International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2735-2742
© International Research Publication House. http://www.irphouse.com

Sparsity and Cold Start Recommendation System Challenges Solved by Hybrid Feedback

Marwa Hussien Mohamed[1]
Information systems Department
October 6 University,
Fayoum University, Cairo, Egypt.

Mohamed Helmy Khafagy[2]
Computer Sciences Department
Fayoum University
Fayoum, Egypt.

Heba Elbeh[3]
Computer Sciences Department
Menoufia University
Shebin Elkom, Egypt.

Ahmed Mohamed Abdalla[4]
Electronics Department
Collage of technological study,
PAAET, Kuwait.

ORCID: 0000-0003-3794-6767
ORCID: 0000-0003-0479-0516
ORCID: 0000-0003-2664-3533
ORCID: 0000-0002-9100-7587

Abstract

The Recommendation system is an important issue to E-commerce business for online websites today to increase sales of products to users. The Recommender system helps users to find his interests more easily according to his favorite movies, song and products using rated history items or matching profiles with other users. Collaborative filtering is one of recommendation system techniques using the similarity between users. It's suffered from little rating on items from users this cause sparsity problem and it's a challenge led to poor accuracy of recommendation. In this research, we use association rule with clustering based to solve accuracy and sparsity challenges of recommended items. We solve cold start problem by using root mean square error to find the average of rated songs. Our experiments applied to last.FM music data sets with user's implicit interaction records and explicit rating records of items named by hybrid feedback. We use association rule to count every purchase per transaction and compute similarities by cosine vector similarity to make a recommendation. Our experimental results show that the new proposed algorithm has better performance and more accuracy compared to basic collaborative filtering techniques when data are spare using precision, recall and F-metrics evaluations and recommend novel items to users.

Keywords: Recommender systems, collaborative filtering, content-based filtering, association rule mining, root mean square error and cold start problem.

1. INTRODUCTION

Recommender systems became a good call tool or assistant to dump such undesirable tasks like extraction, analysis, and process formidably long time operations. Worse yet, activities involving humans are inevitably subject to human errors which will cause poor or wrong choices[1].

Technology is developing thus quick and dissemination of information has exaggerated, as well as the needs of consumers that are more complex[2].

Manufacturers and suppliers had an issue in providing services that meet the users' desires for the convenience of shopping for the service attributable to a business that makes the competition has a lot of activity. During this era of competition, complicated information causes overload issues that successively are time intense [1].

Recommendation systems are data filtering system that aids users in predicting rating or preference of an associate degree item below users' thought. The systems supply users alternate picks while not having to figure out all the main points by themselves [3].

All recommender systems (RS) need a model of the users' interests to operate. We do this using explicitly or implicitly data, without ambiguity categorical their interests in things. On the opposite hand, implicit feedback is generated by the RS itself, through inferences it makes concerning the user's behavior. What constitutes an implicit feedback depends on the application domain: generally, it'll be one or multiple evident and measurable parameters that arise out of the user's interactions with the RS [4]. In this research, we merge and use these two data implicit and explicit data (Hybrid feedback)from a million song data set (Last. FM).

A recommender system [5] has different filtering techniques:

- Collaborative filtering is divided into item based and model based its find the similarities between users' preferences.
- Content based filtering finds the similarities between items.
- Demographic filtering recommend items to users based on demographic information similarities like age, gender, living area, education, etc.
- Hybrid filtering it's a combination between collaborative filtering, content based filtering and hybrid filtering.

Figure 1 shows a Model recommendation process to find similarity between items or users and recommend items though recommendation engine.

In this paper, we focus on discuss recommender system steps to build a model and talk about recommender system challenges. We list some of the recommender system public datasets like million song datasets. We explain collaborative filtering techniques and types and how to compute the similarity measures. We explain our new proposed algorithm that's used to solve sparsity, accuracy and recommend to user's novel items or songs while using precision, recall, and F measure metric. We use new technique to recommend top three rated songs to the new user to solve cold start problem.

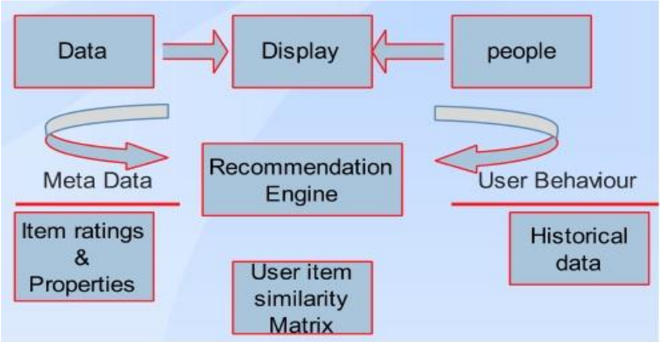

Fig 1. The Recommendation Process Model.

