

# Performance Comparison of Similarity Measures Used in Recommendation Systems

Berna Seref<sup>1\*</sup> and Erkan Bostanci<sup>2</sup>

<sup>1</sup>Ankara University, Computer Engineering Department, Ankara, Turkey. Email: [bseref@ankara.edu.tr](mailto:bseref@ankara.edu.tr)

<sup>2</sup>Ankara University, Computer Engineering Department, Ankara, Turkey. Email: [ebostanci@ankara.edu.tr](mailto:ebostanci@ankara.edu.tr)

\*Corresponding Author

**Abstract:** There are more data on the web, thus it is hard to get relevant data and make good decisions. Recommendation systems provide suggestions to users about the various items. They are classified into four groups which are collaborative filtering, content-based filtering, knowledge-based recommender systems, and hybrid recommendation systems. There are some similarity measures such as Pearson Correlation, Euclidean, Uncentred Cosine, and LogLikelihood to calculate similarity between users or items. In this study, a user-based collaborative filtering recommendation system is developed on Eclipse platform using mahout library. To develop a recommendation system, different similarity measures such as Pearson Correlation, Euclidean, Uncentred Cosine, and LogLikelihood are used. After that, recommendation performances of them are compared. Movielens datasets are used to train and test the system. As a result, it is seen that while the best mean average error and the best root mean square error performances belong to Uncentred Cosine similarity measure, the best precision, recall, and f-measure performances belong to Pearson Correlation measurement.

**Keywords:** Euclidean, Loglikelihood, Pearson, Recommendation, Similarity.

## I. INTRODUCTION

Recommender systems are the systems that recommend an item or data using background data such as ratings from users of items and features of items in order to cope with information overload problem. This problem occurs when there is more data that exceed processing capability of the system [1]. As a result, it is hard to make efficient decisions [2]. On the other hand, people need intelligent techniques to filter and get relevant data. Recommender systems are applications that present intelligent suggestions on items and cope with information overload problem [1, 3, 4].

Recommender systems are classified into four groups which are based on used techniques and information to make

recommendations. These are collaborative filtering, content-based filtering, knowledge-based recommender systems, and hybrid recommendation systems [1, 5-7].

Collaborative filtering systems make recommendations only by using users' ratings [8]. They assume that users with similar interests and opinions tend to prefer similar items [1]. These systems are classified into two groups: memory-based and model-based algorithms. In the memory-based algorithm, similarities between users or items are calculated directly. On the other hand, in the model-based algorithm, firstly, predictive model from the user database is constructed, and secondly, it is used for making a prediction [9].

Content-based filtering systems recommend items with similar features that were much preferred by users in past [1].

Knowledge-based recommender systems make recommendations based on the knowledge which is based on how certain item features and attributes satisfy users' needs [1, 4].

Hybrid recommendation systems make recommendations using two or more techniques [1].

According to Gunawardana and Shani [10], recommendation systems have two tasks: the prediction task and the recommendation task. In the prediction task, user opinion such as ratings is predicted. In the recommendation task, relevant items are recommended to the users.

Computer technology is getting more developed day by day. As a result, use of the internet for reading news or articles and e-commerce rates have been increasing. People want to get relevant data as soon as possible. Therefore, they need recommendation systems which recommend relevant data such as books, DVDs, clothes, news, or articles. In order to get relevant data and make good decisions, lot of studies are carried on and lot of recommendation systems are developed.

Collaborative filtering recommendation systems have some challenges such as cold start problem, data sparsity, scalability, synonymy, gray sheep, and shilling attacks [11]. Cold start problem occurs when it is impossible to make a recommenda-

tion for a new user because of having insufficient information [12, 13].

