*International Journal of Research in Advent Technology, Vol.3, No.4, April 2015*
*E-ISSN: 2321-9637*

# Personalized Recommendation on Web Data

Prof. Avinash Palve[1], Akshay Tamhane[2], Ashwin Iyer[3], Akshay Mitkal[4], Prasad Satam[5]
*Computer Engg,TCOER[1, 2, 3, 4,5], Savitribai Phule Pune University,India[1, 2, 3, 4,5]*
*Email: avi.palve@gmail.com[1] , akshay.tamhane14@gmail.com[2]*

**Abstract-**E-commerce and web-based services are more effective if recommendations are provided. Recommendations are carried out dynamically on web data. In this paper, such recommendation technique is proposed, which has information about rating and profile contents. These two are used to maintain a relation. The user preferences are recorded for this recommendation. Along with the preferences, the features are adaptively weighted for personalized recommendation. Using recommendation technique helps in avoiding the skipping of the end users' preferred item**.**

**Index Terms-** sparse data; recommendation; clustering

## 1. INTRODUCTION

Nowadays the internet has become an important part of human lives. It is a platform for sharing information across the world. With increase in its use, the data is expanding at a faster rate. The data over the internet is scattered which is also known as sparse data. Here comes the challenge of providing the precise content to the end user. This is where personalized recommendation comes into play for the end users' satisfaction.

Our task is to gather the data keeping the users' interest in mind. This can be achieved by taking a note of the users' activity. Accordingly, ratings are given to each item. To keep a track of the above mentioned attributes user profile is taken into account. Based on their views and activity the algorithm will analyze user interest domain. Every users history will help to know the users tendency and interest and it will vary based on their usage.

Recommendation of items relevant to users' interest must be done dynamically. But the sparse data creates hurdles while doing so. Thus, there is a need to manage the data. To keep the data ready for recommendation, clusters are to be formed. The tracking of all the items are done and processed to form a cluster based on the similar items in that domain. We use a clustering algorithm to cluster or group the products under one domain. These clusters will encompass all the similar items and thus, making it easier to recommend the relevant items. When the clusters are in place, the chances of missing out any data becomes close to nothing.

Fig (1). alongside gives an idea on how the closely related items are bound in a cluster. The items within a cluster have similar features which is judged by a threshold value.

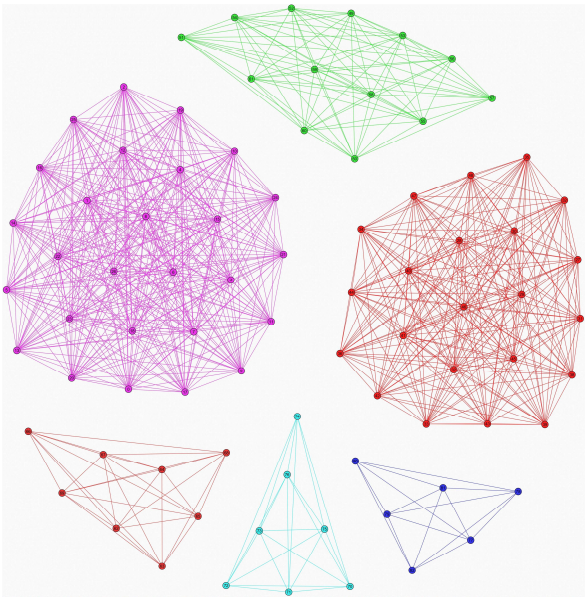

Fig. (1) Formation of clusters

## 2. THE PROPOSED METHOD

Firstly we will be forming clusters of the similar items to recommend similar items followed by an association rule to recommend the items associated in a transaction at a given time.

## 3. CLUSTERING

In this paper we will be seeing clustering using the cosine similarity algorithm. Whenever any user browses through any product, system automatically process and finds out whether the product falls under any cluster or not. If it falls in any defined cluster then it looks after the products there in the cluster to get any missing item. The missing item is then

recommended to the user so that user gets to buy that product.

$$\text{similarity} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad \ldots(1)$$

### 3.1 *Algorithm*

i. Find list P of products viewed or purchased by current user
ii. for each pair of products $A_i$, $B_i$ in P
iii. take features of 2 products as input
iv. Calculate cosine c = sum ($A_i$. $B_i$) where $A_i$ = ith feature of product A and $B_i$ = ith feature of product B.
v. add cosine to list product_cosine
vi. Calculate threshold = avg ( values in product_cosine )
vii. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster
viii. Compute distances (similarities) between the new cluster and each of the old clusters.

## 4. ASSOCIATION RULE

As said earlier, the association rule is solely responsible for recommending the items related or as the name of the rule says, associated with each other. In this paper we will see the Apriori algorithm used as an association rule.

