IEEE International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), August 01-02, 2014, Dr. Virendra Swarup Group of Institutions, Unnao, India

Web-based Personalized Hybrid Book Recommendation System

Salil Kanetkar B.E. Computer
Akshay Nayak B.E. Computer
Sridhar Swamy B.E. Computer
Gresha Bhatia Deputy H.O.D.
Department of Computer Engineering, VES Institute of Technology, Mumbai, India.

***Abstract*—Recommender Systems have been around for more than a decade now. Choosing what book to read next has always been a question for many. Even for students, deciding which textbook or reference book to read on a topic unknown to them is a big question. In this paper, we try to present a model for a web-based personalized hybrid book recommender system which exploits varied aspects of giving recommendations apart from the regular collaborative and content-based filtering approaches. Temporal aspects for the recommendations are incorporated. Also for users of different age, gender and country, personalized recommendations can be made on these demographic parameters. Scraping information from the web and using the information obtained from this process can be equally useful in making recommendations.**

***Index Terms*—Recommender system, Collaborative filtering, Content filtering, Demographic filtering, Time, Web scraping**

I. Introduction

THE job of a Recommender System (RS) is to deliver suitable recommendations to a user based on his likings, interests and preferences. A RS makes these predictions by using certain filtering techniques. These techniques include Collaborative filtering, Content-based filtering and Knowledge-based filtering. These methods help a user to discover relevant information in the complex mesh of the World Wide Web. Collaborative filtering involves filtering out users who have similar likings, and predicting new items for the user based on the filtered results. Collaborative filtering is of two types: Model-based and Memory-based. Model-based filtering, also called offline recommendation engine, involves the creation of a model using different techniques and algorithms to make recommendations. Memory-based, on the other hand, uses a rating matrix to make predictions and is often used for large datasets. Content-based filtering uses the information, description and keywords about the items and user profiles to make recommendations for the users. Knowledge-based recommender engines use different constraints and cases for making recommendations based on explicitly defined rules and similarity measures respectively. A recommender platform with the incorporation of all these techniques is also possible. It is called a Hybrid Recommender System [1].

Different techniques have been developed over time to give accurate recommendations. Apart from the regular filtering techniques, other approaches are being adopted. Ontology-based recommendations, Demographic-based recommendations have gained importance in recent times. Natural Language Processing is also being incorporated nowadays to analyze user feedback. Context aware recommendations too are gaining popularity. Even the temporal aspect is of equal importance in making accurate recommendations plausible with the current time. Scraping information from the web and using the traditional filtering techniques is becoming highly popular [3].

The system we propose takes into account the personalization of the recommendations. A user resorts to any website with a sense of trust. If recommendations given vary too much from the user's likes and tastes, he/she may simply stop using the system. Hence in order to build trust, recommendations needs to be personalized. Demographic recommendations are a good way of giving personalized recommendations. Filtering the results of a collaborative approach is a good way of making better recommendations. Recommendations suited to the user's age, region, gender can be made to make them more personalized. The cold start problem is a major issue in many recommendation systems. In such a scenario, the system is unable to give appropriate predictions until it has a better idea about the user's preferences. Demographic recommendations could help alleviate this problem to some extent, if not entirely in case of a newly added user [4].

A user always would like to stay abreast of the most popular books in a particular category. The traditional filtering techniques may not always be able to keep a user updated about the recent trends in books. Web scraping, could be of major help to users with such preferences. Information from e-commerce sites which have a dynamic list of popular and most purchased books, could be useful in giving recommendations.

Temporal aspects are of equal importance when it comes to recommendations for books. Especially for academically oriented books, old ratings and recommendations often become obsolete and lead to false predictions. Hence, a timestamp attribute for every rating the user gives is necessary. Even the recommendations can have a timestamp. Filtering out old recommendations helps make more accurate predictions. If a newer version of a particular book is available, then the user should be recommended the newer one rather than the older one. Also the users who had previously rated the old version, should be prompted to rate the new one too. Hence the ratings repository remains updated and obsolete recommendations are filtered out of the system over a period of time [2].

