The Design and Implementation of an Intelligent Online Recommender System

Rosario Sotomayor, Joe Carthy and John Dunnion

Intelligent Information Retrieval Group
Department of Computer Science
University College Dublin
Dublin, Ireland

Abstract

This paper describes the general design and architecture of an intelligent recommendation system aimed mainly at supporting a user in her navigation through the massive amounts of information that she has to cope with in order to find the right information. Alternative recommender system techniques are needed to retrieve quickly high quality recommendations even from a huge amount of data. Singular Value Decomposition-Collaborative Filtering (SVD-CF) methods are the techniques that are used in order to solve some recommender system problems by reducing the dimensionality of the product space, therefore producing better recommendations. Thanks to these techniques we can capture important latent associations between users and items. Also, users can benefit from the extension of their recommendation lists by taking into consideration the purchase of products that tend to be bought together.

Introduction

Recommender systems have been introduced to provide a solution to navigating the huge volume of information already available and growing at an explosive rate. The amount of information available in electronic form, such as news, movies, books, advertisements and other online information is overwhelming us. Recommender systems are computer-based techniques that can be utilized to efficiently provide personalized services in many e-business domains. In a domain where customers and suppliers interact, both groups reap the benefits. The customer benefits by receiving feedback from the system with some suggestions on items that she is likely to buy. At the same time, the business benefits with an increase in its sales.

One of the most promising recommender technologies is Collaborative Filtering (Resnick 1994) (Shardanand and Maes 1995). This technology works by building a database of items with users' opinions on them. Then a specific user is matched against this database in order to find her neighbors, those with whom she shares similar tastes. Collaborative Filtering has been used successfully by e-commerce sites and in the area of information filtering (Schafer, Konstan and Riedl 1999) (Sarwar et al. 2000a).

Despite the success of this technology there are still some challenges to overcome. The first issue is to improve the scalability of the collaborative filtering algorithms. Scalability problems occur during the search process in real-time as the process struggles with a high data load (Sarwar et al. 2001). The second problem that faces this technology is to improve the quality of the recommendation feedback. Inaccurate recommendations could potentially spoil a good relationship between an "e-tailer" company and its users.

A good example of the application of collaborative filtering technology is Amazon.com (Linden, Smith and York 2003), where it is used to help customers obtain a good recommendation for particular types of products. Accurate recommendation will win the loyalty of users, encouraging them to browse more and buy more. Amazon.com is perhaps the most representative example of the e-tailer phenomenon. However, this e-business company does not use all its available data to make its recommendations; in fact, it uses less than one percent of it (Sarwar et al. 2001).

As common recommender algorithms are not able to handle the shortcomings of recommender systems there is a need to look for other more capable algorithms. This has prompted us to investigate the use of dimensionality reduction technology (Berry, Dumais and O'Brian 1995) (Anton and Rorres 2000) in an attempt to improve the performance of recommender systems. Specifically, we have investigated a technology called Singular Value Decomposition (SVD) in order to reduce the dimensionality of recommender systems databases (Sarwar et al. 2000b) (Vozalis 2003).

This paper outlines the design, general architecture and function of an intelligent recommendation system aimed at supporting and facilitating a demanding customer in the task of searching for the right items to purchase

(Herlocker 2000) (Swearingen and Sinha 2002). It does so by bundling a set of products and creating a feedback list, thereby anticipating the wishes of the customer. The system allows the user to identify her own recommendation list and it personalizes the shopping experience by aggregating new items that were unknown to her. All these processes take cognizance of the issue of timeliness. Finally, this paper highlights how the technologies of collaborative filtering and singular value decomposition complement each other and how the advantages of each are combined in order to overcome some of the issues with other recommender systems.

An SVD-CF Combination System Approach

This paper presents a singular value decomposition - collaborative filtering recommendation system that addresses the following objectives:

Retrieval Time Reduction: The system reduces the time required for the feedback process, thus avoiding a bottleneck in the search process, which could cause user dissatisfaction and perhaps the loss of her loyalty towards the company. Our system, the Intelligent Online Recommender System (IORS), utilizes relational databases to generate timely feedback.

Search Shaping: The arduous process of searching for the preferred items is reduced by IORS. The system makes use of recommender technology and the purchase database to anticipate user wishes. With the help of dimensionality reduction we can reduce the noise generated by large quantities of data. The user is supported in the process of selecting a personalised item through the receipt of recommended feedback on items that they are likely to purchase.

Unveiling of New Preferences: Customers can take advantage of new relationships among users and products, allowing them to find new preferences that were previously unknown. Mainly, thanks to the dimensionality reduction techniques we can capture latent associations between items and users.

Extension of Recommendation Lists: Users can benefit from the extension of their recommendation lists by taking into consideration the purchase of products that tend to be bought together. This option allows us to anticipate user needs.

