




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

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Matrix factorization (MF) is a widely used approach to extract significant patterns in a data matrix. MF is formalized as the approximation of a data matrix X by the matrix product of two factor matrices U and V . Because this formalization has a large number of degrees of freedom, some constraints are imposed on the solution. Non-negative matrix factorization (NMF) imposing a non-negative solution for the factor matrices is a widely used algorithm to decompose non-negative matrix data matrix. 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 (Cichocki et al., 2009).

A discrete version of NMF can also be considered by imposing a binary solution (e.g., $\{0,1\}$) for the factor matrices extracted from the data matrix and it is called binary matrix factorization (BMF) (Z. Zhang et al., 2007). BMF is recently featured in some data science domains such as market basket data, document-term data, Web click-stream data, DNA microarray expression profiles, or protein-protein complex interaction networks.

Although BMF is becoming more used, in the current data analysis, further extensions are required. For example, we may need a ternary solution (e.g., $\{0,1,2\}$) instead of a binary one. Here, I call it ternary matrix factorization (TMF). TMF would contribute to the extraction of ordered patterns, such as stages of disease severity. It is also possible to apply the discretization to only one of the two factor matrices (U or V) and here I call it semi-binary matrix factorization (SBMF) (Ma et al., 2021) or semi-ternary matrix factorization (STMF). This extension contributes to the extraction of discrete patterns in continuous-valued matrix data. Finally, there is a growing demand to extend MF to the simultaneous factorization of multiple matrices or tensors (high-dimensional arrays) (Cichocki et al., 2009). Such heterogeneous data sets are obtained when multiple measurements with a common data structure are performed under different experimental conditions. Therefore, it is very convenient if discretization is available to such heterogeneous data structures. 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 are some tools to perform BMF such as Nimfa, libmf, recosystem, and Origa mi.jl but there is no implementation to perform TMF, SBFM, STMF, or extensions of MF to multiple matrices or tensor. For this reason, I 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:

- MF against a matrix data

- dNMF: Discretized Non-negative Matrix Factorization (Cichocki et al., 2009; Lee & Seung, 1999)
- dSVD: Discretized Singular Value Decomposition (Tsuyuzaki et al., 2020)
- MF against multiple matrices data
 - dsiNMF: Discretized Simultaneous Non-negative Matrix Factorization (Badea, 2008; Cichocki et al., 2009; Yilmaz, 2010; C.-C. Zhang S. Liu et al., 2012)
 - djNMF: Discretized Joint Non-negative Matrix Factorization (Cichocki et al., 2009; Yang & Michailidis, 2016)
 - dPLS: Discretized Partial Least Squares (Arora, 2012)
- Tensor Decomposition
 - dNTF: Discretized Non-negative CP Decomposition (Cichocki et al., 2007, 2009)
 - dNTD: Discretized Non-negative Tucker Decomposition (Cichocki et al., 2009; Kim & Choi, 2007)

