Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 50 (2015) 229 – 234

Procedia Computer Science

2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

Performance and Quality Assessment of Similarity Measures in Collaborative Filtering Using Mahout

Saikat Bagchi*
Indian Institute of Technology
Kharagpur, India
sbagchi1982@gmail.com

ABSTRACT

Recommendation systems use knowledge discovery and statistical methods for recommending items to users. In any recommendation system that uses collaborative filtering methods, computation of similarity metrics is a primary step to find out similar users or items. Different similarity measuring techniques follow different mathematical approaches for computation of similarity. In this paper, we have analyzed performance and quality aspects of different similarity measures used in collaborative filtering. We have used Apache Mahout in the experiment. In past few years, Mahout has emerged as a very effective and important tool in the area of machine learning. We have collected the statistics from different test conditions to evaluate the performance and quality of different similarity measures.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement Techniques, Performance attributes

General Terms

Performance, Measurement

Keywords

Performance and Quality of Similarity Measures, Performance of Mahout-based Recommendation, Performance of User-based Recommendation, Analysis of Similarity Measures, Similarity Measures in Collaborative Filtering

1. INTRODUCTION

Recommendation systems use knowledge discovery and statistical methods for recommending different kind of items to users. At present e-commerce systems offer millions of products for sale. Customers of e-commerce systems often have very little or no knowledge about all offerings provided by those. e-Commerce systems have to predict preferences of customers and recommend products to them to optimize sales. A recommendation system may collect preferences of a customer for different items and recommend new products to him/her predicting his/her preferences for those products. Recommendation techniques play very important role in social networking and other online services like online news service, music/movie service etc. where presentation of personalized items to users is a very important aspect of business. There are various types of techniques for recommendation. Collaborative filtering, content-based recommendation, hybrid recommendation etc. are well-known approaches for generating recommendations. In collaborative filtering approaches of recommendation, items are recommended to a customer by assessing preferences of other customers who are in the neighborhood based on their historically similar taste to the first customer, so similarity-measure is a significant aspect of collaborative filtering.

In this paper we are going to analyze performance and quality aspects of recommendation using different types of similarity measures provided by Apache Mahout. Apache Mahout is an open-source project, which provides scalable implementations of machine learning techniques like collaborative filtering, clustering, classification etc. We will use Movie Lens data from Group Lens dataset for the experiment.

In sections 2 and 3 we will mention summary about related work, overview about recommendation system, definitions of similarity measures and Mahout, which are used in our assessment. In section 4 we will explain our work on performance and quality assessment of similarity measures used in recommendation system.

2. RELATED WORK

Recommender systems have emerged to help users to navigate through large volume of online content. Many online search systems, e-commerce websites, online news services, online multimedia services etc. are exploiting the benefits of recommendation systems in providing extra mileage to their business. Works on evaluation of recommendation systems include Herlocker et al.'s [8] survey and Shani and

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

* Student in Master of Technology at Indian Institute of Technology, Kharagpur, India.

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)
doi:10.1016/j.procs.2015.04.055

Gunawardana's book [13]. There have been several other works on this topic. In almost 50% of the studies on benchmarking of recommendation systems, open data sets have been used; almost similar amount of studies presented information on test/training splits. Very less number of studies used open dataset, open framework, and provided all necessary details for replication of experiments and results. Algorithmic details have been disclosed in almost 25% of the studies. Said et al. [12] have performed comparative study and benchmarking of recommendation systems implemented using separate open source frameworks and open data sets and tried to address the issues related to replication of experimental results. Owen et al. has provided some details about comparative analysis of different similarity measures using Mahout in their book [1], but that is not complete w.r.t. the above mentioned parameters for replication. There is a need of comparative analysis of similarity measure algorithms with open dataset, open framework with disclosure of full details about algorithms and environments for facilitating future study and validation on benchmarking of recommendation systems.

