2016 Doctoral Thesis

Extensions of nonnegative matrix factorization for exploratory data analysis

Graduate School of Culture and Information Science,
Doshisha University

48141001

Superviser Prof. Hiroshi Yadohisa

Abstract

Nonnegative matrix factorization (NMF) is a matrix decomposition technique to analyze nonnegative data matrices, which are matrices of which all elements are nonnegative. The technique has been widely applied to various fields: image recognition, music sound analysis, genetic analysis, text mining, recommender systems, marketing analysis, etc. The reason for the wide application of NMF is that a nonnegative data matrix is easily obtainable. In addition, NMF is simple and easy to use and has the ability to extract interpretable information from a given nonnegative data matrix. However, NMF encounters difficulties when the data matrix has the following characteristics: it contains outliers or many zero elements or a combination of these two characteristics. The existence of outliers in a nonnegative data matrix sometimes leads to a meaningless result following matrix decomposition. Moreover, a data matrix that contains a number of zero elements, which is also referred to as a zero-inflated situation, results in poor approximation of factorization to the given data matrix. To address these difficulties, we focus on using divergence as an error criterion between the given data matrix part and the factorization part. In this regard, β-divergence is one type of divergence that has proven to be robust against outliers and has been applied to NMF by many researchers. The use of β-divergence is known to correspond to using a Tweedie distribution as an error distribution method. A specific case of the Tweedie distribution is compound Poisson-gamma (CP) distribution, which is a Poisson mixture of the gamma distribution. The CP distribution can be extended to a zero-inflated model such as that based on a zero-inflated Poisson distribution; hence, in this study we employ the zero-inflated CP (ZICP) distribution as an error distribution technique for NMF. NMF based on the ZICP distribution is potentially robust against outliers and a zero-inflated situation. This research also focuses on NMF used in combination with an orthogonal constraint (ONMF) to improve the interpretability of the factor matrix. A nonnegative factor matrix with an orthogonal constraint has a simple structure, and hence, the role of this factor matrix is similar to that of an indicator matrix used in k-means clustering. Although previous studies involving ONMF led to the proposal of algorithms to solve the ONMF problem, most of these algorithms experience difficulties when estimating factor matrices. In this study, we solve the ONMF problem using a k-means-like algorithm that produces estimates of an acceptable accuracy. Furthermore, the use of Poisson and CP distributions in the k-means-like algorithm enables us to solve the ONMF problem. This approach to ONMF is valuable because most previous algorithms of ONMF have employed the normal distribution and there are few studies about ONMF based on Poisson and CP distribution. The NMF problem can be divided into

two- and three-factor NMF: the former involves decomposition to two-factor matrices, whereas the latter entails decomposition to three-factor matrices. Three-factor NMF is assumed to comprise two types of factors: the row and column objects of the data matrix. The ZICP distribution and orthogonal constraint mentioned above can be applied not only to two-factor NMF but also three-factor NMF. The aim of this study is to present a comprehensive discussion of NMF. Especially, the discussion concentrates on the four features of NMF: two-factor vs. three-factor NMF, orthogonal constraints, distributions and divergences, and the zero-inflated model. Moreover, we present details of the model setting, the derivation of the updating rules, and an estimation algorithm for NMF with and without these features. We also include a simulation study and apply our proposed solution to real data to capture the features of these NMFs.

Contents

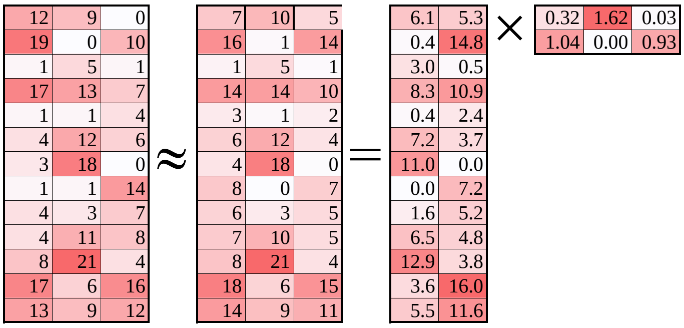
Figure Contents

Chart Contents

Chapter 1

Introduction

Many types of data in the world consist of only nonnegative values, e.g., pixel values of an image, power values with respect to frequency, microarray-based gene expression profiling, term frequency of documents, ratings on an ascending risk scale of 1 to 5, rainfall, insurance, sales, and sales quantities. These data are often presented in the form of a matrix; the elements are values corresponding to the combination of two finite sets of objects; for example, in a document-term data matrix, the row and column objects correspond to the documents and terms, and each of the entries in the matrix corresponds to a frequency of the term existing in the document. Such a matrix is referred to as a nonnegative matrix, and it is known that nonnegative matrix factorization (NMF) is one of the approaches suitable to analyze data of a particular format. NMF is employed for approximating a given nonnegative data matrix by using the product of some nonnegative matrices, which are referred to as nonnegative factor matrices. Basically, NMF is mostly used for two factor matrices, which is described as two-factor NMF in this thesis. Fig. 1.1 shows an example of two-factor NMF. NMF enables us to obtain an understanding of the co-occurrence relation between two sets of objects. For example, in the two factor matrices in Fig. 1.1, both the 1st and 3rd columns simultaneously have large values for the 2nd, 12th, and 13th rows. NMF has two advantages. First, in many cases, the estimated factor matrices have sparsity thanks to the nonnegative constraints. This sparsity leads



$$\begin{bmatrix} 12 & 9 & 0 \\ 19 & 0 & 10 \\ 1 & 5 & 1 \\ 17 & 13 & 7 \\ 1 & 1 & 4 \\ 4 & 12 & 6 \\ 3 & 18 & 0 \\ 1 & 1 & 14 \\ 4 & 3 & 7 \\ 4 & 11 & 8 \\ 8 & 21 & 4 \\ 17 & 6 & 16 \\ 13 & 9 & 12 \end{bmatrix} \approx \begin{bmatrix} 7 & 10 & 5 \\ 16 & 1 & 14 \\ 1 & 5 & 1 \\ 14 & 14 & 10 \\ 3 & 1 & 2 \\ 6 & 12 & 4 \\ 4 & 18 & 0 \\ 8 & 0 & 7 \\ 6 & 3 & 5 \\ 7 & 10 & 5 \\ 8 & 21 & 4 \\ 18 & 6 & 15 \\ 14 & 9 & 11 \end{bmatrix} = \begin{bmatrix} 6.1 & 5.3 \\ 0.4 & 14.8 \\ 3.0 & 0.5 \\ 8.3 & 10.9 \\ 0.4 & 2.4 \\ 7.2 & 3.7 \\ 11.0 & 0.0 \\ 0.0 & 7.2 \\ 1.6 & 5.2 \\ 6.5 & 4.8 \\ 12.9 & 3.8 \\ 3.6 & 16.0 \\ 5.5 & 11.6 \end{bmatrix} \times \begin{bmatrix} 0.32 & 1.62 & 0.03 \\ 1.04 & 0.00 & 0.93 \end{bmatrix}$$

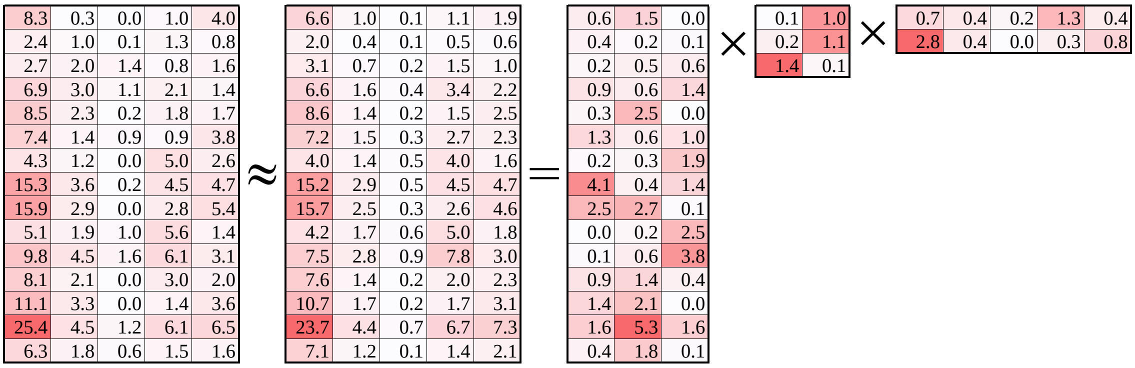
Figure 1.1: Example of two-factor NMF. The red color represents the magnitude of values: the darkest red and lightest red (i.e., white) are the maximum and minimum values in each matrix.

to an easy interpretation of the given nonnegative matrix. Lee and Seung (1999) calls this characteristic a “parts-based representation.” In face image decomposition, Lee and Seung (1999) demonstrates that parts of face can be extracted by NMF rather than principal component analysis or a vector quantization technique. Second, we can easily interpret a nonnegative factor matrix by NMF because of its nonnegativity. When a given data matrix has only nonnegative entries, the negative values in the decomposed factor matrix are difficult to explain. This is based on the idea that the basis vectors of a given nonnegative data matrix should also be nonnegative.

Few studies relating to NMF were reported until Lee and Seung (1999, 2001) developed simple and efficient algorithms to address the two-factor NMF problem. They used an auxiliary function method, also referred to as the “majorize-minimization” or “minorize-maximization” method, for developing algorithms to obtain estimates of the two factor matrices. In these algorithms, the two factor matrices are iteratively updated by multiplicative updating, in which the updates are obtained by multiplying their current values by some scalar values. Derivation of the update rule of the factor matrices necessitates the determination of the divergence as an error between the data matrix and factorization parts. Lee and Seung (2001) proposed two multiplicative updating algorithms using two divergences: the first is the Euclidean distance and the other is the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951), also referred to as I-divergence.

Following these studies, the use of NMF became widespread, prompting many researchers to extend the technique in various ways. One of the extensions involves divergence. The first generalization of the NMF algorithms Lee and Seung (2001) was published by Kompass (2007). This author pointed out the similarity between the two multiplicative update rules of the original algorithm Lee and Seung (2001) in terms of the Euclidean distance and KL divergence, found a new divergence that is between and includes the two divergences, and developed an update rule by using the new divergence. In fact, this divergence is the so-called β-divergence (Basu et al., 1998). The β-divergence has a hyper-parameter $\beta \in \mathbb{R}$, and the divergence of the generalized NMF by Kompass (2007) is the β-divergence for $\beta \in [1, 2]$. The β-divergences for $\beta = 2$ and $\beta = 1$ are the Euclidean distance and KL divergence, respectively. Subsequently, other researchers Févotte et al. (2009) proposed the NMF algorithm with the Itakura-Saito (IS) divergence (Itakura and Saito, 1968) to analyze a sound spectrogram. Févotte et al. (2009) observed that the IS divergence is a case of β-divergence such that $\beta \to 0$. Moreover, Févotte et al. (2009) found that the update rule proposed by Kompass (2007) is not only available in the case of $\beta \in [1, 2]$ but also for $\beta < 1$ and $\beta > 2$; however, the proof was not provided. The perfect proof of NMF with the β-divergence is provided by Nakano et al. (2010). Another generalized divergence to be employed in NMF is the α-divergence (Chernoff, 1952). NMF with the α-divergence was proposed by Cichocki et al. (2008). Both the β- and α-divergence are generalizations of the KL-divergence, and NMF with these generalized divergences has the advantage in that it is robust to outliers (Cichocki and Amari, 2010). From a statistical perspective, minimization of the β-divergence and the divergences of its cases are interpreted as the maximization of the log-likelihood under the assumption of the corresponding probability distribution as described in Section 3.3. In this study, we focus on the compound Poisson-gamma (CP) distribution, which corresponds to the β-divergence for $\beta \in (0, 1)$. A CP distributed random variable is the sum of n independently identically gamma-distributed random variables and the number of random variables n is Poisson distributed.

Another extension of NMF is three-factor NMF, where the data matrix is decomposed into three factor matrices. Fig. 1.2 is an example of three-factor NMF. Three-factor NMF with a column orthogonal constraint to the matrices on the left- and right- sides was proposed Ding et al. (2006). These authors Ding et al. (2006) considered three-factor NMF with no constraint, except for nonnegativity, to be equivalent to two-factor NMF, because the matrix produced as a product of the factor matrices on the left and in the center could be interpreted as being the matrix on the left in two-factor NMF. However, such a three-factor NMF is not insignificant: the estimates given by three-factor NMF are different from those given by two-factor NMF, and the approximation matrix is not the same as that of two-factor NMF. Three-factor NMF can be regarded as a two-way data case of nonnegative tensor factorization. Other researchers (Cichocki et al., 2007, 2009; Kim and Choi, 2007; Kim et al., 2008) based their work on the Tucker3 decomposition style (Tucker, 1966), in which the multi-array data is decomposed into factor matrices consisting of the objects in each array and one core tensor.



$$\begin{bmatrix} 8.3 & 0.3 & 0.0 & 1.0 & 4.0 \\ 2.4 & 1.0 & 0.1 & 1.3 & 0.8 \\ 2.7 & 2.0 & 1.4 & 0.8 & 1.6 \\ 6.9 & 3.0 & 1.1 & 2.1 & 1.4 \\ 8.5 & 2.3 & 0.2 & 1.8 & 1.7 \\ 7.4 & 1.4 & 0.9 & 0.9 & 3.8 \\ 4.3 & 1.2 & 0.0 & 5.0 & 2.6 \\ 15.3 & 3.6 & 0.2 & 4.5 & 4.7 \\ 15.9 & 2.9 & 0.0 & 2.8 & 5.4 \\ 5.1 & 1.9 & 1.0 & 5.6 & 1.4 \\ 9.8 & 4.5 & 1.6 & 6.1 & 3.1 \\ 8.1 & 2.1 & 0.0 & 3.0 & 2.0 \\ 11.1 & 3.3 & 0.0 & 1.4 & 3.6 \\ 25.4 & 4.5 & 1.2 & 6.1 & 6.5 \\ 6.3 & 1.8 & 0.6 & 1.5 & 1.6 \end{bmatrix} \approx \begin{bmatrix} 6.6 & 1.0 & 0.1 & 1.1 & 1.9 \\ 2.0 & 0.4 & 0.1 & 0.5 & 0.6 \\ 3.1 & 0.7 & 0.2 & 1.5 & 1.0 \\ 6.6 & 1.6 & 0.4 & 3.4 & 2.2 \\ 8.6 & 1.4 & 0.2 & 1.5 & 2.5 \\ 7.2 & 1.5 & 0.3 & 2.7 & 2.3 \\ 4.0 & 1.4 & 0.5 & 4.0 & 1.6 \\ 15.2 & 2.9 & 0.5 & 4.5 & 4.7 \\ 15.7 & 2.5 & 0.3 & 2.6 & 4.6 \\ 4.2 & 1.7 & 0.6 & 5.0 & 1.8 \\ 7.5 & 2.8 & 0.9 & 7.8 & 3.0 \\ 7.6 & 1.4 & 0.2 & 2.0 & 2.3 \\ 10.7 & 1.7 & 0.2 & 1.7 & 3.1 \\ 23.7 & 4.4 & 0.7 & 6.7 & 7.3 \\ 7.1 & 1.2 & 0.1 & 1.4 & 2.1 \end{bmatrix} = \begin{bmatrix} 0.6 & 1.5 & 0.0 \\ 0.4 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.6 \\ 0.9 & 0.6 & 1.4 \\ 0.3 & 2.5 & 0.0 \\ 1.3 & 0.6 & 1.0 \\ 0.2 & 0.3 & 1.9 \\ 4.1 & 0.4 & 1.4 \\ 2.5 & 2.7 & 0.1 \\ 0.0 & 0.2 & 2.5 \\ 0.1 & 0.6 & 3.8 \\ 0.9 & 1.4 & 0.4 \\ 1.4 & 2.1 & 0.0 \\ 1.6 & 5.3 & 1.6 \\ 0.4 & 1.8 & 0.1 \end{bmatrix} \times \begin{bmatrix} 0.1 & 1.0 \\ 0.2 & 1.1 \\ 1.4 & 0.1 \end{bmatrix} \times \begin{bmatrix} 0.7 & 0.4 & 0.2 & 1.3 & 0.4 \\ 2.8 & 0.4 & 0.0 & 0.3 & 0.8 \end{bmatrix}$$

Figure 1.2: Example of three-factor NMF. The red color represents the magnitude of values: the darkest red and lightest red (i.e., white) are the maximum and minimum values in each matrix.

NMF has also been expanded such that it includes some constraints for simple factor matrices. The most well-known constraint is orthogonality to factor matrices. We refer to NMF with an orthogonal constraint as "ONMF." There are two types of ONMF: two-factor and three-factor ONMF. Two-factor ONMF is two-factor NMF with an orthogonal constraint imposed on one factor matrix. Although two-factor ONMF has been applied mainly for document and term clustering because of its efficient result, it has also been adopted in some other fields (Kim et al., 2011; Mauthner et al., 2010; Wang et al., 2016). Because a nonnegative column-orthogonal matrix plays a role analogous to an indicator matrix in k-means clustering, ONMF is considered a clustering method. On the other hand, three-factor ONMF is three-factor NMF with column orthogonality to both of the factor matrices on the left and the right (Ding et al., 2006; Yoo and Choi, 2009, 2010b). Owing to the relationship between the column-orthogonal nonnegative factor matrices and clustering mentioned above, three-factor ONMF is considered to be a bi-clustering method capable of detecting the row and column clusters of a data matrix simultaneously; it has been adopted for use in document-term clustering, collaborative filtering, etc (Chen et al., 2009; Costa and Ortale, 2014). Almost all algorithms for ONMF are multiplicative updating algorithms (Choi, 2008; Ding et al., 2006; Li et al., 2010; Yoo and Choi, 2010a; Yoo and Choi, 2008, 2009). However, multiplicative updating algorithms in ONMF have two drawbacks. First, column orthogonality is not exactly (but only approximately) obtained despite the column orthogonality constraints. Second, although the objective function value tends to be non-increasing in the early stages, it is not exactly monotonically non-increasing. Mirzal (2014) pointed out the second drawback and proposed a new convergent ONMF algorithm using an additive updating rule, but there is no guarantee that a perfectly orthogonal factor matrix will be obtained. Kimura et al. (2014) proposed a new ONMF algorithm using a hierarchical alternating least-squares algorithm, rather than a multiplicative algorithm, which is a faster algorithm than the previous multiplicative algorithms but which continues to experience the above-mentioned two drawbacks. On the other hand, Pompili et al. (2014) proposed an iterative updating algorithm for ONMF in which the orthogonality and monotonically non-increasing property of the objective function value are exactly maintained. Pompili et al. (2014) found the optimization problem of ONMF to be similar to that of spherical k-means (Banerjee et al., 2003) and refers to this problem as weighted spherical k-means. In this study, we develop new two- and three-factor ONMF with KL and β-divergence in a fashion similar to that of the ONMF of Pompili et al. (2014). Of course, the ONMF has perfect orthogonal and non-increasing properties. This extension is valuable because the ONMF of Pompili et al. (2014), as well as almost all of the other ONMFs, employs a normal distribution as its error distribution; furthermore, an ONMF algorithm with KL and β-divergence has not yet been proposed.

Sometimes we encounter difficulties when working with a given sparse data matrix, in other words, a zero-inflated data matrix that contains many zero values. In fact, situations such as this often occur with larger data matrices, in which case the approximation tends to be worse than for a non-zero-inflated data matrix. In such a situation, the zero-inflated model is available to improve the goodness of approximation of the data matrix. Thus, Lambert (1992) proposed a zero-inflated Poisson (ZIP) model for count data containing many zero values, and Simchowitz (2013) developed an efficient Bayesian NMF technique for a zero-inflated nonnegative data matrix that assumes the ZIP model for nonnegative data and applied it to collaborative filtering in a recommender system. In this study, we extend NMF proposed by Simchowitz (2013) to NMF based on zero-inflated CP (ZICP) distribution. Moreover, we develop new two- and three-factor ONMF by employing zero-inflated CP distribution.

The extensions mentioned in the above discussion, that is, CP distribution, three-factor NMF, orthogonal constraint, and zero-inflated model, are appropriate for application to real-world nonnegative data. For example, data acquired to maintain records of human behavior, e.g., point of sales with customer ID and web access logs, can have outliers because of some abnormal behavior. Hence, the assumption that a matrix consisting of count values or the sum of nonnegative values for the combination of two objects given by such data displays a Poisson or CP distribution is appropriate. If it is necessary to classify objects in two sets, three-factor NMF is useful. In addition, a data matrix containing data relating to human behavior contains many zeros because the number of samples tends to be extremely small in comparison to the number of all the combinations of the objects in the two sets. Therefore, the zero-inflated model is also available to process this data. Furthermore, estimates that are easy to interpret can be obtained by using the orthogonal constraint for factor matrices.

