Middle-East Journal of Scientific Research 23 (10): 2592-2602, 2015
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2015.23.10.22795

Recommendation System for the Long Tail Problem Using Factorization Through Latent Dirichlet Allocation

T.K. Thivakaran and R. Nedunchelian

Department of Computer Science and Engineering,
Sri Venkateswara College of Engineering, Sriperumnudur, Tamilnadu, India

Abstract: Long Tail Problem is one major issue in providing effective recommendation system, since long tail problem have only a few user ratings. The proposed system solves the issue by clustering tht identified items according to their popularity. It is identified that tail items are clustered based on the ratings of clustered groups while the head items are based on the ratings of individual item or group. This method is applied to movie lens data sets and the results are compared with those of the non grouping and fully grouped methods in terms of recommendation accuracy and scalability. The results show that implementing factorization through latent dirichlet allocation considerably reduces the recommendation error rates for the tail items by maintaining reasonable computational performance.

Key words: Adaptive Clustering · Recommender Systems · Long Tail Problem · Latent Dirichlet Allocation · Matrix factorization

INTRODUCTION

Long Tail Problem: The term long tail has gained popularity in recent times as describing the retailing strategy of selling a large number of unique items with relatively small quantities sold of each-usually in addition to selling fewer popular items in large quantities. Anderson elaborated the concept of long tail. The distribution and inventory costs of businesses successfully applying this strategy allow them to realize significant profit out of selling small volumes of hard –to-find items to many customers instead of only selling large volumes of a reduced number of popular items. The total sale of this large number of "non-hit items" is called "the long tail". The long tail concept has found some ground for application, research and experimentation. It is a term used in online business, mass media, micro-finance, user-driven innovation and social network mechanisms economic models, marketing a frequency distribution with a long tail has been studied by statisticians since at least 1946.The term has also been used in the finance and insurance business for many years [1].

Objective:

- To address the long Tail Recommendation Problem (LTRP) in the recommendation system in order to reduce the error rate.
- To overcome the problem of insufficiency of tail items in Each Item (EI) method and to overcome the problem of excessive items of head items in Total Clustering (TC) method Factorization through Latent Dirchlet allocation technique is proposed and implemented.

Clustering: Clustering can be considered the most important unsupervised learning problem so as every other problem of this kind it deals with finding a structure in a collection of unlabeled data. A loose definition of clustering could be the process of organizing objects into groups whose members are similar in some way. A cluster is therefore a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters.

Recommended Systems: Seek to predict the 'rating' or 'preference' that user would give to an item such as music, books, or movies or social element they had not

Corresponding Author: T.K. Thivakaran, Department of Computer Science and Engineering,
Sri Venkateswara College of Engineering, Sriperumnudur, Tamilnadu, India.

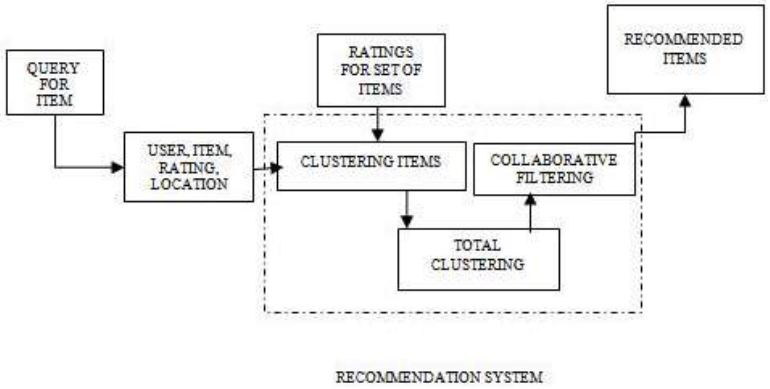

Fig. 2.1: Workflow of Existing Recommender System

yet considered, using a model built from the characteristics of an item content based approaches or the user's social environment collaborative filtering approaches. Recommender systems have become extremely common in recent years. Fig. 2.1 refers the workflow diagram of the recommender system.

Adaptive Clustering: Adaptive Clustering method clusters the item with other similar items when it has only small amount of data, but groups to lesser extent or does not group when it has considerable amount of data.

The main features of adaptive clustering are

- It provides only the necessary amount of data to both the head and tail items and does not exceed this amount.
- It builds predictive rating models for Each Item method(EI)(unlike(TC) Total Clustering method) by providing more data to the tail items(unlike the EI method)
- One item can be clustered into several different groups redundantly as similar items [2].

Problem Statement: Many recommended systems ignore unpopular or newly introduced items, having only a few ratings and focusing on those items with enough ratings to be of real use in the recommendation algorithms. Alternatively, such unpopular or newly introduced items can remain in the system but require special handling.work flow of existing system architecture is given below.

