Recommending Top-n Research Papers (Based on With, Without and Boolean Items Preferences: An User Base Collaborative Filtering Approach in Mahout)

Ashishkumar B. Patel, Niravkumar B. Suthar and Jitendra S. Dhobi

Abstract— *Recommender systems are programs that aim to present items like songs, books and friends that are likely to be interesting for a user. In web systems, like e-shops or community servers there are usually multiple data sources we can use for recommending, as user and item attributes, user-item rating or implicit feedback from user behaviour. Researchers often spend a considerable amount of time searching for published papers and articles relevant to their interest, dissertation and research work. Despite having similar interests, they conduct independent, time consuming searches. While they may share the results afterwards, they are unable to leverage previous search results during the search process. We propose a research paper recommender system that avoids such time consuming searches by augmenting existing search engines with recommendations based on previous searches performed by others. We explore various User Similarity Models for dataset with Paper Preferences, without Paper Preferences and Boolean Preferences. We tested all Similarity Models for User Based Collaborative filtering techniques for recommending top-n Research Papers in Apache Mahout and find out best similarity model. (Abstract)*

Keywords--- *Recommender Systems, User Similarity Models, Paper Preferences, Collaboratice Filtering, Top-n, Mahout*

I. INTRODUCTION

EVERYONE can bring to mind examples of recommendation. We might think of reading movie reviews in a magazine to decide which movie to see. Or we might recall visits to our local bookstore, where we've talked to the owner about our interests and current mood, and we then recommended a few books we'd probably like. Finally, we might remember walking through our neighbourhood and noticing that a particular sidewalk hotel always is crowded. We think that its popularity must be a good sign, so we decide to give it a try.

Reflecting on these examples helps to clarify the concept of recommendation. A person is faced with a decision, which for our purposes is a choice among a universe of alternatives. The universe typically is quite large, and the person probably doesn't even know what all the alternatives are, let alone how to choose among them. If the person doesn't have sufficient personal knowledge to make the choice, he or she may seek recommendations from others.

Recommendation is based on the *preferences* of the recommender (and perhaps the seeker and other individuals).

For my purposes, a preference is an individual mental state concerning a subset of items from the universe of alternatives. Individuals form preferences based on their experience with the relevant items, such as listening to music, watching movies, testing food, etc.

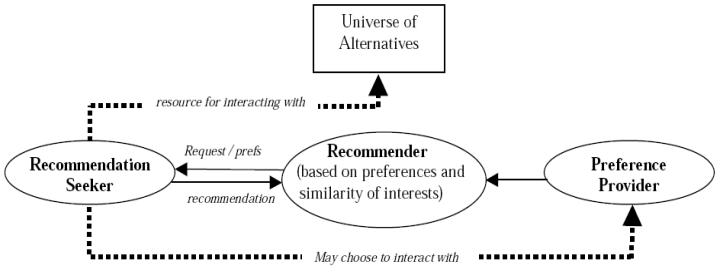

Figure 1: Model of the Recommendation Process

Figure 1 summarizes these concepts and situates them in a general model of recommendation. A recommendation seeker may ask for a recommendation, or a recommender may produce recommendations. Seekers may volunteer their own preferences, or recommenders may ask about them. Based on a set of known preferences – his/her own, the seeker's, and those of other people, often people who received recommendations in the past – the recommender recommends items the seeker probably will like. In addition, the recommender may identify people with similar interests. The seeker may use the recommendation to select items from the universe or to communicate with like-minded others.

II. MAHOUT - A COLLABORATIVE FILTERING LIBRARY

Mahout[1] is an open source machine learning library from Apache. The algorithms it implements fall under the broad umbrella of machine learning or collective intelligence. This can mean many things, but at the moment for Mahout it means

Ashishkumar B. Patel, Department of ME Computer Science and Engineering, Govt. Engineering College, Modasa, Gujarat, India. E-mail: ashishmemod@gmail.com

Niravkumar B. Suthar, Department of ME Computer Science and Engineering, Govt. Engineering College, Modasa, Gujarat, India. E-mail: niravsuthar@live.com

Jitendra. S. Dhobi, Department of Computer Engineering & IT, Govt. Engineering College, Modasa, Gujarat, India. E-mail: jsdhobi@yahoo.com

ISBN 978-93-82338-02-4 | © 2012 Bonfring

primarily recommender engines (collaborative filtering), clustering, and classification. It's also scalable. Mahout aims to be the machine learning tool of choice when the collection of data to be processed is very large, perhaps far too large for a single machine. In its current incarnation, these scalable machine learning implementations in Mahout are written in Java, and some portions are built upon Apache's Hadoop distributed computation project.

A Mahout-based collaborative filtering engine takes users' preferences for items and returns estimated preferences for other items. Mahout provides a rich set of components from which we can construct a customized recommender system from a selection of algorithms.

III. DATA SET USED

MovieLens[2] data sets were collected by the GroupLens Research Project at the University of Minnesota.

This data set consists of:

- 100,000 ratings (1-5) from 943 users on 1682 movies.
- Each user has rated at least 20 movies.

