www.ijecs.in
*International Journal Of Engineering And Computer Science ISSN:2319-7242*
*Volume 4 Issue 7 July 2015, Page No. 13306-13310*

# An effective Research Paper Recommender System based on Subspace Clustering

***K. Naga Neeraja[#1], Dr. B. Prajna[#2]***

#1 Dept. of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, naganeeraja.k@gmail.com
#2 Associate Professor, Dept. of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, prajna.mail@gmail.com

**Abstract: -** There are an increasing number of research papers getting published day by day. It becomes difficult for a researcher to closely examine all the research papers in their research field and find out the papers that are related to their research work for guidance. Recommender system helps the researcher by recommending papers based on the ratings given by other researchers in that field. Collaborative filtering is one of the most successful technologies for building recommender systems and is extensively used in many commercial recommender systems. Unfortunately the computational complexity of these methods grows linearly with the number of users and number of items, which in typical research paper domain can be several millions. To address these scalability issues, we present an effective recommender system based on the subspace clustering, which analyzes the researcher-paper matrix to discover the relation between different researchers and uses these relations to compute the list of research papers to recommend.

*Keywords*: ***collaborative filtering, subspace clustering***

## I. Introduction

There is an increasing need of recommender systems due to the increase in the online publication of research papers in several conferences and journals. Due to the emerging popularity of e-commerce many older research papers are converted into electronic versions rapidly. A researcher has to study all these papers in their field of research for their research work and then select the papers that are related to their current work. It will be difficult for the researcher to go through all the thousands of papers in their field. A recommender system is a personalized information filtering technology [3] used to identify a set of papers that will be of interest to a certain researcher. It predicts a set of research papers for a certain researcher, based on the preferences of other researchers with similar interests in the same field.

Content-based filtering (CBF) and collaborative filtering (CF) [5] are the two main branches of the recommender systems. Inherent characteristics of items are taken into consideration for recommendation in CBF. A user profile is built with the keywords or attributes. Items are ranked by how closely they match the user attribute profile, and the best matches are recommended. CF approach uses a database of ratings/preferences for items by users to predict additional topics or products a user might like.

In high dimensional data, many of the dimensions are often irrelevant. Subspace clustering algorithms [2] restrict the search for relevant dimensions allowing them to find clusters that exist in multiple, possibly overlapping subspaces.

The remainder of the paper is organized as follows. Section 2 explores the details about the CF approach. Subspace clustering techniques are discussed in Section 3. The proposed method and the algorithm are described in detail in Section 4. Results and the performance comparison are discussed in Section 5. The conclusion and the future work are presented in Section 6 and Section 7.

## II. CF Approach

CF approach predicts the utility of items to a particular user (active user) based on the ratings/votes of other users. CF technic assumes that the users who have similar preferences in the past are likely to have similar preferences in the future.

Recommender systems based on CF technique use a database about user preferences to predict additional items/products a new user might like, based on the preferences of other users have expressed for those items. User-based CF approach predicts what the current user might like by finding the users whose past rating behavior is similar to the current user and use their ratings on other items for prediction. Item-based CF [4] approach uses similarities between the rating patterns of items to predict preferences.

Memory-based CF [5] algorithms utilize the entire database every time to generate recommendations. Model-based CF [5] algorithms extracts some information from the dataset, and uses that as a "model" to make recommendations without having to use the complete dataset every time. But building a model is often a time-consuming and resource-consuming process. It is usually more difficult to add data to model-based systems, making them inflexible. In model-based CF we are not using all the information (the whole dataset) available to us, so there is a chance we may don't get predictions as accurate as with model-based systems.

## III. Subspace Clustering

Subspace clustering is an extension of traditional clustering algorithms that seeks to find clusters in different subspaces within a dataset. A point might be a member of multiple clusters, each existing in a different subspaces. Traditional clustering algorithms consider all the dimensions of an input dataset in an attempt to learn as much as possible about each instance described. In real world data many dimensions are irrelevant and can mask existing clusters in noisy data. Feature selection removes irrelevant and redundant dimensions by analyzing the entire dataset.

