International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 4 Issue 11, November-2015

Personalized Movie Recommendation System with User Interest in Social Network

Anil Rathod[1], Dr. Indiramm M[2]

[1]M. Tech. (CSE), Department of CSE, BMS College of Engineering, Bangalore, Karnataka, India
[2]Professor, Department of CSE, BMS College of Engineering, Bangalore, Karnataka, India

ABSTRACT

Recommender systems have grown extremely in day to day life, and it is very useful in a plenty of applications. Although the recommender system is very popular, there are some serious problems like cold start (unavailability of data for modeling algorithms) and sparsity (initial rating not available). This made people to step back in analyzing the functionality of those algorithms which leads to little decrease in creativity and optimizations in data mining algorithms and recommendation system. With the changed people and excessive growth of social network, people are accepting to share their personal interests via ratings, reviews, and likes on social networks like facebook, twitter etc. To recommend user interested movie and to solve the cold start and sparsity problem of input data, we develop Personalized Movie Recommendation System (PMRS). In PMRS model we used personal interest, similarity interest, and trust value to forms an optimized method of recommendation system.

Keywords: Personalized Movie Recommendation System (PMRS), Personal interest, Rating, Social Network

I. INTRODUCTION

Extracting hidden data has always been a fascinating science of study. It's not only fascinating but it also has miracle application that has changed the world who we were, who we are and who we will be. Recommendation system (RS) has been the source of income in Ecommerce, it is important to handling large inputs, such as recommending user preferred items and products. A perception from reviews has data that no less than 20 percent of the deals in Amazon originate from the work of the RS [1].

For example, it is observed that 90% of people have trust on book recommended by friends is good, and 75% of people have trust that the recommendation is useful from friends. People who like night life are likely to buy things related night life. These kinds of factors motivated to develop a recommendation system which includes factors like user interest, social influence, inter personal influence. The purpose of Recommendation Framework is to actually make things to be proposed automatically (Movie, Music, Books, etc.) according to their historical behavior and reduce their seeking time on the web.

A. PERSONALIZED MOVIE RECOMMENDATION SYSTEM

Movie recommendation system is broadly used application combined with online multimedia platforms which helps customers to get favored movies wisely from a large set of movie database. The most of existing recommendation systems is based on collaborative filtering (CF) mechanism [12] which has been successfully developed in the past couple of years. There are different well known online multimedia platforms (e.g., movielens.org, youtube.com, Netflix.com, anddouban.com) fused with CF strategy to recommend multimedia products (Movie, Music) to their customers. However, conventional recommendation system experience some limits: cold start, data sparsity and poor scalability problems [1, 7-8]. Various works have created to manage these issues and demonstrated the advantages on prediction accuracy in RS [4-5, 11].

In Personalized Recommendation System we utilize the ratings given by the user to the products to train a model which is then used to produce online recommendations. However, extracting similar interested users from large dimensional rating of data becomes very difficult and hence we calculate similarity interest of each user based on ratings given by other users. Personalized recommendation systems proved that it gives better result than Content-based filtering and Collaborative filtering both in terms of prediction accuracy and efficiency.

In this project, we use an optimized method for learning mechanism. To train learning model, we use training data set based on user-item ratings. The entire system comprises of two phases, one is an online phase, and other one is an offline phase. In offline phase, a learning model is trained to update user interest towards a particular movie by calculating user latent feature and items genre inclination by calculating item latent feature. In online phase, a TOP-N movies list is recommended to the users based on calculated ratings of movies. We use the genetic algorithm to quantify TOP-N movies in our new approach to achieve high quality recommendations. We also cross checked our new recommendation model with the help of Movieslens dataset to prove accuracy, we analyzed the results and proved that this new method is able of provide trustworthy movie recommendation comparing with present techniques.

II. LITERATURE SURVEY

Learning is a process of gaining knowledge by finding special characters from a set of actions. If we can impart that knowledge and learning to machines then we can help people to achieve better life style. The learned features will be again given as input to the learning process along with the data for better learning results.

