A Hybrid Approach to Improving Scalability in Collaborative Filtering

Pritha Ghosh
Dr.Sudhir Chandra Sur Degree Engineering College
540, Dumdum Road, SurerMath, Kolkata-700074, INDIA

Prosenjit Gupta
NIIT University
Neemrana, Rajasthan 301705, INDIA

ABSTRACT

The process of filtering information or patterns using techniques involving collaboration among multiple agents or data sources is known as collaborative filtering [14]. Applications of collaborative filtering typically involve very large data sets. Techniques of Collaborative filtering have been applied to many different fields such as sensing and monitoring data in mineral exploration, environmental sensing over large areas, financial data, such as financial service institutions or in web applications where the focus is on user information. It is based on the concept that everything is related to everything else [9]. One such popular field of development of collaborative filtering is Recommendation Systems. Recommendation systems were developed to guide users in a personalized way to a large set of possible options matching their choices and requirements. A content-based recommender system matches the attributes of a user‘s preferences and interests to the attributes of an object (item). On the other hand a collaborative filtering takes the approach of matching one user‘s choices to the choices of another user. The basic assumption behind this method is that other users‘ opinions can be selected and aggregated in such a way as to provide a reasonable prediction of the active user‘s preference. Hence a new hybrid and scalable recommendation system has been proposed in this research that combines techniques from Content-Based Recommender Systems, Collaborative Filtering, Location Aware Recommender Systems and Spatial Autocorrelation.

General Terms

Collective intelligence, Similarity search, Collaborative search engine , Content discovery platform, Decision support system.

Keywords

Recommendation System, Location Based Services, Collaborative Filtering, Movie Recommendation System, Hybrid Recommendation System

1. INTRODUCTION

Online movie recommendation sites have become one of the most promising applications that sustain numerous Internet based movie releases and promotions. Previously a large number of movies after their release were only available to a specific or limited group of viewers (e.g.: only to the citizens of a specific country). But now with the gradual growth of movie recommendation sites movies are available to a large group of internet explorers from all around the world. Now a user can get a list of recommended movies from a large set of movies, ranging from recent top releases to movies matching her preferences or previously made choices. This has been made possible by filtering the movie request made by the user and extracting similarities between various registered users to make as near and distinct recommendations as possible. Recommender systems often follow collaborative filtering that involves predictive models, heuristic search, data collection, user interaction and model maintenance. The system usually needs to be updated periodically with newly added ratings, items and users. In other words a recommender system is an information filtering technology designed to determine items that are most likely to the customer‘s tastes. After the best items have been determined they are recommended to the user. Recommender systems studies user preferences and form a profile of each customer usually based on ratings of items.

2. PRELIMINARIES

2.1 Collaborative Filtering

The gradual growth in the world of Internet has made it extremely difficult to effectively extract useful information from all the available online data. A popular techniques used for dealing with efficient extraction of information is called collaborative filtering[12]. This method follows a pattern of matching similarity between different sets of users to generate a new recommendation. A collaborative system consists of a database which contains the users’ ratings and is updated as new users are added or the existing users’ interacts with the system, (Ekstrand et al., 2010) [11].

The basic principle of collaborative filtering is to (i) to locate a subset of users having similar choices, tastes and preferences to that of the active user. (ii) Offering recommendations based on the subset of users located with similar choices. The basic principles of assumptions are: (i) Users with similar interest have common preferences. (ii) Sufficiently large number of user preferences is available for research.

Users’ preferences are compared based on their ratings and similarities are obtained while differences are used to predict recommendations to the current/active user. Collaborative filtering suffers from the sparsity problem. As not all users rate an item after viewing it, the available rating data is therefore typically very sparse, especially when a user or the item is new. A primary virtue of collaborative filtering is that it can surprise the user with relevant items that are not explicitly similar to items in the users’ profile also known as ‘outside the box’ recommendation ability (Burke, 2002) [4]. This is possible because it uses people-to-people correlations. Computing correlations between all pairs of users or items is an expensive and crucial step and can be avoided by using appropriate data partitioning techniques.