The rest of this paper is organized as follows: a recommender system foundation in section 2. Collaborative filtering is discussed in section 3. New proposed techniques described in detail in Section 4. Cold start problem in section 5. Experimental evaluation is discussed in section 6. Finally, a conclusion is given in Section 7.

2. RECOMMENDER SYSTEM FOUNDATIONS

In this section we will talk about recommendation system design guidelines and challenges face recommender system. In this research, We will focus on sparsity, novelty and accuracy to be solved in our new proposed algorithm and recommend novel items to users and solve cold start problem.

A. *Design tips for Recommendation System*

There are several works within the literature that build accessible several principles to construct a recommender system. However, They deliver general principles that ought to guide the creating of a recommender once the initial technical selections are created [6]. During this section, several questions are addressed:

- A way to build a smart choose to once begin by coming up with the system?
- Procedures: that recommendation procedure to apply?
- Design: a way to deploy system, whether or not distributed or centralized ?
- User summary: The information concerning users?
- Users: Active users and their objectives?
- Records: style of data set and its features?
- Use: wherever will we have a tendency to apply the recommender system?
- Why to develop: The motivation towards creating a recommender system?
- Categories: the kind of suggestions provided by recommendation?
- Metrics: Performance measures and metrics?
- Item ratings: the kind of ratings given by users to products?

Taking into concern, an organized execution of the construct advice system is finished in Java.

B. *Recommendation system challenges [5]:*

1. Cold start problem: it's happening when we have a new product and new user to the system.
2. Novelty and diversity: we have in the system new products to which user we will recommend this item and make our recommendation has more diversity for the user rather than which products he had seen on the system.
3. Scalability: the ability for the system to work when the number of users and products on the system increased.
4. Accuracy: we need to compute the accuracy of our prediction compared to the training data in the system.
5. Sparsity: many users on the system buy a product and didn't rate the items they watch or purchase.
6. Gray shape: it happens, if a malicious user or competitor enters into a system and starts giving false ratings on some items either to increase the item popularity or to diminish its popularity.

3. COLLABORATIVE FILTERING

A collaborative filtering algorithm is a type of recommendation system widely used and successfully algorithms it's building a model using user rating to gain more novelty and diversity between recommended items to the user. However, collaborative filtering faces challenges of sparsity and recommendation accuracy [9].

A. *Collaborative filtering algorithm guidelines:*

There are general main steps to build an algorithm:

1. Every rated item with active users is retrieved.
2. We compute the similarity by a set of retrieved items to find the targeted users. We will find the k nearest neighbors of target items with similarities.
3. Target item prediction it computed by summing weighted for the active users rate on the k nearest similar items.

Collaborative filtering [10] divided into two categories:

1. Memory-based (Also, named as user based): It takes into concern the previous user-item rating to make the next prediction for this user. It has two main sections to compute the similarity: A user- item filtering fined the similarity between users' rating items. In contrast, item–item filtering its compute similarity between users like this item and other items similar to this item's specification rated by other users in the system.
2. Model-based: it’s used to predict rates of items we will recommend to the user using machine learning this algorithm shown in figure 2 and broken into 3 sub types.

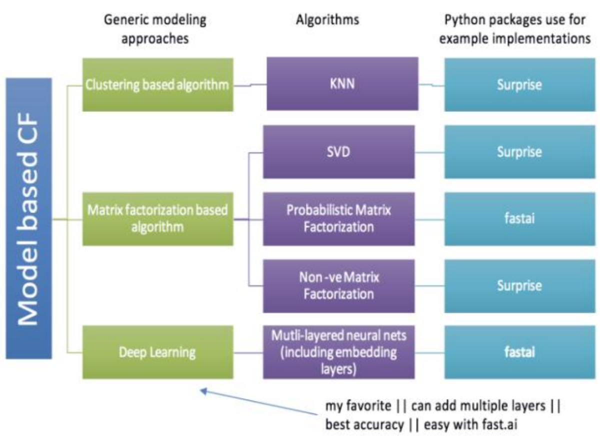

Fig 2. Model based algorithm types of collaborative filtering approach.

