International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Data Analytics and Recommendation Engine for E-Book Portal using Python and LAMP

Sandeep H[1], Chidanand H[2]

Department of CSE, RYMEC, Bellary, Karnataka, India

Abstract: ***There is a large growth in the internet and increasingly more number of websites are coming every day and trying to sell various things and services. It is very important for these web sites to analyze the data based on the user interactions. These data should help the management team to take the right decisions and it indicates whether their business is performing as per the goals. Apart from the analytics, as an extension to it, the recommendation engine plays a major role for each website to increase the sales. There are various techniques for generating recommendations such as item based collaborative filtering, user based collaborative filtering, hybrid collaborative filtering, etc.***

Keywords: Recommender Systems, Collaborative Filtering, Smoothing, Similarity Fusion, Amazon web services(AWS)

1. Introduction

Pustaka Digital Media Pvt. Ltd., is a Bangalore based company which publishes e-books in Indian vernacular languages. The website (http://www.pustaka.co.in) is already functional and 1000+ books with 100+ authors had been already published.

The current problem which the management facing is, the visibility of all the activities happening in the web site. Google Analytics tool provides information about the traffic but do not provide the information specific to the portal. The existing portal does not provide

Dashboard which shows the data analytics in a presentable format.
The existing portal does not display any recommendations to the end users and hence there is no motivation provided to the end user to buy more books. Also, when there are more number of books, the user does not have a clear idea on what can be purchased. In general, most of the retail web sites suggests the items based on the review done by the user, items purchased by the user and combining with the similar user's purchase patterns.

This project is intended to develop a dashboard for the management to view the data by analyzing the data and building a recommendation engine to generate the recommendations for the end users.

2. Literature Survey

Jun Wang, Arjen P. de Vries and Marcel J.T. Reinders, "Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion"

Memory-based methods for collaborative filtering predict new ratings by averaging (weighted) ratings between, respectively, pairs of similar users or items. In practice, a large number of ratings from similar users or similar items are not available, due to the sparsity inherent to rating data. Consequently, prediction quality can be poor. This paper reformulates the memory-based collaborative filtering problem in a generative probabilistic framework, treating individual user-item ratings as predictors of missing ratings. The final rating is estimated by fusing predictions from three sources: predictions based on ratings of the same item by other users, predictions based on different item ratings made by the same user, and, third, ratings predicted based on data from other but similar users rating other but similar items. Existing user-based and item-based approaches correspond to the two simple cases of our framework. The complete model is however more robust to data sparsity, because the different types of ratings are used in concert, while additional ratings from similar users towards similar items are employed as a background model to smooth the predictions. Experiments demonstrate that the proposed methods are indeed more robust against data sparsity and give better recommendations.

Adomavicius G., Tuzhilin A., Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Trans. on Knowledge and Data Engineering, 2005, 17(6): 734-749

Collaborative Filtering(CF) algorithms are widely used in a lot of recommender systems, however, the computational complexity of CF is high thus hinder their use in large scale systems. In this paper, we implement user-based CF algorithm on a cloud computing platform, namely Hadoop, to solve the scalability problem of CF. Experimental results show that a simple method that partition users into groups according to two basic principles, i.e., tidy arrangement of mapper number to overcome the initiation of mapper and partition task equally such that all processors finish task at the same time, can achieve linear speedup.

Y. Fan , Hebei Univ. of Eng., Handan and Y. Shen ; J. Mai. "Study of the Model of E-commerce Personalized Recommendation System Based on Data Mining", 978-0-7695-3258-5, pp. 647-651.

The integration of multiple recommendation algorithms using various data and the real-time requirement are pressing problems in the development of e-commerce

personalized service. This paper introduces recommendation methods and the timing and manners of recommendation result display, presents multiple recommendation algorithms that reflect the latest achievements in data mining research, designs a model of the e-commerce personalized recommendation system based on data mining. In the model, the rule type library and the recommendation method library are employed and the libraries are designed independently for the recommendation rule types, therecommendation algorithms, the recommendation moderules, the recommendation methods, effectively guaranteeing the real-time, efficient operation of multiple recommendation algorithms using various data, and the quality and efficiency of the personalized recommendation system.

