Package: CFF (via r-universe)

October 17, 2024

Title Simple Similarity for User-Based Collaborative Filtering Systems

Version 1.0

Date 2020-02-25

Maintainer Farimah Houshmand Nanehkaran <hoshmandcomputer@gmail.com>

Description A simple, fast algorithm to find the neighbors and similarities of users in user-based filtering systems, to break free from the complex computation of existing similarity formulas and the ability to solve big data.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.0.2

NeedsCompilation no

Author Farimah Houshmand Nanehkaran [aut, cre] (<https://orcid.org/0000-0003-1687-1719>, University=Islamic Azad University of Kashan,Kashan,Iran), Seyed Mohammad Reza Lajevardi [ctb] (<https://orcid.org/0000-0002-4744-2784>, University=Islamic Azad University of Kashan,Kashan,Iran), Mahmoud Mahlouji Bidgholi [ctb] (<https://orcid.org/0000-0001-8895-8501>, University=Islamic Azad University of Kashan,Kashan,Iran)

Date/Publication 2020-03-04 10:00:05 UTC

Repository https://hoshmandcomputer.r-universe.dev

RemoteUrl https://github.com/cran/CFF

RemoteRef HEAD

RemoteSha 1d642bac05e7be4fbc3fc3a8f2d8464a06441cf7

Contents

CFF-package *Simple Similarity for User-Based Collaborative Filtering Systems*

Description

A simple, fast algorithm to find the neighbors and similarities of users in user-based filtering systems, to break free from the complex computation of existing similarity formulas and the ability to solve big data.

Details

The DESCRIPTION file:

Package:	CFF
Title:	Simple Similarity for User-Based Collaborative Filtering Systems
Version:	1.0
Date:	2020-02-25
Authors@R:	c(person(given="Farimah", family="Houshmand Nanehkaran", role = c("aut", "cre"), email="hoshmandcomp
Maintainer:	Farimah Houshmand Nanehkaran <hoshmandcomputer@gmail.com>
Description:	A simple, fast algorithm to find the neighbors and similarities of users in user-based filtering systems, to brea
License:	GPL (>= 2)
Encoding:	UTF-8
RoxygenNote:	7.0.2
Author:	Farimah Houshmand Nanehkaran [aut, cre] (<https://orcid.org/0000-0003-1687-1719>, University=Islamic A

Index of help topics:

CFF-package	Simple Similarity for User-Based Collaborative Filtering Systems
Score_replace	Replacing of Neighbor Users' Ratings on Non-Rated Items By The Active User
simple_predict	Prediction Unseen Items For The Active User
simple_similarity	Finding Neighbor Users And Their Similarity Values

User-Based Collaborative Filtering Systems

Author(s)

NA

Maintainer: Farimah Houshmand Nanehkaran <hoshmandcomputer@gmail.com>

References

Kumar, P., Kumar, V., & Thakur, R. S. (2019). *A new approach for rating prediction system using collaborative filtering*. Iran Journal of Computer Science, vol.2, no. 2, pp. 81-87.

Zhang, P., Zhang, Z., Tian, T., & Wang, Y. (2019). *Collaborative filtering recommendation algorithm integrating time windows and rating predictions*. Applied Intelligence, vol. 49, no. 8, pp. 3146-3157.

Gadekula, S. K., Rao, U. P., Vyas, R. K., Dontula, A. L., & Gaikwad, S. V. (2019). *Improved Pearson Similarity for Collaborative Filtering Recommendation System*. In 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1047-1054, IEEE.

