Anadolu Üniversitesi Bilim ve Teknoloji Dergisi A- Uygulamalı Bilimler ve Mühendislik
Anadolu University Journal of Science and Technology A- Applied Sciences and Engineering

2017 - Volume: 18 Number: 1
Page: 225 - 237
DOI: 10.18038/aubtda.273802
Received: 06 December 2016 Revised:28 February 2017 Accepted: 06 March 2017

IMPROVING ACCURACY OF MULTI-CRITERIA COLLABORATIVE FILTERING BY NORMALIZING USER RATINGS

Alper BİLGE [1], Alper YARGIÇ [1,*]

[1] Department of Computer Engineering, Engineering Faculty, Anadolu University, 26555, Eskişehir, TURKEY

ABSTRACT

Multi-criteria collaborative filtering schemes allow modeling user preferences in a more detailed manner by collecting ratings on various aspects of a product or service. Although preferences are expressed by numerical ratings within a predetermined scale, it is not guaranteed that users comprehend such scale identically. As a result, profiles of users with similar tastes might turn out to be unrelated. Besides, distinct criteria might have different rating scales creating an essential incompatibility with the rating schemes of users which in turn conceals proper relation between main criterion and sub-criteria. Since users rate items based on their personal rating habits, it is essential to determine user similarities according to their rating patterns by normalizing ratings to an identical scale. In this paper, two different normalization methods are studied, i.e., *z*-score normalization and decoupling normalization, in order to improve accuracy of multi-criteria collaborative filtering systems. In particular, two normalization methods are employed by modifying the state-of-the-art memory-based multi-criteria recommender schemes so that similarities among users are calculated based on preference models rather than pure numerical ratings. Real world data-based experimental results show that both methods, especially decoupling normalization method, provide significant improvements on accuracy of estimated multi-criteria predictions and outperform previous pure numerical ratings-based approach.

Keywords: Multi-criteria collaborative filtering; *Z*-score normalization; Decoupling normalization, Accuracy

1. INTRODUCTION

As online services become more prevalent on the Internet, tendencies of people are developed toward realizing daily routines over such services, e.g., following daily news, shopping, and booking a hotel [1]. With rapid expansion of such digital data sources, information filtering becomes vital to discover useful information and avoid redundant data. Collaborative filtering (CF) is a well-known recommender technique which avoids items of disinterest and puts interesting ones forward based on preference patterns of users [2]. In daily life, people intrinsically tend to rely on impressions from others having the experience of a product. CF systems imitate and automate such humane word-of-mouth approach [3]. There are various online service providers, e.g., Amazon, Spotify, TripAdvisor etc. employing CF methods in order to please their customers and increase sales by discovering relevant products based on preference histories.

Traditional CF systems *collect* user preferences in terms of ratings based on their experiences where such ratings are stored in a 2-dimensional user-item matrix [4, 5]. Based on collected repository, the service providers estimate the likelihood of a user's inclination towards a particular product using a two-step process: (*i*) locating neighbors of the user in question as the most similar ones in the database and (*ii*) estimating a prediction based on ratings of designated neighbors. A relatively new approach called Multi-Criteria CF (MCCF), however; collects user preferences not only as a general liking degree, but over multiple sub-criteria with the purpose of better personalization [2, 5-7]. A hotel recommender system, for instance, might collect ratings on cleanliness, location, and staff hospitality sub-criteria in

*Corresponding Author: ayargic@anadolu.edu.tr

addition to an overall liking criterion. The popular restaurant guide Zagat.com [8] allows users to give ratings based on food, service, décor, and cost criteria which leads to a more fine-grained personalization of user's tendencies. MCCF systems leverage such extra preference data to discover hidden relations among users and produce more personalized referrals.

One major problem of MCCF systems is that the whole process is based on direct similarity of rating histories among users. However, different users interpret predetermined rating scales diversely and users with similar tastes might assign different ratings for the same item. Although they have similar tastes, such similarity would not be detectable by direct computations on numeric preference data. While an easy-going user assigns relatively higher ratings, a strict user might tend to rate lower even though they have similar experiences of item. Besides, such incompatibility occurs more often and in a deeper manner with larger rating scales. In addition, the main criterion and sub-criteria in MCCF systems might have different rating scales which complicates the problem even further since users' understanding of the rating scales among criteria might also diverge. Resnick et al. [9] addresses such problem in traditional single-criterion CF systems by normalizing user ratings to a predefined Gaussian distribution. In another study, Jin et al. [10] discuss modeling ratings and preference patterns in a decoupled manner and Jin and Si [11] further compare these two approaches within CF domain. Based on conclusions of these studies, effects of such phenomenon tend to be more visible in MCCF due to inherent relation among ratings for sub-criteria and the main criterion.

