Package 'poismf'

September 5, 2019

Type Package

Title Factorization of Sparse Counts Matrices Through Poisson Likelihood

Version 0.1.3

Date 2019-09-04

Author David Cortes

Maintainer David Cortes <david.cortes.rivera@gmail.com>

Description Creates a low-rank factorization of a sparse counts matrix by maximizing Poisson likelihood with l1/l2 regularization
with all non-negative latent factors (e.g. for recommender systems or topic modeling) (Cortes, David, 2018, <arXiv:1811.01908>).
Similar to hierarchical Poisson factorization, but follows an optimization-based approach with regularization instead of a hierarchical
structure, and is fit through either proximal gradient or conjugate gradient instead of variational inference.

License BSD_2_clause + file LICENSE

Imports Rcpp (>= 0.12.19), Matrix, SparseM, methods, nonneg.cg

LinkingTo Rcpp

RoxygenNote 6.1.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-09-05 02:20:06 UTC

R topics documented:

poismf 2
predict.poismf 3
predict_all 5
print.poismf 5
summary.poismf 6

Index **7**

poismf *Factorization of Sparse Counts Matrices through Poisson Likelihood*

Description

Creates a low-rank non-negative factorization of a sparse counts matrix by maximizing Poisson likelihood with L1/L2 regularization, using optimization routines based on proximal gradient iteration.

Usage

```
poismf(X, k = 50, l1_reg = 0, l2_reg = 1e+09, niter = 10,
  nupd = 1, step_size = 1e-07, init_type = "gamma", seed = 1,
  nthreads = -1)
```

Arguments

X	The matrix to factorize. Can be: a) a 'data.frame' with 3 columns, containing in this order: row index (starting at 1), column index, count value (the indices can also be character type, in wich case it will enumerate them internally, and will return those same characters from 'predict'); b) A sparse matrix in COO format from the 'SparseM' package; c) a full matrix (of class 'matrix' or 'Matrix::dgTMatrix'); d) a sparse matrix from package 'Matrix' in triplets format.
k	Dimensionality of the factorization (a.k.a. number of latent factors).
l1_reg	Strength of the l1 regularization
l2_reg	Strength of the l2 regularization.
niter	Number of iterations to run.
nupd	Number of updates per iteration.
step_size	Initial step size to use (proximal gradient only). Will be decreased by 1/2 after each iteration.
init_type	One of "gamma" or "uniform" (How to intialize the factorizing matrices).
seed	Random seed to use for starting the factorizing matrices.
nthreads	Number of parallel threads to use. Passing a negative number will use the maximum available number of threads

Value

An object of class 'poismf' with the following fields of interest:

Fields

A : the user/document/row-factor matrix (as a vector, has to be reshaped to (nrows, k)).

B : the item/word/column-factor matrix (as a vector, has to be reshaped to (ncols, k)).

References

Cortes, David. "Fast Non-Bayesian Poisson Factorization for Implicit-Feedback Recommendations." arXiv preprint arXiv:1811.01908 (2018).

See Also

predict.poismf predict_all

Examples

```
library(poismf)

### create a random sparse data frame in COO format
nrow <- 10 ** 2
ncol <- 10 ** 3
nnz  <- 10 ** 4
set.seed(1)
X <- data.frame(
    row_ix = as.integer(runif(nnz, min = 1, max = nrow)),
    col_ix = as.integer(runif(nnz, min = 1, max = ncol)),
    count = rpois(nnz, 1) + 1)
X <- X[!duplicated(X[, c("row_ix", "col_ix")]), ]

### factorize the randomly-generated sparse matrix
model <- poismf(X, nthreads = 1)

### predict functionality
predict(model, 1, 10) ## predict entry (1, 10)
predict(model, 1, topN = 10) ## predict top-10 entries "B" for row 1 of "A".
predict(model, c(1, 1, 1), c(4, 5, 6)) ## predict entries [1,4], [1,5], [1,6]
head(predict(model, 1)) ## predict the whole row 1

#all predictions for new row/user/doc
head(predict(model, data.frame(col_ix = c(1,2,3), count = c(4,5,6)) ))
```