Existing System Architecture

Methods in Existing System: Two conventional recommender systems are the each item (EI) method and the total clustering (TC) method.

EI Method: The EI method builds rating-predictive models for each individual item, thus resulting models are highly customized for EI.

Total Clustering (TC) Method: To solve the long tail recommendation problem (LTRP) that arises in the EI method, we next cluster the whole item set into different groups and build rating-predictive models for the resulting group using more data than with the EI. We call this method the TC recommendation method. The error rates of the TC method are significantly lower than the basic EI method in many cases, especially for the items in the long tail. In other words, the TC method does not have the LTRP. However, the TC method has the limitation that overly increases the data size for the head items, which already have an adequate amount of data. This excessive data impede the scalability of the TC method without significant performance improvement in the head [3].

Drawbacks:

- EI method often has the long tail problem because of lack of data to build good predictivemodels in tail items.
- TC method has the limitation that overly increases the data size for the head items, whichalready have an adequate amount of data.

Adaptive Clustering: A new recommender method called the adaptive clustering (AC) recommendation method that adaptively groups items according to their popularities. Popularity is defined as the number of ratings provided by customers for that item. In other words, if the item has only a small amount of data, then the AC method clusters it with other similar items more intensively; on the other hand, if it has a large amount of data, then the AC method clusters it to a much lesser extent. The suggested AC

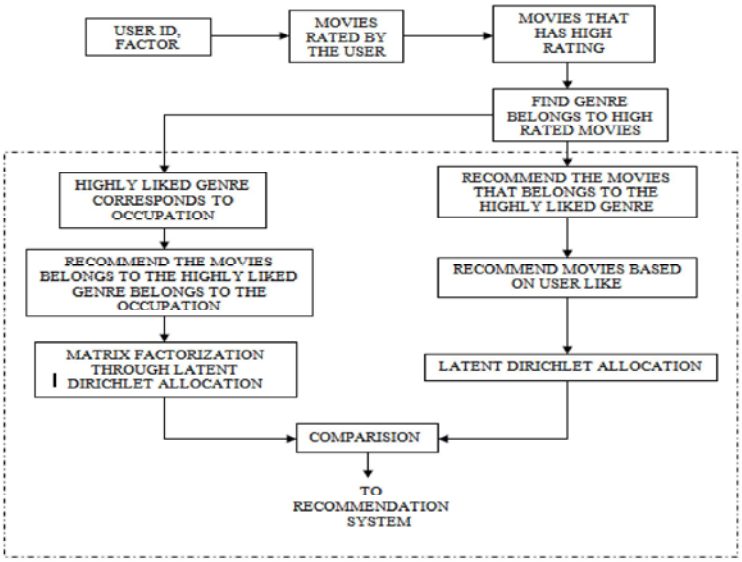

Fig. 3.1: Work Flow of Proposed Recommender System

method solves the LTRP in the sense that the error rates of the items, especially in the tail, are significantly lower than those for the basic EI method without overly increasing the amount of data for the head items. Moreover, the AC method builds a more customized predictive rating model than EI and TC method [4].

Proposed System

Latent Dirichlet Allocation: Latent Dirichlet allocation (LDA) is a generative probabilistic model for collections of discrete data such as text. LDA is a three level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document. The report results in document modeling, text classification and collaborative filtering, comparing to a mixture of unigrams model and the probabilistic model.LDA overcomes both of these problems by treating the topic mixture weights as a k-parameter hidden random variablerather than a large set of individual parameters it is also intended to be illustrative of the way in which probabilistic models can be scaled up to provide useful inferential machinery in domains involving multiple levels of structure [5].

As a probabilistic module LDA can be readily embedded in a more complex model LDA comes from allowing mixtures of Dirichlet distributions in the place of the single Dirichlet of LDA. Finally, a variety of extensions of LDA can be considered in which the distributions on the topic variables are elaborated. For example, arrange the topics in a time series, essentially relaxing the full exchangeability assumption to one of partial exchangeability. Also consider partially exchangeable models in which the condition on exogenous variables thus, for example, the topic distribution could be conditioned on features such as "paragraph" or "sentence," providing a more powerful text model that makes use of information obtained from a parser [6].

