**List of similarity measures in collaborative filtering**

Loc Nguyen

Loc Nguyen’s Academic Network, Vietnam

Ali Amer

TAIZ University, Yemen

# Abstract

Recommendation system (RS) is the system which recommends items to users among many existing items/products in database. Two main approaches for RS are content-based filtering (CBF) and collaborative filtering (CF). Recommendation system is a system which recommends items to users among many existing items in database. A popular algorithm in CF is nearest neighbors (NN) algorithm which is to find out nearest neighbors of an active user and then to recommend active user items that these neighbors may like. The essence of NN algorithm is to calculate similarity measures like cosine and Pearson for determining nearest neighbors of an active rating vector. This report focuses on descripting similarity measures which are implemented in SIM framework which is available at http://sim.locnguyen.net. Note that SIM framework provides recommendation algorithms and other machine learning algorithms along with tools to evaluate and deploy them.

|  |  |
| --- | --- |
|  | (99.99) |

# 0. Introduction

Before describing popular similarity measures, we need to concern some important conventions and notations. 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 0.1 is an example of user-based rating matrix in which missing values are denoted by question masks (Do, Nguyen, & Nguyen, 2010, p. 218) and ratings values range from 1 to 5. In Table 0.1, 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 0.1.** 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 0.2 is the item-based rating matrix which is transposed from the user-based rating matrix shown in table 0.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 0.2.** Item-based rating matrix

In table 0.2, 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 0.1, 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. The NN algorithm which acts on user-based rating matrix is called user-based NN algorithm and the NN algorithm which acts on item-based rating matrix is called item-based NN algorithm. Although ideology of user-based NN algorithm and item-based NN algorithm is the same, their implementations are slightly different. User-based NN algorithm is mentioned by default. In general, nearest neighbors (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 items whose predictive values are high enough are recommended to active user.

The essence of NN algorithm is to use similarity measures in order to find out nearest neighbors of an active rating vector with note that similarity measures are calculated in step 1. 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 whose indices belong to are rated by both user 1 and user 2. In other words, all items whose indices belong to co-exist in vectors *u*1 and *u*2. All items whose indices belong to are rated by user 1 or user 2. Similarly, let *Jk* be the set of users who rated the given item *k*. Notation |x| indicates absolute value of number, length of vector, length of geometric segment, or cardinality of set, which depends on context. Pair of braces “{…}” denotes a set. Similarly, Let *U*1 and *U*2 be set of indices of user from which item 1 (*v*1) and item 2 (*v*2) are received ratings, respectively. Of course, we have intersection set and union set .

Let sim(*u*1, *u*2) denote the similarity of *u*1 and *u*2, which is the general notation for all similarity measures, for example, sim(*u*1, *u*2) = cos(*u*1, *u*2) denotes cosine measure. Before describing similarity measures, we should consider how to compute predictive values for missing values in step 2 of NN algorithm. Given the similarity of *u*1 and *u*2 denoted sim(*u*1, *u*2), the larger sim(*u*1, *u*2) is, the more the user 2 is near to active user 1. Hence, sim(*u*1, *u*2) 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 (predicted) based on ratings of nearest neighbors and similarities according to step 2 of NN algorithm (Torres Júnior, 2004, p. 18).

|  |  |
| --- | --- |
|  | (0.1) |

Where and are mean values of *u*1 and *ui*, respectively. Equation 0.1 above is called prediction formula or estimation formula.

Where *Ii* is the set of indices of items that user *i* rated. The missing value *r*1*j* of *u*1 can be predicted more simply as follows:

|  |  |
| --- | --- |
|  | (0.2) |

In general, similarity measure is the heart of NN algorithm because prediction formulas are based on similarity measures. Next sections mention similarity measures built in SIM framework.

It is convenient for us to systemize and describe these measures that they are classified into indexed groups whose names are names of representative measures. In other words, measures in the same group are often variants of a representative measure or mutually relative in their formulation. For instance, the first section of this report is “*1.Jaccard*” which is the group named “Jaccard” indexed 1, including measures related to the representative Jaccard measure. They are often variants of Jaccard measures, such as Dice, JaccardMulti, Improved Jaccard (IJ), JacLMH, RatingJaccard, relevant Jaccard, JacRA, PNCR, and JacEDS. Each measure in a group is specified by a equation as its formulation which is indexed by two numbers *major* and *minor* in form of *major*.*minor* where *major* is the group index and *minor* is the index of such measure in such group. For example, Dice in group “Jaccard” is specified by equation 1.2 in which the major number 1 is the index of the group Jaccard and the minor number 2 is the index of Dice in its group Jaccard. However, a measure is usually specified by its group index and its name. For instance, IJ measure is often called “*1.IJ*”, which is the most convenient way to keep track of describing and testing measures. Therefore, please pay attention that groups are conventionally specified by their indices listed as sections in this report.

Because this report only lists basic, main, or important measures whereas there are a huge number of variants and combinations which derived from measure groups, *paired label* and *compound name* are proposed to organize measures in more structural way, especially in testing plan. For instance, a paired label *XY* is a pair of two letters *X* and *Y* where *X* taken from {*A*, *B*, *C*,…} and *Y* taken from {*I*, *U*}. For *X* letter, set *A* includes basic measures and set *B* includes variants of basic measures in set *A*. There will be more new similarity measures and their sets are named *C*, *D*, *E* and *F*, etc. For *Y* letter, *I* indicates that measure is tested with item-based NN algorithm whereas *U* indicates that measure is tested with user-based NN algorithm. Thus, paired label *XY* represents sub-groups. When a new measure is derived from two or more measures by some combinations or associations, its name is created by string concatenation of super measures like *measure1.measure2*. For example, *AU01.Jaccard* implies Jaccard measure in Jaccard group indexed 1 tested with user-based NN algorithm in set *A* (basic set) and *BI01.Dice* implies Dice measure in Jaccard group indexed 1 tested with item-based NN algorithm in set *B* whereas *BU07.NHMS.Amer* is a compound measure which combines NHMS measure and Amer measure in PSS group indexed 7 tested with user-based NN algorithm in set *B*. In general, the association of measure name, group index, and paired label make formal names of measures as ***XYindex.measure*1*….measuren*** for both descriptions and testing plans. Within the formal name convention, measure equation is named in form of ***X*(*Y*)*major*.*minor*** where *major* is the group index and *minor* is the index of such measure in which *X*=*A* is ignored and *Y* is often ignored. Note that some main measures are named in set *B* but they can indexed in set *A* as default, for example BU01.IJ is described in section “1.Jaccard” in set *A*, which is explained that it is required to test basic measures in high priority in testing plan. Therefore, the two most important parts in the name convention are group index (major number) and measure name and besides, an equation name always points to a measure exactly. The following table lists some formal names of measures.