Scalability is the ability to make the recommendation of a system in real time or near real time [12, 14-15]. Sparsity problem occurs when a new user or item enters the system. The system cannot make good recommendation because of having insufficient information [11]. Synonymy is the tendency of a number of the same or very similar items which are represented with different names or entities. Recommender systems are not able to discover these words and, as a result, the performance of collaborative filtering systems is decreasing [11]. Gray sheep problem occurs when there are people whose tastes are different from others in the system. These people may affect recommender systems in a negative way and may not get a good recommendation [16]. Attacks on recommender systems are called shilling attacks. Push and nuke attacks are the two most common types of these attacks. In a push attack, the attacker wants to make specified item to be recommended more often by injecting biased ratings. On the other hand, in nuke attacks, the attacker wants to make specified item less often recommended [17].

In this study, a user-based recommendation system is developed using mahout library on Eclipse platform. Pearson, Euclidean, Uncentred Cosine, and LogLikelihood similarity measures are used to calculate similarities between users. Recommendation performance of these measures are focused. MovieLens 100K Dataset which has 100,000 ratings and 1 million ratings of MovieLens 20M Dataset are used to train and test the system. While randomly 70% of the dataset is used as a training test, 30% of the dataset is used as a test set. Performance of the system is evaluated using mean average error, root mean square error, precision, recall, and f-measure performance evaluation criterions.

The rest of the paper is structured as follows: Section II gives information about studies carried on about recommender systems. Section III explains materials and methods. Section IV shows results and Section V concludes the paper.

## II. LITERATURE REVIEW

There are a lot of studies about recommender systems. In general, they use fuzzy or genetic algorithm to make recommendations.

For example, Siddiquee *et al.* [18] make recommendations using fuzzy logic. As a similarity measure, Euclidean Distance, Manhattan Distance, Pearson Coefficient, and Cosine Similarity measures are used. As a result, it is observed that results with Euclidean Distance show the best performance. On the other hand, Parvin *et al.* [19] make recommendations using genetic algorithm and trust statements. As a results, it is observed that the proposed two-step Collaborative Filtering method called TCFGGA shows better performance. Alhijawi and Kilani [20] used genetic algorithm in their studies. They proposed a

recommender system which is called Simgen. They calculate similarity between users using genetic algorithm. As a result, 46 and 38% improvements in prediction quality and performance, respectively, are observed when it is compared with other techniques. On the other hand, Verma *et al.* [21] and Jeon *et al.* [22] propose a recommendation system using fuzzy logic. Verma *et al.* [21] propose a recommender system based on collaborative filtering and fuzzy c-means clustering algorithms. This system handles sparsity and cold start problems. As a result, it is observed that fuzzy clustering algorithm shows better performance over k-means method. Jeon *et al.* [22] propose a prediction system based on collaborative filtering and fuzzy system. As a result, it is observed that fuzzy system complements defects of collaborative filtering.

There are some studies which combine fuzzy logic and genetic algorithm. For example, Shivhare *et al.* [23] propose a recommendation system based on fuzzy c-means clustering and genetic algorithm-based weighted similarity measure. After the experiments, it is observed that recommender system using fuzzy c-means clustering and weighted similarity measure with genetic algorithm shows better performance than the recommender system using weighted similarity measure with genetic algorithm.

Scalability is defined as ability of making recommendation of a system in real time or near real time [12, 14-15]. Shou-qiang and Ming [24] focus on this problem and propose paralleled transform based on the Hadoop MapReduce. In addition to it, they develop Hadoop cluster recommender system using hybrid collaborative filtering algorithm based on Genetic algorithm optimization. After the experiments, it is observed that performance of the system is improved.

Content-based filtering systems recommend items with similar features that preferred by users in the past [1]. Kim *et al.* [25] propose a recommender system based on content-based filtering technique and genetic algorithm. After the experiments, it is observed that the proposed system can recommend suitable items.