3. OVERVIEW ABOUT CONCEPTS & TOOLS USED IN THE ASSESSMENT

3.1 Recommendation System

Recommendation System, a sub-class of information filtering system, helps in predicting top-N preferred items for a user. Recommendation techniques follow mainly following approaches: collaborative filtering, content-based recommendation and hybrid recommendation. Collaborative filtering methods build a model using information about past purchases or ratings provided by users. A model may also be created based on decisions (preference ratings or selection of items) taken by similar users. This model may be used for prediction of preference rating for a given item. In content-based methods, features of an item are compared against features of other items to recommend items. In collaborative filtering process a large amount of information on a user is required to make accurate predictions (cold-start problem), where as content-based recommendation needs very little information to get started. Following subsection gives a summary about collaborative filtering method.

3.1.1 Collaborative Filtering

Collaborative filtering methods analyze large amount of information about preferences of users and predict preferences of similar users for recommending items. In collaborative filtering method an accurate prediction of preferences of a user and recommendation of items is possible without any need for detailed analysis of item features. A basic assumption in collaborative filtering is that users would like similar kinds of items as they have liked in past.

Collaborative filtering methods suffer from issues like – cold start, scalability and sparsity.

Following section describes about similarity measurement techniques, which are used in collaborative filtering methods.

3.2 Similarity Measures

A similarity measure or similarity function is a real-valued function that quantifies the similarity between two objects. Although no single definition of a similarity measure exists, usually similarity measures are in some sense the inverse of distance metrics: they take on large values for similar objects and either zero or a negative value for very dissimilar objects.

One of the preferred approaches to collaborative filtering (CF) recommenders is to use k-Nearest-Neighborhood (kNN) classifier, which is dependent on defining an appropriate similarity or distance measure. Definitions[1] of some popular similarity measures, which are used in our experiment, are given below:

3.2.1 Euclidean distance

Mathematical definition of Euclidean distance measure is given below for two objects x and y:

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$$

Here n is number of dimensions (attributes) and x_k and y_k are k^{th} attributes (components) of data objects x and y

3.2.2 Minkowski distance

Minkowski distance is a generalized distance measure and is represented mathematically as below:

$$d(x,y) = (\sum_{k=1}^{n}|x_k - y_k|^r)^{\frac{1}{r}}$$

Here r is degree of distance. Depending on the value of r, generic Minkowski distance is known with specific names:

- For r = 1, City block (Manhattan, taxicab or L_1 norm) distance
- For r = 2, Euclidean distance
- For $r \to \infty$, Supremum (L_{max} norm or L_∞ norm) distance, which corresponds to computing the maximum difference between any dimensions of k objects.

3.2.3 Cosine similarity or L_2 Norm

Cosine similarity is the measure of similarity between two vectors of an inner product space that measures the cosine of angle between them.

$$\cos(x,y) = \frac{(x \bullet y)}{\|x\|\|y\|}$$

Here • indicates vector dot product and ||x|| is the norm of vector x.

3.2.4 Pearson correlation

Pearson correlation score checks how highly 2 variables are correlated. A Pearson correlation coefficient is represented as below:

$$Pearson(x,y) = \frac{\sum(x,y)}{\sigma_x \times \sigma_y}$$

Here $\sum$ is the covariance of data points x and y and σ is the standard deviation.

[1] http://en.wikipedia.org

Recommendation Systems generally use either cosine similarity or Pearson correlation or one of their many variations through, for instance, weighting schemes.

3.2.5 Spearman correlation

Spearman correlation coefficient, ρ, is defined as the Pearson correlation between ranked variables.

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

For a sample of size n, the n raw scores X_i, Y_i are converted to ranks x_i, y_i. $d_i = x_i - y_i$, is difference between ranks

3.2.6 Tanimoto coefficient

If samples X and Y are bitmaps, then Tanimoto similarity ratio is defined as:

$$T_s(X,Y) = \frac{\sum_i (X_i \wedge Y_i)}{\sum_i (X_i \vee Y_i)}$$

Tanimoto distance coefficient is defined as:

$$T_d(X,Y) = -\log_2(T_s(X,Y))$$

3.2.7 Log likelihood similarity

A likelihood ratio test is a statistic test used to compare the fit of two models, one of which (the null model) is a special case of the other (the alternative model). The test is based on the likelihood ratio, which expresses how many times more likely the data are under one model than the other. This likelihood ratio, or equivalently its logarithm, can then be used to compute a p-value or compared to a critical value to decide whether to reject the null model in favor of the alternative model. When the logarithm of the likelihood ratio is used, the statistic is known as a log-likelihood ratio statistic, and the probability distribution of this test statistic, assuming that the null model is true, can be approximated using Wilk's theorem.