In research paper domain there are many research papers than the number of researchers. In high dimensional data like research paper domain many dimensions are irrelevant. For example, we are not interested in the papers that are rated as poor. Subspace clustering algorithms only makes use of relevant dimensions allowing them to find clusters that exist in multiple, possibly overlapping subspaces.

Subspace algorithms can be broadly categorized based on their search method, top-down or bottom-up. Top-down approaches (e.g., PROCLUS, FINDIT, etc.) search in the full dimensional space and search smaller and smaller subspaces recursively. They first searches for all dimensions in the dataset. But in a research paper domain it is time-consuming process to check each and every dimension in which some are irrelevant. So they are not suited for recommender systems.

Bottom-up approaches (e.g., CLIQUE, MAFIA [2], etc.) first search for interesting areas in one dimension and search high dimensional subspaces only when there may be clusters in those higher-dimensional subspaces [1].

## IV. Proposed Method

We track the log of researchers to generate a researcher-paper database that consists of the data like researcher, the research paper and the rating that is given to the research paper by that researcher. We consider the 0-5 rating system. A subspace cluster of researchers contains researchers having similar interests in a particular research field. Our proposed algorithm is a five-step process. When the researcher enters into the system he has to first enter the field of his research work. Then the list of papers in that particular field are retrieved and displayed to the user. The input to the system is a researcher-paper matrix corresponding to the active user's research field and the papers that are rated by the researcher. The output is a list of research papers that are highly rated in that field by other researchers except those that are read by the current researcher. The detailed description of each step is given below.

### A. Transformation of researcher-paper matrix

The number of research papers published in web is more than the number of researchers. Here we have to eliminate irrelevant dimensions. For that we only consider the papers that are highly rated. For dealing with the sparse data in research paper domain, the rows in the researcher-paper

matrix are transformed into strings containing positions of the papers that are highly rated (here we used 0-5 ratings so we take 4 and 5 as high ratings). For example if a researcher gave rating for paper1 as 2, for paper2 as 4, for paper3 as 5, and for paper4 as 2 then that row of researcher is transformed as 2 and 3. The result of this transformation is a list of strings representing the ids of research papers that are rated highly by researchers. Let the transformed dataset as T and rows in it as $row_{id}$.

**B. Finding Intersection between rows in T**

For each row in the transformed dataset T, compare each row with each its successive rows. Initialize a hash table with null. If there is any intersection between rows then update that intersection into a hash table along with its $row_{id}$. If the intersection is already there in the hash table then update the count value. For example if transformed data for the researcher1 is 2 4 5 8 and the transformed data for researcher2 is 4 6 8 10 then the intersection 4 8 is placed in the hash table. The result of this step is a collection of intersections or subspaces, S.

**C. Removing redundancy among intersections or subspaces**

The intersections or subspaces in S are sorted according to their size in descending order. For each row in the dataset S, compare each subspace with each of its successive subspaces. If it is a subset or equivalent of $s_i$ then remove it from the subspaces list S.

**D. Finding coinciding or similar subspaces**

Find the similarity or overlap between the given subspace and the other subspaces in the list of subspaces S, to form clusters. A threshold parameter is used to control the degree of overlapping that indicates the percentage of dimensions/elements that match. If the similarity between the two subspaces is above the given threshold then the subspace is selected as a member of the cluster. This process is repeated for each element from the list of subspaces, resulting in large number of clusters of subspaces.

**E. Recommending research papers**

When a researcher enters into the system, the papers that are rated are taken. All of the subspaces containing the current researcher's papers are collected and the matching subspaces are retrieved from S. The retrieved papers or subspaces are ranked based on how similar they are with the current researcher's selection. The subspaces that are ranked higher are taken and the papers in those subspaces that are not rated and read by the researcher are recommended to the active user.

**Algorithm for recommending research papers**

Data: Researcher-paper matrix and list of fields in research.
Result: A list of papers.
Select the research field.
Collect the list of papers in that field.

Step 1:
Take the researcher-paper matrix in that field.
For each row r1 in the matrix do
    Transform the r1 into a string containing the id's of papers that are highly rated (>= 4) by the researcher.