There are some techniques to calculate similarities between users and items such as Cosine-based similarity, correlation-based similarity, adjusted cosine similarity, Euclidean distance, Manhattan distance, and Minkowski distance. While Sarwar *et al.* [18, 26] design an item-based recommender system and use cosine-based similarity, correlation-based similarity and adjusted cosine similarity techniques. Roy and Kundu [18, 27] recommend a recommendation system using collaborative filtering and data clustering, and measure the similarities between users using Euclidean distance, Manhattan distance, and Minkowski distance.

Clustering algorithms are used to find similarity and nearest neighborhood in recommendation systems. Li and Zhou [21, 28] and Gong [9] use clustering algorithms in their study. Li and Zhou [21, 28] use content-based and collaborative filtering technique for a recommendation. Firstly, they discover similar

users using k-means clustering algorithm. Secondly, they find the contents which are rated high by users in the same cluster. Then, they add the content to the content list and use fuzzy c-means algorithm for the contents with the aim of finding contents in the same cluster which is requested. Gong [9] proposes a new collaborative recommendation system based on fuzzy similar-priority comparison and fuzzy clustering. While fuzzy similar-priority comparison is used to compute user similarity, fuzzy clustering technology is used to find nearest neighborhood. After that, recommendations are made. Experiments show that the system's recommendation quality and accuracy of the predicted values are increased.

Summary of used datasets and performance evaluation criterions in the literature are shown in Table I.

TABLE I: DATASETS AND PERFORMANCE EVALUATION CRITERIONS

| Paper                        | Dataset                               | Performance Evaluation Criterion                    |
|------------------------------|---------------------------------------|---|
| Siddiquee <i>et al.</i> [18] | Movielens                             | Root mean square error                              |
| Parvin <i>et al.</i> [19]    | FilmTrust and Epinions datasets       | Root mean square error and mean absolute error      |
| Alhijawi and Kilani [20]     | Synthetic data and Movielens datasets | Mean absolute error, precision, and recall          |
| Verma <i>et al.</i> [21]     | Movielens                             | Root mean square error and mean absolute error      |
| Jeon <i>et al.</i> [22]      | Dataset is offered by Netflix         | Root mean square error                              |
| Shivhare <i>et al.</i> [23]  | Movielens                             | Mean absolute error, precision, recall and coverage |
| Li and Zhou [21, 28]         | Movielens                             | Mean absolute error                                 |
| Gong [9]                     | Movielens                             | Mean absolute error                                 |

### III. MATERIALS AND METHODS

In this study, a user-based recommendation system is developed using mahout libraries on Eclipse platform. Movielens dataset is used to train and test the system. While 70% of the data is used as a training set, 30% of the data is used as a test set. All the experiments are carried on three times with training and test sets which are created randomly. After that, the average of the results is taken.

There are a lot of similarity measures to calculate similarity between users or items. Pearson Correlation, Cosine Vector

Similarity, Euclidean, LogLikelihood and Extended Jaccard Coefficient are some of them [29]. In this study, Pearson, Euclidean, Uncentred Cosine, and LogLikelihood similarity measures are used to calculate similarities between users. Mean average error, root mean square error, precision, recall, and f-measures are used as performance evaluation criteria.

#### A. Distance Measures

##### 1) Pearson Correlation

Pearson correlation measures the linear correlation between two vectors of ratings. It is formulated in Eq. (1) [9].

$$Sim(i, j) = \frac{\sum_{c \in I_{i,j}} (R_{i,c} - A_i)(R_{j,c} - A_j)}{\sqrt{\sum_{c \in I_{i,j}} (R_{i,c} - A_i)^2 * \sum_{c \in I_{i,j}} (R_{j,c} - A_j)^2}} \quad (1)$$

In Eq. (1),  $R_{i,c}$  is the rating of item  $c$  by user  $i$ .  $A_i$  is the average rating of user  $i$  for all the co-rated items.  $I_{i,j}$  is the items set both rated user  $i$  and  $j$ .  $R_{j,c}$  is the rating of item  $c$  by user  $j$ .  $A_j$  is the average rating of user  $j$  for all the co-rated items.

