Recommender Systems

Matthew Gatland
University of Auckland
Mgat014@ec.auckland.ac.nz

ABSTRACT
This paper gives an overview of recommender systems and their uses in a variety of fields. Key problems are discussed and several examples are examined.

INTRODUCTION
Recommender Systems are systems that give useful suggestions to users.

Recommender systems are widely used in E-commerce. However, they have applications in any situation where a user is searching for something and cannot easily find or identify items which they want.

Recommender systems are commonly used to recommend products for purchase to a customer at an E-commerce website. There are two common systems for implementing this type of recommender system in E-commerce.

The first uses relationships between data items (usually products in E-commerce). When a user has purchased or otherwise expressed interest in a product, similar products can be suggested to the user. Product similarity can be calculated from a database of attributes stored about each product.

The second system instead uses a large database of users to compare users to each other. A user is presented with recommendations which other users similar to them have expressed an interest in.

Each system has its own problems which are discussed below.

Recommender systems can be sorted into a taxonomy with two axis, according to how a use interacts with them. The automation axis related to the amount of effort the user must expend in order to receive useful recommendations. The persistence axis relates to how much the recommendations are influenced by data collected about the user in the past rather than at the moment the recommendation is made.

BODY
Recommender systems are now crucial to E-commerce and widely used by large E-commerce Web sites.

A recommender system uses information previously obtained about a customer to recommend products that she is likely to find most valuable.

Modern companies need to provide customers with more customisation to address consumers' individual needs. As a consequence, the job of the consumer becomes more complex as there are more factors to consider when making a purchase.

Recommender systems can reduce information overload by presenting customers with a personalised product list that only shows products that meet certain criteria relevant to the customer.

Recommendations can be based on demographic information about the customer, data collected from the customer's previous purchases, or by simply displaying the store's most popular products.

Recommender systems provide an automatic and cost-effective way for an online store to be personalised for an individual.

The paper (Schafer, J. B., Konstan, J., & Riedl, J., 1999) identifies three ways in which recommender systems enhance E-commerce sales:

Converting browsers into buyers: Helps customers find products they wish to purchase, rather than browsing other products and not purchasing.

Cross-selling: Suggesting additional products to a customer making a purchase can increase the average order size.

Loyalty: Recommender systems become more accurate as more information about a customer is gathered. This provides an incentive for a customer to continue to frequent the same E-commerce Web site. Recommender systems can also be used to facilitate interaction between customers, creating a community which adds to customer retention.

One problem with collaborate filtering is that of free-riding. The problem is especially strong in collaborative filtering of bulletin board comments.

A user can avoid the unimportant comments by waiting until others have provided evaluations, allowing them to only read the best comments, and avoid the worst. It benefits any user to wait until others have read and rated comments before reading themselves. Thus, all users are discouraged from reading and rating comments, which weakens the rating system. (Avery, C., & Zeckhauser, R. 1997)

A Taxonomy for recommendation systems:
(Schafer, J. B., Konstan, J., & Riedl, J., 1999) describes a taxonomy to provide a user-focused analysis of recommender systems.

The taxonomy has two axes. The automation axis refers to how much explicit effort is required of the customer to receive recommendations. Whether the recommendations are generated automatically or not is not relevant to this measure, which only represents how the customer perceives the system.

The persistence axis describes whether recommendations are based on data collected over a long period of time, or information from a single session or Web page.

There are four recommendation techniques:

Non-personalized recommendations: These recommendations do not depend on information from the customer and are the same for all visitors. They simply use aggregated information from other customers. These are automatic and ephemeral (from the point of view of the person receiving the information). An example is feedback rating for sellers on eBay.

Attribute-based recommendations: Based on product attributes. These are manual, as the customer must specifically request a list of products with a certain attribute (for example, novels in the 'historical romance' genre.) These systems are ephemeral unless they remember the user's previous requests.

Item-to-item correlation: These are usually ephemeral, and recommend additional products based on a products already selected by the customer. These can be used to recommend products complimentary to those a customer has placed in her shopping basket. When used in this way, item-to-item correlation is an automatic system as it requires no special action on the user's part.