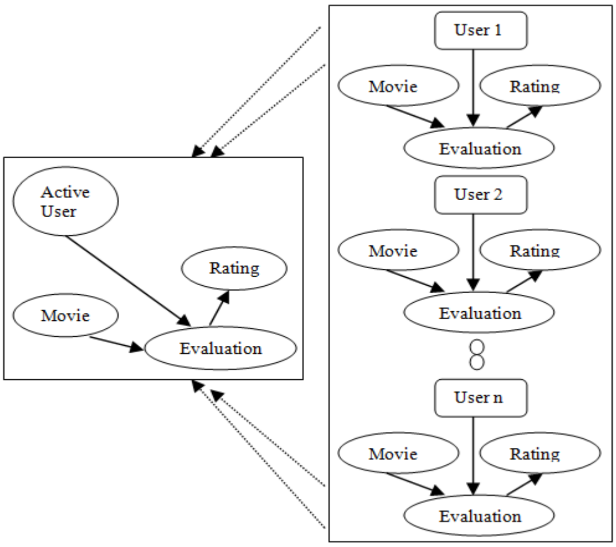


Fig 1: Collaborative Filtering Based Recommendation System

2.2 Content Based Methods

Another method for recommendation is content based information. In this approach filtering is done by comparing the attributes of an object (in this case movie) in the dataset to the attributes of the object rated by a user. Content-based recommendation systems analyze attributes of items described in the datasets to spot a match of items that are of particular interest to the user B.

There are various approaches to illustrate on the aspects of content based recommendation system. Some of them are as follows:

(A) Item Representation: Items that are recommended to the user are represented by a set of features, also called attributes or properties. For example in a movie recommendation system movie name, genre, rating, cast are considered as attributes to the object movie. Recommendations are based on such attributes.

(B)Keyword Based Matching: Most content-based recommender systems use keyword matching. Here the query which the user enters is broken down into tokens. Each token is a 'keyword'.The keywords are then matched with the data collection in the dataset.

The similarities between items and the ratings information obtained are used to predict whether the user will like or dislike the new item (Basu, Hirsh, and Cohen, 1998) [6], where the information of one recommender acts as features for the other. The main disadvantage of this method is that it is dependents upon machine's representations of items, which may be difficult to obtain. Content based methods cannot provide surprise recommendations to the user, because it uses the feature values of the items that the user has rated and will not recommend an item that does not share any of these values. Just as collaborative filtering, content based methods also suffer from the scarcity problem.

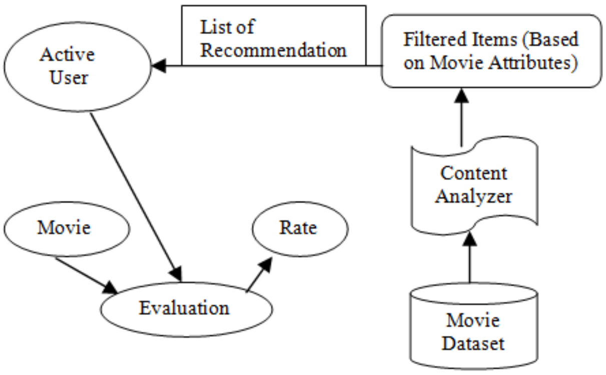


Fig 2: Content Based Recommendation System

2.3 Location Based Methods

LARS is a location-aware recommender system which is an application of location based systems, which uses location-based ratings to produce Recommendations to the users, which is form of spatial autocorrelation (Lo, C.P., Yeung, 2007)[10].It exploits user rating locations through *user partitioning*, which influences recommendations with ratings spatially close to the active user in a manner that maximizes system scalability while not sacrificing recommendation quality (J. Das, S. Majumder, and P. Gupta)[13][2][3]. In this location based approach the total user set has been divided into smaller subsets based on locations. Each set contains a few numbers of users and a similarity graph is established between them to generate a recommendation set. LARS can apply these techniques separately, or together, depending on the type of location-based rating available.

Based on the information delivery system, LBS are primarily of two types: pull, push. In case of a *pull service* the user issues a request to be automatically positioned and access the LBS he wants. For example a tourist roaming in a foreign land might desire to receive information about the nearest restaurants to his current location. He uses his mobile to issue an appropriate request (e.g. via SMS or WAP), which helps the network to locate his current position and respond with a list of restaurants located near his position. But in case of a push service, the request is issued by the Service Provider and not by the user himself. A practical example of push services is location based advertising, where users are informed about products of their interest, located at nearby stores.

