Improving item-based recommendation accuracy with user's preferences on Apache Mahout

Ammar Jabakji
Management Information Systems
Kadir Has University,
Istanbul, Turkey
ammarjabakji@gmail.com

Hasan Dağ
Management Information Systems
Kadir Has University,
Istanbul, Turkey
hasan.dag@khas.edu.tr

***Abstract*—Recommendation systems play a critical role in the Information Science application domain, especially in e-commerce ecosystems. In almost all recommender systems, statistical methods and machine learning techniques are used to recommend items to the users. Although the user-based collaborative filtering approaches have been applied successfully in many different domains, some serious challenges remain especially in regards to large e-commerce sites, for recommender systems need to manage millions of users and millions of catalog products. In particular, the need to scan a vast number of potential neighbors makes it very hard to compute predictions. Many researchers have been trying to come up with solutions like using neighborhood-based collaborative filtering algorithms, model-based collaborative filtering algorithms, and text mining algorithms. Others have proposed new methods or have built various architectures/frameworks. In this paper, we proposed a new data model based on users'preferences to improve item-based recommendation accuracy by using the Apache Mahout library. We also present details of the implementation of this model on a dataset taken from Amazon. Our experimental results indicate that the proposed model can achieve appreciable improvements in terms of recommendation quality.**

***Keywords*-Recommendation Systems; Collaboration Filtering; Mahout; Mean Absolute Error (MAE).**

I. Introduction

Studies have shown that word of mouth is still the most powerful marketing tool - even in an online age when a friend telling us about something interesting, we listen to him/her with open ears. Many people still make their decision regarding a purchase based on the actions of the others. It is safe to say that when we see most of the people recommending an item or a product and saying they have to have it, we tend to feel in the same way [1].

Product reviews are acknowledged to have great influences on customer buying behavior, to a certain extent, they can be considered a new electronic form of word-of-mouth. Marketers enable and encourage consumers to post product reviews and opinions on their e-retail sites in a form of ratings and comments [2, 3].

The available data on the internet is increasing at immense speeds, and the need to extract useful information from big date has emerged. Recommender systems provide customers with easy access to products/services and help boost e-commerce sales based on users'browsing history, searches, purchases, and preferences.

In the past 20 years, recommender systems (RS) have developed ways of finding products and information. They help consumers by selecting products they will probably like or might buy based on analyzing and discovering the patterns of other customers'behavior. Amazon uses different RS techniques to recommend new products to their customers. Facebook, Twitter, and Linkedin are using RS techniques to suggest people we might know. TripAdvicor is another company that has taken advantage of recommender systems to give advice on a wide variety of travel choices around the globe, and Eharmony which helps to match people together.

Among the widely used recommendation systems, collaborative filtering (CF) techniques are proven efficient. The two main collaborative filtering techniques are user-based and item-based. A recent survey, which is conducted inside recommendation system, has concluded that item-based CF provides better recommendations [4]. Along with the importance of the survey, some other studies have shown that item-based CF algorithm is of superior quality than user-based CF algorithm. In addition, it is claimed that it is scalable over very large datasets [5, 6, 7, 8].

In late 2006 Netflix announced a prize of 1 million dollars to the first team or person that could improve the accuracy of its movie recommendation system by 10 percent [9]. Since then, the challenge of improving the accuracy of recommendation systems has raised noticeable attention in the research community.

The aim of the present work is to construct a model that enhances e-commerce recommendation accuracy by using the helpfulness score of consumer product reviews. In particular, we aim to improve item-based recommendations for Amazon.com taking into account the rating system and users' opinions toward reviews of products. Also, we test and evaluate the proposed model and compare its results to six similarity metrics.

The proposed model takes three inputs; the total number

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

of feedbacks, the number of helpful feedbacks, and the rating given by a customer to a particular item or product. Then, it calculates a new rating called the adjusted rating, taking into account other factors, such as customer dissatisfaction to other reviews.

We empirically evaluate the proposed model on e-commerce review dataset from amazon.com. Experimental results show that the data model can improve the e-commerce quality of recommendation for specific similarity measures.