In this paper, we describe details of NMF, especially the model, derivations of the updating rules for parameter estimation, and updating algorithms, through the perspectives of the number of factor matrices, orthogonal constraint, distribution and divergence, and zero-inflated model. These perspectives are important for exploratory data analysis using nonnegative matrices. First, in some situations, a matrix may contain some extremely large values, and this may have a significantly negative effect on estimates. One way to solve this problem is to trim rows or columns containing these large values, but this may cause important information to be lost. Moreover, it is difficult to determine what values are outliers. In this situation, a better way is to use an appropriate distribution for the robust estimation. CP distribution can be one of the solutions for robust estimation. Second, a nonnegative matrix in the real world may contain many zero entries, and the NMF model may not approximate it well. This means that some of the zero entries are unexplainable using a nonnegative linear combination of the nonnegative basis. The zero-inflated model can be one approach to handling such zero-inflated matrices. Third, in many cases nonnegative matrix analysis, the objective is to derive a simple result. The orthogonality constraint leads to a simple structure of the factor matrix and this enables us to easily interpret how the nonnegative matrix is generated. Table 1.1 shows a classification of the existing NMFs discussed in this chapter as well as our proposed NMFs. N3ONMF, P3ONMF, and P2ONMF are our original NMFs, but there are existing NMFs that have a model assumption that is the same as these. In contrast , CP2ONMF, CP3ONMF, and all four NMFs based on ZICP are completely original. The proposed NMF with orthogonal constraints is an extension of Pompili et al. (2014) and is referred to as N2ONMF. The proposed NMFs based on the ZICP distribution are extensions of Simchowitz (2013). Moreover, we observe the advantages and disadvantages and characteristics of NMFs through some simulation studies and by application to real-world data.

This paper is organized as follows. Chapter 2 provides selected notations used in this paper. Chapter 3 presents a comprehensive explanation of NMF from the perspective of the number of factor matrices, the orthogonal constraint, distribution and divergence, and the zero-inflated model. From chapter 4 to 7, we introduce details of various NMFs from the point of these four views. Chapter 8 reports simulations of the estimation accuracy and goodness of approximation of these NMF. Chapter 9 presents an example of an analysis using document and term data and point-of-sale data. Chapter 10 concludes our study and discusses open questions.

Table 1.1: Classification of NMFs into four types. The colored NMFs are discussed in this thesis. Red indicates our proposed NMFs, and blue indicates existing NMFs. The name in parentheses is the name of the NMF of the citation listed above it. For example, N2NMF is the NMF proposed by Lee and Seung (2001).

Distribution	two-factor NMF		three-factor NMF	
	non-orthogonal	orthogonal	non-orthogonal	orthogonal
normal	Lee and Seung (1999) Lee and Seung (2001) (N2NMF)	Ding et al. (2006) Yoo and Choi (2008) Choi (2008) Yoo and Choi (2010a) Li et al. (2010) Mirzal (2014) Kimura et al. (2014) Pompili et al. (2014) (N2ONMF)	Kim and Choi (2007) Cichocki et al. (2009) (N3NMF)	Ding et al. (2006) Yoo and Choi (2010b) N3ONMF
Poisson	Lee and Seung (2001) (P2NMF)	Li et al. (2010) P2ONMF	Cichocki et al. (2009) (P3NMF)	Yoo and Choi (2009) P3ONMF
exponential	Févotte et al. (2009)	-	-	-
Tweedie(CP)	Kompass (2007) Nakano et al. (2010) (CP2NMF) Févotte and Idier (2011)	CP2ONMF	Cichocki et al. (2007) Cichocki et al. (2009) (CP3NMF)	CP3ONMF
ZIP	Simchowitz (2013)	-	-	-
ZICP	ZICP2NMF	ZICP2ONMF	ZICP3NMF	ZICP3ONMF

Chapter 2

Notations and definitions

In this chapter, we introduce the notation employed in this thesis. We use bold uppercase letters, e.g., $\boldsymbol{M}$, to denote a matrix, and a lowercase letter, e.g., m_{ij}, for its i, jth element. An element i, j of a complicated matrix is expressed as $[\cdot]_{ij}$. Further, we use $\boldsymbol{m}_i$ and $\boldsymbol{m}_{(j)}$ as the vertical vector of the i-th row and j-th column of $\boldsymbol{M}$, respectively. We use the prime symbol and "-1" to express a transposed matrix and an inverse matrix, e.g., $\boldsymbol{M}'$ and $\boldsymbol{M}^{-1}$, respectively. The trace and diagonal parts of a square matrix $\boldsymbol{M}$ are denoted by $\mathrm{tr}(\boldsymbol{M})$ and $\mathrm{diag}(\boldsymbol{M})$, respectively. The Euclidean norm of a matrix or vector is represented as $\|\boldsymbol{M}\| = \sqrt{\mathrm{tr}(\boldsymbol{M}'\boldsymbol{M})}$. $\boldsymbol{D}_{\boldsymbol{M}}$ and $\boldsymbol{D}_{\boldsymbol{M}'}$ are diagonal matrices in which each diagonal element is $\|\boldsymbol{m}_{(j)}\|$ and $\|\boldsymbol{m}_i\|$, respectively. $\mathit{\Delta}(\boldsymbol{M})$ is the first left-side singular vector in which all elements are converted into nonnegative values when $\boldsymbol{M}$ is decomposed using singular value decomposition. We use $\odot$ as the Hadamard product. The element-wise quotient of two matrices is denoted by fraction notation; e.g., $\dfrac{\boldsymbol{M}}{\boldsymbol{N}}$ or $\boldsymbol{M}/\boldsymbol{N}$ is the element-wise quotient of $\boldsymbol{M}$ and $\boldsymbol{N}$. $\boldsymbol{M}^{\beta}$ is the element-wise β power of the matrix $\boldsymbol{M}$. The vectorization of the $n \times p$ matrix $\boldsymbol{M}$ is defined as $(\boldsymbol{m}'_{(1)}\ \boldsymbol{m}'_{(2)}\ \cdots\ \boldsymbol{m}'_{(p)})'$. The Kronecker product of an $n \times p$ matrix $\boldsymbol{M}$ and $\boldsymbol{N}$ are defined as

$$\boldsymbol{M} \otimes \boldsymbol{N} = \begin{pmatrix} m_{11}\boldsymbol{N} & m_{12}\boldsymbol{N} & \cdots & m_{1p}\boldsymbol{N} \\ m_{21}\boldsymbol{N} & m_{22}\boldsymbol{N} & \cdots & m_{2p}\boldsymbol{N} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1}\boldsymbol{N} & m_{n2}\boldsymbol{N} & \cdots & m_{np}\boldsymbol{N} \end{pmatrix}. \tag{2.1}$$

We denote $\boldsymbol{1}_n$ and $\boldsymbol{E}_{n\times p}$ as the n-length vector and $n \times p$ matrix of which all of the elements are 1 and denote $\boldsymbol{0}_n$ as the n-length vector of which all the elements are 0. Finally, $\mathbb{R}^{n\times p}$ is a set of $n \times p$ matrices and $\mathbb{R}_+^{n\times p}$ is a set of $n \times p$ matrices consisting of nonnegative elements. We refer to a vector and matrix consisting of nonnegative elements as a "nonnegative vector" and "nonnegative matrix," respectively.

Now, we describe the definition of an auxiliary function technique to be used for derivation of updating rules of NMFs. An auxiliary function method is a technique with which to solve the parameters such that the objective function value becomes smaller than the current value. Let $f(\theta)$ be a function to be minimized with respect to θ. Then, auxiliary function $f_{\mathrm{aux}}(\theta, \theta^*)$ is a function that satisfies the following:

$$f(\theta) \leq f_{\mathrm{aux}}(\theta, \theta^*) \text{ for all } \theta \text{ and } \theta^* \tag{2.2}$$

$$f(\theta) = f_{\mathrm{aux}}(\theta, \theta^*) \text{ if } \theta = \theta^*. \tag{2.3}$$

This definition implies that we have to find the new function using an inequality property. When we derive the auxiliary function and the solution of $\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\{f_{\mathrm{aux}}(\theta, \theta^*)\}$, we have

$$f(\theta^*) = f_{\mathrm{aux}}(\theta^*, \theta^*) \geq f_{\mathrm{aux}}(\hat{\theta}, \theta^*) \geq f(\hat{\theta}) \tag{2.4}$$

from (2.2) and (2.3). It is noted that, rather than minimizing the objective function value with respect to θ, $\hat{\theta}$ decreases the value of this function. It is important that the derived auxiliary function can easily be differentiated with respect to θ and that the optimal θ for the auxiliary function can be obtained.

Chapter 3

Various perspectives of NMF

In this chapter, we discuss the four perspectives of NMF. NMFs start from the description of a statistical model of a given data matrix with some parameters, as well as other statistical methods. The description can be divided into two parts: the structure the expected value of the data is represented by and the type of probability distribution the data follow. Section 3.1 and 3.2 are topics about the former: the number of factor matrices and orthogonal constraint. Section 3.3 and 3.4 are about the latter: the distributions and divergences and the zero-inflated model. These topics are related to some versions of our proposed NMF described from Chapter 4 to 7.

Before providing the details in each of the sections, we define and explain the NMF problem. Let $\boldsymbol{Y} \in \mathbb{R}_{+}^{n \times p}$ be a $n \times p$ data matrix that contains a nonnegative real value as each of its entries, and let $\boldsymbol{X}(\boldsymbol{\theta})$ be a representation of the matrix decomposition by some factor matrices $\boldsymbol{\theta}$. Then NMF can be defined as a problem approximating $\boldsymbol{Y}$ by $\boldsymbol{X}(\boldsymbol{\theta}) \in \mathbb{R}_{+}^{n \times p}$, that is,

$$\boldsymbol{Y} \approx \boldsymbol{X}(\boldsymbol{\theta}). \tag{3.1}$$

For the following discussions, we denote $\boldsymbol{X}$ as $\boldsymbol{X}(\boldsymbol{\theta})$ for brevity. The approximation made possible by setting the measure to evaluate the approximation and which measure to use, depends on the definition of the distribution of y_{ij}. Therefore, we can formulate the NMF problem as follows:

$$y_{ij} \overset{\text{cid}}{\sim} f(x_{ij}) \; (i = 1, \ldots, n; \; j = 1, \ldots, p), \tag{3.2}$$

where $\overset{\text{cid}}{\sim}$ signifies “conditionally independently distributed” and $f(\cdot)$ is a density or probability function of the probability distribution. In (3.2), x_{ij} is the expected value of y_{ij}. In section 3.1 and 3.2, we provide details of how x_{ij} can be specified. Then, in section 3.3 and 3.4, we describe how the $f(\cdot)$ can be defined and what the $f(\cdot)$ means.

3.1 Two-factor NMF vs three-factor NMF

A representation of the matrix decomposition by NMF, that is, $\boldsymbol{X}$, is made by some factor matrices. There are two types of matrix decomposition by NMF: two-factor and three-factor NMF.

Two-factor NMF

Two-factor NMF is a method to approximate $\boldsymbol{Y}$ by $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{A}'$, where $\boldsymbol{F} \in \mathbb{R}_{+}^{n\times k}$ is a $n \times k$ left-side nonnegative factor matrix and $\boldsymbol{A} \in \mathbb{R}_{+}^{p\times k}$ is a $p \times k$ right-side nonnegative factor matrix. Here, k is interpreted as the number of factors or clusters and recommended to be set as $k \ll \min(n, p)$. From a geometrical point of view, the i-th nonnegative sample vector $\boldsymbol{y}_i$ $(i = 1, \ldots, n)$ is approximated by using a nonnegative linear combination of nonnegative basis vectors $\boldsymbol{a}_{(m)}$ $(m = 1, \ldots, k)$ as follows:

$$\boldsymbol{y}_i \approx \sum_{m=1}^{k} f_{im}\boldsymbol{a}_{(m)}. \tag{3.3}$$

In other words, the goal of two-factor NMF is to search a convex cone low space such that all samples can be approximated by its space.

Some two-factor NMF methods are available in the "*NMF*" R package (Gaujoux and Seoighe, 2010). In this package, seven methods are implemented: "brunet" (Brunet et al., 2004; Lee and Seung, 2001), "lee" (Lee and Seung, 2001), which uses the Euclidean distance, "ls-nmf" (Wang et al., 2006), "nsNMF" (Pascual-Montano et al., 2006), "offset" (Badea, 2008), "pe-nmf" (Zhang et al., 2008), and "snmf" (Kim and Park, 2007). The "lee" and "brunet" methods have the same updating rules of Lee and Seung (2001) for Euclidean distance and KL divergence, and they are referred to as N2NMF and P2NMF, respectively, here. All other methods except "snmf" are modified version of these two methods: "ls-nmf", "offset", and "pe-nmf" are based on Lee and Seung (2001) for Euclidean distance and "nsNMF" is based on Lee and Seung (2001) for KL divergence. The "snmf" method is based on an alternating least squares approach.

Three-factor NMF

On the other hand, three-factor NMF is a method to approximate $\boldsymbol{Y}$ by $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{S}\boldsymbol{A}'$, where $\boldsymbol{F} \in \mathbb{R}_{+}^{n\times k}$ is a $n \times k$ left-side nonnegative factor matrix, $\boldsymbol{S} \in \mathbb{R}_{+}^{k\times \ell}$ is a $k \times \ell$ nonnegative factor matrix in the center, and $\boldsymbol{A} \in \mathbb{R}_{+}^{p\times \ell}$ is a $p \times \ell$ right-side nonnegative factor matrix. Here, k and ℓ are the number of factors of row objects $(i = 1, \ldots, n)$ and column objects $(j = 1, \ldots, p)$. The center factor matrix $\boldsymbol{S}$ can be interpreted as the relationship between each of the factors of the two sets of objects.

Appropriate use of two-factor NMF and three-factor NMF

Two-factor NMF has the same number of factors for row and column objects. Hence, it is recommended that two-factor NMF is used when the objects in a single row or column are of interest and the other objects are independent samples. For example, two-factor NMF has been adopted for image recognition tasks in which the row objects of the data matrix are the image samples and the column objects are the pixels of these images. The main objective of this task is to extract parts of the images, which are represented as $\boldsymbol{A}$, from the image samples, as opposed to clustering images and their pixels.

In contrast, three-factor NMF has a different number of factors for the row and column objects; hence, it can be useful when both the row and column objects are of interest and a different number of clusters is required for each of the two sets. For example, three-factor NMF has been applied to many document and term clustering problems. In this problem, the rows and columns of the data matrix are the documents and terms, respectively, and the data matrix contains frequencies corresponding to the terms and the documents. It is often assumed that the types of groups of documents are different from those of the terms: documents are preferably categorized by their content or topic, whereas terms are preferably classified not only by their content or meaning but also their function.

3.2 Orthogonal constraint

Techniques for imposing constraints on factor matrices do exist. The main aim of these constraints is to obtain a sparse estimate of the factor matrix. One of these techniques involves imposing a column orthogonal constraint. A well-known statistical technique based on the use of these constraints is principal component analysis (PCA). However, the factor matrix in PCA does not have nonnegative constraints imposed thereupon; hence, an orthogonal constraint for NMF differs slightly from that of PCA. If all entries of a matrix are nonnegative and the matrix is orthogonal, only one entry has a non-zero value and each of the others has a zero value in each row vector such that all inner products of the two column vectors are 0 with each other, then we can find that such a matrix is similar to the indicator matrix to be used in k-means clustering. That is, the matrix only contains 0 or 1 and the sum of the entries in each row is 1. Let R_m $(m = 1, \ldots k)$ be a subset of row objects $(1, \ldots, n)$ belonging to the m-th cluster of row objects; we refer to R_m as an "m-th row cluster." (3.4) and (3.5) are examples of a nonnegative matrix with column orthogonality and an indicator matrix, respectively:

$$\begin{pmatrix} \boldsymbol{b}_1 & \mathbf{0}_{|R_1|} & \cdots & \mathbf{0}_{|R_1|} \\ \mathbf{0}_{|R_2|} & \boldsymbol{b}_2 & \cdots & \mathbf{0}_{|R_2|} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{|R_k|} & \mathbf{0}_{|R_k|} & \cdots & \boldsymbol{b}_k \end{pmatrix}, \tag{3.4}$$

$$\begin{pmatrix} \mathbf{1}_{|R_1|} & \mathbf{0}_{|R_1|} & \cdots & \mathbf{0}_{|R_1|} \\ \mathbf{0}_{|R_2|} & \mathbf{1}_{|R_2|} & \cdots & \mathbf{0}_{|R_2|} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{|R_k|} & \mathbf{0}_{|R_k|} & \cdots & \mathbf{1}_{|R_2|} \end{pmatrix}. \tag{3.5}$$

where $\boldsymbol{b}_m$ $(m = 1, \ldots, k)$ is a $|R_m|$-length nonnegative vector and $|R_m|$ is the number of objects in R_m. From (3.6), the indicator matrix $\boldsymbol{U}$ for row objects has the following property.

$$u_{im} = \begin{cases} 1 & (i \in R_m) \\ 0 & (i \notin R_m) \end{cases} \quad (i = 1, \ldots, n;\ m = 1, \ldots, k). \tag{3.6}$$

On the other hand, from (3.4), a nonnegative and column orthogonal matrix $\boldsymbol{F}$ has the following property similar to an indicator matrix:

$$f_{im} = \begin{cases} b_i > 0 & (i \in R_m) \\ 0 & (i \notin R_m) \end{cases} \quad (i = 1, \dots, n;\ m = 1, \dots, k), \tag{3.7}$$

where $b_i > 0$ $(i = 1, \dots, n)$. From this perspective, NMF with the combination of nonnegativity and column orthogonality is considered as a method substantially similar to a non-hierarchical clustering method.

Three-factor NMF enables us to impose a nonnegative and column orthogonal constraint on the left- and right-side factor matrices, that is, $\boldsymbol{F}$ and $\boldsymbol{A}$. These constraints mean that both the row and column objects are simultaneously clustered. This is known as the bi-clustering method. Apart from row objects, we define a nonnegative and column orthogonal factor matrix for column objects as follows: Let C_q $(q = 1, \dots \ell)$ be a subset of column objects $\{1, \dots, p\}$ belonging to the q-th cluster of column objects; we refer to C_q as an "q-th column cluster." Similarly, we have (3.7) for the factor matrix $\boldsymbol{A}$:

$$a_{jq} = \begin{cases} d_j > 0 & (j \in C_q) \\ 0 & (j \notin C_q) \end{cases} \quad (j = 1, \dots, p;\ q = 1, \dots, \ell), \tag{3.8}$$

where $d_j > 0$ $(j = 1 \dots, p)$. We denote a set of row and column clusters as $\mathcal{R} = \{R_1, \dots, R_k\}$ and $\mathcal{C} = \{C_1, \dots, C_\ell\}$, respectively.