D = -2 ln (likelihood for null model/ likelihood for alternative model)

= -2 ln (likelihood for null model) + 2 ln (likelihood for alternative model)

Next section will provide overview about different features of Apache Mahout, which will be used for performance and quality assessments of similarity measures in our experiment.

3.3 Recommendation Framework of Mahout[2]

We have used open source libraries of Apache Mahout for our experiment. Mahout is an open source project to provide free implementations of scalable and distributed machine learning algorithms in the areas of collaborative filtering, clustering and classification. Mahout is written in Java and has version 0.9 at the time of writing. It provides both non-distributed and distributed (Map-Reduce) algorithms for recommendation. In this work, the non-distributed algorithms of Mahout have been used. Mahout has several algorithms for similarity measures, neighborhood computation and evaluation. We are going to use the non-distributed algorithms for user-based recommendation.

Mahout uses various types of similarity measures, neighborhood computation and evaluation techniques in collaborative filtering methods of the recommendation processes.

In Mahout user-based recommendation the interaction between high-level components is displayed in Figure 3-1[1]:

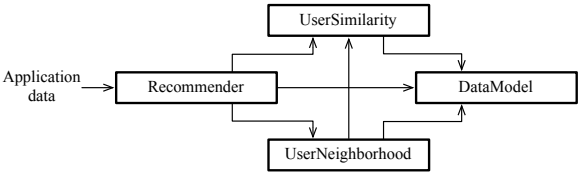

Figure 3-1: Interaction between components in Mahout user-based recommendation

A *DataModel* implementation stores and provides access to all the preference, user and item data needed for computation. A *UserSimilarity* implementation provides similarity details of two users using one of the different similarity measuring algorithms available in Mahout. A *UserNeighborhood* implementation provides the group of users who are most similar to a given user. A *Recommender* uses all the above components together to recommend items to users.

3.4 Similarity measures supported in Mahout

Apache Mahout supports following similarity measure algorithms:

- Euclidean distance similarity
- City-block similarity
- Un-centered cosine similarity
- Pearson correlation similarity
- Spearman correlation similarity
- Tanimoto coefficient similarity
- Log likelihood similarity

3.5 Evaluation of a recommender

Most of the recommendation systems, while recommending items to a user, try to estimate the ratings for some or all other items, which are not already rated by the given user. Quality of estimated preference values can be evaluated by measuring how closely estimated preferences match actual preferences. We used standard methods – Evaluation score, Precision and Recall, for evaluation of the quality of experiment. A brief details about the methods are given below.

3.5.1 Evaluation of score

As actual preferences of items by a user don't exist for items, which have not been already rated by the user, a simulation technique needs to be used for the evaluation. A small part of real data (with preference values) is set aside as test data. Another part of real data is used as training data. A recommendation system is asked to estimate the preference values for the test data and the results are compared with actual preference values to measure the quality of recommendation.

A score can be generated for a recommender from evaluation. Average difference (mean absolute error, MAE) between estimated and actual preferences or root-mean-square (RMSE) of the differences can be used for calculation of scores. Lower score

[2] https://mahout.apache.org

is better as that indicates that estimates are closer to actual preference values.

For example when average difference is used for evaluation score computation, a score of 1.0 signifies that, on average, the recommender estimates a preference that deviates from the actual preference by 1.0.

3.5.2 Precision and Recall

In addition to the above-mentioned evaluation technique of scoring, recommenders can also be evaluated by information retrieval metrics like precision and recall. These terms are used for search engines, where best results out of many possible results are provided as a query output. Precision is the proportion of top results that are relevant for some definition of relevance. Recall is the proportion of all relevant results included in the top results.