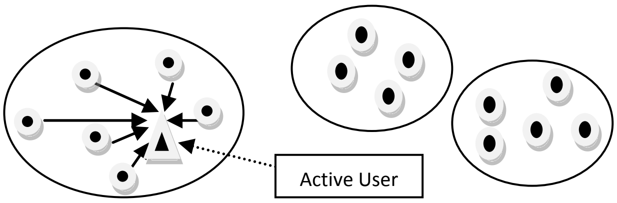


Fig 3: Location Based Recommendation System

2. HYBRID RECOMMENDATION APPROACH.

In this generated approach of recommendation a function is used to decide when to switch from one recommender to the other (Tran & Cohen, 2000) [5]. The rating provided by one user is compared to that of another user (collaborative filtering) and is used as an additional feature to the content based filtering when required. This procedure is then applied by location based filtering where the total set of users is divided into subsets based on their location (zip code). (Melville, Mooney, and Nagarajan, 2001) [7] ,used a content boosted collaborative filtering technique, in which they first solved the sparsity problem by making sets of data using content based information, and then used collaborative filtering on a much larger data set. In some cases special features like geographic location information (Brunato, M., Battiti, R., Villani, A., and Delai, A., 2002) [8], with previous Web site access information has also been a form of location based recommendation. One of the most popular amongst hybrids recommendation systems is Burke's taxonomy (Burke, R., 2005) [1] which is widely accepted by researchers.

The example of a Movie Recommendation System has been used to site the working of our Hybrid Recommendation Approach. A Movie Recommendation system (MRS) helps to match a group of users with their preferred movie sets based on attributes of movies stored in the dataset. It not only ease movie information overload but also provides assistance guidance, advisory, persuasion to our active user. A MRS has multiple perspectives:

(i) *Retrieval perspective:* It requires a minimal search cost to provide correct proposals as users know in advance what they want.

(ii)*Prediction perspective:* Based on popular evaluation it predicts to what degree user would like a movie.

(iii)*User perspective:* On study of user's previous history MRS recommends a new set of movies which was previously totally unknown to the user.

(iv*)Interaction perspective:* It not only educates the user about the product domain but also gives a positive feeling to the active user.

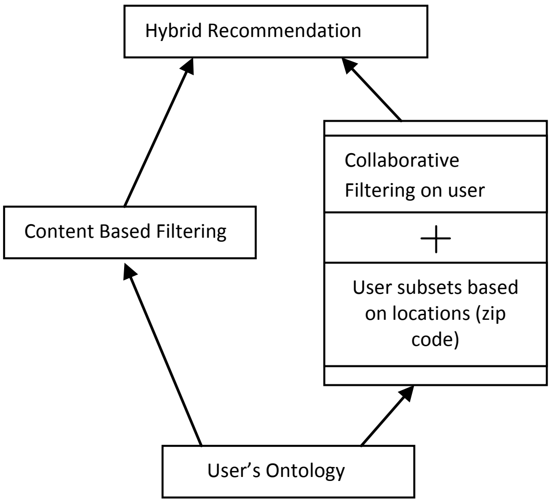

Hybrid Recommendation
Collaborative Filtering on user
+
User subsets based on locations (zip code)
Content Based Filtering
User's Ontology

Fig 4: Hybrid Recommendation System

3.1 Putting It All Together

In the hybrid recommendation approach, data is extracted from the user to match her choices to the preferences of the other users based on parameters such as ratings. If a similar pattern in the preferences is observed, the region to which the user belongs is reviewed based on her zip code. If a match in both the above cases are found, recommendations are made within that particular region, if not region is discarded and recommendation is made only based in ratings. If both the above two cases fail, recommendations are made based on popular demands or recent updated items in the dataset. The algorithm to run this recommendation system is broken up into two sections.

PART I: The initial section consists of the arrangements or basic recommendations that are possible even before the user logs in.

PART II: The second part of the recommendation system begins as soon as the user logs in. From this point hence recommendations will be specialized according to the requirements of the active user.

(I) 'PREPROCESSING ALGORITHM '
STEP A: Construction of Datasets.

- **(i)** The dataset 'Movie' consists of the list of movies with detailed metadata that are available with the recommendation system at the present.
- **(ii)** The dataset 'User_Info' consists of all the personal information of the registered user. This information is essential because it consists of user_id, which uniquely identifies each user and is used for specialized information. Also other information like user's **zipcode** is also important for grouping a user into a region. Email and Password from this table is used for user validation.
- **(iii)** The dataset 'Rate' is mainly used for checking to which group the particular logged in user belongs (establishing spatial correlation). This is used to locate the user to a group id and find users in the same or nearby range (location based services).
- **(iv)** The dataset 'User_Rate' is used to match the preference (here based on ratings of movies he watched) of the current user, with the preferences of other registered users.