|  |  |  |
| --- | --- | --- |
| Measure | Group | Equation |
| AU01.Jaccard | 1.Jaccard | (1.1) |
| AU02.Cosine | 2.Cosine | (2.1) |
| AU03.Pearson | 3.Pearson | (3.1) |
| AU04.MSD | 4.MSD | (4.1) |
| AU05.SRC | 5.SRC | (5.1) |
| AU06.PIP | 6.PIP | (6.1) |
| AU07.PSS | 7.PSS | (7.1) |
| AU08.BCF | 8.BCF | (8.1) |
| AU09.MMD | 9.MMD | (9.1) |
| AU10.Triangle | 10.Triangle | (10.1) |
| AU11.Feng | 11.Feng | (11.1) |
| AU12.Mu | 12.Mu | (12.1) |
| AU14.SMD | 14.SMD | (14.1) |
| AU15.NNSM | 15.NNSM | (15.1) |
| AU16.TA | 16.TA | (16.1) |
| AU17.RA | 17.RA | (17.1) |
| AU18.Entropy | 18.Entropy | (18.1) |
| AU19.KL | 19.KL | (19.1) |
| AU26.URP | 26.URP | (26.1) |
| AU27.STB | 27.STB | (27.1) |
| AU30.Singularity | 30.Singularity | (30.1) |
| BU01.Dice | 1.Jaccard | (1.2) |
| BU01.IJ | 1.Jaccard | (1.4) |
| BU01.JacDual | 1.Jaccard | (1.14) |
| BU01.JacEDS | 1.Jaccard | (1.13) |
| BU01.JacLMH | 1.Jaccard | (1.6) |
| BU01.JacRA | 1.Jaccard | (1.11) |
| BU01.JaccardMulti | 1.JaccardMulti | (1.3) |
| BU01.RatingJaccard | 1.Jaccard | (1.8) |
| BU01.RelevantJaccard | 1.Jaccard | (1.10) |
| BU02.CosineJ | 2.Cosine | (2.5) |
| BU03.CPC | 3.Pearson | (3.2) |
| BU03.IPC | 3.Pearson | (3.5) |
| BU03.IPWR | 3.Pearson | (3.6) |
| BU03.PC | 3.Pearson | (3.8) |
| BU03.PearsonJ | 3.Pearson | (3.7) |
| BU03.SPC | 3.Pearson | (3.4) |
| BU03.WPC | 3.Pearson | (3.3) |
| BU04.MSDJ | 4.MSD | (4.3) |
| BU06.MPIP | 6.PIP | (6.8) |
| BU07.NHMS | 7.PSS | (7.4) |
| BU07.NHMS.Amer | 7.PSS | (B7.2) |
| BU07.NHMS.SMD | 7.PSS | (B7.1) |
| BU07.PSSJ | 7.PSS | (7.3) |
| BU08.BCFJ | 8.BCF | (8.3) |
| BU09.CjacMD | 9.MMD | (9.2) |
| BU10.TMJ | 10.Triangle | (10.2) |
| BU14.Amer | 14.SMD | (14.4) |
| BU14.AmerJ | 14.SMD | (B14.2) |
| BU14.HSMD | 14.SMD | (14.2) |
| BU14.HSMDJ | 14.SMD | (14.3) |
| BU14.QTI | 14.SMD | (14.9) |
| BU14.QTIJ | 14.SMD | (14.10) |
| BU14.SMDJ | 14.SMD | (B14.1) |
| BU15.NNSMJ | 15.NNSM | (15.2) |
| BU16.TAJ | 16.TA | (16.2) |
| BU16.TAN | 16.TA | (16.3) |
| BU16.TANJ | 16.TA | (16.4) |
| CU01.IJ.Amer | 1.Jaccard | (C1.4) |
| CU01.IJ.Cosine | 1.Jaccard | (C1.1) |
| CU01.IJ.PSS | 1.Jaccard | (C1.3) |
| CU01.IJ.Pearson | 1.Jaccard | (C1.2) |
| CU01.IJ.TA | 1.Jaccard | (C1.5) |
| CU14.Amer.CPC | 14.SMD | (C14.3) |
| CU14.Amer.Cosine | 14.SMD | (C14.1) |
| CU14.Amer.NNSM | 14.SMD | (C14.6) |
| CU14.Amer.PSS | 14.SMD | (C14.4) |
| CU14.Amer.Pearson | 14.SMD | (C14.2) |
| CU14.Amer.QTI | 14.SMD | (C14.5) |
| CU14.Amer.TA | 14.SMD | (C14.7) |

**Table 0.3.** Sample conventional measures

Note, equations *B*\* and *C*\* are listed in appendices *B* and *C*.

# 1. Jaccard

The first measure which is described here is Jaccard because of its special feature when it does not concern magnitude of numeric rating values. Jaccard 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):

|  |  |
| --- | --- |
|  | (1.1) |

A simple version of Jaccard is Dice measure as follows:

|  |  |
| --- | --- |
|  | (1.2) |

Although Jaccard is simple, it is effective within spare rating matrix which has many missing values. Another version of Jaccard is (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158):

|  |  |
| --- | --- |
|  | (1.3) |

Liang, Ma, and Yuan (Liang, Ma, & Yuan, 2015) proposed an improved variant of Jaccard based on the concept of singularity. 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, hence, rating values which are greater or equal than *rm* are called relevant ratings and the others which are smaller than *rm* are called non-relevant. Liang et al. called “relevant” and “non-relevant” as “positive” and “negative”, respectively. Given user 1 and user 2, for every item *j*, let:

* If both *r*1*j* and *r*2*j* are positive, item *j* belongs to the predefined set *PA*.
* If both *r*1*j* and *r*2*j* are negative, item *j* belongs to the predefined set *NA*.
* If *r*1*j* and *r*2*j* are different which means that one is positive and the other is negative, item *j* belongs to the predefined set *D*.
* When one user rated item *j* but the other did not rate, if such rating is positive then item *j* belongs to the predefined set *PO*, otherwise item *j* belongs to the predefined set *NO*.

Liang et al. (Liang, Ma, & Yuan, 2015) defined singularity quantities such as positive singularity, negative singularity, and empty singularity denoted *SjP*, *SjN*, and *SjE* as follows:

Where *Pj* is the set of users who gave positive ratings to item *j*, *Nj* is the set of users who gave negative ratings to item *j*, and *Ej* is the set of users who did not rate item *j*. Liang et al. defined Improved Jaccard (IJ) as follows (Liang, Ma, & Yuan, 2015, p. 1634):

|  |  |
| --- | --- |
|  | (1.4) |

As a result, when Liang et al. concerned singularity quantities, they concerned specific features of users so that the recommendation process is more appropriate. For example, given two rating vectors *u*3 = (*r*31=4, *r*32=1, *r*33=5, *r*34=5), and *u*4 = (*r*41=1, *r*42=2, *r*43=?, *r*44=?) from table 0.1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 = ? |

we have *PA* = {}, *NA* = {1, 2}, *D* = {}, *PO* = {3, 4}, *NO* = {}, *P*1=1, *N*1=3, *E*1=0, *P*2=0, *N*2=4, *E*2=0, *P*3=1, *N*3=2, *E*3=1, *P*4=3, *N*4=0, and *E*4=1 when *rm*=3. We also obtain *S*1*P* = 1 – 1/4 = 3/4, *S*1*N* = 1 – 3/4 = 1/4, *S*1*E* = 1 – 0/4 = 1, *S*2*P* = 1 – 0/4 = 1, *S*2*N* = 1 – 4/4 = 0, *S*2*E* = 1 – 0/4 = 1, *S*3*P* = 1 – 1/4 = 3/4, *S*3*N* = 1 – 2/4 = 1/2, *S*3*E* = 1 – 1/4 = 3/4, *S*4*P* = 1 – 3/4 = 1/4, *S*4*N* = 1 – 0/4 = 1, and *S*4*E* = 1 – 1/4 = 3/4. As a result, the IJ measure of *u*3 and *u*4 is:

Lee (Lee, 2017) divided rating range into three sub-intervals, for each sub-interval a sub-Jaccard measure is computed. Finally, the whole Jaccard measure of Lee called JacLMH is the average of such three sub-Jaccard measures. Let *rL* and *rH* be minimum value and maximum value of rating values, for example, if rating values range from 1 to 5 then, *rL* = 1 and *rH* = 5. Let *Lbd* and *Hbd* be boundaries of the three sub-intervals such that *rL* < *Lbd* < *Hbd* < *rH*, the set of items rated by user *i* denoted *Ii* is divided into three sub-sets *ILi*, *IMi*, and *IHi* as follows (Lee, 2017, p. 803):

The three sub-Jaccard measures with regard to such three sub-sets are defined as follows (Lee, 2017, p. 803):

|  |  |
| --- | --- |
|  | (1.5) |

The final JacLMH is average of such three sub-Jaccard measures (Lee, 2017, p. 803):

|  |  |
| --- | --- |
|  | (1.6) |

Traditional Jaccard does not concern numerical rating values and so, Ayub et al. (Ayub, Ghazanfar, Khan, & Saleem, 2020) improved Jaccard measure by numerical equality. Let *NT*(*u*1, *u*2) be the count of ratings that are equal in numerical values, as follows (Ayub, Ghazanfar, Khan, & Saleem, 2020, p. 10003):

|  |  |
| --- | --- |
|  | (1.7) |

Their proposed measure called RatingJaccard is defined as follows (Ayub, Ghazanfar, Khan, & Saleem, 2020, p. 10003):

|  |  |
| --- | --- |
|  | (1.8) |

Ayub et al. also improved the quantity *NT*(*u*1, *u*2) by associating it with a pre-defined threshold *TH* as follows (Ayub, Ghazanfar, Khan, & Saleem, 2020, p. 10003):

|  |  |
| --- | --- |
|  | (1.9) |

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

Bag et al. (Bag, Kumar, & Tiwari, 2019) improved Jaccard by concerning more un-co-rated items which are items rated only one user among two concerned users. According to their viewpoint, the traditional Jaccard is re-formulated as follows (Bag, Kumar, & Tiwari, 2019, p. 57):

They defined and as the numbers of items which are not rated by only user 1 and only user 2, respectively (Bag, Kumar, & Tiwari, 2019, p. 57).

The traditional Jaccard is re-written as follows (Bag, Kumar, & Tiwari, 2019, p. 58):

Therefore, Jaccard measure is proportional to two quantities and . Bag et al. (Bag, Kumar, & Tiwari, 2019) improved such proportion by specify that their Jaccard will be proportional to three quantities , , and . As a result, Bag et all defined the so-called relevant Jaccard as follows (Bag, Kumar, & Tiwari, 2019, p. 59):

|  |  |
| --- | --- |
|  | (1.10) |

Wu, Huang, and Wang (Jindal, Sharma, & Verma, 2022, p. 4) also improved Jaccard measure by concerning numerical values. Hence, they proposed the so-called JacRA as follows:

|  |  |
| --- | --- |
|  | (1.11) |

Percentage of Non-Common Ratings (PNCR) is defined as follows (Gazdar & Hidri, 2019, p. 19):

|  |  |
| --- | --- |
|  | (1.12) |

Note, |***V***| is the number of all items.