`predict.poismf` *Make predictions for arbitrary entries in matrix*

Description

Make predictions for arbitrary entries in matrix

Usage

```
## S3 method for class 'poismf'
predict(object, a, b = NULL, seed = 10, topN = NULL,
  l2_reg = 1000, l1_reg = 0, ...)
```

Arguments

object	An object of class "poismf" as returned by function ‘poismf‘.
a	Row(s) for which to predict. Alternatively, a ‘data.frame‘ (first column being the column indices and second column being the count values) or ‘sparseVector‘ (from package ‘Matrix‘) of counts for one row/user/document, from which predictions will be calculated by producing latent factors on-the-fly.
b	Column(s) for which to predict. If ‘NULL‘, will make predictions for all columns. Otherwise, it must be of the same length as "a", and the output will contain the prediction for each combination of "a" and "b" passed here (unless passing ‘topN‘).
seed	Random seed to use to initialize factors (when ‘a‘ is a ‘data.frame‘ or ‘sparseVector‘)
topN	Return top-N ranked items (columns or IDs from "B") according to their predictions. If passing argument "b", will return the top-N only among those.
l2_reg	When passing to argument ‘a‘ a ‘data.frame‘ or ‘sparseVector‘ and the new factors needs to the calculated on-the-fly, it indicates the L2 regularization strenght to use. Note that in this case, the new factors are optimized through a conjugate-gradient routine, which works better with smaller regulatization values than the proximal-gradient routine used to fit the model.
l1_reg	L1 regularization to use in the same case as above.
...	Not used.

See Also

poismf predict_all

Examples

```
library(poismf)

### create a random sparse data frame in COO format
nrow <- 10 ** 2
ncol <- 10 ** 3
nnz  <- 10 ** 4
set.seed(1)
X <- data.frame(
    row_ix = as.integer(runif(nnz, min = 1, max = nrow)),
    col_ix = as.integer(runif(nnz, min = 1, max = ncol)),
    count = rpois(nnz, 1) + 1)
X <- X[!duplicated(X[, c("row_ix", "col_ix")]), ]

### factorize the randomly-generated sparse matrix
model <- poismf(X, nthreads = 1)

### predict functionality
predict(model, 1, 10) ## predict entry (1, 10)
predict(model, 1, topN = 10) ## predict top-10 entries "B" for row 1 of "A".
predict(model, c(1, 1, 1), c(4, 5, 6)) ## predict entries [1,4], [1,5], [1,6]
```

```
head(predict(model, 1)) ## predict the whole row 1

#all predictions for new row/user/doc
head(predict(model, data.frame(col_ix = c(1,2,3), count = c(4,5,6)) ))
```

`predict_all` *Predict whole input matrix*

Description

Outputs the predictions for the whole input matrix to which the model was fit. Note that this will be a dense matrix, and in typical recommender systems scenarios will likely not fit in memory.

Usage

```
predict_all(model)
```

Arguments

`model`	A Poisson factorization model as output byfunction ‘poismf‘.

Value

A matrix A x B with the full predictions for all rows and column. If the inputs did not have numbers as IDs, the equivalences to their IDs in the outputs are in the ‘poismf‘ object under fields ‘levels_A‘ and ‘levels_B‘.

`print.poismf` *Get information about poismf object*

Description

Print basic properties of a "poismf" object.

Usage

```
## S3 method for class 'poismf'
print(x, ...)
```

Arguments

`x`	An object of class "poismf" as returned by function "poismf".
`...`	Extra arguments (not used).

summary.poismf *Get information about poismf object*

Description

Print basic properties of a "poismf" object (same as ‘print.poismf‘ function).

Usage

```
## S3 method for class 'poismf'
summary(object, ...)
```

Arguments

object	An object of class "poismf" as returned by function "poismf".
...	Extra arguments (not used).

See Also

print.poismf

Index

poismf, 2, *4*
predict.poismf, *3*, 3
predict_all, *3*, *4*, 5
print.poismf, 5, *6*

summary.poismf, 6