B.*Collaborative filtering Techniques used to solve sparsity:*

We are going to talk about different collaborative filtering techniques want to solve the sparsity problems and list some advantages and disadvantages for collaborative filtering recommendation. Data mining like (dimensionality reduction, association rule mining, web mining) used to solve sparsity by taking into a priority to search out find hidden patterns and relationships between items and users to analyze users' shopping behavior.

Data mining techniques[11] have no sparsity problems and sensitive to the modifications of user preferences and it's sensible with hybrid recommendation techniques, however, its main disadvantage is that the issue in a way to understand the techniques and the way we can match inputs on data sets.

Dimensionality reduction disadvantage [12]:

- Whereas reducing the dimensionality may result in lost useful information.
- Clustering based mostly used historical rating data to create clusters, but it ignores other data resources like social connections of users and also the correlations between items.
- It doesn't affect the modification in the user preferences.

Association rule mining [12]:

- It doesn’t take into concern the number of times the user used this item or purchase it per transaction.
- Ability to process with the huge number of items and users while applying association rule mining.
- Take into account only for correlations between items and user's profile data.

Implicit Feedback [12]:

- A Recommendation process based on track users' preferences, behaviors like viewing list, rated items, preferred items in the profile. It's good with hybrid techniques also it needs to analyze user behaviors [13].
- Whereas finding similarity between users and items some information lost while converting implicit data between purchased this item or not.

C. *Collaborative filtering Similarity measures:*

Collaborative filtering [12] calculates the similarity between two users or two items by matching the correlated items rated by both users. The most famous similarity measures are the cosine similarity, Pearson correlation coefficient,.

Cosine Vector Similarity[14] (CVS) equation 1

$$W_{xy} = \frac{x \cdot y}{|x| \times |y|} = \frac{\sum_{i=1}^{n} (r_{xi})(r_{yi})}{\sqrt{\sum_{i=1}^{n} (r_{xi})^2}\sqrt{\sum_{i=1}^{n} (r_{yi})^2}} \qquad (1)$$

Pearson's Correlation Coefficient [15] (PCC) equation

$$W_{xy} = \frac{\sum_{i=1}^{n} \left(r_{xi} - \overline{r_x}\right)\left(r_{yi} - \overline{r_y}\right)}{\sqrt{\sum_{i=1}^{n} \left(r_{xi} - \overline{r_x}\right)^2 \sum_{i=1}^{n} \left(r_{yi} - \overline{r_y}\right)^2}} \qquad (2)$$

Where x and y are users rated a number of items n. The average rating of item x is rx and an average rating of the item y are ry. The weight measures between the preferences /interests of users are WXy ∈ [-1, 1].also rxi and ryi are the ratings of the users x and y on item i. the similarity between active users are measured by any of these two equations. We can use eq (3) to predict the weighted average of rates on items like j by the formula predXj by using all neighbors of active users. The set of neighbors is represented by K.

$$pred_{xj} = \overline{r_x} + \frac{\sum_{y=1}^{k} \left(r_{yj} - \overline{r_y}\right) \times w_{xy}}{\sum_{y=1}^{k} |W_{xy}|} \qquad (3)$$

Although, these equations used to compute the similarity[12] are successful to find the nearest neighbor users, but have some difficulty to find this when data are spare because many users don't like to rate viewed or purchased items this happen with a large number of items on the system to spare of rating.

4. NEW PROPOSED TECHNIQUE

In this section we will discuss our new proposed technique used to increase the accuracy of predicted items will be recommended to users also solve the sparsity problem by using the merging between implicit and explicit data. Our data sets are part of million song data sets is last. FM datasets [16] and use this research paper [17].

Our new techniques are best compared to the basic collaborative filtering techniques. The main advantage of our proposed algorithm is how to find the correlation between items in a spare data this solved by using implicit data(Songs played, number of play counts to a specific song, play ratio for certain category of songs and tagging information) with explicit data rated songs. Also, a similar behavior pattern can compute easily with our new algorithms this lead to a good recommendation accuracy.

The main idea we solve in this technique is sparsity challenge. Where data are spare in this case users don’t have any co-played music and we need to find similar preferences on pop music as an example by matching between other users in the system.

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2735-2742
© International Research Publication House. http://www.irphouse.com

However, such a result is not true in more sparse rating data at basic collaborative filtering. Even though both users do not have any co-played music, both of them are fans of pop music. Thus, we should consider them to be similar in the case that they are sharing a very similar preference for pop music. We will compute the similarity between users on playing songs groups by applying the clustering techniques and association rules to find hidden items between active users or users interested in items available on the system.