Amer defined JacEDS as follows:

|  |  |
| --- | --- |
|  | (1.13) |

Where and are the sets of items that user 1 and user 2 do not rate, respectively. The reasonable proof of JacEDS is that JacEDS proves traditional Jaccard by concerning both items rated by users and items not rated by users whereas Jaccard only focuses on rated items. For example, given two rating vectors *u*3 = (*r*31=4, *r*32=1, *r*33=5, *r*34=5), and *u*4 = (*r*41=1, *r*42=2, *r*43=?, *r*44=?) from table 0.1, we have *I*3 = {1, 2, 3, 4}, *I*4 = {1, 2}, = {}, = {3, 4}, and |***V***| = 4. Therefore, JacEDS of *u*3 and *u*4 is:

Where and are the sets of items that user 1 and user 2 do not rate, respectively. Nguyen proposed dual Jaccard that balances the commonality among rated items and the commonality among non-rated items. Let *P*1 and *P*2 be the sets of items on which user 1 and user 2 gave relevant (positive) ratings, dual Jaccard is defined as follows:

|  |  |
| --- | --- |
|  | (1.14) |

Where *k* is called concerning coefficient or reinforcement factor which is now defined as a half of Jaccard measure.

Note, *N* is the total number of items that both users rated. In equation of JacDual measure, the expression is exactly traditional Jaccard and the expression implies the similarity in rating between two users with regards to implicit inversion. Indeed, the term indicates the number of common rated items in implicit inversion whereas the term indicates cardinality of the union of rated items in implicit inversion. Therefore, the concept “duality” in JacDual measure implies the duality of explicit Jaccard and implicit Jaccard. For example, given two rating vectors *u*3 = (*r*31=4, *r*32=1, *r*33=5, *r*34=5), and *u*4 = (*r*41=1, *r*42=2, *r*43=?, *r*44=?) from table 0.1, we have *I*3 = {1, 2, 3, 4}, *I*4 = {1, 2}, = {}, = {3, 4}, *P*3 = {1, 3}, *P*4 = {}, and *N* = 4. Therefore, dual Jaccard of *u*3 and *u*4 is:

Al-Shamri (Al‑Shamri, 2021) applied hyperbolic tangent function into improving Jaccard measure so as to propose a measure called JacHT. However, before mentioning JacHT, it is necessary to describe a so-called active-driven Jaccard where Jaccard is focused on active user in recommendation process who receives recommended list of items. Particularly, suppose user 1 named *u*1 is active user, the active-driven Jaccard called Jacative is defined as follows (Al‑Shamri, 2021, p. 4):

Al-Shamri applied hyperbolic tangent function into Jacative to define JacHT as follows:

|  |  |
| --- | --- |
|  | (1.15) |

Where *h*(.) is hyperbolic tangent function:

# 2. Cosine

Cosine which is the most popular similarity measure is specified as follows:

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

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. By following the ideology of Jaccard measure, cosine measure is modified as follows:

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

Traditional cosine is only determined by the common set of items which are rated by both users but COJ is more general by concerning items which are rated by at least one user. Amer proposed a so-called numerical nearby similarity measure (NNSM) which is also more general than cosine but it saves computation cost better than COJ. NNSM developed by Amer and implemented by Nguyen is defined as follows:

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

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:

Obviously, CON measure is constrained Pearson correlation (CPC) mentioned later.

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

Adjusted cosine measure (COD) is defined as follows:

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

Jaccard can be combined with any measure. For instance, CosineJ is combinations of Jaccard and cosine as follows:

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

# 3. Pearson

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

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

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. Pearson measure 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):

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

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 is defined as follows (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158):

|  |  |
| --- | --- |
|  | (3.3) |

SPC is defined as follows (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 158):

|  |  |
| --- | --- |
|  | (3.4) |

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

In the determinator of traditional Pearson measures, both vectors *u*1 and *u*2 are taken account of common items *I*1∩*I*2. Alternately, Ayub et al. (Ayub M. , et al., 2019) calculated module of each vector on its full rated item regardless of the other’s rated items. Their measure is called improved Pearson correlation (IPC) is defined as follows (Ayub M. , et al., 2019, p. 7):

|  |  |
| --- | --- |
|  | (3.5) |

Ayub et al. (Ayub M. , et al., 2019) also proposed a new improvement of Pearson measure by combining IPC and rating preference behavior (RPB) measure. Thus, their proposed measure is called improved Pearson correlation with rating preference behavior (IPWR), which is defined as follows (Ayub M. , et al., 2019, p. 8):

|  |  |
| --- | --- |
|  | (3.6) |

Where *α* and *β* are two weights which are applied to RPB and IPC. RPB measure is trigonometric cosine function of statistical deviations, specified as follows (Ayub M. , et al., 2019, p. 6):

Note, and var(*ui*) are mean and variance of *ui*, respectively.

In practice, the variance of *ui* can be replaced by the standard deviation of *ui* as follows (Ayub M. , et al., 2019, p. 6):

PearsonJ is combinations of Jaccard and Pearson as follows:

|  |  |
| --- | --- |
|  | (3.7) |

Choi and Suh (Choi & Suh, 2013) proposed a so-called PC measure which is Pearson measure weighted by similarities of items. In other words, PC measure combines similarities of users and items (Patra, Launonen, Ollikainen, & Nandi, 2015, p. 4). The ideology is excellent. PC measure can be applied into any foundation measures. Each factor in PC measure is weighted by a similarity of active item and another item. Suppose it is necessary to estimate rating values of active item *k*, PC measure (Choi & Suh, 2013, p. 148) is defined as follows:

|  |  |
| --- | --- |
|  | (3.8) |

Where sim(*vk*, *vj*) is similarity of the active item *k* and item *j*. Note, sim(*vk*, *vj*) can be calculated by any measures here. The and are mean values of *u*1 and *u*2, respectively.

Experimental results proved that PC is an effective measure.

# 4. MSD

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):

|  |  |
| --- | --- |
|  | (4.1) |

Another variant of MSD is specified by some authors as follows:

|  |  |
| --- | --- |
|  | (4.2) |

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

|  |  |
| --- | --- |
|  | (4.3) |

# 5. SRC

When rating values are converted into ranks, Spearman’s Rank Correlation (SRC) is defined as follows (Hyung, 2008, p. 39):

|  |  |
| --- | --- |
|  | (5.1) |

Where *dj* is difference between two ranks on item *j* given by user 1 and user 2.

Note, it is easy to convert ratings values to ranks. For example, suppose rating values (bins) are 5, 6, 7, 8, 9 then, we have rank 1 (for value 9), rank 2 (for value 8) , rank 3 (for value 7), rank 4 (for value 6), and rank 5 (for value 5). If user 1 rates value 9 to item *j*, we have *rank*1*j* = 1. The larger the value is, the smaller (higher) the rank is.

# 6. PIP

Ahn (Hyung, 2008) proposed a heuristic measure to solve cold-starting problem which relates to missing data in which there is not enough information to calculate similarities between rating vectors (Hyung, 2008, p. 39). The measure called PIP measure based on concept of “agreement” in rating. If both user 1 and user 2 like or dislike the same item, it is called that they have a rating “agreement” on such item. Let *r*1*j* and *r*2*j* be ratings of user 1 and user 2 on item *j*, respectively, the agreement (Hyung, 2008, p. 43) of them is defined 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. PIP measure (Hyung, 2008, p. 42) is sum of products of triples Proximity, Impact, and Popularity.

|  |  |
| --- | --- |
|  | (6.1) |

Proximity (Hyung, 2008, p. 43) indicates similarity of two ratings, based on agreement and distance between them. The distance is increased twice as a penalty if such two ratings are not agreed.

|  |  |
| --- | --- |
|  | (6.2) |

Where *rmin* and *rmax* are minimum rating value and maximum rating value, respectively. If two ratings are agreed, their impact (Hyung, 2008, p. 43) is proportional to difference between them and rating median. If two ratings are disagreed, their impact is inverse of such difference.

|  |  |
| --- | --- |
|  | (6.3) |

Popularity (Hyung, 2008, p. 43) indicates difference between ratings given by active users and the average rating.

|  |  |
| --- | --- |
|  | (6.4) |

Note, *μj* is average rating of item *j*, which is same mean of rating values of item *j*. Experimental results proved that cold-starting problem is solved well by PIP measure (Hyung, 2008, p. 47).

