Research Journal of Applied Sciences, Engineering and Technology 10(7): 812-815, 2015
DOI:10.19026/rjaset.10.2434
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: February 16, 2015 Accepted: March 7, 2015 Published: July 10, 2015

Research Article

Inspecting the Proficiency of Novel Algorithms on Sparse Data Domains for Efficient Recommendation-A Glance

Chandni Suresh, B. Krishya Yedugiri, P. Sini Raj and Souparnika Sreedhar
Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham (Amrita University), Ettimadai, India

Abstract: Recommender systems have played a major role in almost all the domains today where human interaction happens with system. Depending on the user's choice, a recommender system presents some promising suggestions by observing all the activities of the user on the web and thus, helps to find out similar users and interested products. All the ratings provided by the user is stored in a rating matrix. Sometimes it so happens that the users may view the item, but not always rate it; which makes the dataset sparse. Performing operations on such sparse datasets by recommender engines may not give precise suggestions to the user. This study aims to make such sparse datasets denser by applying the two novel methods, FST and UTAOS; and thereby implementing any of the collaborative filtering techniques upon it to showcase the efficiency in recommendation. Results reveal that FST outperforms the UTAOS approach in terms of sparse datasets which paves the way for an efficient recommendation.

Keywords: Feature subspace transfer, recommender engine, sparse datasets, user's tree accessed on subspaces

INTRODUCTION

Nowadays enterprises benefit a lot from the information received from the people who are using a certain system. They can know about the personal tastes of the users and recommend products, which help them to grow business. There are two major ways to recommend an item to a user. One is by using Content based and other is Collaborative based filtering methods (Li *et al*., 2009; Han and Kamber, 2008). To perform these methods, we need to introduce the rating matrix. Rating matrix is basically the cluster of users and the items they have rated for, represented in a table format. The ample number of RS available make predictions based on the rating matrix (Wang and Ke, 2014).

Coming on to the features of a rating matrix, there are two kinds. Item based and user based rating matrix. In item based rating matrix, the recommendation happens by considering the rating of items and combination of items in it. In user based rating matrix, user preferences are a significant factor that decides how to make the recommendation.

Speaking of which, Item rating matrix is important for content based recommendation method as it analyses a set of features of a particular item, which had highest ratings; and using these features, matches similar items having the same features. This is how content based filtering works. On the other hand, collaborative filtering recommendation technique focuses on the user than the item and finds out users who have similar tastes.

Well, the basic drive behind writing this study is still not visited. It is the data sparsity issue that needs to be brought to light. A lot of datasets are available which are sparse or have ambiguity in their values. It may arise due to a variety of factors like incorrect recording of data, failing to capture results carefully etc. But our concern is on the rating matrices on which we must perform operations. If we divert our attention to a few data sets, it is observed that most users have reviewed the item but not rated it. This phenomenon that leads to sparse data sets is called as "cold start" problem (Li *et al*., 2009; Wang and Ke, 2014). The users who do so i.e., review the item and do not rate it are called the "cold start users". This could only lead to more number of missing values which significantly reduces the efficiency of results and incorrect output is generated.

LITERATURE REVIEW

While developing this study we had read through many papers which were related to sparse problem that is now becoming an evident factor to be considered during the data mining process.

Coming to a conclusion, we observed that a method could reduce the sparsity issue significantly. This is by finding out who among the users share similar interests. By doing so, we can easily predict the missing value in a matrix with the help of ratings given

Corresponding Author: Chandni Suresh, Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham (Amrita University), Ettimadai, India

by other users for the common items. Operations can be performed to find out the neighbour users and what is their ratings. Once this is known, algorithms are applied over the dataset to fill in with real values and hence data sparsity issue is alleviated.