In many previous studies of the two-factor or three-factor orthogonal NMF problem, a factor matrix with an orthogonal constraint is updated by using the multiplicative updating algorithm (MUA); the factor matrix $\boldsymbol{F}$ is updated by the element-wise product to its current matrix $\boldsymbol{F}^*$:

$$\boldsymbol{F} \leftarrow \boldsymbol{F}^* \odot \boldsymbol{M}, \tag{3.9}$$

where "←" denotes substitution of right-side material into the left-side. $\boldsymbol{M}$ is calculated by the data matrix $\boldsymbol{Y}$, $\boldsymbol{F}^*$, and/or the other factor matrices. MUA is a well-known updating algorithm and is widely adopted in many NMFs because Lee and Seung (2001), whose work resulted in NMF becoming widely known, derived an MUA for two-factor NMF (see Section 4.1 and 4.2). Subsequently, Ding et al. (2006) and Yoo and Choi (2010b) proposed well-known MUAs for three-factor NMF with an orthogonal constraint. Both of these groups of researchers derived update rules of three-factor matrices to solve the following optimization problem:

$$\begin{aligned} &\underset{\boldsymbol{F},\boldsymbol{S},\boldsymbol{A}}{\operatorname{argmin}}\{\|\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{S}\boldsymbol{A}'\|^2\} \\ &\text{subject to } \boldsymbol{F} \in \mathbb{R}_+^{n\times k}, \boldsymbol{S} \in \mathbb{R}_+^{k\times \ell}, \boldsymbol{A} \in \mathbb{R}_+^{p\times \ell}, \boldsymbol{F}\boldsymbol{F}' = \boldsymbol{I}_k, \text{ and } \boldsymbol{A}\boldsymbol{A}' = \boldsymbol{I}_\ell. \end{aligned} \tag{3.10}$$

The first group Ding et al. (2006) solved the problem using a method comprising a Lagrange multiplier (Lagrange, 1788) and the Karush-Kuhn-Tucker condition (Karush, 1939; Kuhn and Tucker, 1951). On the other hand, Yoo and Choi (2010b) solved it as an optimization problem under a Stiefel manifold. These algorithms are presented in Algorithm

Algorithm 1 three-factor NMF by Ding et al. (2006)

```
Input $\boldsymbol{Y} \in \mathbb{R}_{+}^{n \times p}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_{+}^{n \times k}$, $\boldsymbol{S}^{(0)} \in \mathbb{R}_{+}^{k \times \ell}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_{+}^{p \times \ell}$
$t \leftarrow 0$
repeat
    $t \leftarrow t + 1$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{(t-1)} \odot \left( \frac{\boldsymbol{Y}\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}}{\boldsymbol{F}^{(t-1)}\boldsymbol{F}^{(t-1)\prime}\boldsymbol{Y}\boldsymbol{A}^{(t-1)\prime}\boldsymbol{S}^{(t-1)\prime}} \right)^{1/2}$
    $\boldsymbol{S}^{(t)} \leftarrow \boldsymbol{S}^{(t-1)} \odot \left( \frac{\boldsymbol{F}^{(t)\prime}\boldsymbol{Y}\boldsymbol{A}^{(t-1)}}{\boldsymbol{F}^{(t)\prime}\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime}\boldsymbol{A}^{(t-1)}} \right)^{1/2}$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t-1)} \odot \left( \frac{\boldsymbol{Y}^{\prime}\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}}{\boldsymbol{A}^{(t-1)}\boldsymbol{A}^{(t-1)\prime}\boldsymbol{Y}^{\prime}\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}} \right)^{1/2}$
until convergence
```

Algorithm 2 three-factor NMF by Yoo and Choi (2010b)

```
Input $\boldsymbol{Y} \in \mathbb{R}_{+}^{n \times p}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_{+}^{n \times k}$, $\boldsymbol{S}^{(0)} \in \mathbb{R}_{+}^{k \times \ell}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_{+}^{p \times \ell}$
$t \leftarrow 0$
repeat
    $t \leftarrow t + 1$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{(t-1)} \odot \frac{\boldsymbol{Y}\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}}{\boldsymbol{F}^{(t-1)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime}\boldsymbol{Y}^{\prime}\boldsymbol{F}^{(t-1)}}$
    $\boldsymbol{S}^{(t)} \leftarrow \boldsymbol{S}^{(t-1)} \odot \frac{\boldsymbol{F}^{(t)\prime}\boldsymbol{Y}\boldsymbol{A}^{(t-1)}}{\boldsymbol{F}^{(t)\prime}\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime}\boldsymbol{A}^{(t-1)}}$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t-1)} \odot \frac{\boldsymbol{Y}^{\prime}\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}}{\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t)\prime}\boldsymbol{F}^{(t)\prime}\boldsymbol{Y}\boldsymbol{A}^{(t-1)}}$
until convergence
```

1 and 2. However, the MUA for the NMF with an orthogonal constraint is problematic in two ways. First, column orthogonality is not exactly (but only approximately) obtained despite the column orthogonality constraints. Second, although the objective function value tends to be non-increasing in the early stages, it is not exactly monotonically non-increasing. On the other hand, Pompili et al. (2014) proposed a k-means–based algorithm for two-factor orthogonal NMF, in which the column orthogonality is retained in all steps in the algorithm and the objective function is monotonically decreased in each of the steps. This k-means–based algorithm is derived using the property of a nonnegative orthogonal factor matrix described in (3.7). From (3.7), the optimization problem of $\boldsymbol{F}$ (or $\boldsymbol{A}$) is divided into an optimization problem of $\mathcal{R}$ (or $\mathcal{C}$) and that of f_{im} ($i \in R_m;\ m = 1, \ldots, k$) (or $a_{j\ell}$ ($j \in C_q;\ q = 1, \ldots, \ell$)). Details of the method of Pompili et al. (2014) are introduced in Section (6.1). The other methods with an orthogonal constraint described in Chapter 6 and 7 are based on the k-means algorithm.

3.3 Distributions and divergences

Solving the NMF problem requires us to determine the criteria we use to approximate model part $\boldsymbol{X}$ to data matrix $\boldsymbol{Y}$. The well-used setting is an element-wise divergence between $\boldsymbol{Y}$ and $\boldsymbol{X}$. Table 3.1 presents selected well-known divergences in NMF. The most well-known and commonly used divergence is the Euclidean distance. The Euclidean distance offers easy and convenient way to solve an optimization problem, and is intuitively clear because of its high affinity for the real world. The other representative divergences are the KL and IS divergences. These three divergences are generalized to the β-divergence. Specifically, the Euclidean distance, KL divergence, and IS divergence are specific cases of the β-divergence for which $\beta = 2$, $\beta = 1$, and $\beta = 0$, respectively. We define the divergence

Table 3.1: Divergences between y and x and corresponding probability distribution assumptions.

Divergence		Probability distribution assumption for y
Euclidean distance (Lee and Seung, 2001)	$d_{\mathrm{EUC}}(y, x) = (y - x)^2$	normal
KL divergence (Lee and Seung, 2001)	$d_{\mathrm{KL}}(y, x)$ $= y \log(y/x) - y + x$	Poisson
IS divergence (Févotte et al., 2009)	$d_{\mathrm{IS}}(y, x)$ $= y/x - \log(y/x) - 1$	gamma (exponential)
β-divergence (Févotte and Idier, 2011) (Nakano et al., 2010)	$d_{\beta}(y, x)$ $= y(y^{\beta-1} - x^{\beta-1})/(\beta-1)$ $- (y^{\beta} - x^{\beta})/\beta$	Tweedie (Compound Poisson-gamma for $\beta \in (0, 1)$)

between two matrices as follows:

$$d_{\mathrm{EUC}}(\boldsymbol{Y},\boldsymbol{X}) = \sum_{i=1}^{n}\sum_{j=1}^{p} d_{\mathrm{EUC}}(y_{ij},x_{ij}), \tag{3.11}$$

$$d_{\mathrm{KL}}(\boldsymbol{Y},\boldsymbol{X}) = \sum_{i=1}^{n}\sum_{j=1}^{p} d_{\mathrm{KL}}(y_{ij},x_{ij}), \tag{3.12}$$

$$d_{\mathrm{IS}}(\boldsymbol{Y},\boldsymbol{X}) = \sum_{i=1}^{n}\sum_{j=1}^{p} d_{\mathrm{IS}}(y_{ij},x_{ij}), \tag{3.13}$$

$$d_{\beta}(\boldsymbol{Y},\boldsymbol{X}) = \sum_{i=1}^{n}\sum_{j=1}^{p} d_{\beta}(y_{ij},x_{ij}) \tag{3.14}$$

As seen in Table 3.1, these divergences correspond to a distribution for which y_{ij} follows a given expected value x_{ij}. The Euclidean distance, KL divergence, IS divergence, and β-divergence are derived from the assumption of normal, Poisson, gamma (exponential), and Tweedie distributions, respectively, by using maximum-likelihood procedures. Similarly to the divergences, the Tweedie distribution is a generalization of the normal, the Poisson, and the gamma (exponential) distributions (Dunn and Smyth, 2001; Jorgensen, 1997). When a random variable y follows the Tweedie distribution, we denote $y \sim TW(\mu,\phi,\beta)$. The probability density function for a random variable in the Tweedie distribution is given by

$$f(y;x,\phi) = a_{\beta}(y,\phi)\exp\left\{(y\theta(x)-\kappa(x))/\phi\right\}, \tag{3.15}$$

$$\theta(x) = \begin{cases} \dfrac{x^{\beta-1}-1}{\beta-1} & (\beta \neq 1) \\ \log x & (\beta = 1) \end{cases}, \quad \kappa(x) = \begin{cases} \dfrac{x^{\beta}-1}{\beta} & (\beta \neq 0) \\ \log x & (\beta = 0) \end{cases}, \tag{3.16}$$

where x and ϕ are the mean and dispersion parameters, respectively. $\beta \in (-\infty,\ 1] \cup [2,\ \infty)$ is the index that determines the distribution. The variance is $V(y) = \phi x^{2-\beta}$. As mentioned above, the normal ($\beta = 2$), the Poisson ($\beta = 1$), and the gamma ($\beta = 0$) distributions are specific cases of the Tweedie distribution. $a_{\beta}(y_{ij},\phi)$ varies with β, and cannot be written in closed form except in the special cases mentioned above. For $0 < \beta < 1$, the Tweedie distribution is continuous for $y > 0$, and has a mass at $y = 0$. The distribution of this range of β is referred to as a compound Poisson-gamma (CP) distribution, which is a Poisson mixture of gamma distributions. If $y = 0$, the CP distribution is a Poisson distribution at $y = 0$, that is,

$$P(y=0) = \exp\{-\lambda\}, \tag{3.17}$$

and if $y > 0$, the density function of the CP distribution is

$$f(y;x,\phi) = \sum_{n=1}^{\infty}\left[\frac{\lambda^{n}\exp\{-\lambda\}}{n!}\right]\left[\frac{b^{n\alpha}}{\Gamma(n\alpha)}y^{n\alpha-1}\exp\{-bx\}\right], \tag{3.18}$$

where

$$\lambda = \frac{x^{\beta}}{\phi\beta}, \tag{3.19}$$

$$b = \frac{x^{\beta-1}}{\phi(1-\beta)}, \tag{3.20}$$

$$\alpha = \frac{\beta}{1-\beta}. \tag{3.21}$$

Equations (3.19), (3.20), and (3.21) are given by comparing the two forms of cumulant generating function from the two ways of defining the CP distribution; one is a Poisson mixture of the gamma distribution and the other is the power variance assumption of the exponential dispersion model. The proof is given by Jorgensen (1997) and Şimşekli et al. (2013). Substituting (3.19) and (3.20) into (3.18), we have

$$f(y;x,\phi) = \frac{1}{y}\sum_{n=1}^{\infty}\frac{y^{n\alpha}(1-\beta)^{-n\alpha}}{\phi^{n(1+\alpha)}\beta^{n}n!\Gamma(n\alpha)}\exp\left\{\frac{1}{\phi}\left(\frac{yx^{\beta-1}}{\beta-1}-\frac{x^{\beta}}{\beta}\right)\right\} \tag{3.22}$$

$$= h(y,\phi,\beta)\exp\left\{\frac{1}{\phi}\left(\frac{yx^{\beta-1}}{\beta-1}-\frac{x^{\beta}}{\beta}\right)\right\}. \tag{3.23}$$

Here, we define $h(y, \phi, \beta)$ as

$$h(y,\phi,\beta) = \frac{1}{y}\sum_{n=1}^{\infty}\frac{y^{n\alpha}(1-\beta)^{-n\alpha}}{\phi^{n(1+\alpha)}\beta^{n}n!\Gamma(n\alpha)}. \tag{3.24}$$

From (3.15) and (3.16), we have

$$f(y;x,\phi) = a_{\beta}(y,\phi)\exp\left\{-\frac{y}{\beta-1}+\frac{1}{\beta}\right\}\exp\left\{\frac{1}{\phi}\left(\frac{yx^{\beta-1}}{\beta-1}-\frac{x^{\beta}}{\beta}\right)\right\}. \tag{3.25}$$

Hence, we have

$$a_{\beta}(y,\phi) = \exp\left\{\frac{y}{\beta-1}-\frac{1}{\beta}\right\}h(y,\phi,\beta). \tag{3.26}$$

Fig. 3.1 shows plots of the probability density functions of the Tweedie distribution for various values of β, where $\mu = 1$ and $\phi = 1$.

In NMF, the index parameter β affects the robustness of parameter estimation (2010). Fig. 3.2 shows graphs of β divergence $d_{\beta}(y,x) = y(y^{\beta-1} - x^{\beta-1})/(\beta-1) - (y^{\beta} - x^{\beta})/\beta$ given $y = 10$ (left side) and $y = 100$ (right side) for various values of β. The value of $d_{\beta}(y,x)$ is lower about $x = 100$, given $y = 100$, than about $x = 10$ given $y = 10$, for $\beta < 2$. In NMF, this means that extremely large values are not taken into account in parameter estimation for values of $\beta < 2$.

In this study, we focus on NMF using normal, Poisson, and compound Poisson-gamma distribution. The reason is as follows: our proposed NMFs are developed considering a two-way table consisting of a count (such as a contingency table) or a gross summation of the nonnegative values of a pair of objects in two sets. Count data is commonly used with Poisson distribution, and a gross summation of the nonnegative values is compatible

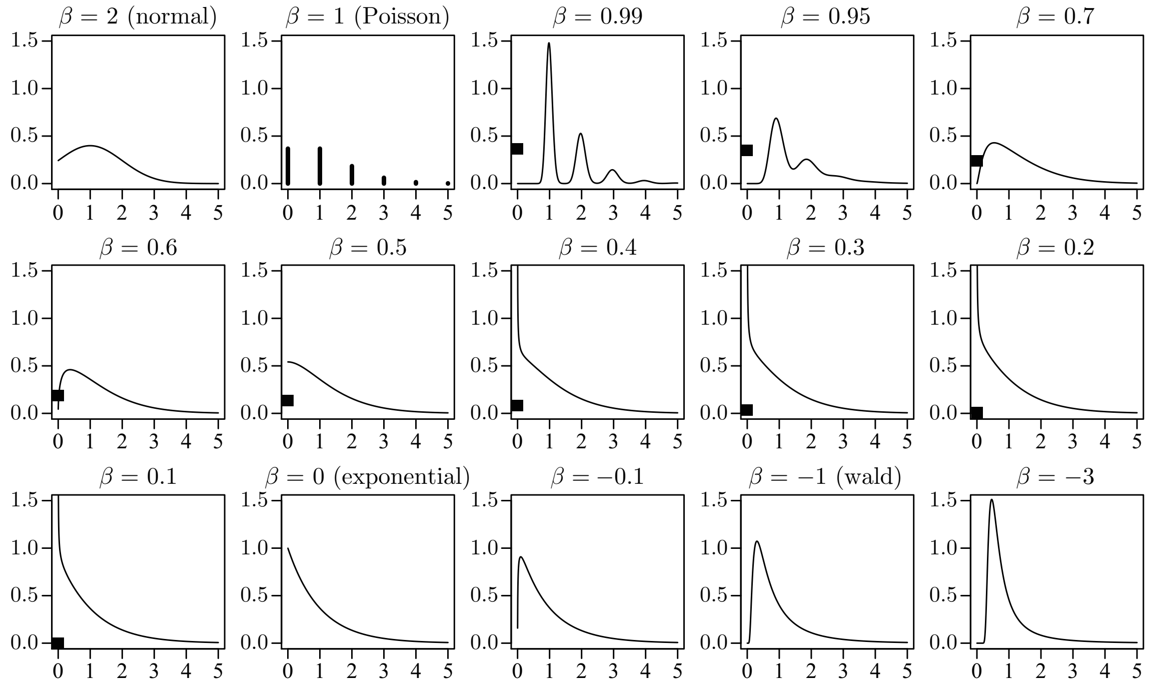


Figure 3.1: Probability density functions of the Tweedie distribution for various values of β. The black square represents the probability at $y = 0$.

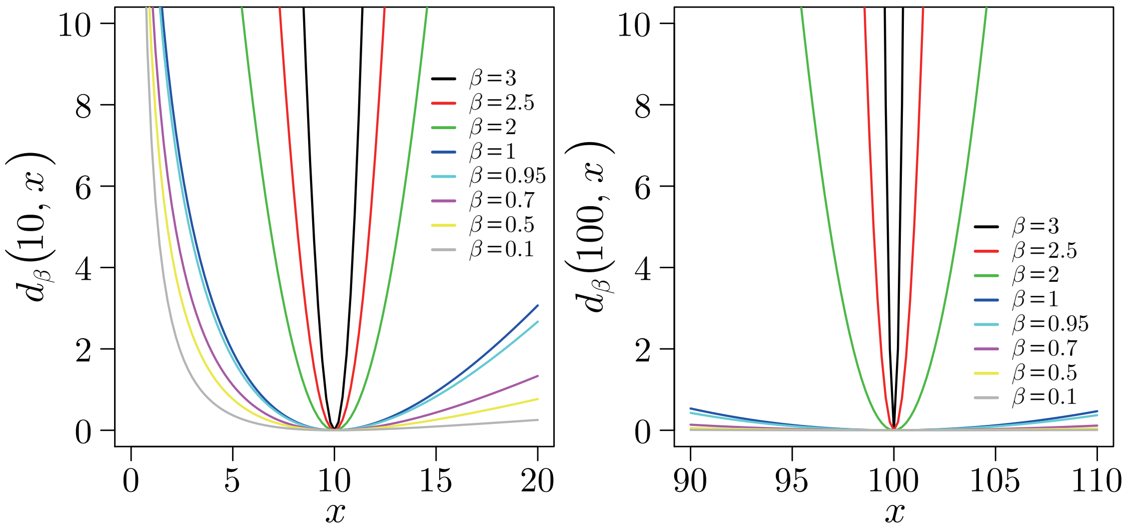


Figure 3.2: Graphs of β divergence $d_\beta(y, x) = y(y^{\beta-1} - x^{\beta-1})/(\beta - 1) - (y^\beta - x^\beta)/\beta$, given $y = 10$ (left side) and $y = 100$ (right side), for various values of β. The horizontal and vertical axes represent x and $d_\beta(y, x)$, respectively.

with the generate model of CP. Henceforth, we introduce the probability or density function of these distributions, likelihood under these distribution assumptions, and objective function of the NMF derived as a minus logarithm of their likelihood. These topics form an introduction to the NMFs described in Chapter 4 to 7.