This paper is organized as follows: After a brief introduction to the topic, we give a short literature review in Section 2. Followings are a general overview of recommendation systems including a special attention to the Mahout's similarity measure and evaluation of recommender system. The proposed data model is given in Section 4. We describe our datasets, system configuration, similarity measures used in the assessment, and test results in Section 5. We present discussion and concluding remarks in Sections 6 and 7 respectively.

II. Literature review

We briefly review the related work and research literature. First, we outline the previous methods utilized to improve item-based recommender systems. Second, we summarize the approaches that take into account the helpfulness of consumer product reviews.

Sarwar et al. [5] analyze different item-based recommendation generation algorithms and present a new algorithm for CF-based recommender systems.Their results show that item-based techniques hold the promise of allowing CF-based algorithms to scale to large data sets and at the same time produce high-quality recommendations. Later, in 2003, in the paper by Linden et al. [7], it is stated clearly that "At Amazon.com, we use recommendation algorithms to personalize the online store for each customer." They conclude that item-to-item collaborative filtering is able to react immediately to changes in a user's data, and makes compelling recommendations for all users regardless of the number of purchases and ratings. Additional studies conclude that the item-based algorithms have superior recommendation quality over user-based algorithms, especially when we are dealing with scalability, real-time performance, and computational complexity as given by Linden et al. [7], Papagelis and Plexousakis [8], Ricci et al. [10].

As a basic definition, the number of helpful feedback over the total number of feedbacks is frequently called the helpfulness of on-line user reviews by previous studies [11, 12, 13, 14, 15]. Raghavan et al. [16] integrate the helpfulness scores of product reviews with probabilistic matrix factorization to improve the performance of recommender systems. Wang et al. [17] investigate the dual roles of users in the recommender systems, the first as a reviewer and the second as a rater who rates the helpfulness scores of reviews. Also, the author proposes a framework using matrix factorization method to exploit the dual roles of users. Some work has been done to recommend useful product reviews. As an example, Zhang and Tran [18] propose an information gain approach for modeling the helpfulness of on-line product reviews. Other studies like Ghose and Ipeirotis [19] examine the helpfulness and economic impact of product reviews by using text mining and reviewer characteristics.

III. Preliminaries

A. Recommendation Systems

Recommender systems are software tools and machine learning techniques that provide suggestions for items which might be interesting to other users [20].

Recommendations are an integral part of our daily experience on the Internet. Products are recommended to us on an e-commerce site, news items on a news portal, and videos are recommended on sites such as YouTube. Filtering out relevant information is essential to make the right decisions and realize new information. Recommender systems usually produce a list of recommendations by using one of two algorithms.

- Collaborative filtering
- Content-based filtering

However, it is possible to combine both techniques together it is called as hybrid recommendation systems. Collaborative filtering is a popular recommendation algorithm that bases its predictions and recommendations on either the similarity between users past behavior or the similarity between items in the system. The basic idea behind it is that if many users shared the same interests in the past they might also have similar tastes in the future. In this method a recommendation model has to be built based on similar behavior between users such as browsing or purchasing same products, giving almost identical ratings to items. The recommendations are then automatically generated for items that a user has not yet rated or given any preference for. Basically, this approach is based on collecting and analyzing a large amount of user data. The model built using past behavior of the users can then be used to recommend new items to them [5]. Content-based filtering approaches utilize a series of discrete characteristics of an item, in order to recommend additional items with similar properties. This technique requires a lot of effort for feature extraction and textual similarity in metadata, especially when the item set is huge like that of Amazon e-commerce site. And it is also relatively less precise than collaborative filtering approaches [21].

B. Mahout

Apache Mahout is an open source machine learning library that produces free implementations of both distributed (MapReduce) and non-distributed algorithms focused mainly in the areas of recommendation, clustering, and classification [22].

Mahout recommenders support many different similarity and neighborhood formation calculations. Recommendation prediction algorithms include item-based, user-based, and Slope-One and Singular Value Decomposition (SVD). It also incorporates Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) evaluation methods. Mahout is readily extensible and provides a wide range of Java classes for customization. It had reached version 12.2 at the time of writing. In this work, we conducted the study using non-distributed mode for item-based recommendation.

The building elements of an item-based recommender in Mahout are as follows.

