Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 3419 – 3437
© TÜBİTAK
doi:10.3906/elk-2004-184

# Binary multicriteria collaborative filtering

**Emre YALÇIN**[1,2,*], **Alper BİLGE**[3]
[1]Department of Computer Engineering, Faculty of Engineering, Sivas Cumhuriyet University, Sivas, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, Eskişehir Technical University, Eskişehir, Turkey
[3]Department of Computer Engineering, Faculty of Engineering, Akdeniz University, Antalya, Turkey

**Received:** 27.04.2020 • **Accepted/Published Online:** 24.09.2020 • **Final Version:** 30.11.2020

**Abstract:** Collaborative filtering is specialized in suggesting appropriate products and services to the users concerning personal characteristics and past preferences without requiring any effort of users. It might be more efficient to collect preferences of users based on multiple subcriteria of products and services. For this purpose, researchers propose multicriteria recommender systems that are convenient for more accurate and useful evaluation of items. In such systems, it might be preferable to collect binary ratings instead of numerical ones due to the large number of subcriteria. However, there is a gap in the literature to satisfy a binary preferences-based multicriteria recommender system. In this study, the applicability of multicriteria recommender systems based on binary ratings is investigated. Firstly, recommendations for users on the overall criterion are produced by employing naïve Bayes classifier. In order to improve the quality of recommendations, user- and item-based similarity models are proposed enabling the formation of more successful neighborhoods. Such models are further improved by integrating a concordance measure between overall preference and subcriteria ratings, which helps to provide more personalized and meaningful similarities among users. Finally, a hybrid model is proposed employing user- and item-based models together and real data-based experimental outcomes demonstrate that the quality of estimated binary referrals is improved statistically significantly.

**Key words:** Multicriteria recommender systems, naïve Bayes classifier, collaborative filtering, binary data

## 1. Introduction

Recommender systems are evolving personalization tools that help individuals and businesses to find relevant information within a massive collection of data by automating the fundamental human behavioral characteristic of "word-of-mouth" [1]. Being a significant approach to recommender systems, collaborative filtering (CF) guides their users to unexperienced amenities based on the correlation of past behavioral patterns with other participants of the system [2]. Principally, CF processes rely on the fundamental assumption that who agreed in the past are tend to agree in the future decisions and operate by collecting personal preference information on products and services. Researchers have long been studying improving CF systems in quantitative terms such as accuracy, coverage, and scalability [3].

A qualitative and relatively recent approach for enhancing personalization measure of CF-based recommendations encourage collecting not only general preference values but also detailed ratings on distinctive dimensions of a product/service to constitute fine-grained feedback [4]. These multicriteria collaborative filtering (MCCF) systems allow users to rate an item on multiple subaspects possibly along with a general preference and operate on such multidimensional data to produce better personalized recommendations. Although two

*Correspondence: eyalcin@cumhuriyet.edu.tr users seem quite similar to each other based on their overall tastes, they might easily differentiate why they like/dislike a particular product/service. Since CF systems rely on the correlation among past preferences, such distinction turns out to be vital in discovering latent conformance among users.

However, collecting multidimensional ratings might be challenging since it requires an extensive and time-consuming evaluation of a product and service by customers. The hotel booking platform HRS, for instance, collects criteria ratings in more than a dozen dimensions in numerical scales of 1-to-10 and an overall comfortableness value of 1-to-3 [5]. On the other hand, compared to numerical values, providing binary preferences representing taste status over multiple criteria on the product might be preferable by users due to convenience. Moreover, such a rating mechanism fits specific criteria better, for example, voting a hotel for its accessibility. Therefore, it becomes vital for recommender systems to determine the likelihood of a user will either like or dislike a particular item rather than estimating the exact level of liking as a final or auxiliary step in the recommendation process.

Naïve Bayesian classifier (NBC) is a simple yet efficient binary classifier that is used in traditional single-criterion CF systems to determine class labels (like or dislike) of items subject to recommendation [6, 7]. NBC requires a small amount of data to operate which makes it suitable for sparse collections of preferences and it is based on the assumption that given the class label, the value of a particular feature is independent of the value of any other feature. In other words, it assumes all features are independent. It is also highly scalable since it requires some parameters linearly dependent on the number of features, i.e. the number of users or items in a recommendation system. NBC can be used as a decision-making tool for MCCF schemes operating on multidimensional preference data.

### 1.1. Contributions and organization

In this study, we concentrate on building a framework to produce binary multicriteria ratings-based referrals by employing NBC as a recommendation algorithm. We elaborate on how to combine multidimensional subpredictions to determine the overall likelihood of a product/service to be liked by a user. To achieve this aim, we propose several approaches building on top of each other. The following summarizes the main contributions of this study.

1. We introduce two similarity aggregation-based binary multicriteria recommendation algorithms utilizing NBC algorithm. These algorithms incorporate subcriteria ratings along with overall preferences for estimating similarities among either users or items, which contributes to locating appropriate neighborhoods and consequently improving recommendation quality.

2. We further propose to utilize concordance as a measure to grasp correlations between subcriteria and overall ratings for each user firmly. Considering such correlations leads to having a better-personalized system and provides obtaining more convenient neighborhoods, which is crucial to producing high-quality referrals.

3. We introduce a hybrid algorithm that shrinks the user-item matrix by using the combination of the previously proposed user- and item-based approaches, which enhances classification skills of the NBC algorithm.

The rest of the study is organized as follows: The following section presents a brief literature summary on CF and MCCF. Section 3 explains NBC algorithm and MCCF techniques. Section 4 introduces the proposed approaches, and the following section demonstrates experimental work and obtained results. Finally, Section 6 concludes the study and presents future research directions.

## 2. Related work

Adomavicius and Kwon [8] divide MCCF approaches into two main categories as aggregation function-based and similarity-based approaches, as explained detailed in Section 3.2. Aggregation function-based approach utilizes an aggregation function, which is formed based on relationships between the overall preferences and subcriteria ratings and helps to produce recommendations on the overall preferences. On the other hand, the aggregation similarity-based approach estimates overall similarities among users/items by aggregating the computed similarity values for each criterion.

In order to learn aggregation functions by extracting relations between overall preferences and subcriteria ratings, several techniques are utilized in MCCF domain such as genetic algorithms [9], support vector machines [10], neural networks [11, 12], fuzzy techniques [13, 14], matrix factorization [15], autoencoders [16], and linear regression [15]. In addition, a considerable amount of research aims to improve prediction quality in the MCCF domain by considering similarities among user ratings provided for each criterion [17–19]. Hu et al. [20] introduce a neighborhood selection technique using preference relations between users in order to determine the overall strength of one user's preference over that of another. Liu et al. [21] propose to divide users into groups according to their important criteria and provide recommendations by utilizing users in the same group. Lakiotaki et al. [22] introduce a technique that clusters users according to derived user weight vectors from a model that is constructed by linear programming. In addition, Hu [23] suggests a neighborhood selection method utilizing a grey relational analysis technique to estimate relationships between either users or items. Finally, Bilge and Kaleli [24] develop an item-based MCCF framework that selects the appropriate neighborhoods by considering similarities among items.

In the literature, there are various traditional CF approaches based on binary data [7, 25, 26]. Miyahara and Pazzani [7] utilize NBC algorithm to produce recommendations in single-criterion rating systems based on binary data. To improve prediction quality, they propose two variants of the NBC algorithm, namely sparse and transformation data model. They also propose a NBC-based CF framework in which similarities among either users or items are computed from negative and positive ratings separately [27]. Finally, Hwang [25] propose the Pearson correlation-based method that employs both separated terms and separated terms with proportions in order to improve predictions with CF on binary market basket data.

We conclude this review by noting that, to the best of our knowledge, no research has been conducted on multicriteria rating using binary data in the field of recommender systems. In this paper, we address this issue particularly investigating how to achieve the NBC-based MCCF schemes on binary data. Also, we focus on developing novel similarity-based algorithms to improve prediction quality in MCCF that relies on binary data.

## 3. Preliminaries

In this section, we briefly introduce essential preliminary information on NBC-based CF algorithm and MCCF framework to inform the reader on the background of this study.