Another interesting way was through Transfer Learning Techniques (Li *et al*., 2009). Transfer learning techniques have been used from a very long while, to induce the transfer of useful knowledge from a cluster of information. Basically transfer learning can be extremely helpful in the case of sparse data sets as the rating matrix that we know may not have all the values. Since we cannot just randomly choose some numeric value and fill in the missing places, we can check in some useful knowledge that may be present elsewhere. So it is the process of linking two rating matrices with a thread of useful knowledge in order to complete the matrix which is sparse (Pan *et al*., 2010).

Our aim is to do a comparison of the above said techniques that alleviate missing value problems in data sets. Under thorough analysis and the implementation of these techniques on the Movie Lens data set, we have achieved results which will be presented shortly. The upcoming topics will discuss on these techniques in detail and describe the algorithm we have chosen in each case.

MATERIALS AND METHODS

The two major algorithms that we put in a balance are UTAOS (Ramezani *et al*., 2014) and FST (Wang and Ke, 2014). UTAOS algorithms stands for User's Tree Accessed on Subspaces. As the name sounds its major functionality revolves around users and their ratings. FST is Feature Subspace Transfer. Both deals with integrating subspaces and sharing knowledge from them to use on other matrix.

A flowchart can demonstrate the aim of the paper. We have a dataset which is sparse and is used for both these algorithms. For now, we chose the Movie Lens dataset. This dataset it taken and both algorithm are performed on it individually. After the dataset is retrieved by applying UTAOS and FST, a recommendation method is applied and finally they both are compared on the basis of MAE. Figure 1 below will explain you the process.

Our query is not to find out which algorithm works best for recommendation, but it is focused on the aim to find out which algorithm can make a sparse data set denser. Whether it is UTAOS or FST, the results can be seen later.

Implementation of algorithms:

UTAOS: The implementation of the UTAOS algorithm is simple. But there are some concepts to be noted here. We know that the aim of CF is to suggest items to its active users depending on its neighbours, as we have discussed above. Hence to do this there are two ways one by using some similarity measures like PCC (Pearson Correlation Coefficient) and NCS (New user

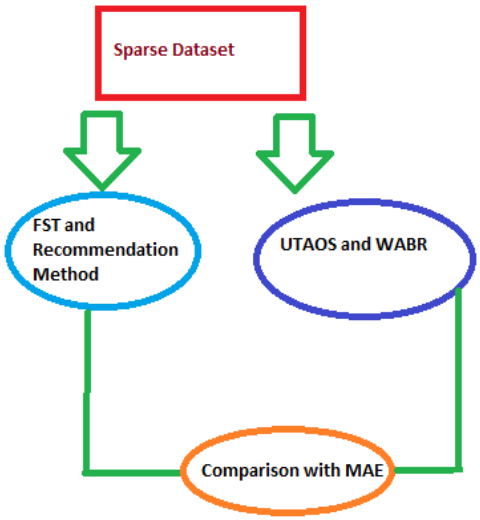

Fig. 1: The proposed idea of working

Cold Start) (Suresh *et al*., 2014) problems. The other way is by using clustering algorithms like K-means Clustering (Ramezani *et al*., 2014; Ren *et al*., 2013) and so on. Users who are in the same cluster of the active use are selected as neighbour users.

The UTAOS algorithm produces a novel method for finding neighbour users in sparse data sets and thus filling in for them. The non-redundant subspaces are identified and the systems finds list of users who share a common interest in each space. The users in single subspace have similar interesting patterns. Also a new recommendation method (WABR-Weighted Average Based Recommendation) (Ramezani *et al*., 2014) is used whilst the process is over.

The steps to perform this algorithm is listed below:

- Consider the rating matrix which contains ratings of m users on n items; here we use the matrix of MovieLens data set.
- This rating matrix is converted to binary matrix where, the ratings 4 and 5 are converted to 1 and the remaining ones to 0. This is done since rating 4 and 5 are considered as interesting items.
- The missing values are changed to 0.
- From the matrix, only the interesting items are taken i.e., items with rating 4 and 5.
- In order to find the subspaces of the interesting items which is used for finding a group of similar users, each user in the rating matrix is compared with the others users.
- A local table is created to store the subsets of each user. An intersection operation is done here.
- At the end of the iteration, the user data is put in the global table.
- A subspace list is generated.
- The redundant subspaces are removed which makes recommendation faster.