Although there is positive evidence that rating domain normalization helps discovering hidden similarities within traditional single-criterion CF systems, it is not investigated how such normalization methods perform on intrinsic/domain-specific data handling problems of MCCF systems. In this paper, we investigate two normalization methods, namely *z*-score and decoupling normalizations, in order to improve accuracy by regularizing multi-criteria user ratings and relieve effects of incompatible rating domains for distinct users and criteria. We study how to apply normalization methods on multi-criteria preference data and compare them with respect to their significance in improving accuracy by producing predictions based on normalized preferences. We perform real data-based experiments to demonstrate effects of normalization approaches against traditional data handling methods within neighborhood-based prediction generation schemes. According to the experimental results obtained, both normalization methods improve multi-criteria ratings-based referrals statistically significantly where decoupling normalization approach achieves greater enhancements compared to *z*-score normalization.

2. RELATED WORK AND PROBLEM DEFINITION

The main purpose of recommender systems is to aid individual users in discovering relevant information among a huge collection which helps coping with the information overload problem. In order to perform such functionality, CF systems utilize information filtering techniques based on previous evaluations of users with similar tastes relying on the assumption that people who agreed in the past are tend to agree in the future, as well [9]. A relatively new approach for CF-based recommender systems is to collect additional ratings on various sub-criteria of the product/service and extend context of provided recommendations. There are also some successful service providers, such as TripAdvisor and Yahoo!Movies, collecting multi-criteria preferences [8].

Recommender systems' performance is directly related to both quantity and quality of collected user preferences [4]. However, they often face with insufficient amount of noisy preference data. Therefore, estimating accurate and dependable predictions based on such unqualified collections becomes a significant challenge, especially for commercial recommender systems [4]. In practice, MCCF systems operate on a huge user-item matrix which consists of preferences of users on varying products and services [6]. However, an average user typically rates a very small fraction of all available items which renders an extremely sparse user-item matrix. Average sparsity level for well-known MCCF databases are higher than 98% [7,12]. Such phenomenon obstructs discovering correlations among users since

similarity calculations are solely performed on commonly rated items between users. Besides the sparsity of collections, variety of users' rating patterns complicates determining like-minded profiles. Such rating patterns are time-varying and affected by personal interpretation of voting ranges. MCCF systems need to deal with consequences of inconsistent voting habits, too [10].

Prediction estimation is performed not only according to individual user ratings but also users' preferential habits [11]. In order to create accurate predictions, it is significant to reveal such preferential patterns by normalizing user ratings into a predefined range. The idea of normalization in the sense of utilizing possessed data effectively is a well-known method in CF domain [9, 11]. In order to discover hidden relations among customers independent of their understanding of rating scales, collected data are transformed using a normalization method. Effects of users' rating habits are aimed to be eliminated by bringing user ratings into an order. Table 1 demonstrates an example of varying rating habits and their effects on discovering true relations among users. In the example, u_1 is tend to give more accepting ratings compared to u_2 and u_3. While generating a prediction for u_1 on i_6, u_2 will be determined as the nearest neighbor based on raw preference data and resulting prediction will agree with u_2's rating on i_6, which is 2. However, if the rating profile is evaluated as a whole, it can be observed that u_1 is a tolerant user submitting only higher ratings and u_3 has a perfectly fitting preference trend to u_1 except that she is not easily satisfied. By matching u_1 and u_3's patterns, the resulting prediction for i_6 would be 4. As can be observed from the example, using raw preference data may result inconsistent predictions. Thus, we examine effects of two well-known normalization methods on accuracy improvements by normalizing user ratings in MCCF systems.

Table 1. An example user-item matrix

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	4	4	5	5	4	?
u_2	4	4	5	4	3	2
u_3	1	1	3	3	1	1

One possible normalization method for preference datasets is to subtract mean of the profile from each user rating; however, it is not an appropriate solution for all kinds of users, such as the ones who only utilize the highest and the lowest ratings possible [13-16]. Other than such straightforward approach, Sarwar et al. [17] propose to employ dimension reduction techniques and Traupman and Wilensky [18] utilize factor analysis methods for rating normalization.

GroupLens [9] propose another approach of normalization by computing average deviation from the mean of a user's individual past ratings, based on other user's evaluations. This method is utilized by [12, 15, 19, 20] to obtain normalized data across all other user's mean ratings. Apart from these, simple mean and weighted mean normalization techniques were proposed for various neighborhood-based methods with the aim of analyzing design choices in multi-attribute utility collaborative filtering systems [21,22]. In another approach, *z*-score normalization method is applied for adjusting each previous vote in the system to the rating distribution of the active user who requests a prediction [12, 13, 19]. Experimental results demonstrate that *z*-score normalized data yields relatively better prediction accuracy than deviation from mean approach [19, 12].

In some of the previous studies on CF systems, varying user preferential patterns are normalized into the same scale based on the assumption that they roughly resemble Gaussian distributions [9, 11]. Such normalization method is applied on both memory- and model-based algorithms successfully [9]. Jin et al. [10] further replace Gaussian normalization by a decoupling method which extracts information about a distribution for the preference values instead of the Gaussian assumption. In a subsequent study, Jin and Si [11] compare Gaussian and decoupling normalization approaches within CF domain and their experimental results demonstrate that decoupling normalization is more effective than Gaussian

normalization method in terms of prediction accuracy. As followed by the literature review, *z*-score and decoupling normalization methods help achieving better prediction accuracy compared to other normalization methods. In this study, we examine effects of two well-known normalization methods, namely *z*-score and decoupling normalization methods, on improving accuracy by normalizing multi-criteria user ratings.