Normal distribution

When random variable y follows a normal distribution, we denote $y \sim N(x, \sigma^2)$, where x and σ^2 are an expected value and a variance of y, respectively. Then, its density function is defined as follows:

$$f_{\mathrm{N}}(y|x, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}(y-x)^2 \right\}. \tag{3.27}$$

We assumed that all elements of $\boldsymbol{Y}$, i.e., y_{ij} $(i = 1, \ldots, n;\ j = 1, \ldots, p)$, are conditionally independent normal distributed random variables with mean $x_{ij}(\boldsymbol{\theta})$, that is:

$$y_{ij} \overset{\text{cid}}{\sim} N(x_{ij}, \sigma^2)\ (i = 1, \ldots, n;\ j = 1, \ldots, p). \tag{3.28}$$

Then, the likelihood with respect to $\boldsymbol{\theta}$ and σ^2 is:

$$L(\boldsymbol{\theta}, \sigma^2|\boldsymbol{Y}) = \prod_{i=1}^{n}\prod_{j=1}^{p} f_{\mathrm{N}}(y_{ij}|x_{ij}, \sigma^2) = \prod_{i=1}^{n}\prod_{j=1}^{p} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}(y_{ij}-x_{ij})^2 \right\}. \tag{3.29}$$

The objective function to be minimized with respect to $\boldsymbol{\theta}$ and σ^2 is obtained as the minus logarithm of (3.29) as follows:

$$\begin{aligned} Q(\boldsymbol{\theta}, \sigma^2) &= -\log\left\{ \prod_{i=1}^{n}\prod_{j=1}^{p} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}(y_{ij}-x_{ij})^2 \right\} \right\} \\ &= \frac{np}{2}\log\{\sigma^2\} + \frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{p}(y_{ij}-x_{ij})^2 + \text{const}, \end{aligned} \tag{3.30}$$

where "const" denotes terms that are independent of the parameters to be optimized. The optimal σ^2 is derived by differentiating (3.30) with respect to σ^2 and setting them to zero as follows:

$$\hat{\sigma}^2 = \frac{1}{np}\sum_{i=1}^{n}\sum_{j=1}^{p}(y_{ij}-x_{ij})^2 = \frac{1}{np}\|\boldsymbol{Y} - \boldsymbol{X}\|^2. \tag{3.31}$$

Poisson distribution

When random variable y follows a Poisson distribution, we denote $y \sim Po(x)$, where x is an expected value (and a variance) of y. Then, its density function is defined as follows:

$$f_{\mathrm{P}}(y|x) = \frac{x^y \exp\{-x\}}{y!}. \tag{3.32}$$

We assumed that all elements of $\boldsymbol{Y}$, i.e., y_{ij} $(i = 1, \ldots, n;\ j = 1, \ldots, p)$, are conditionally independent Poisson distributed random variables with mean $x_{ij}(\boldsymbol{\theta})$, that is:

$$y_{ij} \overset{\text{cid}}{\sim} Po(x_{ij})\ (i = 1, \ldots, n;\ j = 1, \ldots, p). \tag{3.33}$$

Then, the likelihood with respect to $\boldsymbol{\theta}$ is:

$$L(\boldsymbol{\theta}|\boldsymbol{Y}) = \prod_{i=1}^{n}\prod_{j=1}^{p} f_{\mathrm{P}}(y_{ij}|x_{ij}) = \prod_{i=1}^{n}\prod_{j=1}^{p} \frac{x_{ij}^{y_{ij}}\exp\{-x_{ij}\}}{y_{ij}!}. \tag{3.34}$$

The objective function to be minimized with respect to $\boldsymbol{\theta}$ is obtained as the minus logarithm of (3.34) as follows:

$$\begin{aligned} Q(\boldsymbol{\theta}) &= -\log\left\{\prod_{i=1}^{n}\prod_{j=1}^{p} \frac{x_{ij}^{y_{ij}}\exp\{-x_{ij}\}}{y_{ij}!}\right\} \\ &= -\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}\log\{x_{ij}\} + \sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij} + \text{const}. \end{aligned} \tag{3.35}$$

Compound Poisson-gamma distribution

When random variable y follows a CP distribution, we denote $y \sim CP(x, \phi, \beta)$, where x is an expected value, $\beta \in (0, 1)$ is an index parameter, and ϕ is a dispersion parameter. Then, its density function is defined as follows:

$$f_{\mathrm{CP}}(y|x, \phi, \beta) = h(y, \phi, \beta)\exp\left\{\frac{1}{\phi}\left(\frac{yx^{\beta-1}}{\beta-1} - \frac{x^{\beta}}{\beta}\right)\right\}. \tag{3.36}$$

Function $h(y, \phi, \beta)$ is defined in (3.24). We assumed that all elements of $\boldsymbol{Y}$, i.e., y_{ij} $(i = 1, \ldots, n;\ j = 1, \ldots, p)$, are conditionally independent CP distributed random variables with mean $x_{ij}(\boldsymbol{\theta})$, that is:

$$y_{ij} \overset{\text{cid}}{\sim} CP(x_{ij}, \phi, \beta)\ (i = 1, \ldots, n;\ j = 1, \ldots, p). \tag{3.37}$$

Then, the likelihood with respect to $\boldsymbol{\theta}$ and ϕ is:

$$\begin{aligned} L(\boldsymbol{\theta}, \phi|\boldsymbol{Y}) &= \prod_{i=1}^{n}\prod_{j=1}^{p} f_{\mathrm{CP}}(y_{ij}|x_{ij}, \phi, \beta) \\ &= \prod_{i=1}^{n}\prod_{j=1}^{p} h(y_{ij}, \phi, \beta)\exp\left\{\frac{1}{\phi}\left(\frac{y_{ij}x_{ij}^{\beta-1}}{\beta-1} - \frac{x_{ij}^{\beta}}{\beta}\right)\right\}. \end{aligned} \tag{3.38}$$

The objective function to be minimized with respect to $\boldsymbol{\theta}$ and ϕ is obtained as the minus logarithm of (3.38) as follows:

$$Q(\boldsymbol{\theta}, \phi) = -\sum_{i=1}^{n}\sum_{j=1}^{p}\log\{h(y_{ij}, \phi, \beta)\} - \frac{1}{\phi}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\frac{y_{ij}x_{ij}^{\beta-1}}{\beta-1} - \frac{x_{ij}^{\beta}}{\beta}\right). \tag{3.39}$$

The optimal ϕ cannot be obtained analytically because of the $h(y_{ij}, \phi, \beta)$ term in (3.39). However, if we use the BFGS quasi-Newton method (Byrd et al., 1995) with constraints $\phi > 0$, we always obtain the optimal value of ϕ. This property has not been proved yet, but from our experience, this could be made available.

It is noted that β is considered as a hyper-parameter in this study; β is not estimated. One of the ways to obtain the optimal β is to use a numerical optimization method in the same manner as for ϕ. However, as mentioned above, the variance in the random variable in the Tweedie distribution is $V(y) = \phi x^{\beta}$. This shows that ϕ and β are closely related. From (3.39), we can also find that x_{ij} does not depend on ϕ, but on β instead. Hence, if we optimize β, we should optimize ϕ simultaneously. Unfortunately, as mentioned above, the normalization term $h(y, \phi, \beta)$ in the density function of the CP cannot be analytically calculated; thus, a simultaneous search by a numerical optimization method is computationally time consuming. Moreover, the change in β values indicates a change in divergence. This means that we change the manner of estimating factor matrices. Therefore, β should be determined in advance based on some prior knowledge. These facts explain the difficulty of estimating the index parameters. If β has to be estimated, we may use an approach to approximate the log likelihood, or the Bayesian approach proposed by (Zhang, 2013) in a generalized linear model. A future task would involve extending this procedure to NMF.

3.4 Zero-inflated model

One of the aspects on which we focus in this study is to analyze the two-way table of the count or the gross sum of nonnegative values. Such data often contain many zeroes, resulting in a sparse matrix $\boldsymbol{Y}$. In this situation, the accuracy of the approximation tends to be poor. Fig. 3.3 shows two examples of two-factor NMF using a non-zero-inflated matrix (left side) and a zero-inflated matrix (right side). When a data matrix has many zero entries, most of its data vectors are located along the edge of the first quadrant, as shown on the right side of the three-dimensional (3D) plot in Fig. 3.3. Hence, most data vectors cannot be approximated by the linear subspace, which leads to a worsening

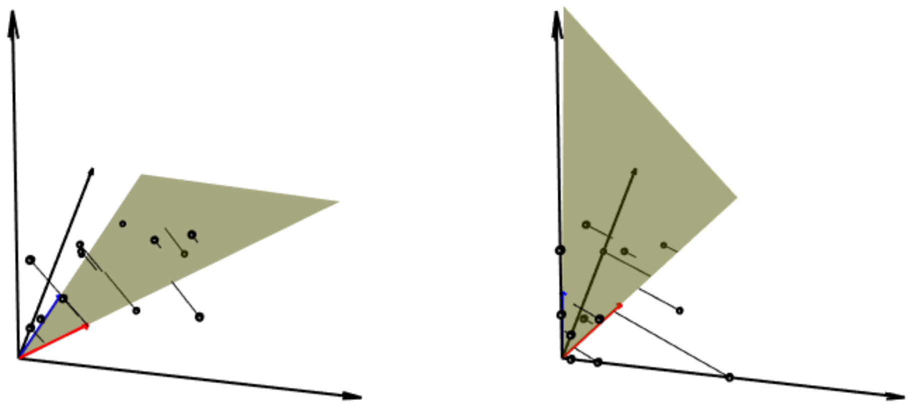


Figure 3.3: Examples of two-factor NMF using a non-zero-inflated matrix (left side) and zero-inflated matrix (right side). The size of both matrices is 13 × 3. The convex cones represent the column space of $\boldsymbol{A}$, estimated using the respective matrices.

approximation of the basic NMF. Simchowitz (2013) proposed a zero-inflated Poisson NMF based on a Bayesian method to estimate unvalued points in nonnegative preference matrices for collaborative filtering in a recommender system. This is an extended NMF model based on the Poisson distribution. In our study, we propose an NMF model based on the ZICP distribution. Since the CP distribution is a generalization of the Poisson distribution, our proposed model is a generalization of Simchowitz's NMF model. We used a numerical simulation to demonstrate the accuracy of approximation of the proposed model for a sparse data matrix $\boldsymbol{Y}$.

Here, we introduce the ZICP distribution and the objective function for NMF. When random variable y follows a ZICP distribution, we denote $y \sim ZICP(x, \phi, \beta, w)$, or

$$\begin{cases} y \sim 0 & \text{with probability } w \\ y \sim CP(x, \phi, \beta) & \text{with probability } 1 - w, \end{cases} \tag{3.40}$$

where x is an expected value, $\beta \in (0, 1)$ is an index parameter, ϕ is a dispersion parameter, and $w \in (0, 1)$ is a mixture ratio. Then, its density function is defined as follows:

$$f_{\text{ZICP}}(y|x, \beta, \phi, w) = wI(y = 0) + (1 - w)f_{\text{CP}}(y|x, \phi, \beta), \tag{3.41}$$

where $I(\cdot)$ is an indicator function. We assumed that all elements of $\boldsymbol{Y}$, i.e., y_{ij} $(i = 1, \ldots, n;\ j = 1, \ldots, p)$, are conditionally independent zero-inflated compound Poisson-gamma distributed random variables with mean $x_{ij}(\boldsymbol{\theta})$, that is:

$$y_{ij} \overset{\text{cid}}{\sim} ZICP(x_{ij}, \phi, \beta, w)\ (i = 1, \ldots, n;\ j = 1, \ldots, p). \tag{3.42}$$

Then, the likelihood with respect to $\boldsymbol{\theta}$, ϕ, w is:

$$\begin{aligned} L(\boldsymbol{\theta}, \phi, w|\boldsymbol{Y}) &= \prod_{i=1}^{n}\prod_{j=1}^{p}[wI(y_{ij} = 0) + (1 - w)f_{\text{CP}}(y_{ij}|x_{ij}, \phi, \beta)] \\ &= \prod_{i=1}^{n}\prod_{j=1}^{p}\left[wI(y_{ij} = 0) + (1 - w)h(y_{ij}, \phi, \beta)\exp\left\{\frac{1}{\phi}\left(\frac{y_{ij}x_{ij}^{\beta-1}}{\beta - 1} - \frac{x_{ij}^{\beta}}{\beta}\right)\right\}\right], \end{aligned} \tag{3.43}$$

The objective function to be minimized with respect to $\boldsymbol{\theta}$, ϕ, and w is obtained as the minus logarithm of (3.43) as follows:

$$Q(\boldsymbol{\theta}, \phi, w) = -\sum_{i=1}^{n}\sum_{j=1}^{p}\log\left[wI(y_{ij} = 0) + (1 - w)h(y_{ij}, \phi, \beta)\exp\left\{\frac{1}{\phi}\left(\frac{y_{ij}x_{ij}^{\beta-1}}{\beta - 1} - \frac{x_{ij}^{\beta}}{\beta}\right)\right\}\right]. \tag{3.44}$$

We are not directly using (3.44) to derive the update rules for $\boldsymbol{\theta}$, ϕ, and w because (3.44) is difficult to differentiate with respect to these parameters. Instead of using the likelihood (3.43), we use the complete likelihood function in the expectation-maximization (EM) algorithm (Dempster et al., 1977; McLachlan and Krishnan, 2007). We consider latent variables z_{ij} $(i = 1, \ldots, n;\ j = 1, \ldots, p)$ such that

$$z_{ij} = \begin{cases} 1 & \text{if } y_{ij} \sim 0 \\ 0 & \text{if } y_{ij} \sim CP(x_{ij}, \phi, \beta), \end{cases} \tag{3.45}$$

and assume that z_{ij} $(i = 1, \dots, n;\ j = 1, \dots, p)$ is an identically and independently Bernoulli-distributed random variable, that is,

$$z_{ij} \sim Be(w), \tag{3.46}$$

and hence the probability function of z_{ij} is

$$f_{z_{ij}}(z_{ij}|w) = w^{z_{ij}}(1-w)^{1-z_{ij}}. \tag{3.47}$$

We also define the conditional distribution of y_{ij} given z_{ij} from (3.45) as follows:

$$f_{y_{ij}|z_{ij}}(y_{ij}|z_{ij}, \boldsymbol{\theta}, \phi) = \{I(y_{ij}=0)\}^{z_{ij}} \left\{ h(y_{ij}, \phi, \beta) \exp\left\{ \frac{1}{\phi} \left(\frac{y_{ij} x_{ij}^{\beta-1}}{\beta - 1} - \frac{x_{ij}^{\beta}}{\beta} \right) \right\} \right\}^{1-z_{ij}}. \tag{3.48}$$

From (3.47) and (3.48), the joint distribution of y_{ij} and z_{ij} is as follows:

$$\begin{aligned} & f_{y_{ij}, z_{ij}}(y_{ij}, z_{ij}|\boldsymbol{\theta}, \phi, w) \\ & = f_{y_{ij}|z_{ij}}(y_{ij}|z_{ij}, \boldsymbol{\theta}, \phi) f_{z_{ij}}(z_{ij}|w) \\ & = \{wI(y_{ij}=0)\}^{z_{ij}} \left\{ (1-w) h(y_{ij}, \phi, \beta) \exp\left\{ \frac{1}{\phi} \left(\frac{y_{ij} x_{ij}^{\beta-1}}{\beta - 1} - \frac{x_{ij}^{\beta}}{\beta} \right) \right\} \right\}^{1-z_{ij}}. \end{aligned} \tag{3.49}$$

The complete likelihood is defined as the joint distribution of $\boldsymbol{Y}$ and $\boldsymbol{Z}$ as follows:

$$\begin{aligned} & L(\boldsymbol{\theta}, \phi, w|\boldsymbol{Y}, \boldsymbol{Z}) \coloneqq f_{\boldsymbol{Y},\boldsymbol{Z}}(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\theta}, \phi, w) \\ & = \prod_{i=1}^{n} \prod_{j=1}^{p} f_{y_{ij}, z_{ij}}(y_{ij}, z_{ij}|\boldsymbol{\theta}, \phi, w) \\ & = \prod_{i=1}^{n} \prod_{j=1}^{p} \{wI(y_{ij}=0)\}^{z_{ij}} \left\{ (1-w) h(y_{ij}, \phi, \beta) \exp\left\{ \frac{1}{\phi} \left(\frac{y_{ij} x_{ij}^{\beta-1}}{\beta - 1} - \frac{x_{ij}^{\beta}}{\beta} \right) \right\} \right\}^{1-z_{ij}}. \end{aligned} \tag{3.50}$$

Then, the new objective function is defined using the minus logarithm of (3.50), instead of (3.44), as follows:

$$\begin{aligned} Q_{\mathrm{comp}}(\boldsymbol{\theta}, \phi, w) = - \sum_{i=1}^{n} \sum_{j=1}^{p} \Bigg[& \hat{z}_{ij} \log\{w\} + (1 - \hat{z}_{ij}) \Bigg\{ \log\{1-w\} \\ & + \log\{h(y_{ij}, \phi, \beta)\} + \frac{1}{\phi} \left(\frac{y_{ij} x_{ij}^{\beta-1}}{\beta - 1} - \frac{x_{ij}^{\beta}}{\beta} \right) \Bigg\} \Bigg] \end{aligned} \tag{3.51}$$

where $\hat{\boldsymbol{Z}}$ is a conditional expected value of $\boldsymbol{Z}$ given $\boldsymbol{Y}$, such that

$$\begin{aligned} \hat{z}_{ij} & = E[z_{ij}|y_{ij}] \\ & = \sum_{z_{ij} \in \{0,1\}} z_{ij} \frac{f_{y_{ij}, z_{ij}}(y_{ij}, z_{ij}|\boldsymbol{\theta}, \phi, w)}{f_{y_{ij}}(y_{ij}|\boldsymbol{\theta}, \phi, w)} \\ & = \frac{wI(y_{ij}=0)}{wI(y_{ij}=0) + (1-w) f_{\mathrm{CP}}(y_{ij}|x_{ij}, \phi, \beta)} \\ & = \begin{cases} \dfrac{w}{w + (1-w) h(0, \phi, \beta) \exp\{-x_{ij}^{\beta}/(\phi\beta)\})} & \text{if } y_{ij} = 0 \\ 0 & \text{if } y_{ij} \neq 0. \end{cases} \end{aligned} \tag{3.52}$$

(3.52) is known as the update rule of Estep in the EM algorithm. From (3.51), the objective function with respect to w is as follows:

$$Q_w(w) = -\log\{w\}\sum_{i=1}^{n}\sum_{j=1}^{p}\hat{z}_{ij} - \log\{1-w\}\sum_{i=1}^{n}\sum_{j=1}^{p}(1-\hat{z}_{ij}). \tag{3.53}$$

The partial derivative of (3.53) is as follows:

$$\frac{\partial Q_w(w)}{\partial w} = -\frac{1}{w}\sum_{i=1}^{n}\sum_{j=1}^{p}\hat{z}_{ij} + \frac{1}{1-w}\sum_{i=1}^{n}\sum_{j=1}^{p}(1-\hat{z}_{ij}). \tag{3.54}$$

The optimal $\hat{w}$ is obtained by setting (3.54) to 0 and solving the equation with respect to w as follows:

$$\hat{w} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{p}\hat{z}_{ij}}{np}. \tag{3.55}$$

From (3.51), the objective function to be minimized with respect to ϕ is as follows:

$$Q_\phi(\phi) = -\sum_{i=1}^{n}\sum_{j=1}^{p}\hat{z}^*_{ij}\log\{h(y_{ij},\phi,\beta)\} - \frac{1}{\phi}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\frac{\hat{z}^*_{ij}y_{ij}x_{ij}^{\beta-1}}{\beta-1} - \frac{\hat{z}^*_{ij}x_{ij}^{\beta}}{\beta}\right). \tag{3.56}$$

The optimal ϕ cannot be obtained analytically for the same reason described in Section 3.3. Hence, we use the BFGS quasi-Newton method (Byrd et al., 1995) in the same manner as for CP.

From Chapter 4 to 7, we introduce an algorithm for each of the NMFs based on a maximum likelihood estimation method as listed in Table 3.2. N2NMF and P2NMF are the same as the NMFs proposed in Lee and Seung (2001). CP2NMF is an NMF of Nakano et al. (2010) for $\beta \in (0,1)$. N2ONMF is the same as the NMF proposed in Pompili et al. (2014). The algorithms of N3NMF, P3NMF, and CP3NMF are introduced in Cichocki et al. (2009). The others NMFs written on colored cells in Table 3.2 are proposed by us.

Table 3.2: NMFs presented in this paper. The NMFs in gray cells are proposed methods.

Distribution	two-factor		three-factor	
	non-orthogonal	orthogonal	non-orthogonal	orthogonal
normal	N2NMF (Section 4.1)	N2ONMF (Section 6.1)	N3NMF (Section 5.1)	N3ONMF (Section 7.1)
Poisson	P2NMF (Section 4.2)	P2ONMF (Section 6.2)	P3NMF (Section 5.2)	P3ONMF (Section 7.2)
CP	CP2NMF (Section 4.3)	CP2ONMF (Section 6.3)	CP3NMF (Section 5.3)	CP3ONMF (Section 7.3)
ZICP	ZICP2NMF (Section 4.4)	ZICP2ONMF (Section 6.4)	ZICP3NMF (Section 5.4)	ZICP3ONMF (Section 7.4)

It is noted that the convergence of this algorithm is determined by the log-likelihood value. However, we can use the corresponding divergence (3.11), (3.12), and (3.14) to determine the convergence of all algorithms except for that of using the zero-inflated model.

Chapter 4

Two-factor NMF

In this chapter, we describe four two-factor NMFs. The aim of all methods in this chapter is to obtain estimates of the factor matrices, $\boldsymbol{F} \in \mathbb{R}_+^{n\times k}$ and $\boldsymbol{A} \in \mathbb{R}_+^{p\times k}$, such that $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{A}'$ is approximated to a given data matrix $\boldsymbol{Y} \in \mathbb{R}_+^{n\times p}$. All of the methods are based on a maximum likelihood estimation method; hence, our goal is to obtain the optimal parameters that minimize the objective function defined as the minus logarithm of the likelihood. Since we cannot derive the global solution of all parameters simultaneously, we attempt to derive an update rule for each parameter to at least decrease the objective function given the other parameters, and to develop an iterative algorithm to optimize the objective function.

4.1 Normal distribution

In this section we present details of two-factor NMF based on a normal distribution, named N2NMF. The objective function is defined as (3.30), where $\boldsymbol{\theta} = \{\boldsymbol{F}, \boldsymbol{A}\}$ and $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{A}'$.

Update rules

We present the updates rules of the parameters, $\boldsymbol{F}$ and $\boldsymbol{A}$. Note that the update rule of σ^2 is (3.31).