Architecture

Matrix Factorization Through Latent Dirichlet Allocation: (FLDA) Matrix factorization through LDA is a method to predict ratings in recommender system applications where a bag of words representation for item meta-data is natural. Such scenarios are commonplace in web applications like content recommendation, ad targeting and web search where items are articles, ads and web pages respectively. Because of data sparseness, regularization is key to good predictive accuracy. Our method works by regularizing both user and item factors simultaneously through user features and the bag of words associated with each item. Specially, each word in an item is associated with a discrete latent factor often referred to as the topic of the word item; they are obtained by averaging topics across all words in an item. To avoid over fitting, user and item factors are regularized through Gaussian linear regression and Latent Dirichlet Allocation (LDA) priors respectively. As a by-product, fLDA also identifies interesting topics that explains user item interactions. The method also generalizes a recently

proposed technique called supervised LDA to collaborative filtering applications. While estimates item topic vectors in a supervised fashion for a single regression, fLDA incorporates multiple regressions (one for each user) in estimating the item factors. The key idea of our method is to let the user factors take values in an Euclidean space as in existing factorization models, but assign item factors through a richer prior based on Latent Dirichlet Allocation (LDA) [7].

Each Item Method (EI)

Description: The EI recommendation method builds data mining models for each individual item i in Item to estimate unspecified ratings for i. In other words, the EI method does not group an item with the other similar items at all and builds predictive models only by using the data in EI. For example, in the case of the Movie Lens data set, the EI method builds a predictive model for each of the 841 movies using the ratings of each particular movie. The main problem with the EI recommendation method is that only a few ratings are available in the long tail, so the predictive models for the tail items are learned from only a few training examples using the EI method. Fig 4.1 shows the rating prediction model for the targeted movie and user which represents the user defined collaborative count to predict the similarity and neighbor items to provide original rating.

Algorithm

Input:

Target Movie
Target User
CF Count – user defined

Output:

Predicted Rating
Original Rating
Long tail

Process:

- Select co-rated users
- Select the rating (R_{it}) for the target movie (t) provided by the co-rated users
- Select the rating (R_{ir}) for the remaining movies (r) provided by the co-rated users (m)
- Calculate similarity as follows

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

where

- R_{it} is the rating of the target item t by user i,
- $R_{ir}$is the rating of the remaining item r by user i and
- M is the number of all rating users to the item t and item r.
- Sim (t, r) – similarity between target item and remaining item
- The rating of the target user u to the target item t is as following:

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

where

- R_{ui} is the rating of the target user u to the neighbour item i,
- $sim_{(t,i)}$ is the similarity of the target item t and the neighbour it user i for all the co-rated items and
- m is the number of all rating users to the item t and item r.
- Put – Predicted rating for the target item for the target user
- Neighbor item – high similar item to target item

Screen Shots: Fig. 4.1 shows the rating for the targeted movie and user by setting the collaborative count which user can prefer from the predictive model for similar and neighbor items is calculated to predict the original rating and similarly Fig 4.2 shows the reduction of long tail with itemrate in x axis and frequency of popularity in yaxis.

Graph

Total Clustering Method (TC): The LTRP problem is caused by a lack of data to build good predictive models in the tail and therefore, clustering items can be a reasonable solution. The TC recommendation method