We will discuss proposed algorithms phases:

Phase I: Pre-processing Reducing data dimensionality for the rule mining part

1. User-Song Matrix:
   Present listening history of users and rated songs in the form (user, song, play-count).
2. Cluster sets for song dataset:
   Group/Cluster songs in a hierarchical structure based on features including tags and song duration.
3. The Model is listening history of users on clustered songs:
   Build a profile of a user's taste on clustered songs based on the music he/she has played and rated (matching song features/cluster of tags and duration to user profiles). Express ratio of user play counts for a certain group of songs.
4. Output:
   Cluster sets for users listening Grouped songs and rated songs groups and the profile of the user's tastes on clustered songs as input for association rule mining.

Phase 2: Prediction process on users' unknown preferences based on association rule mining

1. Association rules generator:
   Running an association rule mining and generate rules according to users' listening history (user play counts on song clusters).
2. Extract association rules on clusters:
   Compute similarity between tastes of users on clustered songs based on their overlapping of group songs listened (Listened group songs overlap ratio). Discovering the similar listening behaviors between users (users' overlapping interactions in song clusters) and song duration
3. Select rules to satisfy listening history of the active user:
   Find rules that their antecedent side (left side of a rule) is according to the user's previous preference list.
4. Predict user' preferences on clustered songs:
   Predict users' preferences on clusters. Via is extracting the consequent side of the rules in which antecedent side of the rules is according to the user's previous preference list. Then, the consequent side of the rules is refined and just the rules with the highest preference level value between users (highest play counts) is selected.
5. Output:
   Predicted list of users' preferences on the song categories (clusters) that have a high score against the users' profiles (highest play counts).

Phase III: Recommendation a list of songs available in each cluster that match the user' preferences

1. Calculating similarity between songs based on songs' features and rated songs:
   Calculate the similarities between pairs of songs using rated songs and the information about song's features which are the artist, year, title, release, song popularity, artist familiarity, duration, and tags to acquire a similarity degree between items.
2. Finding a set of songs like those already present in the user profile and rated by users:
   We will compute a threshold value to find the similar songs compared with user preferences. Then, match the user' profile (listening history of the user) and rated songs if available with song features to recommend songs which are similar to those she or he has liked in the past.
3. Output:
   Recommend similar songs from each cluster based on threshold Songs with bigger similarity value than the calculated threshold are selected for recommendation to the active user.

We will discuss our new algorithm three phases from pre-processing phase, prediction phase using association rule to the recommendation phase which common items to users in figure 3.

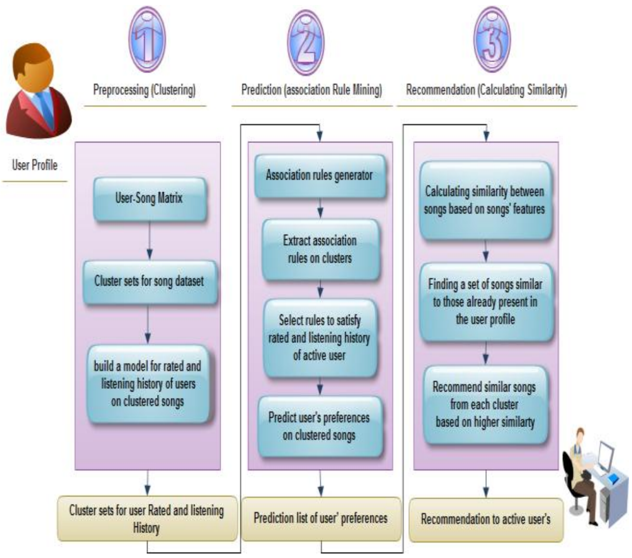

Fig 3 . New algorithm three phases.

5. The proposed algorithm to solve cold start problem

One of the most recommendation system challenges is which item we will recommend to new users on our system and which user we will suggest the new product to this user.

Some of the researchers using ask to rate techniques to help system know the users' preferences throw suggest some of items and ask user to rate it. Also, another researches used demographic recommendation as we ask the user to make a profile and we divide users according to their living area place or age, etc. to recommend items similar to users in his ageing or with the same culture.

In this paper, we will solve cold start problem for new user by finding the similarity between rated songs by others registered users:

- Creating hash map to find the common rated songs and but the result in an array.
- Our data sets are divided to 70% training sets and 30% test sets.
- Find the similarity between songs using cosine vector similarity.
- Create a sheet contain all similar songs together.
- Make a prediction for rates items based on songs similarity and user rating on songs.
- Compute MAE mean absolute error to songs.
- Select the highest rated songs from the list and select the highly 10 songs.
- Recommend the top 3 rated songs to users from the system.