- DataModel: Implementations of this method represent a repository of information about users and their associated preferences for items.
- ItemSimilarity: Implementations of this method define a notion of similarity between two items. It should return values in the range -1.0 to 1.0, with 1.0 representing perfect similarity.
- Recommender: Implementations of this method can recommend items for a user [23].

C. Similarity Measures

One critical step in the item-based CF algorithm is to compute the similarity between items and then to select the most similar items. There are a number of different ways to compute the similarity between items. We summarize six such methods for the sake of coherence. These similarity measures are based on how much the ratings by common users for a pair of items deviate from average ratings for those items. These are Euclidean distance similarity, Pearson correlation similarity, Uncentered cosine similarity, Tanimoto coefficient, Log-likelihood similarity, and City-block similarity.

1) Pearson correlation similarity: The Pearson correlation coefficient is a measure of the strength of linear relationship between two variables. The correlation between two items i and j is computed by the following:

$$sim(i,j) = \frac{\sum_{u \in U}(r_{u,i}-\bar{r_i})(r_{u,j}-\bar{r_j})}{\sqrt{\sum_{u \in U}(r_{u,i}-\bar{r_i})^2}\sqrt{\sum_{u \in U}(r_{u,j}-\bar{r_j})^2}} \quad (1)$$

where: $r_{u,i}$ is the rating of user u on item i, r_i is the average rating of the i-th item [5].

2) Euclidean distance similarity: The Euclidean distance measure is computed as *1/ (1+d)*, where *d* is the Euclidean distance between two user points. Larger values mean more-distant or less similar. It computed using the following formula.

$$d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + ... + (p_n-q_n)^2} \quad (2)$$

3) Uncentered Cosine similarity: In this case, two items are thought of as two vectors in the m dimensional user-space. The similarity between them is measured by computing the cosine of the angle between these two vectors. $0°$ means that the two items are similar. Similarity between items i and j, denoted by $sim(i,j)$ is given by

$$sim(i,j) = cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\left\|\vec{i}\right\|_2 * \left\|\vec{j}\right\|_2} \quad (3)$$

where "."denotes the dot-product of the two vectors [5].

4) Tanimoto coefficient : Also called Jaccard similarity coefficient; Interesting, the rating score is ignored in this similarity measure only that the user expresses a preference. It uses the ratio of the intersecting items to the union set as the measure of similarity. Thus it equals to zero if there are no overlap between items and equals to one if all products intersected.

$$t = \frac{N_c}{N_a = N_b - N_c} \quad (4)$$

where:
N_a is the number of users that rated item a.
N_b is the number of users that rated item b.
N_c is the number of users that rated both [24].

5) Log likelihood similarity: Log-likelihood-based similarity is like the Tanimoto coefficient-based similarity. It's another metric that doesn't take into account of individual preference values. Which based on the number of products in common between two users, similar to Tanimoto coefficient, but its value is more valuable of how unlikely it is for two users to have so much overlap, given the total number of items in the dataset and the number of items each user has a preference for [25].

D. Evaluation of Recommender

Statistical accuracy metrics measure the closeness between recommendation results provided by the system and the numerical ratings entered by the user for the same items. Recommendation accuracy has been evaluated in many different ways. One popular way is using mean absolute error (MAE). It used to measure the ability of a system to correctly predict a user's preference for a particular item [26].

MAE is a metric of the deviation of recommendations from their true user-given values. For each ratings-prediction pair p_i, q_i this metric calculates the absolute error between them i.e. p_i, q_i equally. The MAE is computed by first summing these absolute errors of the N corresponding ratings-prediction pairs and then computing the average. This can be illustrated as follows;

$$MAE = \frac{\sum_{i=1}^{k} |p_i - q_i|}{N} \quad (5)$$

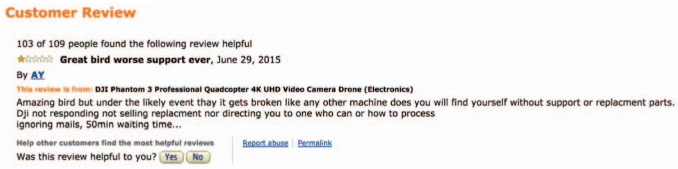

Figure 1. Amazon Sample Review

Lower MAE score means more accurate recommendations. Root Mean Squared Error (RMSE), is another statistical accuracy metric. Decision support accuracy evaluates how effectively recommendations help a user select high-quality items from the huge group of items [27]. MAE metric was used in our evaluation process of recommendation.