A. LEARNING METHODOLOGIES

There are different type of learning like supervised, unsupervised and reinforcement.

1) Supervised

In this we'll be given with inputs and outputs. We have to create a function that will map inputs with outputs. We will try to reduce the gap between the hypothetical function and original function. In supervised learning we will have a teacher who will be supervising the task of mapping with inputs and outputs.

2) Unsupervised

In this we will be given with data where we will try to get some inputs and outputs. We have to create a function that will map inputs with outputs. We will try to reduce the gap between the hypothetical function and original function. In unsupervised learning we don't have a teacher who will be supervising the task of mapping with inputs and outputs. The system learns by itself.

3) Reinforcement

In reinforcement learning the system is taken to a dynamic environment and will be given a goal to perform. The system will have no inputs and outputs except the goal. The system will have least help. The system will be provided with logic to act in situations.

B. BASIC FUNCTIONING

Equations

In the following equation x represent input in training set data, h is hypothesis, and theta, theta1 represent learning parameters.

$$\mathbf{h(x) = \theta + \theta_1 x} \text{ .... } \mathbf{(2.1)}$$

Basic learning depends on how you choose theta and theta1. The idea is to choose those values such that hypothesis is close to output values for input x.

Let y be the training data output value for training data input x then we choose theta1 and theta2 such that for all m values of x, h(x)-y is minimum. This can be represented as follows.

$$\mathbf{\min_{\theta,\theta_1} \sum_{i=1}^{m} h(x^i - y^i)} \text{ .... } \mathbf{(2.2)}$$

The above function will be called as cost function. So cost function is the parameter which affects learning process.

Cost function will change based on the application we are running. One general cost function is squared error function.

$$\mathbf{\min_{\theta,\theta_1} {}^{1}\!/_{2m} \sum_{i=1}^{m} (h(x^i) - y^i)^2} \text{ .... } \mathbf{(2.3)}$$

C. LEARNING ALGORITHMS

1) Gradient Descent Algorithm

Gradient descent is industry standard for minimization. It is not only used for regression but many other algorithms use this gradient descent.

For gradient descent learning the learning process is done as follows

$$\mathbf{temp0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)} \text{ .... } \mathbf{(2.4)}$$

$$\mathbf{temp1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)} \text{ .... } \mathbf{(2.5)}$$

$$\mathbf{\theta_0 = temp0} \text{ .... } \mathbf{(2.6)}$$

$$\mathbf{\theta_1 = temp1} \text{ .... } \mathbf{(2.6)}$$

If we consider cost function as a curve then the gradient descent algorithm moves the point of inputs and outputs to a value where height of curve is zero and that's to say where the hypothesis fits training set.

The algorithm generally comes to stop when one of following satisfies

1. Cost function values is zero
2. Cost function reaches some minimum value
3. Number of iterations reaches a constant.

The stopping condition is application specific.

2) Matrix Factorization

Matrix factorization is one of the most successful methods for finding latent features. With matrix factorization we will calculate latent features for both users and items and the product of these latent features will give us the hypothetical rating with our hypothetical values for user and item latent features.

If U is considered as user latent features and I as item latent features then rating R can be given by

$$\mathbf{R_{ij}=U_iI_j^T} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \mathbf{(2.7)}$$

where i represent i^{th} user and j represent j^{th} item.

3) Clustering

Usually it includes centroid-based and hierarchical approaches to form a cluster. In centroid based method a cluster is form with first element as center and add other elements to that cluster and updates the centroid with the help of RMS (root mean square) formula to decide whether new cluster is formed or the element is added in the same cluster. The method used in hierarchical is the data to be organized into the groups with maximum commonality. The useful clustering algorithms are: k-Means, k-Medians.

4) Association Rule Learning Algorithms

In Association rule learning method the data is extracted based on rules framed to identify relationships between various data. With the aid of rules, we can identify very important feature form large multidimensional datasets. The best association rule learning algorithms is: Apriori.