Examples

```
ratings <- matrix(c(  2,    5,  NaN,  NaN,  NaN,    4,
                    NaN,  NaN,  NaN,    1,  NaN,    5,
                    NaN,    4,    5,  NaN,    4,  NaN,
                      4,  NaN,  NaN,    5,  NaN,  NaN,
                      5,  NaN,    2,  NaN,  NaN,  NaN,
                    NaN,    1,  NaN,    4,    2,  NaN),nrow=6,byrow=TRUE)
active_users <- c(1:dim(ratings)[2])

time_all <- c(rep(NaN, length(active_users)))

ratings3 <- ratings

for (ac in 1:length(active_users))
{
  cat("=========== user",active_users[ac], "==================", "\n","\n")
  ##1
  T1_start <- Sys.time()
  sim <- simple_similarity(ratings, max_score=5, min_score=1, ac)
  T1_end <- Sys.time()

  cat("    Similar Users =", sim$sim_index,                      "\n","\n")
  cat("Similarity Values =", sim$sim_x,                          "\n","\n")

  ##2
  T2_start <- Sys.time()
  ratings2 <- Score_replace(ratings, sim_index= sim$sim_index, ac)
  T2_end   <- Sys.time()

  cat(" Predicted Scores =", ratings2[,ac],                      "\n","\n")

  ##3
  T3_start <- Sys.time()
  predictedItems <- simple_predict(ratings, ratings2, ac)
  T3_end <- Sys.time()

  cat("  Predicted Items =", predictedItems,                    "\n","\n")

  ##4
  time_all[ac] <- (T1_end - T1_start) + (T2_end - T2_start) + (T3_end - T3_start)
```

```
  cat("             Time =", time_all[ac],                      "\n","\n")

  ##5
  ratings3[,ac] <- ratings2[,ac]
}

Mean_Time <- mean(time_all)

cat("=========== Mean Time ==================",               "\n","\n")

cat("         Mean Time =", Mean_Time,                         "\n","\n")

cat("       Full Matrix =",                                   "\n","\n")

print(ratings3)
```

`Score_replace` *Replacing of Neighbor Users' Ratings on Non-Rated Items By The Active User*

Description

The ratings of each user that has more similar to the active user are directly replaced in his unseen items.

Usage

```
Score_replace(ratings, sim_index, ac)
```

Arguments

`ratings`	A rating matrix whose rows are items and columns are users.
`sim_index`	Descending sorted indexes based on similarity to the active user who is a vector of integers.
`ac`	The id of an active user as an integer ($1 \leq ac \leq lengthofusers$).

Details

The unseen items of the active user are filled by the ratings of the similar users, respectively. Each element remains unchanged after one placement.

Value

`ratings2`	A matrix the size of the original user-item matrix in which the active user's empty elements are filled.

Author(s)

Farimah Houshmand Nanehkaran

Maintainer: Farimah Houshmand Nanehkaran <hoshmandcomputer@gmail.com>

References

Gadekula, S. K., Rao, U. P., Vyas, R. K., Dontula, A. L., & Gaikwad, S. V. (2019). *Improved Pearson Similarity for Collaborative Filtering Recommendation System*. In 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1047-1054, IEEE.

Examples

```
ratings <- matrix(c(  2,    5,  NaN,  NaN,  NaN,    4,
                    NaN,  NaN,  NaN,    1,  NaN,    5,
                    NaN,    4,    5,  NaN,    4,  NaN,
                      4,  NaN,  NaN,    5,  NaN,  NaN,
                      5,  NaN,    2,  NaN,  NaN,  NaN,
                    NaN,    1,  NaN,    4,    2,  NaN),nrow=6,byrow=TRUE)

sim <- simple_similarity(ratings, max_score=5, min_score=1, ac=1)

ratings2 <- Score_replace(ratings, sim_index= sim$sim_index, ac=1)
```

---

simple_predict *Prediction Unseen Items For The Active User*

Description

In the predicted items list, items with more scores replace in top of the list.

Usage

```
simple_predict(ratings, ratings2, ac)
```

Arguments

ratings	A rating matrix whose rows are items and columns are users.
ratings2	A matrix the size of the original user-item matrix in which the active user's empty elements are filled.
ac	The id of an active user as an integer ($1 \leq ac \leq lengthofusers$).

Details

Collaborative filtering is a recommender system for predicting the missing ratings that an active user might have given to an item. These ratings have been calculated and accumulate in a vector by this function.

Value

predictedItems A sorted vector of predicted items based on the scores.

Author(s)

Farimah Houshmand Nanehkaran

Maintainer: Farimah Houshmand Nanehkaran <hoshmandcomputer@gmail.com>

References

Song, B., Gao, Y., & Li, X. M. (2020, January). *Research on Collaborative Filtering Recommendation Algorithm Based on Mahout and User Model*. In Journal of Physics: Conference Series, Vol. 1437, no. 1, p. 012095, IOP Publishing.