IV. The Proposed Method

The proposed method and the process of building a new data model aims to improve item-based recommendation results. The primary way to measure the accuracy of a recommendation system is to compare the evaluation score (MAE) between the original rating values given by a user and the adjusted data where we have implemented our data model.

We modify the original rating that the user had given to a product and we name it the adjusted rating. The modification can be done by adding one to (or subtracting one from) the original rating value or just keeping the value as it is.

When we look at Amazon reviews, we can see that each customer has given a rating score from 1 to 5 to a product. Importantly, we also notice a question at the bottom of each review ”Was this review helpful to you?”. An example of this can be seen in Fig 1 where 103 of 109 shoppers find the product review to be helpful.

The number of users finding a review helpful out of the total number of feedbacks for a review is called the helpfulness score, which is, 94.49 in our example. In our proposed model, we take advantage of the helpfulness score to improve recommendation accuracy. Another factor that we take into account is positive and negative reviews. A positive consumer review is given when an item is rated 5 or 4 out of 5 stars by at least one reviewer. A negative review, on the other hand, is where at least one reviewer has rated the item a 1 or a 2 out of 5 stars. Taken together, we calculate the new rating or the adjusted rating as follows;

$$\hat{r} = \begin{cases} d + r, & r < 3 \\ |d - r|, & r > 3 \end{cases} \tag{6}$$

where:
$\hat{r}$: is the adjusted rating
r : is the original rating given by the user
d : is dissatisfaction score
The dissatisfaction score, which takes a value from 0 to 1, can be calculated as follows.

$$d = 1 - (\frac{h}{t}) \tag{7}$$

where:
h : is the number of helpful feedbacks
t : is the total number of feedbacks

UserID	ProductID	Total Feeds	Helpful Feeds	Rating	Adjusted Rating
16605	2	11	9	5	5
16791	2	38	21	1	1
16831	2	35	17	1	2
16905	2	9	5	3	3
16972	2	24	14	5	5
17020	2	0	0	5	5
17029	2	20	12	2	2
17223	2	7	3	4	3
17328	2	8	3	3	3

Figure 2. Sample Example

In our model, when we get a negative rating while the dissatisfaction score is more than 0.5 i.e. the majority of reviewers do not agree with the negative rating, we add the dissatisfaction score to the original rating and round this to the nearest integer. On the other hand, when we a get positive rating while the dissatisfaction score is more than 0.5 i.e. the majority of reviewers do not agree with the positive rating. We subtract the dissatisfaction score from the original rating and round this to the nearest integer. But when the rating is 3 and the dissatisfaction score is high, we can not increase or decrease the value. Therefore, we don’t apply our model on a rating of 3. More examples of the proposed method being utilized can be found in Fig. 2.

The item-based algorithm works in four main steps. First, computes similarities over all pairs of products using one of the similarity measures. Second, the engine determines the most similar products relevant to the target user. Third, the engine fills the gaps, computes the predictions for all similar products that the target user has no rating for. And then, generates a list of recommendations based on high prediction scores.

V. Experiments and Evaluation

In this section, we present the results of the quality assessment of six different similarity measures. We have performed this assessment using Mahout’s item-based recommendation algorithm.

A. Datasets

The experimental data comes from amazon product reviews and consists of 4 ratings provided by 1.5 million users for over 1 million items, from different categories. The dataset has been used before for opinion spam and fake review detection [28, 29]. To evaluate the accuracy of item-based recommendation algorithm, we extracted six different datasets of different sizes from the original large dataset as follows; 100k, 500k, 1M, 2M, 3M, and 4M. The datasets contain meta data, e.g. member id, product id, date, the number of helpful feedbacks, the total number of feedback, rating, the title of the review, the body of the review and the date.

B. System Configuration

We carried out the assessment of the proposed method using a MacBook Pro, for which the system configuration is given in Table I.

Table I
SYSTEM AND OTHER CONFIGURATION

Processor	2.2 GHz Intel Core i7
RAM	16 GB 1600 MHz DDR3
Operating System	OS X El Capitan 10.11.2
Java version	1.8
Mahout	Apache Mahout 0.11.1
Dataset	Amazon reviews dataset
Rating scales	1 to 5

C. Similarity Measures Used in Assessment

The item-based APIs of Mahout, shown in Table II, are used during the assessment of the proposed method. Six different similarity measurement methods are compared for the proposed method.

