Recommender Systems

Edward Marriott

Department of Electrical and Computer Engineering
University of Auckland, Auckland, New Zealand
emar048@ec.auckland.ac.nz

ABSTRACT

This paper provides an overview of how recommender systems work and the improvements which have been made to make recommendations more accurate. The filtering methods used in traditional recommender systems such as content-based and collaborative filtering have numerous limitations which reduces the effectiveness of the systems they are used in. A number of recommendation techniques such as multi-criteria and review-based ratings, along with participation incentives and profile context help to provide the user with better recommendations. Current and proposed implementations of recommender systems which improve the filtering of information are reviewed.

INTRODUCTION

Recommender systems attempt to provide the user with personalized recommendations of a product which they may be interested in. The systems try to match a product with what it perceives to be the preferences of the user, obtained in a number of ways. In doing this, it attempts to reduce the information overload a user receives when trying to traverse a large product range. A recommender system obtains data by storing the ratings a user has submitted in the past. These ratings can then be used to recommend an item similar to items rated highly by the user. This method of recommendation is called content-based recommendation. Another method of recommendation is called collaboration-filtering recommendation. In this case, the system recommends an item based on ratings given by other users which it perceives to be similar to the current user. Most recommender systems can be found in use as part of Internet-based applications, but they are not restricted to this domain. This research report will detail various methods which have been developed to improve the effectiveness of recommender systems.

MOTIVATIONS

The amount of information available to people as a result of expanded communication systems is enormous. As of April 2008, online encyclopedia Wikipedia has over 2,300,000 articles in English. Such a large amount of information makes it difficult for a person to find material relevant or of interest to them. This problem is known as information overload. Recommender systems are commonly used as a solution to this problem, suggesting items based on information about the user. Be that as it may, recommender systems are still a rarity when compared to other types of community sites, such as forums or blogs. The need for effective recommender systems is growing (Frankowski et al., 2007).

CHARACTERISTICS OF RECOMMENDER SYSTEMS

Many of the recommender systems in use today have similar characteristics and methods for recommending items.

Content-Based Filtering

Content-based filtering is one of the simpler methods of providing users with recommendations. It works by analyzing the ratings that a user has given to items in the past. Items which were rated highly by the user will be compared to other items, and the recommender system will determine if these other items are similar to the one that was preferred by the user. The similar items will then be recommended to the user. Many of the current content-based systems are focused on recommending simple textual items, such as documents and web sites (Adomavicius & Tuzhilin, 2005). The content of these systems is typically expressed as keywords, with each keyword having a relevant importance and weighting value.

Content-based filtering methods are limited by a number of factors. The comparison of two objects to determine if they are similar is limited to the attributes given to each object. For example, to determine if two movie items are similar, the accuracy is limited by the number of attributes stored in the system for each movie (actor, director, genre, etc.). Another problem this provides is when two differing items have the same attribute set, meaning that the system will not be able to differentiate between the two items. Content-based methods also suffer from a problem known as overspecialization. This means that the system is restricted to recommending items that are strictly similar to what a user has rated in the past. This eliminates a high percentage of the item domain which may have been of interest to the user. Conversely, users may also receive recommendations which are too similar to items they have rated, such as a different article describing the same event. An additional drawback of the content-based method is that a new user can not receive accurate recommendations until they have rated a sufficient number of items.

Collaborative Filtering

Many of the recommender systems currently in use employ collaborative filtering methods (Adomavicius & Tuzhilin, 2005). In contrast to content-based filtering, collaborative filtering predicts the usefulness of an item for a particular user based on the ratings given to the item by other users. The ratings of one user are compared to the ratings of another user to establish whether or not they have the same 'tastes' and can be considered similar. If they are found to be similar, an item rated highly by one of the users can be recommended to the other user if they have not rated that item.

Just like the content-based method, collaborative filtering has its limitations. It suffers from the new user problem (also known as the cold-start problem), meaning the system must have enough ratings from a user to enable it to learn their preferences. This problem also applies to new items being added to the system, where the item must be rated by a considerable amount of users before it can be recommended. Another limitation faced by collaborative filtering methods is that of data sparsity. The number of ratings a system can utilize is usually very small when compared to the total number of items in the domain. Therefore, if a certain item only has a few ratings, it will not get recommended very often, even if it has been given particularly high ratings. In a similar way, if a user has especially unusual ratings it would mean that there are very few other users who would be considered similar, and as a result the aforementioned user would not receive accurate recommendations.