Y. Ma, S. Ji ; Y. Liang ; J. Zhao ; Y. Cui, "A Hybrid Recommendation List Aggregation Algorithm for Group Recommendation", 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), pp. 405-408

In group-oriented recommendation filed, the design of a commonly acceptable recommendation list is a tough task. Traditional group recommendation algorithms often realize group recommendation list aggregation according to item ranking or item score of group members' recommendation lists. The factors considered in these algorithms are relatively one-sided. This paper puts forward a new HAaB aggregation algorithm for list aggregation, which considers the item ranking as well as the item score of the members' recommendation lists. Experimental results show that HAaB algorithm can obviously outperform the traditional group recommendation algorithms when recommending for various common combinations of groups.

Jing Wang, Sch. of Int. Programs, Neusoft Inst. of Inf. Technol., Foshan, China, Jian Yin. "Combining user-based and item-based collaborative filtering techniques to improve recommendation diversity". Biomedical Engineering and Informatics (BMEI), 2013 6th International Conference. 978-1-4799-2760-9. Page(s): 661 - 665
Print ISBN: 978-1-4799-2760-9
INSPEC Accession Number: 14117661
Conference Location: Hangzhou
DOI: 10.1109/BMEI.2013.6747022
Publisher: IEEE

Nowadays collaborative filtering technologies are widely used in many websites, while the majority research literatures focused on improving recommendation accuracy. However, it had been recognized that improving recommendation accuracy was not the only requirement for achieving user satisfaction. One important aspect of recommendation quality, recommendation diversity gained focus recently. It was important that recommending a diverse set of items for improving user satisfaction since it provided each user with a richer set of items to choose from and increased the chance of discovering potential interest. In this study, a synthetically collaborative filtering model was proposed, which combined the user-based and item-based collaborative filtering techniques. This model gave each user an option to adjust the diversity of their own recommendation list by using the prevalence rate and novelty rate parameters. Experiments using real-world rating datasets indicated the proposed model had effectively increased the recommendation diversity with little decrease in accuracy and surpassed the traditional collaborative filtering techniques.

3. Objective

This project is extending the existing functionality to support the analytics. In the existing Pustaka portal, the users can search books (based on language, author, genre, title) to buy or rent. Apart from the simple search, it lacks other functionality of Web 2.0 features. This project is to move the current portal to Web 2.0 which will enable the users to collaborate each other and interact with the portal. The new portal should be a social media platform for the readers.

In the new portal, the end user is not only a user of the application, but also a participant by social networking, tagging, social book marking, etc. There will be more features added in the project such as recommendations, followers, friends, ratings and reviews about the books and authors.

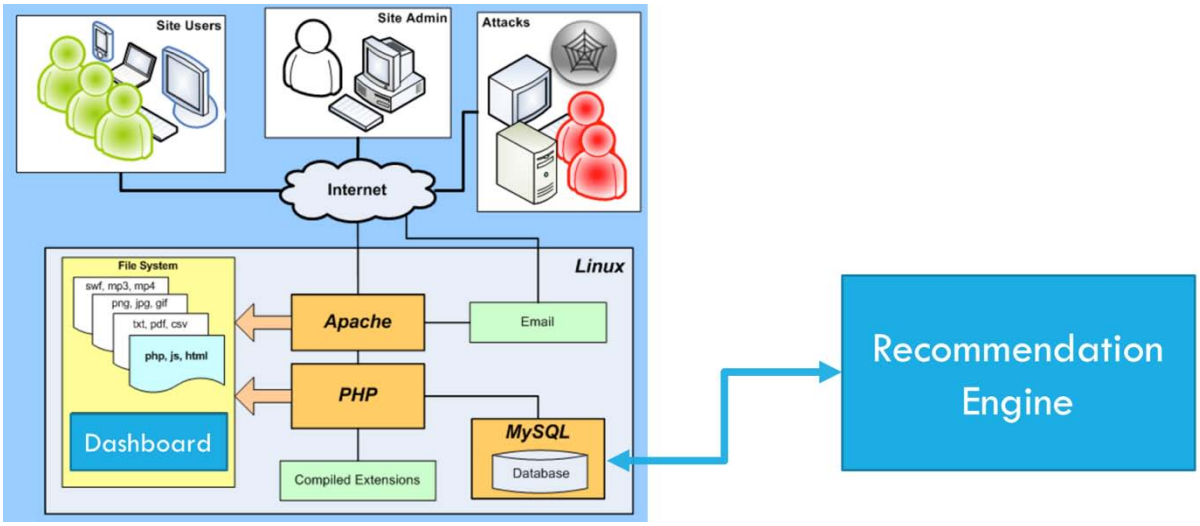
In order to support the new social web, the administrator and the management needs lot of analytics data to manage the portal better. The analytics engine will have the following features:

- Reading trends based on ratings, review comments
- Author trends
- Language trends
- Analysis of recommendations
- Generating recommendations for each reader

4. Possible Outcomes

- Implementation of the dashboard using LAMP technology which will display the content in the web browser.
- The dashboard will be accessible by only the management users which will be protected by user name and password.
- This will provide the administrator to view the trends happening for a specific book, author, etc.
- The trends for author and the language will be based on the sales happening in the system. These data shall be displayed using appropriate charts.
- For each user, the recommendation will be generated. These recommendations will be displayed to the user which will motivate them for purchasing the similar books.
- The recommendation engine will be implemented using hybrid based collaborative filtering by considering the user and the items purchased by the user.

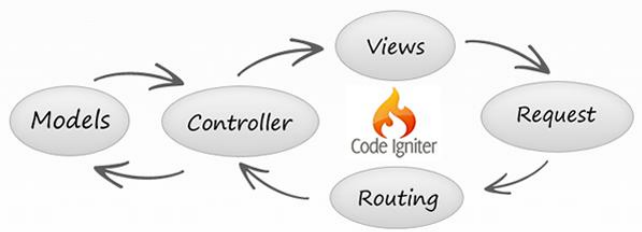
5. System Architecture



Scope of the project: Dashboard, Recommendation Engine

6. Methodology

The current system architecture of Pustaka web site is based on LAMP (Linux, Apache, MySQL and PHP) technologies. It also uses the code igniter framework which is based on MVC (Model, View and Controller) architecture. CodeIgniter is a popular PHP framework which is widely used in the industry which provides a set of libraries on top of PHP. These libraries provide the developers to develop the website in PHP quickly. The Dashboard is developed using CodeIgniter framework and the architecture of it is as shown below:



As part of the implementation, the following components were developed:

- Controller files which routes the request to the appropriate model and forwards the result to the view to get the display
- Models which will interface with the database to store and retrieve the information. The methods defined in the models will be invoked by the controllers based on the user requests.
- The view files which displays the user interface to the management team.

7. Recommendation Engine

Personalized recommendation systems can help people to find interesting things and they are widely used with the development of electronic commerce. Many recommendation systems employ the collaborative filtering technology, which has been proved to be one of the most successful techniques in recommender systems in recent years. With the gradual increase of customers and products in electronic commerce systems, the time consuming nearest neighbor collaborative filtering search of the target customer in the total customer space resulted in the failure of ensuring the real time requirement of recommender system. At the same time, it suffers from its poor quality when the number of the records in the user database increases. Sparsity of source data set is the major reason causing the poor quality. To solve the problems of scalability and sparsity in the collaborative filtering, this paper proposed a personalized recommendation approach joins the user clustering technology and item clustering technology. Users are clustered based on users' ratings on items, and each users cluster has a cluster center. Based on the similarity between target user and cluster centers, the nearest neighbors of target user can be found and smooth the prediction where necessary. Then, the proposed approach utilizes the item clustering collaborative filtering to produce the recommendations. The recommendation joining user clustering and item clustering collaborative filtering is more scalable and more accurate than the traditional one.

User Item Rating:

The task of the traditional collaborative filtering recommendation algorithm concerns the prediction of the target user's rating for the target item that the user has not given the rating, based on the users' ratings on observed items. And the user-item rating database is in the central. Each user is represented by item-rating pairs, and can be summarized in a user-item table, which contains the ratings Rij that have been provided by the ith user for the jth item, the table as following:

TABLE I USER-ITEM RATINGS TABLE

Item / User	Item1	Item2	… …	Itemn
User1	R11	R12	… …	R1n
User2	R21	R22	… …	R2n
… …	… …	… …	… …	… …
Userm	Rm1	Rm2	… …	Rmn

Measuring the Rating Similarity

Collaborative filtering approaches have been popular for both researchers and practitioners alike evidenced by the abundance of publications and actual implementation cases. Although there have been many algorithms, the basic common idea is to calculate similarity among users using some measure to recommend items based on the similarity. The collaborative filtering algorithms that use similarities among users are called user based collaborative filtering.