Interactive GUI feedback: IORS presents a visualization browser, which allows the user to filter the recommendations interactively from best to worst item. Additionally, this list can be filtered in different ways, such as by genre of item. Recommender systems should help a customer find and discover new, relevant and interesting items.

The IORS Process Model

The IORS system currently works with the movie recommendation domain, but can be easily tailored to other domains. This domain was chosen for two main reasons: Firstly, for the abundant availability and usage of movie recommenders online. Secondly, and more importantly, the movie domain has been the scenario of recognized research groups such as Movielens and Eachmovie, which may help us to compare and evaluate the efficacy of our results.

The IORS system modeling was realized from the perspective of the user's requirements. In order to design a successful e-tailer application we need to take in to consideration the users' needs and expectations.

Goals	Users	System
Unveil unknown items	Must be aware of new items	Highlight new items that might be a possible preferred item.
Trustful recommendation	Expect a useful feedback. Don't just overwhelm users with recommendations in order to sell something	Feedback must be presented in a way such that users can identify familiar items as well as new relevant items.
Transparent System	Must find the recommendation process logical.	Design of a clear and friendly system in order to make purchasing a pleasure.
Refined recommendations	Should be able to refine recommendation and exclude particular genres.	Provide a flexible search by using filters on different genres.
Evaluation of items	Need to express their opinion about their preferences.	Allow users to evaluate items by asking them to rate as many times as is needed.
Expansion of preferences	Must be allowed to explore their tastes and expand the horizons of their preferences.	Use of audio and video in order to let users experience and discover new items.
Purchase education	Want to know more about their items.	Provide information about items as well as ratings, pictures.
Appeal of shopping experience	Purchase experience must be enjoyable and entertaining	Design of an engaging process. Mix of questions and feedback.

Table 1: System Design Requirements

We have elaborated a structured table (see Table 1) to outline the problems or goals with respect to the expectations of the users (Swearingen and Sinha 2002). In other words, this table forms the requirements of the system and will be a guideline for the design of the application.

Following on from this, we present our main model for recommendation. Figure 1 shows the basic model as a UML class diagram. The most important class in this model is *MovieRecomSVD-CF*. As the name implies, this represents the recommendation process that will be given to a determinate user. This process is assisted by the use of SVD-CF techniques. This class generates personalized lists.

OfferList is the class that generates the final recommendation list that will be recommended to a determinate customer.

The *Profile* class is defined as an aggregation of the *User* class. In the case of the *Profile* class we can see that users will have individual profiles that will differentiate them from other customers, for example the different tastes and preferences that might exist between a housewife and a schoolgirl.

As we can see from this model the association between the *User* and *Movie* class itself has properties. The *Selection* class represents the properties of that relationship. Therefore, the *Selection* class becomes an association class of the *Movie* and *User* classes. The relation between the *Selection* class and *MovieRecomSVD-CF* plays an important role in the similarity process.

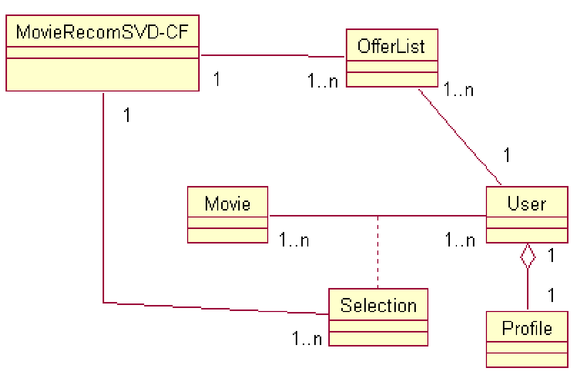

Figure 1: Recommendation main model.

We now concentrate on one of the most important aspects of the whole recommendation scenario - how the user is supported in the selection of products from a catalogue of items and how this process leads to the building and storage of a personalized list. Figure 2 gives a snapshot of the recommendation process that is resourced with refined data from a previous singular value decomposition technique.

Let us assume that the user wants to select a movie with some particular features. The user starts by entering some partial description of the movie (genre, favorite actors and so on). If the user is a registered user, the system will recognize her automatically because of the storage profile, and will know some description of herself, such as: age, occupation, tastes and so on. In the second step, the IORS system searches the catalogue of items and retrieves those items that satisfy those conditions. This search is based on the similarity technique of collaborative filtering with pre processing using singular value decomposition. Consequently, the previously recorded personalized data is used to rank these items. The movie that is most similar to those that similar users have expressed their opinion of in the past is ranked first. Finally, movies that the user expressed her opinion of, whether they were made in an explicit or implicit fashion, are recorded by the system for future reference.