RECOMMENDATION TECHNIQUES

Traditional recommender systems are limited by a number of factors including how the ratings of users are structured, and how the information is obtained. This section will detail several techniques developed to improve the way information about an item is stored, and different ways of obtaining this information.

Multi-Criteria Ratings

Traditionally, recommender systems allow for a single rating for each item, usually given by the user in numeric form. A new approach is to use a multi-criteria system whereby each item has several criteria, each with its own rating (Adomavicius & Kwon, 2007). This allows greater insight into the opinions of the user: the overall rating they give to an item gives an impression of how much they liked it, and the individual criteria ratings give an impression of why they liked it. This allows the recommender system to make a more accurate assumption as to which users are similar to each other.

Review-Based Ratings

An extension of the multi-criteria ratings system is to have a recommender system which obtains its data from consumer reviews. At present, content-based and collaborative methods collect data by asking users to input their rating of a particular item in singular form, normally as a numeric rating. However, many users prefer to use a textual review to express their opinion on an item, and these are usually written to a discussion board or forum. A recommender system that could extract this information for use in its recommendations could provide more accurate recommendations. The proposed system breaks down these consumer reviews into a structured form using translation ontology (Aciar, Zhang, Simoff & Debenham, 2006). This basically extracts keywords from the review onto a previously established representation of an object. From there, the system has access to these ratings as a set of criteria, and uses them in the same way as a multi-criteria system.

Participation Incentives

A recommender system will only work properly if users submit their own ratings of an item, and the more ratings the system can access, the more reliable the recommendations will be. It is common for users to ignore this process as it consumes their own time and there is no incentive for them to do so. The Jiminy architecture model attempts to provide participation incentives for users of recommender systems (Kostovinos, Zerfos, Piratla, Cameron & Agarwal, 2006). It makes use of an honesty metric to determine how honest a user is when they submit a rating on an item. This prevents users from submitting a large number of arbitrary ratings in order to receive the rewards. The reward model involves the user being credited for each rating they contribute to the system. If the user's honesty metric is below the required amount, they do not receive a reward for submitting a rating. In order to continue to receive rewards, their honesty metric must increase above the honesty threshold.

Physical and Social Context

Context is something that current recommender systems rarely take into account. The system does not account for things such as physical location or social networks when recommending an item to a user. The system will treat each user identically, and only base its recommendations on ratings. Incorporating physical and social context into recommendations will greatly increase the accuracy and appropriateness of a recommendation (Woerndl & Groh, 2007). An example given in Woerndl & Groh (2007) is that of a restaurant recommendation system being run on a mobile PDA. In this instance, since the user is in a mobile scenario, the restaurant's location would become more important because the user would be more inclined to choose a restaurant nearer to their current location. With regards to social context, it is important to differentiate between taste-related domains such as movies and music, and rational domains such as computers and technology. Influencing factors will be extremely different between both domains. In taste-related domains, social factors such as culture or religion may have a heavy bearing on what is considered an appropriate recommendation. Conversely, rational domains will lean towards logic and analytical comparisons in its recommendations. The implementation of physical context is fairly trivial, with mobile scenarios making use of GPS technology and internet applications using user profiles to obtain geographical information. Obtaining information to use in a social context is slightly more complicated. One option is to create or utilize existing social networks when analyzing user similarity. Other users which belong to the same social networks as the current user will be given more weight when determining if they are similar.

IMPROVED RECOMMENDER SYSTEMS

This section focuses on a number of implementations which rectify some of the disadvantages of traditional recommender systems

Trust-Based Recommender System

One research institute has proposed a Recommender System based on trust metrics, called the Trust-aware Recommender System (Massa & Avesani, 2007). The motivation for creating this system stems from the limitations experienced when working with Recommender Systems which are based on Collaborative Filtering. Due to the fact that most object domains are very large, and the number of items rated by a single user is generally small, it is unlikely that two users selected at random will have submitted a rating on the same object, meaning that their similarity can not be established. The Collaborative Filtering method also suffers from the fact that it can subject the Recommender System to malicious activity, whereby an attacker can create a profile which is considered similar to the target user and can influence their recommendations.

The proposed Recommender System aims to address these weaknesses by providing a way for the user to explicitly convey trust information. Users can determine which of the other users of the system they consider trustworthy, and which users they do not consider trustworthy. The context of the trust between users is related to how much they consider the ratings of the other user to be relevant to them. The trust information is organized into a trust network, enabling the system to link users through a trust metric, in order to establish a gauge of the trustworthiness of other users. In effect, instead of searching for 'similar' users like Collaborative Filtering does, the system searches for users it determines to be 'trustable' by the current user.