A set of similarity measures are presented and a metric of relevance between two vectors. When the values of these vectors are associated with a user's model then the similarity is called user based similarity, whereas when they are associated with an item's model then it is called item based similarity. The similarity measure can be effectively used to balance the ratings significance in a prediction algorithm and therefore to improve accuracy.

There are several similarity algorithms that have been used in the collaborative filtering recommendation algorithm: Pearson correlation, cosine vector similarity, adjusted cosine vector similarity, meansquared difference and Spearman correlation. Pearson's correlation, as following formula, measures the linear correlation between two vectors of ratings.

$$sim(i,j)=\frac{\sum_{c\in I_{ij}}(R_{i,c}-A_i)(R_{j,c}-A_j)}{\sqrt{\sum_{c\in I_{ij}}(R_{i,c}-A_i)^2\sum_{c\in I_{ij}}(R_{j,c}-A_j)^2}}$$

Where Ri,c is the rating of the item c by user i, Ai is the average rating of user i for all the co-rated items, and Iij is the items set both rating by user i and user j. The cosine measure, as following formula, looks at the angle between two vectors of ratings where a smaller angle is regarded as implying greater similarity.

$$sim(i,j)=\frac{\sum_{k=1}^{n}R_{ik}R_{jk}}{\sqrt{\sum_{k=1}^{n}R_{ik}^{\ 2}\sum_{k=1}^{n}R_{jk}^{\ 2}}}$$

Where Rik is the rating of the item k by user i and n is the number of items co-rated by both users. And if the rating is null, it can be set to zero. The adjusted cosine, as following formula, is used in some collaborative filtering methods for similarity among users where the difference in each user's use of the rating scale is taken into account.

$$sim(i,j)=\frac{\sum_{c\in I_{ij}}(R_{ic}-A_c)(R_{jc}-A_c)}{\sqrt{\sum_{c\in I_{ij}}(R_{ic}-A_c)^2*\sum_{c\in I_{ij}}(R_{jc}-A_c)^2}}$$

Where Ri,c is the rating of the item c by user i, Ac is the average rating of user i for all the co-rated items, and Ii,j is the items set both rating by user i and user j. Literature provides rich evidence on the successful performance of collaborative filtering methods. However, there are some shortcomings of the methods as well. Collaborative filtering methods are known to be vulnerable to data sparsity and to have cold-start problems. Data sparsity refers to the problem of insufficient data, or sparseness. Cold-start problems refer to the difficulty of recommending new items or recommending to new users where there are not sufficient ratings available for them.

User Clustering:

User clustering techniques work by identifying groups of users who appear to have similar ratings. Once the clusters are created, predictions for a target user can be made by averaging the opinions of the other users in that cluster. Some clustering techniques represent each user with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. Once the user clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller.

The idea is to divide the users of a collaborative filtering system using user clustering algorithm and use the divide as neighborhoods, as shown in the below figure. The clustering algorithm may generate fixed sized partitions, or based on some similarity threshold it may generate a requested number of partitions of varying size.

	Item-1	Item-2	...	Item-n
User-1	R11	R12		R1n
User-2	R21	R22		R2n
...				
User-m	Rm1	Rm2		Rmn

user clustering

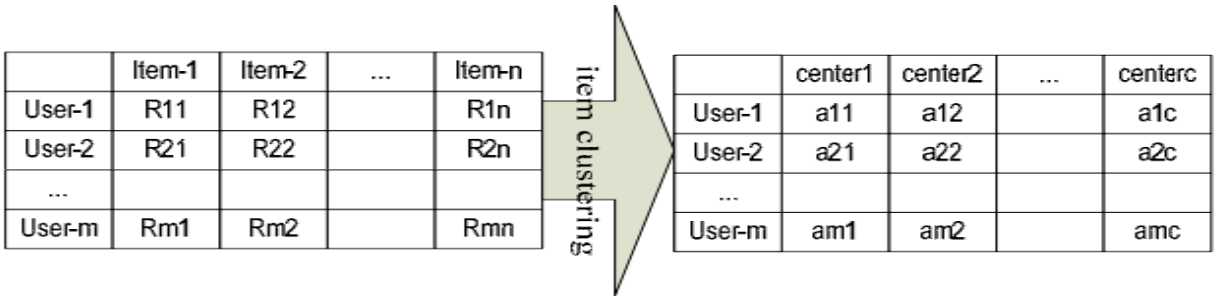
	Item-1	Item-2	...	Item-n
center1	a11	a12		a1n
center2	a21	a22		a2n
...				
centerc	ac1	ac2		acn

Item Clustering

Item clustering techniques work by identifying groups of items who appear to have similar ratings. Once the clusters are created, predictions for a target item can be made by averaging the opinions of the other items in that cluster. Some clustering techniques represent each item with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. Once the item clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller.