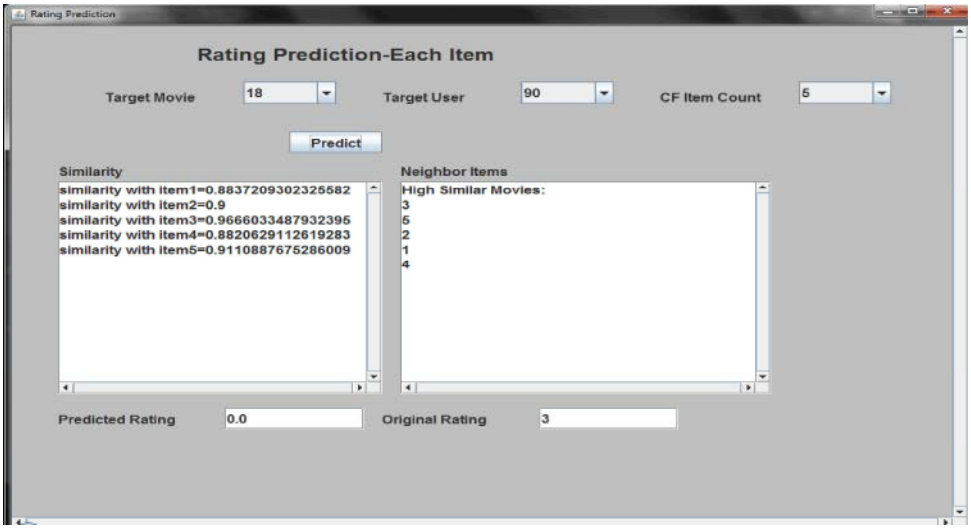

Fig. 4.1: Each Item Method

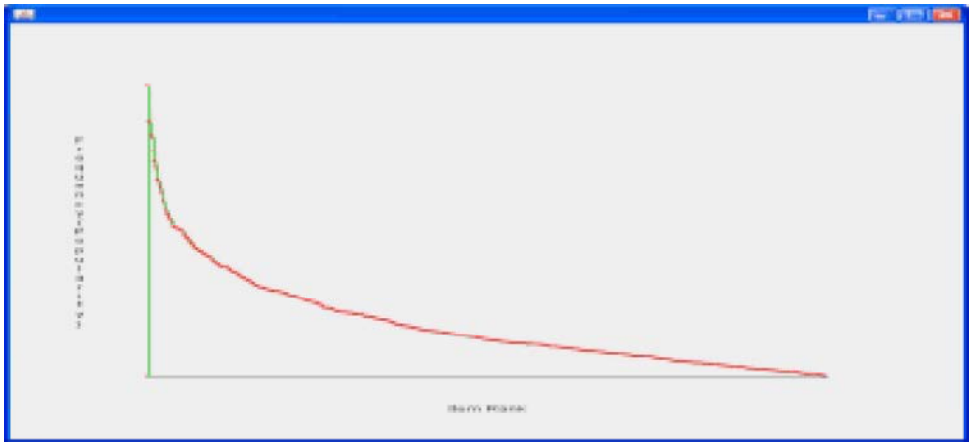

Fig. 4.2: Graphs for Each Item Method

clusters the whole item set I into different groups by applying conventional clustering methods such as k-means clustering and building rating-predictive models for each resulting group. For example, in the case of the Movie Lens data set, the TC method clusters 841 movies into groups using the k-means clustering method and builds a predictive model [8].

It is concluded that the TC method can significantly outperform the EI method by providing more data to tail items. However, it attains this achievement inefficiently by providing an excessive amount of training data to the head items without performance improvements. Fig 5.1 shows the predicted rating for the targeted movie and user by setting the collaborative filtering count constant to provide original rating from the rating prediction model and Fig 5.2 shows the required graph for the total clustering method.

Algorithm:

Input**:**

Target Movie

Target User

CF Count – 100

Output:

Predicted Rating

Original Rating

Reduced Long tail compared to Each Item

Process**:**

- Select co-rated users
- Select the rating (R_{it}) for the target movie (t) provided by the co-rated users
- Select the rating (R_{ir}) for the remaining movies (r) provided by the co-rated users (m)
- Calculate similarity as followed by the steps in the Each Item method.