**Support**

The rule $X \Rightarrow Y$ holds with support s if s% of transactions in D contain $X \cup Y$. Rules that have a s greater than a user-specified support is said to have minimum support.

$$\text{supp}(X) = \frac{\text{number\_of\_transactions\_containing\_X}}{\text{number\_of\_transactions}} \quad \ldots(2)$$

**Confidence**

The rule $X \Rightarrow Y$ holds with confidence c if c% of the transactions in D that contain X also contain Y. Rules that have a c greater than a user-specified confidence is said to have minimum confidence.

$$\text{conf}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)} \quad \ldots(3)$$

### 4.1 *Algorithm*

**<u>3 steps to be followed:</u>**

**i. <u>Boolean Matrix Generation:</u>**
This step is to convert the data's over the web and the incoming instances of data into Boolean value (either 0"s or 1"s). It will first prepare a matrix named as pdb(suppose) based on transaction where no. Of row would be no. Of transaction and no. Of column would be name of item. If product present in transaction then it will mark as 1 else 0 and based on this it will generate matrix having value as 0 or 1.

**ii. <u>Frequent Itemset Generation:</u>** In this step it finds frequent items(product) from existing transaction based on support value. It first checks whether sum of each product column is greater than support value if yes then it will take that column as f, else it will delete that column from generated pdb table same summation would be followed for row, it will sum each row and check if its value would be less than 2 then it will delete that row from pdb matrix and after this process it will make combinations of items and result frequent item sets.

**iii. Association Rule:**
This is used to generate association rules from the already generated frequent item sets. The algorithm checks if a given set is a subset of another set or not. To perform this operation each item in an item set is represented as an integer where a bit corresponding to as item is set to 1. Suppose there are 2 sets named as fk and fm, **if fk is subset** of fm then it will increment found by 1 and **conf**=support(fm)/ support(fk) and fk will get deducted for further process as fk = (fm - fk) support=support(fm) **else** found=0. This process will run till it reaches to maxsize-1.

## 5. CONCLUSION

In this paper, we have seen that using cosine similarity we can form clusters of closely related sparse data over the internet. Also, we can recommend those items to the users using clusters and the association between them.

## FUTURE WORK

There is a scope for betterment of the data mining algorithms which will only help to give accurate result. Formulation of improved algorithm with faster recommendation using improved association can be achieved, thus, making the database access quite less.

**Acknowledgments**

We are grateful to people who have helped us in completing our paper, without whom, the completion of this document would not have been possible. Firstly we would like to thank our esteemed guide, Prof. Avinash Palve, for his guidance at all the times and that he provided his support and guidance without complains at any time of the day and also for the bright ideas and inputs that they gave for the document.

We would also like to thank Prof. S. B. Chaudhari (HOD COMPUTER) for his keen interest and Cooperation towards this document. We wish to express our sincere gratitude to TCOER, Pune, for providing facilities like Internet and Library. We would also like to convey thanks to all those who directly and indirectly helped us in completing this document.

## REFERENCES

[1] IEEE Transactions on K nowledge and Engineering, VOL. XX, NO. YY, 2011" "clustering with multi view point based similarity measure"
[2] Cole, A. J. & Wishart, D. (1970). An improved algorithm for the Jardine-Sibson method of generating overlapping clusters. The Computer Journal 13(2):156-163.
[3] D'andrade,R. 1978, "U-Statistic Hierarchical Clustering" Psychometrika, 4:58-67.
[4] R. Agrawal, C. Faloutsos, and A. Swami. Ef-cient similarity search in sequence databases. In Proc. of the Fourth International Conference on Foundations of Data Organization and Algorithms, Chicago, October 1993.
[5] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2(2), 1987.
[6] J. Han, Y. Cai, and N. Cercone. Knowledge discovery in databases: An attribute oriented approach. In Proc. of the VLDB Conference, pages 547{559, Vancouver, British Columbia, Canada, 1992.
[7] WEI Yong-qing, YANG Ren-hua, LIU Pei-yu, "An Improved Apriori Algorithm for Association Rules of Mining" IEEE(2009)
[8] Mrs. R. Sumithra, Dr (Mrs). Sujni Paul, "Using distributed apriori association rule and classical apriori mining algorithms for grid based knowledge discovery", 2010 Second International conference on Computing, Communication and Networking Technologies, IEEE.
[9] Zhuang Chen, Shibang CAI, Qiulin Song and Chonglai Zhu, "An Improved Apriori Algorithm Based on Pruning Optimization and Transaction Reduction", IEEE 2011.
[10] D. Kerana Hanirex, and M.A. Dorai Rangaswamy. 2011. Efficient Algorithm for Mining Frequent Itemsets using Clustering Techniques. International Journal on Computer Science and Engineering (IJCSE) Vol. 3 No. 3 March 2011.