##### 2) Cosine Similarity

Cosine similarity focus on angle between two vectors of ratings. Smaller angle means better similarity. It is formulated in Eq. (2) [9].

$$Sim(i, j) = \frac{\sum_{k=1}^n R_{i,k} * R_{j,k}}{\sqrt{\sum_{k=1}^n R_{i,k}^2 * \sum_{k=1}^n R_{j,k}^2}} \quad (2)$$

In Eq. (2),  $R_{i,k}$  is rating of item  $k$  by user  $i$ .  $R_{j,k}$  is rating of item  $k$  by user  $j$ .  $n$  is the number of items co-rated by both users.

##### 3) Euclidean Distance Measurement

It is formulated in Eq. (3) [18].

$$D_{uv} = \sqrt{\sum_{i=1}^n |a_{ui} - a_{vi}|^2} \quad (3)$$

In Eq. (3), while  $U$  is set of all users,  $D_{uv}$  is distance between user  $u$  and  $v$  based on their movie ratings.  $a_{ui}$  and  $a_{vi}$  are ratings of user  $u$  and  $v$  for item  $i$ .

##### 4) LogLikelihood

If series  $y$  has Fourier coefficients  $(p_i, q_i)$ , periodogram of  $y$  is equal to  $a_i = p_i^2 + q_i^2$ . Loglikelihood is defined in Eq. (4) [30].

$$l(a) = \frac{\sum_{i=1}^{n-1} a_i}{2a_i} \log(2a_i) \quad (4)$$

##### 5) Manhattan Distance Measurement

Manhattan measurement measures dissimilarity between users. Dissimilarity between user  $u$  and  $v$  is defined in Eq. (5) [18].

$$Duv = \sum_{i=1}^n |aui - avi| \quad (5)$$

### 6) Uncentred Cosine Similarity

Uncentred cosine similarity measures cosine of the angle formed between the two preference vectors. In addition, instead of centering its data, similarity shifts the user's preference values to make each of their means 0 [31].

## B. Performance Evaluation Criteria

### 1) Root Mean Squared Error (RMSE)

It is defined in Eq. (6) [32].

$$RMSE = \sqrt{\frac{1}{N} * \sum_{u,i} (pi(u) - ri(u))^2} \quad (6)$$

In Eq. (6), while  $N$  is the total number of ratings,  $pi(u)$  is the predicted rating of user  $u$  on item  $i$ .  $ri(u)$  is actual rating. Smaller values of root mean square error means better performance [31].

### 2) Mean Average Error

In this study to calculate mean absolute error, mahout classes and methods are used. Average Absolute Difference Recommender Evaluator class, Abstract Difference Recommender Evaluator class and evaluate() method are one of them [33, 34].

This method evaluates the recommendation quality of the system. Lower values mean better performance. 0 means the best evaluation.

Recommender Evaluator calculates average absolute difference between predicted and actual ratings for users which is called mean average error [35].

### 3) Precision

Precision is defined in Eq. (7).

$$precision = \frac{tp}{tp+fp} \quad (7)$$

In Eq. (7),  $tp$  is the number of movies which are liked by active users and recommended in the recommended list.  $fp$  is the amount of movies which are unliked by active users, but not recommended in recommended list [36].

### 4) Recall

Recall is defined in Eq. (8).

$$recall = \frac{tp}{tp+fn} \quad (8)$$

In Eq. (8),  $tp$  is the number of movies which are liked by active users and recommended in the recommended list.  $fn$  is the amount of movies which are liked by active users, but not recommended in recommended list [36].