People-to-people correlation or collaborative filtering: Products are recommended to a customer based on the opinions or purchasing behavior of similar customers. Calculating similarity can be done in a number of ways. This system is persistent and fairly automatic, although it moves towards the manual end of the axis if users must explicitly rate products.

Example systems
(Schafer, J. B., Konstan, J., & Riedl, J., 1999) examples several commercial approaches.

Amazon.com:

Customers who bought (x) Like: On the information page for each book is a section of recommendations. This lists books frequently purchased by customers who also purchased the book being viewed, and lists authors whose books are frequently purchased by customers who also purchased books by the author of the book being viewed. This is an example of collaborative filtering. It is automatic and ephemeral on the taxonomy.

Amazon Eyes: Customers create a query for certain types of books based on subject, author, and other information, and receive email notification when products making their search are added to Amazon.com. This is an example of attribute-based recommendation, and is persistent and relatively manual in the taxonomy.

Book Matcher: Customers rate books on a five point scale and then may request recommendations, which they can rate to receive further recommendations. This is persistant and relatively manual in the taxonomy.

CDNOW:

Album Advisor: The user can select a single album, or up to three artists, and have the system recommend related albums.

My CDNOW: Customers indicate which albums they own (and like). When requested, the system will present 6 albums the customer might like based on their previous purchases. The customer can interact with the recommendation list to improve it, moving an album to their wish list, indicating that they do not want it, or indicating that they already own it.

eBay:

Feedback Profile: Buyers and sellers can contribute feedback about each transaction they have with another customer. Buyers may view the feedback of a seller before bidding on one of the seller's auctions, creating a simple recommender system. This system uses aggregation and does not tailor its results to a specific user.

Recommendation Interfaces and ways to make money:
Browsing: Users who are browsing for products that meet a certain criteria (for example "a comedy video from the 50s") can more quickly narrow their search and find the products that they are likely to want, making them more likely to buy.

Similar items: Recommending similar items exposes customers to items that they are likely to want but might have forgotten or not known about. This exposes customers to more of the product line and helps increase order size.

Ways of collecting user input
Main methods of user input (Schafer, J. B., Konstan, J., & Riedl, J., 1999) :

Purchase data: What the customer has previously purchased.

Likert: A manual evaluation, typically on a numbered scale.

Text: Written comments, not normally interpreted by a computer system, but which can be displayed to other customers in aggregate to give a group opinion.

Editor's Choice: Selections made by human editors.

Limitations of the different models

Collaborative filtering recommender systems only work once enough data is collected. This can make the system less useful for new users, slowing adoption.

One solution would be for a group of non-competing companies to share their customer information.

Another solution is used in the PTV system.

Example: Personal TV listings

(Smyth, B., & Cotter, P. 1999) explains that the next generation of TV systems will provide users with an unprecedented level of programme choice. The current techniques of programme selection via reading complete TV listings or "channel surfing" (inspecting available channels for desirable content) will not work sufficiently when content is so abundant.

"A 10 second per channel surf over even a modest 200 channel service will take about 35 minutes to complete!"

Personalized TV (PTV) listings are proposed as a solution to the problem of information overload which will otherwise make it difficult for a person to select TV programming that they want to watch.

PTV uses case-based reasoning and user-profiling to produce personalised viewing guides. PTV relies on an accurate database of user profiles, encoding a user's TV preferences. Factors include preferred viewing time, liked and disliked programmes, subject preferences, etc.

Information is collected at registration time. This is to overcome the [FIXME FIXME FIXME] slow start problem outlined in another paper. More information is collected as the user evaluates proposed recommendations.

The PTV system stores a database of programme cases describing programmes with associated information such as the title, genre, director's name and cast list.

A schedule database for all channels is generated automatically from schedules available elsewhere.

The PTV recommender combines case-based and collaborative recommendation strategies to create a list of recommended programmes. A user's custom TV schedule for a given day is generated by displaying occurrences of recommended programmes from the channel schedules for that day. This is displayed as an HTML page.