978-1-4799-6393-5/14/$31.00 ©2014 IEEE

II. Current Book Recommendation Systems

There are several recommendation websites already in existence for various domains. The methodology adopted for giving recommendations by these sites may vary but have still have a lot in common. Item-based collaborative filtering and User-based collaborative filtering are the two commonly adopted techniques. In Item-based recommendations, the similarity between items is taken into account and then predictions are made. Whereas in the latter, users with similar tastes are found and on the basis of their ratings, predictions are made. Different algorithms like Cosine Similarity Measure, Pearson Correlation Similarity Measure are used for the same [7]. Book recommendation websites have flourished over the web over the past decade. Huddersfield Book Recommender system, BookPsychic, WhatshouldIreadnext.com, LibraryThing, Goodsreads.com and Bookexplorer.com are some of the popular ones. Content-based recommendations are also provided by some of the mentioned systems [2].

III. Proposed Recommendation System

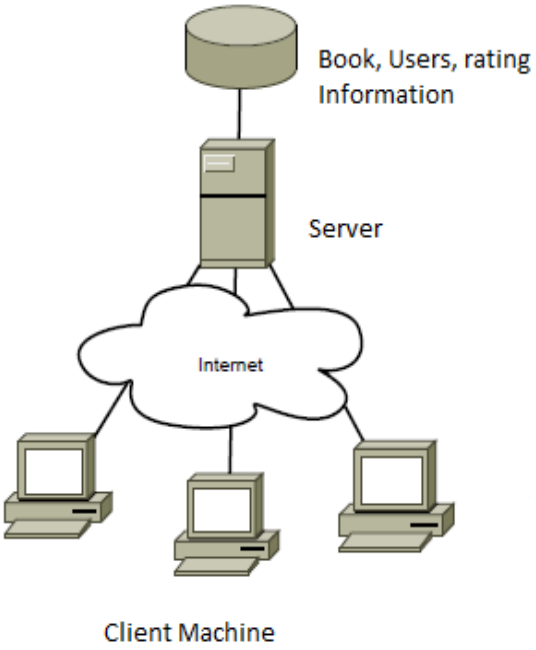

Fig. 1.Recommender System Overview

The figure above shows the basic overview of the web based recommender system. There would be a main server computer where the entire processing work would be done. The server would be connected to the main repository where the user and the demographic information would be stored. Another table would may consist of the entire database of books. This may include the title, images and other information of the book. A third table has the ratings given by the user to the books. An important field of this table is also the timestamp column. The temporal aspect of the ratings gets incorporated in this field. As mentioned before, we need to keep refining the recommendations at a regular interval as old ratings become obsolete. Hence with the help of timestamp field this would be possible. The web scraped data would be stored in another table of the database. Apache Mahout gives us the option of implementing the backend either using Apache Hadoop or a relational database like MySQL. Mahout attempts to provide open source implementations capable of solving problems at large scale using Hadoop. Hence scalability is not an issue when using Mahout. Accessing records from relational databases however can be much slower in comparison to in-memory representations due to the additional overhead of having to process data repeatedly.