### 5) F-measure

F-measure is the harmonic mean of precision and recall. It is shown in Eq. (9) [37].

$$F - measure = \frac{2 * precision * recall}{precision + recall} \quad (9)$$

## C. Dataset

In this study, Movielens 100K and a part of Movielens 20M [38] datasets are used. Movielens 100K dataset consists of 100000 ratings from 1000 users on 1700 movies. All of the data of Movielens 100K are used. Movielens 20M dataset consists of 20 million ratings and 465000 tag applications applied to 27000 movies by 138000 users. In this study, 1 million of the data are used.

## IV. RESULTS

Results are shown in Tables II and III.

TABLE II: RESULTS FOR THE DATASET WITH 100000 RATINGS

|                        | Pearson   | Euclidean | LogLikelihood | Uncentred Cosine |
|------------------------|-----------|-----------|---------------|------------------|
| Mean Average Error     | 0.903667  | 0.894758  | 0.815834      | 0.830392         |
| Root Mean Square Error | 1.122772  | 1.123979  | 1.037744      | 1.04070          |
| Precision              | 0.015982  | 0.014580  | 0.000507      | 0.000253         |
| Recall                 | 0.0148900 | 0.012035  | 0.000508      | 0.000254         |
| F-measure              | 0.015417  | 0.013186  | 0.000508      | 0.000254         |

TABLE III: RESULTS FOR THE DATASET WITH 1000000 RATINGS

|                        | Pearson  | Euclidean | LogLikelihood | Uncentred Cosine |
|------------------------|----------|-----------|---------------|------------------|
| Mean Average Error     | 0.812165 | 0.789041  | 0.738430      | 0.728047         |
| Root Mean Square Error | 1.044703 | 1.03509   | 0.9579285     | 0.951552         |
| Precision              | 0.002436 | 0.005986  | 0.000013      | 0.000018         |
| Recall                 | 0.002442 | 0.005589  | 0.000010      | 0.000011         |
| F-measure              | 0.00244  | 0.005936  | 0.000011      | 0.000010         |

As it seen in Table II, the performance of Loglikelihood similarity measure is better than the others based on mean average error and root mean square error. On the other hand, performance of Pearson similarity measure is better than the others based on precision, recall, and f-measure performance evaluation criteria.

As it is seen in Table III, uncentred cosine similarity measure shows the best performance based on mean average error and root mean square error. The best precision, recall, and f-measure values are obtained using Euclidean similarity measure.

When Tables II and III are compared, it is seen that the best mean average error and root mean square error belong to uncentred cosine similarity measure. The best precision, recall, and f-measure belong to Pearson measurement.

## V. CONCLUSIONS

In this study, a user-based recommendation system is developed using Pearson, Euclidean, uncentred cosine, and LogLikelihood similarity measures. Recommendation performances of these similarity measurements are observed. Mean average error, root mean square error, precision, recall, and f-measure are used as performance evaluation criteria. Movielens dataset is used to train and test the system.

In the future, it is planned to develop a recommender system using genetic algorithm and fuzzy clustering algorithms.