### 3.1. Collaborative filtering with the naïve Bayesian classifier

Despite its simplicity, NBC is one of the most efficient generative classifiers [28] which strives for learning a model that estimates the unknown joint probability $P(X,Y)$ of the inputs $< X_1, X_2, \cdots, X_n >$ and the label $Y$. These classifiers are based on the Bayes rule to calculate $P(Y|X)$, and then pick the most likely label $Y$. Considering the complexity of Bayesian classifiers, NBC strongly assumes independence between features to reduce such computational costs. Given the classification task $P(Y|X)$ where $X =< X_1, X_2, \cdots, X_n >$, NBC makes the assumption that each $X_i \in X$ is conditionally independent of $X_j$ given $Y$ where $j = 1, 2, \cdots, n$ and $j \neq i$, and also conditionally independent of each subset of $X_j$ given $Y$. Therefore, if $X$ has $n$ attributes that satisfy the conditional independence, we obtain Eq. 1 to estimate the probability of $Y$ will take on its $k$th possible value.

$$Y \leftarrow \arg\max_{y_k} \{P(Y = y_k) \prod_i P(X_i|Y = y_k)\} \tag{1}$$

where $P(Y = y_k)$ and $P(X_i|Y = y_k)$ values can be estimated from training data.

Miyahara and Pazzani [7] employ NBC for producing CF-based recommendations where preferences of users are depicted in binary form as like (1) and dislike (0). In such scenario, an active user's ratings for items are considered as class labels of the training examples, other users represent features, and votes represent values of those features in the user-item matrix. In their model, the probability of an item belonging to $class_j$, where $j \in \{like, dislike\}$, given its $n$ feature values can be rewritten as in Eq. 2, where only known features are assumed to be informative for the classification process, and ratings for corated items are considered solely to estimate conditional probabilities.

$$Class = \arg\max_{class_j \in \{like, dislike\}} \{P(class_j) \prod_i P(U_i = class_k|class_j)\} \tag{2}$$

### 3.2. Traditional MCCF framework

With the improvements in recommendation services, MCCF have been widely used for personalized recommendation generation purposes. In order to produce better-personalized systems and collect users' preferences on multiple perspectives, traditional MCCF systems request from users not only an overall rating on items but also subcriteria ratings on several aspects of these items, which enables obtaining a detailed evaluation of the product by the user. Therefore, MCCF systems utilize a multiperspective preference vector $(r_0, \cdots, r_c)$ to predict the rating function $R$ as explained in Eq. 3.

$$R : Users \times Items \rightarrow r_0 \times r_1 \times \cdots \times r_c \tag{3}$$

where $r_i$ $(i = 1, 2, \cdots, c)$ denotes the possible rating values for each individual criterion $i$ on different evaluation formats (e.g., binary, numeric scale etc.) and $r_0$ indicates the possible overall rating values.

In order to incorporate and to leverage multicriteria rating information for CF tasks, Adomavicius and Kwon [8] propose aggregation function-based and similarity aggregation-based approaches. In aggregation function-based approach, it is assumed that the overall preference has a special relationship with each individual subcriterion ratings. Such relationship can be specified by statistical or machine learning techniques, such as linear regression, and denoted as the aggregation function, $f$, and presented in Eq. 4. Then, the overall preference level of the user for a particular item is estimated by aggregating prediction values using $f$ for each subcriterion calculated by any traditional single-vote recommendation algorithm.

$$r_0 = f(r_1, r_2, \cdots, r_c) \tag{4}$$

However, in the similarity aggregation-based approach, the similarity among users or items can be calculated in one of two ways, i.e. using multidimensional distance metrics and aggregating individual criterion similarity values. In the former approach, multicriteria ratings are interpreted as points in a multidimensional space, and the distance among them is calculated by multidimensional distance metrics such as Manhattan, Euclidean, or Chebyshev. Then keeping in mind that similarity and distance are inversely related, these distance values are converted into similarity values. Alternatively, one can use similarity metrics such as Cosine similarity or Pearson's correlation coefficient to estimate the similarity between distinct criteria. Finally, these similarity values are aggregated to obtain an overall similarity value between entities by either averaging them as denoted in Eq. 5 or considering the worst case scenario by taking their minimum as denoted in Eq. 6.

$$\overline{sim(a,u)} = \frac{1}{c+1} \sum_{i=0}^{c} sim_i(a,u) \tag{5}$$

$$\lfloor sim(a,u) \rfloor = \min_{i=0,1,\ldots,c} sim_i(a,u) \tag{6}$$

where $sim_i(a,u)$ denotes the subsimilarity value between users $a$ and $u$ based on the $i^{th}$ criterion. It is also possible to compute subsimilarity values between item vectors.

## 4. NBC-based binary multicriteria collaborative filtering

Modern recommender systems collect user opinions relying on multiple aspects of products or services in many domains, such as hotels, restaurants, music, and movie recommendation scenarios. For that reason, MCCF systems aim to provide maximizing the overall satisfaction of users based on an aggregation of their choices on various features [8]. Moreover, since collecting personal preferences is a challenge both in terms of convenience and sentimentality, these systems might be relieved by enabling the collection of binary ratings.

NBC is an efficient classification algorithm that is used for CF purposes on single-rating recommendation systems [7]. In this section, we describe the proposed NBC-based algorithms for the multicriteria domain to enable these systems to deal with binary ratings and consequently improve recommendation quality. In order to form a baseline and demonstrate the effects of utilizing multicriteria ratings, we first employ the NBC algorithm purely on collected overall ratings as a single-criterion system, which is denoted as Pure-NBC. Then, we develop two similarity aggregation-based binary multicriteria recommendation algorithms employing NBC and improve these algorithms with concordance between the overall preference of an item and its subcriteria ratings. Finally, we introduce the hybrid algorithm producing recommendations by combining these algorithms to enhance the accuracy performance of the system.

### 4.1. Similarity aggregation-based binary MCCF algorithms

CF algorithms mainly depend on two fundamental steps of (*i*) forming an appropriate neighborhood and (*ii*) estimating a prediction based on preferences of neighbors. Therefore, the success of these algorithms strongly dependent on how well the neighborhood is formated. Subcriteria ratings might help to form a better neighborhood to produce more accurate recommendations. We propose two similarity aggregation-based MCCF algorithms that rely on binary data, as explained in the following.

Aggregation of user-based similarities *($AGG_U$)*: This algorithm is a form of the traditional user-based nearest neighbor MCCF technique [8], which determines neighbors of an active user and produces recommendations utilizing their ratings for the overall criterion. The $AGG_U$ first extracts user rating vectors for each subcriterion and the overall criterion and employs them to calculate similarities between the active user and other users based on each criterion. Since the extracted user vectors include binary ratings, similarities can be calculated via binary similarity measures, which are more robust and suitable for binary vectors compared to classic similarity/distance metrics [29]. Therefore, in order to perform the estimation of similarities, $AGG_U$ employs binary similarity metrics described in Section 5.2. Having computed the similarities for each criterion, $AGG_U$ merges such similarities by either averaging them ($\overline{AGG_U}$) or taking their minimum ($\lfloor AGG_U \rfloor$), as presented in Eqs. 5 and 6, respectively. Finally, $AGG_U$ determines a set of users as neighbors of the active user based on the aggregated similarities and generates recommendations by performing the NBC algorithm on the selected neighbors' ratings for the overall criterion.

Aggregation of item-based similarities *($AGG_I$)*: Another commonly used method to produce recommendations in the MCCF domain is selecting the most similar items to a target item and utilizing the active user's ratings to these items, which is referred to as item-based nearest neighbor MCCF technique [24]. Therefore, the $AGG_I$ algorithm initially extracts item rating vectors for each criterion and utilizes them to compute per criterion similarities between the target item and the remaining items. Note that binary similarity metrics are utilized to estimate similarities among item vectors. Then, the $AGG_I$ aggregates the calculated similarities values for each criterion to obtain an overall similarity value between items by either averaging them ($\overline{AGG_I}$) or taking their minimum ($\lfloor AGG_I \rfloor$), as presented in Eqs. 5 and 6, respectively. Finally, it selects a set of items as neighbors of the target item based on the estimated aggregated similarities, then applies the NBC algorithm on overall ratings of such set of neighbor items to generate recommendations.

### 4.2. Decorating $AGG_U$ and $AGG_I$ by concordance effects

In a multicriteria rating based system, the importance of subcriteria on the overall satisfaction of users usually differs. In other words, each subcriterion has a diverse effect on the general impression of users on services/products. For example, in a multicriteria based hotel reservation system, while the cleanliness of a hotel might be the most critical subcriterion for a user, the location might be crucial to another user. Therefore, we propose to extract the importance level of each subcriteria for users by considering correlations between subcriteria ratings and overall preference, which then will be utilized in the similarity computation phase. It leads to having a better personalized system and provides to obtain more convenient neighborhoods, which is critical to producing high-quality referrals.