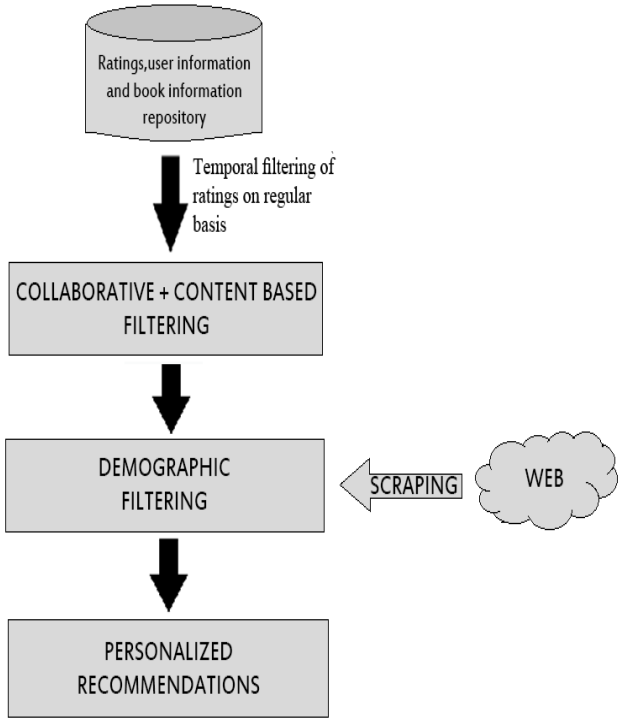

Fig. 2.Workflow of the Recommendation Process

The above figure explains the flow of the recommender system process. Let us consider a scenario where the users' ratings have been already stored in the repository. Now it's time to give him recommendations. Two major kinds of filtering techniques would be applied to give recommendations: Collaborative and Content. The collaborative filtering could use either item-item similarity or user-user similarity which is explained in detail further. A further filtering procedure based on the demographic aspects of the user like his age, gender etc. would help in making the recommendations more personalized. Apart from these recommendations, a user would have constant access to the popular set of books scraped from e-commerce websites.

IV. Filtering Techniques

A. Collaborative-based filtering

Collaborative filtering, as mentioned before, is mainly of two major types: Memory-based and Model-based. Memory based is further of two types: Item-Item Similarity and User-User Similarity. The similarity values between items are measured by observing all the users who have rated both the items**.** As shown in the diagram below, the

978-1-4799-6393-5/14/$31.00 ©2014 IEEE

similarity between two items is dependent upon the ratings given to the items by users who have rated both of them.

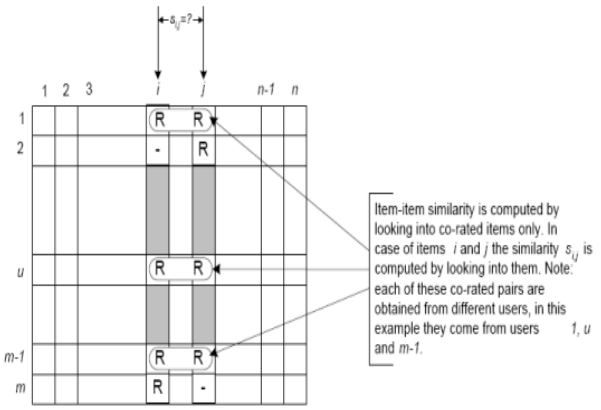

Fig. 3.Item-Item Similarity Technique Functioning

User-user similarity works in a different fashion. Suppose there is a user XYZ who has not seen an item A, then the algorithm finds those sets of users who liked same items as user XYZ in the past and also have rated the item A. This technique is carried out for all the items XYZ hasn't rated and the best-rated items are then recommended. Pearson correlation is a popular similarity measure used for this purpose. PearsonCorrealtionSimilarity and GenericUserBasedRecommender are two different classes in Apache Mahout which help in implementing this similarity measure [8]. The output is between the ranges -1 to 1, where a -1 means a total dissimilarity and a 1 means a total similarity. The mathematical formula for the same is:

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

a, b: Users
$r_{a,p}$: Rating of user a for item p
P: Set of items, rated both by a and b

B. *Content-based filtering*

Content-based recommendation in contrast to collaborative-based filtering, requires additional information about the items to give recommendations. This type of a recommender learns from the user preferences and suggests items that are similar to the likes of the user. In case of a book recommender, apart from the rating given by the users to the book, additional information like author, genre, publisher, cost, number of pages of the book, etc. may also be stored in the central repository [6]. A content-based recommender mainly learns from the different aspects. Hence, choosing a learning method is the most important step. Apart from the efficiency of the method, its time and space complexities also need to be considered, as these factors affect the overall performance of the system. The learning method needs to perform two tasks: Exploitation and Exploration. In the exploitation phase, the system tries to predict books based on the preferences already expressed by the user. In real world, the inclination of users towards items often varies and changes drastically. Hence the method adopted needs to explore newer options and try predicting a favorable item for the user.