Manochandar and Punniyamoorthy (Manochandar & Punniyamoorthy, 2020) proposed the modified PIP (MPIP) measure which is still the sum of products of triples Proximity, Impact, and Popularity but they improved such triples. For instance, the Proximity quantity is improved as follows (Manochandar & Punniyamoorthy, 2020, p. 595):

|  |  |
| --- | --- |
|  | (6.5) |

Note, *rm* is the median; for example, if rating values range from 1 to 5, the median is *rm* = (1+5) / 2 = 3. By similar inference, *rm*+ is the specified median that is larger than *rm*; for example, if rating values range from 1 to 5 then, *rm*+ = (4+5) / 2 = 4.5. Thus, *rm*– is the specified median that is smaller than *rm*; for example, if rating values range from 1 to 5 then, *rm*– = (1+2) / 2 = 1.5. The parameter *δ* is rating scale as follows (Manochandar & Punniyamoorthy, 2020, p. 595):

Manochandar and Punniyamoorthy modified the Impact quantity as follows (Manochandar & Punniyamoorthy, 2020, p. 596):

|  |  |
| --- | --- |
|  | (6.6) |

They also modified the Popularity quantity as follows (Manochandar & Punniyamoorthy, 2020, p. 596):

|  |  |
| --- | --- |
|  | (6.7) |

Note, *μj* is average rating of item *j*. As a result, MPIP is specified as follows:

|  |  |
| --- | --- |
|  | (6.8) |

# 7. PSS and NHMS

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):

|  |  |
| --- | --- |
|  | (7.1) |

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:

|  |  |
| --- | --- |
|  | (7.2) |

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.

PSS associated with Jaccard produces a so-called PSSJ measure as follows:

|  |  |
| --- | --- |
|  | (7.3) |

Where Jaccard is specified by equation 1.1, respectively. Liu et al. (Liu, Hu, Mian, Tian, & Zhu, 2013, p. 161) proposed a new heuristic similarity model (NHSM) as triple product of PSS measure, URP measure, and Jaccard2 measure.

|  |  |
| --- | --- |
|  | (7.4) |

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

# 8. BCF

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 coefficient for items is calculated as follows (Patra, Launonen, Ollikainen, & Nandi, 2015, p. 5):

|  |  |
| --- | --- |
|  | (8.1) |

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, item BC coefficient concerns two items. In table 0.2, rating vectors of item 3 and item 4 are *v*3 = (1, 2, 5, ?) and *v*4 = (5, 4, 5, ?), respectively with note that rating values range from 1 to 5 and so we have:

According to Patra et al. (Patra, Launonen, Ollikainen, & Nandi, 2015, p. 5), user BC similarity is sum of products of item BC coefficients and local similarities as follows:

|  |  |
| --- | --- |
|  | (8.2) |

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 similarity in CF (BCF) as sum of user BC similarity and Jaccard measure as follows:

|  |  |
| --- | --- |
|  | (8.3) |

Singh et al. (Singh, Sinha, & Choudhury, 2022) improved BCF measure by applying BCF measure into both user vector and item vector. Let BC(*v*1, *v*2) be BC similarity of item *v*1 and item *v*2 as follows:

Note, bc(*k*, *l*) is user BC coefficient for users:

Where #*k* and #*l* are the numbers of items which are rated by users *k* and *l*, respectively whereas #*hk* and #*hl* are numbers of items which are rated with rating value *h* by users *k* and *l*, respectively. The local similarity loc(*rk*1, *rl*2) with regard to items is specified as follows:

Singh et al. (Singh, Sinha, & Choudhury, 2022) proposed BCF2 measure by browsing BC similarity over both users and items, as follows:

|  |  |
| --- | --- |
|  | (8.4) |

Where local similarity is improved by mutual associating with BC similarity as follows:

|  |  |
| --- | --- |
|  | (8.5) |

Obviously, Singh et al. (Singh, Sinha, & Choudhury, 2022) improved BCF measure by operating BCF measure by itself. Singh et al. (Singh, Sinha, & Choudhury, 2022) also added Jaccard measure into their measure BCF2 for improving BCF2 more, as BCF2J measure as follows:

|  |  |
| --- | --- |
|  | (8.6) |

# 9. MMD

Suryakant and Mahara (Suryakant & Mahara, 2016) proposed a so-called Cosine-Jaccard-Mean Measure of Divergence (CjacMD) based on Mean Measure of Divergence (MMD) to solve the problem of sparse rating matrix. Because MMD measure takes advantages of statistical aspects, it can alleviate sparsity. MMD focuses on personal habits which are ignored by nonstatistical measures (Suryakant & Mahara, 2016, p. 453). Recall that 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 *X* = (*x*1, *x*2,…, *xb*) and *Y* = (*y*1, *y*2,…, *yb*) be count vectors of user 1 and user 2, respectively where *xj* (*yj*) is the number of items to which user 1 (user 2) gives bin *j* with note that *b* is the number of bins. For example, rating vectors of user 1 and user 2 in table 0.1 are *u*1 = (1, 2, 1, 5) and *u*2 = (2, 1, 2, 4), respectively with note that rating values ranges from 1 to 5. We have:

MMD measure is defined as follows (Suryakant & Mahara, 2016, p. 453), (Harris & Sjøvold, 2018, p. 87):

|  |  |
| --- | --- |
|  | (9.1) |

Where *θ*1\* and *θ*2\* are Grewal’s transformations (Harris & Sjøvold, 2018, p. 85) of *X* and *Y*, respectively.

In fact, CjacMD (Suryakant & Mahara, 2016, p. 453) combines three other measures such as cosine, Jaccard, and MMD together.

|  |  |
| --- | --- |
|  | (9.2) |

Experimental result proved that CjacMD model is effective similarity model.

# 10. Triangle

Sun et al. (Sun, et al., 2017) proposed a so-called Triangle similarity measure which considers both angle and lengths of rating vectors. For instance, given two user vectors *u*1 and *u*2 are considered as two vector OA = *u*1 and OB = *u*2 and hence, OAB forms a triangle. TS measure is ratio of the length |AB| to the sum of lengths |OA| + |OB|. Of course, |AB| is always less than or equal to |OA| + |OB| according to triangle inequality. The idea is excellent. TS measure (Sun, et al., 2017, p. 6) is defined as follows:

|  |  |
| --- | --- |
|  | (10.1) |

Sun et al. also combined Triangle measure and Jaccard measure to form a new measure called Triangle multiplying Jaccard (TMJ) measure. The integrated TMJ (Sun, et al., 2017, p. 6) is defined as follows:

|  |  |
| --- | --- |
|  | (10.2) |

Experimental result proved that TMJ is effective measure.

# 11. Feng

To solve the problem of sparse rating matrix, Feng et al. (Feng, Fengs, Zhang, & Peng, 2018) proposed a new model of similarity which includes three parts such as *S*1, *S*2, and *S*3. The *S*1 (Feng, Fengs, Zhang, & Peng, 2018, p. 6) is normal similarity and they choose cosine as *S*1.

Where *ρ* is sparsity threshold which is proposed by Feng et al. The *S*2 (Feng, Fengs, Zhang, & Peng, 2018, p. 6) punishes user pairs whose co-rated items are few.

The *S*3 (Feng, Fengs, Zhang, & Peng, 2018, p. 6) focuses on statistical feature of user ratings, which reflects essential user favorites. *S*3 is aforementioned URP measure.

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. The similarity model of Feng et al. (Feng, Fengs, Zhang, & Peng, 2018, p. 5) is product of *S*1, *S*2, and *S*3 as follows:

|  |  |
| --- | --- |
|  | (11.1) |

Experimental result proved that Feng model is effective similarity model.

# 12. Mu

Mu et al. (Mu, Xiao, Tang, Luo, & Yin, 2019) combined local measures (Pearson and Jaccard) with global measure to solve the problem of sparse rating matrix. The global measure is Hellinger (Hg) distance which estimates similarity of two probabilistic distributions. In fact, Hg is inverse of BC coefficient in discrete distributions as follows (Mu, Xiao, Tang, Luo, & Yin, 2019, p. 419):

|  |  |
| --- | --- |
|  | (12.1) |

Note, #1 and #2 are the numbers of item which are rated by user 1 and user 2, respectively whereas #*h*1 and #*h*2 are numbers of items which receive rating value *h* from user 1 and user 2, respectively. For example, rating vectors of user 1 and user 2 in table 0.1 are *u*1 = (1, 2, 1, 5) and *u*2 = (2, 1, 2, 4), respectively with note that rating values range from 1 to 5 and so we have:

Given weight *α*, the Mu measure (Mu, Xiao, Tang, Luo, & Yin, 2019, p. 419) combines Pearson, Jaccard, and Hg as follows:

|  |  |
| --- | --- |
|  | (12.2) |

Experimental result proved that Mu measure is effective similarity model.

