**New similarity measures for collaborative filtering**

Ali A. Amer

TAIZ University, Yemen

Loc Nguyen

Loc Nguyen’s Academic Network, Vietnam

**Abstract**

Collaborative filtering (CF) is an important approach for recommendation system. CF recommends an item to a user if her/his neighbors are interested in such item. One of popular algorithms in CF is nearest neighbors (NN) algorithm. Finding neighbors of a user is main problem of NN algorithm. As usual, similarity measures between two users such as cosine and Pearson correlation are used to determine nearest neighbors of a user. This research proposes three new similarity measures called Amer, Amer2, and QTIJ with note that QTIJ is an extension of Amer measure and it follows ideology of term frequency (TF) and inverse document frequency (IDF) in information retrieval. As a convention, Amer family includes measures related to Amer measure. We test Amer family with many other measures. From experiments, Amer family is preeminent when its measures are in top-5 lists of good measures in which Amer is the best with dataset which has more missing values and QTIJ is the best with dataset which has fewer missing values.

**Keywords:** collaborating filtering (CF), term frequency (TF), inverse document frequency (IDF), similarity measure, nearest neighbors (NN) algorithm, rating matrix.

**1. Introduction**

Recommendation system is a system which recommends items to users among many existing items in database. Item is anything which users consider, such as product, book, and newspaper. There are two main approaches for recommendation such as content-based filtering (CBF) and collaborative filtering (CF). CF recommends an item to a user if her/his neighbors (other users like her/him) are interested in such item. One of popular algorithms in CF is nearest neighbors (NN) algorithm. The essence of NN algorithm (Torres Júnior, 2004, pp. 16-18) is to find out nearest neighbors of a regarded user (called active user) and then to recommend active user items that these neighbors may like. Let ***U*** = {*u*1, *u*2,…, *um*} be the set of users and let ***V*** = {*v*1, *v*2,…, *vn*} be the set of items. User-based rating matrix is the matrix in which rows indicate users and columns indicate items and each cell is a rating which a user gave to an item. In other words, each row in user-based rating matrix is a rating vector of a specified user. Rating vector of active user is called active user vector. As a convention, rating matrix implies user-based rating matrix if there is no additional explanation. Table 1.1 is an example of user-based rating matrix in which missing values are denoted by question masks (Do, Nguyen, & Nguyen, 2010, p. 218). In table 1.1, active vector is *u*4 = (*r*41=1, *r*42=2, *r*43=?, *r*44=?), which is shaded.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Item 1 | Item 2 | Item 3 | Item 4 |
| User 1 | *r*11 = 1 | *r*12 = 2 | *r*13 = 1 | *r*14 = 5 |
| User 2 | *r*21 = 2 | *r*22 = 1 | *r*23 = 2 | *r*24 = 4 |
| User 3 | *r*31 = 4 | *r*32 = 1 | *r*33 = 5 | *r*34 = 5 |
| User 4 | *r*41 = 1 | *r*42= 2 | *r*43 = ? | *r*44 = ? |

**Table 1.1.** User-based rating matrix

User-based rating matrix can be transposed into item-based rating matrix in which each row is a rating vector of a specified item. Table 1.2 is the item-based rating matrix which is transposed from the user-based rating matrix shown in table 1.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | User 1 | User 2 | User 3 | User 4 |
| Item1 | *r*11 = 1 | *r*21 = 2 | *r*31 = 4 | *r*41 = 1 |
| Item2 | *r*12 = 2 | *r*22 = 1 | *r*32 = 1 | *r*42 = 2 |
| Item3 | *r*13 = 1 | *r*23 = 2 | *r*33 = 5 | *r*43 = ? |
| Item4 | *r*14 = 5 | *r*24 = 4 | *r*34 = 5 | *r*44 = ? |

**Table 1.2.** Item-based rating matrix

In table 1.2, active item vectors are *v*3 = (*r*13=1, *r*23=2, *r*33=5, *r*43=?) and *v*4 = (*r*14=5, *r*24=4, *r*34=5, *r*44=?), which are shaded.

In table 1.1, there are four rating vectors *u*1 = (1, 2, 1, 5), *u*2 = (2, 1, 2, 4), *u*3 = (4, 1, 5, 5), and *u*4 = (1, 2, *r*43=?, *r*44=?). Suppose the active rating vector is *u*4, NN algorithm will find out nearest neighbors of *u*4 and then compute the predictive values for *r*43 and *r*44 based on similarities between these neighbors and *u*4. NN algorithm acts on user-based rating matrix is called user-based NN algorithm and NN algorithm acts on item-based rating matrix is called item-based NN algorithm. Although the ideology of user-based NN algorithm and item-based NN algorithm is the same, their implementations are slightly different. We mention user-based NN algorithm by default. In general, NN algorithm includes two steps (Torres Júnior, 2004, pp. 17-18):

1. Find out nearest neighbors of the active user by calculating similarities between active vector and other vectors. The more the similarity is, the nearer two users are. Given a threshold, users whose similarities between them and active user are equal to or larger than a threshold are considered as nearest neighbors of active user.
2. Compute predictive values for missing ratings of active vector. The computation is based on ratings of nearest neighbors and similarities calculated in step 1.

