Package 'CFilt'

January 20, 2025

Title Recommendation by Collaborative Filtering

Version 0.3.0

Description Provides methods and functions to implement a Recommendation
System based on Collaborative Filtering Methodology.
See Aggarwal (2016) <doi:10.1007/978-3-319-29659-3> for an overview.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

LazyData true

Imports methods

NeedsCompilation no

Author Jessica Kubrusly [aut, cre] (<https://orcid.org/0000-0003-0465-4629>),
Thiago Lima [ctb],
Lucas Oliveira [ctb]

Maintainer Jessica Kubrusly <jessicakubrusly@id.uff.br>

Depends R (>= 3.5.0)

Repository CRAN

Contents

`CF-class`   *Recommendation by Collaborative Filtering*

Description

CF is a class of objects that stores information about a recommendation system. This information includes the consumption or rating of each (user, item) pair in the utility matrix MU, the similarities between each pair of users in the similarity matrix SU, the similarities between each pair of items in the similarity matrix SI, the number of items consumed and/or rated by each user in the vector n_aval_u, the number of users who consumed and/or rated each item in the vector n_aval_i, the average rating value of each user in the vector averages_u, the average rating value received by each item in the vector averages_i, the number of items consumed in common by each pair of users in the matrix Int_U, and the number of users in common for each pair of items in the matrix Int_I. The class contains methods such as addNewUser, addNewEmptyUser, deleteUser, addNewItem, addNewEmptyItem, deleteItem, newRating and deleteRating, which modify the object's structure by altering users, items, or consumption data. The class also includes functions such as kClosestItems, topKUsers, and topKItems, which return items to recommend to a user or users to whom an item should be recommended. An object of the CF class is created using the CFBuilder function.

Fields

MU The Utility Matrix, a matrix that contains all the users' ratings. The rows comprise users and the columns, items.

SU The user similarity matrix.

SI The item similarity matrix

IntU A symmetric matrix that records the number of users in common who consumed each pair of items.

IntI A symmetric matrix that records the number of items in common that have been consumed by each pair of users.

averages_u A vector that contains the averages of users' ratings.

averages_i A vector that contains the averages of items' ratings.

n_aval_u A vector that stores the number of items rated by each user.

n_aval_i A vector that stores the number of users who consumed each item.

datatype A character that indicates the type of data, which can be either "consumption" or "rating".

Methods

addnewemptyitem(Id_i) A method that adds a new item that has not yet been consumed by any existing user in the recommendation system. Id_i: a character, the new item ID; To add more than one new user, lists can be used. Id_i: a list of characters;

addnewemptyuser(Id_u) A method that adds a new user who has not yet consumed any existing items in the recommendation system. Id_u: a character, the new user ID; To add more than one new user, lists can be used. Id_u: a list of characters;

addnewitem(Id_i, Ids_u, r = NULL) A method that adds a new item that has been consumed by already existing users in the recommendation system. Id_i: a character, the new item ID; Ids_u: a character vector, the IDs of the users who consumed the new item; r: a numeric vector, the ratings given by the users for the new item (only for ratings datatype). To add more than one new item, lists can be used. Id_i: a list of characters; Ids_u: a list of characters vectors; r: list of numeric vectors.

addnewuser(Id_u, Ids_i, r = NULL) A method that adds a new user who consumed items already existing in the recommendation system. Id_u: a character, the new user ID; Ids_i: a character vector, the IDs of the items consumed by the user; r: a numeric vector, the ratings of the items consumed by the new user (only for ratings datatype). To add more than one new user, lists can be used. Id_u: a list of characters; Ids_i: a list of characters vectors; r: list of numeric vectors.

changerating(Id_u, Id_i, r = NULL) A method that changes a rating or consumption of a user for an item that has already been rated by them. Id_u: a character, the user ID; Id_i: a character, the item ID; r: a numeric, the rating given by Id_u for Id_i (only for ratings datatype). To change more than one ratings, lists can be used. Id_u: a list of characters; Id_i: a list of characters; r: list of numeric vectors.

deleteitem(Id_i) A method that deletes an item from the recommendation system. Id_i: a character, the item ID; To delete more than one item, lists can be used. Id_i: a list of characters;

deleterating(Id_u, Id_i) A method that deletes a existing rating or consumption of a user for an item. Id_u: a character, the user ID; Id_i: a character, the item ID; To deletes more than one ratings, lists can be used. Id_u: a list of characters; Id_i: a list of characters.

deleteuser(Id_u) A method that deletes an user from the recommendation system. Id_u: a character, the user ID; To delete more than one user, lists can be used. Id_u: a list of characters;

newrating(Id_u, Id_i, r = NULL) A method that adds a new rating or consumption of an existing user for an existing item that had not yet been rated by them. Id_u: a character, the user ID; Id_i: a character, the item ID; r: a numeric, the rating given by Id_u for Id_i (only for ratings datatype). To add more than one new ratings, lists can be used. Id_u: a list of characters; Id_i: a list of characters; r: list of numeric vectors.

