinions. Thus, complex ratings that do not stem directly from item properties are very difficult to handle in a content-based RS. If a new item is introduced in the data set, a CF RS can not make any prediction for this item until a certain number of users have rated the item (the *new item problem*). In contrast, content-based RS can extend rating information to new items based on similarities with previously rated items. However, the content-based approach can make predictions only in specific regions of the domain space: any item that is not “very” similar to the items rated by the user so far can not be recommended. In contrast, CF RS can easily handle these *serendipitous predictions* effortlessly (Herlocker et al., 1999) once sufficient number of similar users have rated the item.

Hybrid approaches combine the CF (user-centered in general) approach and the content-based (item-centered in general) approach in an attempt to take the best of the two worlds. (Sarwar et al., 1998) proposes the use of several content-based analysis *filterbots* as pseudo-users in a CF RS for newsgroups.

These filterbots are user independent and analyze the content of each item to produce a rating. The filterbots prediction output can be designed to simulate certain category of users (e.g. users that are not interested in advertisement) or certain user preferences (e.g. users that prefer newsgroup emails that have more new content than reply content). Their experiments with very simple filterbots are promising. As expected, the coverage increases whenever a filterbot is added to a standard CF system. With respect to rating accuracy, only the Spell Checker filterbot provided consistent improvement. A big advantage of using filterbots is that users biases for certain item attribute values (like the preference for a smaller price) that could be modeled only with utility-based or knowledge-based approaches, can be easily encoded in this way. Moreover, a CF can be jumpstarted with a set of filterbots. Similarly, (Cohen & Fan, 2000) show how a CF can be improved by mining preference association between music artists from the web pages (though they do not acknowledge the filterbot concept)

(Good et al., 1999) extends the filterbots approach by creating a set of filterbots for each user. This filterbots mine the content information from items already rated by the active user and pose as pseudo-users in a CF system. They show that a CF that combines a limited number of users (50) and 23 filterbots performs better than a CF system that combines the 23 filterbots which, in turn, is better than a standard CF that combines just the 50 users. Their results emphasis one of the CF engines strengths: throw in the available data and the CF engine will sort out which information is useful.

(Pazzani, 1999) combines mean vector item-centered approach with CF in the so called collaboration via content. In his proposal the active user and the users that have already rated the active item are first converted to a mean vector representation using the words from the item description. Next, the mean vectors are used as rating vectors and CF is applied. His collaboration via content approach performs better than standard content-based and CF approaches. Somewhat similar, (O'Sullivan, Smyth, Wilson, McDonald, & Smeaton, 2004) improve CF by using item-item similarities mined from the rating matrix using the Apriori association rules discovery algorithm. Their CF system that makes use of these similarities performs better than a standard CF system.

All hybrid approaches discussed so far try to improve CF by tackling the data sparsity problem. But other types of RS can be improved using hybrid models. (Burke, 2002) shows how implicit ratings (in forms of user actions) in a knowledge-based system can be used to improve the standard knowledge-based system. According to their results, users arrive faster to the restaurant of interest if previous users' actions are used when ranking the items to be displayed in the next interaction.

A number of techniques inspired from other domains have been used to combine RS. For example, latent variable models are known to produce good results for unsupervised topic detection. The same method is used in (Schein, Popescul, Ungar, & Pennock, 2002). First, they train a latent model that relates users and movies through a latent factor. This model is used to generate data for training a second latent model that relates users and movie actors. Their evaluation results are mixed but promising future directions are identified. Finally, inspired by the good results of ensemble techniques for machine learning, (Pazzani, 1999) combines five distinct approaches to RS using a simple ensemble method: each item is ranked based on the sum of ranks predicted by each of the individual systems. Their results indicate that the ensemble method improves the accuracy over the best performing individual RS.

5. Issues in Recommendation Systems

In the previous sections, we described various approaches to RS. In this section we address a number of issues that should be taken into consideration when designing and implementing a RS.

Cold start problems

The phrase cold start has been used in the RS community to identify the situations where nothing or almost nothing is known about either the user or the item. These situations occur whenever a new user

starts using the system (*the new user problem*) or whenever a new item is added to the RS database (*the new item problem*).

The new user problem is the most costly problem in terms of user's experience with the RS. Making bad recommendation can frustrate the user, ultimately leading to his lack of confidence in the system ability to make recommendations. There are two ways to learn more about a new user: letting the user navigate and interact with the domain space or asking the user about his interests.