The idea is to divide the items of a collaborative filtering system using item clustering algorithm and use the divide as neighborhoods, as shown in the below figure. The clustering

algorithm may generate fixed sized partitions, or based on some similarity threshold it may generate arequested number of partitions of varying size.



Algorithm

There are many algorithms that can be used to create item clustering. In this paper, we choose the k means algorithm as the basic clustering algorithm. The number k is an input to the algorithm that specifies the desired number of clusters. Firstly, the algorithm takes the first k

Items as the centers of k unique clusters. Each of the remaining items is then compared to the closest center. In the following passes, the cluster centers are re-computed based on cluster centers formed in the previous pass and the cluster membership is re-evaluated.

Specific algorithm as follows:
Input: clustering number k, user-item rating matrix
Output: item-center matrix
Begin
Select user set U={U1, U2, …, Um};
Select item set I={I1, I2, …, In};
Choose the top k rating items as the clustering
CI={CI1, CI2, …, CIk};
The k clustering center is null as c={c1, c2, …, ck};
do
for each item Ii∈I
for each cluster center CIj∈CI
calculate the sim(Ii, CIj);
end for
sim(Ii, CIx)=max{sim(Ii, CI1), sim(Ii, CI2), …,
sim(Ii, CIk);
cx=cx∪Ii
end for
for each cluster ci∈c
for each user Ij∈I
CIi=average(ci, Ij);
end for
end for
while (CU and c is not change)
End

We use the pearson's correlation, as following formula, to measure the linear correlation between two vectors of ratings as the target item t and the remaining item r.

$$sim(t,r)=\frac{\sum_{i=1}^{m}(R_{it}-A_t)(R_{ir}-A_r)}{\sqrt{\sum_{i=1}^{m}(R_{it}-A_t)^2\sum_{i=1}^{m}(R_{ir}-A_r)^2}}$$

Producing Recommendations

Since we have got the membership of item, we can calculate the weighted average of neighbors' ratings, weighted by their similarity to the target item. The rating of the target user u to the target item t is as following:

$$P_{ut}=\frac{\sum_{i=1}^{c}R_{ui}\times sim(t,i)}{\sum_{i=1}^{c}sim(t,i)}$$

Data Set:

The data set to perform the above recommendation is based on the following date from Pustaka:

- User information
- Book information
- Rating provided by the users

At the moment, there are 5000 users, 1000 books in the system along with rating information.

System Requirements

Hardware Requirements

- System :pentium IV 2.4 GHz.
- hard Disk :40 GB.
- Floppy Drive: 1.44 Mb.
- Monitor: 15 VGA Colour.
- Mouse : Logitech.
- RAM : 512 Mb.

Software Requirements

- Operating system:Windows XP/7.
- Coding Language : PHP, CodeIgniter
- IDE : Notepad++
- Database : MYSQL

8. Conclusions

We proposed a novel algorithm to unify the user-based and item-based collaborative filtering approaches to overcome limitations specific to either of them. We showed that user-based and item-based approaches are only two special cases in our probabilistic fusion framework. Furthermore, by using a linear interpolation smoothing, other ratings by similar

users towards similar items can be treated as a background model to smooth the rating predictions. The experiments showed that our new fusion framework is effective in improving the prediction accuracy of collaborative filtering and dealing with the data sparsity problem. In the future, we plan to conduct better formal analyses of the fusion model and more complete comparisons with previous methods.

References

[1] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proc. of UAI, 1998.
[2] J. Canny. Collaborative filtering with privacy via factor analysis. In Proc. of SIGIR, 1999.
[3] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst., 22(1):143–177, 2004.
[4] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins.Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval Journal, 4(2):133–151, July 2001.
[5] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In Proc. of SIGIR, 1999.
[6] D. Hiemstra. Term-specific smoothing for the languagemodeling approach to information retrieval: the importance of a query term. In Proc. of SIGIR, pages 35–41, 2002.