The GroupLens Research Project is a research group in the Department of Computer Science and Engineering at the University of Minnesota. Members of the GroupLens Research Project are involved in many research projects related to the fields of information filtering, collaborative filtering, and recommender systems.

The full data set is of about 90,500 ratings by 943 users on 1682 items. Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered. This is a tab separated list of user id | item id | rating | timestamp.

In the case of recommending Research Papers, this data set is used with assumption that the item id of MovieLens is representing the Paper Id. By using this dataset several algorithms are tested. In this implementation the output will display the top-n items with their item id. Based on the item id of the top-n recommended items, their respective paper titles can be displayed.

IV. IMPLEMENTING AND EVALUATING USER BASED RECOMMENDER

This is the approach described in some of the earliest research[5] in the field, and it's certainly implemented within Mahout. The defining characteristic of a user-based recommender algorithm is that it is based upon some notion of similarities between users. In fact, we've probably encountered this type of algorithm in everyday life.

A. Experiment on Dataset: Item Preferences are considered

The Dataset have the following format:

user id | item id | rating | timestamp.

Here we used item rating for recommendation purpose.

V. (A) IMPLEMENTING GENERIC USER BASED RECOMMENDER

The necessary components of the standard user-based recommender algorithm are:

- UserSimilarity encapsulates some notion of similarity among users
- UserNeighborhood encapsulates some notion of a group of most-similar users.

i.e. GenericUserBasedRecommender(model,neighborhood, similarity);

For User Similarity, two similarity models are implemented and tested for the proposed system, Pearson Correlation Similarity and Uncentered CosineSimilarity.

A. Cosine Similarity

In this similarity[3] the result is the cosine of the angle formed between the two preference vectors. In this similarity metric, the paper rating is used as a vector to find the normalized dot product of the two users. By determining the cosine similarity, the user is effectively trying to find cosine of the angle between the two users.

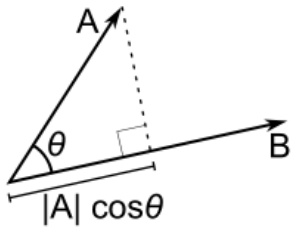

Figure 2: Cosine Similarity

Cosine similarity is literally the angular difference between two vectors. Cosine Similarity is expressed by the formula:

$$Similarity = \cos(\theta) = \frac{\Sigma Ai * Bi}{\sqrt{\Sigma (Ai)^2} * \sqrt{\Sigma (Bi)^2}}$$

The result of this calculation will always be a value between 0 and 1, where 0 means 0% similar, and 1 means 100% similar Experiment Gives the top-10 recommended papers with Precision of 0.005229793978 and Recall of 0.005847357432.

B. Pearson Correlation Similarity

A correlation is a number between -1 and +1 that measures the degree of association between two variables (call them X and Y). A positive value for the correlation implies a positive association (large values of X tend to be associated with large values of Y and small values of X tend to be associated with small values of Y). A negative value for the correlation implies a negative or inverse association (large values of X tend to be associated with small values of Y and vice versa).

Suppose we have two users X and Y, then

$$Pearson(x,y) = \frac{\sum xy - \frac{\sum x \sum y}{N}}{\sqrt{(\sum x^2 - \frac{(\sum x)^2}{N})(\sum y^2 - \frac{(\sum y)^2}{N})}}$$

where, (x,y) refers to the data objects and N is the total number of attributes.

Experiment Gives the top-10 recommended papers with

ISBN 978-93-82338-02-4 | © 2012 Bonfring

Precision of 0.016481774960 and Recall of 0.018598596332 .

VI. Experiment on Dataset: Item Preferences are Not Considered

Here we will not use item rating for recommendation purpose.

user id | item id

VII. (A) Experiment on Dataset: Generic Boolean Preference (yes/no) are considered

Sometimes, the preferences that go into a recommender engine have no values. That is, a user and item are associated, but there's no notion of the strength of that association[6]. For example, imagine a news site recommending articles to users based on previously viewed articles. A view establishes some association between the user and the item, but that's about all that's available. It isn't common for users to rate articles. It's not even common for users to do anything more than view an article. All that's known in this case is which articles the user is associated with, and little more.

The necessary components of the standard user-based recommender algorithm are:

- UserSimilarity encapsulates some notion of similarity among users
- UserNeighborhood encapsulates some notion of a group of most-similar users.

i.e.GenericBooleanPrefUserBasedRecommender

(model,neighborhood, similarity);

A. *Loglike similarity*

In this similarity two users are similar when they rate or are associated to many of the same items.

However, a certain overlap may or may not be meaningful - it could be due to chance, or due to the fact that we have similar tastes. For example if you and I have rated 100 items each, and 50 overlap, we're probably similar. But if we've each rated 1000 and overlap in only 50, maybe we're not.

The log-likelihood metric is just trying to formally quantify how unlikely it is that our overlap is due to chance. The less likely, the more similar we are.

So it is comparing two likelihoods, and just looking at their ratio. The numerator likelihood is the null hypothesis: we're not similar and overlap is due to chance. The denominator is the likelihood that it's not at all due to chance - that the overlap is completely explained is perfectly explained because our tastes are similar and the overlap is exactly what we'd expect given that.

