The Journal of Open Source Software

dcTensor: An R package for discrete matrix/tensor decomposition

Koki Tsuyuzaki[1, 2]

1 Department of Artificial Intelligence Medicine, Graduate School of Medicine, Chiba University, Japan **2** Laboratory for Bioinformatics Research, RIKEN Center for Biosystems Dynamics Research, Japan

DOI: DOIunavailable

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Reviewers:

- @Pending Reviewers

Submitted: N/A
Published: N/A

Summary

Non-negative matrix factorization (NMF) is a widely used algorithm to decompose non-negative matrix data into factor matrices. Due to the interpretability of its non-negativity and the convenience of using decomposition results as clustering, there are many applications of NMF in image processing, audio processing, and bioinformatics (CICHOCK, 2009).

We can consider the discrete version of NMF by imposing a binary solution ($\{0,1\}$) or ternary solution ($\{0,1,2\}$) for the factor matrices. Here, we will refer to these as binary matrix factorization (BMF) or ternary matrix factorization (TMF). It is also possible to apply such a constraint to only one of the two factor matrices, and we will call these (semi-) binary matrix factorization (SBMF) and (semi-) ternary matrix factorization (STMF).

A BMF algorithm was first proposed by Zhang, Z. et al. in 2007 (Z. et al. Zhang, 2007), and the algorithm is based on a binary regularization against two factor matrices. Although their work focused on only BMF, the formulization can be applied to TMF, SBMF, or STMF. Besides, there is a growing demand to apply such (semi-) binary or (semi-) ternary decomposition to more heterogeneous non-negative data such as multiple matrices and tensors (high-dimensional arrays), which are higher-order data structures than matrices (CICHOCK, 2009). To meet these requirements, I originally developed `dcTensor`, which is an R/CRAN package to perform some discrete matrix/tensor decomposition algorithms (https://cran.r-project.org/web/packages/dcTensor/index.html).

Statement of need

There is no package to perform (semi-) binary or (semi-) ternary matrix/tensor decomposition. We originally implemented such discrete matrix/tensor decomposition algorithms in R language, which is one of the popular open-source programming languages.

`dcTensor` provides the matrix/tensor decomposition functions as follows:

- `dNMF`: Discretized Non-negative Matrix Factorization (CICHOCK, 2009; Lee & Seung, 1999)
- `dSVD`: Discretized Gradient Descent Singular Value Decomposition (Tsuyuzaki, 2020)
- `dsiNMF`: Discretized Simultaneous Non-negative Matrix Factorization (Badea, 2008; CICHOCK, 2009; Yilmaz, 2010; S. et al. Zhang, 2012)
- `djNMF`: Discretized Joint Non-negative Matrix Factorization (CICHOCK, 2009; Zi, 2016)
- `dPLS`: Discretized Partial Least Squares (Arora, 2012)
- `dNTF`: Discretized Non-negative CP Decomposition (CICHOCK, 2009; Cichocki, 2007)
- `dNTD`: Discretized Non-negative Tucker Decomposition (CICHOCK, 2009; Kim, 2007)

Example

The SBMF and plots in Figure 1 can be easily reproduced on any machine where R is pre-installed by using the following commands in R:

```
# Install package required (one per computer)
install.packages("dcTensor")

# Load required package (once per R instance)
library("dcTensor")
library("nnTensor")
library("fields")

# Load Toy data
data <- toyModel("NMF")

# Perform SBMF
set.seed(1234)
out <- dNMF(data, Bin_U=1E+6, J=5)

# Reconstruction of the data matrix
rec.data <- out$U %*% t(out$V)

# Visualization
layout(rbind(1:2, 3:4))
image.plot(data, main="Original Data", legend.mar=8)
image.plot(rec.data, main="Reconstructed Data", legend.mar=8)
hist(out$U, breaks=100)
hist(out$V, breaks=100)
```