D. Recommendation Algorithms

The most famous recommendation algorithms are collaborative and content-based filtering.

1) Collaborative filtering:

In this approach, construct a model from a user's past behavior (items purchased in the past and rated the items in past) along with similar choice made by different users, then use this model to expect items from the database which matches the interest of new user.

Collaborative filtering method does not depend on machine analyzable content and hence it is capable of recommending complex items without the knowledge of what the item is. This approach has three major drawbacks: cold start, scalability, and sparsity.

Cold Start: Huge amount of data is required to recommend accurately. In many applications we fail to get large database in initial stage of the business.

Scalability: To generate accurate recommendation we require to extract data from millions of transaction. This method fails to calculate the recommended items from millions of users and products.

Sparsity: Though the large number of items are sold, we have lest information of ratings and reviews given by the other users.

2) Content-based filtering:

This model uses a plenty of distinct character of an item so as to suggest other items which have similar functionality.

Usually, this model uses an item summary (i.e., a collection of distinct character and attributes) differentiating the item inside the method. The method constructs a content-based summary of users depends on a weighted vector of item features. The weights represent the significance of every feature to the user and it can be calculated using rated content vector. One of the most common method used is averaging values of the item rated vector and other difficult method uses machine learning mechanism such as cluster analysis, Bayesian Classifiers, artificial neural networks and decision trees in order to approximation that the user is going to like the items.

3) Clustering-based Collaborative Recommendation

This approach helps to overcome the problem of scalability and gives more accuracy in movie recommendation system. Several works have identified that clustering based collaborative filtering have more benefits than other recommendation system. The objective of this method is to segregate the items into the cluster that minimizes the distance between items among the same cluster to extract similar items. In this method k-nearest neighbor (k-NN) algorithm is used to construct an offline model. Usually, the different cluster is formed based similarity ratings given by the other users. With the help of different cluster we are able to find like-minded people who can buy the similar products. When a interested customer appeared online, this model suggests items based on weight assigned in offline mode and rating predicted by the user instead of searching in full user space.

III. ARCHITECTURE AND MODULES

A. Machine Learning Mechanism

In this learning process, we are predicting user ratings for a movie based on personal interest, user similarity interest, and trust values. We are calculating similarity interest, personal interest and trust values with help of movies data set used from MovieLense web site. The data set is divided into two parts that is training set and test set. Training set is used to train the algorithm, and find approximate value of trust set. After training, we apply hypothesis on test set to find accurate values of user rating. Once we satisfy with the result, then the result is updated into database.