Table II
MAHOUT APIS USED FOR SIMILARITY MEASUREMENT

Similarity Measure	Mahout API
Pearson Correlation	PearsonCorrelationSimilarity
Euclidean Distance	EuclideanDistanceSimilarity
Uncentered Cosine	UncenteredCosineSimilarity
Tanimoto Coefficient	TanimotoCoefficientSimilarity
Log Likelihood	LogLikelihoodSimilarity
City Block	CityBlockSimilarity

D. Evaluation Score of Similarity Measures

We compare the evaluation score (MAE) of recommendation between two datasets, each one consists of 2 million records (rows) 3. The first dataset is the original data, i.e., without any modification, and the dataset modified with respect to the proposed method. We use the term ”Adjusted” our presentation for the later. The comparison results about the quality of the six different similarity measure algorithms in recommending items to users is given in Figures 4 - 9. A lower score is better as that indicates that estimates are closer to actual preference values.

Studies have shown that the optimum value for training dataset is 80% for evaluating CF-based recommender

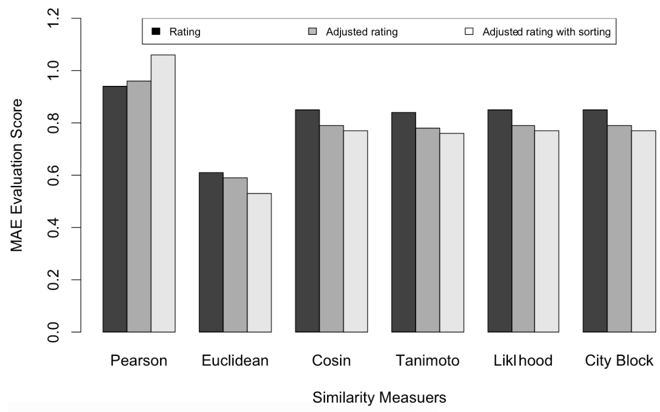

Figure 3. Evaluation scores (MAE) of different similarity measures for training data to 80% and testing data to 20%

systems [5]. For this reason, we chose training/test ratio as 80/20 for all assessments. It is observed that Euclidean distance similarity measure performs far better than other similarity measures. Most importantly, our proposed model gives smaller error score for all similarity measures except Pearson correlation similarity as shown in Fig. 3.

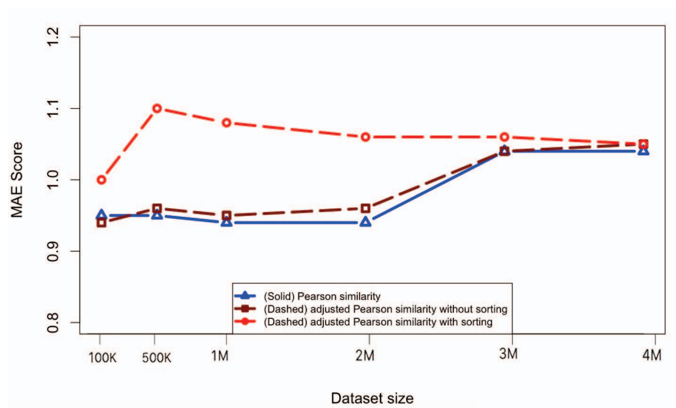

Figure 4. Evaluation scores (MAE) of Pearson Correlation Similarity for training data to 80% and testing data to 20%

As can be seen in Fig. 4 the adjusted Pearson correlation similarity falls below evaluation expectations. However, by increasing the size of dataset, the results tend to be close to each other. The recommender estimates a preference that deviates from the actual preference by an average of 0.01. In other words, our method fails using this measure with an average of 1 percent deterioration. Consequently, we can say that it is not recommended to adopt our model for item-based recommendation system using Pearson correlation similarity measure.

As shown in Fig. 5, the adjusted Euclidean distance similarity outperforms the Euclidean distance similarity. Specifically, the recommender estimates a preference that deviates from the actual preference by an average of 0.02. This relates to about 2 percent improvement. However, when we increase the size of dataset the results tend to be close to each other.