The trust metric is an algorithm which uses the trust network to determine the trustworthiness of users which have not been given a trust evaluation by the current user. In this way the metric reduces the social complexity of the system by predicting if the user would be considered trustworthy by the current user. An example of a trust metric in commercial systems is online auction marketplace eBay.com, which uses a trust metric with users being able to submit feedback ratings.

The Trust-aware Recommender System is implemented using two input matrices and one output matrix. The first input matrix is the ratings matrix which is of size n x m, where n = the number of users in the system, and m = the number of items in the domain. The ratings matrix represents all of the ratings given by the users to the items. The second input is the trust matrix, of size n x n. The trust matrix represents the trust information between all users. The output matrix is of size n x m, which represents all of the estimated ratings for each user, which is then utilized in providing the recommendations.

By being able to link users by trust propagation in the trust network, the system is able to generate more information about the trustworthiness of other users in the system. This reduces the problem of data sparsity, which is a weakness of the Collaborative Filtering method. In particular, this benefit is seen when a new user is added to the system. The user only needs to explicitly state its trust for one other user, and the system can then base its recommendations on the trusted user and other users that the trusted user also trusts. There are also fewer chances for attacks to be made on the Recommender System. Users who are attempting to exploit the system are not explicitly trusted by any of the users they are attempting to manipulate, and are therefore not considered for recommendations.

WikiLens

The GroupLens research group from the University of Minnesota has created an application which allows anyone to create a community maintained recommender system for any type of item. They have achieved this through the use of a 'wiki', which is a collection of web pages that can be accessed and modified by anyone. Following the success of sites such as Wikipedia and YouTube which allow user-contributed content, WikiLens permits users to contribute information needed to recommend items.

The challenge faced in developing this application was that many of the best known recommender systems were developed for large user bases, whereas many of the recommender systems being created through WikiLens would initially be for small communities (Frankowski et al., 2007). The research group solved this challenge by implementing 'small world recommenders', which helped to alleviate the lack of preference data.

Visual Interactive Recommendation

A publication by O'Donovan, Smyth, Gretarsson, Bostandjiev and Höllerer (2008) introduces a new type of recommender system called PeerChooser. PeerChooser seeks to address the black-box nature of most collaborative filtering applications by providing the user with a graphical explanation interface. Along with providing the user with a recommendation, the system also graphically explains how it arrived at that recommendation.

The system visualizes collaborative filtering by displaying a graph which is centred on the current user. The nodes connected to the user represent other users in the system, with the length of the connection between the two users representing the similarity of their ratings. The complete graph represents a neighbourhood of users, with the closer neighbours reflecting the current users taste more closely. The system also allows for a user to modify their neighbourhood to allow their recommendations to reflect their preferences at that time. Various icons representing features of a domain can be manipulated and moved closer or further from the user's node in the graph, indicating the user's current opinion of that feature. For example, in the domain of movies, various genre icons surround the user's node on the graph. If the user was to move a certain genre icon closer to their node, it would indicate that they currently prefer that genre to others.

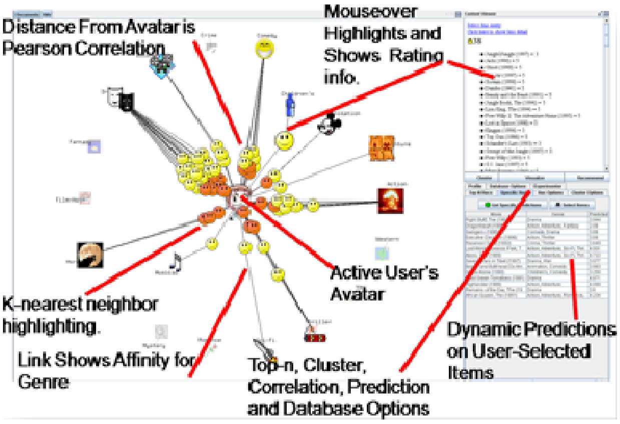

Figure 1: Screenshot of the PeerChooser interface.