Step 2:
Initialize the hash-table h with null.
For each row r1 in the transformed matrix do
    Compare with next row r2.
    If there is an intersection between r1 and r2 then
    Put that intersection in h along with researcher id and update count.
Initialize a threshold value.
For each intersection i in h do
    If size of i >= threshold then
        Add i to subspace list s.

Step 3:
Sort s in descending order according to size.
For each subspace s1 in s do
    Compare with next subspace s2.
    If the s2 is the subset or equivalent of s1 then
        Remove the s2 from s.

Step 4:
Collect the papers that are highly rated by the active user let us call the list as q.
For each subspace s1 in s do
    If there is an overlapping between the s1 and q and the overlap is >= threshold then
        S1 is selected as a member of cluster c.

Step 5:
Rank the members of c based on the coverage of the query.
The subspaces ranked higher are taken.
The elements in the subspaces that are not a part of the query are recommended to the active user.

## V. Results

We created a synthetic data with 20 research fields and each field containing more than 200 research papers with more than 150 researchers. The system is tested by creating and by inserting new research papers. The system performed well since it doesn't depend on the number of dimensions. In every case clusters were formed completely, no extra clusters were reported.

If the new user only read one paper and rated that then the quality of recommendations will be low. But this a rare case we come across with, since any researcher has some idea on his research work and has some idea on which papers to read. Hence the overall quality of the system will not be affected.

Precision and recall are widely used measures to evaluate the quality. We define precision as the ratio between the number of relevant research papers returned and the total number of returned research papers. Recall is defined as the ratio between the number of relevant papers returned and the number of true relevant papers.

Precision and recall also depends on the number of papers that are rated by the active researcher. If the researcher rated many papers then the result will be more accurate than the case in which the researcher rated only some papers.

Let R be the number of researchers, P be the number of Papers and C be the number of papers rated by the current researcher. We conduct experimental analysis to compare the recommendation quality, by varying the values of R, P and C. The quality of recommendations varies with the value of C. The results are as shown in Table 1.

Table 1: Results of the analysis

| R | P | C | Precision | Recall |
|---|---|---|---|---|
| 20 | 30 | 5 | 0.895 | 0.980 |
| 20 | 30 | 10 | 0.925 | 0.989 |
| 50 | 100 | 10 | 0.883 | 0.892 |
| 50 | 100 | 30 | 0.861 | 0.897 |
| 150 | 300 | 30 | 0.841 | 0.864 |
| 150 | 300 | 90 | 0.867 | 0.875 |
| 250 | 500 | 50 | 0.815 | 0.803 |
| 250 | 500 | 150 | 0.824 | 0.793 |
| 350 | 700 | 70 | 0.786 | 0.794 |
| 350 | 700 | 210 | 0.779 | 0.789 |

## VI. Conclusion

In this paper, we recommended research papers to the researchers in an effective way by using subspace clustering, which processes only relevant dimensions. Papers are recommended based on the ratings given by the researches in that field. Thus, this provides high quality recommendations and is fast. We explained in detail how subspaces are formed and used for recommending research papers to increase quality of our system.

## VII. Future Work

In future, in order to improve the quality to higher extent, we can use subjective user ratings or by taking opinion of the researcher on the paper he read.

## VIII. References


[1] Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data: a review.

[2] Karam Gouda and Mohammed J. Zaki. Efficiently Mining Maximal Frequent Itemsets.

[3] Joonseok Lee, Kisung Lee, Jennifer G. Kim. Personalized Academic Research Paper Recommendation System.

[4] Mukund Deshpande and George Karypis. Item-based top-n Recommendation Algorithms.

[5] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems.

[6] Robin van Meteren and Maarten van Someren. Using Content-Based Filtering for Recommendation.

[7] Cheng-Fa Tsai, Han-Chang Wu, and Chun-Wei Tsai. A new Data Clustering Approach for Data Mining in large Databases.

[8] Joeran Beel, Stefan Langer, Marcel Genzmehr, Bela Gipp, Corinna Breitinger, Andreas Nurnberger. Research Paper Recommender System Evaluation: A Quantitative Literature Servey.

[9] Michael J. Pazzani and Daniel Billsus. Content-based Recommendation Systems.