3. AN IMPROVED NORMALIZATION-BASED MCCF FRAMEWORK

One major problem of memory-based CF systems is to understand user tendencies about how to rate an item and match varying types of users with different voting habits for the purpose of collaboration. Normalization methods are known to perform practical and helpful for classification algorithms such as nearest-neighbor classification and clustering [23] and are effective tools for eluding system dependency on varying voting habits and scales. Normalization of raw and entangled user ratings enables CF systems to recognize hidden matchings among rating patterns.

In this study, we analyzed effects of two different normalization techniques, namely *z*-score and decoupling normalization, in order to eliminate varying voting tendencies, discovering hidden user profile similarities and hence improve accuracy of memory-based MCCF systems. We provide a normalization-based preprocessing framework for providing high-quality multi-criteria ratings-based referrals.

3.1. *z*-Score Normalization

Although a clearly distinctive preference scale and their meanings are supplied to system users, some individuals averse to submit strongly high and/or low ratings on an item [8]. In statistics, *z*-score is the signed number of deviations from mean indicating that a datum is above the mean if positive and below the mean otherwise. It is mostly suitable for situations where minimum and maximum of item ratings are unknown. In user-based CF techniques, the original rating of user u for item i, r_{ui}, is *z*-score normalized to z_{ui} as given in Eq. 1 [19].

$$z_{ui} = \frac{r_{ui} - \overline{r_u}}{\sigma_u} \tag{1}$$

where $\overline{r_u}$ and σ_u denote mean and standard deviation of u, respectively. Considering multi-criteria ratings domain, *z*-scores of user ratings in a database can be calculated as explained in Procedure 1.

Procedure 1. *z*-Score Normalization Procedure.
Require: Item List ($I_{1\times m}$), Criteria list ($C_{1\times k}$), User×Item×Criteria matrix ($U_{n\times m\times k}$)
Estimate mean scores and standard deviations ($\rightarrow \overline{U}$, σ_U)
1 : **for all** users **in** *U* (*i* ← 1 to *n*) **do**
2 : **for all** criteria **in** *C* (*c* ← 1 to *k*) **do**
3 : $\overline{u_{i,c}} \leftarrow mean(U(i, :, c))$;
4 : $\sigma_{u,c} \leftarrow std(U(i, :, c))$;
5 : **end for**
6 : **end for**
Estimate mean scores and standard deviations ($\rightarrow Z$)
7 : **for all** users **in** *U* (*i* ← 1 to *n*) **do**
8 : **for all** items **in** *I* (*j* ← 1 to *m*) **do**
9 : **for all** criteria **in** *C* (*c* ← 1 to *k*) **do**
10: $Z_{i,j,c} \leftarrow (U_{i,j,c} - \overline{u_{i,c}}) / \sigma_{u,c}$
11: **end for**
12: **end for**
13: **return** *Z*

In order to exemplify how two apparently dissimilar users' rating patterns are strongly correlated, we provide multi-criteria rating profiles of three users, i.e., u_1, u_2, and u_3, on an allowed voting range of 1-13 in Table 2 and cross distance-based similarities in Table 3.

Table 2. An example user-item matrix with raw and *z*-score normalized ratings

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	7	7	8	8	9	9	9	11	12	12
z_1	-1.1741	-1.1741	-0.6404	-0.6404	-0.1067	-0.1067	-0.1067	0.9606	1.4943	1.4943
u_2	1	1	2	2	3	4	3	5	6	6
z_2	-1.2179	-1.2179	-0.6884	-0.6884	-0.1589	0.3707	-0.1589	0.9002	1.4297	1.4297
u_3	7	7	9	8	10	10	10	12	12	13
z_3	-1.3348	-1.3348	-0.3814	-0.8581	0.0953	0.0953	0.0953	1.0488	1.0488	1.5255

As can be followed in Table 2, within such large distribution range for votes, u_1 and u_3 prefers to rate all items with a more tolerant manner while u_2 prefers to rate harsh ratings. Although 6 is an exceptionally high rating for u_2 and reflects more appreciation, 7 is the lowest given rating for u_1 and u_3 which indicates disapproval. Although preferences based on raw evaluations seem like they are not similar, *z*-score transformed forms of preferences display high correlation which helps discovering similar rating patterns. Calculated similarity values based on raw and *z*-score transformation based data are given in Table 3.

Table 3. Similarity values based on raw and *z*-score normalized ratings

$sim(u_1, u_2)$	$sim(z_1, z_2)$	$sim(u_1, u_3)$	$sim(z_1, z_3)$
0.1449	0.9128	0.6250	0.8355

Although inclinations of u_1 and u_2 are numerically far from each other, they show very similar behavioral patterns on their preferences. However, as can be followed in Table 3, such analogous patterns render irrelevant based on the raw ratings and considered close neighbors based on their *z*-score transformed profile similarities. On the other hand, u_1 and u_3 seem similar due to their tolerance; however, their patterns not as analogous as with u_2. As a result, *z*-score normalizing raw user ratings helps discovering hidden close relationship between u_1 and u_2, even more than with u_3.