STEP-B: Pre-recommendations for guest users

- **(v)** Display a list of top ten movies from each genre, as an attempt to make random recommendation.
- **(vi)** The pre-recommendations which are made are not based on any particular calculation, but on basic popular demands, or study of recent release of movies and so on. The pre-recommendations made are not user specific and consists of no user specialization.

(II) ALGORITHM FOR REGISTERED USERS.

STEP 1: User logs in and enters a query to search.

STEP 2: Check query with database *'Movie'*.

2.1 If (match==0) // query entered does not match with the exact name of any movie present in the dataset.

Then

2.1.1 Call the function '***Tokenization (Query)'***

Else

2.1.2 Select and Display data from the database **'*Movie'*.**

STEP 3: Make Recommendations to the user:

Recommendation is made to the user if even function tokenization (query) fails or the user has viewed some movie from the dataset and is expected to visit again. In both cases

3.1. Call function '*Recommend (string).'*

3.1 Display Result

FUNCTION TOKENIZATION (QUERY).

If the search query (e.g. movie name) which the user entered fails to match with the exact movie name, the string is broken down to tokens and each token is matched with subsets or sets of strings of movie names from the dataset 'Movie'. In case the user has entered only one alphabet, then that single alphabet is matched with all the sets and the alphabet itself is considered as a token. For e.g. if an user is searching for Toy Story and enters only Toy, all movie names with the subset word 'Toy' is displayed to the user.

FUNCTION RECOMMEND (STRING).

STEP A: Check whether user is a guest or new user or registered user from the database 'USER_INFO'. If an entry is found in the dataset the user is a registered user, if not she's a guest user.

CASE 1**:** User is a new user or guest.

ACTION 1: Display the pre-recommended movies to the user. This is usually the top 10 movies of a few popular genres.

ACTION 2: Redirect user to the subscription page to make future searches more optimized and user specific.

CASE 2: User is a registered user.

If the user is a registered user, recommendation can be more specialized. In our work we consider the rating pattern combined with its location with respect to other users to compute a similarity pattern and generate recommendations to the user. To compute this similarity, Geary's Index is used [2]:

Geary's Index: Geary's Index: The index measures the similarity of i's *(current user)* and j's attributes, c_{ij} which can be calculated as follows:

$$c_{ij} = (z_i - z_j)^2 \text{............ (Equation I)}$$

Here z_i and z_j are the values of the attributes of interest for object i and j. A location similarity w_{ij} (region) was used by Geary, where w_{ij} = 1 if *i* and *j* shared a common region and w_{ij} = 0 if they don't. Geary's index is expressed as follows

$$c = \frac{\sum_i \sum_j w_{ij} c_{ij}}{2 \sum_i \sum_j w_{ij} \sigma^2} \text{ .... (Equation II)}$$

Here σ^2 is the variance of the attribute z values, or

$$\sigma^2 = \frac{\sum_i (z_i - \bar{z})^2}{(n-1)} \text{ .... (Equation III)}$$

In the above equation $\bar{z}$ stands for

$$\bar{z} = \frac{\sum_i^n z_i}{n} \text{ .... (Equation IV)}$$

If the resultant value of 'c'=0, attributes are distributed independent of locations. If however the value for 'c'>0, similar attributes coincide within the region to which the user belongs.

CASE A: If(c==1)

If the value of 'c' is 1, it means that the ratings of the users in the current users' region are independent of her ratings. So the user is redirected to a page where the user is displayed recommendations based on users matching her ratings but not necessarily of her region.

CASE B: Else If(c < 1)

If the value of 'c' is less than 1, it means that the ratings of the current user have high correlation with the ratings of the users' in her region. So the user is redirected to a page where she is displayed recommendations based on matching from her region to which she has been grouped.

CASE C: In case where c > 1 indicating no correlation between ratings and regions, recommendations are based on basic assumptions of popular choices without considering regions or rate level preferences.

3. IMPLEMENTATION

Whether user is guest or registered is checked by extracting user's log in information using sessions and matching them to the dataset user_info.