A user's programme preferences are stored simply as a two lists of programme titles, one of programmes the user liked and one of programmes the user did not like. Domain preferences are also stored. These are the user's available channels, preferred viewing times and some other general information.

Researchers have found that users will provide domain preferences at registration time but will not provide complete programme preferences. To obtain more complete programme preferences, the user's personalised TV schedule has interactive rating buttons next to each listed programme which allow the user to easily update their programme preferences. Over a long period of time, this feedback is also used to alter domain preferences, for example a preference for morning or prime-time programmes.

PTV uses case-based recommendation to recommend items that are similar to items on the user's positive list and dissimilar to those on the negative list.

Using case-based recommendation alone is not ideal. It is necessary to create profiles for all programmes and develop complex similarity models to compare them. Moreover, since new recommendations are always similar to previous preferences, there is reduced diversity in recommendations.

PTV uses collaborative filtering to recommend programmes to a user that other similar users have liked. PTV uses a lazy-approach where the recommendations for a target user are based on the preferences of a fixed number of similar users. Potential recommendations are weighed be the similarity of the users who prefer them to the target user.

Collaborative filtering does not require expensive content profiling and does not lead to the lack of diversity associated with case-based recommendation. However, it has different problems. Recommendations cannot be made until sufficient data is gathered about a user and about a programme. For TV listings, there is a need to recommend one-off programmes for which no user preferences may be known.

PTV uses a combination of both techniques to compliment each other. Case-based recommendations cover new and one-off programmes, while collaborative recommendations ensure diversity.

A study was performed on PTV users. In the study, each user was presented with three new recommendations each day and asked to rate them. The recommendations came from either collaborative filtering, case-based filtering, or random selection. 96% and 78% of users received at least one good new programme suggestion per day, depending on whether the guide was generated using collaborative filtering techniques or case-based methods. The paper suggests that this is a very positive result, especially given that the recommendations given each day are drawn from a limited pool of programmes so there is a finite number of good recommendations possible.

Example Relevance-feedback:

In a typical information system, a user must specify what information they want to retrieve. Unfortunately, people seeking information frequently cannot specify exactly what information they want. Moreover, when accessing an unfamiliar system, users may have no knowledge of the underlying database structure or the vocabulary used to describe items in the database.

As an example, a user searching for obituary information about well-known Americans in a system that relies on text content search may find no relevant results when searching for the word “obituary” because that word is never used in the text of an obituary.

In a situation like the one described above, it may be beneficial for the system to make a recommendation to the user, not on what information they are searching for, but about how they should perform their search.

John Rocchio suggested “relevance-feedback” as a form of query reformulation that can be done with no special knowledge of the system. The technique assumes that a user can determine whether objects returned by a search are relevant to their problem or not. By marking the relevant items, the user gives provides information that can be used to improve the query to produce more relevant results.

An alternative form of query reformulation is enabled by providing the user with their original query and list of new terms which might be useful to improve the query, and allowing them to manually reformat the query. These suggested terms can be terms commonly associated with the user’s query terms in previous searches, or associated with information items which are associated with the user’s query terms.

The latter technique can be thought of as “user-controlled” as the user has more direct control over how the query is reformulated. The former technique is “system-controlled” as the user only influences the query indirectly by selecting relevant information items.

Early results showed that relevance feedback worked well, but was improved when users had increased knowledge of how it worked and was given increased control by being presented with and able to alter the query terms that the system determined would help them.

However, users seemed to prefer the straight term-suggestion system because it did not require the additional effort of their selecting relevant results. Users preferred control over the primary task of query suggestion, but did not mind giving up control of the secondary suggestion mechanism in exchange for expending less effort.

Having an understanding of how the suggested terms are determined seems to be important to users. Terms that were not perceived by the user as being related to their search were distracting and made users uncomfortable. These conditions mean that users are only able to use the system efficiently when they have sufficient trust in it. When users have sufficient trust, they are comfortable giving up some degree of control.

Example: Recommender Systems for evaluation Computer Messages

One problem with collaborate filtering is that of free-riding. The problem is especially strong in collaborative filtering of bulletin board comments.