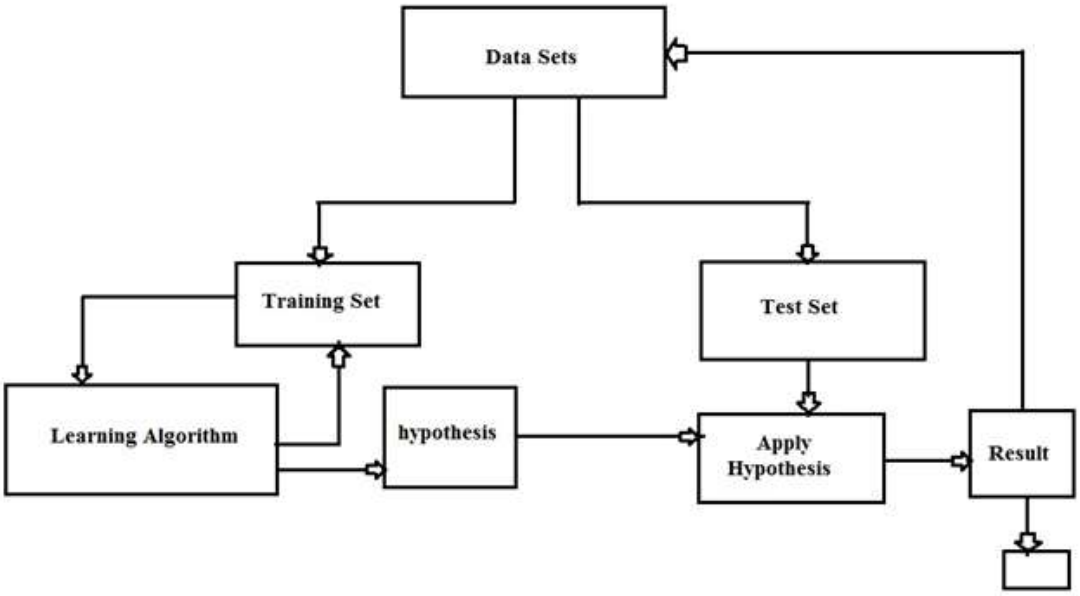

Figure 1. Learning architecture of PMRS

Here we work with multivariable learning (number of variable). For learning user habits we will start with users learning with matrix factorization methods. The learning algorithm used here is gradient decent for calculating user and item latent features. The learning process involves

1. Assume a hypothesis and identify learning parameters.
2. Calculate output values for given inputs with hypothesis
3. Calculate the cost function
4. Update hypothesis

1) Assuming hypothesis

In this step we frame an equation by assuming some constants as follows.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n n \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (4.1)$$

The above equation is the general for of an hypothesis and in that equation we have on the left side the notation for hypothesis and on the right side of assignment operator we have parameters separated by a addition operator and each term will have a constant which involve theta and an input variables with different powers.

The number of terms in the equation explains the complexity of algorithm and along with accuracy of prediction, if the number of terms is more, then the equation is better as more number of factors is considered for making equation.

2) Calculate output values for given inputs with hypothesis

Our next step is to find all the output values with our hypothesis equation by substituting the input values we have for x and calculating the value of hypothesis for all theta values and inputs. After this step we will have two collections of values for inputs and outputs.

3) Cost Function

Our idea of machine learning is to find a hypothesis that is close to original equation. So technically, we have to choose theta values such that the value of hypothesis(x) is close to y values for our training set examples. So what we are solving a minimization problem over theta values.

Therefore, the equation will be like

$$\min_{\theta_0\theta_1}(\mathbf{h}(\mathbf{x}) - \mathbf{y})^2 \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \;(4.2)$$

Since we are doing this for all training samples, the equation will become

$$\min_{\theta_0\theta_1} \quad \sum_{i=0}^{m}(\mathbf{h}(\mathbf{x}) - \mathbf{y})^2 \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.3)$$

4) Updating Hypothesis

In this part we are updating the values of theta so that the equation going to get a place where the values of

1. Cost function becomes 0 (Zero) or
2. Cost function doesn't changes or
3. We reach a specific cost function value (some value specified) or
4. We reach a specific number of iterations