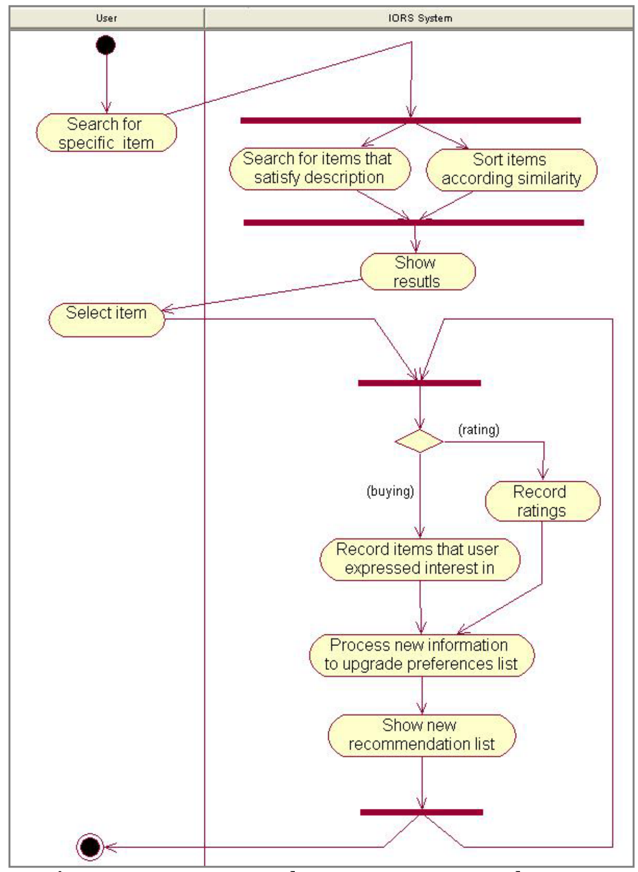

Figure 2: Recommendation main activity diagram.

The IORS Architecture

In this section, the general architecture of the IORS system is depicted as a layered architecture. More specifically, this section focuses on a recommender system constructed with the IORS recommendation framework. This framework serves as a structure within which recommendations can be made. It does so through a three-layer process, see Figure 3.

The *Presentation Layer* of the IORS system architecture serves as the connection between the *user* and the *Application Logic Layer*. It is the visible part of the user interface through which users can ask for assistance and view recommendations. From this layer, the system validates and translates the user requirements, transferring this information to the *Application Logic* module, and formats the returned feedback list to provide useful recommendations to the user. We decided upon using JavaServer Pages (JSP) technology as it provides a simplified, efficient way to create dynamic web content. Also, JSP technology enables the rapid development of web-based applications that are server and platform independent. This response returned in an acceptable time might make a difference to the retailers keeping the loyalty of their users (Linden, Smith and York 2003).

The *Application Logic Layer* of our system architecture is where the recommendation content from the *Data Layer* and the user's requirements are combined in order to obtain recommendations consisting of an ordered list of recommended items with their respective scores. This list of recommendations is returned to the *Presentation Layer* for display to the user. It is important to highlight that in order to integrate information contained in relational databases, our approach exploits SQL which interacts with Java. We decided upon PL/SQL as this extends SQL by adding constructs found in procedural languages, resulting in a structural language that is more powerful than stand alone SQL.

The *Data Layer* is where the IORS system acquires the required data, that is to say, a database of available customer preferences. The *Application Logic Layer* stores and upgrades data in this layer in response to input from the *Presentation Layer.*

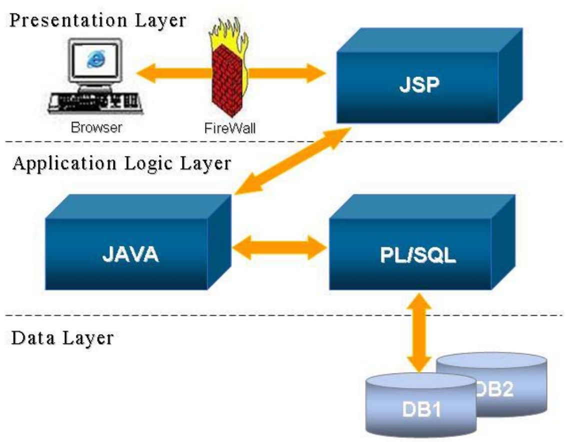

Figure 3: IORS Architecture diagram

An SVD-CF Approach in the Recommender System Domain

The remainder of this paper concentrates on providing a further explanation of the similarity process used by the IORS system. The particular techniques used are singular value decomposition and collaborative filtering.

Latent Semantic Indexing (LSI) is a statistical information retrieval method that has been widely used in information retrieval (IR) to solve the problems of synonymy and polysemy (Deerwester et al. 1990). LSI uses singular value decomposition as its underlying matrix factorization algorithm. The reduced orthogonal dimensions resulting from singular value decomposition are less noisy than the original data, capturing the latent association between the terms and documents (Berry, Dumais and O'Brian 1995).