Concordance measure is a practical similarity estimation methodology especially utilized for not breaking user privacy in traditional CF approaches [30]. More specifically, this technique defines the correlations between users/items by counting the number of concordant pairs of ratings in their rating vectors. Therefore, we propose adopting the concordance measure to discover the significance of the subcriteria on the overall preference for users and employing them to improve the similarity estimation procedure in both $AGG_U$ and $AGG_I$ algorithms. Given a pair of binary rating values, $r_1$ and $r_2$, such pair is considered to be concordant if they are identical, i.e. $r_1 = 1$ and $r_2 = 1$, or $r_1 = 0$ and $r_2 = 0$.

In order to understand how concordance measure works, a small user-item matrix is given in Table 1, containing binary ratings like or dislike of a user on four subcriteria and the overall preference for five distinct movies. By counting the number of concordant pairs between the overall preference and subcriteria, concordance value of each criterion ($C$) are calculated as follows: $C_{story} = 5$, $C_{acting} = 3$, $C_{directing} = 1$, and $C_{visuals} = 0$. Thus, it can be followed that each criterion has a varying effect on the overall impression. Note that, for this particular user to shape her tastes story excel as the most significant criterion, followed respectively by acting and directing. At the same time, visual effects do not impress her at all. As a result, we suggest employing such different tendencies of users to compute better personalized and convenient similarities, which leads to obtaining a more dependable neighborhood formation.

**Table 1**. An example of multicriteria rating matrix.

| | Story | Acting | Directing | Visuals | Overall |
|---|---|---|---|---|---|
| Titanic | 1 | 1 | 0 | 0 | 1 |
| Matrix | 1 | 0 | 0 | 0 | 1 |
| Inception | 0 | 0 | 1 | 1 | 0 |
| Kill Bill | 0 | 1 | 1 | 1 | 0 |
| Star Wars | 1 | 1 | 1 | 0 | 1 |

To improve the neighborhood selection process, we decorate similarity computation procedure with concordance effects. In doing so, we first calculate the number of concordant pairs between overall preferences and each subcriterion ratings for the active user ($C_a$). Since the number of rated items may differ for each user, we further normalize these $C_a$ values, as given in Eq. 7.

$$\hat{C}_{a,k} = \frac{C_{a,k}}{\sum\limits_{i=1}^{c} C_{a,i}}, k = 1, \dots, c \tag{7}$$

where $\hat{C}_{a,k}$ is the normalized concordance value for the active user on criterion $k$ and $c$ is the number of subcriteria.

We then improve the $AGG_U$ and the $AGG_I$ algorithms via decorating them with $\hat{C}_a$, as explained in the following.

Concordance decorated $AGG_U$ ($CAGG_U$): This algorithm weights the similarity values between the active user ($a$) and another user ($u$) for each subcriterion by the active user's normalized concordance values ($\hat{C}_a$). Having these similarity values weighted, the $CAGG_U$ estimates the ultimate similarity value $\omega(a,u)$ between $a$ and $u$ by averaging the weighted similarities, as denoted in Eq. 8.

$$\omega(a,u) = \frac{\sum\limits_{i=1}^{c} [sim_i(a,u) \times \hat{C}_{a,i}] + sim_{ovr}(a,u)}{c+1} \tag{8}$$

where $sim_i(a,u)$ and $sim_{ovr}(a,u)$ are the similarity values between $a$ and $u$ for the $i^{th}$ and overall criterion, respectively.

Concordance decorated $AGG_I$ ($CAGG_I$): Similar to the $CAGG_U$, this algorithm weights the similarity values between the target item ($q$) and another item ($t$) for each subcriterion by the active user's normalized concordance values ($\hat{C}_a$). Having these similarity values weighted, the $CAGG_I$ estimates the ultimate similarity value $\omega(q,t)$ between $q$ and $t$ by averaging the weighted similarities, as denoted in Eq. 9.

$$\omega(q,t) = \frac{\sum_{i=1}^{c}[sim_i(q,t) \times \hat{C}_{a,i}] + sim_{ovr}(q,t)}{c+1} \tag{9}$$

where $sim_i(q,t)$ and $sim_{ovr}(q,t)$ are the similarity values between $q$ and $t$ for the $i^{th}$ and overall criterion, respectively.

### 4.3. The hybrid algorithm

We propose several similarity calculation algorithms based on user- and item-based nearest neighbor MCCF approaches in the previous sections. Rather than the entire data set, these algorithms aim to produce recommendations by performing the NBC algorithm on a subset of the user-item matrix, which is formed based on neighbors of either the active user or the target item. To enhance the performance of the NBC algorithm, we suggest utilizing the proposed user- and item-based algorithms together at the neighborhood selection step, which is referred to as the *Hybrid* algorithm. More specifically, this algorithm combines the $CAGG_U$ and $CAGG_I$ to select a subset containing the ratings of both the most similar users and items at the same time. After the selection of neighborhoods, it produces a new form of the user-item matrix with the size of $N_{user} \times N_{item}$, where $N_{user}$ and $N_{item}$ are the number of selected neighbors for the active user and the target item, respectively. Then, the hybrid algorithm performs the NBC algorithm on the filtered relatively smaller user-item matrix. Such hybridization leads both to improve the classification skills of the NBC algorithm and reduce the computation time. Figure 1 presents an overview of the recommendation generation procedure of all the proposed algorithms as a framework.

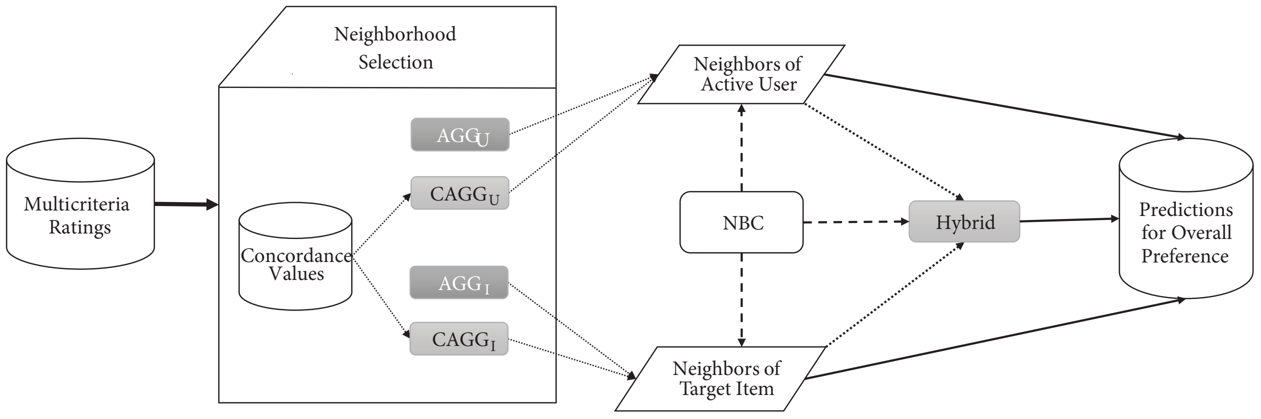


**Figure 1**. An overview of the proposed algorithms.

## 5. Experimental studies

In order to evaluate the performance of the proposed algorithms, we performed several experiments on real-world datasets.

### 5.1. Datasets and evaluation metrics

For experimental purposes, we employ two versions of a famous multicriteria movie rating dataset provided by Yahoo!Movies platform [5]. In the dataset, users have multicriteria preferences for items on four subfeatures of the movie domain, i.e. story, acting, directing, and visuals, along with an overall preference value. Also, the original ratings are in the form of lexical representations (A+ to F) with a 13-level scale. Since the collection is extremely sparse (about 99.98%), we utilize two subsets of the collection in which each user and item has at least 10 and 20 ratings, referred to as YM10 and YM20, respectively. Table 2 gives detailed information about YM10 and YM20.

**Table 2**. Detailed information about YM10 and YM20.

| Dataset | #Users | #Items | #Ratings (per criterion) |
|---|---|---|---|
| YM10 | 1827 | 1471 | 48,026 |
| YM20 | 429 | 491 | 18,504 |

Since the proposed NBC-based algorithms rely on binary data, we perform a preprocessing step in which the lexical evaluations are transformed into binary ratings. More specifically, we first convert the lexical scores into numeric ratings and then transform them into binary preferences as follows: We assign 1 if a rating value of the user for an individual criterion is above the average of his/her ratings for the corresponding criterion and 0 otherwise. Also, this process is performed for each individual subcriterion in addition to overall preferences. For example, assume that [5, 8, 6] are the average ratings of a user for $criterion_1$, $criterion_2$, and $overall$, respectively. Also, assume that [6, 7, 8] are the ratings of the user for the corresponding criteria. Then, such ratings are transformed as [1, 0, 1].

