

Package ‘BNMF’

October 7, 2017

Type Package

Title Package of the Non Negative Matrix Factorization Model

Version 1.0

Date 2017-05-27

Author Priscila Valdiviezo <pmvaldiviezo@utpl.edu.ec>, Jesus Bobadilla <jesus.bobadilla@upm.es>, Sergio Gil <sergio.gil.borras@alumnos.upm.es>, Antonio Hernando <ahernando@upm.es>

Maintainer Priscila Valdiviez <pmvaldiviezo@utpl.edu.ec>

Description This Package is used for the recommendation of items in Collaborative Filtering

License GPLv2

Suggests knitr

VignetteBuilder knitr

R topics documented:

BNMF-package	1
BNMF	3
computeLearningParameters	3
computeMatrixPredictions	4
computeOuput	5
initializeModel	5
loadData	6
mae	6
PrecisionRecall	7
splitData	7
Index	8

BNMF-package	<i>BNMF: Bayesian probabilistic model of non-negative factorization for collaborative filtering</i>
--------------	---

Description

Algorithm for predicting the tastes of users in recommender systems based on a Bayesian probabilistic model of non-negative factorization for collaborative filtering. BNMF is based on factorizing the rating matrix into two non negative matrices.

Usage

```
BNMF(iter,R,k,alpha,eta)
```

Arguments

The input of the algorithm is a matrix of ratings R, and some parameters as the following:

iter: setting number of iterations
k: number of latent factors (or number of groups)
R: matrix of ratings (user x items)
alpha: control of group overlap
eta: evidence that a group of users likes an item

Details

Package: BNMF
Type: Package
Version: 1.0
Date: 2017-05-27
License: GPLv2

This package use a matrix of ratings R and three parameters of setting, k, alpha, and eta.

Author(s)

Priscila Valdiviezo <pmvaldiviezo@utpl.edu.ec>, Antonio Hernando <ahernando@upm.es>, Jesus Bobadilla

Maintainer: Priscila Valdiviezo <pmvaldiviezo@utpl.edu.ec>

References

Hernando, A., Bobadilla, J., & Ortega, F. (2016). A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. Knowledge-Based Systems, 97. <http://doi.org/10.1016/j.knosys.2015.12.018>

Examples

```
library(BNMF)
#loading dataset of training and testing
data(ratings)
dim(R)
#System setting parameters
k<-6 #Number of groups (latent factors)
alpha<-0.8 #Control of group overlap
eta<-5 #Evidence that a group of users likes an item
iter<-20 #setting number of iterations
output<-BNMF(iter,R,k,alpha,eta)
#Matrix associated to users
output$au.k
#Matrix associated to items
```

```

output$bk.i
#Predictions of the ratings
predictions<-output$pred
predictions

####Prediction Accuracy
mae(R.tst,predictions)
####Precision/Recall
vectPredictions <- c(5,10,20,40)
PrecisionRecall(predictions,vectPredictions)

```

BNMF	<i>Main function which invoke others functions for the run of the algorithm</i>
------	---

Description

Main function which invoke others functions for the run of the algorithm

Usage

```
BNMF(steps, R, k, alpha, eta)
```

Arguments

steps	number iterations
R	matrix of ratings
k	number of latent factors
alpha	for learning the algorithm
eta	f or learning the algorithm

Value

matrix outR au.k bk.i

computeLearningParameters	<i>Functions of BNMF model.</i>
---------------------------	---------------------------------

Description

Main functions used to the learning of parameters of the model

Usage

```
computeLearningParameters(m.pos, m.neg, gamma, sum.gam, e.pos, e.neg, N, M)
```

Arguments

m.pos	positive matrix
m.neg	negative matrix
gamma	gamma matrix
sum.gam	Summation of gammas
e.pos	positive epsilons matrix
e.neg	negative epsilons matrix
N	number of users
M	number of items

Value

matrix lambda

Examples

```
computeLearningParameters<-function(m.pos,m.neg,gamma,sum.gam,e.pos,e.neg,N,M)
```

```
computeMatrixPredictions
```

Predictions

Description

Compute the prediction the tastes of users: pu,i

Usage

```
computeMatrixPredictions(k, au.k, bk.i, R)
```

Arguments

k	number of groups (latent factors)
au.k	matrix associated to users
bk.i	matrix associated to items
R	ratings matrix

Value

matrix outP outR

Examples

```
computeMatrixPredictions<-function(k,au.k,bk.i,R)
```

computeOuput

Output Matrices

Description

Compute the Output Matrices of the algorithm

Usage

```
computeOuput(k, gamma, e.positive, e.negative, N, M)
```

Arguments

k	number of groups (latent factors)
gamma	gamma matrix
e.positive	positive epsilons matrix
e.negative	negative epsilons matrix
N	number of users
M	number of items

Value

matrix a b

Examples

```
computeOuput<-function(k,gamma,e.positive, e.negative,N,M)
```

initializeModel

Initialization of parameters of the model.

Description

Utilities of BNMF software required to initialize parameters of the model.

Usage

```
initializeModel(k, R)
```

Arguments

k	number of groups (latent factors)
R	ratings matrix

Value

matrix e.pos e.neg m.pos m.neg l

Examples

```
initializeModel<-function (k,R)
```

loadData	<i>Loading the dataset</i>
----------	----------------------------

Description

Alternative code to download the dataset from the official site

- UserID: numeric
- MovieID: numeric
- Rating: rating, numeric in a scale from 1-5
- Timestamp: date

Usage

```
loadData()
```

Source

<https://grouplens.org/datasets/movielens/> ,

mae	<i>Evaluation of Predictions</i>
-----	----------------------------------

Description

Function to evaluate predictions

Usage

```
mae(R.tst, pred)
```

Arguments

R.tst	matrix of test
pred	matrix of predictions

Examples

```
mae(R.tst,output$pred)
```

PrecisionRecall	<i>Compute Precision and Recall</i>
-----------------	-------------------------------------

Description

Functions to evaluate recommendations

Usage

```
PrecisionRecall(predictions, vectPredictions, ratings.tst)
```

Arguments

predictions	matrix of predictions
vectPredictions	vector TopN of recommendations
ratings.tst	dataframe of test ratings

Examples

```
vectPredictions <- c(5,10,20,40)
PrecisionRecall(output$pred,vectPredictions,ratings.tst)
```

splitData	<i>Compute a Random partitioning from input data</i>
-----------	--

Description

Function to determine a random partition of the input dataset

Usage

```
splitData(pctSplit, ratings.dat)
```

Arguments

pctSplit	training percentage
ratings.dat	dataset (format dataframe)

Examples

```
data(MovieLensLatest)
ratings.dat<-MovieLensLatest
pctSplit<-0.7 #for test
splitData(pctSplit,ratings.dat)
```

Index

BNMF, [3](#)
BNMF (BNMF-package), [1](#)
BNMF-package, [1](#)

computeLearningParameters, [3](#)
computeMatrixPredictions, [4](#)
computeOuput, [5](#)

initializeModel, [5](#)

loadData, [6](#)

mae, [6](#)

PrecisionRecall, [7](#)

splitData, [7](#)