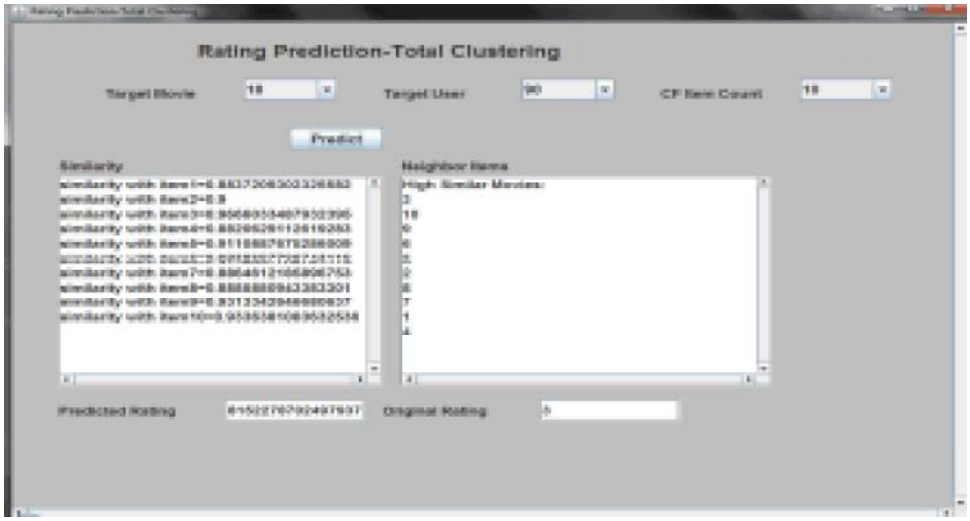

Fig. 5.1: Total Clustering Method

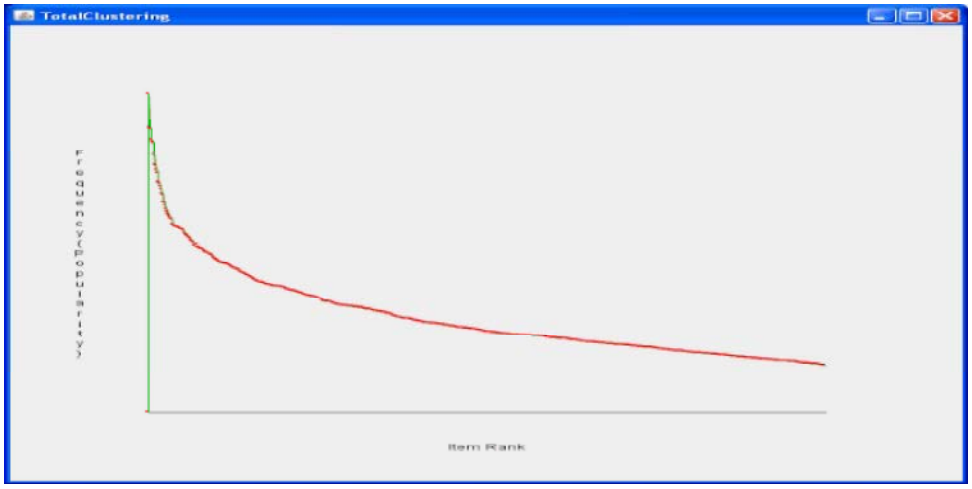

Fig. 5.2: Graphs for Total Clustering Method

Screen Shots

Graph

Adaptive Clustering (AC): A new recommender system called the AC method that clusters items according to their popularities. The suggested method clusters an item with other similar items one by one until the resulting group size reaches the criterion number of rating α.

The AC method clusters the item with other similar items when it has only a small amount of data, but groups to a lesser extent or does not group at all when it has a considerable amount of data. In the case of the Movie Lens data set, all movies from that data set are ordered based on the popularity for each movie. Then, the popularity of each movie is compared with the criterion number of ratings α [9-11].

If it is larger than α, then the AC method does not apply any clustering method; instead, it keeps the basic EI approach. However, if it is smaller than α, then the AC method clusters the movie with other similar movies one by one until the resulting group size reaches α. After that, the AC method builds rating predictive models using the resulting group for EI.

Fig. 6.1 represents the rating for the targeted movie and user by setting a criterion number to predict the original rating from the predictive model and Fig 6.2 shows the required graph for the adaptive clustering technique then Table 1 shows the comparison about the three different methods and represents the count of the popular and unpopular items and Table 2 represents how the popularity and rank is increased when compared it to the existing methods.

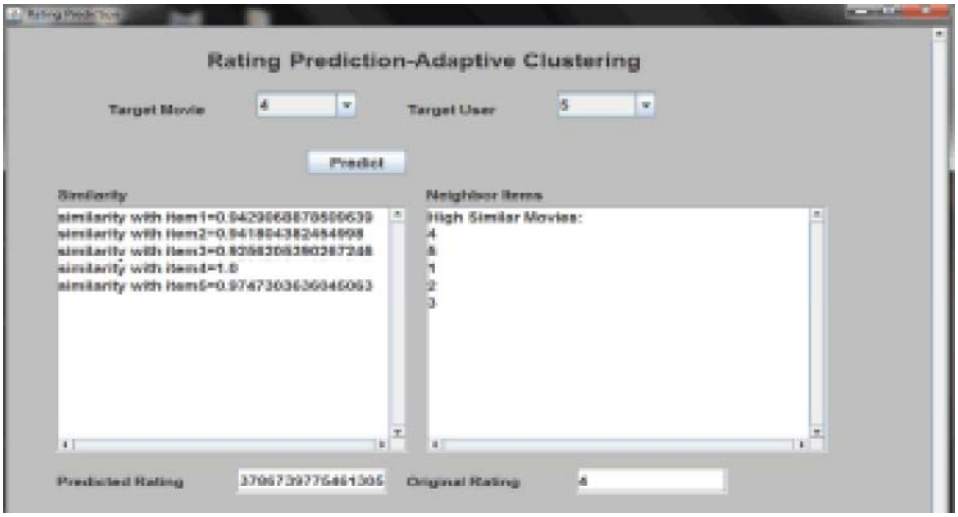

Fig. 6.1: Adaptive Clustering Method

Fig. 6.2: Graphs for Adaptive Clustering Method

Table 2: Comparison Table For Three Methods (Performance Evaluation)