For evaluating the success of the proposed algorithms, we utilize classification accuracy (CA) and F1 measures, which are widely used in CF research to measure the quality of binary predictions [27]. Concretely, CA is defined as the ratio of correct classifications to all classifications. On the other hand, F1-score is the harmonic mean of precision (the ratio of correctly classified positive observations to the total classified positive observations) and recall (the ratio of correctly classified positive observations to all observations in actual class). Thus, the higher CA value and F1-score, the more successful the produced predictions are.

### 5.2. Experimentation methodology

In this study, for the full utilization of datasets, we follow a leave-one-out cross-validation experimentation methodology to evaluate the proposed algorithms. To achieve the cross-validation, we select one active user as the test user and employ the remaining users as the train set. Then, we produce predictions for the actual ratings of the active user by applying the NBC algorithm on the overall ratings of her neighbors, which are estimated by the proposed similarity aggregation-based MCCF algorithms. We repeatedly perform this process for each user in the dataset and compare the generated predictions with the actual ratings by CA and F1-score measures. Note that we perform several experiments with varying neighborhood size ($N$) values ranging from 15 to 300 where we classify the size of selected neighbors as small ($N < 30$), medium ($30 \leq N < 100$), and large ($100 \leq N$).

In the literature, several binary similarity measurements (BSMs) are introduced to calculate correlations between binary vectors [29]. In our experiments, we employ 21 BSMs to calculate similarities among users in the proposed algorithms. However, for the sake of clarity and space constraints, we present the results of the most successful five of them, i.e. Jaccard, Czekanowski, Simpson, Kulczynski, and Johnson. In Table 3, these metrics are formulated based on four pairwise comparison patterns, which refer to the mutual matching status of the vector values. Here, given $i$ and $j$ as binary vectors, $S_{11}$ is the number of pairwise values of $i$ and $j$ are both 1 (positive matches), $S_{01}$ is the number of pairwise values of $i$ is 0 and $j$ is 1 ($i$ absence mismatches), $S_{10}$ is the number of pairwise values of $i$ is 1 and $j$ is 0 ($j$ absence mismatches), and finally $S_{00}$ is the number of pairwise values of both $i$ and $j$ are 0 (negative matches). Note that, in case the pairwise values of $S_{10}$ and $S_{01}$ are zero between $i$ and $j$ vectors, the similarity value cannot be calculated with Kulczynski metric as the denominator value ends up with zero. To deal with this problem, we smooth the calculation process of the Kulczynski metric utilizing a Laplacian prior [27], where we increment both the nominator and denominator values by one before calculating a final similarity value. For example, if rating vectors of $i$ and $j$ are $[1,0,0]$ and $[1,0,0]$, respectively, then the similarity value between $i$ and $j$ is calculated as $1/0$ by the original Kulczynski metric; however, the similarity value will be $2/1 = 2$ by the Kulczynski smoothed by Laplacian method.

**Table 3**. Description of the selected BSMs.

| BSM | Description |
|---|---|
| Jaccard | $\frac{S_{11}}{S_{11}+S_{10}+S_{01}}$ |
| Czekanowski | $\frac{2\times S_{11}}{2\times S_{11}+S_{10}+S_{01}}$ |
| Simpson | $\frac{S_{11}}{min(S_{11}+S_{10},\ S_{11}+S_{01})}$ |
| Kulczynski | $\frac{S_{11}}{S_{10}+S_{01}}$ |
| Johnson | $\frac{S_{11}}{S_{11}+S_{01}} + \frac{S_{11}}{S_{11}+S_{10}}$ |

## 5.3. Benchmark algorithms

We set both user- and item-based $k$-nearest neighbor (kNN) CF methods as the benchmark algorithms to evaluate the efficiency of our proposed algorithms as they are the most prominent approaches in traditional recommender systems [2]. More specifically, for a target item $q$ that is not evaluated by the active user $a$, the user- and item-based kNN algorithms predict an estimate $\hat{r}_{a,q}$ using the formula given in Eqs. 10 and 11, respectively.

$$\hat{r}_{a,q} = \overline{r_a} + \frac{\sum_{u\in U}(r_{u,q} - \overline{r_u}) \times w_{a,u}}{\sum_{u\in U}|w_{a,u}|} \tag{10}$$

where $w_{a,u}$ is the similarity value between the active user $a$ and another user $u$, $r_{u,q}$ is the rating of $u$ for the target item $q$, and $\overline{r_a}$ and $\overline{r_u}$ are the average ratings of users $a$ and $u$, respectively. Also, $U$ denotes the set of neighbors of active user $a$, which is formed by selecting the most similar users from those who rated the target item $q$.

$$\hat{r}_{a,q} = \frac{\sum_{i\in I}(r_{a,i} \times w_{q,i})}{\sum_{i\in I}|w_{q,i}|} \tag{11}$$

where $w_{q,i}$ is the similarity value between the target item $q$ and another item $i$, and $r_{a,i}$ is the rating of user $u$ for the item $i$. Also, $I$ denotes the set of neighbors of target item $q$, which is formed by selecting the most similar items from those rated by the user $u$.

In traditional settings, these algorithms produce a numeric prediction for the active user or target item. However, we focus on providing predictions that rely on binary ratings in this study. Therefore, to compare our findings with the user- and item-based kNN algorithms, the predicted scores of these algorithms are rounded to the nearest binary rating.

## 5.4. Experimental results

### 5.4.1. Effects of $AGG_U$ and $AGG_I$

For evaluating the accuracy performance of the proposed similarity aggregation-based MCCF algorithms, we perform several experiments for varying neighborhood size ($N$) and different BSMs. The CA and F1-score results of the conducted experiments for the $AGG_U$ algorithm including $\overline{AGG_U}$ and $\lfloor AGG_U \rfloor$ methods on YM10 and YM20 datasets are presented in Tables 4 and 5, respectively. Also, we compare empirical outcomes against the pure-NBC algorithm, which denotes the baseline method and applies the NBC algorithm solely on whole overall preferences without filtering any neighborhood, as demonstrated in Section 4.

**Table 4**. CA and F1-score values of the $AGG_U$ and pure-NBC for YM10.

| | | CA | | | | | | F1-score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Pure-NBC | | 70.5 | | | | | | 0.70 | | | | | |
| | N | 15 | 30 | 60 | 100 | 200 | 300 | 15 | 30 | 60 | 100 | 200 | 300 |
| $\overline{AGG_U}$ | Jaccard | 63.1 | 69.2 | 75.1 | 79.0 | 79.8 | 76.7 | 0.54 | 0.65 | 0.74 | 0.79 | 0.83 | 0.82 |
| | Czekanowski | 63.1 | 69.1 | 74.9 | 78.3 | 78.2 | 74.8 | 0.54 | 0.65 | 0.74 | 0.79 | 0.82 | 0.80 |
| | Simpson | 61.8 | 65.9 | 69.1 | 70.4 | 70.6 | 70.6 | 0.62 | 0.70 | 0.75 | 0.77 | 0.78 | 0.78 |
| | Kulczynski | 69.3 | 69.7 | 69.2 | 68.4 | 67.6 | 67.5 | 0.74 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 |
| | Johnson | 63.1 | 69.1 | 74.7 | 77.9 | 77.8 | 74.7 | 0.55 | 0.65 | 0.73 | 0.79 | 0.81 | 0.80 |
| $\lfloor AGG_U \rfloor$ | Jaccard | 60.2 | 65.3 | 69.8 | 71.7 | 70.2 | 67.9 | 0.49 | 0.59 | 0.67 | 0.72 | 0.75 | 0.75 |
| | Czekanowski | 60.2 | 65.3 | 69.8 | 71.7 | 70.2 | 67.9 | 0.49 | 0.59 | 0.67 | 0.72 | 0.75 | 0.75 |
| | Simpson | 59.4 | 63.2 | 66.1 | 67.2 | 67.4 | 67.4 | 0.58 | 0.66 | 0.71 | 0.73 | 0.74 | 0.74 |
| | Kulczynski | 61.4 | 62.2 | 62.1 | 61.9 | 62.1 | 62.3 | 0.64 | 0.67 | 0.69 | 0.69 | 0.70 | 0.70 |
| | Johnson | 60.2 | 65.3 | 69.7 | 71.7 | 70.4 | 68.2 | 0.49 | 0.59 | 0.67 | 0.72 | 0.75 | 0.75 |

The empirical outcomes for both datasets demonstrate that the $AGG_U$ algorithm clearly outperforms the pure-NBC when the Jaccard measure and a sufficient amount of neighborhood is utilized, and $\overline{AGG_U}$ performs better than $\lfloor AGG_U \rfloor$. More specifically, if a medium- or large-size neighborhood is utilized for YM10, both CA and F1-score values are enhanced. The best results are obtained for $N = 200$ where CA is improved by 13.2% (from 70.5 to 79.2) and F1-score by 18.6% (from 0.70 to 0.83). Also, a medium-size neighborhood is useful for YM20, where CA is improved by 10.2% and F1-score by 6.4% for $N = 60$. Moreover, these improvements appear to be statistically significant at 99.9% confidence level according to the performed statistical significance $t$-tests.