B. Modules Implemented

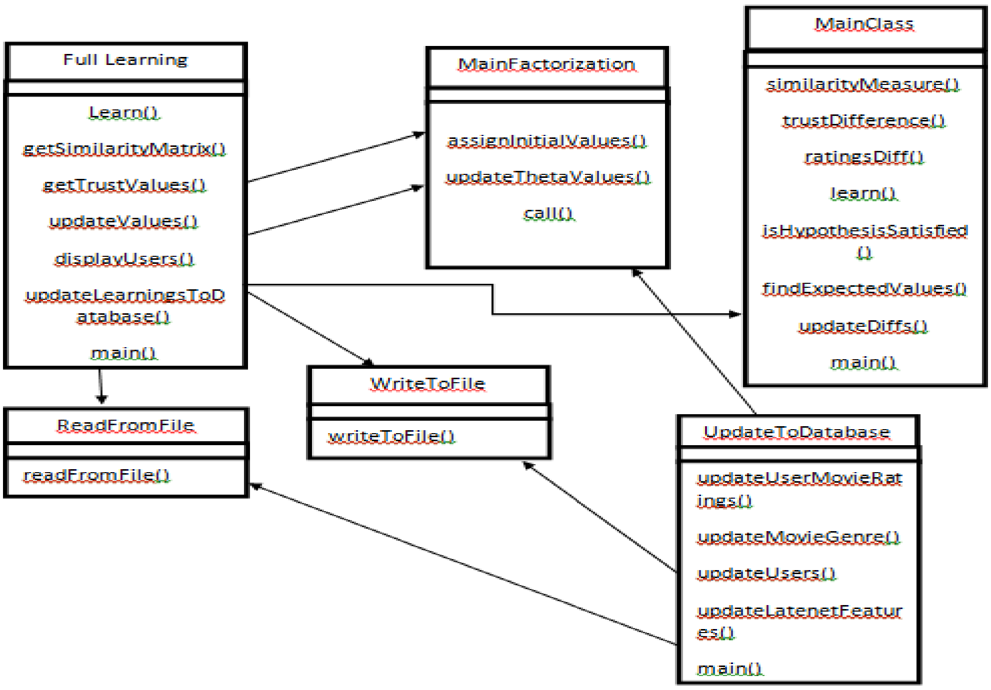

Figure 2. Modules implemented

IV. EXPERIMENTS AND RESULTS

We used the computer with Intel core I7 3.5GHz Processor, 4.0GB RAM to demonstrate all the experiment conducted.

A. DataSet

We used the dataset collected from Movieslens web site which is freely available online. It consists of 1000000 rating given by the users for 1682 different movies. It has two sets of data, one is training set and other one is test set. We used the training set to build machine learning mechanism and the test data is used to predict the interested movies based on different features. To verify the quality of recommended items we use the root mean square (RMS) method which is commonly used method to measure the accuracy.

B. Tables used

Table 1: Users database table details

Table Name: User		
Field Name	**Type**	**Description**
UserId	integer, primary key	Unique id of the user
Age	Integer	Age of the user
Gender	Boolean	Gender of the user
Occupation	character(30)	Occupation of the user
ZipCode	character varying(10)	Postal code
Adventure	double precision	Movie Genre
Animation	double precision	Movie Genre
Children	double precision	Movie Genre
Comedy	double precision	Movie Genre
Crime	double precision	Movie Genre
Documentry	double precision	Movie Genre
Drama	double precision	Movie Genre
Fantasy	double precision	Movie Genre
Musical	double precision	Movie Genre
Mystery	double precision	Movie Genre
Romance	double precision	Movie Genre
Sci_Fic	double precision	Movie Genre
Thriller	double precision	Movie Genre
War	double precision	Movie Genre
Western	double precision	Movie Genre

Table 2: Movie Genre database table details

Table Name: Movie Genre		
Field Name	**Type**	**Description**
MovieId	integer, primary key	Unique id of the user
Adventure	double precision	Movie Genre
Animation	double precision	Movie Genre
Children	double precision	Movie Genre
Comedy	double precision	Movie Genre
Crime	double precision	Movie Genre
Documentry	double precision	Movie Genre
Drama	double precision	Movie Genre
Fantasy	double precision	Movie Genre
Film_Noir	double precision	Movie Genre

Horror	double precision	Movie Genre
Musical	double precision	Movie Genre
Mystery	double precision	Movie Genre
Romance	double precision	Movie Genre
Sci_Fic	double precision	Movie Genre
Thriller	double precision	Movie Genre
War	double precision	Movie Genre
Western	double precision	Movie Genre
Name	Text	Name of the movie

Table 3: UserMovie Rating database table details

Table Name: Movie Genre		
Field Name	**Type**	**Description**
UserId	integer NOT NULL	User Id
MovieId	integer NOT NULL	Movie Id
Rating	integer NOT NULL	Rating given by the user
timestamp	Numeric	

