

Package ‘recosystem’

September 11, 2014

Type Package

Title Recommender System using Matrix Factorization

Version 0.2.3

Date 2014-09-08

Maintainer Yixuan Qiu <yixuan.qiu@cos.name>

Description This package is an R wrapper of the libmf library (<http://www.csie.ntu.edu.tw/~cjlin/libmf/>) for recommender system using matrix factorization. It is typically used to approximate an incomplete matrix using the product of two matrices in a latent space. Other common names for this task include “collaborative filtering”, “matrix completion”, “matrix recovery”, etc.

License BSD_3_clause + file LICENSE

Copyright see file COPYRIGHTS

URL <https://github.com/yixuan/recosystem>

BugReports <https://github.com/yixuan/recosystem/issues>

SystemRequirements C++11

Depends methods

Imports Rcpp (>= 0.11.0)

Suggests knitr

LinkingTo Rcpp

VignetteBuilder knitr

Author Yixuan Qiu [aut, cre], Chih-Jen Lin [aut, cph] (original C++ code), Yu-Chin Juan [aut, cph] (original C++ code), Yong Zhuang [aut, cph] (original C++ code), Wei-Sheng Chin [aut, cph] (original C++ code)

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-09-11 10:00:26

R topics documented:

convert	2
predict	3
Reco	4
train	5

Index	7
--------------	----------

convert	<i>Read data file and convert to binary format</i>
---------	--

Description

These methods are member functions of class "RecoSys" that convert training and testing data files into binary format. The conversion is a preprocessing step prior to the model training part, since data with this binary format could be accessed more efficiently.

The common usage of these methods is

```
r = Reco()
r$convert_train(rawfile, outdir, verbose = TRUE)
r$convert_test(rawfile, outdir, verbose = TRUE)
```

Arguments

r	Object returned by <code>Reco()</code>
rawfile	Path of data file, see section 'Data format' for details
outdir	Directory in which the output binary file will be generated. If missing, <code>tempdir()</code> will be used.
verbose	Whether to show detailed information. Default is TRUE.

Data format

The data file required by these methods takes the format of sparse matrix in triplet form, i.e., each line in the file contains three numbers

```
row col value
```

representing a number in the rating matrix with its location. In real applications, it typically looks like

```
user_id item_id rating
```

NOTE: row and col start from 0. So if the first user rates 3 on the first item, the line will be

```
0 0 3
```

NOTE: For testing data, the file also needs to contain three numbers each line. If the rating values are unknown, you can put any number as placeholders.

Example data files are contained in the `recoSystem/dat` directory.

Author(s)

Yixuan Qiu <<http://statr.me>>

References

LIBMF: A Matrix-factorization Library for Recommender Systems. <http://www.csie.ntu.edu.tw/~cjlin/libmf/>

Y. Zhuang, W.-S. Chin, Y.-C. Juan, and C.-J. Lin. A Fast Parallel Stochastic Gradient Method for Matrix Factorization in Shared Memory Systems. Technical report 2014.

See Also

[train](#), [predict](#)

Examples

```
trainset = system.file("dat", "smalltrain.txt", package = "recoSystem")
testset = system.file("dat", "smalltest.txt", package = "recoSystem")
r = Reco()
r$convert_train(trainset)
r$convert_test(testset)
print(r)
```

predict

Recommender model predictions

Description

This method is a member function of class "RecoSys" that predicts unknown entries in the rating matrix. Prior to calling this method, model needs to be trained by calling `$train()`, and testing data also must be set through `$convert_test()`. Prediction results will be written into the specified file, one value per line, corresponding to the testing data.

The common usage of this method is

```
r = Reco()
r$predict(outfile, verbose = TRUE)
```

Arguments

r	Object returned by <code>Reco()</code>
outfile	Name of the output file for prediction
verbose	Whether to show detailed information. Default is TRUE.

Author(s)

Yixuan Qiu <<http://statr.me>>

References

LIBMF: A Matrix-factorization Library for Recommender Systems. <http://www.csie.ntu.edu.tw/~cjlin/libmf/>

Y. Zhuang, W.-S. Chin, Y.-C. Juan, and C.-J. Lin. A Fast Parallel Stochastic Gradient Method for Matrix Factorization in Shared Memory Systems. Technical report 2014.