The results for both datasets evidently indicate that Jaccard outperforms other BSMs since it is more sensitive to positive matches by excluding negative matches. The reason for this consequence is that positive matches between two vectors are more precious than the negative matches for the success of the similarity

**Table 5**. CA and F1-score values of the $AGG_U$ and pure-NBC for YM20.

| | | CA | | | | | | F1-score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Pure-NBC | | 71.7 | | | | | | 0.78 | | | | | |
| | N | 15 | 30 | 60 | 100 | 200 | 300 | 15 | 30 | 60 | 100 | 200 | 300 |
| $\overline{AGG_U}$ | Jaccard | 69.0 | 76.6 | 79.0 | 75.8 | 72.0 | 71.1 | 0.66 | 0.78 | 0.83 | 0.82 | 0.79 | 0.78 |
| | Czekanowski | 69.2 | 76.7 | 78.1 | 74.6 | 71.7 | 71.1 | 0.66 | 0.78 | 0.83 | 0.81 | 0.79 | 0.78 |
| | Simpson | 63.6 | 68.9 | 73.0 | 74.0 | 72.7 | 71.7 | 0.67 | 0.75 | 0.79 | 0.80 | 0.79 | 0.78 |
| | Kulczynski | 73.3 | 73.7 | 72.3 | 71.2 | 70.6 | 70.7 | 0.78 | 0.80 | 0.80 | 0.79 | 0.78 | 0.78 |
| | Johnson | 68.9 | 76.2 | 78.4 | 75.1 | 72.0 | 71.3 | 0.66 | 0.78 | 0.83 | 0.81 | 0.79 | 0.78 |
| $\lfloor AGG_U \rfloor$ | Jaccard | 67.5 | 73.0 | 72.4 | 70.5 | 69.2 | 69.1 | 0.65 | 0.76 | 0.78 | 0.78 | 0.76 | 0.76 |
| | Czekanowski | 67.5 | 73.0 | 72.4 | 70.5 | 69.2 | 69.1 | 0.65 | 0.76 | 0.78 | 0.78 | 0.76 | 0.76 |
| | Simpson | 62.3 | 67.5 | 71.3 | 72.0 | 70.4 | 69.3 | 0.65 | 0.73 | 0.77 | 0.78 | 0.77 | 0.76 |
| | Kulczynski | 68.9 | 68.6 | 67.7 | 67.7 | 67.4 | 67.5 | 0.74 | 0.75 | 0.75 | 0.75 | 0.75 | 0.74 |
| | Johnson | 67.3 | 73.1 | 73.1 | 71.0 | 69.8 | 69.2 | 0.64 | 0.76 | 0.79 | 0.78 | 0.77 | 0.76 |

estimation procedure. Besides, it can be concluded that the performances of all BSMs significantly decrease and converge to a particular level with the diminishing neighborhood size. Finally, $\overline{AGG_U}$ performs better than $\lfloor AGG_U \rfloor$ since the former considers the information obtained from all criteria, but the latter discards most criteria and contracts the algorithms to a small neighborhood.

Tables 6 and 7 present CA and F1-score results of the $AGG_I$ algorithm for YM10 and YM20, respectively. The most substantial results of conducted experiments for both datasets is that the $AGG_I$ algorithm significantly outperforms the pure-NBC and also slightly better than the $AGG_U$ algorithm for almost all neighborhood size.

Similar to the outcomes of the $AGG_U$ algorithm, it can be concluded that the best results of the $AGG_I$ algorithm are obtained when the neighborhood size is large for YM10 dataset and medium for YM20 dataset. Also, it can be followed from Tables 6 and 7, $AGG_I$ variations are more robust than $AGG_U$ even with small neighborhood size, and $\overline{AGG_I}$ method is more successful than $\lfloor AGG_U \rfloor$ in the aggregation of distinct similarity values. Finally, the outcomes also demonstrate that Jaccard is more successful in comparison to other BSMs, and the improvements obtained by the best configuration are statistically significant at 99.9% confidence level.

### 5.4.2. Effects of $CAGG_U$ and $CAGG_I$

For evaluation of the proposed similarity aggregation-based MCCF algorithms decorated with concordance effects, we conducted several experiments for varying neighborhood size. In these experiments, we utilize Jaccard BSM to estimate similarities among users and employ the averaging technique to aggregate individual similarities since they outperform their rivals, as demonstrated in the previous subsection. We compare the accuracy results of the $CAGG_U$ and $CAGG_I$ algorithms against the $AGG_U$ and $AGG_I$ for both datasets in Tables 8 and 9, respectively.

As can be seen in both Tables 8 and 9, decorating $AGG_U$ and $AGG_I$ algorithms with concordance effects enhance the performance of these algorithms significantly, which emphasizes the positive contribution of utilizing concordance as an upholder in neighborhood formation. Obtained improvement levels with the best configurations in terms of CA for YM10 are 4.6% (user-based) and 6.1% (item-based) and for YM20 6.2%

**Table 6**. CA and F1-score values of $AGG_I$ and pure-NBC for YM10.

| | | CA | | | | | | F1-score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Pure-NBC | | 70.5 | | | | | | 0.70 | | | | | |
| | N | 15 | 30 | 60 | 100 | 200 | 300 | 15 | 30 | 60 | 100 | 200 | 300 |
| $\overline{AGG_I}$ | Jaccard | 71.5 | 74.0 | 76.8 | 78.9 | 79.8 | 77.7 | 0.77 | 0.79 | 0.81 | 0.83 | 0.84 | 0.83 |
| | Czekanowski | 71.5 | 74.0 | 76.8 | 78.7 | 78.8 | 76.3 | 0.77 | 0.79 | 0.81 | 0.83 | 0.83 | 0.82 |
| | Simpson | 66.2 | 66.5 | 67.0 | 67.3 | 68.2 | 68.6 | 0.74 | 0.74 | 0.75 | 0.75 | 0.76 | 0.76 |
| | Kulczynski | 72.2 | 72.7 | 72.8 | 72.2 | 71.3 | 70.6 | 0.78 | 0.79 | 0.79 | 0.79 | 0.79 | 0.78 |
| | Johnson | 71.5 | 74.0 | 76.6 | 78.4 | 78.0 | 75.4 | 0.77 | 0.79 | 0.81 | 0.83 | 0.83 | 0.81 |
| $\lfloor AGG_I \rfloor$ | Jaccard | 71.4 | 73.7 | 75.7 | 76.5 | 74.7 | 73.0 | 0.77 | 0.79 | 0.81 | 0.82 | 0.81 | 0.80 |
| | Czekanowski | 71.4 | 73.7 | 75.7 | 76.5 | 74.7 | 73.0 | 0.77 | 0.79 | 0.81 | 0.82 | 0.81 | 0.80 |
| | Simpson | 66.2 | 66.5 | 67.1 | 67.3 | 68.2 | 68.9 | 0.74 | 0.74 | 0.75 | 0.75 | 0.76 | 0.76 |
| | Kulczynski | 70.7 | 70.5 | 69.1 | 68.1 | 66.7 | 66.2 | 0.77 | 0.77 | 0.77 | 0.76 | 0.75 | 0.75 |
| | Johnson | 71.3 | 73.7 | 75.6 | 76.1 | 74.3 | 72.4 | 0.77 | 0.79 | 0.81 | 0.81 | 0.81 | 0.80 |