Popularity Table											
Adaptive Clustering Method				Latent Dirichlet Allocation				Matrix Factorization			
Movieid	Popularity	Rank	Category	Movieid	Popularity	Rank	Category	Movieid	Popularity	Rank	Category
210	456	24	popular	210	556	24	popular	210	656	24	popular
211	283	300	unpopular	211	381	305	popular	211	481	305	popular
212	192	352	popular	212	292	352	unpopular	212	392	352	unpopular
213	329	145	popular	213	409	194	popular	213	509	194	popular
	294	254	unpopular	214	389	280	popular	214	489	280	popular
215	287	284	unpopular	215	386	290	popular	215	486	290	popular
	342	117	popular	216	465	57	popular	216	565	57	popular
	295	245	unpopular	217	395	255	unpopular	217	495	255	popular
	346	108	unpopular	218	446	97	popular	218	546	97	popular
219	286	287	popular	219	386	290	popular	219	486	290	popular

Table 1: Comparison Table

Algorithm	No of Movies	Category	Popular	Unpopular
Each Item Method(EI)	1048	50	32	809
Total Clustering(TC)	1048	50	117	724
Adaptive Clustering(AC)	1048	50	221	620
Latent Dirichlet Allocation(LDA)	1048	50	324	517
Matrix Factorization (MF)	1048	50	838	4

Algorithm:

Input**:**
Target Movie
Target User
CF Count – 100 (popularity < 200)
CF Count – 75 (popularity > 200 &&< 400)
CF Count – 50 (popularity > 400 &&< 600)
CF Count – 25 (popularity > 600)

Output**:**

Predicted Rating
Original Rating
Reduced Long tail compared to Each Item and Total Clustering

Process:

- Select co-rated users
- Select the rating (Rit) for the target movie (t) provided by the co-rated users
- Select the rating (Rir) for the remaining movies (r) provided by the co-rated users (m)
- Calculate similarity as followed by the steps in the Total Clustering Method.

Screen Shots: Fig. 6.1 shows the original rating of the targeted movie and user by setting the criterion number to find the original rating from the predictive model. Fig 8 shows the perfect reduction of tail to solve the problem.

Graph

Latent Dirichlet Allocation (LDA): Proposed Latent Dirichlet allocation(LDA), algorithm is implemented for collections of discrete data such as movie.. The central goal of a title is to provide a thematic summary" of a collection of set of movies. A collection of news articles could discuss e.g. stuntl, romance and business related themes. The central goal of title modelling is to automatically discover the title from a collection of movies.

Algorithm:

Input:
user ID
Output:
Recommended Movies based on user like
Reduced Long tail compared to adaptive clustering

Process:

1. Select the movies rated by the user
2. Find the movies that has high rating
3. Find the genre belongs to the high rated movies
4. Find the genre that encountered frequently – highly liked genre
5. Recommend the movies that belongs to the highly liked genre

Screen Shots

Graph

Matrix Factorization (FLDA): The fLDA (factorization through latent dirichlet allocation), a novel matrix factorization method to predict ratings in recommender system Specially, each word in an item is associated with a discrete latent factor often referred to as the title of the movies.Movie titlesare obtained by averaging title across all words in an item. Then user rating on an item is modeled as user's affinity to the item's title where user affinity to title (user factors) and topic assignments to words in items (item factors) are learned jointly in a supervised fashion. As a byproduct, fLDA also identifies interesting titles that explains user-item interactions [12-17].

Algorithm:

Input:
User ID + Occupation
Output:
Recommended Movies based on user like + factor