Update rule for $\boldsymbol{F}$

From (3.30), the objective function to be minimized with respect to $\boldsymbol{F}$ is as follows:

$$Q_{\boldsymbol{F}}(\boldsymbol{F}) = -2\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij} \sum_{m=1}^{k} f_{im}a_{jm} + \sum_{i=1}^{n}\sum_{j=1}^{p}\left(\sum_{m=1}^{k} f_{im}a_{jm}\right)^2. \tag{4.1}$$

It is difficult to obtain an optimal f_{im} by differentiating (4.1) with respect to f_{im} because the simultaneous equation in f_{im} ($i = 1, \ldots, n$; $m = 1, \ldots, k$) is complicated by the summation of f_{im} for m in the square function in the objective function. In this regard, Lee and Seung (2001) tried to obtain an update rule of f_{im} using the auxiliary function method described in Chapter 2. Now, we show the derivation of an update rule of f_{im} by using the auxiliary function method. We can find that the following function

$$f(x) = x^2 \tag{4.2}$$

is in the second term of (4.1). Obviously, this function is convex. Hence, we can use the Jensen inequality to derive the auxiliary function of $Q_{\boldsymbol{F}}(\boldsymbol{F})$. Then, we have

$$\left(\sum_{m=1}^{k} f_{im}a_{jm}\right)^2 = \left(\sum_{m=1}^{k} \lambda_{ijm}\frac{f_{im}a_{jm}}{\lambda_{ijm}}\right)^2 \leq \sum_{m=1}^{k} \lambda_{ijm}\left(\frac{f_{im}a_{jm}}{\lambda_{ijm}}\right)^2 = \sum_{m=1}^{k} \frac{f_{im}^2 a_{jm}^2}{\lambda_{ijm}}$$
$$(i = 1, \ldots, n;\ j = 1, \ldots, p) \tag{4.3}$$

where $\lambda_{ijm} > 0$ $(i = 1, \ldots, n;\ j = 1, \ldots, p;\ m = 1, \ldots, k)$ and $\sum_{m=1}^{k} \lambda_{ijm} = 1$ $(i = 1, \ldots, n;\ j = 1, \ldots, p)$. The equality is satisfied if and only if

$$\frac{f_{i1}a_{j1}}{\lambda_{ij1}} = \frac{f_{i2}a_{j2}}{\lambda_{ij2}} = \cdots = \frac{f_{ik}a_{jk}}{\lambda_{ijk}}\ (i = 1, \ldots, n;\ j = 1, \ldots, p). \tag{4.4}$$

If we define $c_{ij} = f_{im}a_{jm}/\lambda_{ijm}$ then $\lambda_{ijm} = f_{im}a_{jm}/c_{ij}$, and hence we have $c_{ij} = \sum_{m=1}^{k} f_{im}a_{jm}$ from $\sum_{m=1}^{k} \lambda_{ijm} = 1$. Therefore, (4.4) implies

$$\lambda_{ijm} = \frac{f_{im}a_{jm}}{\sum_{u=1}^{k} f_{iu}a_{ju}}\ (i = 1, \ldots, n;\ j = 1, \ldots, p;\ m = 1, \ldots, k). \tag{4.5}$$

Let f_{im}^* be current value of f_{im}. If we replace $(\sum_{m=1}^{k} f_{im}a_{jm})^2$ in (4.1) with $\sum_{m=1}^{k}(f_{im}^2 a_{jm}^2/\lambda_{ijm})$, that is, the final term in (4.3), and substitute

$$\lambda_{ijm} = \frac{f_{im}^* a_{jm}}{\sum_{u=1}^{k} f_{iu}^* a_{ju}}\ (i = 1, \ldots, n;\ j = 1, \ldots, p;\ m = 1, \ldots, k) \tag{4.6}$$

into $\sum_{m=1}^{k}(f_{im}^2 a_{jm}^2/\lambda_{ijm})$, we obtain the following auxiliary function of $Q_{\boldsymbol{F}}(\boldsymbol{F})$:

$$Q_{\boldsymbol{F}}^{\mathrm{aux}}(\boldsymbol{F}, \boldsymbol{F}^*) = -2\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}\sum_{m=1}^{k} f_{im}a_{jm} + \sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{m=1}^{k} f_{im}^2 a_{jm}\frac{\sum_{u=1}^{k} f_{iu}^* a_{ju}}{f_{im}^*}. \tag{4.7}$$

Of course it is satisfied that $Q_{\boldsymbol{F}}(\boldsymbol{F}) \leq Q_{\boldsymbol{F}}^{\mathrm{aux}}(\boldsymbol{F}, \boldsymbol{F}^*)$ for all $\boldsymbol{F}$ and $\boldsymbol{F}^*$ and $Q_{\boldsymbol{F}}(\boldsymbol{F}) = Q_{\boldsymbol{F}}^{\mathrm{aux}}(\boldsymbol{F}, \boldsymbol{F}^*)$ if and only if $\boldsymbol{F} = \boldsymbol{F}^*$. Then, we derive an optimal $\hat{f}_{im}$ that minimizes $Q_{\boldsymbol{F}}^{\mathrm{aux}}(\boldsymbol{F}, \boldsymbol{F}^*)$ with respect to f_{im}. The partial derivative of $Q_{\boldsymbol{F}}^{\mathrm{aux}}(\boldsymbol{F}, \boldsymbol{F}^*)$ with respect to f_{im} is as follows:

$$\frac{\partial Q_{\boldsymbol{F}}^{\mathrm{aux}}(\boldsymbol{F}, \boldsymbol{F}^*)}{\partial f_{im}} = -2\sum_{j=1}^{p} y_{ij}a_{jm} + 2\frac{f_{im}}{f_{im}^*}\sum_{j=1}^{p} a_{jm}\left(\sum_{u=1}^{k} f_{iu}^* a_{ju}\right). \tag{4.8}$$

The optimal $\hat{f}_{im}$ is obtained by setting (4.8) to 0 and solving the equation with respect to f_{im} as follows:

$$\hat{f}_{im} = f_{im}^*\frac{\sum_{j=1}^{p} y_{ij}a_{jm}}{\sum_{j=1}^{p}(\sum_{u=1}^{k} f_{iu}^* a_{ju})a_{jm}}. \tag{4.9}$$

(4.9) is an update rule of f_{im} given the current value f_{im}^* $(m = 1, \ldots, k)$ and a_{jm} $(j = 1, \ldots, p;\ m = 1, \ldots, k)$. The matrix form of this update rule (4.9) is as follows:

$$\hat{\boldsymbol{F}} = \boldsymbol{F}^* \odot \frac{\boldsymbol{Y}\boldsymbol{A}}{\boldsymbol{F}^*\boldsymbol{A}'\boldsymbol{A}}. \tag{4.10}$$

Update rule for $\boldsymbol{A}$

The minimization of the objective function (4.1) with respect to $\boldsymbol{A}$ takes the same form as (4.1); in addition, we can obtain this update rule in a manner similar to that of $\boldsymbol{F}$:

$$\hat{a}_{jm} = a^*_{jm} \frac{\sum_{i=1}^{n} y_{ij} f_{im}}{\sum_{i=1}^{n} (\sum_{u=1}^{k} f_{iu} a^*_{ju}) f_{im}}. \tag{4.11}$$

The matrix form of (4.11) is

$$\hat{\boldsymbol{A}} = \boldsymbol{A}^* \odot \frac{\boldsymbol{Y}'\boldsymbol{F}}{\boldsymbol{A}^*\boldsymbol{F}'\boldsymbol{F}}. \tag{4.12}$$

Algorithm

From (4.10), (4.12), and (3.31), the N2NMF algorithm is presented in Algorithm 3. Here, τ is a threshold to terminate the algorithm and υ is the maximum number of iterative cycles. Through Algorithm 3, the sequence of log-likelihood $L^{(0)}, L^{(1)}, \ldots$ is of course monotonically non-decreasing.

Algorithm 3 N2NMF Algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_+^{n \times p}$, $k \in \mathbb{N}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_+^{n \times k}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p \times k}$, $\tau > 0$, and $\upsilon \in \mathbb{N}$
$t \leftarrow 0$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
$(\sigma^{(t)})^2 \leftarrow \frac{1}{np}\|\boldsymbol{Y} - \boldsymbol{X}^{(t)}\|^2$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{N}}\left(y_{ij} \middle| x_{ij}^{(t)}, (\sigma^{(t)})^2\right)$
repeat
    $t \leftarrow t + 1$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{(t-1)} \odot \frac{\boldsymbol{Y}\boldsymbol{A}^{(t-1)}}{\boldsymbol{F}^{(t-1)}\boldsymbol{A}^{(t-1)\prime}\boldsymbol{A}^{(t-1)}}$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t-1)} \odot \frac{\boldsymbol{Y}'\boldsymbol{F}^{(t)}}{\boldsymbol{A}^{(t-1)}\boldsymbol{F}^{(t)\prime}\boldsymbol{F}^{(t)}}$
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $(\sigma^{(t)})^2 \leftarrow \frac{1}{np}\|\boldsymbol{Y} - \boldsymbol{X}^{(t)}\|^2$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{N}}\left(y_{ij} \middle| x_{ij}^{(t)}, (\sigma^{(t)})^2\right)$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(t)}$, and $(\sigma^{(t)})^2$
```

4.2 Poisson distribution

In this section we present details of two-factor NMF based on a Poisson distribution, named P2NMF. The objective function is defined as (3.35), where $\boldsymbol{\theta} = \{\boldsymbol{F}, \boldsymbol{A}\}$ and $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{A}'$.

Update rules

Below, we show the update rules of the parameters, $\boldsymbol{F}$ and $\boldsymbol{A}$.

Update rule for $\boldsymbol{F}$

From (3.35), the objective function to be minimized with respect to $\boldsymbol{F}$ is as follows:

$$Q_{\boldsymbol{F}}(\boldsymbol{F}) = \sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{m=1}^{k} f_{im}a_{jm} + \sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}\left\{ -\log\left\{ \sum_{m=1}^{k} f_{im}a_{jm} \right\}\right\}. \tag{4.13}$$

It is difficult to obtain an optimal f_{im} by differentiating this objective function with respect to f_{im} because the summation of f_{im} for m exists in the minus logarithm function in the objective function. Lee and Seung (2001) provides an update rule of this f_{im} in a manner similar to that of the method based on normal distribution described in Section 4.1 using the auxiliary function method. We can find that the following function

$$f(x) = -\log\{x\} \tag{4.14}$$

is in the second term of (4.13). This function is convex and we can use the Jensen inequality to derive an auxiliary function. Hence, we have

$$-\log\left\{ \sum_{m=1}^{k} f_{im}a_{jm} \right\} = -\log\left\{ \sum_{m=1}^{k} \lambda_{ijm}\frac{f_{im}a_{jm}}{\lambda_{ijm}} \right\} \leq -\sum_{m=1}^{k} \lambda_{ijm}\log\left\{ \frac{f_{im}a_{jm}}{\lambda_{ijm}} \right\} \tag{4.15}$$

with equality if and only if (4.5). If we replace $-\log\{\sum_{m=1}^{k} f_{im}a_{jm}\}$ in (4.13) with $-\sum_{m=1}^{k}\lambda_{ijm}\log\{f_{im}a_{jm}/\lambda_{ijm}\}$, that is, the final term in (4.15), and substitute (4.6) into $-\sum_{m=1}^{k}\lambda_{ijm}\log\{f_{im}a_{jm}/\lambda_{ijm}\}$, we obtain the following auxiliary function of $Q_{\boldsymbol{F}}(\boldsymbol{F})$:

$$\begin{aligned} Q_{\boldsymbol{F}}^{\text{aux}}(\boldsymbol{F}, \boldsymbol{F}^*) =& \sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{m=1}^{k} f_{im}a_{jm} \\ &- \sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}\sum_{m=1}^{k}\left(\frac{f_{im}^*a_{jm}}{\sum_{u=1}^{k} f_{iu}^*a_{ju}} \right)\log\left\{ \frac{f_{im}(\sum_{u=1}^{k} f_{iu}^*a_{ju})}{f_{im}^*} \right\}. \end{aligned} \tag{4.16}$$

It is satisfied that $Q_{\boldsymbol{F}}(\boldsymbol{F}) \leq Q_{\boldsymbol{F}}^{\text{aux}}(\boldsymbol{F}, \boldsymbol{F}^*)$ for all $\boldsymbol{F}$ and $\boldsymbol{F}^*$ and $Q_{\boldsymbol{F}}(\boldsymbol{F}) = Q_{\boldsymbol{F}}^{\text{aux}}(\boldsymbol{F}, \boldsymbol{F}^*)$ if and only if $\boldsymbol{F} = \boldsymbol{F}^*$. Then, we derive an optimal $\hat{f}_{im}$ that minimizes $Q_{\boldsymbol{F}}^{\text{aux}}(\boldsymbol{F}, \boldsymbol{F}^*)$ with respect to f_{im}. The partial derivative of $Q_{\boldsymbol{F}}^{\text{aux}}(\boldsymbol{F}, \boldsymbol{F}^*)$ with respect to f_{im} is as follows:

$$\frac{\partial Q_{\boldsymbol{F}}^{\text{aux}}(\boldsymbol{F}, \boldsymbol{F}^*)}{\partial f_{im}} = \sum_{j=1}^{p} a_{jm} - \frac{f_{im}^*}{f_{im}}\sum_{j=1}^{p}\left(\frac{y_{ij}}{\sum_{u=1}^{k} f_{iu}^*a_{ju}} \right)a_{jm}. \tag{4.17}$$

The optimal $\hat{f}_{im}$ is obtained by setting (4.17) to 0 and solving the equation with respect to f_{im} as follows:

$$\hat{f}_{im} = f_{im}^*\frac{\sum_{j=1}^{p}\{y_{ij}/(\sum_{u=1}^{k} f_{iu}^*a_{ju})\}a_{jm}}{\sum_{j=1}^{p} a_{jm}}. \tag{4.18}$$

The matrix form of this update rule (4.18) is as follows:

$$\hat{\boldsymbol{F}} = \boldsymbol{F}^* \odot \frac{\{\boldsymbol{Y}/(\boldsymbol{F}^*\boldsymbol{A}')\}\boldsymbol{A}}{\boldsymbol{E}_{n\times p}\boldsymbol{A}}. \tag{4.19}$$

Update rule for $\boldsymbol{A}$

The minimization of the objective function (3.35) with respect to $\boldsymbol{A}$ takes the same form as (4.13); in addition, we can obtain this update rule in a manner similar to that of $\boldsymbol{F}$:

$$\hat{a}_{jm} = a^*_{jm} \frac{\sum_{i=1}^{n} \{y_{ij}/(\sum_{u=1}^{k} f_{iu} a^*_{ju})\} f_{im}}{\sum_{i=1}^{n} f_{im}}. \tag{4.20}$$

The matrix form of (4.20) is

$$\hat{\boldsymbol{A}} = \boldsymbol{A}^* \odot \frac{\{\boldsymbol{Y}/(\boldsymbol{F}\boldsymbol{A}^{*\prime})\}'\boldsymbol{F}}{\boldsymbol{E}_{p\times n}\boldsymbol{F}}. \tag{4.21}$$

Algorithm

From (4.19) and (4.21), the P2NMF algorithm is presented in Algorithm 4.

Algorithm 4 P2NMF Algorithm