**Table 7**. CA and F1-score values of $AGG_I$ and pure-NBC for YM20.

| | | CA | | | | | | F1-score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Pure-NBC | | 71.7 | | | | | | 0.78 | | | | | |
| | N | 15 | 30 | 60 | 100 | 200 | 300 | 15 | 30 | 60 | 100 | 200 | 300 |
| $\overline{AGG_I}$ | Jaccard | 77.7 | 80.4 | 80.9 | 79.3 | 75.4 | 73.1 | 0.81 | 0.84 | 0.85 | 0.84 | 0.81 | 0.79 |
| | Czekanowski | 77.7 | 80.2 | 80.1 | 78.6 | 74.6 | 73.0 | 0.82 | 0.84 | 0.84 | 0.83 | 0.81 | 0.79 |
| | Simpson | 63.5 | 64.1 | 65.9 | 66.9 | 71.2 | 72.5 | 0.71 | 0.72 | 0.73 | 0.75 | 0.78 | 0.79 |
| | Kulczynski | 75.8 | 76.6 | 76.5 | 75.7 | 73.4 | 72.3 | 0.81 | 0.82 | 0.82 | 0.81 | 0.80 | 0.79 |
| | Johnson | 77.4 | 80.0 | 79.6 | 77.2 | 73.8 | 72.7 | 0.81 | 0.84 | 0.84 | 0.82 | 0.80 | 0.79 |
| $\lfloor AGG_I \rfloor$ | Jaccard | 77.8 | 78.7 | 77.4 | 75.4 | 73.1 | 72.1 | 0.82 | 0.83 | 0.82 | 0.81 | 0.80 | 0.79 |
| | Czekanowski | 77.8 | 78.7 | 77.4 | 75.4 | 73.1 | 72.1 | 0.82 | 0.83 | 0.82 | 0.81 | 0.80 | 0.79 |
| | Simpson | 63.5 | 64.1 | 65.7 | 67.6 | 71.9 | 72.0 | 0.71 | 0.72 | 0.73 | 0.75 | 0.79 | 0.79 |
| | Kulczynski | 75.6 | 74.6 | 73.1 | 71.5 | 70.0 | 69.4 | 0.81 | 0.80 | 0.80 | 0.79 | 0.77 | 0.77 |
| | Johnson | 77.1 | 78.0 | 75.9 | 73.8 | 72.5 | 72.0 | 0.81 | 0.83 | 0.82 | 0.80 | 0.79 | 0.79 |

(user-based) and 4.9% (item-based). Also, such improvements are statistically significant at 99% confidence level. Finally, $CAGG_I$ performs more successful and robust than $CAGG_U$ algorithm for both datasets.

### 5.4.3. Effects of the hybrid algorithm

We conducted the last set of experiments to evaluate the effects of performing NBC algorithm on the reduced user-item matrix filtered by neighbors of both the active user $(a)$ and the target item $(q)$ together. Such neighbors are determined by utilizing both user- and item-based nearest neighbor MCCF algorithms simultaneously, as mentioned in Section 4.3. In these experiments, we utilize the Jaccard metric for computing similarities among users and employ $CAGG_U$ and $CAGG_I$ algorithms to locate neighbors of the active user and the target item, respectively.

As can be followed from the previous section, $CAGG_U$ algorithm performs best with $N = 200$ for YM10 and $N = 60$ for YM20. Similarly, $CAGG_I$ algorithm achieves the best outcomes with $N = 200$ for YM10

**Table 8**. CA values of $CAGG_U$ and $CAGG_I$.

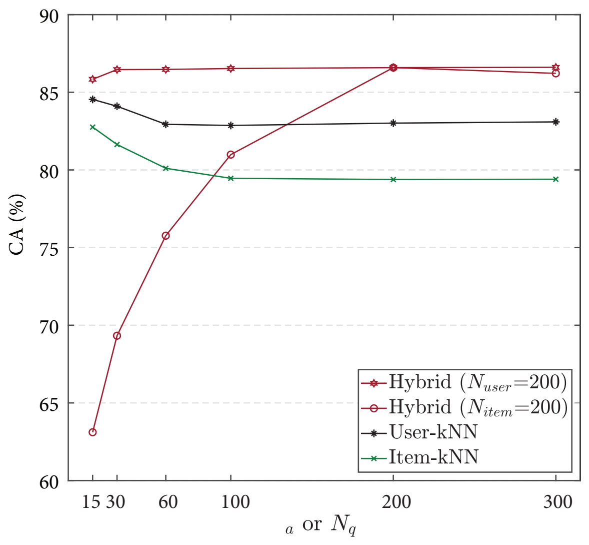
| | | CA | | | | | |
|---|---|---|---|---|---|---|---|
| | $N$ | 15 | 30 | 60 | 100 | 200 | 300 |
| $AGG_U$ | YM10 | 63.1 | 69.2 | 75.1 | 79.0 | 79.8 | 76.7 |
| | YM20 | 69.0 | 76.6 | 79.0 | 75.8 | 72.0 | 71.1 |
| $CAGG_U$ | YM10 | 63.1 | 69.3 | 75.8 | 80.9 | 83.5 | 79.6 |
| | YM20 | 70.3 | 70.3 | 80.0 | 83.9 | 79.4 | 72.6 |
| $AGG_I$ | YM10 | 71.5 | 74.0 | 76.8 | 78.9 | 79.8 | 77.7 |
| | YM20 | 77.7 | 80.4 | 80.9 | 79.3 | 75.4 | 73.1 |
| $CAGG_I$ | YM10 | 71.7 | 74.6 | 78.5 | 82.0 | 84.7 | 82.1 |
| | YM20 | 81.1 | 84.9 | 84.5 | 81.8 | 76.3 | 73.5 |

**Table 9**. F1-score values of $CAGG_U$ and $CAGG_I$.

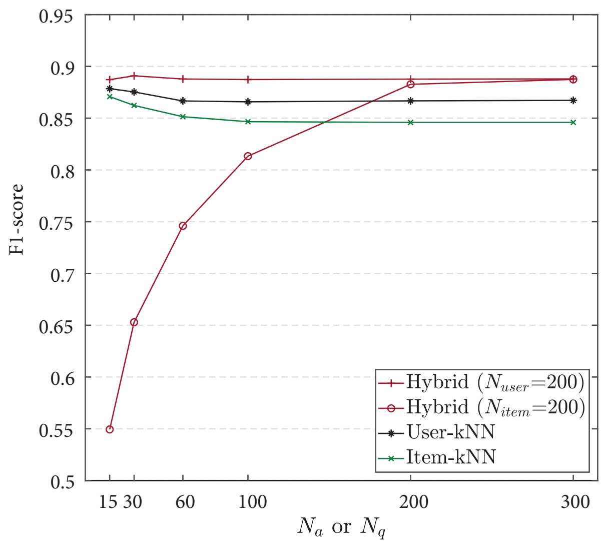
| | | F1-score | | | | | |
|---|---|---|---|---|---|---|---|
| | N | 15 | 30 | 60 | 100 | 200 | 300 |
| $AGG_U$ | YM10 | 0.54 | 0.65 | 0.74 | 0.79 | 0.83 | 0.82 |
| | YM20 | 0.66 | 0.78 | 0.83 | 0.82 | 0.79 | 0.78 |
| $CAGG_U$ | YM10 | 0.54 | 0.65 | 0.74 | 0.81 | 0.86 | 0.85 |
| | YM20 | 0.66 | 0.80 | 0.86 | 0.84 | 0.79 | 0.78 |
| $AGG_I$ | YM10 | 0.77 | 0.79 | 0.81 | 0.83 | 0.84 | 0.83 |
| | YM20 | 0.81 | 0.84 | 0.85 | 0.84 | 0.81 | 0.79 |
| $CAGG_I$ | YM10 | 0.77 | 0.80 | 0.83 | 0.85 | 0.87 | 0.84 |
| | YM20 | 0.84 | 0.87 | 0.87 | 0.86 | 0.82 | 0.80 |

and $N = 30$ for YM20. For this set of experiments, since there are two conditional variables to test, i.e. utilized neighborhood size of the active user and the target item, we fix one of them in each trial and vary the other to see mutual effects. Therefore, for YM10 dataset, we first fix the size of neighbors of the active user ($N_{user}$) to 200 and vary the size of neighbors of the target item ($N_q$). Then, we fix the size of neighbors of the target item ($N_{item}$) to 200 and investigate the performance of the hybrid algorithm with varying the size of neighbors of the active user ($N_a$). Similarly, for YM20, we mutually fix $N_{user}$ to 60 and $N_{item}$ to 30. The CA and F1-score results of the conducted experiments are depicted in Figures 2 and 3 for YM10 and YM20 datasets, respectively. We also compare the empirical results against both benchmark algorithms, i.e. user- and item-based kNN, which are applied solely on the overall preferences without considering other subcriteria ratings. Note that, these algorithms utilize Jaccard metric to calculate similarity among users or items, as well.