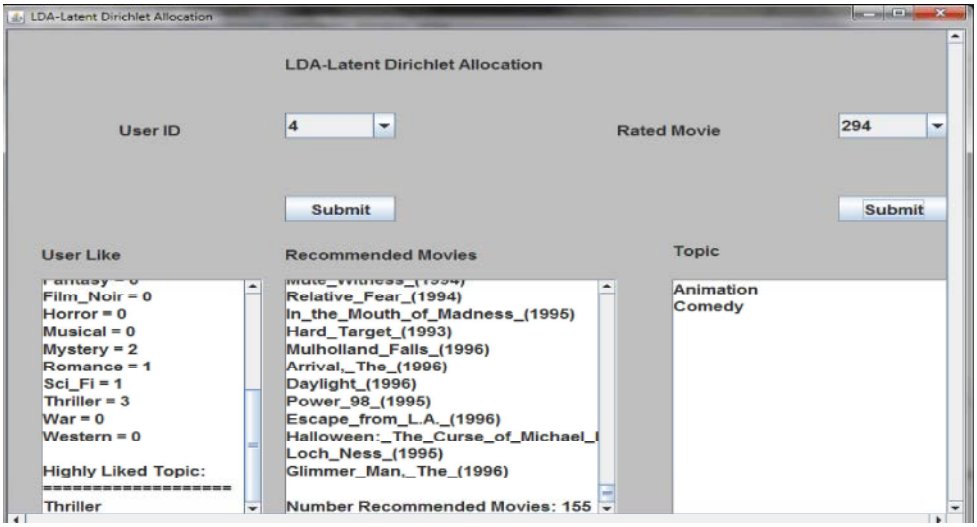

Fig. 7.1: LatentDirichlet Allocation

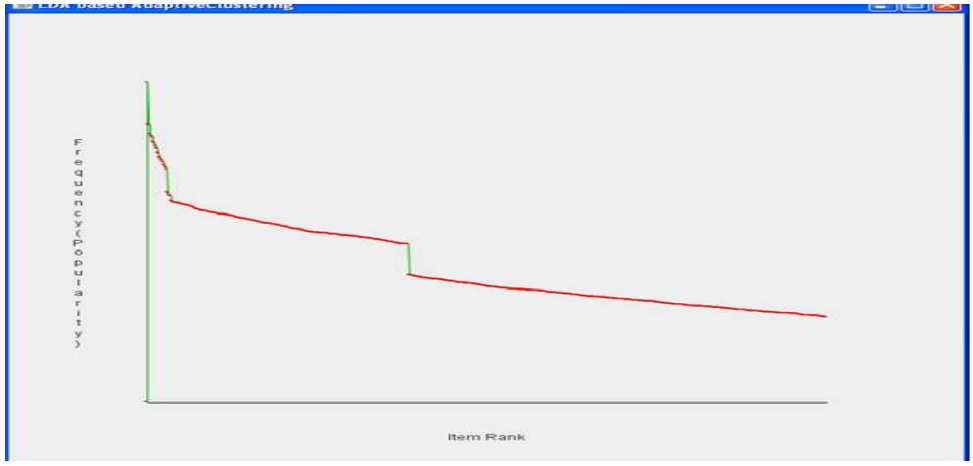

Fig. 7.2: Graph For Latent Dirichlet Allocation

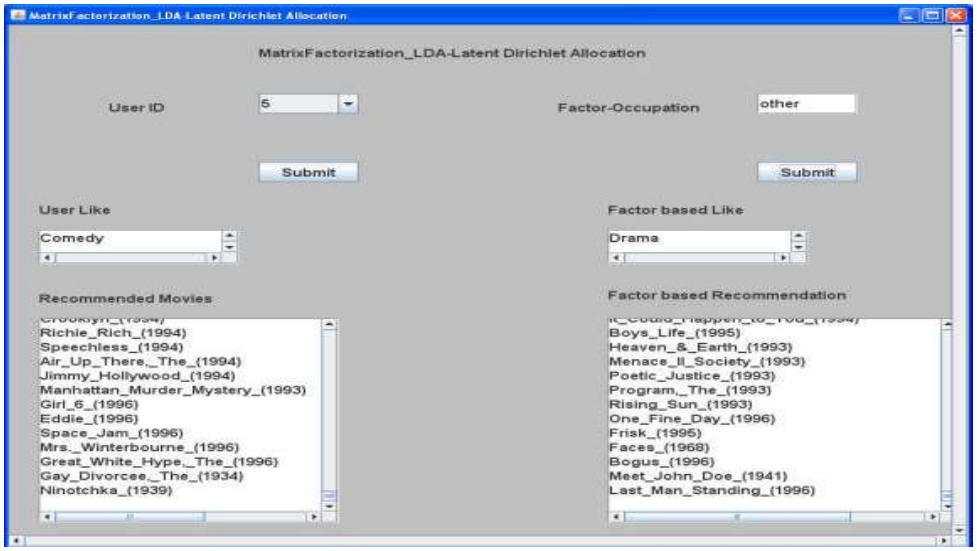

Fig. 8.1: Matrix Factorization through Latent Dirichlet Allocation

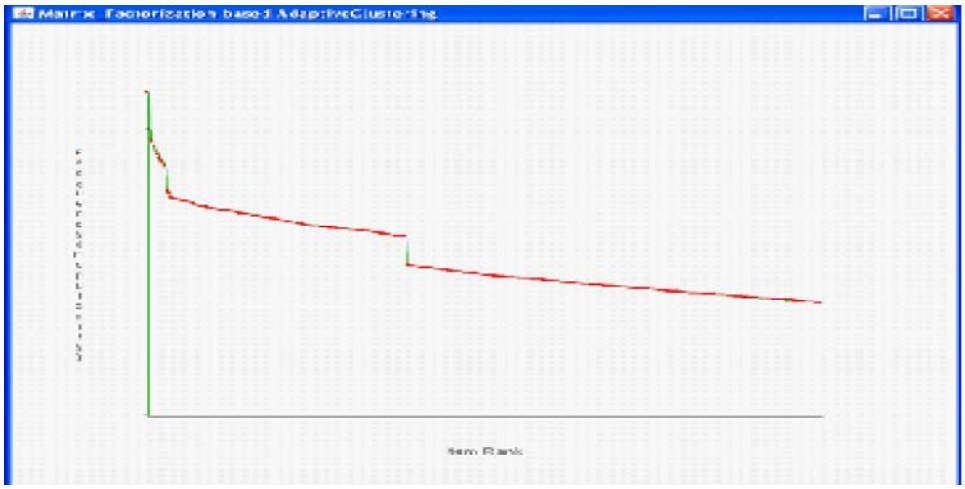

Fig. 8.2: Graph for Matrix Factorization through Latent Dirichlet Allocation

Process:

1. Select the movies rated by the user
2. Find the movies that has high rating
3. Find the genre belongs to the high rated movies
4. Find the genre that encountered frequently – highly liked genre
5. Recommend the movies that belongs to the highly liked genre
6. Select the factor – Occupation
7. Find the genre that encountered frequently for the occupation – highly liked genre corresponds to occupation
8. Recommend the movies that belongs to the highly liked genre corresponds to occupation

Screenshots

Graph: From the above Table 1 it is inferred that Matrix Factorization reduces the tail portion of long tail by bringing the unpopular to popular side. Sample of data of ten values is shown in the Table 2 from Table 1.

CONCLUSION

The proposed work deals with the Long Tail Recommendation Problem (LTRP) responsible for improving the error rates in the tail of the item distribution of the recommendation system. This issue is overcome by using factorization through latent dirichlet allocation. In this technique, the recommender system considers both the user likes genre and occupation factor recommending movies belonging to a certain genre corresponds to the respective occupation. This technique when implemented in movielens dataset results in reduction of tail part. This means that unpopular items are reduced by considering both the user liked genre and occupation factor so that more number of movies are recommended and the tail part is reduced with increase in ratings.

The future work would include the addition of user location as feature in order to provide location based recommendation for further reduction in tail problem.

REFERENCES

1. Linden, G., B. Smith and J. York, 2003. Amazon.com Recommendations: Item-to-Item Collaborative Filtering, IEEE Internet Computing, 7(1): 76-80.