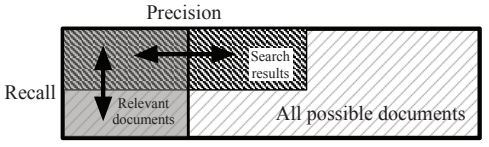

Figure 3-2: Precision and Recall in context of search results

In the context of recommendations Precision is the proportion of top recommendations that are good recommendations and Recall is the proportion of good recommendations that are top recommendations.

4. EXPERIMENTAL EVALUATION

In this section we present the results of performance and quality assessment of similarity measures. We have performed this assessment using user-based recommendation of Apache Mahout.

4.1 System configuration

We carried out the assessment using following configuration:

Table 1: System and other configuration

Processor	*2.53 GHz Intel Core 2 Duo*
RAM	*8 GB 1067 MHz DDR3*
Operating System	*Macintosh OS X 10.9.3*
JVM	*JRE 1.6.0_65*
Mahout	*Apache Mahout 0.9*
Data file	*Movie lens data of 27.2 MB*
User-item preference rating scales	*1 to 5*

4.2 APIs used for model, neighborhood, evaluator

We used following APIs of Mahout during the assessment:

Table 2: Used Mahout APIs

Parameter	Mahout API
Data Model	*FileDataModel*
Neighborhood Algorithm	*NearestNUserNeighborhood*
Evaluator	*AverageAbsoluteDifferenceRecommenderEvaluator*
IR stats evaluator (Precision/Recall)	*GenericRecommenderIRStatsEvaluator*

4.3 Similarity Measures Used in Assessment

We have used following similarity measures for evaluation in the experiment.

Table 3: Mahout APIs used for similarity measurement

Similarity Measure & abbreviated form	Mahout API
Euclidean Distance (EDS)	*EuclideanDistanceSimilarity*
Pearson Correlation (PCS)	*PearsonCorrelationSimilarity*
Tanimoto Coefficient (TCS)	*TanimotoCoefficientSimilarity*
Uncentered Cosine (UCS)	*UncenteredCosineSimilarity*
City Block (CBS)	*CityBlockSimilarity*
Log Likelyhood (LLS)	*LogLikelihoodSimilarity*
Spearman Correlation (SCS)	*SpearmanCorrelationSimilarity*

4.4 Results of Performance & Quality Assessment

Our observation about the performance and quality of different similarity measure algorithms in recommending items is given below. Abbreviated names of the similarity measures have been used here (as mentioned in Table 3). Figure 4-1 shows the recommendation time for recommending top 3 items.

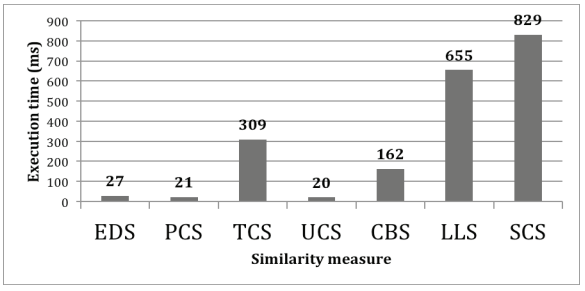

Figure 4-1: Average execution time taken by different similarity measure algorithms for recommending items to a user

It was observed that Euclidean distance, Pearson correlation and un-centered cosine similarity measure performed far better than other similarity measures under the situation described above. The average execution time for Spearman correlation was worst followed by Log likelihood and Tanimoto coefficient measures.

As mentioned in section 3.3, user-based recommendation uses three key steps for recommending items to a user – 1) measurement of similarity between the given user and other users, 2) detection of users who belong to neighborhood of the given user with respect to their similarity scores 3) recommendation of items from the list of preferred items of users in neighborhood.

It is surprising to see that recommendation using City Block similarity took more time than recommendation using Euclidean distance similarity.