The best accuracy results for YM10 are obtained when $N_a$ and $N_q$ are selected as 200 and 300, respectively, which yields CA as 86.60% and F1-score as 0.88. Thus, the empirical results demonstrate that the hybrid algorithm enhances by 3.7% (wrt CA) compared to $CAGG_U$ and 2.2% (wrt CA) compared to $CAGG_I$. Similarly, the best accuracy outcomes for YM20 are achieved when $N_a$ and $N_q$ are 60 and 200, respectively, which yields CA as 86.16% and F1-score as 0.88. Thus, the hybrid algorithm enhances by 2.7% (wrt CA) compared to $CAGG_U$ and 1.5% (wrt CA) compared to $CAGG_I$. Moreover, all of the reported improvements are statistically significant, at least at 95% confidence level.

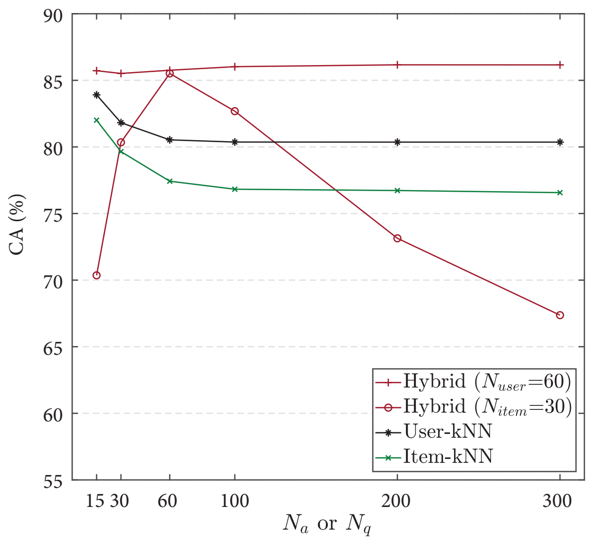


(a) CA Values

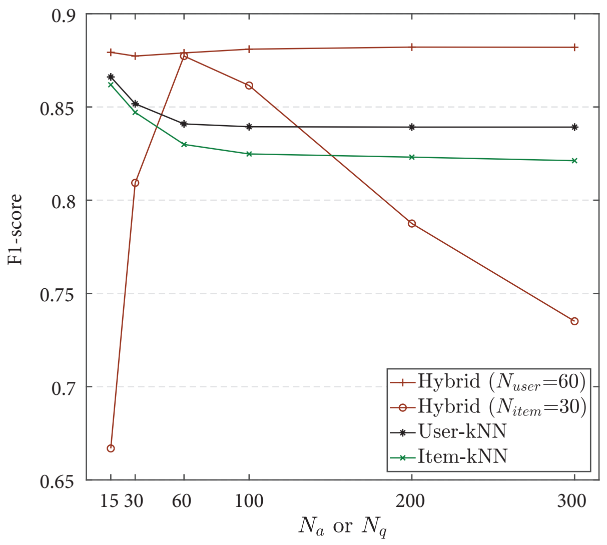


(b) F1-score Values

**Figure 2**. CA and F1-score values of hybrid algorithm for YM10.



(a) CAValues



(b) F1-scoreValues

**Figure 3**. CA and F1-score values of hybrid algorithm for YM20.

As can be seen in both Figures 2 and 3, the hybrid algorithm that fixes $N_{user}$ exhibits relatively better performance in comparison with the hybrid algorithm that fixes $N_{item}$. More specifically, the former algorithm follows almost a stable trend when $N_q$ varies from 15 to 300. On the other hand, the latter algorithm reaches its peak when $N_q$ is 200 and 60 for YM10 and YM20, respectively. Thus, it can be concluded that the dominating factor of the hybrid algorithm is $N_{user}$. Also, the former hybrid algorithm generally outperforms both user-kNN and item-kNN algorithms, and the enhancements are statistically significant at a 95% confidence level for all $N_a$ or $N_q$ values.

In addition to obtained accuracy improvements, the hybrid algorithm drastically reduces the computation time compared to pure-NBC algorithm since it shrinks the user-item matrix in both dimensions. As a result, it can be concluded that the hybrid algorithm is useful for locating neighborhoods appropriately and consequently improving the prediction quality.

#### 5.4.4. Coverage of the proposed algorithms

Up to now, we have performed several trials with varying $N$ values to investigate the accuracy performance of our proposed algorithms. However, it might be challenging to find enough neighbors as $N$ increases, which in turn leads to diminishing the coverage of the system. Therefore, we also present the coverage results of all utilized algorithms in the previously conducted experiments for both datasets in Table 10. Note that, since the $CAGG_U$ and $CAGG_I$ are the variants of $AGG_U$ and $AGG_I$, respectively, they achieve identical coverage results.

**Table 10**. Coverage results of the our proposed and benchmark algorithms for both dataset.

| | | Coverage (%) | | | | | |
|---|---|---|---|---|---|---|---|
| | | N | | | | | |
| Dataset | Algorithm | 15 | 30 | 60 | 100 | 200 | 300 |
| YM10 | User-kNN | 98.1 | 88.2 | 71.3 | 55.2 | 33.5 | 23.3 |
| | Item-kNN | 96.0 | 81.2 | 63.5 | 50.6 | 33.9 | 25.3 |
| | $AGG_U$ or $CAGG_U$ | 96.0 | 95.9 | 95.7 | 95.2 | 92.7 | 87.6 |
| | $AGG_I$ or $CAGG_I$ | 99.7 | 99.4 | 98.8 | 97.9 | 94.6 | 90.0 |
| | Hybrid ($N_{user} = 200$) | 96.2 | 96.0 | 95.7 | 95.2 | 93.6 | 91.3 |
| | Hybrid ($N_{item} = 200$) | 95.3 | 95.3 | 95.2 | 94.9 | 93.6 | 91.1 |
| YM20 | User-kNN | 100.0 | 95.4 | 71.4 | 49.5 | 26.1 | 17.4 |
| | Item-kNN | 100.0 | 95.5 | 76.6 | 59.1 | 36.3 | 25.2 |
| | $AGG_U$ or $CAGG_U$ | 98.6 | 98.6 | 98.6 | 98.4 | 97.0 | 89.8 |
| | $AGG_I$ or $CAGG_I$ | 99.6 | 99.6 | 99.5 | 99.2 | 97.3 | 92.5 |
| | Hybrid ($N_{user} = 60$) | 99.1 | 99.1 | 99.0 | 98.9 | 98.0 | 95.6 |
| | Hybrid ($N_{item} = 30$) | 99.1 | 99.1 | 99.1 | 99.0 | 98.2 | 94.6 |

As can be followed by Table 10, the coverage of the user- and item-based kNN algorithms drastically decrease with increasing $N$. The reason for this consequence is that the former algorithm forms neighbors of the active user by selecting the most similar users from the set of those only rated the target item. In other words, it locates neighborhoods by filtering users who provide a rating for the target item. Similarly, the latter algorithm performs the neighborhood selection process by considering only the set of items that are rated by the active user. Thus, $N$ has a negative effect on the coverage since the chance of finding a greater number of neighbors obviously gets harder due to the sparse nature of preference data.

Although coverage outcomes of our proposed algorithms slightly decrease with increasing $N$, they still achieve significantly better coverage than the benchmark algorithms. This is because the proposed algorithms form neighborhoods by considering all users or items for which a similarity value can be calculated. Also, among the proposed algorithms, the best coverage outcomes are generally obtained with the $CAGG_I$ and $AGG_I$ algorithms. Although the hybrid algorithm selects neighbors of both the active user and the target item at the same time, it achieves comparable coverage results with other proposed algorithms, which indicates the robustness of the hybrid algorithm in terms of coverage.

## 6. Conclusion

Multicriteria preference collection is a useful tool that encourages exploring the interests of users on services/products within multiple aspects. Multicriteria collaborative filtering algorithms utilize such preference collections to capture personal tendencies and produce better personalized referrals. However, collecting multidimensional user preferences might be challenging since it requires a detailed and time-consuming evaluation of a product/service by customers. Gathering user tastes on products via a binary preferences based rating system might be preferable to ease such an evaluation process. In this study, we mainly investigate the applicability of multicriteria collaborative filtering systems relying on binary ratings and introduce novel multicriteria collaborative filtering algorithms.