Probabilistic method like that of Naïve Bayes Classifier, Linear Classifiers like Rocchio method and Support Vector Machines and also other methods like Term Frequency-Inverse Document Frequency, Nearest Neighbor and Relevance Feedback are used for content-based recommendations. Apache Mahout does not possess classes to implement content-based filtering as it has for collaborative-based techniques. However there are in-built classes for implementing Classification technique algorithms [5].

C. *Demographic-based filtering*

Multiple levels of filtering help give better recommendations. Demographic filtering combined with results of content and collaborative filtering, results in a better experience.

Clustering techniques can be highly advantageous in improving the efficiency of recommending algorithms [13]. Instead of taking the entire set of users registered with the website as a training set, we could cluster users based on their demographic aspects. Men and women usually tend to have their own respective choices of books which are often different from each other. We could create different clusters based on the gender. Age can be another attribute for clustering. Clusters can be based on different age groups. Kids, teenagers, adults can be accordingly then suggested books pertaining to their age groups. Classes like SequenceFile and different distance measure classes like EuclideanDistanceMeasure, ManhattanDistanceMeasure are used for implementing clustering in Mahout [5]. The figure below shows how different clusters look graphically after the implementation is performed on the target data.

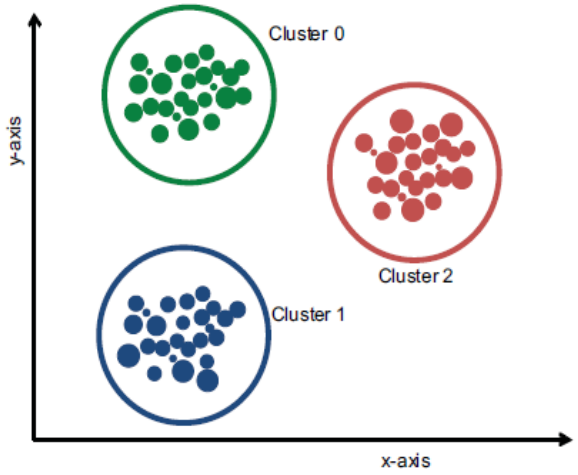

Fig. 4.Graphical Representation of Clusters

Location can also be a good parameter for filtering. This can be mainly done in two ways. Clustering of users according to their location and clustering of books according to the origin of the author. People often prefer native authors over foreign authors due to the ease of the language. Filtering can then be done only on the clusters rather than on the entire set. Thus the demographic parameters can be of great use in personalization of

978-1-4799-6393-5/14/$31.00 ©2014 IEEE

recommendations and reducing the time required for processing by use of clusters.

V. Web Scraping Technique

With the growth of the web to a large extent, the idea of using its resources to the aid of the recommending process has been gaining a boost over the past few years. Every possible information is available on the web, in many formats. The term Web Scraping means retrieving or getting data out of the web. It is a type of web content retrieval. The flowchart below explains the entire process of web scraping in brief [3].