3.2. Decoupling Normalization

Decoupling normalization is a probabilistic data normalization technique used in recommender systems context. Rather than labeling liking level of a user for a product or service based on a sole vote, it assumes a probabilistic measurement for the rated item to be favored by the user. Such likelihood of favoring is obtained based on the following two principles:

1. As the higher the percentage a user rates items as less than or equal to rating category *R*, the more the likelihood that those items rated as *R* to be favored by the user.
2. As the higher the percentage a user rates items as exactly equal to rating category *R*, the less the likelihood that those items rated as *R* to be favored by the user.

Jin et al. [10] propose combining these two principles in order to approximate the likelihood of a user preferring a particular rating category *R* by utilizing a halfway accumulative distribution approach as explained in Eq. 2:

$$\Pr(R \text{ is preferred}) = \Pr(U \leq R) - \Pr(U = R) / 2 \quad (2)$$

where *R* is the particular rating category, *U* is the profile vector, $\Pr(U \leq R)$ and $\Pr(U = R)$ terms represent the percentage of items that are rated at most with category *R* and exactly rated with category *R*,

respectively. Considering multi-criteria ratings domain, decoupling normalization of user ratings in a database can be calculated as explained in Procedure 2.

Procedure 2. Decoupling Normalization Procedure.
Require: User×Item×Criteria matrix ($U_{n\times m\times k}$)
1 : **for all** users **in** U ($i \leftarrow 1$ to n) **do**
Estimate probabilities of users' ratings being exactly or less than or equal to rating category R
2 : **for all** criteria **in** C ($c \leftarrow 1$ to k) **do**
3 : **for each** rating category r in R
4 : $\Pr(U_{i,c} = r) \leftarrow |U_{i,:,c} = r| \,/\, |U_{i,:,c} \neq \emptyset|$
5 : $\Pr(U_{i,c} \leq r) \leftarrow |U_{i,:,c} \leq r| \,/\, |U_{i,:,c} \neq \emptyset|$
6 : **end for**
Estimate probabilities of rating category R is preferred
7 : **for each** v **in** $U_{i,:,c} \neq \emptyset$
8 : **switch** $U_{i,:,c}$
9 : **case** $\forall r$ **in** R
10: $D_{i,v,c} \leftarrow \Pr(U_{i,c} \leq r) - \Pr(U_{i,c} = r) / 2$
11: **end switch**
12: **end for**
13: **end for**
14: **end for**

In order to exemplify how decoupling normalization technique handles the problem of modeling a user' preference patterns on items independently from user's pure rating scheme, we provide the same example of multi-criteria rating profiles in Table 2 with decoupled ratings in Table 4.

Table 4. An example user-item matrix with raw and decoupled normalized ratings

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	7	7	8	8	9	9	9	11	12	12
d_1	0.1	0.1	0.3	0.3	0.55	0.55	0.55	0.75	0.9	0.9
u_2	1	1	2	2	3	4	3	5	6	6
d_2	0.1	0.1	0.3	0.3	0.5	0.65	0.5	0.75	0.9	0.9
u_3	7	7	9	8	10	10	10	12	12	13
d_3	0.1	0.1	0.35	0.25	0.55	0.55	0.55	0.8	0.8	0.95

As can be followed in Table 4, decoupled rating patterns demonstrate resemblance between d_1 and d_2 while they do not with u_1 and u_2. Estimated cross distance-based similarities of u_1 to u_2 and u_3 along with d_1 to d_2 and d_3 are given in Table 5.

Table 5. Similarity values based on raw and decoupled normalized ratings

$sim(u_1, u_2)$	$sim(d_1, d_2)$	$sim(u_1, u_3)$	$sim(d_1, d_3)$
0.1449	0.9804	0.6250	0.9709

As can be followed by Table 5, similar to the case with z-score normalized ratings, u_1 and u_2 would be considered as close neighbors based on their decoupled normalized ratings-based similarity, i.e., $\text{sim}(d_1,d_2)$. Similarly, decoupling normalization helps discovering a precious hidden correlation between u_1 and u_2 which was unclear with raw ratings-based profiles.

3.3. Similarity Estimation

In a single-criterion neighborhood-based CF system, similarities between users can be calculated using Pearson's correlation coefficient and a close neighborhood is formed accordingly. For that purpose, similarities between the active user (a) for whom the prediction will be estimated and a system user (u) is calculated as in Eq. 3.

$$sim(a,u) = \frac{\left(\sum_{i \in I(a,u)} R(a,i) R(u,i)\right)}{\left(\sqrt{\sum_{i \in I(a,u)} R(a,i)^2} \sqrt{\sum_{i \in I(a,u)} R(u,i)^2}\right)} \tag{3}$$

where I corresponds to the set of commonly rated items between a and u, $R(a, i)$ and $R(u, i)$ corresponds to the ratings of a and u to item i in the list I, respectively. Similarities based on raw and decoupled ratings-based profiles can be calculated using Eq. 3.