# 13. SMTP

Similarity Measure for Text Processing (SMTP) was developed by Lin, Jiang, and Lee (Lin, Jiang, & Lee, 2013), originally used for computing the similarity between two documents in text processing. Here documents are considered 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):

|  |  |
| --- | --- |
|  | (13.1) |

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:

|  |  |
| --- | --- |
|  | (13.2) |

# 14. SMD

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*. Amer defined a so-called SMD measure in binary representation as follows:

|  |  |
| --- | --- |
|  | (14.1) |

Now we interpret SMD measure in context of rating matrix. Let *I*1 and *I*2 be set of indices of items that user 1 and user 2 rated, respectively, we have:

Therefore, we obtain:

Let,

Where,

Amer also defined another so-called *HSMD measure* in numerical representation in which values *rij* are kept in numerical values as rating values, as follows:

|  |  |
| --- | --- |
|  | (14.2) |

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 SMD and HSMD 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 SMD measure, we have *N*12=3, *N*=6, *F*=3, *N*1=5, and *N*2=4. Hence, SMD measure is calculated according to equation 14.1, as follows:

According to HSMD measure, we have *R*1 = 5+7 = 12, *R*2 = 5, and *G* = (2 + 5 + 7 + 8 + 9) \* (9 + 6 + 5 + 1) = 651. Hence, HSMD measure is calculated according to equation 14.2, as follows:

When HSMD measure is combined with Jaccard measure, it is called HSMDJ which is specified by equation 14.3.

|  |  |
| --- | --- |
|  | (14.3) |

Another extension of SMD called Amer measure is developed by Amer and Nguyen. Amer is more general than SMD because it concerns rating values with the aspect of relevant values and non-relevant values. 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, hence, rating values which are greater or equal than *rm* are called relevant ratings and the others which are smaller than *rm* are called non-relevant. Let inc1(*u*1, *u*2, *j*) and inc2(*u*1, *u*2, *j*) be two increment functions on user 1, user 2, and item *j* which is defined as follows:

Essentially, Amer is derived from SMD by replacing the intersection cardinality by sum of inc1(*u*1, *u*2, *j*) and inc2(*u*1, *u*2, *j*) over *j*. We define:

The factor ½ occurs in *A* because the sum of inc1(*u*1, *u*2, *j*) and inc2(*u*1, *u*2, *j*) aims to fold double the cooccurrence of two rated items. Therefore, Amer measure is defined as follows:

In other words, Amer is specified more succinctly as follows:

|  |  |
| --- | --- |
|  | (14.4) |

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), according to Amer measure, we have:

We also have inc1(*u*1, *u*2, 1) = 1, inc2(*u*1, *u*2, 1) = 0, inc1(*u*1, *u*2, 2) = 0, inc2(*u*1, *u*2, 2) = 0, inc1(*u*1, *u*2, 3) = 1, inc2(*u*1, *u*2, 3) = 0, inc1(*u*1, *u*2, 4) = 1, inc2(*u*1, *u*2, 4) = 1, inc1(*u*1, *u*2, 5) = 0, inc2(*u*1, *u*2, 5) = 0, inc1(*u*1, *u*2, 6) = 1, inc2(*u*1, *u*2, 6) = 0. Therefore,

We develop a new measure which is based on HSMD measure and ideology of TF and IDF. Firstly, we research deeply HSMD 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 HSMD measure is a *quasi-IDF*. Here we re-define the quasi-IDF as follows:

|  |  |
| --- | --- |
|  | (14.5) |

Note, notation “\” denotes complement operator in set theory. The quasi-IDF measure associated with Jaccard derives quasi-IDFJ measures as follows:

|  |  |
| --- | --- |
|  | (14.6) |

Recall that Jaccard measure is mentioned in previous section. Similarly, following ideology of term frequency (TF), we define a so-called quasi-TF as follows:

|  |  |
| --- | --- |
|  | (14.7) |

Note, notation “” denotes intersection operator in set theory. The quasi-TF measure associated with Jaccard derives quasi-TFJ measures as follows:

|  |  |
| --- | --- |
|  | (14.8) |

The new measure called quasi-TF-IDF (QTI) is product of the quasi-TF and the quasi-IDF as follows:

|  |  |
| --- | --- |
|  | (14.9) |

QTI is associated with Jaccard measure, which derives QTIJ measure as follows:

|  |  |
| --- | --- |
|  | (14.10) |

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:

EMX proposed by Amer is an improvement of Amer measure by combining Amer measure with other measures such as URP, NNSM, Coco, and STB as well as standard deviations are removed from equation of EMX. The integrations of Amer and URP, NNSM are called EMX1 and EMX2 defined as follows:

|  |  |
| --- | --- |
|  | (14.11) |

|  |  |
| --- | --- |
|  | (14.12) |

Where measures URP and NNSM are defined by equations 26.1 and 15.1. The first part in the equations above specifies URP. The integrations of Amer and URP, Coco are called EMX3 and EMX4 defined as follows:

|  |  |
| --- | --- |
|  | (14.13) |

|  |  |
| --- | --- |
|  | (14.14) |

Where Coco is defined by equation 25.1. The integrations of Amer and URP, STB are called EMX5 and EMX6 defined as follows:

|  |  |
| --- | --- |
|  | (14.15) |

|  |  |
| --- | --- |
|  | (14.16) |

Where STB is defined by equation 27.1. Equations of EMX measures are repeated in section 26.

# 15. NNSM

Jaccard measure, which is an effective similarity measures, focuses on whether items are rated but it does not concern magnitude rating values like other measures. We overcome this drawback by proposing a so-called numerical nearby similarity measure (NNSM) which concerns magnitude rating values and keeps strong point of Jaccard measure. In other words, NNSM combines sums of rating values and cardinalities of item sets. Equation 15.1 specifies NNSM.

|  |  |
| --- | --- |
|  | (15.1) |

Note that |*I*1 ∩ *I*2| is the number of items rated by both user 1 and user 2, |*I*1| is the number of items rated by only user 1, and |*I*2| is the number of items rated by only user 2. It is easy to recognize that NNSM is an interesting advanced variant of cosine measure with support of Jaccard measure. However, NNSM is totally different from combination of cosine and Jaccard as CosineJ. Experimental section will mention evaluation of NNSM and CosineJ. Anyway, NNSM is simpler than CosineJ.

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 NNSM as an example. Of course, we have *I*1 = {1, 2, 3, 4, 6} and *I*2 = {1, 4, 5, 6}. We also have:

NNSM is associated with Jaccard measure, which derives NNSMJ measure as follows:

|  |  |
| --- | --- |
|  | (15.2) |

# 16. TA

Cosine measure is effective but it has a drawback that there may be two end points of two vectors which are far from each other according to Euclidean distance, but their cosine is high. This is negative effect of Euclidean distance which decreases accuracy of cosine similarity. Therefore, a so-called triangle area (TA) measure (Nguyen & Amer, 2019) is proposed as an improved version of cosine measure. Figure 17.1 illustrates TA measure.

A close up of a map

Description automatically generated

**Figure 17.1.** Triangle area (TA) measure with 0 ≤ *α* ≤ *π*/2

TA measure uses ratio of basic triangle area to whole triangle area as reinforced factor for Euclidean distance so that it can alleviate negative effect of Euclidean distance whereas it keeps simplicity and effectiveness of both cosine measure and Euclidean distance in making similarity of two vectors. TA is considered as an advanced cosine measure. TA is defined by equation 16.1 (Nguyen & Amer, 2019):

|  |  |
| --- | --- |
|  | (16.1) |

Where, as usual:

Let TAJ denote the combined measure which combines TA measure and Jaccard measure. TAJ measure is defined as follows:

|  |  |
| --- | --- |
|  | (16.2) |

Let *rm* be median of rating values, TA measure is normalized as TAN measure as follows:

|  |  |
| --- | --- |
|  | (16.3) |

By combined with Jaccard measure, TAN measure becomes TANJ measure as follows:

|  |  |
| --- | --- |
|  | (16.4) |

As a convention, TA family includes TA, TAJ, TAN, and TAJ. Hence, equation 16.1 is the key of TA family.

# 17. RA

Chen et al. (Chen, Zhang, Liu, Gao, & Zhou, 2016) consider the rating matrix as a user-item (user-object) bipartite network in which every link in the network represents the rating that a user rated on an item. Chen et al. (Chen, Zhang, Liu, Gao, & Zhou, 2016, p. 608) stated that “the resource-allocation (RA) process is equivalent to the one-step random walk in the user-object bipartite network starting from the common neighbors”. Therefore, the RA index between user 1 and user 2 is (Chen, Zhang, Liu, Gao, & Zhou, 2016, p. 608):

|  |  |
| --- | --- |
|  | (17.1) |

Where |*vj*| is module of item *j*,

Recall that *Jj* is the set of users who rated on item *j*. In the original article, Chen et al. actually calculated the RA index between two items, but their model can be extended to both item-based NN algorithm and user-based NN algorithm. Chen et al. (Chen, Zhang, Liu, Gao, & Zhou, 2016, pp. 608 - 609) combine cosine measure and RA index to derive a new measure called CosRA as follows:

|  |  |
| --- | --- |
|  | (17.2) |

It is interesting to extend CosRA measure of Chen et al. as a combination of RA index and Pearson measure as follows:

|  |  |
| --- | --- |
|  | (17.3) |

# 18. Entropy

Lee (Lee, Entropy-weighted similarity measures for collaborative recommender systems, 2018) used entropy to improve similarity measure; exactly Lee attached information entropy to measures such as cosine and Pearson. Let *rmin* and *rmax* be the minimum value and maximum value of ratings and let *r*(*j*) denote the rating for item *j* (users who rated such item *j* are not concerned yet), the entropy of item *j* denoted *E*(*j*) is defined as follows (Lee, Improving Jaccard Index for Measuring Similarity in Collaborative Filtering, 2017, p. 3):

|  |  |
| --- | --- |
|  | (18.1) |

Note, *P*(*r*(*j*)=*v*) is the probability that users rated item *j* with rating value *v* (Lee, Improving Jaccard Index for Measuring Similarity in Collaborative Filtering, 2017, p. 3).