```
**Input** $\boldsymbol{Y} \in \mathbb{R}_+^{n\times p}$, $k \in \mathbb{N}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_+^{n\times k}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p\times k}$, $\tau > 0$, and $\upsilon \in \mathbb{N}$
$t \leftarrow 0$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{P}}\left(y_{ij} \middle| x_{ij}^{(t)}\right)$
**repeat**
    $t \leftarrow t + 1$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{(t-1)} \odot \frac{\{\boldsymbol{Y}/(\boldsymbol{F}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})\}\boldsymbol{A}^{(t-1)}}{\boldsymbol{E}_{n\times p}\boldsymbol{A}^{(t-1)}}$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t-1)} \odot \frac{\{\boldsymbol{Y}/(\boldsymbol{F}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})\}'\boldsymbol{F}^{(t)}}{\boldsymbol{E}_{p\times n}\boldsymbol{F}^{(t)}}$
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{P}}\left(y_{ij} \middle| x_{ij}^{(t)}\right)$
**until** $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
**Output** $\boldsymbol{F}^{(t)}$ and $\boldsymbol{A}^{(t)}$
```

4.3 Compound Poisson-gamma distribution

In this section we present details of two-factor NMF based on a compound Poisson-gamma distribution, named CP2NMF. The objective function is defined as (3.39), where $\boldsymbol{\theta} = \{\boldsymbol{F}, \boldsymbol{A}\}$ and $\boldsymbol{X} \coloneqq \boldsymbol{F}\boldsymbol{A}'$.

Update rules

We show the update rules of the parameters, $\boldsymbol{F}$ and $\boldsymbol{A}$. ϕ is obtained as described in Section 3.3.

Update rule for $\boldsymbol{F}$

From (3.39), the objective function to be minimized with respect to $\boldsymbol{F}$ is as follows:

$$Q_{\boldsymbol{F}}(\boldsymbol{F}) = \sum_{i=1}^{n}\sum_{j=1}^{p}\left(\frac{(\sum_{m=1}^{k} f_{im}a_{jm})^{\beta}}{\beta} - \frac{y_{ij}(\sum_{m=1}^{k} f_{im}a_{jm})^{\beta-1}}{\beta-1}\right). \tag{4.22}$$

It is difficult to obtain an optimal f_{im} by differentiating this objective function with respect to f_{im} because the summation of f_{im} for m exists in the β and $\beta - 1$ power functions. Actually, this objective function is the same objective function of two-factor NMF based on β-divergence, and Févotte and Idier (2011) and Nakano et al. (2010) derive the update rule of the factor matrices given some ranges of β. First, Févotte and Idier (2011) derived it for the case of $1 \leq \beta \leq 2$, and then, Nakano et al. (2010) provided it for the other case of β. Both of these research groups used an auxiliary function method to derive the update rule in the same manner as section 4.1 and section 4.2. However, Févotte and Idier (2011) only used the Jensen inequality to derive the update rule because both of the functions in the first and second terms of (4.22), that is, $f(x) = x^{\beta}/\beta$ and $f(x) = -x^{\beta-1}/(\beta-1)$, respectively, are convex if $1 \leq \beta \leq 2$. On the other hand, Nakano et al. (2010) pointed out that the former function is concave if $\beta < 1$ and the latter function is also concave if $\beta > 2$, in which case they not only used the Jensen inequality but also the inequality of a concave function in response to the value of β. In this section, we only present the case of $0 < \beta < 1$, which pertains to the compound Poisson-gamma distribution. The function in the first term, $f(x) = x^{\beta}/\beta$, is concave if $0 < \beta < 1$. Hence, we have

$$f(x) \leq f(\lambda) + f'(\lambda)(x - \lambda) \tag{4.23}$$

for any λ with equality if and only if $x = \lambda$. From this inequality, we have

$$\frac{(\sum_{m=1}^{k} f_{im}a_{jm})^{\beta}}{\beta} \leq \frac{\eta_{ij}^{\beta}}{\beta} + \eta_{ij}^{\beta-1}\left(\sum_{m=1}^{k} f_{im}a_{jm} - \eta_{ij}\right) = \eta_{ij}^{\beta-1}\sum_{m=1}^{k} f_{im}a_{jm} + \eta_{ij}^{\beta}\left(\frac{1}{\beta} - 1\right)$$
$$(i = 1, \ldots, n;\ j = 1, \ldots, p). \tag{4.24}$$

The equality is satisfied if and only if

$$\eta_{ij} = \sum_{m=1}^{k} f_{im}a_{jm}\ (i = 1, \ldots, n;\ j = 1, \ldots, p). \tag{4.25}$$

On the other hand, the function in the second term, $f(x) = -x^{\beta-1}/(\beta-1)$, is convex if $0 < \beta < 1$, and hence we can use the Jensen inequality. Therefore, we have

$$\begin{aligned} -\frac{(\sum_{m=1}^{k} f_{im}a_{jm})^{\beta-1}}{\beta-1} &= -\frac{1}{\beta-1}\left(\sum_{m=1}^{k}\lambda_{ijm}\frac{f_{im}a_{jm}}{\lambda_{ijm}}\right)^{\beta-1} \\ &\leq -\frac{1}{\beta-1}\sum_{m=1}^{k}\lambda_{ijm}\left(\frac{f_{im}a_{jm}}{\lambda_{ijm}}\right)^{\beta-1} \\ &= -\frac{1}{\beta-1}\sum_{m=1}^{k}\left(\frac{f_{im}^{\beta-1}a_{jm}^{\beta-1}}{\lambda_{ijm}^{\beta-2}}\right)\ (i = 1, \ldots, n;\ j = 1, \ldots, p). \end{aligned} \tag{4.26}$$

The equality is satisfied if and only if (4.5). If we replace

$$\frac{(\sum_{m=1}^{k} f_{im}a_{jm})^{\beta}}{\beta} \quad \text{and} \quad -\frac{(\sum_{m=1}^{k} f_{im}a_{jm})^{\beta-1}}{\beta-1}$$

in (4.22) with

$$\eta_{ij}^{\beta-1}\sum_{m=1}^{k} f_{im}a_{jm} + \eta_{ij}^{\beta}\left(\frac{1}{\beta-1}\right) \quad \text{and} \quad -\frac{1}{\beta-1}\sum_{m=1}^{k}\left(\frac{f_{im}^{\beta-1}a_{jm}^{\beta-1}}{\lambda_{ijm}^{\beta-2}}\right),$$

that is, the final term in (4.24) and (4.26), respectively, and substitute

$$\eta_{ij} = \sum_{m=1}^{k} f_{im}^{*}a_{jm} \; (i = 1, \ldots, n; \; j = 1, \ldots, p) \tag{4.27}$$

and (4.6) into the replaced (4.22), we obtain the following auxiliary function of $Q_{\boldsymbol{F}}(\boldsymbol{F})$:

$$\begin{aligned} Q_{\boldsymbol{F}}^{\text{aux}}(\boldsymbol{F}, \boldsymbol{F}^{*}) = \sum_{i=1}^{n}\sum_{j=1}^{p}\Bigg\{ &\left(\sum_{u=1}^{k} f_{iu}^{*}a_{ju}\right)^{\beta-1}\sum_{m=1}^{k} f_{im}a_{jm} + \left(\sum_{u=1}^{k} f_{iu}^{*}a_{ju}\right)^{\beta}\left(\frac{1}{\beta}-1\right) \\ &- \frac{1}{\beta-1}y_{ij}\left(\sum_{u=1}^{k} f_{iu}^{*}a_{ju}\right)^{\beta-2}\sum_{m=1}^{k}\left(\frac{f_{im}^{\beta-1}a_{jm}}{f_{im}^{*\beta-2}}\right)\Bigg\} \end{aligned} \tag{4.28}$$

Then, we derive an optimal $\hat{f}_{im}$ that minimizes $Q_{\boldsymbol{F}}^{\text{aux}}(\boldsymbol{F}, \boldsymbol{F}^{*})$ with respect to f_{im}. The partial derivative of $Q_{\boldsymbol{F}}^{\text{aux}}(\boldsymbol{F}, \boldsymbol{F}^{*})$ with respect to f_{im} is as follows:

$$\frac{\partial Q_{\boldsymbol{F}}^{\text{aux}}(\boldsymbol{F}, \boldsymbol{F}^{*})}{\partial f_{im}} = \sum_{j=1}^{p}\left(\sum_{u=1}^{k} f_{iu}^{*}a_{ju}\right)^{\beta-1} a_{jm} - f_{im}^{\beta-2}\sum_{j=1}^{p} y_{ij}\left(\sum_{u=1}^{k} f_{iu}^{*}a_{ju}\right)^{\beta-2}\frac{a_{jm}}{f_{im}^{*\beta-2}}. \tag{4.29}$$

The optimal $\hat{f}_{im}$ is obtained by setting (4.29) to 0 and solving the equation with respect to f_{im} as follows:

$$\hat{f}_{im} = f_{im}^{*}\left\{\frac{\sum_{j=1}^{p} y_{ij}(\sum_{u=1}^{k} f_{iu}^{*}a_{ju})^{\beta-2}a_{jm}}{\sum_{j=1}^{p}(\sum_{u=1}^{k} f_{iu}^{*}a_{ju})^{\beta-1}a_{jm}}\right\}^{\frac{1}{2-\beta}}. \tag{4.30}$$

The matrix form of this update rule (4.30) is as follows:

$$\hat{\boldsymbol{F}} = \boldsymbol{F}^{*} \odot \left[\frac{\{\boldsymbol{Y} \odot (\boldsymbol{F}^{*}\boldsymbol{A}')^{\beta-2}\}\boldsymbol{A}}{\{(\boldsymbol{F}^{*}\boldsymbol{A}')^{\beta-1}\}\boldsymbol{A}}\right]^{\frac{1}{two-\beta}}. \tag{4.31}$$

Update rule for $\boldsymbol{A}$

The minimization of the objective function (3.39) with respect to $\boldsymbol{A}$ takes the same form as (4.22); in addition, we can obtain this update rule in a manner similar to that of $\boldsymbol{F}$:

$$\hat{a}_{jm} = a_{jm}^{*}\left\{\frac{\sum_{i=1}^{n} y_{ij}(\sum_{u=1}^{k} f_{iu}a_{ju}^{*})^{\beta-2}f_{im}}{\sum_{i=1}^{n}(\sum_{u=1}^{k} f_{iu}a_{ju}^{*})^{\beta-1}f_{im}}\right\}^{\frac{1}{2-\beta}}. \tag{4.32}$$

The matrix form of (4.32) is

$$\hat{\boldsymbol{A}} = \boldsymbol{A}^* \odot \left[\frac{\{\boldsymbol{Y}' \odot (\boldsymbol{A}^* \boldsymbol{F}')^{\beta-2}\}\boldsymbol{F}}{\{(\boldsymbol{A}^* \boldsymbol{F}')^{\beta-1}\}\boldsymbol{F}} \right]^{\frac{1}{2-\beta}}. \tag{4.33}$$

Together (4.31) and (4.33) indicate that CP2NMF is a generalization of N2NMF or P2NMF. When $\beta = 1$, these update rules are equivalent to (4.19) and (4.21), respectively. On the other hand, when $\beta = 2$, the formulas of these update rules are equivalent to (4.10) and (4.12), respectively except for the exponent part, $1/(2-\beta)$. Nakano et al. (2010) found that their formulas are the same for all β except for the exponent part.

Algorithm

The CP2NMF algorithm, which is derived from (4.31), (4.33), and the discussion in Section 3.3 about optimal ϕ, is presented in Algorithm 5. Note that we limit the number of times ϕ is updated to prevent the computational time from becoming excessively large. We update ϕ for the first δ iterations; then, for the remaining iterative cycles, we update it at every κ-th iteration.

Algorithm 5 CP2NMF Algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_+^{n\times p}$, $\beta \in (0,1)$, $k \in \mathbb{N}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_+^{n\times k}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p\times k}$, $\phi^{(0)} > 0$, $\tau > 0$, $\upsilon \in \mathbb{N}$, $\delta \in \mathbb{N}$, and $\kappa \in \mathbb{N}$
$t \leftarrow 0$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{CP}}\left(y_{ij} \middle| x_{ij}^{(t)}, \phi^{(t)}, \beta\right)$
repeat
    $t \leftarrow t + 1$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{(t-1)} \odot \left[\frac{\{\boldsymbol{Y} \odot (\boldsymbol{F}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-2}\}\boldsymbol{A}^{(t-1)}}{\{(\boldsymbol{F}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-1}\}\boldsymbol{A}^{(t-1)}}\right]^{\frac{1}{2-\beta}}$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t-1)} \odot \left[\frac{\{\boldsymbol{Y}' \odot (\boldsymbol{A}^{(t-1)}\boldsymbol{F}^{(t)\prime})^{\beta-2}\}\boldsymbol{F}^{(t)}}{\{(\boldsymbol{A}^{(t-1)}\boldsymbol{F}^{(t)\prime})^{\beta-1}\}\boldsymbol{F}^{(t)}}\right]^{\frac{1}{2-\beta}}$
    if $t \le \delta$ or $t \bmod \kappa = 0$ then
        $\phi^{(t)}$ is obtained as the optimal $\phi$ that optimizes $Q_\phi(\phi)$ given $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(t)}$, and $\beta$ using the BFGS quasi-Newton method with constraints $\phi > 0$
    end if
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{CP}}\left(y_{ij} \middle| x_{ij}^{(t)}, \phi^{(t)}, \beta\right)$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(t)}$, and $\phi^{(t)}$
```

4.4 Zero-inflated compound Poisson-gamma distribution

In this section we present details of two-factor NMF based on a zero-inflated compound Poisson-gamma distribution, named ZICP2NMF. The objective function is defined as (3.44), where $\boldsymbol{\theta} = \{\boldsymbol{F}, \boldsymbol{A}\}$ and $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{A}'$. However, the update rule of $\boldsymbol{F}$ and $\boldsymbol{A}$ is obtained as these optimizers, which minimize (3.51). This method was proposed by Abe and Yadohisa (2016).

Update rules

We show the update rule of the parameters, $\boldsymbol{F}$ and $\boldsymbol{A}$. The update rules of $\hat{\boldsymbol{Z}}$ and w is obtained as (3.52) and (3.55). ϕ is obtained as described in Section 3.3. The estimation of w, $\boldsymbol{F}$, $\boldsymbol{A}$, and ϕ is known as the Mstep in the EM algorithm.

Update rule for $\boldsymbol{F}$

From (3.51), the objective function to be minimized with respect to $\boldsymbol{F}$ is as follows:

$$Q_{\boldsymbol{F}}(\boldsymbol{F}) = \frac{1}{\phi}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\frac{\hat{z}_{ij}^{*}(\sum_{m=1}^{k} f_{im}a_{jm})^{\beta}}{\beta} - \frac{\hat{z}_{ij}^{*}y_{ij}(\sum_{m=1}^{k} f_{im}a_{jm})^{\beta-1}}{\beta-1}\right), \tag{4.34}$$

where $\hat{z}_{ij}^{*} := 1 - z_{ij}^{*}$. This objective function is similar to that of (4.22); however, the weight value in this case is $\hat{z}_{ij}^{*}$. Since z_{ij}^{*} is positive, we can use the Jensen inequality to convert the second term into a function existing in the upper bound of the second term. The first term can also be converted into a function existing in the upper bound of the first term using inequality (4.23). Hence, the update rule of $\boldsymbol{F}$ is derived in the same manner as described in Section 4.3. These update rules of both the element and matrix forms are as follows:

$$\hat{f}_{im} = f_{im}^{*}\left\{\frac{\sum_{j=1}^{p}\hat{z}_{ij}^{*}y_{ij}(\sum_{u=1}^{k} f_{iu}^{*}a_{ju})^{\beta-2}a_{jm}}{\sum_{j=1}^{p}\hat{z}_{ij}^{*}(\sum_{u=1}^{k} f_{iu}^{*}a_{ju})^{\beta-1}a_{jm}}\right\}^{\frac{1}{2-\beta}}, \tag{4.35}$$

$$\hat{\boldsymbol{F}} = \boldsymbol{F}^{*} \odot \left[\frac{\{\hat{\boldsymbol{Z}}^{*} \odot \boldsymbol{Y} \odot (\boldsymbol{F}^{*}\boldsymbol{A}')^{\beta-2}\}\boldsymbol{A}}{\{\hat{\boldsymbol{Z}}^{*} \odot (\boldsymbol{F}^{*}\boldsymbol{A}')^{\beta-1}\}\boldsymbol{A}}\right]^{\frac{1}{2-\beta}}. \tag{4.36}$$

Update rule for $\boldsymbol{A}$

The minimization of the objective function (3.51) with respect to $\boldsymbol{A}$ takes the same form as (4.34); in addition, we can obtain this update rule in a manner similar to that of $\boldsymbol{F}$:

$$\hat{a}_{jm} = a_{jm}^{*}\left\{\frac{\sum_{i=1}^{n}\hat{z}_{ij}^{*}y_{ij}(\sum_{u=1}^{k} f_{iu}a_{ju}^{*})^{\beta-2}f_{im}}{\sum_{i=1}^{n}\hat{z}_{ij}^{*}(\sum_{u=1}^{k} f_{iu}a_{ju}^{*})^{\beta-1}f_{im}}\right\}^{\frac{1}{2-\beta}}. \tag{4.37}$$

The matrix form of (4.37) is

$$\hat{\boldsymbol{A}} = \boldsymbol{A}^{*} \odot \left[\frac{\{\hat{\boldsymbol{Z}}^{*} \odot \boldsymbol{Y}' \odot (\boldsymbol{A}^{*}\boldsymbol{F}')^{\beta-2}\}\boldsymbol{F}}{\{\hat{\boldsymbol{Z}}^{*} \odot (\boldsymbol{A}^{*}\boldsymbol{F}')^{\beta-1}\}\boldsymbol{F}}\right]^{\frac{1}{2-\beta}}. \tag{4.38}$$

When $\hat{z}_{ij} = 0$ for all i and j, that is, all elements of $\boldsymbol{Y}$ are compound Poisson-gamma distributed, (4.36) and (4.38) are the same as (4.31) and (4.33), respectively. Therefore, ZICP2NMF is a generalized method of CP2NMF.

Algorithm

The ZICP2NMF algorithm, which is based on (3.52), (3.55), (4.36), (4.38), and the discussion in Section 3.3 about an optimal ϕ, is presented in Algorithm 6. Note that we limit the number of times ϕ is updated in the same manner as for CP2NMF.