Recommendation is not specialized for guest or visiting user. Once user logs in, her personal details are obtained and the process to optimize the search result begins. To improve accuracy, the user is grouped into certain regions by studying the zip code of the user. Users with nearby zip codes are grouped into one region. Once grouping of registered users is completed, we compute certain values by studying user's history. Some of these values are as follows:

i. Mean value of the ratings of movies that the current user has viewed or rated.

ii. Selected ratings of movies given by other users apart from the current user.

iii. Check whether the users ratings obtained from step (ii) are from users lying in the same region as that of the current user. This is done by extracting the current user's location by session and matching it with the locations of others extracted directly from the dataset.

iv. If regions match, subtract the mean value obtained in step (i) from the ratings of the users' belonging to the same region and perform a square of the value.

v. Now we consider a variable 'w' whose value would be 1, if the current user finds any other users in her own region, or 0 if no other user exists in her region.

vi. Once 'w' is obtained, we multiply the value with the value obtained in step (iv).

vii. We next compute the value for sigma which is square of the difference between last rated movies of the user with the other user ratings from her region

viii. Finally we compute Geary's index by using the above obtained values to find similar users with similar preferences.

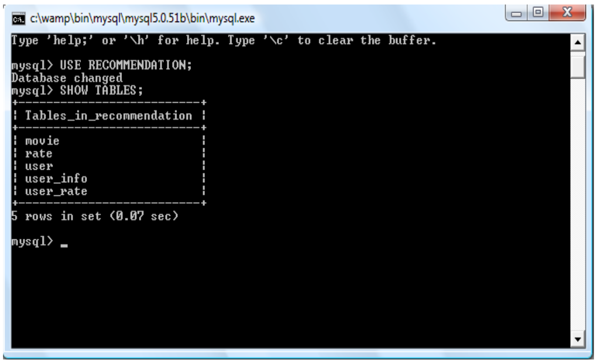
ix. Once obtained we now make the recommendations to the current user.

5. EXAMPLE USE CASES

To study the effect of the algorithm on experimental data, datasets are created. Datasets here is the collection of movies, user information, user preferences and their related information that are used in the pilot database to make recommendations to the users. Creation and manipulation of datasets has been done using MySql. In this project we will be using a '**Pilot Dataset'** extracted from the original dataset of 'MOVIELENS'.

Pilot Dataset: A pilot dataset is a small set of experimental data which may or may not be accurate to the real world information. Pilot dataset is used, instead of real life data sets in the earlier stages of project development because, due to the small size of the dataset it is easier to work with as manipulation and error detection in such sets are convenient.

Following are the datasets used to study the algorithm:



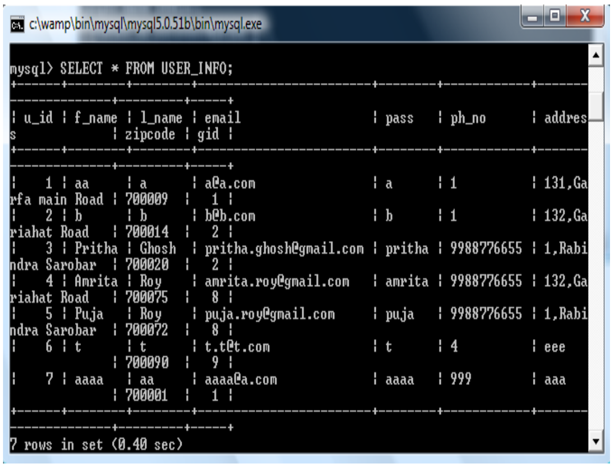
```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> USE RECOMMENDATION;
Database changed
mysql> SHOW TABLES;
+---------------------------+
| Tables_in_recommendation |
+---------------------------+
| movie                     |
| rate                      |
| user                      |
| user_info                 |
| user_rate                 |
+---------------------------+
5 rows in set (0.07 sec)

mysql> _
```

Snapshot of Table in Database Recommendation

A look into the table consisting of user's personal information:

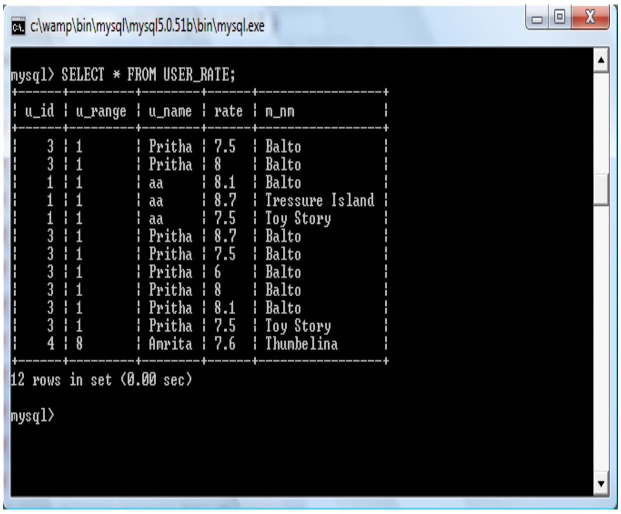


```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe

mysql> SELECT * FROM USER_INFO;
+------+--------+--------+------------------------+---------+------------+-------
---------------+---------+-----+
| u_id | f_name | l_name | email                  | pass    | ph_no      | addres
s                | zipcode | gid |
+------+--------+--------+------------------------+---------+------------+-------
---------------+---------+-----+
|    1 | aa     | a      | a@a.com                | a       | 1          | 131,Ga
rfa main Road | 700009  |   1 |
|    2 | b      | b      | b@b.com                | b       | 1          | 132,Ga
riahat Road    | 700014  |   2 |
|    3 | Pritha | Ghosh  | pritha.ghosh@gmail.com | pritha  | 9988776655 | 1,Rabi
ndra Sarobar   | 700020  |   2 |
|    4 | Amrita | Roy    | amrita.roy@gmail.com   | amrita  | 9988776655 | 132,Ga
riahat Road    | 700075  |   8 |
|    5 | Puja   | Roy    | puja.roy@gmail.com     | puja    | 9988776655 | 1,Rabi
ndra Sarobar   | 700072  |   8 |
|    6 | t      | t      | t.t@t.com              | t       | 4          | eee
               | 700090  |   9 |
|    7 | aaaa   | aa     | aaaa@a.com             | aaaa    | 999        | aaa
               | 700001  |   1 |
+------+--------+--------+------------------------+---------+------------+-------
---------------+---------+-----+
7 rows in set (0.40 sec)
```

Snapshot of Table USER_INFO (user's personal information)

A look into the table consisting of active user's movie preferences based on ratings of movies watched:



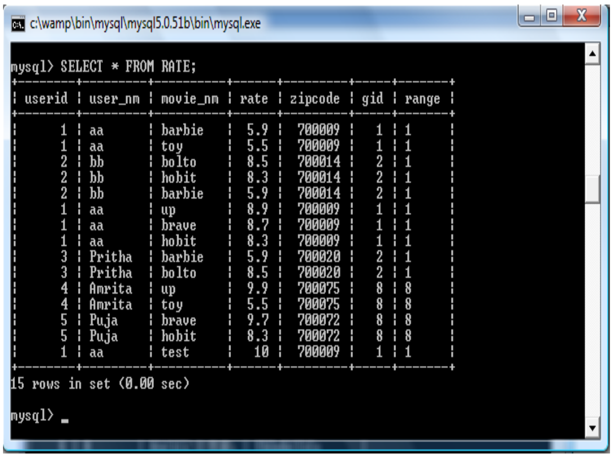
```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe

mysql> SELECT * FROM USER_RATE;
+------+---------+--------+------+-----------------+
| u_id | u_range | u_name | rate | m_nm            |
+------+---------+--------+------+-----------------+
|    3 | 1       | Pritha | 7.5  | Balto           |
|    3 | 1       | Pritha | 8    | Balto           |
|    1 | 1       | aa     | 8.1  | Balto           |
|    1 | 1       | aa     | 8.7  | Tressure Island |
|    1 | 1       | aa     | 7.5  | Toy Story       |
|    3 | 1       | Pritha | 8.7  | Balto           |
|    3 | 1       | Pritha | 7.5  | Balto           |
|    3 | 1       | Pritha | 6    | Balto           |
|    3 | 1       | Pritha | 8    | Balto           |
|    3 | 1       | Pritha | 8.1  | Balto           |
|    3 | 1       | Pritha | 7.5  | Toy Story       |
|    4 | 8       | Amrita | 7.6  | Thumbelina      |
+------+---------+--------+------+-----------------+
12 rows in set (0.00 sec)

mysql>
```

Snapshot of Table USER_RATE (shows user ratings to movies viewed):