Note, *ruj* is the rating that user *u* rated on item *j*. Finally, Lee (Lee, Improving Jaccard Index for Measuring Similarity in Collaborative Filtering, 2017, p. 3) developed the entropy cosine and entropy Pearson by attaching the entropy to cosine and Pearson as follows:

|  |  |
| --- | --- |
|  | (18.2) |

|  |  |
| --- | --- |
|  | (18.3) |

# 19. Kullback - Leibler divergence

The drawback of traditional measures such as cosine and Pearson is that they are calculated based on the common items that two concerned users rated. Therefore, Deng et al. (Deng, et al., 2018, p. 571) proposed the similarity model which does not require such common rated items, as follows:

|  |  |
| --- | --- |
|  | (19.1) |

The quantity *s*(*r*1*j*, *r*2*k*) is called local similarity function based on sigmoid function, which is defined as follows (Deng, et al., 2018, p. 571):

|  |  |
| --- | --- |
|  | (19.2) |

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

The quantity simitem(*j*, *k*) is the similarity measure of item *j* and item *k* which does not require common (co-rating) users. Deng et al. applied Kullback - Leibler (KL) divergence into calculating the similarity simitem(*j*, *k*) as follows (Deng, et al., 2018, p. 572):

|  |  |
| --- | --- |
|  | (19.3) |

Note, *rmin* and *rmax* are the minimum value and maximum value of ratings, respectively. Where *pjv* (*pkv*) is the probability of rating value *v* that item *j* (item *k*) received among its ratings (Deng, et al., 2018, p. 572).

Therefore, the measure model of Deng et al. is also called KL measure. Note, KL divergence is a concept in information theory like entropy.

# 20. OS

Gazdar and Hidri (Gazdar & Hidri, 2019) proposed a so-called Our developed Similarity (OS) measure when taking account kernel function. Their OS is product of the so-called Percentage of Non-Common Ratings (PNCR) and the Absolute Difference of Ratings (ADR), as follows (Gazdar & Hidri, 2019, p. 19):

|  |  |
| --- | --- |
|  | (20.1) |

Where (Gazdar & Hidri, 2019, p. 19),

|  |  |
| --- | --- |
|  | (20.2) |

|  |  |
| --- | --- |
|  | (20.3) |

Note, |***V***| is the number of all items and max(*r*1*j*, *r*2*j*) is the maximum value among *r*1*j* and *r*2*j*. Moreover, quantities PNCR(*u*1, *u*2) and ADR(*u*1, *u*2) can be considered as independent similarity measures.

# 21. SMC

Verma and Aggarwal (Verma & Aggarwal, 2019) proposed the Simple Matching Coefficient (SMC) measure by using statistical concept “statistical ratio match”. 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, two rating values *r*1*j* and *r*2*j* match together if there are three cases as follows (Verma & Aggarwal, 2019, p. 42):

1. They are equal as *r*1*j* = *r*2*j*.
2. Both of them are smaller than *rm* as *r*1*j* < *rm* and *r*2*j* < *rm*.
3. Both of them are larger than *rm* as *r*1*j* > *rm* and *r*2*j* > *rm*.

Therefore, Verma and Aggarwal defined SMC as the ratio of the number of matched items to the total number of items as follows (Verma & Aggarwal, 2019, p. 42):

|  |  |
| --- | --- |
|  | (21.1) |

# 22. ESim

ESim measure developed by Ali Amer and implemented by Loc Nguyen alleviates incomplete problem by ignoring missing values instead of aligning non-missing values of both rating vectors. Moreover, ESim removes second-order operators from its equation. In particular, ESim is defined as follows:

|  |  |
| --- | --- |
|  | (22.1) |

It is easy to recognize that ESim, which is similar to NNSM, is a variant of cosine measure with support of Jaccard measure except that ESim does not concern the length of rating vector when it focuses on the additional module of rating vectors. Recall that the equation of NNSM is:

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), we calculate ESim as follows:

ZSim is a variant of ESim in which its denominator is replaced by product of rating values, as follows:

|  |  |
| --- | --- |
|  | (22.2) |

It is easy to recognize that ZSim is more similar to Cosine than ESim. 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), we calculate ZSim as follows:

ESim2 is a variant of ZSim where its nominator is improved by ignoring missing values as follows:

|  |  |
| --- | --- |
|  | (22.3) |

Where,

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), we calculate ESim2 as follows:

Both ZSim and ESim2 are strengthened with the number of commonly rating values . ESim3 is another variant of ZSim, which is strengthened by the number of non-missing values for both user 1 and user 2, as follows:

|  |  |
| --- | --- |
|  | (22.4) |

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), we calculate ZSim as follows:

Measure IZSM developed is a variant of ZSim measure, in which missing values are replaced by 1 in case that rating values are greater than 1.

Where,

IZSM will be only effective if rating values are larger than 1. 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), we calculate IZSM as follows:

ESim is modified so as to derive a so-called Nearby measure as follows:

Indeed, Nearby measure is NNSM measure.

# 23. RES

By inspiring the physical resonance of simple harmonic motion, Tan and He (Tan & He, 2017) proposed a new similarity resonance similarity (RES) measure. The believed that user rating behavior is similar to the simple harmonic motion of particles in vibration system. RES is product of three components such as consistence component, distance component, and Jaccard component.

|  |  |
| --- | --- |
|  | (23.1) |

Note, *vj* denotes item *j* whereas *k*1, *k*2, *k*3, and *k*4 are predefined constants. Let *rmin* and *rmax* be the minimum value and maximum value of ratings, respectively and 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. Let *μ* be the mean over all rating values and let and be rating means of user *i* and item *j*.

Consistency component *C*(*u*1, *u*2, *vj*, *k*1) reflecting the harmonic motion is calculated based on the initial phase *φ* of harmonic vibration *x*(*t*) = *A*cos(*ωt* + *φ*), as follows:

|  |  |
| --- | --- |
|  | (23.2) |

Where,

Note,

Distance component reflecting distance between two ratings is defined as follows:

|  |  |
| --- | --- |
|  | (23.3) |

Where,

Jaccard component is Jaccard measure powered with constant *k*4.

|  |  |
| --- | --- |
|  | (23.4) |

# 24. SM

Singularity implies an item that very few users rated it and so it is often ignored in calculating similarity but such item is valuable to recognizing two similar users. For instance, if both users rated the same singular items, they are highly similar together. Bobadilla et al. (Bobadilla, Ortega, & Hernando, 2012) propose a so-called singularity measure (SM) to concern singular items. 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, hence, rating values which are greater or equal than *rm* are called relevant ratings and the others which are smaller than *rm* are called non-relevant. Let *Rj* and *Nj* are sets of users who gave relevant ratings and non-relevant ratings on item *j*, respectively. Thus, *SRj* is defined as the singularity of item *j* with regard to relevant values and *SNj* is defined as the singularity of item *j* with regard to non-relevant values.

|  |  |
| --- | --- |
|  | (24.1) |

Essentially, SM is an extension of the mean squared difference (MSD) measure with support of singularity quantities like *SRj* and *SNj*, which is defined as follows:

|  |  |
| --- | --- |
|  | (24.2) |

Where *A* is the set of items which are received relevant ratings from both user 1 and user 2, *B* is the set of items which are received non-relevant ratings from both user 1 and user 2, and *C* is the set of items which are received relevant ratings from one user and non-relevant ratings from the other.

# 25. Coco

Cosine measure is based on dot product which is to make multiplication on pairs of elements obtained from two rating vectors. Because cosine measure only concerns commonly rated items, gaps between two vectors can be enlarged when there are a lot of elements inside the two vectors but there are few common elements. Derived from ideology of dot product, a so-called circle product aims to filling spaces inside pairs of vector elements by taking multiplication over all vector elements, as follows:

|  |  |
| --- | --- |
|  | (25.1) |

From the equation above, it is easy recognize that circle product is only valid if elemental dimensions have the same meaning. In the context of rating matrix, all rating values have the same meaning as ratings that users gave on items within the same scale. Indeed, circle product is the product of sum of two vectors.

The so-called circle length is defined based on circle product, as follows:

|  |  |
| --- | --- |
|  | (25.2) |

In this research, a so-called Coco measure is proposed based on circle product and normal vector length as follows:

|  |  |
| --- | --- |
|  | (25.3) |

In other words, Coco measure replaces dot product by circle product, which is its different aspect from cosine. 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), their Coco measure is:

Coco associated with Jaccard produces CocoJ measure as follows:

|  |  |
| --- | --- |
|  | (25.4) |

# 26. URP

User rating preference (URP) measure is defined based on difference between two user ratings regarding their means and standard deviation as follows:

|  |  |
| --- | --- |
|  | (26.1) |

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.