Algorithm 6 ZICP2NMF Algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_+^{n \times p}$, $\beta \in (0,1)$, $k \in \mathbb{N}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_+^{n \times k}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p \times k}$, $w^{(0)} \in (0,1)$, $\phi^{(0)} > 0$, $\tau > 0$, $\upsilon \in \mathbb{N}$, $\delta \in \mathbb{N}$, and $\kappa \in \mathbb{N}$
$t \leftarrow 0$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
$z_{ij}^{(t)} \leftarrow \begin{cases} \dfrac{w^{(t)}}{w^{(t)} + (1 - w^{(t)})h(0, \phi^{(t)}, \beta)\exp\{-(x_{ij}^{(t)})^{\beta}/(\phi^{(t)}\beta)\}} & \text{if } y_{ij} = 0 \\ 0 & \text{if } y_{ij} \neq 0 \end{cases}$
$(i = 1, \ldots, n;\ j = 1, \ldots, p)$
$\boldsymbol{Z}^{*(t)} \leftarrow \boldsymbol{E}_{n \times p} - \boldsymbol{Z}^{(t)}$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log\left\{w^{(t)}I(y_{ij} = 0) + (1 - w^{(t)})f_{\mathrm{CP}}(y_{ij}|x_{ij}^{(t)}, \phi^{(t)}, \beta)\right\}$
repeat
    $t \leftarrow t + 1$
    $w^{(t)} \leftarrow \dfrac{\sum_{i=1}^{n}\sum_{j=1}^{p} z_{ij}^{(t-1)}}{np}$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{(t-1)} \odot \left[\dfrac{\{\boldsymbol{Z}^{*(t-1)} \odot \boldsymbol{Y} \odot (\boldsymbol{F}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-2}\}\boldsymbol{A}^{(t-1)}}{\{\boldsymbol{Z}^{*(t-1)} \odot (\boldsymbol{F}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-1}\}\boldsymbol{A}^{(t-1)}}\right]^{\frac{1}{2-\beta}}$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t-1)} \odot \left[\dfrac{\{\boldsymbol{Z}^{*(t-1)} \odot \boldsymbol{Y}' \odot (\boldsymbol{A}^{(t-1)}\boldsymbol{F}^{(t)\prime})^{\beta-2}\}\boldsymbol{F}^{(t)}}{\{\boldsymbol{Z}^{*(t-1)} \odot (\boldsymbol{A}^{(t-1)}\boldsymbol{F}^{(t)\prime})^{\beta-1}\}\boldsymbol{F}^{(t)}}\right]^{\frac{1}{2-\beta}}$
    if $t \leq \delta$ or $t \bmod \kappa = 0$ then
        $\phi^{(t)}$ is obtained as the optimal $\phi$ that optimizes $Q_{\phi}(\phi)$ given $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(t)}$, $\boldsymbol{Z}^{*(t-1)}$, and $\beta$ using the BFGS quasi-Newton method with constraints $\phi > 0$
    end if
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $z_{ij}^{(t)} \leftarrow \begin{cases} \dfrac{w^{(t)}}{w^{(t)} + (1 - w^{(t)})h(0, \phi^{(t)}, \beta)\exp\{-(x_{ij}^{(t)})^{\beta}/(\phi^{(t)}\beta)\}} & \text{if } y_{ij} = 0 \\ 0 & \text{if } y_{ij} \neq 0 \end{cases}$
    $(i = 1, \ldots, n;\ j = 1, \ldots, p)$
    $\boldsymbol{Z}^{*(t)} \leftarrow \boldsymbol{E}_{n \times p} - \boldsymbol{Z}^{(t)}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log\left\{w^{(t)}I(y_{ij} = 0) + (1 - w^{(t)})f_{\mathrm{CP}}(y_{ij}|x_{ij}^{(t)}, \phi^{(t)}, \beta)\right\}$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(\upsilon)}$, $\boldsymbol{Z}^{(t)}$, and $\phi^{(\nu)}$
```

Chapter 5

Three-factor NMF

In this chapter, we describe four three-factor NMFs. The aim of all methods in this chapter is to obtain estimates of the factor matrices, $\boldsymbol{F} \in \mathbb{R}_+^{n\times k}$, $\boldsymbol{S} \in \mathbb{R}_+^{k\times \ell}$, and $\boldsymbol{A} \in \mathbb{R}_+^{p\times \ell}$, such that $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{S}\boldsymbol{A}'$ is approximated to a given data matrix $\boldsymbol{Y} \in \mathbb{R}_+^{n\times p}$. Basically, the update rules of $\boldsymbol{F}$ and $\boldsymbol{A}$ have the same form as those of the two-factor NMFs described in Chapter 4. On the other hand, the center factor matrix $\boldsymbol{S}$ is not as straightforward.

5.1 Normal distribution

In this section we present details of the three-factor NMF based on a normal distribution, named N3NMF. The objective function is defined as (3.30), where $\boldsymbol{\theta} = \{\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}\}$ and $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{S}\boldsymbol{A}'$.

Update rules

We show the update rules of the parameters, $\boldsymbol{F}$, $\boldsymbol{S}$, and $\boldsymbol{A}$. Note that the update rule of σ^2 is (3.31).

Update rule for $\boldsymbol{F}$

If we treat $\boldsymbol{A}\boldsymbol{S}'$ as the right hand factor matrix in a two-factor NMF, the form of the objective function with respect to $\boldsymbol{F}$ is the same as that of (4.1). Hence, we can obtain the update rule of $\boldsymbol{F}$ in the same form as (4.9) and (4.10) as follows:

$$\hat{f}_{im} = f_{im}^* \frac{\sum_{j=1}^p y_{ij} \sum_{q=1}^\ell s_{mq} a_{jq}}{\sum_{j=1}^p (\sum_{r=1}^k \sum_{c=1}^\ell f_{ir}^* s_{rc} a_{jc}) \sum_{q=1}^\ell s_{mq} a_{jq}}, \tag{5.1}$$

$$\hat{\boldsymbol{F}} = \boldsymbol{F}^* \odot \frac{\boldsymbol{Y}\boldsymbol{A}\boldsymbol{S}'}{\boldsymbol{F}^*\boldsymbol{S}\boldsymbol{A}'\boldsymbol{A}\boldsymbol{S}'}. \tag{5.2}$$

Update rule for $\boldsymbol{A}$

We can obtain an update rule of $\boldsymbol{A}$ as a same form of $\boldsymbol{F}$:

$$\hat{a}_{jm} = a_{jm}^* \frac{\sum_{i=1}^n y_{ij} \sum_{m=1}^k f_{im} s_{mq}}{\sum_{i=1}^n (\sum_{r=1}^k \sum_{c=1}^\ell f_{ir} s_{rc} a_{jc}^*) \sum_{m=1}^k f_{im} s_{mq}}. \tag{5.3}$$

$$\hat{\boldsymbol{A}} = \boldsymbol{A}^* \odot \frac{\boldsymbol{Y}'\boldsymbol{F}\boldsymbol{S}}{\boldsymbol{A}^*\boldsymbol{S}'\boldsymbol{F}'\boldsymbol{F}\boldsymbol{S}}. \tag{5.4}$$

Update rule for $\boldsymbol{S}$

From (3.30), the objective function to be minimized with respect to $\boldsymbol{S}$ is as follows:

$$Q_{\boldsymbol{S}}(\boldsymbol{S}) = -2\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq} + \sum_{i=1}^{n}\sum_{j=1}^{p}\left(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq}\right)^2. \tag{5.5}$$

It is difficult to obtain an optimal s_{mq} by differentiating this objective function with respect to s_{mq} because the summation of s_{mq} for m and q exists in the square function in (5.5). However, we can use the auxiliary function method in a manner similar to N2NMF because the second term in (5.5) contains a square function that is convex. From the Jensen inequality, we have

$$\begin{aligned}\left(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq}\right)^2 &= \left(\sum_{m=1}^{k}\sum_{q=1}^{\ell} \lambda_{ijmq}\frac{f_{im}s_{mq}a_{jq}}{\lambda_{ijmq}}\right)^2 \\ &\leq \sum_{m=1}^{k}\sum_{q=1}^{\ell} \lambda_{ijmq}\left(\frac{f_{im}s_{mq}a_{jq}}{\lambda_{ijmq}}\right)^2 = \sum_{m=1}^{k}\sum_{q=1}^{\ell}\frac{f_{im}^2 s_{mq}^2 a_{jq}^2}{\lambda_{ijmq}} \\ &(i = 1, \ldots, n;\ j = 1, \ldots, p),\end{aligned} \tag{5.6}$$

where

$$\lambda_{ijmq} > 0\ (i = 1, \ldots, n;\ j = 1, \ldots, p;\ m = 1, \ldots, k;\ q = 1, \ldots, \ell)$$
$$\text{and}\quad \sum_{m=1}^{k}\sum_{q=1}^{\ell}\lambda_{ijmq} = 1\ (i = 1, \ldots, n;\ j = 1, \ldots, p).$$

The equality is satisfied if and only if

$$\begin{aligned}&\frac{f_{i1}s_{11}a_{j1}}{\lambda_{ij11}} = \frac{f_{i1}s_{12}a_{j2}}{\lambda_{ij12}} = \cdots = \frac{f_{i1}s_{1\ell}a_{j\ell}}{\lambda_{ij1\ell}} \\ =&\frac{f_{i2}s_{21}a_{j1}}{\lambda_{ij21}} = \frac{f_{i2}s_{22}a_{j2}}{\lambda_{ij22}} = \cdots = \frac{f_{i2}s_{2\ell}a_{j\ell}}{\lambda_{ij2\ell}} \\ = \cdots =&\frac{f_{ik}s_{k1}a_{j1}}{\lambda_{ijk1}} = \frac{f_{ik}s_{k2}a_{j2}}{\lambda_{ijk2}} = \cdots = \frac{f_{ik}s_{k\ell}a_{j\ell}}{\lambda_{ijk\ell}}\ (i = 1, \ldots, n;\ j = 1, \ldots, p).\end{aligned} \tag{5.7}$$

If we define $c_{ij} = f_{im}s_{mq}a_{jq}/\lambda_{ijmq}$ then $\lambda_{ijmq} = f_{im}s_{mq}a_{jq}/c_{ij}$ and hence we have

$$c_{ij} = \sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq}$$

from $\sum_{m=1}^{k}\sum_{q=1}^{\ell}\lambda_{ijmq} = 1$. Therefore, (5.7) implies

$$\lambda_{ijmq} = \frac{f_{im}s_{mq}a_{jq}}{\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s_{rc}a_{jc}}\ (i = 1, \ldots, n;\ j = 1, \ldots, p;\ m = 1, \ldots, k;\ q = 1, \ldots, \ell). \tag{5.8}$$

Let s^*_{mq} be the current value of s_{mq}. If we replace $(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq})^2$ in (5.5) with $\sum_{m=1}^{k}\sum_{q=1}^{\ell} f^2_{im}s^2_{mq}a^2_{jq}/\lambda_{ijmq}$, that is, the final term in (5.6), and substitute

$$\lambda_{ijmq} = \frac{f_{im}s^*_{mq}a_{jq}}{\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s^*_{rc}a_{jc}} \; (i = 1, \dots, n;\; j = 1, \dots, p;\; m = 1, \dots, k;\; q = 1, \dots, \ell) \tag{5.9}$$

into $\sum_{m=1}^{k}\sum_{q=1}^{\ell} f^2_{im}s^2_{mq}a^2_{jq}/\lambda_{ijmq}$, we obtain the following auxiliary function of $Q_{\boldsymbol{S}}(\boldsymbol{S})$:

$$\begin{aligned} Q^{\mathrm{aux}}_{\boldsymbol{S}}(\boldsymbol{S}, \boldsymbol{S}^*) = & -2\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq} \\ & + \sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s^2_{mq}a_{jq}\frac{\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s^*_{rc}a_{jc}}{s^*_{mq}}. \end{aligned} \tag{5.10}$$

Of course it is satisfied that $Q_{\boldsymbol{S}}(\boldsymbol{S}) \leq Q^{\mathrm{aux}}_{\boldsymbol{S}}(\boldsymbol{S}, \boldsymbol{S}^*)$ for all $\boldsymbol{S}$ and $\boldsymbol{F}^S$ and $Q_{\boldsymbol{S}}(\boldsymbol{S}) = Q^{\mathrm{aux}}_{\boldsymbol{S}}(\boldsymbol{S}, \boldsymbol{S}^*)$ if and only if $\boldsymbol{S} = \boldsymbol{S}^*$. Then, we derive an optimal $\hat{s}_{mq}$ that minimizes $Q^{\mathrm{aux}}_{\boldsymbol{S}}(\boldsymbol{S}, \boldsymbol{S}^*)$ with respect to s_{mq}. The partial derivative of $Q^{\mathrm{aux}}_{\boldsymbol{S}}(\boldsymbol{S}, \boldsymbol{S}^*)$ with respect to s_{mq} is as follows:

$$\frac{\partial Q^{\mathrm{aux}}_{\boldsymbol{S}}(\boldsymbol{S}, \boldsymbol{S}^*)}{\partial s_{mq}} = -2\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}f_{im}a_{jq} + 2\frac{s_{mq}}{s^*_{mq}}\sum_{i=1}^{n}\sum_{j=1}^{p} f_{im}a_{jq}\left(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s^*_{rc}a_{jc}\right). \tag{5.11}$$

The optimal $\hat{s}_{mq}$ is obtained by setting (5.11) to 0 and solving the equation with respect to s_{mq} as follows:

$$\hat{s}_{mq} = s^*_{mq}\frac{\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}f_{im}a_{jq}}{\sum_{i=1}^{n}\sum_{j=1}^{p}(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s^*_{rc}a_{jc})f_{im}a_{jm}}. \tag{5.12}$$

(5.12) is an update rule of s_{mq} given its current value s^*_{mq} $(m = 1, \dots, k;\; q = 1, \dots, \ell)$, f_{im} $(i = 1, \dots, n;\; m = 1, \dots, k)$, a_{jq} $(j = 1, \dots, p;\; q = 1, \dots, \ell)$. The matrix form of this update rule (5.1) is as follows:

$$\mathrm{vec}(\hat{\boldsymbol{S}}) = \mathrm{vec}(\boldsymbol{S}^*) \odot \frac{(\boldsymbol{A} \otimes \boldsymbol{F})'\mathrm{vec}(\boldsymbol{Y})}{(\boldsymbol{A} \otimes \boldsymbol{F})'(\boldsymbol{A} \otimes \boldsymbol{F})\mathrm{vec}(\boldsymbol{S})}. \tag{5.13}$$

(5.13) is derived from another perspective. The approximation $\boldsymbol{Y} \approx \boldsymbol{FSA'}$ can be rewritten using the vectorization form as $\mathrm{vec}(\boldsymbol{Y}) \approx \mathrm{vec}(\boldsymbol{FSA'}) = (\boldsymbol{A} \otimes \boldsymbol{F})\mathrm{vec}(\boldsymbol{S})$. On the other hand, when we focus on the column vector of $\boldsymbol{Y}$, that is, $\boldsymbol{y}_{(j)}$, we can rewrite the approximation equation of N2NMF as $\boldsymbol{y}_{(j)} \approx \boldsymbol{F}\boldsymbol{a}_j$, and hence the update equation of $\boldsymbol{a}_j$ is described as

$$\hat{\boldsymbol{a}}_j = \boldsymbol{a}^*_j \odot \frac{\boldsymbol{F}'\boldsymbol{y}_{(j)}}{\boldsymbol{F}'\boldsymbol{F}\boldsymbol{a}^*_j} \tag{5.14}$$

from (4.12). A comparison of the two approximation,

$$\mathrm{vec}(\boldsymbol{Y}) \approx \mathrm{vec}(\boldsymbol{FSA'}) = (\boldsymbol{A} \otimes \boldsymbol{F})\mathrm{vec}(\boldsymbol{S})$$

of N3NMF and $\boldsymbol{y}_{(j)} \approx \boldsymbol{F}\boldsymbol{a}_j$ of N2NMF, and (5.14) enable us to derive (5.13).

Algorithm

The N3NMF algorithm, which is based on (5.2), (5.4), (5.13), and (3.31), is presented in Algorithm 7.

Algorithm 7 N3NMF Algorithm

```
**Input** $\boldsymbol{Y} \in \mathbb{R}_{+}^{n\times p}$, $k \in \mathbb{N}$, $\ell \in \mathbb{N}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_{+}^{n\times k}$, $\boldsymbol{S}^{(0)} \in \mathbb{R}_{+}^{k\times \ell}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_{+}^{p\times \ell}$, $\tau > 0$, and $\upsilon \in \mathbb{N}$
$t \leftarrow 0$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
$(\sigma^{(t)})^2 \leftarrow \frac{1}{np}\|\boldsymbol{Y} - \boldsymbol{X}^{(t)}\|^2$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{N}}\left(y_{ij}\middle| x_{ij}^{(t)}, (\sigma^{(t)})^2\right)$
**repeat**
    $t \leftarrow t + 1$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{(t-1)} \odot \frac{\boldsymbol{Y}\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}}{\boldsymbol{F}^{(t-1)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime}\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}}$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t-1)} \odot \frac{\boldsymbol{Y}'\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}}{\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}\boldsymbol{F}^{(t)\prime}\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}}$
    $\mathrm{vec}(\boldsymbol{S}^{(t)}) \leftarrow \mathrm{vec}(\boldsymbol{S}^{(t-1)}) \odot \frac{(\boldsymbol{A}^{(t)} \otimes \boldsymbol{F}^{(t)})'\mathrm{vec}(\boldsymbol{Y})}{(\boldsymbol{A}^{(t)} \otimes \boldsymbol{F}^{(t)})'(\boldsymbol{A}^{(t)} \otimes \boldsymbol{F}^{(t)})\mathrm{vec}(\boldsymbol{S}^{(t-1)})}$
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $(\sigma^{(t)})^2 \leftarrow \frac{1}{np}\|\boldsymbol{Y} - \boldsymbol{X}^{(t)}\|^2$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{N}}\left(y_{ij}\middle| x_{ij}^{(t)}, (\sigma^{(t)})^2\right)$
**until** $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
**Output** $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$, $\boldsymbol{A}^{(t)}$, and $(\sigma^{(t)})^2$
```

5.2 Poisson distribution

In this section we present details of the three-factor NMF based on a Poisson distribution, named P3NMF. The objective function is defined as (3.35), where $\boldsymbol{\theta} = \{\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}\}$ and $\boldsymbol{X} \coloneqq \boldsymbol{F}\boldsymbol{S}\boldsymbol{A}'$.

Update rules

Below, we show the update rules of the parameters, $\boldsymbol{F}$, $\boldsymbol{S}$, and $\boldsymbol{A}$.

Update rule for $\boldsymbol{F}$

As N3NMF, the update rule of $\boldsymbol{F}$ is given in the same form of (4.18) and (4.19) as follows:

$$\hat{f}_{im} = f_{im}^{*}\frac{\sum_{j=1}^{p}\{y_{ij}/(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}^{*}s_{rc}a_{jc})\}\sum_{q=1}^{\ell} s_{mq}a_{jq}}{\sum_{j=1}^{p}\sum_{q=1}^{\ell} s_{mq}a_{jq}}, \tag{5.15}$$

$$\hat{\boldsymbol{F}} = \boldsymbol{F}^* \odot \frac{\{\boldsymbol{Y}/(\boldsymbol{F}^*\boldsymbol{S}\boldsymbol{A}')\}\boldsymbol{A}\boldsymbol{S}'}{\boldsymbol{E}_{n\times p}\boldsymbol{A}\boldsymbol{S}'}. \tag{5.16}$$

Update rule for $\boldsymbol{A}$

We can obtain an update rule of $\boldsymbol{A}$ in a same form as $\boldsymbol{F}$:

$$\hat{a}_{jm} = a^*_{jm}\frac{\sum_{i=1}^{n}\{y_{ij}/(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ru}s_{rc}a^*_{jc})\}\sum_{m=1}^{k} f_{im}s_{mq}}{\sum_{i=1}^{n}\sum_{m=1}^{k} f_{im}s_{mq}}, \tag{5.17}$$

$$\hat{\boldsymbol{A}} = \boldsymbol{A}^* \odot \frac{\{\boldsymbol{Y}/(\boldsymbol{F}\boldsymbol{S}\boldsymbol{A}^{*\prime})\}'\boldsymbol{F}\boldsymbol{S}}{\boldsymbol{E}_{p\times n}\boldsymbol{F}\boldsymbol{S}}. \tag{5.18}$$

Update rule for $\boldsymbol{S}$

From (3.35), the objective function to be minimized with respect to $\boldsymbol{S}$ is as follows:

$$Q_{\boldsymbol{S}}(\boldsymbol{S}) = \sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq} + \sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}\left\{-\log\left\{\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq}\right\}\right\}. \tag{5.19}$$

It is difficult to obtain an optimal s_{mq} by differentiating this objective function with respect to s_{mq} because the summation of s_{mq} for m and q exists in the minus logarithm function in the objective function. However, the auxiliary function method is available to derive the update rule of $\boldsymbol{S}$. As P2NMF, we can apply the Jensen inequality to the second term of (5.19) that contains a minus logarithm function, which is convex. From the Jensen inequality, we have

$$\begin{aligned} -\log\left\{\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq}\right\} &= -\log\left\{\sum_{m=1}^{k}\sum_{q=1}^{\ell}\lambda_{ijmq}\frac{f_{im}s_{mq}a_{jq}}{\lambda_{ijmq}}\right\} \\ &\leq -\sum_{m=1}^{k}\sum_{q=1}^{\ell}\lambda_{ijmq}\log\left\{\frac{f_{im}s_{mq}a_{jq}}{\lambda_{ijmq}}\right\} \end{aligned} \tag{5.20}$$

with equality if and only if (5.8). If we replace $-\log\{\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq}\}$ in (5.19) with $-\sum_{m=1}^{k}\sum_{q=1}^{\ell}\lambda_{ijmq}\log\{f_{im}s_{mq}a_{jq}/\lambda_{ijmq}\}$, that is, the final term in (5.20), and substitute (5.9) into $-\sum_{m=1}^{k}\sum_{q=1}^{\ell}\lambda_{ijmq}\log\{f_{im}s_{mq}a_{jq}/\lambda_{ijmq}\}$, we obtain the following auxiliary function of $Q_{\boldsymbol{S}}(\boldsymbol{S})$:

$$\begin{aligned} &Q^{\mathrm{aux}}_{\boldsymbol{S}}(\boldsymbol{S}, \boldsymbol{S}^*) \\ =&\sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq} \\ &-\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}\sum_{m=1}^{k}\sum_{q=1}^{\ell}\left(\frac{f_{im}s^*_{mq}a_{jq}}{\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ru}s^*_{rc}a_{jc}}\right)\log\left\{\frac{s_{mq}(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s^*_{rc}a_{jc})}{s^*_{mq}}\right\}. \end{aligned} \tag{5.21}$$

Then, we derive an optimal $\hat{s}_{mq}$ that minimize $Q_{\boldsymbol{S}}^{\mathrm{aux}}(\boldsymbol{S}, \boldsymbol{S}^*)$ with respect to s_{mq}. The partial derivative of $Q_{\boldsymbol{S}}^{\mathrm{aux}}(\boldsymbol{S}, \boldsymbol{S}^*)$ with respect to s_{mq} is as follows:

$$\frac{\partial Q_{\boldsymbol{S}}^{\mathrm{aux}}(\boldsymbol{S}, \boldsymbol{S}^*)}{\partial s_{mq}} = \sum_{i=1}^{n}\sum_{j=1}^{p} f_{im}a_{jq} - \frac{s^*_{mq}}{s_{mq}}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\frac{y_{ij}}{\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s^*_{rc}a_{jc}}\right)f_{im}a_{jq}. \tag{5.22}$$

The optimal $\hat{s}_{mq}$ is obtained by setting (5.22) to 0 and solving the equation with respect to s_{mq} as follows:

$$\hat{s}_{mq} = s^*_{mq}\frac{\sum_{i=1}^{n}\sum_{j=1}^{p}\{y_{ij}/(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s^*_{rc}a_{jc})\}f_{im}a_{jq}}{\sum_{i=1}^{n}\sum_{j=1}^{p} f_{im}a_{jq}}. \tag{5.23}$$

The matrix form of this update rule (5.23) is as follows:

$$\mathrm{vec}(\hat{\boldsymbol{S}}) = \mathrm{vec}(\boldsymbol{S}^*) \odot \frac{(\boldsymbol{A}\otimes\boldsymbol{F})'[\mathrm{vec}(\boldsymbol{Y})/\{(\boldsymbol{A}\otimes\boldsymbol{F})\mathrm{vec}(\boldsymbol{S})\}]}{(\boldsymbol{A}\otimes\boldsymbol{F})'\boldsymbol{1}_{np}}. \tag{5.24}$$

Algorithm

The P3NMF algorithm, which is derived from (5.16), (5.18), and (5.24), is presented in Algorithm 8.

Algorithm 8 P3NMF Algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_+^{n\times p}$, $k \in \mathbb{N}$, $\ell \in \mathbb{N}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_+^{n\times k}$, $\boldsymbol{S}^{(0)} \in \mathbb{R}_+^{k\times \ell}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p\times \ell}$, $\tau > 0$, and $\upsilon \in \mathbb{N}$
$t \leftarrow 0$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{P}}\left(y_{ij}\middle|x_{ij}^{(t)}\right)$
repeat
    $t \leftarrow t + 1$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{(t-1)} \odot \frac{\{\boldsymbol{Y}/(\boldsymbol{F}^{(t-1)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})\}\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}}{\boldsymbol{E}_{n\times p}\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}}$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t-1)} \odot \frac{\{\boldsymbol{Y}/(\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})\}'\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}}{\boldsymbol{E}_{p\times n}\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}}$
    $\mathrm{vec}(\boldsymbol{S}^{(t)}) \leftarrow \mathrm{vec}(\boldsymbol{S}^{(t-1)}) \odot \frac{(\boldsymbol{A}^{(t)}\otimes\boldsymbol{F}^{(t)})'[\mathrm{vec}(\boldsymbol{Y})/\{(\boldsymbol{A}^{(t)}\otimes\boldsymbol{F}^{(t)})\mathrm{vec}(\boldsymbol{S}^{(t-1)})\}]}{(\boldsymbol{A}^{(t)}\otimes\boldsymbol{F}^{(t)})'\boldsymbol{1}_{np}}$
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{P}}\left(y_{ij}\middle|x_{ij}^{(t)}\right)$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$, and $\boldsymbol{A}^{(t)}$
```

5.3 Compound Poisson-gamma distribution

In this section we present details of three-factor NMF based on a compound Poisson-gamma distribution, named CP3NMF. The objective function is defined as (3.39), where $\boldsymbol{\theta} = \{\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}\}$ and $\boldsymbol{X} \coloneqq \boldsymbol{F}\boldsymbol{S}\boldsymbol{A}'$.

Update rules

Below, we show the update rules of the parameters, $\boldsymbol{F}$, $\boldsymbol{S}$, and $\boldsymbol{A}$. ϕ is obtained as described in Section 3.3.