Author(s)

Jessica Kubrusly

References

- LINDEN, G.; SMITH, B.; YORK, J. Amazon. com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, v. 7, n. 1, p. 76-80,2003

- Aggarwal, C. C. (2016). Recommender systems (Vol. 1). Cham: Springer International Publishing.

- Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). Mining of massive data sets. Cambridge university press.

See Also

CFbuilder

Examples

```
objectCF_r <- CFbuilder(Data = movies[1:500,], Datatype = "ratings",
similarity = "pearson")
dim(objectCF_r$MU)
colnames(objectCF_r$MU) #movies Id
rownames(objectCF_r$MU) #users Id
dim(objectCF_r$SU)
dim(objectCF_r$SI)
objectCF_r$averages_u
hist(objectCF_r$averages_u)
objectCF_r$averages_i
hist(objectCF_r$averages_i)
objectCF_r$n_aval_u
summary(objectCF_r$n_aval_u)
barplot(table(objectCF_r$n_aval_u))
objectCF_r$n_aval_i
summary(objectCF_r$n_aval_i)
barplot(table(objectCF_r$n_aval_i))
objectCF_r$addnewuser(Id_u = "newuser1",
Ids_i = "The Hunger Games: Catching Fire", r = 5)
rownames(objectCF_r$MU) #users Id
objectCF_r$n_aval_u["newuser1"]
objectCF_r$averages_u["newuser1"]
objectCF_r$addnewuser(Id_u = "newuser2",
Ids_i = c("Frozen","Her","Iron Man 3"),r = c(2,4,3))
rownames(objectCF_r$MU) #users Id
objectCF_r$n_aval_u["newuser2"]
objectCF_r$averages_u["newuser2"]
objectCF_r$addnewuser(Id_u = list("newuser3","newuser4"),
Ids_i = list(c("Lincoln","Monsters University","The Lego Movie","Frozen"),
c("The Wolverine","The Lego Movie")),r = list(c(1,4,5,4),c(4,5)))
rownames(objectCF_r$MU) #users Id
objectCF_r$n_aval_u[c("newuser3","newuser4")]
objectCF_r$averages_u[c("newuser3","newuser4")]
objectCF_r$newrating(Id_u = list("newuser1","newuser1","newuser2","newuser4"),
Id_i = list("The Lego Movie","Wreck-It Ralph","Fast & Furious 6",
"12 Years a Slave"),r = list(4,5,4,2))
objectCF_r$n_aval_u[c("newuser1","newuser2","newuser3","newuser4")]
objectCF_r$averages_u[c("newuser1","newuser2","newuser3","newuser4")]
objectCF_r$addnewitem(Id_i = "Oppenheimer",
Ids_u = c("newuser1","newuser2","newuser3","newuser4","1","2","4","6","10",
"11","20","32"),r = c(1,2,3,1,5,4,5,4,1,3,5,4))
colnames(objectCF_r$MU)
objectCF_r$n_aval_i["Oppenheimer"]
objectCF_r$averages_i["Oppenheimer"]
objectCF_c <- CFbuilder(Data = movies[1:500,-3], Datatype = "consumption",
similarity = "jaccard")
dim(objectCF_c$MU)
colnames(objectCF_c$MU) #movies Id
rownames(objectCF_c$MU) #users Id
dim(objectCF_c$SU)
dim(objectCF_c$SI)
```

```
objectCF_c$averages_u
objectCF_c$averages_i
objectCF_c$n_aval_u
summary(objectCF_c$n_aval_u)
barplot(table(objectCF_c$n_aval_u))
objectCF_c$n_aval_i
summary(objectCF_c$n_aval_i)
barplot(table(objectCF_c$n_aval_i))
objectCF_c$addnewuser(Id_u = "newuser1",
Ids_i = "The Hunger Games: Catching Fire")
rownames(objectCF_c$MU) #users Id
objectCF_c$n_aval_u["newuser1"]
objectCF_c$addnewuser(Id_u = "newuser2",
Ids_i = c("Frozen","Her","Iron Man 3"))
rownames(objectCF_c$MU) #users Id
objectCF_c$n_aval_u["newuser2"]
objectCF_c$addnewuser(Id_u = list("newuser3","newuser4"),Ids_i = list(
c("Lincoln","Monsters University","The Lego Movie","Frozen"),
c("The Wolverine","The Lego Movie")))
rownames(objectCF_c$MU)
objectCF_c$n_aval_u[c("newuser3","newuser4")]
objectCF_c$MU["newuser1","The Lego Movie"]
objectCF_c$newrating(Id_u = list("newuser1","newuser1","newuser2","newuser4"),
Id_i = list("The Lego Movie","Wreck-It Ralph","Fast & Furious 6",
"12 Years a Slave"))
objectCF_c$n_aval_u[c("newuser1","newuser2","newuser3","newuser4")]
objectCF_c$averages_u[c("newuser1","newuser2","newuser3","newuser4")]
objectCF_c$addnewitem(Id_i = "Oppenheimer",
Ids_u = c("newuser1","newuser2","newuser3","newuser4","1","2","4","6","10",
"11","20","32"),r = c(1,2,3,1,5,4,5,4,1,3,5,4))
colnames(objectCF_c$MU)
objectCF_c$n_aval_i["Oppenheimer"]
objectCF_c$averages_i["Oppenheimer"]
```