A look into the table consisting of active user's location (zone or group to which the user belongs)



```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe

mysql> SELECT * FROM RATE;
+--------+---------+----------+------+---------+-----+-------+
| userid | user_nm | movie_nm | rate | zipcode | gid | range |
+--------+---------+----------+------+---------+-----+-------+
|      1 | aa      | barbie   |  5.9 |  700009 |   1 | 1     |
|      1 | aa      | toy      |  5.5 |  700009 |   1 | 1     |
|      2 | bb      | bolto    |  8.5 |  700014 |   2 | 1     |
|      2 | bb      | hobit    |  8.3 |  700014 |   2 | 1     |
|      2 | bb      | barbie   |  5.9 |  700014 |   2 | 1     |
|      1 | aa      | up       |  8.9 |  700009 |   1 | 1     |
|      1 | aa      | brave    |  8.7 |  700009 |   1 | 1     |
|      1 | aa      | hobit    |  8.3 |  700009 |   1 | 1     |
|      3 | Pritha  | barbie   |  5.9 |  700020 |   2 | 1     |
|      3 | Pritha  | bolto    |  8.5 |  700020 |   2 | 1     |
|      4 | Amrita  | up       |  9.9 |  700075 |   8 | 8     |
|      4 | Amrita  | toy      |  5.5 |  700075 |   8 | 8     |
|      5 | Puja    | brave    |  9.7 |  700072 |   8 | 8     |
|      5 | Puja    | hobit    |  8.3 |  700072 |   8 | 8     |
|      1 | aa      | test     |   10 |  700009 |   1 | 1     |
+--------+---------+----------+------+---------+-----+-------+
15 rows in set (0.00 sec)

mysql> _
```

Snapshot of Table RATE (used to extract the locality of user i.e to which zone the user belongs)

Once the datasets are successfully created we are ready to begin the execution of our algorithm.

(i)The first part of the algorithm provides recommendation for a user who is not registered with the site and has entered a search query either with movie name matching exactly with the dataset or the query words matching with some movie names of the dataset. In either case recommendations are made irrespective of the location or rating preferences due to lack of sufficient information about the visitor (Table 1).

Table 1: Recommendation by Tokenization for Visiting/Guest Users

Tressure	**Search**

Movie Name	**Year of Release**	**Cast**	**Genre**	**Rating (Out of 10)**
Tressure Island	1950	Bobby Driscoll,Rober	Adventure , Family	7.7

		t Newton,Basil Sydney		
Tressure Island	2012	Eddie Izzard,Toby Regbo	Adventure	6.4
National Tressure	2004	Nicolas Cage, Diane Kruger	Action, Adventure ,Mystery	6.9
Tressure Planet	2002	Emma Thompson,Joseph Gordon	Animation ,Adventure,Family	7.0
Beethoven's Tressure Tail	2014	Kristy Swanson, Bretten Manley	Family	4.5
The Tressure of the Sierra Madre	1948	Humphrey Bogart	Action,Adventure, Drama	8.3
Tressure of the Lost Lamp	1990	Allan Young,Terence McGovern	Animation ,Adventure,Comedy	7.2

(ii) In case the user is a newly registered user, the recommender system has no previous records on the user's preferences. So if the user query matches that to a dataset object, the result is retrieved and the list of additional recommendations provided for future interest is made by matching the rating of the item of her recent query, to the ratings of items visited or rated by other registered users (collaborative filtering) who might not necessarily belong to the active users' region. As a result movies with similar rating patterns are recommended from different zones (Table 2).

Table 2: Recommendation by Collaborative Filtering for Registered Active User.

Tressure

Search

Movie Name	Year of Release	Cast	Genre	Rating (Out of 10)
Tressure Island	1950	Bobby Driscoll,Robert Newton,Basil Sydney	Adventure , Family	7.7

Average rating of Customer: 8.11

User Id: 1

Group Id: 2

Items Recommended Based on Mean Ratings:

Movie Name	Year of Release	Cast	Genre	Rating (Out of 10)	User Id	Group Id
The Hobbit : The Desolation of Smaug	2013	Ian McKellen, Martin Freeman	Adventure , Fantasy	8.0	4	2
The Matrix	1999	Keanu Reeves, Laurence Fishburne	Action, Sci -Fi	8.7	2	1
The Tressure of the Sierra Madre	1948	Humphrey Bogart	Action,Adventure, Drama	8.3	2	1
Avatar	2009	Sam Worthington, Zoe Saldana	Action,Adventure,Fantasy	8.1	3	3
Inception	2010	Leonardo DiCaprio, Joseph Gordon -Levitt	Action, Mystery, Sci -Fi	8.8	6	2
The Dark Knight Rises	2012	Christian Bale, Tom Hardy	Action, Thriller	8.5	4	2
The Avengers	2012	Robert Downey Jr., Chris Evans	Action, Adventure, Sci-Fi	8.1	5	3

(iii) If the user is a registered user having other users sharing the same or nearby location, we consider the location based collaborative filtering with spatial autocorrelation (Table 3).