We first propose two novel similarity aggregation-based multicriteria recommendation algorithms utilizing the naïve Bayes classifier. These algorithms contribute to forming appropriate neighborhoods by employing several binary similarity measures for estimating similarities among either users or items. Then, we improve them by employing the concordance measure, which helps to discover the harmony between overall preference and subcriteria ratings for each user. Considering the correlations between subcriteria and the overall liking of an item provides computing better personalized similarities among users, which leads to more appropriate neighborhood formation. Finally, we introduce a hybrid algorithm that performs the naïve Bayes classifier on a small rating matrix, which is formed by the incorporation of the proposed user- and item-based similarity algorithms. The hybrid algorithm enhances the classification skills of the naïve Bayes algorithm and reduces the required computation time. The experiments conducted on benchmark datasets demonstrate that each proposed algorithm outperforms its rival, as confirmed by statistical significance tests. Moreover, our ultimate hybrid algorithm achieves more accurate prediction results compared to the state-of-the-art user- and item-based $k$-nearest neighbors collaborative filtering approaches. The empirical outcomes also suggest that the neighborhood size is the critical parameter for producing high-quality referrals. Jaccard similarity measure performs significantly better in comparison to other binary similarity metrics due to its sensitivity to the positive rating matches between user rating vectors.

Although the proposed similarity aggregation-based algorithms can handle binary ratings, future research might include developing novel aggregation function-based approaches, which rely on binary ratings and extract the relationships between the overall preference and subcriteria ratings. Also, other classification techniques, such as support vector machines, can be utilized to produce recommendations in the proposed algorithms.

## 7. Acknowledgement

This work is supported by the Scientific Research Project Fund of Anadolu University under grant no. 1508F588.

## References

[1] Recker Mimi M, Andrew W. Supporting “word-of-mouth” social networks through collaborative information filtering. Journal of Interactive Learning Research 2003; 14 (1): 79-98.

[2] Herlocker J, Konstan JA, Riedl J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval 2002; 5 (4): 287-310. doi: 10.1023/A:1020443909834

[3] Kaleli C, Polat H. P2P collaborative filtering with privacy. Turkish Journal of Electrical Engineering & Computer Sciences 2010; 18 (1): 101-116. doi: 10.3906/elk-0808-21

[4] Adomavicius G, Kwon Y. Multi-criteria recommender systems. In: Ricci F, Rokach L, Shapira B, Kantor P (editors). Recommender Systems Handbook. Boston, MA, USA: Springer, 2015, pp. 847-880.

[5] Jannach D, Karakaya Z, Gedikli F. Accuracy improvements for multi-criteria recommender systems. In: Proceedings of the 13th ACM Conference on Electronic Commerce; Valencia, Spain; 2012. pp. 674-689. doi: 10.1145/2229012.2229065

[6] Bilge A, Polat H. Improving privacy-preserving NBC-based recommendations by preprocessing. In: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology; Toronto, ON, Canada; 2010. pp. 143–147. doi: 10.1109/WI-IAT.2010.109

[7] Miyahara K, Pazzani MJ. Collaborative filtering with the simple Bayesian classifier. In: Pacific Rim International Conference on Artificial Intelligence; Melbourne, Australia; 2000. pp. 679-689.

[8] Adomavicius G, Kwon Y. New recommendation techniques for multi-criteria rating systems. IEEE Intelligent Systems 2007; 22 (3): 48-55. doi: 10.1109/MIS.2007.58

[9] Kaur G, Ratnoo S. Adaptive genetic algorithm for feature weighting in multi-criteria recommender systems. Pertanika Journal of Science & Technology 2019; 27 (1): 1-25.

[10] Fan J, Xu L. A robust multi-criteria recommendation approach with preference-based similarity and support vector machine. In: 10th International Symposium on Neural Networks; Dalian, China; 2013. pp. 385-394. doi: 10.1007/978-3-642-39068-5_47

[11] Hassan M, Hamada M. A neural networks approach for improving the accuracy of multi-criteria recommender systems. Applied Sciences 2017; 7 (9): 868. doi: 10.3390/app7090868

[12] Hamada M, Hassan M. Artificial neural networks and particle swarm optimization algorithms for preference prediction in multi-criteria recommender systems. Informatics 2018; 5 (2): 25. doi: 10.3390/informatics5020025

[13] Farokhi N, Vahid M, Nilashi M, Ibrahim O. A multi-criteria recommender system for tourism using fuzzy approach. Journal of Soft Computing and Decision Support Systems 2016; 3 (4): 19-29.

[14] Kant V, Jhalani T, Dwivedi P. Enhanced multi-criteria recommender system based on fuzzy Bayesian approach. Multimedia Tools and Applications 2018; 77: 1-19. doi: 10.1007/s11042-017-4924-2

[15] Majumder G, Dwivedi P, Kant V. Matrix factorization and regression-based approach for multi-criteria recommender system. In: International Conference on Information and Communication Technology for Intelligent Systems; Ahmedabad, India; 2018. pp. 103-110. doi: 10.1007/978-3-319-63673-3_13

[16] Batmaz Z, Kaleli C. AE-MCCF: an autoencoder-based multi-criteria recommendation algorithm. Arabian Journal for Science and Engineering 2019; 44 (11): 9235-9247. doi: 10.1007/s13369-019-03946-z

[17] Manouselis N, Costopoulou C. Experimental analysis of design choices in multiattribute utility collaborative filtering. International Journal of Pattern Recognition and Artificial Intelligence 2007; 21 (02): 311-331. doi: 10.1142/S021800140700548X

[18] Hu Y, Chiu Y, Liao Y, Li Q. A fuzzy similarity measure for collaborative filtering using nonadditive grey relational analysis. The Journal of Grey System 2015; 27 (2): 93-104.

[19] Naak A, Hage H, Aimeur E. A multi-criteria collaborative filtering approach for research paper recommendation in papyres. In: International Conference on E-Technologies; Ottawa, ON, Canada; 2009. pp. 25-39. doi: 10.1007/978-3-642-01187-0_3

[20] Hu Y, Chiu Y, Tsai J. Establishing grey criteria similarity measures for multi-criteria recommender systems. Journal of Grey System 2018; 30 (1): 194-207.

[21] Liu L, Mehandjiev N, Xu D. Multi-criteria service recommendation based on user criteria preferences. In: Proceedings of the Fifth ACM Conference on Recommender Systems; Chicago, IL, USA; 2011. pp. 77-84. doi: 10.1145/2043932.2043950

[22] Lakiotaki K, Tsafarakis S, Matsatsinis N. UTA-Rec: a recommender system based on multiple criteria analysis. In: Proceedings of the 2008 ACM Conference on Recommender Systems; Lausanne, Switzerland; 2008. pp. 219-226. doi: 10.1145/1454008.1454043

[23] Hu Y. Neighborhood-based collaborative filtering using grey relational analysis. Journal of Grey System 2014; 26 (1): 99-114.

[24] Bilge A, Kaleli C. A multi-criteria item-based collaborative filtering framework. In: 11th International Joint Conference on Computer Science and Software Engineering (JCSSE); Chon Buri, Thailand; 2014. pp. 18-22. doi: 10.1109/JCSSE.2014.6841835

[25] Hwang W. Assessing new correlation-based collaborative filtering approaches for binary market basket data. Electronic Commerce Research and Applications 2018; 29: 12-18. doi: 10.1016/j.elerap.2018.03.002

[26] Verstrepen K, Bhaduriy K, Cule B, Goethals B. Collaborative filtering for binary, positive-only data. ACM SIGKDD Explorations Newsletter 2018; 19 (1): 1-21. doi: 10.1145/3137597.3137599

[27] Miyahara K, Pazzani MJ. Improvement of collaborative filtering with the simple Bayesian classifier. Information Processing Society of Japan 2002; 43 (11): 1-20.

[28] Ng AY, Jordan MI. On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic; Vancouver, BC, Canada; 2001. pp. 841-848.

[29] Choi SS, Cha SH, Tappert CC. A survey of binary similarity and distance measures. Journal of Systemics, Cybernetics and Informatics 2010; 8 (1): 43-48.

[30] Lathia N, Hailes S, Capra L. Private distributed collaborative filtering using estimated concordance measures. In: Proceedings of the 2007 ACM Conference on Recommender Systems; Minneapolis, MN, USA; 2007. pp. 1-8. doi: 10.1145/1297231.1297233