A user can avoid the unimportant comments by waiting until others have provided evaluations, allowing them to only read the best comments, and avoid the worst. It benefits any user to wait until others have read and rated comments before reading themselves. Thus, all users are discouraged from reading and rating comments, which weakens the rating system.

Possible solutions to the problem:

A subscription service, where readers pay to receive the evaluations of others who are paid for the service.

Transactions-based compensation, where some form of money or credit is given to users who provide early evaluations. Users who evaluate most gain a surplus, and users who evaluate least will have to pay. A problem with this system is its complexity.

Early evaluations can be assessed by how closely they conform to later evaluations, and payment can be increased for useful (predictive) evaluations.

Exclusion: Exclude readers who do not provide enough early evaluations. This is similar to informal social arrangements where reciprocity is expected for generous services. (For example, people host dinner parties with the expectation that their guests will later host dinner parties which they will be invited to.)

A problem with the exclusion system is that people who are not good at providing evaluations are forced to provide them anyway, causing inconvenience to them and inaccurate ratings for the group.

(Avery, C., & Zeckhauser, R. 1997) concludes that without some kind of market system to create incentives for evaluation, too few evaluations will be produced.

Future Work

Recommender systems are currently used as virtual salespeople (Schafer, J. B., Konstan, J., & Riedl, J., 1999) . However, they have potential to be used as marketing tools. One barrier to this is that people with marketing experience expect reports on groups of people, not individuals, while recommender systems are tailored to individuals. By aggregating similar individuals dynamically generated market segments, recommender systems may be useful in marketing campaigns.

Some possible developments raise ethical issues. For example, it would be possible for a recommender system to measure each customer's price sensitivity, then offer each customer prices designed to maximize their lifetime value to the company. Alternately, certain customers could be presented with special offers while others are not.

Opportunities for expansion:

Many sites collect implicit positive ratings by assuming that a customer who purchases a product likes that product. However, little has been done to collect implicit negative ratings. These ratings could come from products that a customer has returned. If products are listed on the website at multiple levels of detail, products that a customer chooses not to inspect at a greater level of detail before moving on from may also be given a mild negative rating.

When a customer is looking at a new product, a correlation-based recommender system could be used to provide a list of similar items that the customer already owns, explaining that "this product you're looking at is similar to these other products that you have liked in the past." This lets a customer quickly determine what type of product it is, and encourages them to purchase it.

No current recommender systems use all of the information available to them. Combining demographic information, purchase data, explicit ratings, ownership data in a meaningful way could yield better recommendations.

Recommender systems can be used to meet four of the five goals listed in Joe Pine's book Mass Customization.

"Customize services around standardized products and

services":

"Create customizable products and services":

"Provide point of delivery customization":

"Provide quick response throughout the value chain": - In future, recommender systems may be used to predict demand, allowing a faster response along the supply chain.

While automatic and ephemeral recommendations may seem ideal, Persistent and manual systems give unique benefits. Persistent systems create a relationship with the customer, inciting loyality. Manual systems, by requiring an investment from the customer, also create ties between the customer and Web site.

(Smyth, B., & Cotter, P. 1999, January) suggests that the ideal combination for an E-commerce Web site is a persistent and partially automatic system, requiring some input from customers but providing signification automatic benefits in return.

The PTV work suggests that collaborative filtering and case-based reasoning can be used together to overcome the weaknesses of each.

REFERENCES:

1. Avery, C., & Zeckhauser, R. (1997, March). Recommender systems for evaluating computer messages. *Communications of the ACM*, *40*, 88 - 89.
2. Belkin, N. J. (2000, August). Helping people find what they don't know. *Communications of the ACM*, *43*, 58 - 61.
3. Schafer, J. B., Konstan, J., & Riedl, J. (1999, November). Recommender systems in e-commerce. *Proceedings of the 1st ACM conference on Electronic commerce*, 158-166.
4. Smyth, B., & Cotter, P. (1999, January). Surfing the Digital Wave - Generating Personalised TV Listings using Collaborative, Case-Based Recommendation. *Proceedings of the third international conference on case-based reasoning, Munich, Germany*, 561-571.