Ramakrishnan, G., Saicharan, V., Chandrasekaran, K., Rathnamma, M. V., & Ramana, V. V. (2020). *Collaborative Filtering for Book Recommendation System*. In Soft Computing for Problem Solving, pp. 325-338, Springer, Singapore.

Examples

```
ratings <- matrix(c(  2,    5,  NaN,  NaN,  NaN,    4,
                    NaN,  NaN,  NaN,    1,  NaN,    5,
                    NaN,    4,    5,  NaN,    4,  NaN,
                      4,  NaN,  NaN,    5,  NaN,  NaN,
                      5,  NaN,    2,  NaN,  NaN,  NaN,
                    NaN,    1,  NaN,    4,    2,  NaN),nrow=6,byrow=TRUE)

sim <- simple_similarity(ratings, max_score=5, min_score=1, ac=1)

ratings2 <- Score_replace(ratings, sim_index= sim$sim_index, ac=1)

predictedItems <- simple_predict(ratings, ratings2, ac=1)
```

---

simple_similarity *Finding Neighbor Users And Their Similarity Values*

Description

Steps of calculating the similarity of one user to an active user :

1- Calculating the difference between the desired user ratings with the active user in common items.

2- Calculating the similarity value for each common item.

3- Calculating the mean value of similarities.

Usage

```
simple_similarity(ratings, max_score=5, min_score=1, ac)
```

Arguments

ratings	A rating matrix whose rows are items and columns are users.
max_score	The maximum range of ratings.
min_score	The minimum range of ratings.
ac	The id of an active user as an integer ($1 \leq ac \leq lengthofusers$).

Details

The similarity of the active user with other users is obtained by the following formulas :

$$dif_{(u_i,j)} = |r_{(u_a,j)} - r_{(u_i,j)}|$$

$$sim_{dif_{(u_i,j)}} = \frac{-dif_{(u_i,j)}}{max_score - min_score} + 1$$

$$sim_{(u_a,u_j)} = \frac{\sum_{j=1}^{N_j} sim_{(dif_{(u_i,j)})}}{N_j}$$

j is the row number for the items and i is the column number for the users in the ratings matrix.

u_i is a i*th* column user and u_a is an active user.

$r_{(u_a,j)}$ is the rating of active user in the j*th* row and $r_{(u_i,j)}$ is the rating of the i*th* user in the j*th* row.

$dif_{(u_i,j)}$ is the difference of the rating for the i*th* user with the active user in the j*th* row.

$sim_{dif_{(u_i,j)}}$ is the similarity of the i*th* user with the active user in the j*th* row.

$sim_{(u_a,u_i)}$ is the similarity of the user i, with the active user.

N_j is the number of common items.

For example, suppose active user ratings are: {2, nan, 3, nan, 5} and one user ratings are: {3, 4, nan, nan, 1} then for ratings between 1 and 5:

dif={1, nan, nan, nan, 4} and

sim(dif)={$\frac{-1}{5-1} + 1$, nan, nan, nan, $\frac{-4}{5-1} + 1$}={0.75, nan, nan, nan, 0}

and mean of sim(dif) is sim=0.375.

Value

An object of class "simple_similarity", a list with components:

call	The call used.
sim_x	Neighboring user similarity values in descending order.
sim_index	Number of columns for neighboring users in descending order of similarity.

Author(s)

Farimah Houshmand Nanehkaran

Maintainer: Farimah Houshmand Nanehkaran <hoshmandcomputer@gmail.com>

References

Mongia, A., & Majumdar, A. (2019). *Matrix completion on multiple graphs: Application in collaborative filtering*. Signal Processing, vol. 165, pp. 144-148.

Hong, B., & Yu, M. (2019). *A collaborative filtering algorithm based on correlation coefficient*. Neural Computing and Applications, vol. 31, no. 12, pp. 8317-8326.

Examples

```
ratings <- matrix(c(  2,    5,  NaN,  NaN,  NaN,    4,
                    NaN,  NaN,  NaN,    1,  NaN,    5,
                    NaN,    4,    5,  NaN,    4,  NaN,
                      4,  NaN,  NaN,    5,  NaN,  NaN,
                      5,  NaN,    2,  NaN,  NaN,  NaN,
                    NaN,    1,  NaN,    4,    2,  NaN),nrow=6,byrow=TRUE)#items*users

sim <- simple_similarity(ratings, max_score=5, min_score=1, ac=1)
```

Index