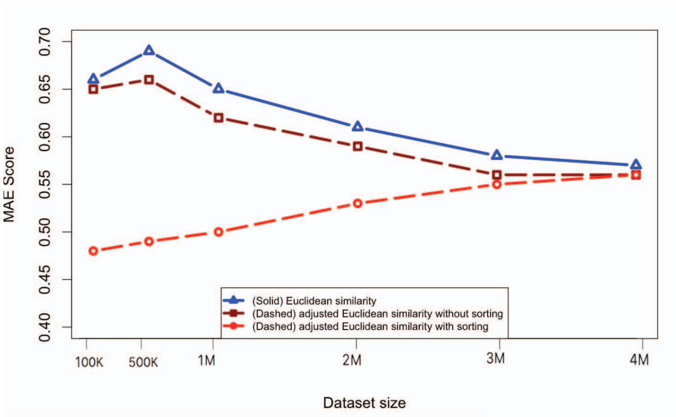

Figure 5. Evaluation scores (MAE) of Euclidean Distance Similarity for training data to 80% and testing data to 20%

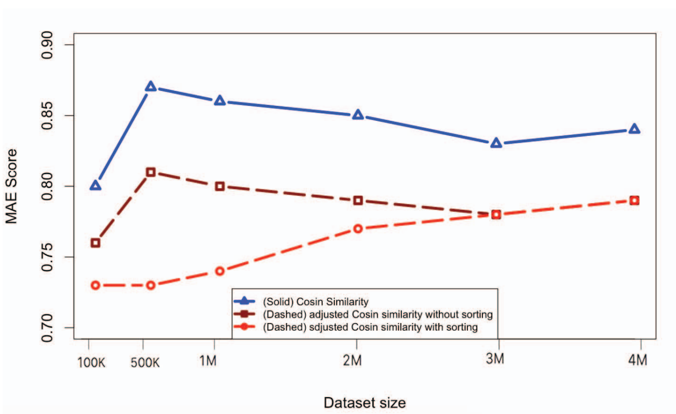

Figure 6. Evaluation scores (MAE) of Uncentered Cosine Similarity for training data to 80% and testing data to 20%

As is clear in Fig. 6 the test results for the adjusted Cosine similarity are much better than the previous two measure. Therefore, our method outperforms the UnCentered Cosine similarity, and the recommender estimates a preference that deviates from the actual preference by an average of 0.053. In other words, we get more accurate recommendation results by 5.3 percent.

The Fig. 7 shows the adjusted LogLikilihood similarity, which is performing better than to the orginal value of the same metric. The recommender estimates a preference that deviates from the actual preference by an average of 0.053. In other words, we get more accurate recommendation results by 5.3 percent.

The adjusted Tanimoto similarity gives notable results and outperforms the Tanimoto similarity, as can be seen in Fig.

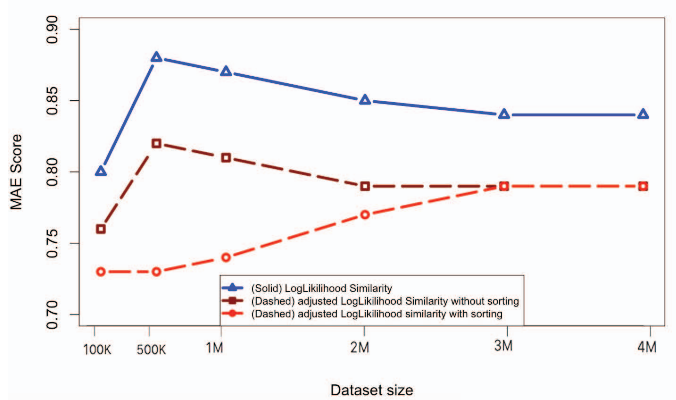

Figure 7. Evaluation scores (MAE) of LogLikilihood Similarity for training data to 80% and testing data to 20%