If items are considered as dimensions and preferences are considered as points along those dimensions, a distance is computed using all items where both users have expressed preferences for that item. Similarity may be computed as 1/(1+distance), so that the resulting score is within 0 and 1. In

Mahout different approach is followed for computing Euclidean distance and City block distance similarities. In Euclidean distance similarity preference arrays are used for getting preferences of users and Euclidean distance between points along dimensions are computed using the preference arrays. On the other hand, City Block similarity is computed by getting the intersection between the sets of preferred items of users. Computational complexities of Euclidean distance similarity, Pearson correlation similarity, Un-centered cosine similarity measures are equivalent and all three of them use preference arrays in Mahout. Computational complexities of Tanimoto coefficient similarity and Log likelihood are higher than Euclidean distance similarity etc. due to use of intersection operations on sets of preferred items, for computation of similarity scores. Spearman correlation similarity use preference arrays in Mahout to store preferences of users, but its computation complexity become higher due to the computation of Pearson correlation between relatively ranked preference values. In Spearman correlation similarity, user preferences are sorted first and then ranked.

4.5 Evaluation Scores of Similarity Measures

Evaluation scores (MAE) of different similarity measures are shown below. Figure 4-2 shows the quality of the similarity measure algorithms when 90% of the input data was used from training and 10% of the input data was used for testing.

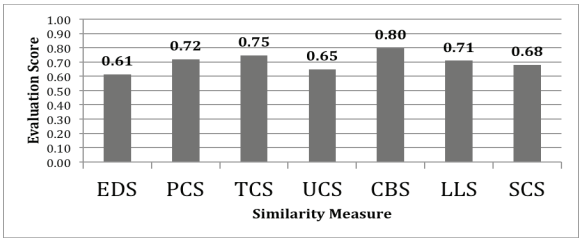

Figure 4-2: Evaluation scores (MAE) of different similarity measures for training data of 90% and testing data of 10%

From the results it is observed that Euclidean distance provides best quality in recommendations followed by Un-centered cosine similarity and Spearman correlation similarity measure algorithms. In case of Euclidean distance measure average difference of the estimated preference rating of test data from the actual preference rating is 0.61 in a rating scale of 1-5. It means EDS provides better estimates for preference scores leading to better recommendations compared to recommendation systems that use other similarity measures.

Detailed assessment of quality of different similarity measures is shown in Figure 4-3 for different training and testing data percentages.

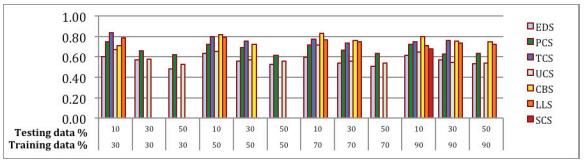

Figure 4-3: Evaluation scores of different similarity measures for different training and testing data %

In general it is observed that Euclidean distance, Pearson correlation and Un-centered cosine similarity measures are consistently performing well and Euclidean distance similarity measure provides slightly better performance compared to Un-centered similarity and Pearson coefficient similarity measures. There are some missing values in the chart for some similarity measures as the algorithms failed to produce any score after executing for very long time and exhausting system resources (heap space) in the computation.

Figure 4-4 shows the evaluation scores (MAE) of Euclidean distance measure for different training and testing set (percentage of total data set)

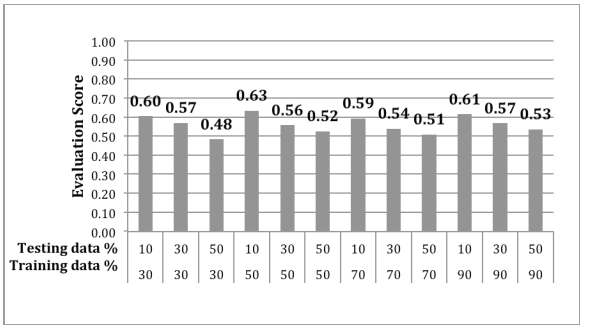

Figure 4-4: Evaluation scores (MAE) of Euclidean Distance Similarity measure for different training and testing data %

4.6 Precision and Recall Scores of Similarity Measures

Precision and recall scores of different similarity measures are shown in Figure 4-5. Higher the precision and recall scores, better the quality of corresponding recommender.

It is observed that Euclidean distance similarity measure provides very high quality results compared to other similarity measures.

The details of Spearman correlation measure is missing as the precision and recall values could not be generated as the Spearman correlation algorithm was taking enormously long duration and exhaustively high amount of system resources.