MovieLens

MovieLens is a concrete implementation of a movie recommender system created by the GroupLens research group (Miller, Albert, Lam, Konstan & Riedl, 2003). The domain of movies is a good example of a domain that produces information overload. As of April 2008, the Internet Movie Database has over 370,000 movies in its database. The MovieLens system simplifies the process of finding movies of interest. The system uses a collaborative filtering model, comparing a user's ratings against similar users' ratings. New users are required to rate at least fifteen movies to begin receiving recommendations, helping to alleviate the cold-start problem. The GroupLens research group provides data sets extracted from MovieLens for other research groups to use as benchmark data for developing recommender systems.

Figure 2: Screenshot of the MovieLens recommendations.

CONCLUSIONS

The use of recommender systems provides an effective solution to the problem of information overload. With the ever increasing amount of information available, recommender systems help to provide people with material which reflects their tastes and preferences. The recommendation techniques mentioned all offer beneficial methods to increase the accuracy and usefulness of recommendations given to users by a recommender system. The inclusion of multi-criteria ratings allows broader scope of ratings, and allows recommender systems to make use of textual reviews. Participation incentives provide encouragement for users to submit reviews, also increasing the information the system can draw on when generating reviews. The utilization of physical and social context will allow for the generated recommendations to be more suitable to the user's situation. The reviews of several improved recommender systems show that the limitations of the traditional filtering models can be resolved, and can also provide users with more accurate recommendations.

REFERENCES

1. Adomavicius, G. & Kwon, Y. (2007). New Recommendation Techniques for Multicriteria Rating Systems. IEEE Intelligent Systems, 22, 48-55. Retrieved March 28, 2008, from IEEE Xplore database.

2. Adomavicius, G. & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6), 734-749. Retrieved March 28, 2008 from IEEE Xplore database.

3. Aciar, S., Zhang, D., Simoff, S. & Debenham, J. (2006). Recommender System Based on Consumer Product Reviews. International Conference on Web Intelligence, 2006, 719-723. Retrieved March 28, 2008 from ACM Digital Library database.

4. Frankowski, D., Lam, S., Sen, S., Harper, F., Yilek, S., Cassano, M. & Riedl, J. (2007). Recommenders Everywhere: The WikiLens Community-Maintained

Recommender System. International Symposium on Wikis, 2007, 47-60. Retrieved April 20, 2008 from ACM Digital Library database.

5. Kotsovinos, E., Zerfos, P., Piratla, N., Cameron, N. & Agarwal, S. (2006). Lecture Notes in Computer Science, 3986, 221-235. Retrieved March 28, 2008 from SpringerLink database.
6. Massa, P. & Avesani, P. (2007). Trust-aware Recommender Systems. ACM Conference On Recommender Systems, 2007, 17-24. Retrieved April 23, 2008 from ACM Digital Library database.
7. Miller, B., Albert, I., Lam, S., Konstan, J. & Riedl, J. (2003). MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System. International Conference on Intelligent User Interfaces, 2003, 263-266. Retrieved April 20, 2008 from ACM Digital Library database.
8. Woerndl, W. & Groh, G. (2007). International Conferences on Web Intelligence and Intelligent Agent Technology, 2007, 123-128. Retrieved March 28, 2008 from IEEE Xplore database.

Web Links:

1. http://ieeexplore.ieee.org.ezproxy.auckland.ac.nz/iel5/9670/4216968/04216980.pdf?tp=&arnumber=4216980&isnumber=4216968
2. http://ieeexplore.ieee.org.ezproxy.auckland.ac.nz/iel5/69/30743/01423975.pdf?tp=&arnumber=1423975&isnumber=30743
3. http://portal.acm.org.ezproxy.auckland.ac.nz/ft_gateway.cfm?id=1249069&type=pdf&coll=ACM&dl=ACM&CFID=22225292&CFTOKEN=13194581
4. http://portal.acm.org.ezproxy.auckland.ac.nz/ft_gateway.cfm?id=1296957&type=pdf&coll=ACM&dl=ACM&CFID=65458127&CFTOKEN=64808504
5. http://www.springerlink.com.ezproxy.auckland.ac.nz/content/c04w073625734677/
6. http://portal.acm.org.ezproxy.auckland.ac.nz/ft_gateway.cfm?id=1297235&type=pdf&coll=ACM&dl=ACM&CFID=65458127&CFTOKEN=64808504
7. http://portal.acm.org.ezproxy.auckland.ac.nz/ft_gateway.cfm?id=604094&type=pdf&coll=ACM&dl=ACM&CFID=65458127&CFTOKEN=64808504
8. http://ieeexplore.ieee.org.ezproxy.auckland.ac.nz/iel5/4427507/4427508/04427555.pdf?tp=&arnumber=4427555&isnumber=4427508