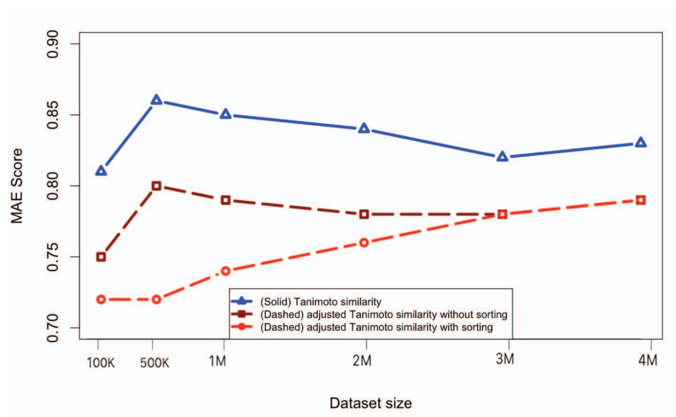

Figure 8. Evaluation scores (MAE) of Tanimoto Similarity Similarity for training data to 80% and testing data to 20%

8. The recommender estimates a preference that deviates from the actual preference by an average of 0.053. In other words, we get more accurate recommendation results by 5.3 percent.

The adjusted City-block similarity, shown in Fig. 9, also gives positive results and outperforms the City-block similarity. The recommender estimates a preference that deviates from the actual preference by an average of 0.051. In other words, we get more accurate recommendation results by 5.1 percent.

VI. Discussion

We make the following observations, based on the experimental results. The proposed model which is based on item-based CF algorithms provide good results with 4.6 percent average improvement. We also notice that the improvements in recommendation quality is inconsistent over different dataset sizes, and provides more accurate recommendations when the dataset size less than 3 million ratings. Another important point is that our model does not perform well when using the Pearson similarity measure.

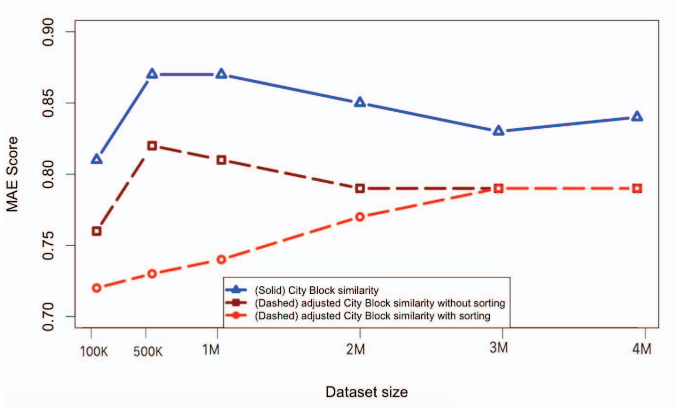

Figure 9. Evaluation scores (MAE) of City Block Similarity Similarity for training data to 80% and testing data to 20%

We think that the unexpected difference in evaluation score MAE between the sorted dataset and the unsorted dataset is attributable to data sparsity. When the dataset size is small and sorted by userID, it becomes less sparse; consequently, the results become more accurate. In other words, the MAE score is less accurate, until it reaches a point where the sparsity level is equal to the level of unsorted dataset. One reason for the sparsity levels is that each user gives a relatively small number of reviews.

VII. Conclusion

Given the importance of recommender systems, this study aims to improve the accuracy of recommendation results in the e-commerce domain. Here, we propose a new data model using the rating system of consumer product reviews. We also evaluate and measure the performance of recommendations with Mahout item-based similarity measures using a real-word dataset taken from Amazon and have found that our proposed model gives accurate recommendations by applying both machine learning techniques and statistical methods. The evaluation outcomes reveal that our model provides more accurate results and an average recommendation improvement of 4.6 percent. Some may argue that this score is low, but in fact and as we stated previously even small improvements in e-commerce recommender systems may result in highly appreciable increases in sales.

Additionally, it was observed that the Euclidean distance similarity measure is the best performing metric among the other Mahout item-based similarity measures. And Pearson similarity measure is the worst performer.

Reviews, which are also called on-line opinions, and ratings of those reviews have a great impact on a purchase decision. Therefore, taking these factors into consideration and analyzing users' reviews using text mining algorithms by extracting keywords from them is potentially valuable for e-commerce systems. It is important to mention that users' opinions of other user' reviews and users' opinions of other user' ratings could be used as an input for recommender systems. By doing so, it could be possible to gain a better understanding of what real value a specific product/item should deserve as a rating score. Eventually, utilizing this method could help improve the quality of predictions or recommendations that the user will receive.