## REFERENCES

- [1] T. K. Paradarami, N. D. Bastian, and J. L. Wightman, "A hybrid recommender system using artificial neural networks," *Expert Systems with Applications*, vol. 83, pp. 300-313, 2017.
- [2] C. Speier, J. S. Valacich, and I. Vessey, "The influence of task interruption on individual decision making: An information overload perspective," *Decision Sciences*, vol. 30, no. 2, pp. 337-360, 1999, doi: 10.1111/j.1540-5915.1999.tb01613.x.
- [3] P. Resnick, and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56-58, 1997.
- [4] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., *Recommender Systems Handbook*, Boston, MA: Springer US, 2011, pp. 1-35.
- [5] G. Adomavicius, and A. Tuzhilin, "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*, vol. 17, no. 6, June 2005, doi: 10.1109/TKDE.2005.99.
- [6] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, November 2002.
- [7] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, 2010, doi: 10.1017/CBO9780511763113.
- [8] R. Yera, and L. Martinez, "Fuzzy tools in recommender systems: A survey," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 776-803, 2017.
- [9] S. J. Gong, "The collaborative filtering recommendation based on similar-priority and fuzzy clustering," *2008 Workshop on Power Electronics and Intelligent Transportation System*, Guangzhou, China, Aug. 2-3, 2008, doi: 10.1109/PEITS.2008.99.
- [10] A. Gunawardana, and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *Journal of Machine Learning Research*, vol. 10, pp. 2935-2962, 2009.
- [11] X. Su, and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, article ID 421425, 19 pages, doi: 10.1155/2009/421425.
- [12] S. Panigrahi, R. K. Lenka, and A. Stitipragyan, "A hybrid distributed collaborative filtering recommender engine using apache spark," *International Workshop on Big Data and Data Mining Challenges on IoT and Pervasive Systems (BigD2M)*, 2016.
- [13] G. Adomavicius, and A. Tuzhilin, "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*, vol. 17, no. 6, pp. 734-749, 2005.
- [14] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the Netflix prize," in *AAIM'08*, pp. 337-348, Berlin, Heidelberg, Springer-Verlag, 2008.
- [15] M. Isard, and Y. Yu, "Distributed data-parallel computing using a high-level programming language," in *SIGMOD*, pp. 987-994, 2009.
- [16] M. A. Ghazanfar, and A. Prugel-Bennett, "Fulfilling the needs of gray-sheep users in recommender systems, a clustering solution," *2011 Int. Conf. on Information Systems and Computational Intelligence*, China, Jan. 18-20, 2011.
- [17] W. Zhou, J. Wen, Q. Qu, J. Zeng, and T. Cheng, "Shilling attack detection for recommender systems based on credibility of group users and rating time series". Available: <https://doi.org/10.1371/journal.pone.0196533>
- [18] M. M. R. Siddiquee, N. Haider, and R. M. Rahman, "A fuzzy based recommendation system with collaborative filtering," *The 8th Int. Conf. on Software, Knowledge, Information Management and Applications (SKIMA 2014)*, Dhaka, Bangladesh, Dec. 18-20, 2014, doi: 10.1109/SKIMA.2014.7083524.
- [19] H. Parvin, P. Moradi, and S. Esmaceli, "A collaborative filtering method based on genetic algorithm and trust statements," *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, Kerman, Iran, Feb. 28-Mar. 2, 2018, doi: 10.1109/CFIS.2018.8336613.
- [20] B. Alhijawi, and Y. Kilani, "Using genetic algorithms for measuring the similarity values between users in collaborative filtering recommender systems," *2016 IEEE/ACIS 15th Int. Conf. on Computer and Information Science (ICIS)*, Okayama, Japan, Jun. 26-29, 2016, doi: 10.1109/ICIS.2016.7550751.
- [21] S. K. Verma, N. Mittal, and B. Agarwal, "Hybrid recommender system based on fuzzy clustering and collaborative filtering," *2013 4th Int. Conf. on Computer and Communication Technology (ICCCT)*, Allahabad, India, Sept. 20-22, 2013, doi: 10.1109/ICCCT.2013.6749613.
- [22] T. Jeon, J. Cho, S. Lee, G. Baek, and S. Kim, "A movie rating prediction system of user propensity analysis based on collaborative filtering and fuzzy system," *2009 IEEE Int. Conf. on Fuzzy Systems*, Jeju Island, South Korea, Aug. 20-24, 2009, doi: 10.1109/FUZZY.2009.5277415.
- [23] H. Shivhare, A. Gupta, and S. Sharma, "Recommender system using fuzzy c-means clustering and genetic algorithm based weighted similarity measure," *2015 IEEE Int. Conf. on Computer, Communication and Control (IC4-2015)*, Indore, India, Sept. 10-12, 2015, doi: 10.1109/IC4.2015.7375707.
- [24] L. Shou-Qiang, Q. Ming, and X. Qing-Zhen, "Research and design of hybrid collaborative filtering algorithm scalability reform based on genetic algorithm optimization," *2016 6th Int. Conf. on Digital Home (ICDH)*, Guangzhou, China, Dec. 2-4, 2016, doi: 10.1109/ICDH.2016.045.
- [25] H.-T. Kim, E. Kim, J.-H. Lee, and C. W. Ahn, "A recommender system based on genetic algorithm for music data," *2010 2nd Int. Conf. on Computer Engineering and Technology*, Chengdu, China, April 16-18, 2010, doi: 10.1109/ICCET.2010.5486161.
- [26] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," *Proc. of the 10th Int. Conf. on World Wide Web*, pp. 285-295, 2001.
- [27] D. Roy, and A. Kundu, "Design of movie recommendation system by means of collaborative filtering," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 4, pp. 67-72, 2013.
- [28] Q. Li, and M. Zhou, "Research and design of an efficient collaborative filtering predication algorithm," *Proc. of the 4th Int. Conf. on Parallel and Distributed Computing, Applications and Technologies*, Chengdu, China, Aug. 29-29, 2003, doi: 10.1109/PDCAT.2003.1236281.
- [29] Y. Ar, and E. Bostanci, "A genetic algorithm solution to the collaborative filtering problem," *Expert Systems with Applications*, vol. 61, pp. 122-128, 2016.
- [30] A. J. Bagnall, G. J. Janacek, and M. Powell, "A likelihood ratio distance measure for the similarity between the fourier transform of time series," *PAKDD'05 Proc. of the 9th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining*, pp. 737-743, 2005, doi:10.1007/11430919\_85.
- [31] <https://mahout.apache.org/docs/0.13.0/api/docs/mahout-mr/org/apache/mahout/cf/taste/impl/similarity/UncenteredCosineSimilarity.html>
- [32] C. Birtolo, D. Ronca, and R. Armenise, "Improving accuracy of recommendation system by means of item-based fuzzy clustering collaborative filtering," *2011 11th Int. Conf. on Intelligent Systems Design and Applications*, Cordoba, Spain, Nov. 22-24, 2011, doi: 10.1109/ISDA.2011.6121638.
- [33] <https://mahout.apache.org/docs/0.13.0/api/docs/mahout-mr/org/apache/mahout/cf/taste/impl/eval/AverageAbsoluteDifferenceRecommenderEvaluator.html> 11.12.2019
- [34] <https://mahout.apache.org/docs/0.13.0/api/docs/mahout-mr/org/apache/mahout/cf/taste/impl/>