The essence of NN algorithm is to use similarity measures in order to find out nearest neighbors of an active rating vector. This research focuses on similarity measures for CF. The most popular similarity measures are cosine and Pearson. Given two rating vectors *u*1 = (*r*11, *r*12,…, *r*1*n*) and *u*2 = (*r*21, *r*22,…, *r*2*n*) of user 1 and user 2, in which user 1 is considered as active user and some *rij* can be missing (empty). Let *I*1 and *I*2 be set of indices of items that user 1 and user 2 rated, respectively. Let denote intersection set of *I*1 and *I*2 and let denotes union set of *I*1 and *I*2. All items which belong to are rated by both user 1 and user 2. In other words, all items which belong to co-exist in vectors *u*1 and *u*2. All items which belong to are rated by user 1 or user 2. Notation |x| indicates absolute value of number, length of vector, length of segment, or cardinality of set, which depends on context. Please pay attention to these denotations.

Let sim(*u*1, *u*2) denote the similarity of *u*1 and *u*2. For instance, the cosine measure of *u*1 and *u*2 is defined as follows (Torres Júnior, 2004, p. 17):

Where |*u*1| and |*u*2| are lengths of *u*1 and *u*2, respectively whereas *u*1•*u*2 is dot product (scalar product) of *u*1 and *u*2, respectively.

If all ratings are non-negative, range of cosine measure is from 0 to 1. If it is equal to 0, two users are totally different. If it is equal to 1, two users are identical. Cosine measure will be mentioned more later. The larger the similarity is, the more the user 2 is near to active user 1. Hence, the similarity is used to determine the list of neighbors of active user. Suppose NN algorithm finds out *k* neighbors of *u*1, let *N* be set of indices of *k* neighbors of *u*1. Of course, we have |*N*| = *k*. A missing value *r*1*j* of *u*1 is computed based on ratings of nearest neighbors and similarities according to step 2 of NN algorithm (Torres Júnior, 2004, p. 18).

Where and are mean values of *u*1 and *ui*, respectively.

Where *Ii* is the set of indices of items that user *i* rated.

Pearson correlation is another popular similarity measure, which is defined as follows (Sarwar, Karypis, Konstan, & Riedl, 2001, p. 290):

Where and are mean values of *u*1 and *u*2, respectively.

The range of Pearson measure is from –1 to 1. If it is equal to –1, two users are totally opposite. If it is equal to 1, two users are identical. Pearson measure is sample correlation coefficient in statistics.

Jaccard measure, which is another popular measure, is ratio of cardinality of common set to cardinality of union set . It measures how much common items both users rated, which is defined as follows (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158):

Another version of Jaccard is (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158):

Cosine measure has some variants. By following the ideology of Jaccard measure, we can modify cosine measure as a so-called COJ measure.

Obviously, the numerator of COJ is equal to or smaller than the denominator of COJ and hence, COJ measure is equal to or smaller than cosine measure.

Let *rm* be median of rating values, for example, if rating values range from 1 to 5, the median is *rm* = (1+5) / 2 = 3. The normalized cosine measure (CON) (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158) is defined as follows:

When the median *rm* is replaced by mean of rating values which are rated on given item, the normalized cosine measure becomes adjusted cosine measure. CON measure is also Constrained Pearson correlation (CPC) measure mentioned in (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158).

Let *vj* = (*vj*1, *vj*2,…, *vjm*) be vector of rating values that item *j* receives from *m* users, for example. The mean of *vj* is:

Adjusted cosine measure (COD) is specified as follows:

Pearson measure also has some variants. Constrained Pearson correlation (CPC) measure considers impact of positive and negative ratings by using median *rm* instead of using the means; for example, if rating values range from 1 to 5, the median is *rm* = (1+5) / 2 = 3. CPC measure is defined as follows (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158):

The similarity will be significant if both users rated more common items. Weight Pearson correlation (WPC) measure and sigmoid Pearson correlation (SPC) measure concern how much common items are. WPC and SPC are formulated as follows (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158):

Where *H* is an threshold and it is often set to be 50 (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158).

Mean squared difference (MSD) is defined as inverse of distance between two vectors. Let MAX be maximum value of ratings, MSD is calculated as follows (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158):

MSD measure combines with Jaccard measure, which derives MSDJ measure as follows (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158):

There are some other researches related to apply similarity measures into CF. Liu et al. (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 156) proposed a new similarity measure called NHMS to improve recommendation task in which only few ratings are available. Their NHMS measure (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 160) is based on sigmoid function and the improved PIP measure as PSS (*Proximity* – *Significance* – *Singularity*). PSS similarity is calculated as follows (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 160):

Where, is intersection set of *I*1 and *I*2. The proximity factor determines similarity of two ratings, based on distance between them; such distance is as less as better. The significance factor determines similarity of two ratings, based on distance from them to rating median; such distance is as more as better. The significance factor determines similarity of two ratings, based on difference between them and other ratings; such difference is as less as better. Followings are equations of these factors based on sigmoid function (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 161).

Note, *rm* be median of rating values, for example, if rating values range from 1 to 5, the median is *rm* = (1+5) / 2 = 3 whereas *μj* is rating mean of item *j*. Liu et al. (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 161) also considered the similarity between two users via URP measure as follows:

Where *μ*1 and *μ*2 are rating means of user 1 and user 2, respectively and *σ*1 and *σ*2 are rating standard deviations of user 1 and user 2, respectively.

Liu et al. (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 161) proposed NHMS as triple product of PSS measure, URP measure, and Jaccard2 measure.

In general, Liu et al. (Liu, Hu, Mian, Tian, & Zhu, 2013) aim to alleviate the problem of few rated common items via their NHMS measure. From experimental result, NHMS gave out excellent estimation.