C. Result

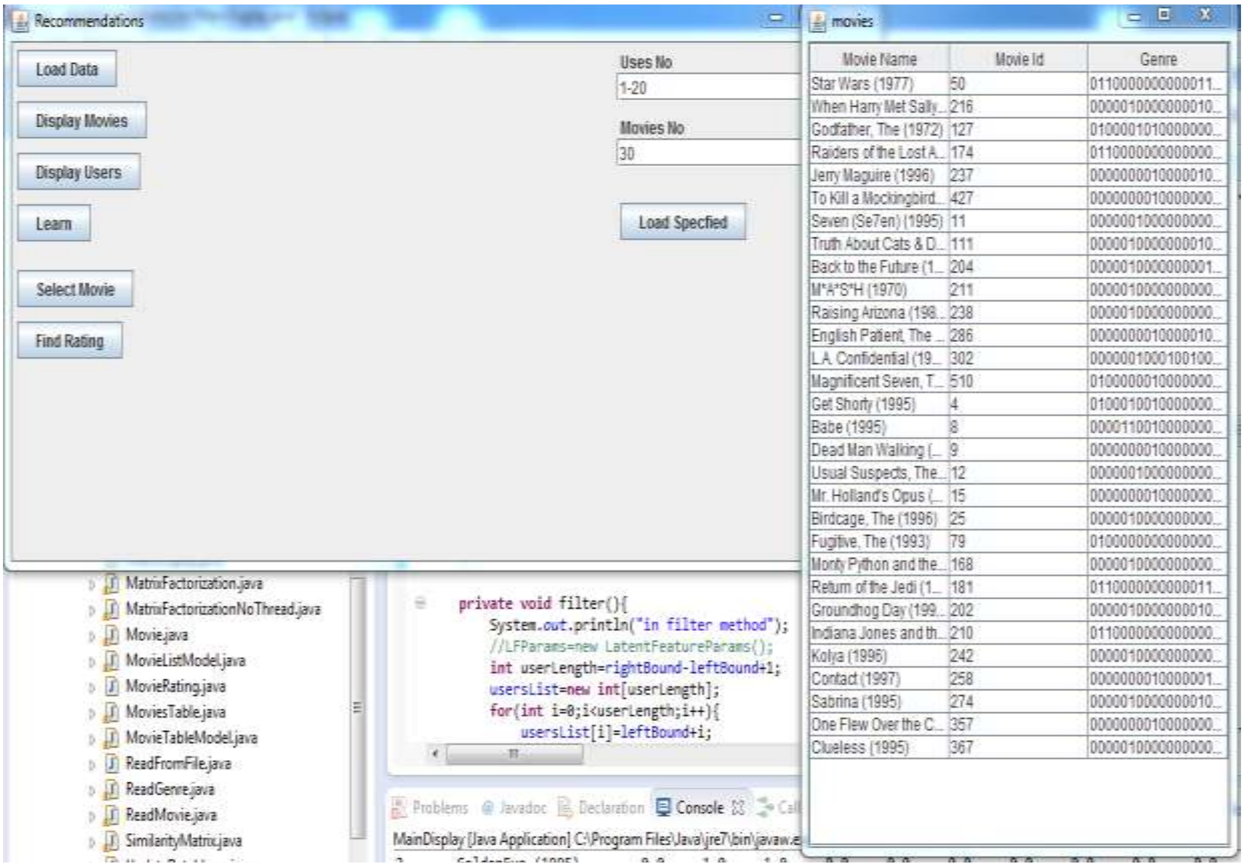

Figure 3. Learning Mechanism

Figure 4. Recommended Movies

V. CONCLUSION AND FUTURE SCOPE

A. Conclusion:

In this project, we represent a personalized movie recommendation system by combining social network factors: personal interest, interpersonal similarity, and trust value. We run the learning algorithm on datasets and calculate the result. The results are better than Content-based filtering and Collaborative filtering methods. We are able to identify characteristics of the users and the movies, which help us to improve the prediction accuracy.

