Package ‘ptf’

June 15, 2021

Type Package

Title Probit Tensor Factorization

Version 0.0.1

Date 2021-06-13

Author Ye Liu

Maintainer Ye Liu <yliu87@ncsu.edu>

Description Efficient algorithms to implement Probit Tensor Factorization (PTF) model for statistical relational learning, which not only inherits the computation efficiency from the classic tensor factorization model but also accounts for the binary nature of relational data. The methodology is based on Ye Liu (2021) <https://repository.lib.ncsu.edu/bitstream/handle/1840.20/37507/etd.pdf?sequence=1> "Computational Methods for Complex Models with Latent Structure".

License GPL

Imports Rcpp (>= 0.12.9), Matrix(>= 1.2), rARPACK (>= 0.11), plyr (>= 1.8.4)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-06-15 14:50:15 UTC

R topics documented:

 package-ptf ................. 2
 ptf .......................... 2
 ptf_sparse .................. 3
 Rescal ....................... 4

Index 6
package: ptf
probit tensor factorization

Description

Package: ptf
Type: Package
Version: 0.0.1
Date: 2020-09-30
License: GPL
LazyLoad: yes
LazyData: yes

Author(s)
Ye Liu Maintainer: Ye Liu <yeliu.nikki@gmail.com>

ptf

Fit a Probit Tensor Factorization Model

Description

Fit a Probit Tensor Factorization Model

Usage

ptf(X, k, n, r = 0, max_iter = 1000, tol = 1e-08, tol_M = 1e-05,
    iter_M_max = 2, print_option = TRUE)

Arguments

X response data, which is a three-way array of size n by n by k
k number of relations
n number of entities
r decomposition rank
max_iter max number of iterations
tol tolerance of absolute change in likelihood
tol_M tolerance of absolute change in the M step
iter_M_max max number of iterations for M step
print_option whether print loss for each iteration or not
**ptf_sparse**

*Fit a Probit RESCAL model (sparse representation).*

**Value**

fitted parameters

**References**

@references Ye Liu, 2021. Computational Methods for Complex Models with Latent Structure. PhD thesis with link at https://repository.lib.ncsu.edu/bitstream/handle/1840.20/37507/etd.pdf?sequence=1

**Examples**

```r
n <- 20
k <- 10
r <- 3
p <- c(n, n, k)
X <- array(rnorm(prod(p)),dim=p)
X_binary <- ifelse(X < -1.5,1,0)
X_binary_with_missing <- X_binary
num_missing <- 200
missing_index <- data.frame(x1=sample(1:n,num_missing,replace=TRUE),
x2=sample(1:n,num_missing,replace=TRUE),
x3=sample(1:k,num_missing,replace=TRUE))
for(i in 1:num_missing){
  X_binary[missing_index[i,1],
            missing_index[i,2],
            missing_index[i,3]] <- NA
}
result <- ptf(X_binary_with_missing,k,n,r,print_option=FALSE)
```

**Description**

Fit a Probit RESCAL model (sparse representation).

**Usage**

```r
ptf_sparse(df, n, k, r = 0, max_iter = 500, tol = 1e-08,
tol_M = 1e-05, iter_M_max = 2, print_option = TRUE)
```

**Arguments**

- **df**
  a four-column dataframe with columns (1) ent1 (2) ent2 (3) relation (4) true, which is an indicator of whether there is such a relation from ent1 to ent2
- **n**
  number of entities
- **k**
  number of relations
- **r**
  decomposition rank
Rescal

Description

RESCAL Model

Usage

Rescal(X, k, n, r, max_iter = 500)
Rescal

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>response data, which is a three-way array of size n by n by k</td>
</tr>
<tr>
<td>k</td>
<td>number of relations</td>
</tr>
<tr>
<td>n</td>
<td>number of entities</td>
</tr>
<tr>
<td>r</td>
<td>decomposition rank</td>
</tr>
<tr>
<td>max_iter</td>
<td>max number of iterations</td>
</tr>
</tbody>
</table>

Value

fitted parameters

References

Index

* factorization
  package-ptf, 2
* model
  package-ptf, 2
* probit
  package-ptf, 2
* tensor
  package-ptf, 2

package-ptf, 2
package-ptf-package (package-ptf), 2
ptf, 2
ptf-package (package-ptf), 2
ptf_sparse, 3

Rescal, 4