Patra et al. (Patra, Launonen, Ollikainen, & Nandi, 2015, p. 143) proposed a new similarity measure called BCF for CF, which uses all ratings made by a pair of users. Proposed measure finds importance of each pair of rated items by exploiting Bhattacharyya (BC) similarity. The BC similarity, which is core of their own measure, measures the similarity between two distributions. So, these distributions are estimated as the number of uses rated on given item. In general, Patra et al. (Patra, Launonen, Ollikainen, & Nandi, 2015, p. 5) combined BC similarity and the local similarity where the local similarity relates to Pearson correlation. It is necessary to survey BC similarity. Bin is a terminology indicating domain of rating values, for example, if rating values range from 1 to 5, we have bins: 1, 2, 3, 4, 5. Let *m* be the number of bins, given items *i* and *j*, item BC similarity for items is calculated as follows (Patra, Launonen, Ollikainen, & Nandi, 2015, p. 5):

Note, #*i* and #*j* are the numbers of users who rated items *i* and *j*, respectively whereas #*hi* and #*hj* are numbers of users who gave rating value *h* on items *i* and *j*, respectively. So, BC similarity concerns two items. According to Patra et al. (Patra, Launonen, Ollikainen, & Nandi, 2015, p. 5), user BC similarity is sum of products of item BC similarities and local similarities as follows:

The local similarity is calculated as a part of constrained Pearson coefficient (CPC) as follows:

Note, *rm* be median of rating values, for example, if rating values range from 1 to 5, the median is *rm* = (1+5) / 2 = 3. Patra et al. (Patra, Launonen, Ollikainen, & Nandi, 2015, p. 5) proposed Bhattacharyya coefficient in CF (BCF) as sum of user BC similarity and Jaccard measure as follows:

This research also implements the Similarity Measure for Text Processing (SMTP) for testing. SMTP was developed by Lin, Jiang, and Lee (Lin, Jiang, & Lee, 2013), which as originally used for computing the similarity between two documents in text processing. Here we consider documents as rating vectors. Given two rating vectors *u*1 = (*r*11, *r*12,…, *r*1*n*) and *u*2 = (*r*21, *r*22,…, *r*2*n*), the function *F* of *u*1 and *u*2 is defined as follows (Lin, Jiang, & Lee, 2013, p. 1577):

Where (Lin, Jiang, & Lee, 2013, p. 1577),

Note that *λ* is the pre-defined number and *σj* is the standard deviation of rating values belonging to field *j* (item *j*). In this research, *λ* is set to be 0.5. Lin, Jiang, and Lee (Lin, Jiang, & Lee, 2013, p. 1577) defined SMTP measure based on function *F* as follows:

In this research, we proposed new measures for CF which follow ideology of term frequency (TF) and inverse document frequency (IDF) in information retrieval.

**2. New similarity measures**

Given two rating vectors *u*1 = (*r*11, *r*12,…, *r*1*n*) and *u*2 = (*r*21, *r*22,… , *r*2*n*) of user 1 and user 2, respectively, in which some *rij* can be missing (empty). In binary representation, *rij* is converted into 1 if it is non-missing (rated) and otherwise, *rij* is converted into 0 if it is missing (not rated). Let *N*12 be the number of common values “1” in both *u*1 and *u*2. Let *N* be the total number of all items under consideration; in this case, *N* = *n*. Let *N*1 and *N*2 be the numbers of values “1” of *u*1 and *u*2, respectively. Let *F* be the number of differences between *u*1 and *u*2; for example, the fact that *r*11 = 0 and *r*21 = 1 contributes one difference to *F*. Ali Amer defined the similarity measure Amer in binary representation as follows:

|  |  |
| --- | --- |
|  | (2.1) |

Let *I*1 or *I*2 be sets of indices of items that user 1 or user 2 rates, respectively. Ali Amer also defined the similarity measure Amer2 in numerical representation in which values *rij* are kept in numerical values as rating values, as follows:

|  |  |
| --- | --- |
|  | (2.2) |

Where, *R*1 (*R*2) is the sum of non-missing values *r*1*j* (*r*2*j*) of *u*1 (*u*2) such that respective values *r*2*j* (*r*1*j*) are missing.

Note, notation “\” denote complement operator in set theory. *G* is product of two sums of non-missing values for both *r*1 and *r*2.

In general, measures Amer and Amer2 are defined firstly for weight vectors of documents in information retrieval, in which every element of a vector is a weight which is product of term frequency (TF) and inverse document frequency (IDF). Here they are applied into CF. For example, given two rating vectors *u*1 = (*r*11=2, *r*12=5, *r*13=7, *r*14=8, *r*15=?, *r*16=9) and *u*2 = (*r*21=9, *r*22=?, *r*23=?, *r*24=6, *r*25=5, *r*26=1). Binary representations of these two vectors are (1, 1, 1, 1, 0, 1) and (1, 0, 0, 1, 1, 1). According to Amer measure, we have *N*12=3, *N*=6, *F*=3, *N*1=5, and *N*2=4. Hence, Amer measure is calculated according to equation 2.1, as follows:

According to Amer2 measure, we have *R*1 = 5+7 = 12, *R*2 = 5, and *G* = (2 + 5 + 7 + 8 + 9) \* (9 + 6 + 5 + 1) = 651. Hence, Amer2 measure is calculated according to equation 2.2, as follows:

When Amer2 measure is combined with Jaccard measure, it is called Amer2J which is specified by equation 2.3.