Knowledge-based RS can be used to address the new user problem. The user will either asked to provide an item of interest (e.g. his favorite movie) or presented with an item of interest (e.g. a popular movie) or with a hierarchy of items. By observing the knowledge-based interaction, a RS can learn about user interests (Burke, 2002). If there is a strong connection between item attributes and user's interest for that item, utility-based approaches can be used. General user profiles (e.g. users prefer less expensive items when all other attributes are the same) and a candidate critique interaction can be used to guide the user to items of interest (Faltings et al., 2004; Linden et al., 1997). By using Bayesian Belief Networks to model the uncertainty about user's interest for item attributes, (Jameson et al., 1995) show how utility-based RS can learn about user interests through the interaction. Static questionnaires and external user resources (e.g. user's location (Krulwich, 1997), user's webpage (Pazzani, 1999)) can be used as demographic information to jumpstart a RS system.

In the CF setting, the only way to learn about user's interests is to ask the user to rate specific items. An important issue in this situation is what items to ask the user to rate. (Goldberg et al., 2001) proposes asking the user to rate a certain set of items (called the gauge set). Unfortunately their proposal works only in domains in which the user can rate items without being familiar with them beforehand (e.g. jokes recommendation). For other domains (e.g. movie recommendation), the system has to propose items for which the user is likely to have an opinion. (Rashid et al., 2002) investigates a number of techniques for selecting such items. They find that if the user effort needs to be minimized (minimize the number of items needed to arrive to a given number of ratings) then popularity and item-item similarity are the right choices. If the quality of future recommendations is of interest, then items should be selected based on their popularity or the combination of popularity and rating entropy. While not investigated in their study, a hybrid method that focuses first on minimizing the user effort and then on improving the prediction quality could yield even better results in terms of user satisfaction with the RS.

Content-based approaches handle well the new item problem as long as the new item is not very different from the other items rated by the user. Thus, content-based RS or hybrid RS that have a content-based component are a good choice for this problem. Among the hybrid based approaches, filterbots seem to be the right candidates for solving this problem (Sarwar et al., 1998). For example, whenever a new movie is added to a movie RS, general or personalized filterbots that use the information like movie genre, cast, release year or description can be used to recommend the movie to the right users.

Evaluation

RS evaluation is an important issue. It can help us measure the quality of the recommendations made by the RS or choose the right RS approach for our data or our domain. Whenever we are making an evaluation, choosing a good baseline is important to gauge the quality of a RS. In many studies, researchers have compared their systems with the non adaptive versions (Billsus & Pazzani, 2000; Bradley, Rafter, & Smyth, 2000) or with a very simple baseline (e.g. random rating, the average rating in the database (Shardanand & Maes, 1995)). If the system has a training phase, then cross-validations are usually used (Sarwar et al., 2001). An important aspect is that the cross-validation has to be done inside user's data: that is, for each user we divide his rating into training ratings and test ratings. (Breese et al., 1998) experiments with various training data sizes: using all users' ratings except one or using only 2, 5 or 10 ratings. Because a certain number of ratings are needed in order to make good predictions for a user, in many cases only part of the user population is used (Good et al., 1999). In general, the results show that as the number of ratings available for each user increase, the prediction quality also increases (Sarwar et

al., 1998). In certain domains that exhibit chronologic and dynamic aspects (e.g. news stories) cross validation might be appropriate (Billsus & Pazzani, 2000). If no training data is need, then a user study is usually performed. (Hirashima, Hachiya, Kashihara, & Toyoda, 1997) tests the quality of several context-sensitive link reordering techniques by asking experimental subjects to rate a list of links.

The metrics used to measure the quality of the recommendations depend on what is being presented to the user. If we want to predict the user's rating for an item (e.g. Netflix's "we think you will rate this movie x stars"), then the mean absolute error is usually used (Sarwar et al., 2001) but other metrics have also been proposed (e.g root mean square error, correlations, etc). (Shardanand & Maes, 1995) argue that some errors weigh more than others: it is important that the RS will recommend items that the user will like not those he will dislike or for which he will have a neutral opinion. If the user is presented with a list of items, then it is important is how many items in the list are actually of interest. Classic measures like precision, recall and F-measure are usually used. Other measures have also been proposed: hit rate (percentage of users with at least one good recommendation (O'Sullivan et al., 2004)), expected utility of a ranked list of recommendations (Breese et al., 1998), whether there was at least on good item in a list of three recommended items (Pazzani, 1999), the sum of ranks for items of interest (Hirashima et al., 1997), rate of purchase for promotional e-mails (Ungar & Foster, 1998).