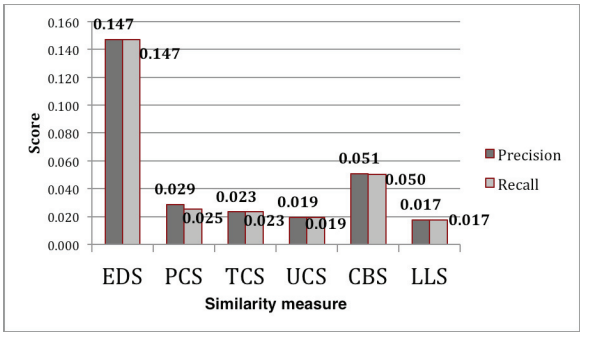

Figure 4-5: Precision and Recall scores of different similarity measures while generating top 2 recommendations

5. CONCLUSION

Recommendation systems play a very important role in e-commerce, social networking etc. Similarity measurement is a key aspect in any recommendation system and performance of a recommendation system is highly dependent on the performance of the similarity measurement steps. In this paper we have assessed the performance and quality of different similarity measuring algorithms in collaborative filtering process. From our assessment it is observed that Euclidean Distance Measure

consistently performs well and produces better quality results compared to other similarity measures.

6. ACKNOWLEDGEMENTS

Thanks to Prof. Pabitra Mitra, Associate Professor, Computer Science & Engineering in Indian Institute of Technology, for motivating, supporting and guiding in this work! Thanks to Prof. Indranil Sen Gupta, Professor, Computer Science & Engineering in Indian Institute of Technology for providing the opportunity to work on this experiment.

7. REFERENCES

[1] Owen S., Anil R., Dunning T. and Friedman E.: "Mahout In Action", 2012. Manning Publications Co. ISBN 978-1-9351-8268-9

[2] Adomavicius G. and Tuzhilin A.: "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions". IEEE Transactions on Knowledge and Data Engineering, 17(6): 734–749, 2005.

[3] Amatriain X., Lathia N., Pujol J.M., Kwak H. and Oliver N.: "The wisdom of the few: A collaborative filtering approach based on expert opinions from the web". In Proc. of SIGIR '09, 2009.

[4] Amatriain X., Pujol J.M., Tintarev N., and Oliver N.: "Rate it again: Increasing recommendation accuracy by user re-rating". In Proc. of RecSys '09, 2009.

[5] Herlocker J.L., Konstan J.A., Borchers A. and Riedl J.: "An Algorithmic Framework for Performing Collaborative Filtering". In Proc. of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval, 1999.

[6] Ricci F., Rokach L. and Shapira B.: "Introduction to Recommender Systems Handbook" (http://www.inf.unibz.it /~ricci/papers/intro-rec-sys-handbook.pdf), Recommender Systems Handbook, Springer, 2011.

[7] Melville P. and Sindhwani V.: "Recommender Systems" (http://www.prem-melville.com/publications/recommender- systems-eml2010.pdf), Encyclopedia of Machine Learning, 2010.

[8] Herlocker J.L., Konstan J.A., Borchers A., and Riedl J.: "Evaluating collaborative filtering recommender systems", January 2004. (http://portal.acm.org/citation.cfm?id=963772). ACM Trans. Inf. Syst. 22 (1): 5–53.

[9] Bhasker B., Srikumar K.: "Recommender Systems in E-Commerce", 2010. (http://www.tatamcgrawhill.com/html/9780070680678.html). CUP. ISBN 978-0-07-068067-8.

[10] Shapira B., Rokach L.: "Building Effective Recommender Systems", June 2012. (http://www.springer.com/computer/ai/book/978-1-4419-0048-7). ISBN 978-1-4419-0047-0.

[11] Jannach D., Zanker M., Felfernig A., Friedrich G.: "Recommender Systems: An Introduction". (http://www.cambridge.org/uk/catalogue /catalogue.asp?isbn=9780521493369). CUP. ISBN 978-0-521-49336-9.

[12] Alan Said, Alejandro Bellogín: "Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks". In Proc. of RecSys'14, 2014.

[13] G.Shani, A.Gunawardana: "Evaluating Recommendation Systems".