|  |  |
| --- | --- |
|  | (2.3) |

We develop a new measure which is based on Amer2 measure and ideology of TF and IDF. Firstly, we research deeply Amer2 measure in which the ratio *R*1*R*2/G indicates difference between *u*1 and *u*2. In other words, such ratio implies uniqueness of each vector, which means that the ratio *R*1*R*2/*G* follows ideology of document frequency (DF) in information retrieval. Hence, essentially Amer2 measure is a *quasi-IDF*. Here we re-define the quasi-IDF as follows:

Note, notation “\” denotes complement operator in set theory. Similarly, following ideology of term frequency (TF), we define a so-called quasi-TF as follows:

Note, notation “” denotes intersection operator in set theory. The new measure called quasi-TF-IDF (QTI) is product of the quasi-TF and the quasi-IDF, according to equation 2.4.

|  |  |
| --- | --- |
|  | (2.4) |

QTI measure combines with Jaccard measure, which derives QTIJ measure according to equation 2.5.

|  |  |
| --- | --- |
|  | (2.5) |

Recall that Jaccard measure is mentioned in introduction section. Equation 2.5 is written in simple way as follows:

Where,

Given two rating vectors *u*1 = (*r*11=2, *r*12=5, *r*13=7, *r*14=8, *r*15=?, *r*16=9) and *u*2 = (*r*21=9, *r*22=?, *r*23=?, *r*24=6, *r*25=5, *r*26=1), we calculate QTIJ as an example. Of course, we have *I*1 = {1, 2, 3, 4, 6} and *I*2 = {1, 4, 5, 6}. We also have:

This implies:

By default, all measures are calculated based on user-based rating matrix in which every vector is user rating vector. When user-based rating matrix is transposed into item-based rating matrix in which every vector is item rating vector, equations for these measures are not changed in semantics. In experimental section, these measures are tested with both user-based rating matrix and item-based rating matrix. NN algorithm for user-based rating matrix becomes user-based NN algorithm and NN algorithm for item-based rating matrix becomes item-based NN algorithm.

**3. Experimental results and discussions**

As a convention, Amer family includes measures such as Amer, Amer2, Amer2J, QTI, and QTIJ. Amer family is tested with cosine family (cosine, COJ, CON, COD), Pearson family (Pearson, WPC, SPC), Jaccard, MSD family (MSD, MSDJ), NHSM, BCF, and SMTP. Two metrics used to test measures are MAE and CC. MAE, which is abbreviation of mean absolute error, which is average absolute deviation between predictive ratings and users’ true ratings. Given tested vector *ut* = (1, 2, 3) having three items, we make *ut* empty as empty vector *u*’ = (?, ?, ?) with missing values. Later, we apply NN algorithm with measures above into predicting (estimating) missing values. As a result, we obtain predictive vector (estimated vector), for example, *up* = (2, 3, 4) having three estimated items. Hence, MAE metric is (|2-1| + |3-2| + |4-3|) / 3 = 1. In general, MAE is calculated by equation 3.1 (Herlocker, Konstan, Terveen, & Riedl, 2004, pp. 20-21) in which *n* is the total number of estimated items while *pj* and *vj* are predictive rating and true rating of item *j*, respectively.

|  |  |
| --- | --- |
|  | (3.1) |

The smaller MAE is, the better the measures are. CC, which is abbreviation of correlation coefficient, is used to evaluate correlation between tested vector and predictive vector. It is really Pearson correlation. The larger CC is, the better the measures are. CC is calculated by equation 3.2 (Herlocker, Konstan, Terveen, & Riedl, 2004, p. 30) in which *n* is the total number of estimated items while *pj* and *vj* are predictive rating and true rating of item *j*, respectively.

|  |  |
| --- | --- |
|  | (3.2) |

Where and are mean values of tested item and predictive item, respectively.

MAE evaluates accuracy of measures whereas CC evaluates adequacy of measures. They are not opposite each other but biases between them can vary in tests. All measures are tested with both user-based NN algorithm (for user-based rating matrix) and item-based NN algorithm (for item-based rating matrix). Although the ideology of user-based NN algorithm and item-based NN algorithm is the same, their implementations are slightly different.

Dataset Movielens (GroupLens, 1998) is used for evaluation, which has 100,000 ratings from 943 users on 1682 movies (items). Every rating ranges from 1 to 5. In the experiments, dataset Movielens is divided into 5 folders and each folder includes training set and testing. Training set and testing set in the same folder are disjoint sets. The ratio of testing set over the whole dataset depends on the testing parameter *r*. For instance, if *r* = 0.1, the testing set covers 10% the dataset, which means that the testing set has 10,000 = 10%\*100,000 ratings and of course the training set has 90,000 ratings. In this testing, parameter *r* has three values 0.1, 0.5, and 0.9. The smaller *r* is, the more accurate measures are because training set gets large if *r* gets small with note that NN algorithm is executed on training set.