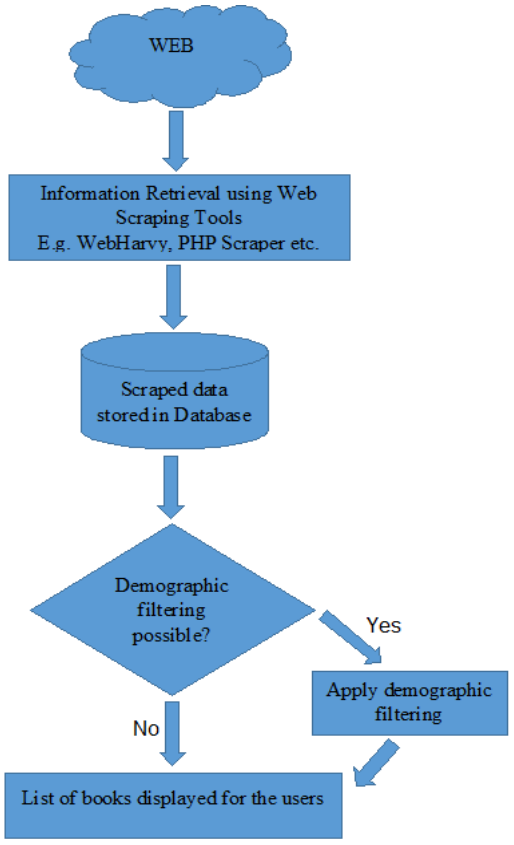

Fig. 5.Flowchart of Web Scraping Process

E-Commerce can be of great help in recommending books. The e-commerce websites reflect the popular and trending books in the market. Websites like amazon.com, flipkart.com provide access to a pool of all-time best sellers, best books of the month and similar such different categories [12]. Such lists can be scraped of the web and further optionally filtered as per the demographic aspects and displayed on the recommending site. The web scraping technique can help solve one of the major issues which the recommender systems face-the cold start problem. The cold start problem is faced by users when they are new to the system. Being new, the system has no or very less knowledge about the user and hence it is unable to recommend books at such a stage. The system needs to be patient and wait until the user has rated at least a few books. A partial solution to this problem can be web scraping. This technique is independent of the user ratings. Every user will have access to the scraped list of books as soon as he registers with the website. People often get allured to the trending books in the market. They will have a constant access to such books via the scraped data. If not nothing, the system is at least suggesting a few set of books which the user is likely to read as per the trends in the market. Later on as time progresses, the other filtering techniques come into picture as the system has enough knowledge about the user by then.

Web scraping can also play a role in suggesting books to *gray sheep* users. These are the set of users for whom the recommender system fails to give predictions due to their varied interests. Hence the web scarping technique helps the users stay in constant touch with the trending popular books in the market.

VI. Building Trust In The Recommender System

It is of utmost importance that the user should trust the recommendation system. Having confidence in the system would help the user get a healthy recommendation experience. The developers should prioritize this task. One way ensuring trust is explaining the process of recommendation to the user. On the website, it is better to have a webpage explaining how recommendations are made. A video of the same would have a larger impact. This would not only help the user understand the entire mechanism, but also develop a sense of trust with the system. Another way of building trust is via *Mirror Behavior*. To mirror someone is to have a profile similar to another user. In case of a profile match, we can allow the user to view the profile of another user with similar traits. However, care should be taken not to intrude into the other users' privacy. Recommender systems need to have a proper balance in this case. Mirror Behavior feature should be permitted only in the case that the user allows his profile to be shared with others, else not.

Security, is another important aspect of every recommender. A web based recommendation system is subject to attack by malevolent users who try to influence the behavior of the system by inserting fake user ratings [9]. Creating fake accounts and profiles is the simplest strategy adopted by these users. They could have two different goals-increase or decrease the rating of an item. Such attacks are called push and nuke attacks respectively [10]. A book recommender system is often subjected to two specialized from of attacks-bandwagon and segment attack. These are similar to the push and nuke attack and mainly attack the blockbuster items. Prevention of such attacks should be necessary, as the recommendations given to the user should not be influenced by the injection of dubious ratings.

One technique is to use hybrid algorithms for giving recommendations. A system solely based on collaborative filtering technique is more susceptible to the kind of attacks mentioned above. Hence, using a hybrid system or multiple layers of filtering techniques can help reduce the effect of these attacks. Our system which mainly deals with three filtering techniques-Collaborative, Content and Demographic would be better resistant to the attacks. Nuke and push attacks are based on profile injection methods. A

978-1-4799-6393-5/14/$31.00 ©2014 IEEE

method to make these attacks more difficult is to increase the profile injection cost. Use of *captchas* makes this possible [13]. Hence during every session, captchas could be used before making recommendations. IP Address could be yet another tool for preventing attacks of malevolent users. If too many registrations are happening from the same IP address, then any activity from that address can be simply blocked to prevent any further attacks.