B. Future Enhancement:

1. Adding user location and hobbies to train dataset.
2. Respond to the user's request for movie recommendations in real time

ACKNOWLEDGMENT

I would like to profoundly thank Management B.M.S College of Engineering for providing such a healthy environment for the successful completion of this work. The work reported in this paper is supported by the college through the TECHNICAL EDUCATION QUALITY IMPROVEMENT PROGRAMME [TEQIP-II] of the MHRD, Government of India

REFERENCES

[1]. X Qian, H Feng, G Zhao, T Mei "Personalized Recommendation Combining User Interest and Social Circle", Knowledge and Data Engineering, IEEE Transactions, 2014.
[2]. X. -W. Yang, H. Steck, and Y. Liu. "Circle-based recommendation in online social networks".KDD'12, pp. 1267-1275, Aug. 2012.
[3]. M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. -W. Zhu and S. -Q. Yang. "Social contextual recommendation". CIKM'12, pp. 45-54, 2012.
[4]. M. Jamali and M. Ester."A matrix factorization technique with trust propagation for recommendation in social networks".In Proc. ACM conference on Recommender systems (RecSys), 2010.
[5]. R. Salakhutdinov and A. Mnih."Probabilistic matrix factorization".In NIPS 2008, 2008.
[6]. G. Adomavicius, and A. Tuzhilin. "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions". Knowledge and Data Engineering, IEEE Transactions on, pp. 734-749, Jun. 2005.
[7]. M. Jahrer, A. Toscher, and R. Legenstein."Combining predictions for accurate recommender systems".KDD'10, pp. 693-702, 2010.

[8]. H. Ma, I. King, and M. R. Lyu.“Learning to recommend with social trust ensemble”.SIGIR’09, 2009.

[9]. X. Yang, Y. Guo and Y. Liu. “Bayesian-inference based recommendation in online social networks”. The 30th Annual IEEE International Conference on Computer Communications (INFOCOM), 2011.

[10]. J. Huang, X. Cheng, J. Guo, H. Shen, and K. Yang.“Social recommendation with interpersonal influence”.In Proceedings of the 19th European Conference on Artificial Intelligence (ECAI), pp. 601–606, 2010.

[11]. H. Ma, H. Yang, M.R. Lyu, and I. King. “Sorec: Social recommendation using probabilistic matrix factorization”. CIKM’08, 2008.

[12]. J. Wang, A.P. de Vries, and M.J.T. Reinders.“Unifying user-based and item-based collaborative filtering approaches by similarity fusion”.SIGIR'06, 2006.

[13]. P. Cui, F. Wang, S. Liu, M. Ou, S. Yang, and L. Sun. “Who should share what? item-level social influence prediction for users and posts ranking”. SIGIR, pp. 185-194, 2011.

[14]. H.R. Kim and P.K. Chan. “Learning Implicit User Interest Hierarchy for Context in Personalization”. In Proc. of International Conference on Intelligent User Interface, 2003.

[15]. O.Phelan, K. McCarthy, and B. Smyth. "Using twitter to recommend realtime topical news."In Proceedings of the third ACM conference on Recommender systems, pp. 385-388.ACM, 2009.

[16]. R. Sinha, and K. Swearingen, “Comparing recommendations made by online systems and friends”. Dublin, Ireland, 2001.

[17]. A. Rahman, and S. Hailes, “Supporting trust in virtual communities”.In Proceedings of the 33rd Hawaii International Conference on System Sciences. Maui, HW, USA (2000).

[18]. H Ma, D Zhou, C Liu, MR Lyu, I King. “Recommender Systems with Social Regularization”. Fourth ACM international conference on Web search and data mining, 2011.

[19]. A Bellogín, P Castells, I Cantador. “Self-adjusting Hybrid Recommenders Based on Social Network Analysis”.34th international ACM SIGIR conference on Research and development in Information Retrieval, 2011.

[20]. Xavier Amatriain “Mining Large Streams of User Data for Personalized Recommendations” ACM SIGKDD Explorations, 2012.

[21]. L Sharma, A Gera “A Survey of Recommendation System: Research Challenges” International Journal of Engineering Trends and Technology, 2013

[22]. GS Sadasivam “A Survey on Personalized Recommendation Techniques”

[23]. Y Chen, C Wu, M Xie, X Guo “Solving the Sparsity Problem in Recommender Systems Using Association Retrieval” , Journal of computers, 2011