Table 3.1 shows all tested measures with *r*=0.1. The parameter value *r*=0.1 implies that testing set is here minimum. Each folder has own tested measures and so tested measures shown here are made average over 5 folders. Shaded cells indicate best values. We should select item-based MAE as the main referred metric because it is smallest and item-based NN algorithm is better than user-based NN algorithm. Some other authors also confirm the preeminence of item-based NN algorithm. The reason may be that items which are goods in e-commerce are more stable than users as customers. Moreover, dataset Movielens has more items than users.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *r* = 1 | MAE  (User-based) | MAE  (Item-based) | CC  (User-based) | CC  (Item-based) |
| Cosine | 0.7532 | 0.7427 | 0.4185 | 0.4217 |
| COJ | 0.7457 | 0.7317 | 0.3881 | 0.4004 |
| CON | 0.7469 | 0.7468 | 0.4339 | 0.4342 |
| COD | 0.8224 | 0.7894 | 0.1810 | 0.4403 |
| Pearson | 0.7395 | 0.7447 | 0.4355 | 0.4429 |
| WPC | 0.7312 | 0.7348 | 0.4519 | 0.4549 |
| SPC | 0.7388 | 0.7434 | 0.4378 | 0.4459 |
| Jaccard | 0.7465 | 0.7300 | 0.4273 | 0.4444 |
| MSD | 0.7529 | 0.7420 | 0.4192 | 0.4226 |
| MSDJ | 0.7457 | 0.7289 | 0.4288 | 0.4457 |
| NHSM | 0.7410 | 0.7219 | 0.4343 | 0.4579 |
| BCF | 0.7984 | 0.7822 | 0.3073 | 0.3978 |
| SMTP | 0.7533 | 0.7465 | 0.4185 | 0.4145 |
| Amer | 0.7524 | 0.7412 | 0.4180 | 0.4246 |
| Amer2 | 0.7481 | 0.7376 | 0.4257 | 0.4321 |
| Amer2J | 0.7427 | 0.7262 | 0.3880 | 0.4132 |
| QTI | 0.7396 | 0.7233 | 0.3912 | 0.4179 |
| QTIJ | 0.7375 | 0.7160 | 0.3941 | 0.4305 |

**Table 3.1.** Measures with *r*=0.1

From table 3.1, QTIJ is the best with lowest item-based MAE and NHSM is the best with highest item-based CC whereas WPC is the best with lowest user-based MAE and highest user-based CC. Top-5 measures according to item-based MAE are QTIJ (1), NHSM (2), QTI (3), Amer2J (4), and MSDJ (5). Top-5 measures according to item-based CC are NHSM, WPC, SPC, MSDJ, and Jaccard. Top-5 measures according to user-based MAE are WPC, QTIJ, SPC, Pearson, QTI. Top-5 measures according to user-based CC are WPC, SPC, Pearson, NHSM, and CON. In general, QITJ in Amer family is the preeminent measure with MAE given *r*=0.1. Moreover, three measures in Amer family such as QTIJ, QTI, and Amer2J are in top-5 measures according to MAE.

Table 3.2 shows all tested measures with *r*=0.5. The parameter value *r*=0.5 implies that testing set is here medium.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *r* = 0.5 | MAE  (User-based) | MAE  (Item-based) | CC  (User-based) | CC  (Item-based) |
| Cosine | 0.7630 | 0.7541 | 0.3784 | 0.3776 |
| COJ | 0.7572 | 0.7450 | 0.3855 | 0.3948 |
| CON | 0.7657 | 0.7681 | 0.3525 | 0.3773 |
| COD | 0.8441 | 0.8099 | 0.0678 | 0.3672 |
| Pearson | 0.7734 | 0.7774 | 0.2905 | 0.3738 |
| WPC | 0.7581 | 0.7613 | 0.3265 | 0.3893 |
| SPC | 0.7708 | 0.7743 | 0.3013 | 0.3770 |
| Jaccard | 0.7583 | 0.7440 | 0.3803 | 0.3950 |
| MSD | 0.7627 | 0.7535 | 0.3793 | 0.3779 |
| MSDJ | 0.7575 | 0.7429 | 0.3822 | 0.3963 |
| NHSM | 0.7545 | 0.7376 | 0.3841 | 0.4040 |
| BCF | 0.8061 | 0.7934 | 0.2530 | 0.3457 |
| SMTP | 0.7629 | 0.7562 | 0.3789 | 0.3698 |
| Amer | 0.7631 | 0.7537 | 0.3777 | 0.3791 |
| Amer2 | 0.7588 | 0.7500 | 0.3835 | 0.3873 |
| Amer2J | 0.7562 | 0.7411 | 0.3799 | 0.4003 |
| QTI | 0.7557 | 0.7400 | 0.3757 | 0.4015 |
| QTIJ | 0.7581 | 0.7380 | 0.3610 | 0.3990 |

**Table 3.2.** Measures with *r*=0.5

From table 3.2, NHSM is the best with lowest item-based MAE, highest item-based CC, and lowest user-based MAE, which is an incredible measure. COJ is the best with highest user-based CC. Top-5 measures according to item-based MAE are NHSM (1), QTIJ (2), QTI (3), Amer2J (4), and MSDJ (5). Top-5 measures according to item-based CC are NHSM, QTI, Amer2J, QTIJ, and MSDJ. Top-5 measures according to user-based MAE are NHSM, QTI, Amer2J, cosine, and MSDJ. Top-5 measures according to user-based CC are COJ, NHSM, Amer2, MSDJ, and Jaccard. From table 3.2, pure TA is better than pure cosine with respect to both MAE and CC. Although QTIJ is not the best over top-5 lists, Amer family is preeminent because most its measures such as QTIJ, QTI, Amer2J, and Amer are in top-5 lists.