In order to estimate similarities between z-score normalized profiles, it is assumed that the list of commonly rated items corresponds to the actual list of rated items for each user. If the terms in the denominator of Eq. 3 are extend to cover the list of all rated items by a and u, respectively, the estimated similarity between these users can be calculated using Eq. 4.

$$sim(z_1, z_2) = z_1 \cdot z_2 \tag{4}$$

On the other hand, MCCF systems hold detailed user ratings over varying criteria along with an overall rating most of the time. Consequently, global similarity value between two users relies on the similarities obtained from sub-criteria and overall criterion ratings. For this purpose, such multiple similarity values are aggregated using two different approaches. In order to estimate global similarity value between a and u, obtained individual similarity values are aggregated in such a way that global aggregation is either the average of individual values or the worst case of them [6]. Average similarity assumes that the global similarity values is dependent equally on all the individual similarity values and is calculated as given in Eq. 5.

$$sim_{avg}(a,u) = \frac{1}{k+1} \sum_{i=0}^{k} sim_i(a,u) \tag{5}$$

where k corresponds to the number of sub-criteria. Worst-case similarity, on the other hand, assumes that the global similarity is bounded to the minimum of individual similarity values as formulated in Eq. 6.

$$sim_{min}(a,u) = \min_{i=0,\ldots,k} sim_i(a,u) \tag{6}$$

3.4. Prediction Estimation

When a requests a prediction for a target item (q), a two-step processes is followed in order to estimate the prediction: (i) construct neighborhood of a based on the similarities between a and all the other users in the system and (ii) aggregate ratings of users in the neighborhood for q in a weighted manner. Such prediction can be estimated for raw ratings-based profiles as given in Eq. 7.

$$P_{a,q} = \frac{\sum_{u \in U} \left(r_{u,q} - \bar{r}_u\right) sim(a,u)}{\sum_{u \in U} sim(a,u)} \tag{7}$$

where U corresponds to the set of neighbors and $sim(a, u)$ corresponds to the similarity value obtained either using Eq. 5 or Eq. 6. However, since z-score normalized ratings are transformed into another domain, prediction obtained using such normalized ratings needs to be de-normalized as given in Eq. 8.

$$P_{a,q} = \bar{r}_a + \sigma_a \frac{\sum_{u \in U} z_{u,q} sim(a,u)}{\sum_{u \in U} sim(a,u)} \quad (8)$$

where $\bar{r}_a$ and σ_a corresponds to the mean and standard deviation of a, respectively. Finally, a prediction for decoupling normalization-based profiles can be estimated using Eq. 7. In addition, such obtained prediction value needs to be de-normalized to the original scale since halfway accumulative distribution process converts raw ratings into likelihood of preference. Therefore, expected preference probability value is de-normalized matching it to the corresponding rating category [10].

4. EXPERIMENTS

In this study, we examined effects of *z*-score and decoupling normalization methods on prediction accuracy for MCCF systems. For this purpose, we employed average and worst-case similarity calculation approaches on well-known memory-based MCCF recommendation schemes. We provide details of experimental outcomes and significance of obtained results in this section.

4.1. Datasets and Evaluation Criteria

We utilized varying versions of a multi-criteria ratings-based dataset crawled from Yahoo!Movies[1] (YM) which is one of the most commonly used dataset for MCCF recommender systems [6,7]. YM dataset consists of four individual specific criteria ratings for movies, namely *Story*, *Acting*, *Directing*, and *Visuals*, along with an *Overall* liking degree rating. All ratings belonging those five criteria are rated by users based on a letter grade rating systems consisting of 13 characters, i.e., [*A*+, *A*, *A*–, *B*+, *B*, *B*–, *C*+, *C*, *C*–, *D*+, *D*, *D*–, *F*], where *F* denotes the worst measure and *A*+ the best. For experimental purposes, we employed numerically converted ratings where 1 represents *F* and 13 represents *A*+. Originally, YM is an extremely sparse dataset where only 0.02% of available items are rated. Jannach et al. [7] constructed three subsets of the original dataset by extracting users and items having at least 5, 10, and 20 ratings for experimental purposes, namely, YM_5_5, YM_10_10, and YM_20_20. Details of the datasets are given in Table 6.

Table 6. Sparsity level of Yahoo!Movies dataset's subsets.

	YM_5_5	***YM_10_10***	***YM_20_20***
Number of Users	4377	1293	202
Number of Items	2565	1164	247
Number of Records	63027	34846	8157
Sparsity Rate	%99.4386	%97.6847	%83.6513