Defining the predict task is also important. Based on the intuition that users usually rate a larger number of items they like than items they do not like, (Ungar & Foster, 1998) proposes three tasks that make distinction between rating an item and purchasing the item: Implicit Rating Prediction (user will rate the item or not), Rating Prediction (user will like the item versus user will not like the item or will not rate it) and Rating Imputation (knowing the user has rated the item, predict how it will be rated). Coverage is also an important aspect (Herlocker et al., 1999). It measures the percentage of items for which the system can make a prediction for a given user. Data sparsity in the CF settings is an important factor that leads to reduced coverage. ROC sensitivity has been proposed to measure the system ability to distinguish between good and bad predictions (Sarwar et al., 1998). While regular ROC measure the quality of predictions for the entire population, (Schein et al., 2002) proposes Customer ROC that measure the customer coverage (the overall quality of recommendation when we are forced to make the same number of predictions for each user).

Interaction with the user

(Brusilovsky, 2001) identifies a number a of user characteristics an Adaptive Hypermedia system should adapt to: context, goals, knowledge, background, preferences, interests (short term and long term), individual traits (cognitive factors, personality factors). In contrast, in RS these characteristics are combined into the notion of rating. There are few studies that attempt to separate them. The fact that users' interests change while interacting with the data has been highlighted in (Hirashima et al., 1997) which investigate various context dependent techniques for recommending links to other relevant articles in an electronic encyclopedia. (Billsus & Pazzani, 1998) acknowledge that user interests change over time (a concept they call concept drift) and build two separate models: a short term model that can adapt very fast to new interests and a long term model that adapts slower but identifies salient user preferences. The fact that in their experiments the hybrid model that combines the short term and the long term model works better than each individual model stresses the fact that we should look beyond ratings in RS.

Users are in general unwilling to provide ratings unless they see an immediate and direct effect of this process (Billsus & Pazzani, 2000). It is well known that users examine more items than they rate. For this reason, researchers have proposed using implicit feedback in RS. Specific user actions can indicate interest or lack of interest: a reply to a newsgroup article is an indicator of positive interest (Goldberg et al., 1992), user actions in a knowledge-based interaction can be interpreted as positive or negative feedback with respect to the item in focus (Burke, 2002). (Claypool, Le, Waseda, & Brown, 2001) show that implicit interest indicators like the time spent on a web page and the amount of scrolling correlate with explicit ratings. (O'Sullivan et al., 2004) shows that RS systems trained on implicit interest indicators

(like playing or recording a TV program) have good recall in the task of predicting TV program ratings. Limitations of the presentation media can also be used to elicit implicit interests. In (Billsus & Pazzani, 2000), the percentage of a news story that a user has requested using a PDA interface is used as an implicit indicator of user's interest in that story. In speech based interactions, (Jameson et al., 1995) uses Bayesian Networks to reason about which attribute of an item to be presented to the user next so that the amount of feedback is maximized. But in general, the implicit feedback information is very noisy thus inappropriate for certain RS techniques (e.g. Pearson correlation) (Wærn, 2004). (Billsus & Pazzani, 2000) show that the system that uses implicit user feedback performs worse than the one that uses explicit user feedback.

An important issue is addressed in (Wærn, 2004): user involvement in correcting profiles. This user action is warranted whenever the RS makes wrong recommendation. In these cases, users typically attempt to fix the system by biasing the ratings. (Wærn, 2004) shows that a system that allows the user to build his profile performs better than one that uses explicit ratings; as the training data increase the difference between the two systems fades. Surprisingly, users can not recognize a good profile by inspecting it nor can they modify a learned profile to improve it.

In utility-based RS, certain interaction principles should be followed. The user should be informed efficiently about the range of attributes (Linden et al., 1997), suboptimal solutions should not be presented (Faltings et al., 2004) and diversity should be used in selecting the list of items that will presented to the user for critiquing (McGinty & Smyth, 2003).

Other issues

Specific characteristics of the underlying domain of a RS place additional requirements on the type RS to be used. In a very dynamic domain where new items are added frequently (e.g. news stories), content based RS are good choice (Billsus & Pazzani, 2000). CF approaches can be used but only when combined with content-based approaches (Sarwar et al., 1998). There are some domains in which very similar items may exist (e.g. news stories). In these domains, recommending items that are too similar is not useful. For this reason, upper bounds on similarity metrics are used (Billsus & Pazzani, 2000). In other domains a large number of items can be produced by combining the attributes (e.g. vacations). Generating all possible items in these domains can be very expensive thus utility-based or knowledge-based interactive approaches are appropriate. The presentation media can limit the amount of information a RS can display. In these cases, a RS should display the information that maximizes user utility (Jameson et al., 1995).

In some domains additional information may exist between items (e.g. hyperlinks). (Mobasher, Dai, Luo, & Nakagawa, 2002) show how browsing logs can be exploited to generate aggregate browsing profiles via clustering. These aggregate profiles are used to predict pages of interest for new users. Unfortunately, the link structure is not exploited in their study. Combinations between IR approaches that exploit the link structure and RS have the potential for improvement in these domains.