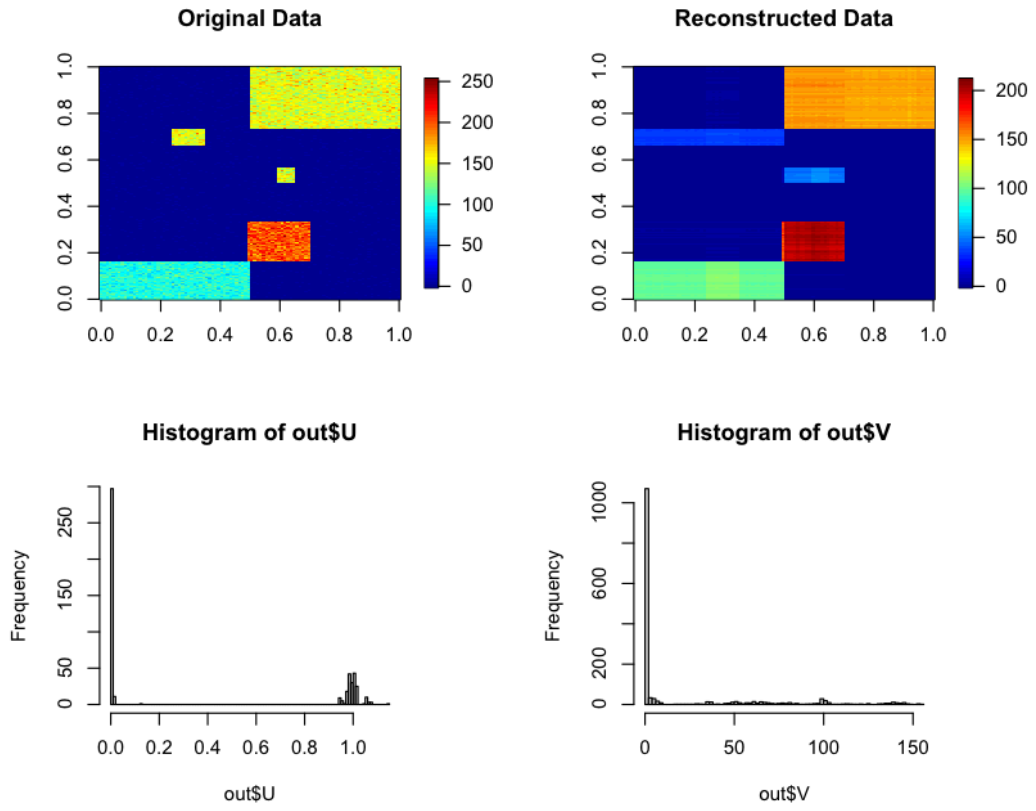

Figure 1: Semi-binary Matrix Factorization (SBMF).

We can see that the factor matrix U is almost binary, but V is not. This solution is imposed by setting a large value against Bin_U, which is the regularization parameter for U.

Related work

There are some packages to perform BMF, such as `libmf`, `recosystem`, and `Origami.jl`, but there is no package to perform TMF, SBMF, STMF, or the extensions for multiple matrices or tensors except for `dcTensor`.

References

Arora, R. (2012). Stochastic optimization for PCA and PLS. *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 861–868.

Badea, L. (2008). Extracting gene expression profiles common to colon and pancreatic adenocarcinoma using simultaneous nonnegative matrix factorization. *Pacific Symposium on Biocomputing*, 279–290.

CICHOCK, A. et al. (2009). *Nonnegative matrix and tensor factorizations*. Wiley.

Cichocki, A. et al. (2007). Non-negative tensor factorization using alpha and beta divergence. *ICASSP '07*, III-1393-III-1396.

Kim, Y.-D. et al. (2007). Nonnegative tucker decomposition. *IEEE CVPR*, 1–8.

Lee, D., & Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.

Tsuyuzaki, K. et al. (2020). Benchmarking principal component analysis for large-scale single-cell RNA-sequencing. *BMC Genome Biology*, *21(1)*, 9.

Yilmaz, Y. K. (2010). Probabilistic latent tensor factorization. *IVA/ICA 2010*, 346–353.

Zhang, S. et al. (2012). Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Research*, *40(19)*, 9379–9391.

Zhang, Z. et al. (2007). Binary matrix factorization with applications. *ICDM 2007*, 391–400.

Zi, et al., Yang. (2016). A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics*, *32(1)*, 1–8.