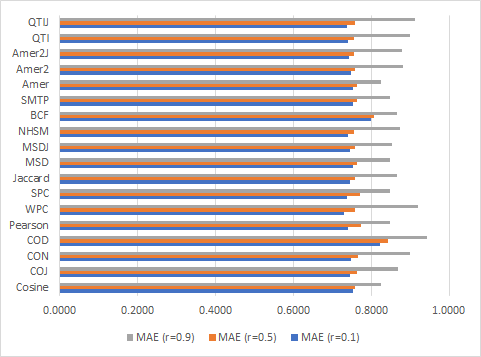
Table 3.3 shows all tested measures with *r*=0.9. The parameter value *r*=0.9 implies that testing set is here maximum.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *r* = 0.9 | MAE  (User-based) | MAE  (Item-based) | CC  (User-based) | CC  (Item-based) |
| Cosine | 0.8255 | 0.8177 | 0.3004 | 0.3154 |
| COJ | 0.8681 | 0.8597 | 0.2051 | 0.2620 |
| CON | 0.8985 | 0.8813 | 0.1403 | 0.2273 |
| COD | 0.9423 | 0.9098 | 0.0178 | 0.2243 |
| Pearson | 0.8473 | 0.8326 | 0.2530 | 0.2785 |
| WPC | 0.9202 | 0.9088 | 0.0569 | 0.2412 |
| SPC | 0.8490 | 0.8341 | 0.2476 | 0.2774 |
| Jaccard | 0.8651 | 0.8566 | 0.2058 | 0.2637 |
| MSD | 0.8471 | 0.8388 | 0.2335 | 0.2714 |
| MSDJ | 0.8538 | 0.8422 | 0.2248 | 0.2673 |
| NHSM | 0.8729 | 0.8583 | 0.2121 | 0.2547 |
| BCF | 0.8658 | 0.8568 | 0.1897 | 0.2815 |
| SMTP | 0.8478 | 0.8425 | 0.2328 | 0.2674 |
| Amer | 0.8253 | 0.8174 | 0.3008 | 0.3158 |
| Amer2 | 0.8812 | 0.8702 | 0.1617 | 0.2515 |
| Amer2J | 0.8791 | 0.8681 | 0.1955 | 0.2553 |
| QTI | 0.8983 | 0.8852 | 0.1832 | 0.2459 |
| QTIJ | 0.9125 | 0.8976 | 0.1732 | 0.2400 |

**Table 3.3.** Measures with *r*=0.9

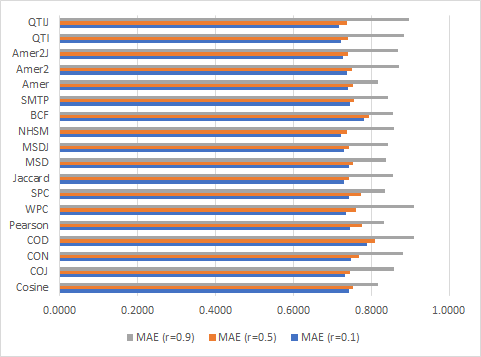
Testing results from table 3.3 are unexpected because the parameter value *r*=0.9 makes noise NN algorithm when the training set is not large enough. Amer is the best with lowest MAE and highest CC. Top-5 measures according to item-based MAE are Amer (1), cosine (2), Pearson (3), SPC (4), and MSD (5). Top-5 measures according to item-based CC are Amer, cosine, BCF, Pearson, and SPC. Top-5 measures according to user-based MAE are Amer, cosine, MSD, Pearson, and SMTP. Top-5 measures according to user-based CC are Amer, cosine, Pearson, SPC, and MSD. Measures like BCF and SMTP whose accuracy is low with *r*=0.1 and *r*=0.5 get now better. BCF aims to take advantages of statistical features and so it can resist lack of data. Similarly, Pearson, SPC, and MSD are now in top-5 measures because they also take advantages of statistical feature (correlation coefficient, sample variance). Amer measure itself proved it as a preeminent measure over top-5 lists. Moreover, the testing results also proved reliability of traditional measures like cosine and Pearson although they are not always preeminent in all cases.

Figure 3.1 shows tested measures with MAE metric and user-based rating matrix related to *r* = 0.1, 0.5, 0.9.



**Figure 3.1.** Measures with user-based MAE related to *r* = 0.1, 0.5, 0.9

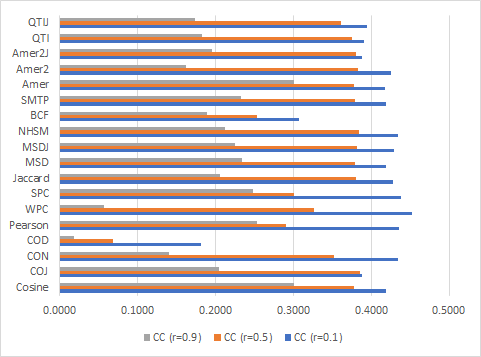
Figure 3.2 shows tested measures with MAE metric and item-based rating matrix related to *r* = 0.1, 0.5, 0.9.



**Figure 3.2.** Measures with item-based MAE related to *r* = 0.1, 0.5, 0.9

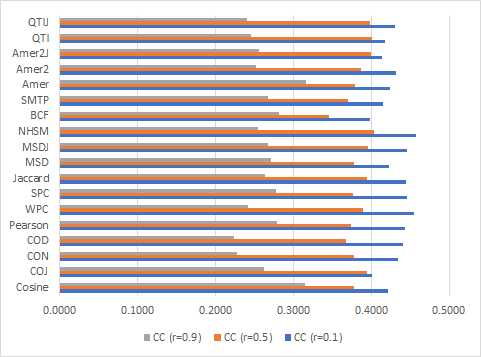
Figures 3.1 and 3.2 show the correlation of measures within MAE when parameter *r* is changed. The accuracy of measures decreases unexpectedly when *r* approaches 0.9. This is reasonable because the parameter value *r*=0.9 implies the training set is too small to train NN algorithm. Conversely, the parameter value *r*=0.1 is adequate to real-time application which has large rating database. The parameter value *r*=0.5 is adequate to testing application. Therefore, the lower the parameter value *r* is, the more adequate to real-time application the measures are.