References

[1] M. Marrs. (2014) Amazon review guide: How to get amazon customer. [Online]. Available: http://www.wordstream.com/blog/ws/2014/04/10/amazon-reviews

[2] D. Godes and D. Mayzlin, "Using online conversations to study word-of-mouth communication," *Marketing science*, vol. 23, no. 4, pp. 545–560, 2004.

[3] S. Sen and D. Lerman, "Why are you telling me this? an examination into negative consumer reviews on the web," *Journal of interactive marketing*, vol. 21, no. 4, pp. 76–94, 2007.

[4] V. Patel, "Inside recommendation system: Survey, research area," *International Journal of Engineering Sciences & Research Technology*, vol. 1, no. 4, pp. 488–491, 2015.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 285–295.

[6] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.

[7] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76–80, 2003.

[8] M. Papagelis and D. Plexousakis, "Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 7, pp. 781–789, 2005.

[9] K. Hafner, "if you liked the movie, a netflix contest may reward you handsomely," *New York Times*, vol. 2, 2006.

[10] F. Ricci, L. Rokach, and B. Shapira, *Introduction to recommender systems handbook*. Springer, 2011.

[11] M. Li, L. Huang, C.-H. Tan, and K.-K. Wei, "Helpfulness of online product reviews as seen by consumers: Source and content features," *International Journal of Electronic Commerce*, vol. 17, no. 4, pp. 101–136, 2013.

[12] Q. Cao, W. Duan, and Q. Gan, "Exploring determinants of voting for the "helpfulness" of online user reviews:

A text mining approach," *Decision Support Systems*, vol. 50, no. 2, pp. 511–521, 2011.
[13] S. M. Mudambi and D. Schuff, "What makes a helpful review? a study of customer reviews on amazon. com," *MIS quarterly*, vol. 34, no. 1, pp. 185–200, 2010.
[14] S. Lee and J. Y. Choeh, "Predicting the helpfulness of online reviews using multilayer perceptron neural networks," *Expert Systems with Applications*, vol. 41, no. 6, pp. 3041–3046, 2014.
[15] N. Korfiatis, E. García-Bariocanal, and S. Sánchez-Alonso, "Evaluating content quality and helpfulness of online product reviews: The interplay of review helpfulness vs. review content," *Electronic Commerce Research and Applications*, vol. 11, no. 3, pp. 205–217, 2012.
[16] S. Raghavan, S. Gunasekar, and J. Ghosh, "Review quality aware collaborative filtering," in *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 2012, pp. 123–130.
[17] S. Wang, J. Tang, and H. Liu, "Toward dual roles of users in recommender systems," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 1651–1660.
[18] R. Zhang and T. Tran, "An information gain-based approach for recommending useful product reviews," *Knowledge and Information Systems*, vol. 26, no. 3, pp. 419–434, 2011.
[19] A. Ghose and P. G. Ipeirotis, "Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 10, pp. 1498–1512, 2011.
[20] T. Mahmood and F. Ricci, "Improving recommender systems with adaptive conversational strategies," in *Proceedings of the 20th ACM conference on Hypertext and hypermedia*. ACM, 2009, pp. 73–82.
[21] J. Withanawasam, *Apache Mahout Essentials*. Packt Publishing Ltd, 2015.
[22] G. Bamnote and S. Agrawal, "Evaluating and implementing collaborative filtering systems using apache mahout," in *Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on*. IEEE, 2015, pp. 858–862.
[23] C. Tiwary, *Learning Apache Mahout*. Packt Publishing, 2015.
[24] S. Mohanty, M. Jagadeesh, and H. Srivatsa, *Big Data imperatives: enterprise Big Data warehouse, BI implementations and analytics*. Apress, 2013.
[25] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in action*. Manning Shelter Island, 2011.
[26] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: an introduction*. Cambridge University Press, 2010.
[27] B. M. Sarwar, J. A. Konstan, A. Borchers, J. Herlocker, B. Miller, and J. Riedl, "Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system," in *Proceedings of the 1998 ACM conference on Computer supported cooperative work*. ACM, 1998, pp. 345–354.
[28] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 2008, pp. 219–230.
[29] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 191–200.