CFbuilder   *The constructor function of the CFilt class.*

Description

The constructor function of the CFilt class.

Usage

```
CFbuilder(Data,Datatype,similarity)

CFbuilder(
Data,
Datatype = ifelse(ncol(Data)==2,"consumption","ratings"),
similarity = ifelse(Datatype == "consumption","jaccard","pearson")
)
```

Arguments

`Data`	a dataframe with 2 or 3 columns. The first column indicates the user ID, the second the item ID and the third the rating (only if Datatype = 'rating').
`Datatype`	a character that indicates the data type: 'rating' or 'consumption'.
`similarity`	a character that indicates the similarity type. For 'datatype='ratings', 'cossine' or 'person'. For datatype='consumption', 'jaccard'.

Value

a CF class object.

Author(s)

Jessica Kubrusly

References

LINDEN, G.; SMITH, B.; YORK, J. Amazon. com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, v. 7, n. 1, p. 76-80,2003

See Also

CF-class

Examples

```
CF1 <- CFbuilder(Data = movies[1:300,], Datatype = "ratings",
similarity = "pearson") #or
CF1_ <- CFbuilder(Data = movies[1:300,])
CF2 <- CFbuilder(Data = movies[1:300,], Datatype = "ratings",
similarity = "cosine") #or
CF2_ <- CFbuilder(Data = movies[1:300,], similarity = "cosine")
CF3 <- CFbuilder(Data = movies[1:300,-3], Datatype = "consumption",
similarity = "jaccard") #or
CF3_ <- CFbuilder(Data = movies[1:300,-3])
```

---

cosine	*Similarity calculation Function*

Description

Functions that returns the cosine similarity between two items or users.

Usage

```
cosine(CF, type, i, j)
```

Arguments

CF	A CF objec
type	"user" or "item"
i	"user" or "item" Id or index
j	"user" or "item" Id or index#'

Author(s)

Jessica Kubrusly

Examples

```
objectCF_r <- CFbuilder(Data = movies[1:500,], Datatype = "ratings",
similarity = "cosine")
cosine(CF=objectCF_r,type = "user",i="1",j="2")
cosine(CF=objectCF_r,type = "item",i="Her",j="Frozen")
```

---

estimaterating	*Recommendation Functions*

Description

Function that provide an estimate of the user's rating for the item.

Usage

```
estimaterating(
  CF,
  Id_u,
  Id_i,
  type = "user",
  neighbors = ifelse(type == "user", nrow(CF$MU) - 1, ncol(CF$MU) - 1)
)
```

Arguments

CF	A CF object
Id_u	the user Id
Id_i	the item Id
type	"user" or "item"
neighbors	number of neighbors in the calculation.

Author(s)

Jessica Kubrusly

Examples

```
objectCF_r <- CFbuilder(Data = movies[1:500,], Datatype = "ratings",
similarity = "cosine")
estimaterating(CF=objectCF_r,Id_u="35",Id_i="Despicable Me 2")
estimaterating(CF=objectCF_r,Id_u="35",Id_i="Her")
```

jaccard   *Similarity calculation Function*

Description

Functions that returns the Jaccard similarity between two items or users.

Usage

```
jaccard(CF, type, i, j)
```

Arguments

CF	A CF objec
type	"user" or "item"
i	"user" or "item" Id or index
j	"user" or "item" Id or index#'

Author(s)

Jessica Kubrusly

Examples

```
objectCF_r <- CFbuilder(Data = movies[1:500,c(1,2)], Datatype = "consumption",
similarity = "jaccard")
jaccard(CF=objectCF_r,type = "user",i="1",j="2")
jaccard(CF=objectCF_r,type = "item",i="Her",j="Frozen")
```

---

kclosestitems   *Recommendation Functions*

Description

Functions that provide items to be recommended to system users.