2. Basu C., H. Hirsh and W. Cohen, 1998. Recommendation as Classification: Using Social and Content Based Information in Recommendation, Proc. Nat'lConf. Artificial Intelligence (AAAI '98), 714-720.
3. Bell, R.M. and K. Yehuda, 2007. Improved Neighborhood-based Collaborative Filtering, Proc. KDD Cup Workshop, pp: 7-14.
4. Hervas-Drane, A., 2007. Word of Mouth and Recommender Systems: A Theory of the Long Tail, working paper, Harvard Business School.
5. Park, Y.J. and A. Tuzhilin, 2008. The Long Tail of Recommender Systems and How to Leverage It, Proc. ACM Conf. Recommender Systems, pp: 11-18.
6. Truong, K.Q., F. Ishikawa and S. Honiden, 2007. Improving Accuracy of Recommender Systems by Item Clustering,IEICE Trans. Information and Systems, 90(9): 1363-1373.

7. Song Jie Gong, 2009. A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering, pp: 697-712.
8. Iba, W. and P. Langley, 1992. Induction of One-Level Decision Trees, Proc. Ninth Int'l Conf. Machine Learning, pp: 233-240.
9. Agarwal, D. and B.C. Chen, 2010. flda: matrix factorization through latent dirichlet allocation. In WSDM, pp: 91-100.
10. Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In KDD, pp: 426-434.
11. Wang, F., S. Ma, L. Yang and T. Li, 2006. Recommendation on item graphs. In ICDM, pp: 1119-1123.
12. Hervas-Drane, A., 2007. Word of Mouth and Recommender Systems: A Theory of the Long Tail, working paper, Harvard Business School.
13. Anderson, C., 2006. The Long Tail. Hyperion Press.
14. Truong K.Q., F. Ishikawa and S. Honiden, 2007. Improving Accuracy of Recommender Systems by Item Clustering, IEICE Trans Information and Systems, 90(9): 1363-1373.
15. Ungar, L.H. and D.P. Foster, 1998. Clustering Methods for Collaborative Filtering, Proc. Workshop Recommendation Systems.
16. Witten, I.H. and E. Frank, 2005. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann.
17. Jeyasree, S. and T.K. Thivakaran, 2014. Precise Recommendation System for the Long Tail Problem Using Adaptive Clustering Technique, International Journal of Innovative Research in Computer & Communication Engineering, 2(4).