- Users similar to active users are found. Active user is the person who has shown his interest in maximum number of items and also rated them.
- Similar users of the active user are represented by a tree. Based on the active user interested items, similar users are found and a tree is constructed by placing active user at the top.
- Then users similar to first level users are found and considered as second level users of the tree. So, finally the tree contains many users which are similar to each other.
- Even if a person does not rate an item at all, he is still placed in the tree so that other neighbours can be a reference for this user.

  Next part is the recommendation part.
- There are 2 methods for recommending the items. One is predicting and other is listing.
- In predicting, the ratings of all items are predicted based on neighbour user rating and top N rated items are recommended to the user. In listing, the items has rating 4 and 5 are recommended to the active user.
- MAE is used for finding the performance of the approach. This result is saved.

Recommendation techniques in UTAOS:

- In the study, the MovieLens dataset has a sparsity level of about 95% and Jester data sets sparsity is 44%. MAE is used for evaluating the performance of the proposed methods. The K-means and Pearson coefficient are compared. We see that K-means gives better results. Although, K-means shows some empty cells which indicates it is unable to recommend the items to the user. UTAOS seems to be one of the best methods for finding the similar users. The values of the MAE must be lesser when compared with other results. As expected, WABR recommendation method gives better results when compared with prediction, 5/4321, average methods.
- The 5/4321 method states that rating 5 and 4 are interesting items, 1 to 4 are disinteresting items. So only the items with ratings 4 and 5 are recommended to the user.
- In average recommendation method, the average rating of an item (liked by similar user) is calculated and based on this value the item is recommended to the active user. In WABR average weight is calculated for an item, using the formula. Then final weight is calculated. Lastly, final score is calculated. If the value of the final score is greater than 4 then that item is recommended to the user.
- Here are some formulas related to calculation of average.

Average: First average = sum (rating*number of neighbour users) /sum (neighbour users).

Calculating final weight: Final weight = Maximum rating - (rating - first average) - 1.

Calculating final score: Final score = Sum ((final weight) 2*rating) /sum (final weight) 2).

FST: We have witnessed one algorithm and its working. Now, another algorithm that we should introduce is FST i.e., Feature Subspace Transfer, which is equally a significant algorithm in reducing data sparsity problem. Before heading to the steps, we should talk about what an auxiliary data set and target data set are.

Auxiliary by the word itself we mean supporting. An auxiliary data set has values which may be appropriate and with a little bit of polishing can bring out useful knowledge.

Target data set is what we call the sparse data set or the pending matrix wherein values will have to be filled.

Transfer learning algorithms work by transferring relevant data values from the auxiliary matrix to the target matrix, so that the sparse target matrix can become denser:

- **Create a user preference structure:** This means that the user decides what data is to be taken from the auxiliary matrix. This structure is developed by solving the Nuclear Norm Least Squares Problem.
- Next thing is the transferring of relevant data to the target matrix. Here optimization algorithms are implemented.
- A confidence parameter α is used to symbolize the amount of relativity between the auxiliary and weighted target matrices.
- An iterative feature subspace algorithm is introduced. Wiberg algorithm is an efficient algorithm due to its good numerical performance. It has a really fast iterative speed also. Implementation of FST completed.
- Successfully transfer all the user features and rating to target matrix.

After computing we get a dense target matrix. Keeping that aside, we implement recommendation methods like PCC, Soft-Impute, RMF etc., on the original MovieLens data set. Having those results as well as results after running the FST algorithm, we now compute MAE to check which one among all has the lowest error rate.