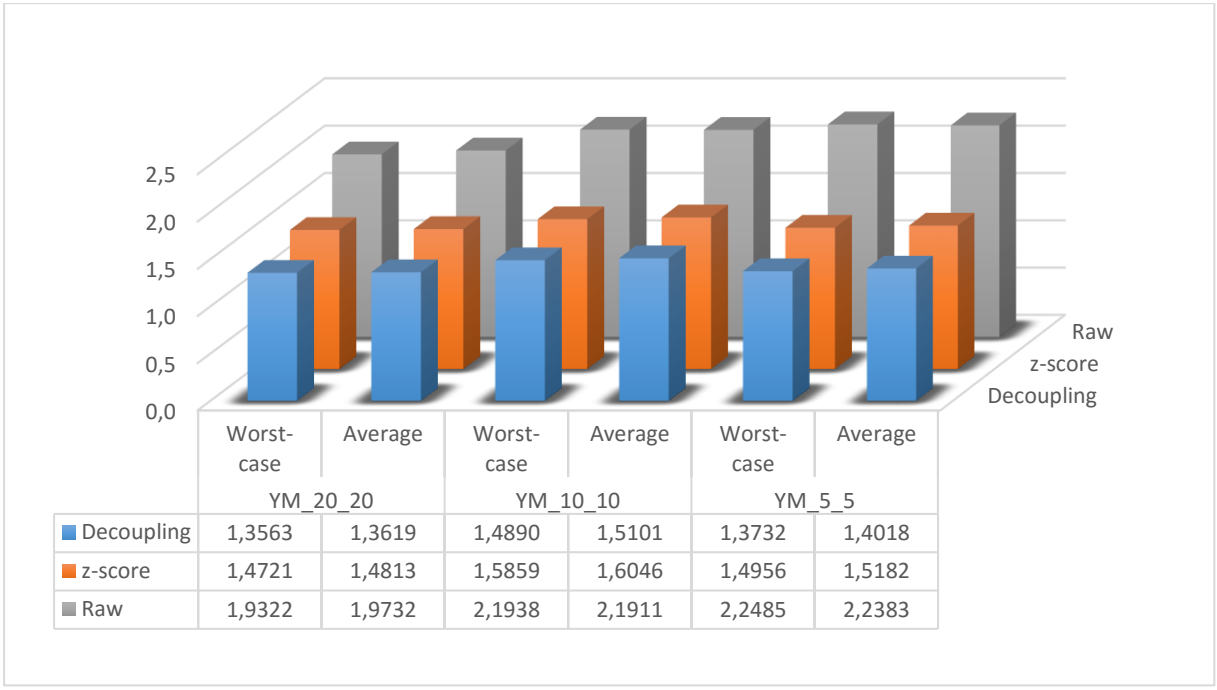
4.2. Experimentation Methodology

During experimental evaluations, all available preference data is subjected to recommendation process using exhaustive leave-one-out cross validation method. According to this method, each user in the dataset is treated as active user as one user at a time and remaining users constructed the train data [7, 9]. A prediction is generated for each submitted rating of individual users by withholding one vote at a time and trying to predict its value by employing remaining votes of the corresponding user via average and worst-case similarity calculation approaches. Therefore, total number of predictions generated for each dataset is equal to the number of total ratings in the corresponding database.

[1] https://www.yahoo.com/movies/

4.3. Experimental Results and Discussion

In order to examine effects of *z*-score and decoupling normalization methods on prediction accuracy of MCCF systems, we performed two different normalization approaches before estimating predictions over three different subsets of YM dataset. For experimental purposes, we set number of neighbors to a constant of 10 for all kinds of experiments which means the most similar 10 users are utilized in the prediction estimation process based on minimum and average similarity values calculated over all criteria. We present experimental outcomes of error levels over varying configurations in Figure 1.



	YM_20_20		YM_10_10		YM_5_5	
	Worst-case	Average	Worst-case	Average	Worst-case	Average
Decoupling	1,3563	1,3619	1,4890	1,5101	1,3732	1,4018
z-score	1,4721	1,4813	1,5859	1,6046	1,4956	1,5182
Raw	1,9322	1,9732	2,1938	2,1911	2,2485	2,2383

Figure 1. MAE results over raw, decoupling normalized, and *z*-score normalized data

Experimental results indicate that both normalization methods have a positive effect on prediction accuracy compared to raw data for all datasets. However, improvements obtained by applying decoupling normalization are even higher than obtained by applying *z*-score normalization. Such slight difference in obtained improvements originate from prediction estimation step which assumes normalized data for *z*-score normalization-based configurations and actual data for decoupling normalization-based configurations. Also, for most of the time worst-case similarity estimation outperforms average similarity estimation except for raw data in big datasets. Therefore, it can be concluded that worst-case similarity estimation helps increasing prediction accuracy when normalization procedures are applied while average similarity estimation is more effective in raw data configurations. One interesting outcome of the experiments is that prediction accuracy is improved in a better way by normalization procedures for larger datasets, i.e., YM_5_5. Although error values increase while the dataset gets larger and sparser for raw data-based experiments, improvement levels get higher when normalization procedures are employed. Improvement levels are about 23.8%, 27.7%, and 33.5% for *z*-score normalization and 29.8%, 32.1%, 38.9% for decoupling normalization for YM_20_20, YM_10_10 and YM5_5, respectively with worst-case similarity estimation. Similarly, 24.9%, 26.8%, and 32.2% for *z*-score normalization and 30.9%, 31.1%, and 32.2% for decoupling normalization for YM_20_20, YM_10_10 and YM5_5, respectively with average similarity estimation. Hence, it can be concluded that effects of normalization get more significant as the datasets get larger which is crucial for recommender systems since they are prone get larger constantly. Such experimental outcomes support employing normalization procedures, especially decoupling normalization, for improving

prediction accuracy and getting MCCF systems more robust against enlarging due to increasing number of users and available products.