Given two variables *x* and *y*, kernel function kernel(*x*, *y*), which is essentially exponential function of deviation between *x* and *y*, measures the similarity between *x* and *y*, which means that the larger the kernel function is, the more the similarity is. The implicit meaning of kernel function is to make the deviation between *x* and *y* to be smooth and inverse. Making some deviation smooth is an excellent aspect of exponential function due to its interpolation ability.

When the kernel function is normalized in unit (1), it can be varied as follows:

The equation above is fundamental of URP and EMX when *x* and *y* are replaced by their means and their standard deviations. EMX proposed by Amer is an improvement of URP by combining URP with other measures such as Amer, NNSM, Coco, and STB as well as standard deviations are removed from equation of EMX. The integrations of URP and Amer, NNSM are called EMX1 and EMX2 defined as follows:

|  |  |
| --- | --- |
|  | (26.2) |

|  |  |
| --- | --- |
|  | (26.3) |

Where measures NNSM and Amer are defined by equations 15.1 and 14.4. The integrations of URP and Coco, Amer are called EMX3 and EMX4 defined as follows:

|  |  |
| --- | --- |
|  | (26.4) |

|  |  |
| --- | --- |
|  | (26.5) |

Where Coco is defined by equation 25.1. The integrations of URP and STB, Amer are called EMX5 and EMX6 defined as follows:

|  |  |
| --- | --- |
|  | (26.5) |

|  |  |
| --- | --- |
|  | (26.7) |

Where STB is defined by equation 27.1.

# 27. STB

Amer proposed a so-called STB measure which is focused on product of sum of rating values called product-sum. In particular, STB is similarity of two rating vectors. Let *X* be product-sum of common part:

Let *Y* be product-sum of different part:

Note, notation “\” denote complement operator in set theory. Let Z be the whole product-sum:

STB is similarity of two rating vectors with regard to product-sums, defined as follows:

# 28. RPB

RPB measure is trigonometric cosine function of statistical deviations, specified as follows (Ayub M. , et al., 2019, p. 6):

|  |  |
| --- | --- |
|  | (28.1) |

Note, and var(*ui*) are mean and variance of *ui*, respectively.

In practice, the variance of *ui* can be replaced by the standard deviation of *ui* as follows (Ayub M. , et al., 2019, p. 6):

# 29. ADR

Absolute Difference of Ratings (ADR) is defined as follows (Gazdar & Hidri, 2019, p. 19),

|  |  |
| --- | --- |
|  | (29.1) |

Note, max(*r*1*j*, *r*2*j*) is the maximum value among *r*1*j* and *r*2*j*.

# 30. Singularity and CLAG

Jin et al. (Jin, Zhang, Cai, & Zhang, 2020) combined local similarity and global similarity with support of singularity factor. As a result, they proposed a so-called Combined Local and Global (CLAG) 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, hence, rating values which are greater or equal than *rm* are called relevant ratings and the others which are smaller than *rm* are called non-relevant. Jin et al. called “relevant” and “non-relevant” as “positive” and “negative”, respectively. Given user 1 and user 2, for every item *j*, let:

* If both *r*1*j* and *r*2*j* are positive, item *j* belongs to the predefined set *PA*.
* If both *r*1*j* and *r*2*j* are negative, item *j* belongs to the predefined set *NA*.
* If *r*1*j* and *r*2*j* are different which means that one is positive and the other is negative, item *j* belongs to the predefined set *D*.

Let *SjP* and *SjN* be positive singularity and negative singularity

Where *Pj* is the set of users who gave positive ratings to item *j* and *Nj* is the set of users who gave negative ratings to item *j*. Jin et al. (Jin, Zhang, Cai, & Zhang, 2020, p. 6) defined their singularity measure as follows:

|  |  |
| --- | --- |
|  | (30.1) |

Where,

Note, is mean of *ui*.

According to Jin et al. (Jin, Zhang, Cai, & Zhang, 2020, p. 6), global measure is defined based on Jaccard measure as follows:

|  |  |
| --- | --- |
|  | (30.2) |

Let {*b*1, *b*2,…, *bn*} be bins of rating values, for instance, if rating values range from 1 to 5, these bins are {1, 2, 3, 4, 5}. Let *Fi* = {*fi*1, *fi*2,…, *fin*} be the frequency vector of user *i* with regard to the bins {*b*1, *b*2,…, *bn*}. Concretely, *fij* is the number of items that user *i* rated with bin *bj*. Jin et al. (Jin, Zhang, Cai, & Zhang, 2020, p. 6) defined the bin measure which is proportioned to dot product of *F*1 and *F*2 regarding user 1 and user 2.

|  |  |
| --- | --- |
|  | (30.3) |

Finally, Jin et al. (Jin, Zhang, Cai, & Zhang, 2020, p. 6) proposed their CLAG measure as the product of singularity measure, global measure and bin measure as follows:

|  |  |
| --- | --- |
|  | (30.4) |

# 31. EXP

Exponential measure are the measures that take advantages of exponential function which can be considered as sigmoid function. They are denoted as EXP measure. Particularly, EXP takes exponential function of MSD measure, which is smooth technique to alleviate small errors in evaluating differences between two vectors. Moghadam et al. (Moghadam, Heidari, Moeini, & Kamandi, 2019) proposed their EXP as follows:

|  |  |
| --- | --- |
|  | (31.1) |

Where *rmin* and *rmax* are minimum rating value and maximum rating value, respectively, for example, if rating values range from 1 to 5, then *rmin* = 1 and *rmax* = 5. The quantity indicates the number of items rated by both users.

# Appendix B

Because this report only lists basic, main, or important measures whereas there are a huge number of variants and combinations which derived from measure groups, this appendix *B* lists some derived measures in set *B* from the paired label *XY*.

Replacing the built-in Jaccard2 in NHMS by SMD and Amer produce measures called NHMS.SMD and NHMS.Amer, respectively as follows:

|  |  |
| --- | --- |
|  | (B7.1) |

|  |  |
| --- | --- |
|  | (B7.2) |

Where PSS, URP, SMD, and Amer are specified by equations 7.1, 26.1, 14.1, and 14.4, respectively.

SMD associated with Jaccard produce a so-called SMDJ measure as follows:

|  |  |
| --- | --- |
|  | (B14.1) |

Where SMD and Jaccard are specified by equations 14.1 and 1.1, respectively.

Amer associated with Jaccard produces measures called AmerJ as follows:

|  |  |
| --- | --- |
|  | (B14.2) |

# Appendix C

Because this report only lists basic, main, or important measures whereas there are a huge number of variants and combinations which derived from measure groups, this appendix C lists some derived measures in set *C* from the paired label *XY*.

IJ associated with cosine, Pearson, PSS, Amer, TA, and Coco produces measures called IJ.Cosine, IJ.Pearson, IJ.PSS, IJ.Amer, IJ.TA, and IJ.Coco, respectively as follows:

|  |  |
| --- | --- |
|  | (C1.1) |

|  |  |
| --- | --- |
|  | (C1.2) |

|  |  |
| --- | --- |
|  | (C1.3) |

|  |  |
| --- | --- |
|  | (C1.4) |

|  |  |
| --- | --- |
|  | (C1.5) |

|  |  |
| --- | --- |
|  | (C1.6) |

Where IJ, cosine, Pearson, PSS, Amer, TA, and Coco are specified by equations 1.4, 2.1, 7.1, 14.4, 16.1, and 25.3, respectively.

Amer associated with cosine, Pearson, CPC, PSS, QTI, NNSM, and TA produces measures called Amer.Cosine, Amer.Pearson, Amer.CPC, Amer.PSS, Amer.QTI, Amer.NNSM, and Amer.TA, respectively as follows:

|  |  |
| --- | --- |
|  | (C14.1) |

|  |  |
| --- | --- |
|  | (C14.2) |

|  |  |
| --- | --- |
|  | (C14.3) |

|  |  |
| --- | --- |
|  | (C14.4) |

|  |  |
| --- | --- |
|  | (C14.5) |

|  |  |
| --- | --- |
|  | (C14.6) |

|  |  |
| --- | --- |
|  | (C14.7) |

Where Amer, cosine, Pearson, CPC, PSS, QTI, NNSM, and TA are specified by equations 14.4, 2.1, 3.1, 3.2, 7.1, 14.5, 15.1, 16.1, respectively.