VII. Conclusion And Future Scope

Apart from just the traditional Collaborative and Content-based filtering techniques, many modern techniques are being exploited nowadays. The hybrid algorithms are a mixture of many techniques. Demographic filtering helps give more personalized recommendations. With the advancement of the web, its use in the process of recommendation can help improve the efficiency. In this paper, we have explored the use of Web Scraping, which is a form of web content mining. The inculcation of Web into the process of recommendation, can help solve many limitations related to the filtering methods. The booming growth of Ontological aspects, semantic and context aware recommendations is sure to improve the quality of recommendations made. Recommendations in ubiquitous domains like that of mobile phones is enhancing the applicability of filtering. A combination of the models presented in the paper and many more can lead to a much greater personalized experience for the users and will also enhance the accuracy of the recommender systems over time.

Acknowledgment

We would like to take this opportunity to thank our mentor and project guide, Mrs Gresha Bhatia, Deputy Head of the Department of Computer Engineering at VESIT, for her continued support and guidance.

References

[1] Introduction to Recommender Systems Handbook, Francesco Ricci, Lior Rokach, Bracha Shapira, F. Ricci et al. (eds.), Recommender Systems Handbook.

[2] Chhavi Rana, Sanjay Kumar Jain, “Building a Book Recommender system using time based content filtering”, WSEAS Transactions on Computers, E-ISSN: 2224-2872, Issue 2, Volume 11, February 2012.

[3] Eloisa Vargiu, Mirko Urru, “Exploiting web scraping in a collaborative filtering based approach to web advertising”, Artificial Intelligence Research, 2013, Vol. 2, No. 1.

[4] Laila Safoury, Akram SalahExploiting, “User Demographic Attributes for Solving Cold-Start Problem in Recommender System”, LNSE.2013.V1.66, Lecture Notes on Software Engineering, Vol. 1, No. 3, August 2013.

[5] Sean Owen, Robin Anil, Ted Dunning, Ellen Friedman, “Mahout in Action”, 2012, Manning Publications, ISBN 9781935182689.

[6] Miguel Ram´ırez J´avega, “Content-based Music Recommender System”, 29th June 2005.

[7] Magnus Mortensen, “Design and Evaluation of a Recommender System”, INF-3981 Master's Thesis in Computer Science, February 5, 2007.

[8] Cataldo Musto, Apache Mahout – Tutorial (2014), 08/01/2014.

[9] Paul-Alexandru Chirita, Wolfgang Nejdl, Cristian Zamfir, “Preventing Shilling Attacks in Online Recommender Systems”, WIDM’05, November 5, 2005, Bremen, Germany.

[10] Bamshad Mobasher, Robin Burke, Runa Bhaumik, Chad Williams, “Towards Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness”, ACM Transactions on Internet Technology (TOIT), Volume 7, Issue 4, October 2007, Article No. 23

[11] Mehmet Kayaalp, Tansel Özyer, Sibel Tariyan Özyer, “A Collaborative and Content Based Event Recommendation System Integrated with Data Collection Scrapers and Services at a Social Networking Site”, 2009 International Conference on Advances in Social Network Analysis and Mining, ASONAM 2009, 20-22 July 2009, Athens, Greece.

[12] Andrei Toma, Radu Constantinescu, Floarea Nastase, “Recommendation system based on the clustering of frequent sets”, WSEAS Transactions on Information Science and Applications, ISSN: 1790-0832, Issue 5, Volume 6, May 2009.

[13] Neil Hurley, “Tutorial on Robustness of Recommender Systems”, RecSys 2011: Tutorial on Recommender Robustness, October 2011.

978-1-4799-6393-5/14/$31.00 ©2014 IEEE