Out of all the algorithms, FST was considered the best algorithm to work on sparse data sets.

RESULTS AND DISCUSSION

After performing both the algorithms on the data set we have computed MAE. The comparison is made for a bunch of values that for a cluster of 5 values, 10 and 20, respectively. It is explained in Table 1 and the results of the comparison is shown in Fig. 2. Results

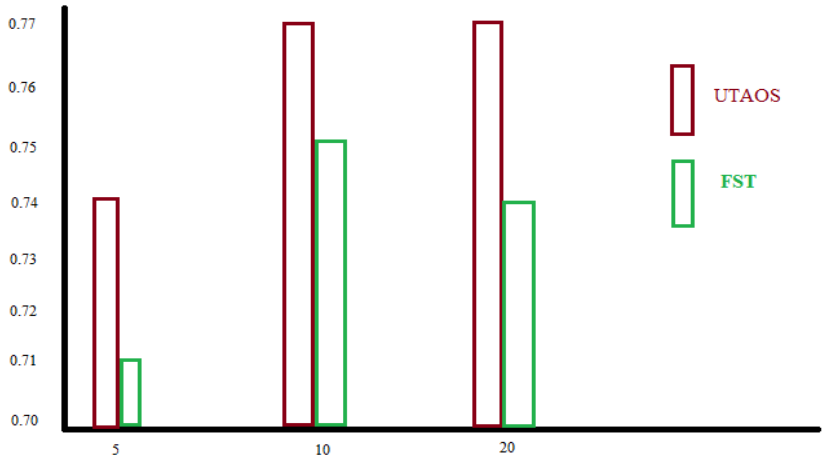

Fig. 2: Graph revealing FST with minimum MAE value

Table 1: Comparison of UTAOS and FST using MAE

Data	Metric	UTAOS+WABR	FST
Movie lens	MAE	0.74 (5)	0.71
		0.77 (10)	0.75
		0.77 (20)	0.74

show that when FST algorithm is used on weighted sparse data domains, it presents a significantly lower MAE value for each cluster or group of MovieLens data source. There is at least a reduction of 4-5% in error rate while using FST.

Table 1 shows the MAE values of UTAOS and FST method, along with recommendation methods.

CONCLUSION

While using both the algorithms on a sparse data set and comparing its Mean Absolute Error, we concluded that the transfer learning technique works better in making a sparse matrix denser. It works better than searching for neighbour users and applying their values on an incomplete matrix. Although there is difference in numerical results produces by the algorithms, FST has one more advantage over UTAOS. FST does not necessarily need user ratings to proceed; it can work with any kind of relevant data which means even item ratings and intermediate values that hold information. It is just the correct transfer of information to yield useful tokens of knowledge.

REFERENCES

Han, J. and M. Kamber, 2008. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, New York.

Li, B., Q. Yang and X. Xue, 2009. Can movies and books collaborate?: cross-domain collaborative filtering for sparsity reduction. Proceeding of the 21 International Joint Conferences on Artificial Intelligence (IJCAI'09), pp: 2052-2057.

Pan, W., E.W. Xiang and Q. Yang, 2010. Transfer learning in collaborative filtering with uncertain ratings. Proceedings of the 26th AAAI Conference on Artificial Intelligence, pp: 230-235.

Ramezani, M., P. Moradi and F. Akhlaghian, 2014. A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains. Physica A, 408: 72-84.

Ren, Y., G. Li, J. Zhang and W. Zhou, 2013. Lazy collaborative filtering for data sets with missing values. IEEE T. Cyb., 43(6): 1822-1834.

Suresh, C., Y.B. Krishya, R.P. Sini and S. Sreedhar, 2014. Recommender systems-a deeper insight. Int. J. Appl. Eng. Res. (IJAER), Research India Publications, India.

Wang, J. and L. Ke, 2014. Feature subspace transfer for collaborative filtering. Neurocomputing, 136: 1-6.