Average rating of Customer: 8.11

User Id: 1

Group Id: 2

Recommended Items Based on Mean Ratings:

Table 3: Recommendation Based on Location Based Collaborative Filtering for Registered Active Users

Movie Name	Year of Release	Cast	Genre	Rating (Out of 10)	User Id	Group Id
The Hobbit : The Desolation of Smaug	2013	Ian McKellen, Martin Freeman	Adventure, Fantasy	8.0	4	2
Inception	2010	Leonardo DiCaprio, Joseph Gordon-Levitt	Action, Mystery, Sci-Fi	8.8	6	2
The Dark Knight Rises	2012	Christian Bale, Tom Hardy	Action, Thriller	8.5	4	2

Since the value of c is greater than 0, we can predict that collaborative filtering can be made within the active users' region.

6. CONCLUSIONS

A novel hybrid recommendation system has been implemented which simultaneously addresses various issues like data sparsity and scalability while taking advantage of spatial autocorrelation in the underlying space. Future research on this subject would include scaling this to larger data sets and experiment with real data sets in other domains. Also an attempt would be made to check the working principality of our algorithm by making finer partitions of the location under consideration and improving our mapping techniques. Later studies would include reducing the time required to fetch the result once a query has been tokenized and a match has been found.

7. ACKNOWLEDGMENTS

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. I am grateful to the co author of this paper, Dr.Prosenjit Gupta, Professor, Department Of CSE, NIIT University for the guidance, inspiration and constructive suggestions that helpful to me in the preparation of this journal. I also thank IJCA who has given me the opportunity to publish this journal.

8. REFERENCES

[1] Burke, R.: Hybrid systems for personalized recommendations. In Mobasher, B., Anand, S.S., eds.:Intelligent Techniques in Web Personalisation. LNAI3169. Springer-Verlag (2005) 133–152.

[2] J. Das, S. Majumder, and P. Gupta. Voronoi based location aware collaborative filtering. In Proceedings of the 3rd IEEE Conference on Emerging Trends and Applications in Computer Science (NCETACS), pages 179{183, 2012}.

[3] P.Gupta, ―Privacy Enhancing Collaborative Social Display Advertising,‖ Workshop on Business Applications of Social Network Analysis, December 2010.

[4] .Bruke, ―Hybrid Recommender Systems: Survey and Experiments‖, User Modeling and User-Adapted Interaction, November 2002.

[5] T.Tran, R.Cohen, ―Hybrid Recommender Systems for Electronic Commerce‖, Knowledge-Based Electronic Markets, 2000.

[6] Basu, C., H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation systems. Papers from 1998 Workshop. Technical Report WS-98-08. AAAI Press, 1998.

[7] P. Melville, R. J. Mooney, and R. Nagarajan. ―Content Boosted Collaborative Filtering for Improved Recommendations, In Proceeding of the 2001 SIGIR Workshop on Recommender Systems, 2001.

[8] Brunato, M., Battiti, R., Villani, A. and Delai, A. A Location-Dependent Recommender System for the web. *Technical report DIT-02-0093, Universita` di Trento* , 2002.

[9] Tobler, W. R. 1970. A computer movie simulating urban growth in the Detroit region. Econ. Geogr. 46: 234 240.

[10] Lo, C.P., Yeung, A.K.W, *Concepts and Techniques of Geographic Information Systems*, Prentice Hall, 2007.

[11] M.J. Ekstrand, J.T. Riedl and J.A. Konstan. CollaborativeFiltering Recommender Systems in Foundations and Trends in Human-Computer Interaction, Vol. 4 (2), 2010, pp. 81-173.

[12] A. Dalmia, J. Das, P. Gupta, S. Majumder, D. Dutta. Scalable Hierarchical Recommendations Using Spatial Autocorrelation, Proceedings, Third ASE International Conference on Big Data Science and Computing, Beijing, China (BigDataScience 2014), August, 2014.

[13] J. Das, S. Majumder, D. Dutta, P.Gupta. Iterative Use of Weighted Voronoi Diagrams to Improve Scalability in Recommender Systems Proceedings, PAKDD 2015, Ho Chi Minh City, Vietnam, Springer Verlag LNAI Vol. 9017 pp. 605—617, May 2015.

[14] Terveen, Loren; Hill, Will (2001). "Beyond Recommender Systems: Helping People Help Each Other"Addison-Wesley. p.6. Retrieved 16 January 2012 .