Usage

```
kclosestitems(CF, Id_i, k = 10)
```

Arguments

CF	A CF objec
Id_i	the item Id
k	an integer

Author(s)

Jessica Kubrusly

Examples

```
objectCF_r <- CFbuilder(Data = movies[1:500,], Datatype = "ratings",
similarity = "pearson")
kclosestitems(CF = objectCF_r, Id_i = "The Lego Movie")
kclosestitems(CF = objectCF_r, Id_i = "Lincoln", k=5)
```

---

movies   *Movie ratings by users*

Description

A dataset containing 7276 ratings for 50 movies by 526 users. This database was created by Giglio (2014).

Usage

```
movies
```

Format

A data frame with 7276 rows and 3 variables:

Id Users Users identifier. Numbers 1 to 526.

Id Items Movies identifier. Movies list:

1. Iron Man 3
2. Despicable Me 2
3. My Mom Is a Character
4. Fast & Furious 6
5. The Wolverine
6. Thor: The Dark World
7. Hansel & Gretel: Witch Hunters
8. Wreck-It Ralph
9. Monsters University
10. The Hangover Part III
11. Vai Que Dá Certo
12. Meu Passado me Condena
13. We're So Young
14. Brazilian Western
15. O Concurso
16. Mato sem Cachorro
17. Cine Holliudy
18. Odeio o Dia dos Namorados
19. Argo
20. Django Unchained
21. Life of Pi
22. Lincoln
23. Zero Dark Thirty
24. Les Miserables
25. Silver Linings Playbook
26. Beasts of the Southern Wild
27. Amour
28. A Royal Affair
29. American Hustle
30. Capitain Phillips
31. 12 Years a Slave
32. Dallas Buyers Club
33. Gravity
34. Her
35. Philomena
36. The Wolf of Wall Street
37. The Hunt

38. Frozen
39. Till Luck Do Us Part 2
40. Muita Calma Nessa Hora 2
41. Paranormal Activity: The Marked Ones
42. I, Frankenstein,
43. The Legend of Tarzan
44. The Book Thief
45. The Lego Movie, , ,
46. Walking With Dinosaurs
47. The Hunger Games: Catching Fire
48. Blue Is The Warmest Color
49. Reaching for the Moon
50. The Hobbit: The Desolation of Smaug

Ratings Movie ratings by users. The ratings follows the Likert scale: 1 to 5.

---

pearson	*Similarity calculation Function*

Description

Functions that returns the pearson similarity between two items or users.

Usage

```
pearson(CF, type, i, j)
```

Arguments

CF	A CF objec
type	"user" or "item"
i	"user" or "item" Id or index
j	"user" or "item" Id or index#'

Author(s)

Jessica Kubrusly

Examples

```
objectCF_r <- CFbuilder(Data = movies[1:500,], Datatype = "ratings",
similarity = "pearson")
pearson(CF=objectCF_r,type = "user",i="2",j="3")
pearson(CF=objectCF_r,type = "item",i="Her",j="Frozen")
```

---

topkitems   *Recommendation Function*

Description

Functions that provide the top k items to be recommended to the user Id_u.

Usage

```
topkitems(CF, Id_u, k = 10, type = "user")
```

Arguments

CF	A CF objec
Id_u	the user Id
k	an integer
type	"user" or "item"

Author(s)

Jessica Kubrusly

Examples

```
objectCF_r <- CFbuilder(Data = movies[1:500,], Datatype = "ratings",
similarity = "pearson")
u1 = rownames(objectCF_r$MU)[1]
topkitems(CF=objectCF_r,Id_u = u1)
u2 = rownames(objectCF_r$MU)[2]
topkitems(CF=objectCF_r,Id_u = u2)
```

---

topkusers   *Recommendation Function*

Description

Functions that provide the top k users to recommend the item Id_i.

Usage

```
topkusers(CF, Id_i, k = 10, type = "user")
```

Arguments

CF	A CF objec
Id_i	the item Id
k	an integer
type	"user" or "item"

Author(s)

Jessica Kubrusly

Examples

```
objectCF_r <- CFbuilder(Data = movies[1:500,], Datatype = "ratings",
similarity = "pearson")
colnames(objectCF_r$MU)
topkusers(CF = objectCF_r, Id_i = "The Lego Movie")
topkusers(CF = objectCF_r, Id_i = "Her")
```

Index