When the numerator is relatively small, the null hypothesis is relatively unlikely, so we are similar.

Experiment gives the top-10 recommended papers with Precision of 0.125673534073 and Recall of 0.125673534073

B. *Tanimoto Coefficient Similarity*

This is one such implementation, based on (surprise) the Tanimoto coefficient/Jaccard Distance[4]. This value is also known as the Jaccard coefficient. It's the number of items that two users express some preference for, divided by the number of items that either user expresses some preference for, as illustrated in figure 3.

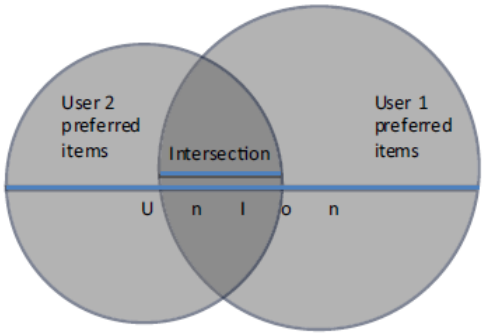

Figure 3: The Tanimoto Coefficient

Experiment gives the top-10 recommended papers with Precision of 0.167038216561 and Recall of 0.210589766810.

VIII. Comparison between all Recommenders

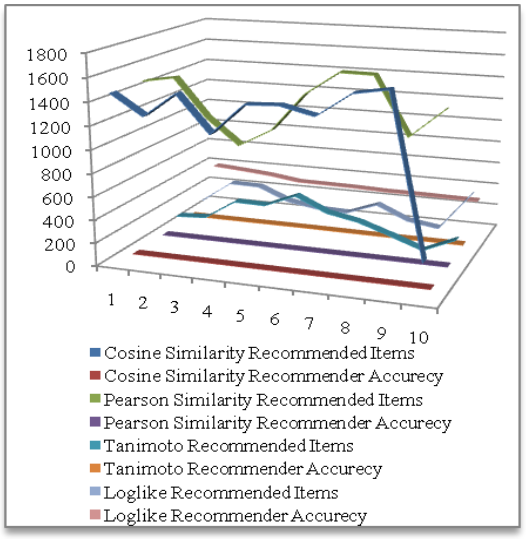

Figure 4: Recommended Items with Recommender Accuracy (all Models)

Table 1: Comparison between all Similarity Models on Precision and Recall

Similarity	Precision	Recall
Cosine Similarity	0.005229793978	0.005847357432
Pearson Similarity	0.016481774960	0.018598596332
Tanimoto Similarity	0.167038216561	0.210589766810
Loglike Similarity	0.125673534073	0.159425703720

ISBN 978-93-82338-02-4 | © 2012 Bonfring

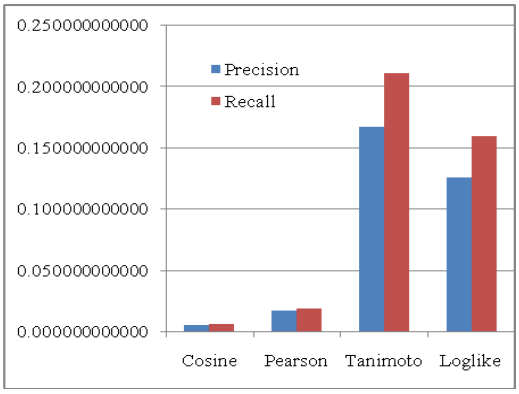

Figure 5: Comparison between all Similarity Models on Precision and Recall

According to above figures it is clear that the recommendations are more accurate when preferences are not considered. Both LogLike and Tanimoto Similarities are superior than similarities with Paper Preferences are considered. Based on Precision and Recall values the Recommender with Tanimoto Coefficient Similarity is good choice for recommending Research Papers.

IX. CONCLUSION

We proposed a User based collaborative filtering approach to implement a Research Paper Recommender system. Our experiments are tested the dataset based on with Paper Preferences, without Paper Preferences and Boolean(yes/no) Paper Preferences. Experiments states that recommender system works well when the Paper Preferences are not considered for paper recommendation. Paper Preferences works well with Tanimoto Coefficient Similarity with compare to the Cosine and Person Correlation Similarity which works on the explicit Paper Preferences.

REFERENCES

[1] https://cwiki.apache.org/confluence/display/mahout/recommender+documentation
[2] http://www.movielens.org/
[3] D. Almazro, M. Kherees, R. Martinez, and W. Nzoukou, "A Survey Paper on Recommender Systems," in Analysis, 2010K. Elissa, "Title of paper if known," unpublished.
[4] "Recommendation Systems," pp. 287-321.
[5] S. M. Mcnee, N. Kapoor, and J. A. Konstan, "Don 't Look Stupid: Avoiding Pitfalls when Recommending Research Papers," Human Factors, 2006.
[6] M. Hahsler, "Developing and Testing Top-N Recommendation Algorithms for 0-1 Data using recommenderlab Collaborative Filtering for 0-1 Data," Most, pp. 1-21, 2011.

ISBN 978-93-82338-02-4 | © 2012 Bonfring