See Also

[convert](#), [train](#)

Examples

```
set.seed(123) # this is a randomized algorithm
trainset = system.file("dat", "smalltrain.txt", package = "recoSystem")
testset = system.file("dat", "smalltest.txt", package = "recoSystem")
r = Reco()
r$convert_train(trainset)
r$convert_test(testset)
r$train(opts = list(dim = 100, niter = 100,
                    cost.p = 0.001, cost.q = 0.001))
outfile = tempfile()
r$predict(outfile)

## Compare the first few true values of testing data
## with predicted ones
print(read.table(testset, header = FALSE, sep = " ", nrows = 10)$V3)
print(scan(outfile, n = 10))
```

Reco

Construct a recommender system object

Description

This function simply returns an object of class "RecoSys" that can be used to construct recommender model and conduct prediction.

Usage

```
Reco()
```

Value

Reco() returns an object of class "RecoSys" equipped with methods `$convert_train()`, `$convert_test()`, `$train()` and `$predict()`, which describe the typical process of reading data, building model and predicting results. See their help documents for details.

Author(s)

Yixuan Qiu <<http://statr.me>>

References

LIBMF: A Matrix-factorization Library for Recommender Systems. <http://www.csie.ntu.edu.tw/~cjlin/libmf/>

Y. Zhuang, W.-S. Chin, Y.-C. Juan, and C.-J. Lin. A Fast Parallel Stochastic Gradient Method for Matrix Factorization in Shared Memory Systems. Technical report 2014.

See Also

[convert](#), [train](#), [predict](#)

train	<i>Train a recommender model</i>
-------	----------------------------------

Description

This method is a member function of class "RecoSys" that trains a recommender model. It will create a model file in the specified directory, containing necessary information for prediction. Training data must have already been converted into binary form through `$convert_train()` before calling this method.

The common usage of this method is

```
r = Reco()
r$train(outdir, opts, verbose = TRUE)
```

Arguments

r	Object returned by <code>Reco()</code>
outdir	Directory in which the model file will be generated. If missing, <code>tempdir()</code> will be used.
opts	Various options and tuning parameters in the model training procedure. See section Options and Parameters for details.
verbose	Whether to show detailed information. Default is TRUE.

Options and Parameters

The `opts` argument is a list that can supply any of the following parameters:

`dim` Integer, the width of the factorized matrix, i.e., the number of latent factors. Default is 40.

`niter` Integer, the number of iterations. Default is 40.

`nthread` Integer, the number of threads for parallel computing. Default is 1.

`cost.p` Nonnegative real number, the regularization cost for P. Default is 1.

`cost.q` Nonnegative real number, the regularization cost for Q. Default is 1.

`cost.ub` Real number, the regularization cost for user bias. Set <0 to disable. Default is -1.

`cost.ib` Real number, The regularization cost for item bias. Set <0 to disable. Default is -1.

gamma Positive real number, the learning rate for parallel SGD. Default is 0.001.
blocks Integer vector of length 2, the number of blocks for parallel SGD. Default is $c(2 * nthread,$
rand_shuffle Logical, whether to enable random shuffle. This should be enabled when data are
 imbalanced. Default is TRUE.
show_tr_rmse Logical, whether to show RMSE on training data. Default is FALSE.
show_obj Logical, whether to show the objective value. This option may slow down the training
 procedure. Default is FALSE.
use_avg Logical, whether to use training data average. Default is FALSE.

Author(s)

Yixuan Qiu <<http://statr.me>>

References

LIBMF: A Matrix-factorization Library for Recommender Systems. <http://www.csie.ntu.edu.tw/~cjlin/libmf/>
 Y. Zhuang, W.-S. Chin, Y.-C. Juan, and C.-J. Lin. A Fast Parallel Stochastic Gradient Method for Matrix Factorization in Shared Memory Systems. Technical report 2014.

See Also

[convert](#), [predict](#)

Examples

```

set.seed(123) # this is a randomized algorithm
trainset = system.file("dat", "smalltrain.txt", package = "recoSystem")
testset = system.file("dat", "smalltest.txt", package = "recoSystem")
r = Reco()
r$convert_train(trainset)
r$convert_test(testset)
r$train(opts = list(dim = 80, cost.p = 0.01, cost.q = 0.01))
print(r)

```

Index

*Topic **models**

Reco, 4

convert, 2, 4–6

convert_test, 3

convert_test (convert), 2

convert_train, 4, 5

convert_train (convert), 2

predict, 3, 3, 4–6

Reco, 2, 3, 4, 5

train, 3–5, 5