Figure 3.3 shows tested measures with CC metric and user-based rating matrix related to *r* = 0.1, 0.5, 0.9.



**Figure 3.3.** Measures with user-based CC related to *r* = 0.1, 0.5, 0.9

Figure 3.4 shows tested measures with CC metric and item-based rating matrix related to *r* = 0.1, 0.5, 0.9.



**Figure 3.4.** Measures with item-based CC related to *r* = 0.1, 0.5, 0.9

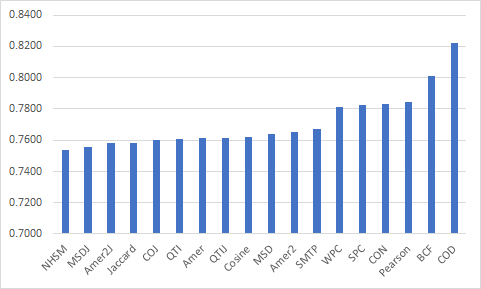
Figures 3.3 and 3.4 show the correlation of measures within CC when the parameter *r* is changed. The chart confirms the unexpected decrease of adequacy when *r* approaches 0.9. The reason was explained above.

Over figures 3.1, 3.2, 3.3, and 3.4, QTIJ measure is dominant with *r*=0.1 and *r*=0.5 whereas Amer measure is dominant with *r*=0.9. In general, Amer family is preeminent. Three values of *r* such as 0.1, 0.5, and 0.9 are enough for us to survey all measures because these values are key values. For instance, the minimum value *r*=0.1 implies large real-time database, the medium value *r*=0.5 implies testing database, and the maximum value *r*=0.9 implies unexpectedly small database. However, we cannot draw which measures are the best in general yet. So, we here test all measures with all values of *r*: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and then calculate average item-based MAE for each measure in order to determine best measures. Table 3.4 shows item-based MAE values of all measures over all values of *r*. Best values are shaded.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *r*=0.1 | *r*=0.2 | *r*=0.3 | *r*=0.4 | *r*=0.5 | *r*=0.6 | *r*=0.7 | *r*=0.8 | *r*=0.9 | Average |
| NHSM | 0.7219 | 0.7251 | 0.7260 | 0.7309 | 0.7376 | 0.7436 | 0.7566 | 0.7843 | 0.8583 | 0.7538 |
| MSDJ | 0.7289 | 0.7320 | 0.7329 | 0.7375 | 0.7429 | 0.7474 | 0.7576 | 0.7782 | 0.8422 | 0.7555 |
| Amer2J | 0.7262 | 0.7293 | 0.7302 | 0.7349 | 0.7411 | 0.7467 | 0.7597 | 0.7876 | 0.8681 | 0.7582 |
| Jaccard | 0.7300 | 0.7330 | 0.7339 | 0.7385 | 0.7440 | 0.7485 | 0.7592 | 0.7815 | 0.8566 | 0.7584 |
| COJ | 0.7317 | 0.7346 | 0.7354 | 0.7398 | 0.7450 | 0.7495 | 0.7600 | 0.7831 | 0.8597 | 0.7599 |
| QTI | 0.7233 | 0.7265 | 0.7278 | 0.7326 | 0.7400 | 0.7473 | 0.7639 | 0.7994 | 0.8852 | 0.7607 |
| Amer | 0.7412 | 0.7438 | 0.7446 | 0.7489 | 0.7537 | 0.7569 | 0.7648 | 0.7792 | 0.8174 | 0.7612 |
| QTIJ | 0.7160 | 0.7200 | 0.7224 | 0.7284 | 0.7380 | 0.7486 | 0.7701 | 0.8121 | 0.8976 | 0.7615 |
| Cosine | 0.7427 | 0.7449 | 0.7456 | 0.7497 | 0.7541 | 0.7573 | 0.7653 | 0.7818 | 0.8177 | 0.7621 |
| MSD | 0.7420 | 0.7443 | 0.7450 | 0.7491 | 0.7535 | 0.7566 | 0.7646 | 0.7811 | 0.8388 | 0.7639 |
| Amer2 | 0.7376 | 0.7403 | 0.7408 | 0.7451 | 0.7500 | 0.7541 | 0.7642 | 0.7871 | 0.8702 | 0.7655 |
| SMTP | 0.7465 | 0.7487 | 0.7488 | 0.7517 | 0.7562 | 0.7589 | 0.7667 | 0.7833 | 0.8425 | 0.7670 |
| WPC | 0.7348 | 0.7393 | 0.7438 | 0.7514 | 0.7613 | 0.7744 | 0.7933 | 0.8245 | 0.9088 | 0.7813 |
| SPC | 0.7434 | 0.7511 | 0.7569 | 0.7649 | 0.7743 | 0.7862 | 0.8022 | 0.8287 | 0.8341 | 0.7824 |
| CON | 0.7468 | 0.7505 | 0.7539 | 0.7604 | 0.7681 | 0.7772 | 0.7905 | 0.8188 | 0.8813 | 0.7831 |
| Pearson | 0.7447 | 0.7531 | 0.7591 | 0.7676 | 0.7774 | 0.7897 | 0.8057 | 0.8314 | 0.8326 | 0.7846 |
| BCF | 0.7822 | 0.7832 | 0.7845 | 0.7884 | 0.7934 | 0.7972 | 0.8034 | 0.8193 | 0.8568 | 0.8009 |
| COD | 0.7894 | 0.7937 | 0.7980 | 0.8031 | 0.8099 | 0.8183 | 0.8290 | 0.8494 | 0.9098 | 0.8223 |