Given a term-document-frequency matrix, two matrices of reduced dimensionality are constructed that represent latent attributes of terms, as reflected by their occurrence in documents, and of documents, as reflected by the terms that occurs within them.

Singular value decomposition is a well-known matrix factorization technique that factors an $m \times n$ matrix R into three matrices as follows:

$$R = U \,.\, S \,.\, V^T$$

Where U and V are orthogonal matrices of sizes $m \times r$ and $r \times n$, respectively; r is the rank of the matrix R. S is a diagonal matrix of size $r \times r$ having all singular values of matrix R as its diagonal entries. It is conceivable to reduce the $r \times r$ matrix S to have only the k largest diagonal values to obtain a matrix S_k , where $k<r$. (Gregorcic 2001) (Sarwar et al. 2000a).

All the entries of matrix S are positive and in decreasing order of magnitude. These matrices are very useful because we can obtain lower rank approximations of the original matrix R (Anton and Rorres 2000).

Following on from the above, we considered a scenario for our similarity model comprising of a set of customers $C=\{c_1,\ldots c_n\}$, a set of products $P=\{p_1,\ldots\ldots,p_m\}$ and a set of possible ratings I. Let us assume that we have data for customers and products (c,p), where $c \in C$ and $p \in P$. So, our approach is based on taking this scenario to compute our final resultant matrices: R_k, $U_k.S_k$, $V_k.S_k$ followed by the collaborative filtering process.

Conclusions

Collaborative Filtering as one of the most successful recommender system technologies, has become widely popular among e-tailers sites. In general though, recommender systems have become stretched by the huge volume of user information and are becoming even more stretched with the growth of the Internet domain across the globe.

Our research shows that singular value decomposition plays a key role in the recommendation process of our IORS system by addressing the gap left by collaborative filtering when it has to deal with the processing of a high quantity of data.

(Sarwar et al. 2000a) has shown that singular value decomposition can dramatically reduce the dimensions of the ratings matrix of a collaborative filtering system. Our approach uses this singular value decomposition as part of the representation step during an offline process in order to alleviate the high load of information to be processed by collaborative filtering. It also obtains the decomposition of the customer and product matrices that will allow us to find possible relations among the customers and products respectively.

It is important for the singular value decomposition method that the derived *k*-dimensional factor space does not reconstruct the original term space perfectly, since the original set is deemed to be unreliable (Deerwester et al. 1990). Further testing is required to understand the different results found when the *k* factor varies.

Finally, further work is required to exploit singular value decomposition for item selection in order to find possible hidden relations among items.

References

Anton, H. and Rorres, C. eds. 2000. *Elementary Linear Algebra*. Von Hoffman Press, Inc., United States of America.

Berry, M. W., Dumais, S.T., and O'Brian, G. W. 1995. Using Linear Algebra for Intelligent Information Retrieval. *SIAM* Review, 37(4): 573-595.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41(6): 391-407.

Gregorcic, G. 2001. The Singular Value Decomposition and the Pseudoinverse, Dept. Electrical Engineering, University College Cork.

Herlocker, J. L. 2000. Understanding and Improving Automated Collaborative Filtering. Ph.D., Faculty of the Graduate School, University of Minnesota.

Linden G., Smith B. and York J. 2003. Amazon.com Recommendations. Item-to-Item Collaborative Filtering. IEEE Internet Computing Press.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of the Computer Supported Collaborative Work Conference. Chapel Hill, NC.

Sarwar, B.M., Karypis, G., Konstan, J.A. and Riedl, J. 2000a. Application of Dimensionality Reduction in Recommender System – A Case Study. In ACM WebKDD 2000 Web Mining for E-commerce Workshop.

Sarwar, B.M., Karypis, G., Konstan, J.A. and Riedl, J. 2000b. Analysis of Recommendation Algorithms for E-Commerce. In *Proceedings of ACM EC'00*, 17-20. Minneapolis, Minnesota. ACM9 .

Sarwar B., Karypis G., Konstan J., and Riedl J. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. WWW10.

Schafer B., Konstan J., Riedl J. 1999. Recommender Systems in E-Commerce. In Proceedings of ACM Conference on Electronic Commerce.

Shardanand, U. and Maes, P. 1995. Social Information Filtering: Algorithms for Automating "Word of Mouth". In Proceedings of CHI., pp. 210-217. Denver, CO.

Swearingen, K., Sinha, R. 2002. Interaction Design for Recommender Systems. *DIS2002*. London

Vozalis, E., Margaritis, K., G. 2003. Analysis of Recommender Systems' Algorithms. Presented at *HERCMA*.