6. EXPERIMENTAL EVALUATIONS

This section we will show the results of our experiments on last.fm million song data sets with merge between implicit and explicit data to improve the accuracy of recommendation systems [16, 17]. We will compare our algorithm with traditional collaborative filtering techniques while changes in the data sparsity level.

In this paper, we will focus on music recommendation only so that we use million song datasets [6] because it's large and free datasets in the music domain. It is constructed from about one million songs and users, in which each user plays a small set of songs. These data sets have an implicit data for users' preferences and have item matrix spare and Last.fm dataset for tagging activity of songs.

A. Experimental methodology and evaluation metrics

We use precision and recall equation to measure the accuracy and evaluate these data sets between our new algorithm and basic collaborative filtering [18,19].

TABLE 2: *DESCRIPTIONS RECOMMENDATION ACCURACY METRICS.*

Predicted items/actual	Relevant	Irrelevant
Recommended	True Positive (TP)	False Positive (FP)
Not recommended	False Negative (FN)	True Negative (TN)

The precision measure [18] the ability of the system to return relevant items among a set of irrelevant and relevant items and it's calculated by the equation (4) and description for accuracy metric seen in table 2.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (4)$$

The Recall measure [18] the ability of the system to return the relevant items only and it's calculated by the equation (5).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (5)$$

Another evaluation metric is an F measure [18] used to find the difference between precision and recall function and in an equal weight of each of them. The metric equation is (equation 6). It's the higher result mean higher accuracy of recommendation.

$$\text{F} - \text{measure} = \frac{(2 * \text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \qquad (6).$$

We focus on number rated songs(this an explicit data) on user profile from 5-1 this will indicate the users, which rate 5 for a song he is very likely it and rate from 2-1 he dislike this song.

Also, we focus on user profile preferences implicit data,We divide the data into categories from 5-1 depending on the high number of players or listing songs in such way the song In range 80-100% this like the rating 5, and songs from 1-20 % this like rating 1. Then we build a user-item rating matrix we will compare with basic collaborative filtering. That's the firstly preprocessing step.in order to find similarities and train models based on user profile preferences like (user-song-play count) 80% as a training dataset and 20 % as a test data set. These steps applied to last.FM million song datasets hybrid feedback.

We cluster data sets into three clusters determine the song's level like (level 0 – all songs), (level 1- songs tags (pop, rock, jazz, etc.)) and (level 3 – song duration (very short- one minute or less, the shorter – from 1 minute to 3, medium form 3 minutes to 5 , long - from 5 minute to 8, very long- more than 8 minute)).this used to help association rule mining by defining optimal numbers of clusters and used the song durations to build it this will be viewed in figure 4. Also, we need to reduce the size data send to the association rule to get the best performance extraction of an association rule.

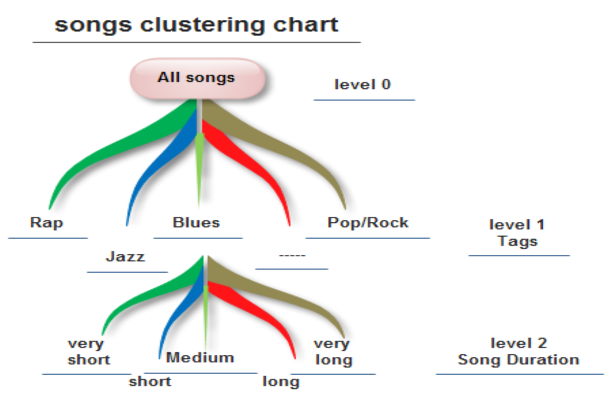

Fig 4 . Songs clustering chart.

After this, firstly preprocessing step we construct datasets with a spare level according to play count for songs. We classify data into ten group sparsity levels from listing records

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2735-2742
© International Research Publication House. http://www.irphouse.com

and it's level between (0.2 to 0.4), (0.4 -0.6),(0.6-0.8)and (0.8-1.0) the last one has the highest sparsity.

Equation 7 calculates the sparsity level [13].

$$\text{Sparsity measure} = 1 - \frac{\text{NR}}{\text{nUsers} - \text{nItems}} \qquad (7)$$

The symbol nR the total number of play counts and nUsers number of users and nItems are number of items or songs on the user item matrix.

B. Experimental environment

We run our experiments on a framework machine 16 GB of RAM and Intel Core I7 CPU and windows 7. We used IntelliJIDEA software to write Java programming code to run our recommender systems. Also, we used the WEKA environment to apply association rule and clustering-based techniques to our new algorithm.