Although normalization procedures outperform raw data-based framework and provide better prediction accuracy results, it is necessary to check whether these improvements are statistically significant. In order to determine the significance of obtained experimental improvements using normalized data against raw data-based configurations, *t*-tests procedures are followed. Results of performed statistical significance tests are displayed in Table 7.

Table 7. Comparison of statistical significance tests between raw data and normalized forms

		YM_5_5	***YM_10_10***	***YM_20_20***
Raw vs. decoupling	Average	t = 91.1101 p = 0.000 *	t = 60.5027 p = 0.000 *	t = 27.0644 p = 0.000*
	Worst-case	t = 96.8730 p = 0.000 *	t = 62.1902 p = 0.000 *	t = 27.6756 p = 0.000*
Raw vs. *z*-score	Average	t = 84.1620 p = 0.000 *	t = 52.577 p = 0.000 *	t = 24.3979 p = 0.000*
	Worst-case	t = 89.9094 p = 0.000 *	t = 55.2852 p = 0.000 *	t = 23.6292 p = 0.000 *

** Significance at %99*

In the light of the results shown in Table 7, obtained improvements by decoupling and *z*-score normalization-based data are statistically significant with a pretty high confidence level of 99% ($p<0.01$ indicated with +) for all YM sub-datasets. Such results confirm that obtained improvements are solid and significantly outperform raw-data based configurations. We further compare error level difference distributions between raw configuration and two normalized forms. Variance distributions for the worst-case similarity estimation in YM_20_20 dataset are displayed in Figure 2.

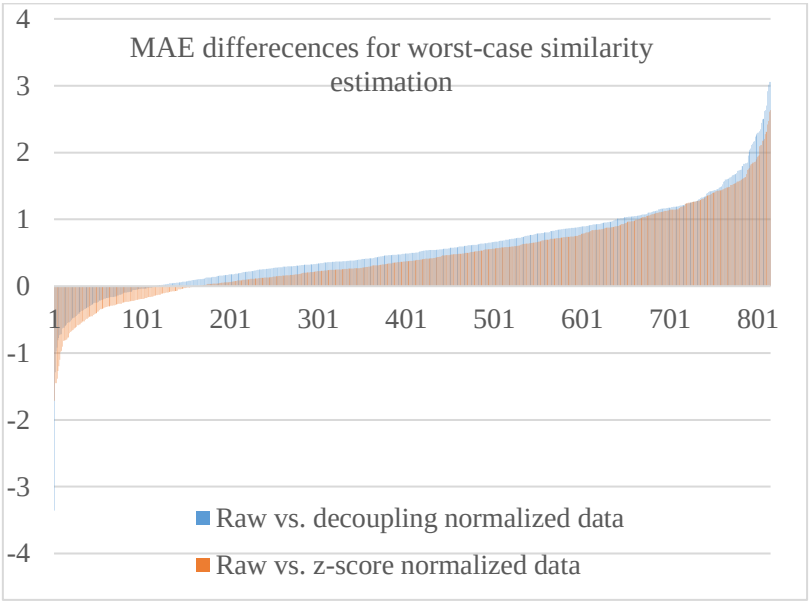


Figure 2: Error level difference distributions between raw data and normalized forms

Finally, in order to evaluate significance of the improvements obtained by normalization methods, distribution of worst-case similarity estimation MAE differences between results of *z*-score and decoupling normalized data are compared. MAE results obtained with the normalized data were subtracted from the MAE results obtained with the raw data separately and then sorted in ascending order. The negative side of Figure 2. denotes the region of the least accurate prediction values and the positive side refers to the most accurate predictions of normalization methods. According to the results seen in Figure 2, it can be concluded that prediction estimations obtained by applying decoupling normalization method are more consistent compared to those obtained by applying *z*-score normalization method.

5. CONCLUSIONS AND FUTURE WORK

Multi-criteria collaborative filtering schemes allow evaluating multiple aspects of a product and/or service in terms of their specific features. Collected data for such services contain either numeric or binary preferences of users expressing users' liking degrees. However, available ranges for varying criteria might be different for each evaluation type. More importantly, each user might interpret such ranges differently which causes an inconsistency among true preferences of users. While a user distributes their preferences uniformly to the range, another user votes items with the highest and the lowest ratings available. Also, similar preference histories might distribute in differing sub-ranges of the criteria. Such inconsistencies complicate the main step of collaborative recommendation process, i.e., detecting similarities in preference histories of users. Normalization procedures help reducing such risk in traditional collaborative filtering schemes. In this study, we discuss applying two well-performing normalization procedures onto multi-criteria preference data in order to improve prediction accuracy. We explain how to apply *z*-score and decoupling normalization techniques onto criteria-based preferences and estimate predictions based on normalized data in order to overcome negative effects of varying voting habits of users. We perform real-world data-based experiments for assessing effectiveness of proposed methods. According to obtained experimental outcomes, both normalization procedures outperform raw data-based configurations. However, due to assumptions in prediction estimation step, decoupling normalization achieves a slightly better improvement compared to *z*-score normalization-based configurations. Also, worst-case similarity estimation outperforms average similarity estimation process for normalized data-based configurations which implies data distortion in normalization process. Moreover, improvements obtained in larger and more inconsistent datasets display a higher trend which emphasizes positive effects of normalization on multi-criteria data. There are also binary ratings-based multi-criteria collaborative filtering systems in the literature for which effects of data normalization procedures must be explored. Also, possible normalization procedures and effects of such normalizations are planned to be explored in implicit data-based configurations where users do not explicitly submit their preferences.

