**List of similarity measures in collaborative filtering**

# 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.

# 1. Introduction

Before descripting popular similarity measures, we need to concern some important conventions and notations. 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. Missing values (missing ratings) are denoted as question mask “?”. For example, we have (user) rating vectors *u*1 = (*r*11=1, *r*12=2, *r*13=1, *r*14=5), *u*2 = (*r*21=2, *r*22=1, *r*23=2, *r*24=4), *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=?). User-based rating matrix can be transposed into item-based rating matrix in which each row is a rating vector of a specified item. For example, we have (item) rating vectors *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=?). 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.

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. 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).

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

Where and are mean values of *u*1 and *ui*, respectively. Equation 1.1 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:

|  |  |
| --- | --- |
|  |  |

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.

# 2. 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):

|  |  |
| --- | --- |
|  |  |

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):

|  |  |
| --- | --- |
|  |  |

Liang, Ma, and Yuan (Liang, Ma, & Yuan, 2015) proposed an improved variant of Jaccard based on the concept of singularity. Their measure called Improved Jaccard (IJ) is specified as follows (Liang, Ma, & Yuan, 2015, p. 1634):

|  |  |
| --- | --- |
|  |  |

Where *SjP*, *SjN*, and *SjE* are positive singularity, negative singularity, and empty singularity of item j, respectively whereas *PA*, *NA*, *D*, *PO*, and *NO* are sets of agreed positive ratings, agreed negative ratings, disagreed ratings, positive ratings, and negative ratings, respectively.

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):

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

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):

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

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):

|  |  |
| --- | --- |
|  |  |

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 is the numbers of items which are 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. 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):

|  |  |
| --- | --- |
|  |  |

# 3. Cosine

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

|  |  |
| --- | --- |
|  |  |

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:

|  |  |
| --- | --- |
|  |  |

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:

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

# 4. Pearson

Pearson correlation is another popular similarity measure besides cosine, 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. 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):

|  |  |
| --- | --- |
|  |  |

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):

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

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):

|  |  |
| --- | --- |
|  |  |

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):

|  |  |
| --- | --- |
|  |  |

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:

|  |  |
| --- | --- |
|  |  |

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:

|  |  |
| --- | --- |
|  |  |

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.

# 5. 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):

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

# 6. SRC

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

|  |  |
| --- | --- |
|  |  |

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.

# 7. 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.

|  |  |
| --- | --- |
|  |  |

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.

|  |  |
| --- | --- |
|  |  |

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.

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

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):

|  |  |
| --- | --- |
|  |  |

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):

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

Note, *μj* is average rating of item *j*.

# 8. 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):

|  |  |
| --- | --- |
|  |  |

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 a new heuristic similarity model (NHSM) 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 NHSM measure. From experimental result, NHSM gave out excellent estimation.

# 9. 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):

|  |  |
| --- | --- |
|  |  |

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 1.2, 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:

|  |  |
| --- | --- |
|  |  |

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:

|  |  |
| --- | --- |
|  |  |

# 10. 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 1.1 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):

|  |  |
| --- | --- |
|  |  |

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.

|  |  |
| --- | --- |
|  |  |

Experimental result proved that CjacMD model is effective similarity model.

# 11. 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:

|  |  |
| --- | --- |
|  |  |

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:

|  |  |
| --- | --- |
|  |  |

Experimental result proved that TMJ is effective measure.

# 12. 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:

|  |  |
| --- | --- |
|  |  |

Experimental result proved that Feng model is effective similarity model.

# 13. 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):

|  |  |
| --- | --- |
|  |  |

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 1.1 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:

|  |  |
| --- | --- |
|  |  |

Experimental result proved that Mu measure is effective similarity model.

# 14. 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):

|  |  |
| --- | --- |
|  |  |

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:

|  |  |
| --- | --- |
|  |  |

# 15. 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:

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

Let *I*1 or *I*2 be sets of indices of items that user 1 or user 2 rates, respectively. 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:

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

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 15.1, 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 15.2, as follows:

When HSMD measure is combined with Jaccard measure, it is called HSMDJ which is specified by equation 15.3.

|  |  |
| --- | --- |
|  | (15.3) |

# 16. NNMS

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 measure (NNMS) which concerns magnitude rating values and keeps strong point of Jaccard measure. In other words, NNMS combines sums of rating values and cardinalities of item sets. Equation 16.1 specifies NNMS.

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

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 NNMS is an interesting advanced variant of cosine measure with support of Jaccard measure. However, NNMS is totally different from combination of cosine and Jaccard as CosineJ. Experimental section will mention evaluation of NNMS and CosineJ. Anyway, NNMS 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 NNMS as an example. Of course, we have *I*1 = {1, 2, 3, 4, 6} and *I*2 = {1, 4, 5, 6}. We also have:

# 17. 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 17.1 (Nguyen & Amer, 2019):

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

Where, as usual:

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

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

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

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

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

|  |  |
| --- | --- |
|  | (17.4) |

As a convention, TA family includes TA, TAJ, TAN, and TAJ. Hence, equation 17.1 is the key of TA family.

# 18. 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):

|  |  |
| --- | --- |
|  |  |

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:

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

# 19. 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):

|  |  |
| --- | --- |
|  |  |

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:

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

# 20. 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:

|  |  |
| --- | --- |
|  |  |

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):

|  |  |
| --- | --- |
|  |  |

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):

|  |  |
| --- | --- |
|  |  |

Note, *rmin* and *rmax* be the minimum value and maximum value of ratings. 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.

# References

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

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

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

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

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

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

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