C. Experimental results

In this section, we run our four experiments according to the sparsity level, as we mention in the previous section to show the difference accuracy result from our new proposed algorithm against basic collaborative filtering techniques. Table 3 shows the result of accuracy of precision, recall and F-measure metric.

Also, we need to mention that we used the merge datasets between implicit and explicit last.Fm data sets hybrid feedback [13] to find a high accuracy level about recommended items or songs to users.

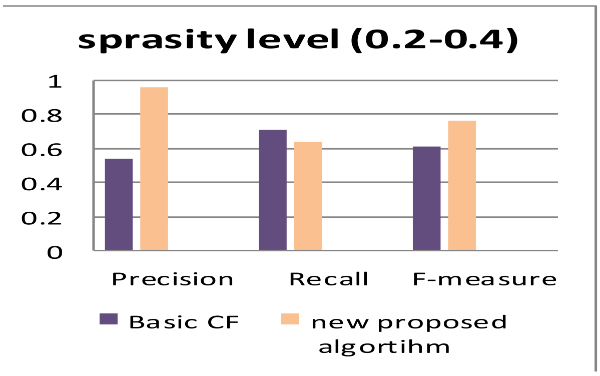

Fig. 5 experimental results while sparsity level from 0.2-0.4.

According to the values in table 5 and the results are shown in figure 5,6,7 and figure 8 through the sparsity level the accuracy with basic CF its decrease while the sparsity increased. But, our new algorithm controls the accuracy through the sparsity level against basic CF and improved by 22% this because the ability to find neighbors and association rule to recommend items to users and merging the implicit and explicit data (rating values to songs).

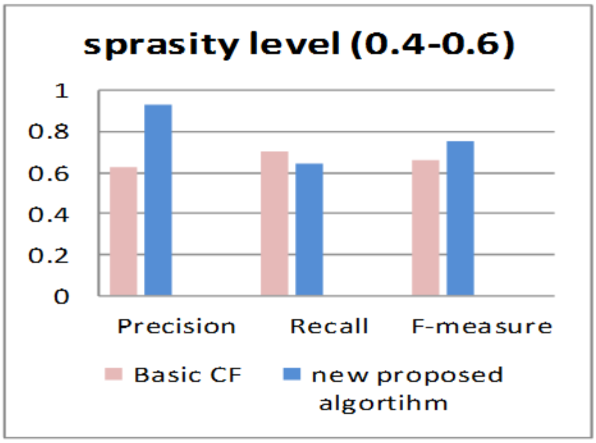

Fig. 6 experimental results while sparsity level from 0.4-0.6.

The value for precision in our proposed algorithm is good because we recommend a less number of not matching songs to the user. We have a higher precision value, it's improved by 37% .The recall based on songs not recommended to users and it's improved in our algorithm by 10%.The F measure it's improved by 17% so that our proposed algorithm is the best in recommended songs to users**.**

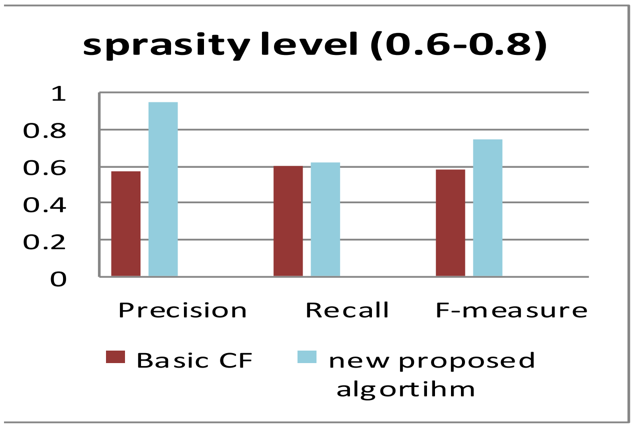

Fig. 7 experimental results while sparsity level from 0.6-8.0.

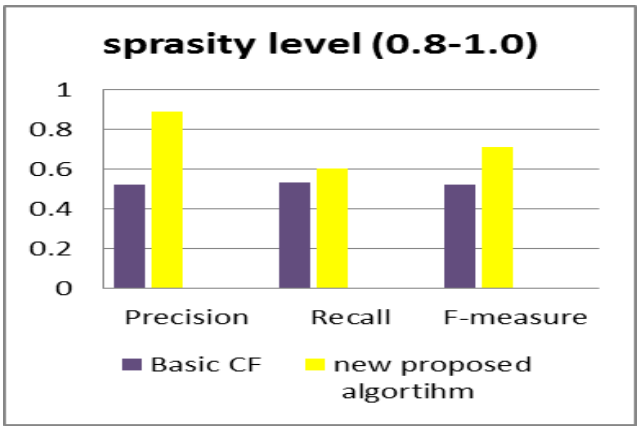

Fig. 8 . Experimental results while sparsity level from 8.0-1.0.