ACKNOWLEDGEMENT

This work was supported in part by the Scientific and Technical Research Council of Turkey (TÜBİTAK) under grant number 215E335 and Anadolu University under grant number 1605F325. The authors are also grateful to Dr. Jannach for his support in providing multi-criteria ratings-based experimental datasets.

REFERENCES

[1] Ricci F, Rokach L, Shapira B. Introduction to recommender systems handbook. Springer US, 2011.

[2] Adomavicius G, 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, 2005; 17.6: pp. 734-749.

[3] Manouselis N, Costopoulou C. Analysis and classification of multi-criteria recommender systems. World Wide Web, 2007; 10.4: pp. 415-441.

[4] Su X, Khoshgoftaar TM. A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009; 2009: 4.

[5] Adomavicius G, Manouselis N, Kwon Y. Multi-criteria recommender systems. In: Recommender systems handbook. Springer US, 2011. pp.769-803.

[6] Adomavicius, G, Kwon Y. New recommendation techniques for multi-criteria rating systems. IEEE Intelligent Systems, 2007; 22.3: pp. 48-55.

[7] Jannach D, Karakaya Z, Gedikli F. Accuracy improvements for multi-criteria recommender systems. In: Proceedings of the 13th ACM Conference on Electronic Commerce. ACM, 2012. pp. 674-689.

[8] Adomavicius G, Kwon Y. Multi-criteria recommender systems. In: Recommender Systems Handbook, Springer US, 2015.

[9] Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM, 1994. pp. 175-186.

[10] Jin R, Si L, Zhai C, Callan J. Collaborative filtering with decoupled models for preferences and ratings. In: Proceedings of the twelfth international conference on Information and knowledge management. ACM, 2003. pp. 309-316.

[11] Jin R, Si L. A study of methods for normalizing user ratings in collaborative filtering. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2004. pp. 568-569.

[12] Nilashi M, Jannach D, bin Ibrahim, O, Ithnin N: Clustering-and regression-based multi-criteria collaborative filtering with incremental updates. Information Sciences, 2015; 293: pp. 235-250.

[13] Akhtarzada A, Calude CS, Hosking J. A multi-criteria metric algorithm for recommender systems. Fundamenta Informaticae, 2011; 110.1-4: pp. 1-11.

[14] Chapphannarungsri K, Maneeroj S. Combining multiple criteria and multi-dimension for movie recommender system. In: Proceedings of the International Multi Conference of Engineers and Computer Scientists. 2009.

[15] Shambour Q, Lu J. Integrating multi-criteria collaborative filtering and trust filtering for personalized recommender systems. In Computational Intelligence in Multi-Criteria Decision-Making (MDCM), 2011 IEEE Symposium on IEEE, 2011. pp. 44-51.

[16] Lakiotaki K, Matsatsinis NF, Tsoukias A. Multi criteria user modeling in recommender systems. IEEE Intelligent Systems, 2011. 26.2: pp. 64-76.

[17] Sarwar B, Karypis G, Konstan J, Riedl J. Application of dimensionality reduction in recommender system-a case study. Minnesota Univ. Minneapolis Dept. of Computer Science, 2000.

[18] Traupman J, Wilensky R. Collaborative quality filtering: Establishing consensus or recovering ground truth?. In: International Workshop on Knowledge Discovery on the Web. Springer Berlin Heidelberg, 2004. pp. 73-86.

[19] Herlocker JL, Konstan JA, Borchers A, Riedl J. An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999. pp. 230-237.

[20] Naak A, Hage H, Aimeur E. A multi-criteria collaborative filtering approach for research paper recommendation in papyres. In: International Conference on E-Technologies. Springer Berlin Heidelberg, 2009. pp. 25-39.

[21] Manouselis N, Kyrgiazos G, Stoitsis G, Stoitsis J. Revisiting the multi-criteria recommender system of a learning portal. In: Proceedings of the 2nd Workshop on Recommender Systems in Technology Enhanced Learning. 2012. pp. 35-48.

[22] Manouselis N, Vuorikari R, Van Assche F. Simulated analysis of MAUT collaborative filtering for learning object recommendation. In: Proceedings of the 1st Workshop on Social Information Retrieval for Technology Enhanced Learning. 2007. pp. 27-35.

[23] Han J, Pei J, Kamber M. Data mining: concepts and techniques. Elsevier, 2011.