Example

For the demonstration, here I show that SBMF can be easily performed 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, zlim=c(0, max(data)))
image.plot(rec.data, main="Reconstructed Data", legend.mar=8, zlim=c(0, max(data)))
hist(out$U, breaks=100)
hist(out$V, breaks=100)
```

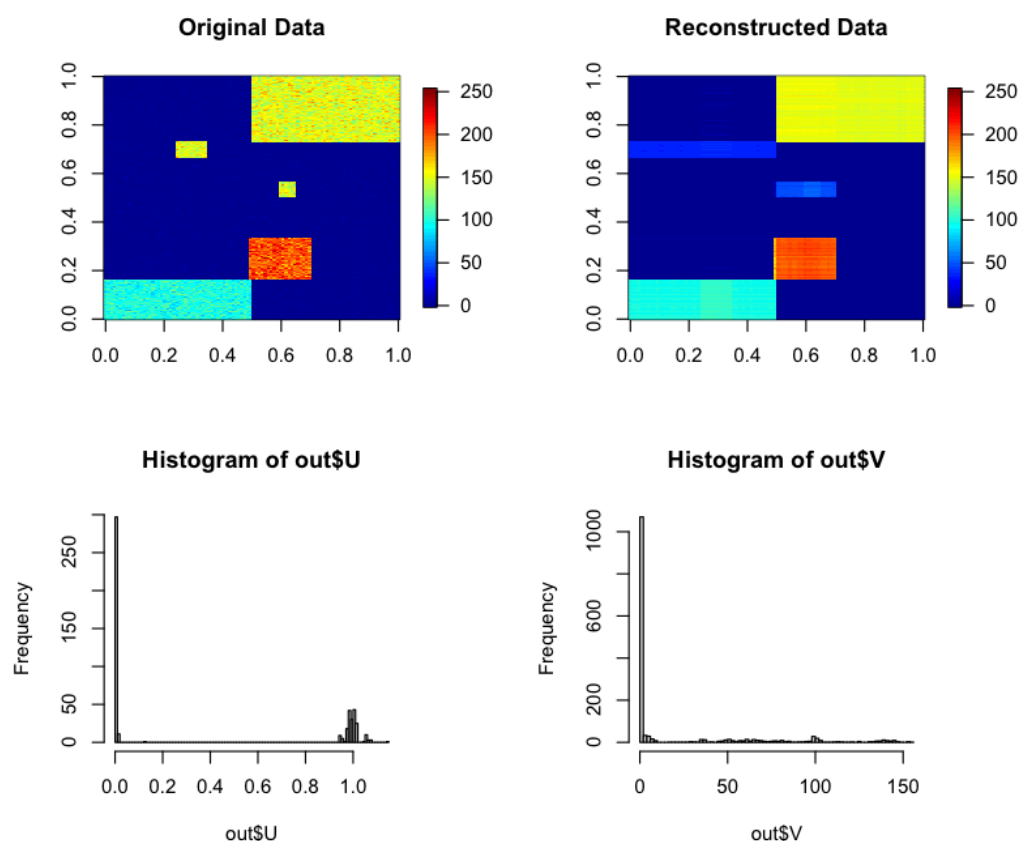


Figure 1: Semi-binary Matrix Factorization (SBMF).

In the top left of Figure 1, we can see that the demo data has five significant patterns as blocks. In the top right of Figure 1, we can see that the reconstructed data, which is the matrix product of the factor matrices U and V , also has the same patterns and this means the optimization of SBMF is properly converged. In the bottom left of Figure 1, we can see that U is binary ($\{0,1\}$), but V is not (the bottom right of Figure 1), which means the solution is semi-binary. This solution is imposed by setting a large value against `Bin_U` argument in `dNMF` function, which is the binary regularization parameter for U . `dNMF` also has `Bin_V` argument, which is the binary regularization parameter for V . Setting large values against `Bin_U` and `Bin_V`, BMF can also be obtained. Likewise, the ternary solutions (TMF and STMF) can be obtained by ternary regularization parameters such as `Ter_U` and `Ter_V`.

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. https://doi.org/10.1142/9789812776136_0027
- Cichocki, A., Zdunek, R., Choi, S., Plemmons, R., & Amari, S. (2007). Non-negative tensor factorization using alpha and beta divergence. *ICASSP '07*, III-1393-III-1396. <https://doi.org/10.1109/icassp.2007.367106>
- Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S. (2009). *Nonnegative matrix and tensor factorizations*. Wiley.

- Kim, Y.-D., & Choi, S. (2007). Nonnegative tucker decomposition. *IEEE CVPR*, 1–8. <https://doi.org/10.1109/cvpr.2007.383405>
- Lee, D., & Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 788–791. <https://doi.org/10.1038/44565>
- Ma, X., Gao, J., Liu, X., Zhang, T., & Tang, Y. (2021). Probabilistic non-negative matrix factorization with binary components. *MDPI Mathematics*, 1189. <https://doi.org/10.3390/math9111189>
- Tsuyuzaki, K., Sato, H., Sato, K., & Nikaido, I. (2020). Benchmarking principal component analysis for large-scale single-cell RNA-sequencing. *BMC Genome Biology*, 21(1), 9. <https://doi.org/10.1186/s13059-019-1900-3>
- Yang, Z., & Michailidis, G. (2016). A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics*, 32(1), 1–8. <https://doi.org/10.1093/bioinformatics/btv544>
- Yilmaz, Y. K. (2010). Probabilistic latent tensor factorization. *IVA/ICA 2010*, 346–353. https://doi.org/10.1007/978-3-642-15995-4_43
- Zhang, C.-C., S. Liu, Li, W., Shen, H., Laird, P. W., & Zhou, X. J. (2012). Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Research*, 40(19), 9379–9391. <https://doi.org/10.1093/nar/gks725>
- Zhang, Z., Li, T., Ding, C., & Zhang, X. (2007). Binary matrix factorization with applications. *ICDM 2007*, 391–400. <https://doi.org/10.1109/icdm.2007.99>