TABLE 3: EXPERIMENTAL RESULTS

Techniques	Sparsity from (0.2-0.4)		
	Precision	Recall	F-measure
Basic CF	0.54	0.71	0.61
New proposed techniques	0.96	0.64	0.76
Techniques	**Sparsity from (0.4-0.6)**		
	Precision	**Recall**	**F-measure**
Basic CF	0. 63	0.7	0.66
New proposed techniques	0.93	0.64	0.75
Techniques	**Sparsity from (0.6-0.8)**		
	Precision	**Recall**	**F-measure**
Basic CF	0.57	0.6	0.58
New proposed techniques	0.95	0.62	0.75
Techniques	**Sparsity from (0.8-1.0)**		
	Precision	**Recall**	**F-measure**
Basic CF	0.52	0.53	0.52
New proposed techniques	0.89	0.6	0.71

The experimental results for a new registered user to our system is shown in table 4 by finding the root mean square error average and we select top 10 songs and recommend to the user top three rated songs then using our new proposed algorithm to recommend items according to his preferences.

Table 4. Recommendations list for user (movies with average ratings >= 3)

Song Name	Average Rating
Ison	3.1666
Body and soul	3.8353
SI Clair	3.33620
Begin to Hope	3.14173
The Boy from Ipanema	4.06091
Off Key	3.31395
When you're not with me	3.27814
This love that I've found	3.58333
Hapless	3.10526
A long and ugly road	3.11764

Table 5. Top 3 Recommendations Based Upon Recent Preference

Song Name	Average Rating
The Boy from Ipanema	4.06091
Body and soul	3.8353
This love that I've found	3.58333

Our proposed algorithm advantages:

1. The ability to recommend items using user preferences not only rated items using user item tags and the number of play count for songs.
2. Merging between explicit data and implicit data to improve accuracy.
3. The ability to control accuracy while sparsity increased.
4. We use number of playing count with association rule to find hidden relationships between users.
5. We can recommend to user's songs similar to his preferences and ability to recommend novelty and diversity songs which is one of recommended items challenges.
6. We solve cold start challenges to the new user by recommending top three songs rated from old users after registration to the site.

7. CONCLUSIONS AND FUTURE WORK

Recommendation systems are the most research areas today. Due to the rapidly increasing data size and social media information about users will help us to recommend items to users more accurate and the importance of big data [19,20].
In this research paper, we used the million song datasets (last.fm) it's a type of implicit data merged with explicit data users rated songs to improve the accuracy of prediction.
We apply our experiments on different levels of sparsity and our new algorithm improves in accuracy through the F-measure by 15% it's relevant to best accuracy result against basic collaborative filtering techniques which suffer from finding neighbors where data are spare and our algorithm also solve the novelty and diversity challenge to recommend songs to users. Help system to recommend to the new users' top three songs rated by other users as one of recommendation system challenges.

In the future, what about applying this algorithm with big data sizes [21] and using spark and flinkto improve recommendation systems[22] .
What about test our algorithm with huge datasets by using HDFS to gain high performance[23]
Also, change the type of data like movies and apply our new algorithm then compare this algorithm with other recommendation system techniques like k-means, hybrid representations, probabilistic learning. We may use HDFS when running our recommendation systems with big data rated items and users [24].

REFERENCES

[1] Phongsavanh Phorasim,Lasheng Yu." Movies recommendation system using collaborative filtering and k-means", International Journal of Advanced Computer Research, Vol 7(29),pp.52-59.

[2] Sajal Halder, A. M. Jehad Sarkar, Young-Koo Lee." Movie Recommendation System Based on Movie Swarm", 2012 Second International Conference on Cloud and Green Computing,(2012).pp.804-809

[3] Marwa Hussien Mohamed, Mohamed Helmy Khafagy, Mohamed Hasan Ibrahim," Two Recommendation System Algorithms used SVD and Association Rule on Implicit and Explicit Data sets", international journal of

scientific & technology research ,volume 8, issue 11, november 2019, ISSN: 2277-8616,pp:508-515.

[4] Gawesh Jawaheer, Martin Szomszor, Patty Kostkova,"Comparison of Implicit and Explicit Feedback from an Online Music Recommendation Service",(2010).doi. 10.1145/1869446.1869453.