The amount of time needed to make a prediction is critical for real-time RS. Memory-based approaches on large databases of users or items can be computationally expensive thus model-based approaches are usually preferred. (Goldberg et al., 2001) proposes a Principal Component Analysis based algorithm that reduces the computation time two orders of magnitude. Due to the large amount of training time needed, in general, model-based approaches exhibit a lag between the time when new ratings are available and the time when these ratings can be used in recommendation. Transferring part of the computational load on the client side has also been proposed (Billsus & Pazzani, 2000; Bradley et al., 2000). In addition, privacy and security can be improved by storing the user profile on the client side (Bradley et al., 2000).

References

- Billsus, D., & Pazzani, M. (2000). User Modeling for Adaptive News Access. *UMUAI, Special Issue on Deployed User Modeling Systems*.
- Billsus, D., & Pazzani, M. J. (1998). *Learning collaborative information filters*. Paper presented at the International Conference on Machine Learning (ICML).
- Bradley, K., Rafter, R., & Smyth, B. (2000). *Case-Based User Profiling for Content Personalization*. Paper presented at the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, (AH2000).
- Breese, J. S., Heckerman, D., & Kadie, C. (1998). *Empirical analysis of predictive algorithms for collaborative filtering*. Paper presented at the Conference on Uncertainty in Artificial Intelligence (UAI).
- Brusilovsky, P. (2001). Adaptive hypermedia. *UMUAI*.
- Burke, R. (2001). Knowledge based recommender systems. *Encyclopedia of Library and Information Science*, 69.
- Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *UMUAI*.
- Claypool, M., Le, P., Waseda, M., & Brown, D. (2001). *Implicit Interest Indicators*. Paper presented at the IUI.
- Cohen, W., & Fan, W. (2000). *Web-collaborative filtering: Recommending music by crawling the web*.
- Faltings, B., Pu, P., Torrens, M., & Viappiani, P. (2004). *Designing example-critiquing interaction*. Paper presented at the IUI.
- Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, 35 (12).
- Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). *Eigentaste: A Constant Time Collaborative Filtering Algorithm*. Paper presented at the Information Retrieval Conference.
- Good, N., Schafer, J., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., et al. (1999). *Combining collaborative filtering with personal agents for better recommendations*. Paper presented at the AAAI.
- Herlocker, J., Konstan, J., Borchers, A., & Riedl, J. (1999). *An algorithmic framework for performing collaborative filtering*. Paper presented at the SIGIR.
- Hirashima, T., Hachiya, K., Kashihara, A., & Toyoda, J. I. (1997). Information Filtering Using User's Context on Browsing in Hypertext. *UMUAI*.
- Jameson, A., Schafer, R., Simons, J., & Weis, T. (1995). *Adaptive provision of evaluation-oriented information: Tasks and techniques*. Paper presented at the IJCAI.
- Krulwich, B. (1997). Lifestyle Finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2).
- Linden, G., Hanks, S., & Lesh, N. (1997). *Interactive assessment of user preference models: The automated travel assistant*. Paper presented at the User Modeling (UM).
- McGinty, L., & Smyth, B. (2003). *On the Role of Diversity in Conversational Recommender Systems*. Paper presented at the International Conference on Case-Based Reasoning (ICCBR).
- Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. *Data Mining and Knowledge Discovery*.
- O'Sullivan, D., Smyth, B., Wilson, D. C., McDonald, K., & Smeaton, A. F. (2004). Improving the Quality of the Personalized Electronic Program Guide. *UMUAI*.
- Pazzani, M. (1999). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, 13(5-6).
- Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., et al. (2002). *Getting to know you: Learning new user preferences in recommender systems*. Paper presented at the IUI.
- Rotaru, M., & Litman, D. (2003). *Exceptionality and Natural Language Learning*. Paper presented at the Computational Natural Language Learning.
- Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms*. Paper presented at the World Wide Web Conference (WWW10).
- Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., & Riedl, J. (1998). *Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system*. Paper presented at the ACM Conference on Computer Supported Cooperative Work (CSCW).
- Schein, A., Popescul, A., Ungar, L., & Pennock, D. (2002). *Methods and metrics for cold-start recommendations*. Paper presented at the SIGIR.
- Shardanand, U., & Maes, P. (1995). *Social Information Filtering: Algorithms for Automating Word of Mouth*. Paper presented at the CHI.

Ungar, L. H., & Foster, D. P. (1998). *Clustering Methods For Collaborative Filtering*. Paper presented at the Workshop on Recommendation Systems.

Wærn, A. (2004). User Involvement in Automatic Filtering: An Experimental Study. *UMUAI*.