- eval/AbstractDifferenceRecommenderEvaluator.html#evaluate-org.apache.mahout.cf.taste.eval.RecommenderBuilder-org.apache.mahout.cf.taste.eval.DataModelBuilder-org.apache.mahout.cf.taste.model.DataModel-double-double 11.12.2019
- [35] <https://mahout.apache.org/docs/0.13.0/api/docs/mahout-mr/org/apache/mahout/cf/taste/impl/eval/AverageAbsoluteDifferenceRecommenderEvaluator.html> 22.07.2019
- [36] S. Jamalzehi, and M. B. Menhaj, "Scalable user similarity estimation based on fuzzy proximity for enhancing accuracy of collaborative filtering recommendation," *2016 4th Int. Conf. on Control, Instrumentation, and Automation (ICCIA)*, Qazvin, Iran, Jan. 27-28, 2016, doi: 10.1109/ICCIAutom.2016.7483164.
- [37] M. A. G. Pinto, R. Tanscheit, and M. Vellasco, "Hybrid recommendation system based on collaborative filtering and fuzzy numbers," *2012 IEEE Int. Conf. on Fuzzy Systems*, Brisbane, QLD, Australia, Jun. 10-15, 2012, doi: 10.1109/FUZZ-IEEE.2012.6251308.
- [38] <https://grouplens.org/datasets/movielens/> 25.08.2018