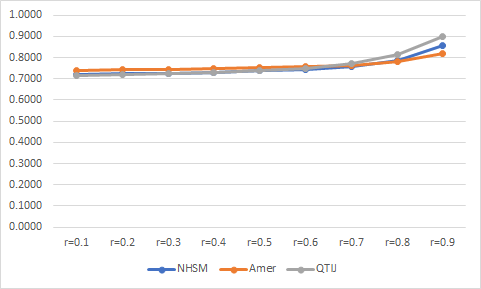
**Table 3.4.** Item-based MAE of all measures over all values of *r*.

The last column in table 3.4 is average MAE values of measures. Hence, measures are sorted according to descending order of their average MAE values. Figure 3.5 shows comparable chart of all measures about their average MAE values. Note, the lower the column in figure 3.5 is, the better the measure is.



**Figure 3.5.** Comparison of all measures regarding average MAE

From table 3.4 and figure 3.5, general top-5 measures are NHSM, MSDJ, Amer2J, Jaccard, and COJ whose average item-based MAE values are 0.7538, 0.7555, 0.7582, 0.7584, and 0.7599, respectively in which NHSM is the best measure. Although Amer2J measure is in top-5 list shown in table 3.4, Amer family has no best measure. Recall that QTIJ measure is preeminent with *r*=0.1, *r*=0.5 and Amer measure is preeminent with *r*=0.9. The reason is that Amer and QTIJ of Amer family are two poles of evaluation. Please see table 3.4 for item-based MAE values of NHSM, Amer, and QTIJ. Figure 3.6 shows comparison of Amer, QTIJ, and NHSM.



**Figure 3.6.** Comparison of Amer, QTIJ, and NHSM regarding MAE

QTIJ is the best from *r*=0.1 to *r*=0.5, NHSM is the best from *r*=0.6 to *r*=0.7, and Amer is the best from *r*=0.8 to *r*=0.9. NHSM is in the middle of QTIJ and Amer and so NHMS is the best overall.

**4. Conclusions**

There is no doubt that Amer family is preeminent when its measures are in top-5 lists. However, it has no best measure overall. QTIJ is the best from *r*=0.1 to *r*=0.5 and Amer is the best from *r*=0.8 to *r*=0.9. When dataset has fewer missing values (*r* < 0.5), QTIJ is better than Amer and otherwise. The reason is that Amer focuses on whether items are rated, and it does not concern numerical rating values. Hence, Amer measure is not affected negatively by missing values with *r* > 0.5. Conversely, QTIJ is harmed by missing values because it takes full advantages of both resemblance (TF) and uniqueness (IDF) of numeric rating vectors. Combination of QTIJ and Amer is a good idea. In future trend, we try our best to combine QTIJ and Amer or research another way to improve QTIJ for resisting missing values. Anyway, QTIJ is most suitable to real applications because it is the best from *r*=0.1 to *r*=0.5 where real rating database is large enough. Hence, Amer family is significant in real applications.

**References**

Do, M.-P. T., Nguyen, D. V., & Nguyen, L. (2010). Model-based Approach for Collaborative Filtering. *Proceedings of The 6th International Conference on Information Technology for Education (IT@EDU2010)* (pp. 217-225). Ho Chi Minh: Ho Chi Minh University of Information Technology. Retrieved from https://goo.gl/BHu7ge

GroupLens. (1998, April 22). *MovieLens datasets*. (GroupLens Research Project, University of Minnesota, USA) Retrieved August 3, 2012, from GroupLens Research website: http://grouplens.org/datasets/movielens

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems (TOIS), 22*(1), 5-53. doi:10.1145/963770.963772

Lin, Y.-S., Jiang, J.-Y., & Lee, S.-J. (2013, January 25). A Similarity Measure for Text Classification and Clustering. (J. Pei, Ed.) *IEEE Transactions on Knowledge and Data Engineering, 26*(7), 1575-1590. doi:10.1109/TKDE.2013.19

Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2013, November 20). A new user similarity model to improve the accuracy of collaborative filtering. (H. Fujita, & J. Lu, Eds.) *Knowledge-Based Systems, 56*(2014), 156-166. doi:10.1016/j.knosys.2013.11.006

Patra, B. K., Launonen, R., Ollikainen, V., & Nandi, S. (2015, March 1). A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. (H. Fujita, & J. Lu, Eds.) *Knowledge-Based Systems, 82*, 163-177. doi:10.1016/j.knosys.2015.03.001

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). Hong Kong: ACM. doi:10.1145/371920.372071

Torres Júnior, R. (2004). *Combining Collaborative and Content-based Filtering to Recommend Research Paper.* Universidade Federal do Rio Grande do Sul, Programa de Pós Graduação em Educação. Porto Alegre: Universidade Federal do Rio Grande do Sul. Retrieved from http://www.lume.ufrgs.br/bitstream/handle/10183/5887/000432990.pdf;sequence=1