[5] Marwa Hussien Mohamed, Mohamed Helmy Khafagy, Mohamed Hasan Ibrahim . "Recommender Systems Challenges and Solutions Survey", 2019 International Conference on Innovative Trends in Computer Engineering (ITCE'2019), Aswan, Egypt, 2-4 February 2019,pp.149-155

[6] Sanjeev Dhawan, Kulvinder Singh, Jyoti." High Rating Recent Preferences Based Recommendation System", 4thInternational Conference on Eco-friendly Computing and Communication Systems, ICECCS 2015, Procedia Computer Science 70(2015), pp. 259 – 264.

[7] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

[8] Mustansar Ali Ghazanfar and Adam Prugel-Bennett." A Scalable, Accurate Hybrid Recommender System". Third International Conference on Knowledge Discovery and Data Mining, WKDD 2010, Phuket, Thailand, 9-10 January 2010.

[9] Mohammed FadhelAljunid and Manjaiah D.H."A SURVEY ON RECOMMENDATION SYSTEMS FOR SOCIAL MEDIA USING BIG DATA ANALYTICS". International Journal of Latest Trends in Engineering and Technology, Special Issue (SACAIM 2017), pp.48 -58.

[10] J. Bobadilla , F. Ortega, A. Hernando, A. Gutierrez," Recommender systems survey". Knowledge-Based Systems 46 , 2013, pp.109–132

[11] Yagnesh G. patel, Vishal P.Patel," A Survey on Various Techniques of Recommendation System in Web Mining", 2015 IJEDR | Volume 3, Issue 4 | ISSN: 2321-9939, pp:696-700.

[12] Maryam Khanian Najafabadi, Mohd Naz'ri Mahrin, Suriayati Chuprat, Haslina Md Sarkan." Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data", Computers in Human Behavior 67 (2017) .pp.113-128

[13] Cheng, L. C., & Wang, H. A.," A fuzzy recommender system based on the integration of subjective preferences and objective information", Applied Soft Computing, 18, 2014,pp:290-301.

[14] Acilar, A. M, Arslan, A " A collaborative filtering method based on artificial immune network". Expert Systems with Applications 36, (2014),pp. 8324-8332.

[15] Huang, S., Ma, J., Cheng, P., & Wang, S." A hybrid multigroup coclustering recommendation framework based on information fusion". ACM Transactions on Intelligent Systems and Technology (TIST) 6(2):27, 2015, pp.1-22.

[16] http://millionsongdataset.com/lastfm

[17] Gunjan Advani,Neha Soni." A Novel Method for Music Recommendation using Social Media Tags", International Journal of Computer Applications (0975 – 8887) Volume 122 – No.2, July 2015,pp.37-43.

[18] Subhash K. Shinde , Uday Kulkarni." Hybrid personalized recommender system using centering-bunching based clustering algorithm". Expert Systems with Applications39, 2012, pp. 1381-1387.

[19] Marwa Hussien Mohamed , Mohamed Helmy Khafagy, Mohamed Hasan Ibrahim . "Hash Semi Join Map Reduce to Join Billion Records in a Reasonable Time." Indian Journal of Science and Technology, vol. 11, no. 18, Jan. 2018, pp. 1–9., doi:10.17485/ijst/2018/v11i18/119112.

[20] Marwa Hussien Mohamed , Mohamed Helmy Khafagy, Mohamed Hasan Ibrahim . "From Two-Way to Multi-Way: A Comparative Study for Map-Reduce Join Algorithms". WSEAS Transactions on Communications, Volume 17, 2018, pp. 129-141.

[21] Marwa Hussien Mohamed and Mohamed Helmy Khafagy. "Hash semi cascade Join for joining multi-way map reduce." 2015 SAI Intelligent Systems Conference (IntelliSys)(2015): pp.355-361.

[22] Radya R. Sahal, Mohamed Helmy Khafagy, Fatma A. omara,"Big data multi-query optimization with apache flink",international journal of web engineering and technology,2018.

[23] Mostafa Rabea Kaseb,Mohamed Helmy Khafagy, Ihab A.Ali, Elsayed M.Saad,"Multi-split HDFS Technique for improving data confidentiality in Big Data Replication",2019 advances in intelligent systems and computing,pp.132-142.

[24] Mostafa rabea Kaseb, Mohamed helmy khafagy,ihab A.Ali,Elsayed M.Saad," Redundent independent files (RIF): A technique for reducing storage and resourses in big data replication ",2018 advances in intelligent systems and computing.