# Appendix E

This appendix lists some examples. Here some examples related to Coco measures including Coco, CocoJ, and CocoIJ are described. For example, given two rating vectors *u*3 = (*r*31=4, *r*32=1, *r*33=5, *r*34=5), and *u*4 = (*r*41=1, *r*42=2, *r*43=?, *r*44=?) from table 0.1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 = ? |

About Jaccard measure, we have *I*3 = {1, 2, 3, 4} and *I*4 = {1, 2} and so we obtain from equation 1.1:

About IJ measure, we have *PA* = {}, *NA* = {1, 2}, *D* = {}, *PO* = {3, 4}, *NO* = {}, *P*1=1, *N*1=3, *E*1=0, *P*2=0, *N*2=4, *E*2=0, *P*3=1, *N*3=2, *E*3=1, *P*4=3, *N*4=0, and *E*4=1 when *rm*=3. We also obtain *S*1*P* = 1 – 1/4 = 3/4, *S*1*N* = 1 – 3/4 = 1/4, *S*1*E* = 1 – 0/4 = 1, *S*2*P* = 1 – 0/4 = 1, *S*2*N* = 1 – 4/4 = 0, *S*2*E* = 1 – 0/4 = 1, *S*3*P* = 1 – 1/4 = 3/4, *S*3*N* = 1 – 2/4 = 1/2, *S*3*E* = 1 – 1/4 = 3/4, *S*4*P* = 1 – 3/4 = 1/4, *S*4*N* = 1 – 0/4 = 1, and *S*4*E* = 1 – 1/4 = 3/4. From equation 1.4, IJ measure of *u*3 and *u*4 is:

From equation 25.3, Coco measure of *u*3 and *u*4 is:

From equation 25.4, CocoJ measure of *u*3 and *u*4 is:

From equation C1.6, the combined measure of Coco and IJ on *u*3 and *u*4 is:

# References

Al‑Shamri, M. H. (2021, October 30). Similarity modifers for enhancing the recommender system performance. *Applied Intelligence*. doi:10.1007/s10489-021-02900-7

Ayub, M., Ghazanfar, A. M., Khan, T., & Saleem, A. (2020, May 12). An Efective Model for Jaccard Coefcient to Increase the Performance of Collaborative Filtering. (B. M. Ali, Ed.) *Arabian Journal for Science and Engineering, 45*(12), 9997 - 10017. doi:10.1007/s13369-020-04568-6

Ayub, M., Ghazanfar, M. A., Mehmood, Z., Saba, T., Alharbey, R., Munshi, A. M., & Alrige, M. A. (2019, August 1). Modeling user rating preference behavior to improve the performance of the collaborative filtering based recommender systems. *PLoS ONE, 14*(8), e0220129. doi:10.1371/journal.pone.0220129

Bag, S., Kumar, S. K., & Tiwari, M. K. (2019, January 11). An Efective Model for Jaccard Coefcient to Increase the Performance of Collaborative Filtering. (W. Pedrycz, Ed.) *Information Sciences, 483*(May 2019), 53 - 64. doi:10.1016/j.ins.2019.01.023

Bobadilla, J., Ortega, F., & Hernando, A. (2012, March). A collaborative filtering similarity measure based on singularities. *Information Processing and Management, 48*(2), 204-217. doi:10.1016/j.ipm.2011.03.007

Chen, L.-J., Zhang, Z.-K., Liu, J.-H., Gao, J., & Zhou, T. (2016, October 3). A vertex similarity index for better personalized recommendation. (H. W. Capel, Ed.) *Physica A, 466*(2017), 607 - 615. doi:10.1016/j.physa.2016.09.057

Choi, K., & Suh, Y. (2013, January). A new similarity function for selecting neighbors for each target item in collaborative filtering. (H. Fujita, & J. Lu, Eds.) *Knowledge-Based Systems, 37*(2013), 146-153. doi:10.1016/j.knosys.2012.07.019

Deng, J., Wang, Y., Guo, J., Deng, Y., Gao, J., & Park, Y. (2018, October 29). A similarity measure based on Kullback-Leibler divergence for collaborative filtering in sparse data. *Journal of Information Science, 45*(5), 656-675. doi:10.1177/0165551518808188

Feng, J., Fengs, X., Zhang, N., & Peng, J. (2018, September 24). An improved collaborative filtering method based on similarity. (H. Wang, Ed.) *PLos ONE, 13*(10), 1-18. doi:10.1371/journal.pone.0204003

Gazdar, A., & Hidri, L. (2019, September 25). A new similarity measure for collaborative filtering based recommender systems. (J. Lu, Ed.) *Knowledge-Based Systems, 188*(2020). doi:10.1016/j.knosys.2019.105058

Harris, E. F., & Sjøvold, T. (2018, September 7). Calculation of Smith's Mean Measure of Divergence for Intergroup Comparisons Using Nonmetric Data. (E. F. Harris, Ed.) *Dental Anthropology: A Publication of the Dental Anthropology Association, 17*(3), 83-93. doi:10.26575/daj.v17i3.152

Hyung, J. A. (2008, January 2). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. (W. Pedrycz, Ed.) *Information Sciences, 178*(1), 37-51. doi:10.1016/j.ins.2007.07.024

Jin, Q., Zhang, Y., Cai, W., & Zhang, Y. (2020, January 10). A New Similarity Computing Model of Collaborative Filtering. *IEEE Access, 8*, 17594 - 17604. doi:10.1109/ACCESS.2020.2965595

Jindal, A., Sharma, N., & Verma, V. (2022). Joyful Jaccard: An Analysis of Jaccard-Based Similarity Measures in Collaborative Recommendations. *International Conference on Artificial Intelligence and Sustainable Engineering: Select Proceedings of AISE 2020.* *1.* Springer Nature. doi:10.1007/978-981-16-8542-2\_3

Lee, S. (2017). Improving Jaccard Index for Measuring Similarity in Collaborative Filtering. In K. Kim, & N. Joukov (Ed.), *Information Science and Applications 2017 (ICISA 2017).* *424*, pp. 799 - 806. Macau: Springer. doi:10.1007/978-981-10-4154-9\_93

Lee, S. (2018). Entropy-weighted similarity measures for collaborative recommender systems. *AIP Conference Proceedings.* *1982*, pp. 1-6. AIP Publishing. doi:10.1063/1.5045417

Liang, S., Ma, L., & Yuan, F. (2015, 10). A singularity-based user similarity measure for recommender systems. *International Journal of Innovative Computing Information and Control, 11*(5), 1629-1638. doi:10.24507/ijicic.11.05.1629

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

Manochandar, S., & Punniyamoorthy, M. (2020, August 22). A new user similarity measure in a new prediction model for collaborative filtering. (M. Ali, & H. Fujita, Eds.) *Applied Intelligence, 51*(1), 586 - 615. doi:10.1007/s10489-020-01811-3

Moghadam, P. H., Heidari, V., Moeini, A., & Kamandi, A. (2019, September 6). An exponential similarity measure for collaborative filtering. *SN Applied Sciences, 1*(10), 1-4. doi:10.1007/s42452-019-1142-8

Mu, Y., Xiao, N., Tang, R., Luo, L., & Yin, X. (2019, January). An Efficient Similarity Measure for Collaborative Filtering. (R. Bie, Y. Sun, & J. Yu, Eds.) *Procedia Computer Science, 147*(2019), 416-421. doi:10.1016/j.procs.2019.01.258

Nguyen, L., & Amer, A. A. (2019, October 17). Advanced Cosine Measures for Collaborative Filtering. (ITS, Ed.) *Adaptation and Personalization (ADP), 1*(1), 21-41. doi:10.31058/j.adp.2019.11002

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

Singh, P. K., Sinha, S., & Choudhury, P. (2022, January 25). An improved item-based collaborative filtering using a modified Bhattacharyya coefficient and user–user similarity as weight. *Knowledge and Information Systems, 64*, 665-701. doi:10.1007/s10115-021-01651-8

Sun, S.-B., Zhang, Z.-H., Dong, X.-L., Zhang, H.-R., Li, T.-J., Zhang, L., & Min, F. (2017, August 17). Integrating Triangle and Jaccard similarities for recommendation. (Q. Zou, Ed.) *PLoS ONE, 12*(8), 1-16. doi:10.1371/journal.pone.0183570

Suryakant, & Mahara, T. (2016, January). A New Similarity Measure Based on Mean Measure of Divergence for Collaborative Filtering in Sparse Environment. (R. K. Venugopal, R. Buyya, & M. L. Patnaik, Eds.) *Procedia Computer Science, 89*(2016), 450-456. doi:10.1016/j.procs.2016.06.099

Tan, Z., & He, L. (2017, November 29). An Efficient Similarity Measure for User-Based Collaborative Filtering Recommender Systems Inspired by the Physical Resonance Principle. *IEEE Access, 5*, 27211 - 27228. doi:10.1109/ACCESS.2017.2778424

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

Verma, V., & Aggarwal, R. K. (2019). A New Similarity Measure Based on Simple Matching Coefficient for Improving the Accuracy of Collaborative Recommendations. (M. He, & Y. V. Bodyanskiy, Eds.) *International Journal of Information Technology and Computer Science, 11*(6), 37-49. doi:10.5815/ijitcs.2019.06.05