Update rule for $\boldsymbol{F}$

For N3NMF and P3NMF, the update rule of $\boldsymbol{F}$ is given in the same form as two-factor NMF, that is, (4.30) and (4.31), as follows:

$$\hat{f}_{im} = f^*_{im}\left\{\frac{\sum_{j=1}^{p} y_{ij}(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f^*_{ir}s_{rc}a_{jc})^{\beta-2}\sum_{q=1}^{\ell} s_{mq}a_{jq}}{\sum_{j=1}^{p}(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f^*_{ir}s_{rc}a_{jc})^{\beta-1}\sum_{q=1}^{\ell} s_{mq}a_{jq}}\right\}^{\frac{1}{2-\beta}}. \tag{5.25}$$

$$\hat{\boldsymbol{F}} = \boldsymbol{F}^* \odot \left[\frac{\{\boldsymbol{Y} \odot (\boldsymbol{F}^*\boldsymbol{S}\boldsymbol{A}')^{\beta-2}\}\boldsymbol{A}\boldsymbol{S}'}{\{(\boldsymbol{F}^*\boldsymbol{S}\boldsymbol{A}')^{\beta-1}\}\boldsymbol{A}\boldsymbol{S}'}\right]^{\frac{1}{2-\beta}}. \tag{5.26}$$

Update rule for $\boldsymbol{A}$

We can obtain an update rule of $\boldsymbol{A}$ in the same form as $\boldsymbol{F}$:

$$\hat{a}_{jm} = a^*_{jm}\left\{\frac{\sum_{i=1}^{n} y_{ij}(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ru}s_{rc}a^*_{jc})^{\beta-2}\sum_{m=1}^{k} f_{im}s_{mq}}{\sum_{i=1}^{n}(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s_{rc}a^*_{jc})^{\beta-1}\sum_{m=1}^{k} f_{im}s_{mq}}\right\}^{\frac{1}{2-\beta}}, \tag{5.27}$$

$$\hat{\boldsymbol{A}} = \boldsymbol{A}^* \odot \left[\frac{\{\boldsymbol{Y} \odot (\boldsymbol{F}\boldsymbol{S}\boldsymbol{A}^{*\prime})^{\beta-2}\}'\boldsymbol{F}\boldsymbol{S}}{\{(\boldsymbol{F}\boldsymbol{S}\boldsymbol{A}^{*\prime})^{\beta-1}\}'\boldsymbol{F}\boldsymbol{S}}\right]^{\frac{1}{2-\beta}}. \tag{5.28}$$

Update rule for $\boldsymbol{S}$

From (3.39), the objective function to be minimized with respect to $\boldsymbol{S}$ is as follows:

$$Q_{\boldsymbol{S}}(\boldsymbol{S}) = \sum_{i=1}^{n}\sum_{j=1}^{p}\left(\frac{(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq})^{\beta}}{\beta} - \frac{y_{ij}(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq})^{\beta-1}}{\beta-1}\right). \tag{5.29}$$

For CP2NMF, we can apply the Jensen inequality and inequality (4.23) to the first term and second term, respectively, to derive the auxiliary function of (5.29) with respect to $\boldsymbol{S}$. From (4.23), we have

$$\begin{aligned}\frac{(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq})^{\beta}}{\beta} &\le \frac{\eta_{ij}^{\beta}}{\beta} + \eta_{ij}^{\beta-1}\left(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq} - \eta_{ij}\right)\\ &= \eta_{ij}^{\beta-1}\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq} + \eta_{ij}^{\beta}\left(\frac{1}{\beta} - 1\right)\\ &\quad (i = 1, \ldots, n;\ j = 1, \ldots, p).\end{aligned} \tag{5.30}$$

The equality is satisfied if and only if

$$\eta_{ij} = \sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq} \; (i = 1,\ldots,n;\; j = 1,\ldots,p). \tag{5.31}$$

On the other hand, from the Jensen inequality, we have

$$\begin{aligned}
-\frac{(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq})^{\beta-1}}{\beta-1} &= -\frac{1}{\beta-1}\left(\sum_{m=1}^{k}\sum_{q=1}^{\ell}\lambda_{ijmq}\frac{f_{im}s_{mq}a_{jq}}{\lambda_{ijmq}}\right)^{\beta-1} \\
&\leq -\frac{1}{\beta-1}\sum_{m=1}^{k}\sum_{q=1}^{\ell}\lambda_{ijmq}\left(\frac{f_{im}s_{mq}a_{jq}}{\lambda_{ijmq}}\right)^{\beta-1} \\
&= -\frac{1}{\beta-1}\sum_{m=1}^{k}\sum_{q=1}^{\ell}\left(\frac{f_{im}^{\beta-1}s_{mq}^{\beta-1}a_{jm}^{\beta-1}}{\lambda_{ijm}^{\beta-2}}\right) \\
&(i = 1,\ldots,n;\; j = 1,\ldots,p).
\end{aligned} \tag{5.32}$$

The equality is satisfied if and only if (5.8). If we replace

$$\frac{(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq})^{\beta}}{\beta} \quad \text{and} \quad -\frac{(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq})^{\beta-1}}{\beta-1}$$

in (5.29) with

$$\eta_{ij}^{\beta-1}\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq} + \eta_{ij}^{\beta}\left(\frac{1}{\beta-1}\right) \quad \text{and} \quad -\frac{1}{\beta-1}\sum_{m=1}^{k}\sum_{q=1}^{\ell}\left(\frac{f_{im}^{\beta-1}s_{mq}^{\beta-1}a_{jm}^{\beta-1}}{\lambda_{ijm}^{\beta-2}}\right),$$

that is, the final term in (5.30) and (5.32), respectively, and substitute

$$\eta_{ij} = \sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}^{*}a_{jq} \; (i = 1,\ldots,n;\; j = 1,\ldots,p) \tag{5.33}$$

and (5.9) into the replaced (5.29), we obtain the following auxiliary function of $Q_{\boldsymbol{S}}(\boldsymbol{S})$:

$$\begin{aligned}
Q_{\boldsymbol{S}}^{\text{aux}}(\boldsymbol{S},\boldsymbol{S}^{*}) = \sum_{i=1}^{n}\sum_{j=1}^{p}\Bigg\{&\left(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s_{rc}^{*}a_{jc}\right)^{\beta-1}\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq} \\
&+\left(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s_{rc}^{*}a_{jc}\right)^{\beta}\left(\frac{1}{\beta}-1\right) \\
&-\frac{1}{\beta-1}y_{ij}\left(\sum_{r=1}^{k}\sum_{q=1}^{\ell} f_{ir}s_{rc}^{*}a_{jc}\right)^{\beta-2}\sum_{m=1}^{k}\sum_{q=1}^{\ell}\left(\frac{f_{im}s_{mq}^{\beta-1}a_{jq}}{s_{mq}^{*\beta-2}}\right)\Bigg\}.
\end{aligned} \tag{5.34}$$

Then, we derive an optimal $\hat{s}_{mq}$ that minimizes $Q_{\boldsymbol{S}}^{\text{aux}}(\boldsymbol{S},\boldsymbol{S}^{*})$ with respect to s_{mq}. The partial derivative of $Q_{\boldsymbol{S}}^{\text{aux}}(\boldsymbol{S},\boldsymbol{S}^{*})$ with respect to f_{im} is as follows:

$$\begin{aligned}
\frac{\partial Q_{\boldsymbol{S}}^{\text{aux}}(\boldsymbol{S},\boldsymbol{S}^{*})}{\partial f_{im}} =&\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s_{rc}^{*}a_{jc}\right)^{\beta-1} f_{im}a_{jq} \\
&- s_{mq}^{\beta-2}\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}\left(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s_{rc}^{*}a_{jc}\right)^{\beta-2}\frac{f_{im}a_{jm}}{s_{mq}^{*\beta-2}}.
\end{aligned} \tag{5.35}$$

The optimal $\hat{s}_{mq}$ is obtained by setting (5.35) to 0 and solving the equation with respect to s_{mq} as follows:

$$\hat{s}_{mq} = s^*_{mq} \left\{ \frac{\sum_{i=1}^{n} \sum_{j=1}^{p} y_{ij} (\sum_{r=1}^{k} \sum_{c=1}^{\ell} f_{ir} s^*_{rc} a_{jc})^{\beta-2} f_{im} a_{jq}}{\sum_{i=1}^{n} \sum_{j=1}^{p} (\sum_{r=1}^{k} \sum_{c=1}^{\ell} f_{ir} s^*_{rc} a_{jc})^{\beta-1} f_{im} a_{jq}} \right\}^{\frac{1}{2-\beta}}. \tag{5.36}$$

The matrix form of this update rule (5.36) is as follows:

$$\mathrm{vec}(\hat{\boldsymbol{S}}) = \mathrm{vec}(\boldsymbol{S}^*) \odot \left[\frac{(\boldsymbol{A} \otimes \boldsymbol{F})'[\mathrm{vec}(\boldsymbol{Y}) \odot \{(\boldsymbol{A} \otimes \boldsymbol{F})\mathrm{vec}(\boldsymbol{S}^*)\}^{\beta-2}]}{(\boldsymbol{A} \otimes \boldsymbol{F})'\{(\boldsymbol{A} \otimes \boldsymbol{F})\mathrm{vec}(\boldsymbol{S}^*)\}^{\beta-1}} \right]^{\frac{1}{2-\beta}}. \tag{5.37}$$

Together, the factorizations (5.26), (5.28), and (5.37), indicate that CP3NMF is a generalization of N3NMF or P3NMF, as is the case with CP2NMF.

Algorithm

The CP3NMF algorithm, which is based on (5.26), (5.28), (5.37), and the discussion in Section 3.3 about optimal ϕ, is presented in Algorithm 9.

Algorithm 9 CP3NMF Algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_+^{n \times p}$, $\beta \in (0,1)$, $k \in \mathbb{N}$, $\ell \in \mathbb{N}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_+^{n \times k}$, $\boldsymbol{S}^{(0)} \in \mathbb{R}_+^{k \times \ell}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p \times \ell}$, $\phi^{(0)} > 0$, $\tau > 0$, $\upsilon \in \mathbb{N}$, $\delta \in \mathbb{N}$, and $\kappa \in \mathbb{N}$
$t \leftarrow 0$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{CP}}\left(y_{ij} \middle| x_{ij}^{(t)}, \phi^{(t)}, \beta\right)$
repeat
    $t \leftarrow t + 1$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{(t-1)} \odot \left[ \frac{\{\boldsymbol{Y} \odot (\boldsymbol{F}^{(t-1)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-2}\}\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}}{\{(\boldsymbol{F}^{(t-1)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-1}\}\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}} \right]^{\frac{1}{2-\beta}}$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t-1)} \odot \left[ \frac{\{\boldsymbol{Y} \odot (\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-2}\}'\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}}{\{(\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-1}\}'\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}} \right]^{\frac{1}{2-\beta}}$
    $\mathrm{vec}(\hat{\boldsymbol{S}}) \leftarrow \mathrm{vec}(\boldsymbol{S}^*) \odot \left[ \frac{(\boldsymbol{A}^{(t)} \otimes \boldsymbol{F}^{(t)})'[\mathrm{vec}(\boldsymbol{Y}) \odot \{(\boldsymbol{A}^{(t)} \otimes \boldsymbol{F}^{(t)})\mathrm{vec}(\boldsymbol{S}^{(t-1)})\}^{\beta-2}]}{(\boldsymbol{A}^{(t)} \otimes \boldsymbol{F}^{(t)})'\{(\boldsymbol{A}^{(t)} \otimes \boldsymbol{F}^{(t)})\mathrm{vec}(\boldsymbol{S}^{(t-1)})\}^{\beta-1}} \right]^{\frac{1}{2-\beta}}$
    if $t \leq \delta$ or $t \bmod \kappa = 0$ then
        $\phi^{(t)}$ is obtained as the optimal $\phi$ that optimizes $Q_\phi(\phi)$ given $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$, $\boldsymbol{A}^{(t)}$, and $\beta$ using the BFGS quasi-Newton method with constraints $\phi > 0$
    end if
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{CP}}\left(y_{ij} \middle| x_{ij}^{(t)}, \phi^{(t)}, \beta\right)$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$, $\boldsymbol{A}^{(t)}$, and $\phi^{(t)}$
```

5.4 Zero-inflated compound Poisson-gamma distribution

In this section we present details of three-factor NMF based on a zero-inflated compound Poisson-gamma distribution, named ZICP3NMF. From the perspective of an approximation matrix, ZICP3NMF is the three-factor version of ZICP2NMF. On the other hand, from the perspective of an error distribution, it is a zero-inflated version of CP3NMF. The objective function is defined as (3.44), where $\boldsymbol{\theta} = \{\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}\}$ and $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{S}\boldsymbol{A}'$. However, the update rule of $\boldsymbol{F}$, $\boldsymbol{S}$, and $\boldsymbol{A}$ is obtained as the optimizer, which minimizes (3.51).

Update rules

We show the update rules of the parameters, $\boldsymbol{F}$, $\boldsymbol{S}$, and $\boldsymbol{A}$. The update rule of $\hat{\boldsymbol{Z}}$ and w is obtained as (3.52) and (3.55). ϕ is obtained as described in Section 3.3.

Update rule for $\boldsymbol{F}$

From (3.44), the objective function to be minimized with respect to $\boldsymbol{F}$ is as follows:

$$\begin{aligned} &Q_{\boldsymbol{F}}(\boldsymbol{F}) \\ &= \frac{1}{\phi}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\frac{\hat{z}^*_{ij}(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq})^{\beta}}{\beta} - \frac{\hat{z}^*_{ij}y_{ij}(\sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq})^{\beta-1}}{\beta-1}\right). \end{aligned} \tag{5.38}$$

This objective function is similar to (5.29) but is weighted by z^*_{ij}. Therefore, the update rule of $\boldsymbol{F}$ is obtained in a form similar to (5.25) and (5.26) but each of the i, j elements is weighted by z^*_{ij}.

$$\hat{f}_{im} = f^*_{im}\left\{\frac{\sum_{j=1}^{p} z^*_{ij}y_{ij}(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f^*_{ir}s_{rc}a_{jc})^{\beta-2}\sum_{q=1}^{\ell} s_{mq}a_{jq}}{\sum_{j=1}^{p} z^*_{ij}(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f^*_{ir}s_{rc}a_{jc})^{\beta-1}\sum_{q=1}^{\ell} s_{mq}a_{jq}}\right\}^{\frac{1}{2-\beta}}, \tag{5.39}$$

$$\hat{\boldsymbol{F}} = \boldsymbol{F}^* \odot \left[\frac{\{\boldsymbol{Z}^* \odot \boldsymbol{Y} \odot (\boldsymbol{F}^*\boldsymbol{S}\boldsymbol{A}')^{\beta-2}\}\boldsymbol{A}\boldsymbol{S}'}{\{\boldsymbol{Z}^* \odot (\boldsymbol{F}^*\boldsymbol{S}\boldsymbol{A}')^{\beta-1}\}\boldsymbol{A}\boldsymbol{S}'}\right]^{\frac{1}{2-\beta}}. \tag{5.40}$$

Update rule for $\boldsymbol{A}$

We can obtain an update rule of $\boldsymbol{A}$ in the same form as $\boldsymbol{F}$:

$$\hat{a}_{jm} = a^*_{jm}\left\{\frac{\sum_{i=1}^{n} z^*_{ij}y_{ij}(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ru}s_{rc}a^*_{jc})^{\beta-2}\sum_{m=1}^{k} f_{im}s_{mq}}{\sum_{i=1}^{n} z^*_{ij}(\sum_{r=1}^{k}\sum_{c=1}^{\ell} f_{ir}s_{rc}a^*_{jc})^{\beta-1}\sum_{m=1}^{k} f_{im}s_{mq}}\right\}^{\frac{1}{2-\beta}}, \tag{5.41}$$

$$\hat{\boldsymbol{A}} = \boldsymbol{A}^* \odot \left[\frac{\{\boldsymbol{Z}^* \odot \boldsymbol{Y} \odot (\boldsymbol{F}\boldsymbol{S}\boldsymbol{A}^{*\prime})^{\beta-2}\}'\boldsymbol{F}\boldsymbol{S}}{\{\boldsymbol{Z}^* \odot (\boldsymbol{F}\boldsymbol{S}\boldsymbol{A}^{*\prime})^{\beta-1}\}'\boldsymbol{F}\boldsymbol{S}}\right]^{\frac{1}{2-\beta}}. \tag{5.42}$$

Update rule for $\boldsymbol{S}$

In the same way as $\boldsymbol{F}$ and $\boldsymbol{A}$, we can obtain the update rule for $\boldsymbol{S}$ similar to that of (5.36) and (5.37).

$$\hat{s}_{mq} = s^*_{mq} \left\{ \frac{\sum_{i=1}^{n} \sum_{j=1}^{p} z^*_{ij} y_{ij} (\sum_{r=1}^{k} \sum_{c=1}^{\ell} f_{ir} s^*_{rc} a_{jc})^{\beta-2} f_{im} a_{jq}}{\sum_{i=1}^{n} \sum_{j=1}^{p} z^*_{ij} (\sum_{r=1}^{k} \sum_{c=1}^{\ell} f_{ir} s^*_{rc} a_{jc})^{\beta-1} f_{im} a_{jq}} \right\}^{\frac{1}{2-\beta}}, \tag{5.43}$$

$$\begin{aligned} \mathrm{vec}(\hat{\boldsymbol{S}}) &= \mathrm{vec}(\boldsymbol{S}^*) \odot \left[\frac{(\boldsymbol{A} \otimes \boldsymbol{F})'[\mathrm{vec}(\boldsymbol{Z}^*) \odot \mathrm{vec}(\boldsymbol{Y}) \odot \{(\boldsymbol{A} \otimes \boldsymbol{F})\mathrm{vec}(\boldsymbol{S}^*)\}^{\beta-2}]}{(\boldsymbol{A} \otimes \boldsymbol{F})'\{(\boldsymbol{A} \otimes \boldsymbol{F})\mathrm{vec}(\boldsymbol{S}^*)\}^{\beta-1}} \right]^{\frac{1}{2-\beta}} \\ &= \mathrm{vec}(\boldsymbol{S}^*) \odot \left[\frac{(\boldsymbol{A} \otimes \boldsymbol{F})'[\mathrm{vec}(\boldsymbol{Z}^* \odot \boldsymbol{Y} \odot \mathrm{vec}(\boldsymbol{F}\boldsymbol{S}^*\boldsymbol{A}')^{\beta-2}]}{(\boldsymbol{A} \otimes \boldsymbol{F})'\mathrm{vec}(\boldsymbol{F}\boldsymbol{S}^*\boldsymbol{A}')^{\beta-1}} \right]^{\frac{1}{2-\beta}}. \end{aligned} \tag{5.44}$$

Algorithm

The ZICP3NMF algorithm, which is based on (3.52), (3.55), (5.40), (5.42), (5.44), and the discussion in 3.3 about optimal ϕ, is presented in Algorithm 10.

Algorithm 10 ZICP3NMF Algorithm

```
**Input** $\boldsymbol{Y} \in \mathbb{R}_{+}^{n\times p}$, $\beta \in (0,1)$, $k \in \mathbb{N}$, $\ell \in \mathbb{N}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_{+}^{n\times k}$, $\boldsymbol{S}^{(0)} \in \mathbb{R}_{+}^{k\times \ell}$ $\boldsymbol{A}^{(0)} \in \mathbb{R}_{+}^{p\times \ell}$, $w^{(0)} \in (0,1)$, $\phi^{(0)} > 0$, $\tau > 0$, $\upsilon \in \mathbb{N}$, $\delta \in \mathbb{N}$, and $\kappa \in \mathbb{N}$
$t \leftarrow 0$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
$z_{ij}^{(t)} \leftarrow \begin{cases} \dfrac{w^{(t)}}{w^{(t)} + (1-w^{(t)})h(0,\phi^{(t)},\beta)\exp\{-(x_{ij}^{(t)})^{\beta}/(\phi^{(t)}\beta)\}} & \text{if } y_{ij} = 0 \\ 0 & \text{if } y_{ij} \neq 0 \end{cases}$
$(i = 1,\ldots,n;\ j = 1,\ldots,p)$
$\boldsymbol{Z}^{*(t)} \leftarrow \boldsymbol{E}_{n\times p} - \boldsymbol{Z}^{(t)}$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log\left\{w^{(t)}I(y_{ij}=0) + (1-w^{(t)})f_{\mathrm{CP}}(y_{ij}|x_{ij}^{(t)},\phi^{(t)},\beta)\right\}$
**repeat**
    $t \leftarrow t+1$
    $w^{(t)} \leftarrow \dfrac{\sum_{i=1}^{n}\sum_{j=1}^{p} z_{ij}^{(t-1)}}{np}$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{(t-1)} \odot \left[\dfrac{\{\boldsymbol{Z}^{*(t-1)} \odot \boldsymbol{Y} \odot (\boldsymbol{F}^{(t-1)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-2}\}\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}}{\{\boldsymbol{Z}^{*(t-1)} \odot (\boldsymbol{F}^{(t-1)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-1}\}\boldsymbol{A}^{(t-1)}\boldsymbol{S}^{(t-1)\prime}}\right]^{\frac{1}{2-\beta}}$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t-1)} \odot \left[\dfrac{\{\boldsymbol{Z}^{*(t-1)} \odot \boldsymbol{Y} \odot (\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-2}\}^{\prime}\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}}{\{\boldsymbol{Z}^{*(t-1)} \odot (\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime})^{\beta-1}\}^{\prime}\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}}\right]^{\frac{1}{2-\beta}}$
    $\mathrm{vec}(\boldsymbol{S}^{(t)})$
    $\leftarrow \mathrm{vec}(\boldsymbol{S}^{(t-1)}) \odot \left[\dfrac{(\boldsymbol{A}^{(t)} \otimes \boldsymbol{F}^{(t)})^{\prime}\mathrm{vec}\{\boldsymbol{Z}^{*(t-1)} \odot \boldsymbol{Y} \odot (\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t)\prime})^{\beta-2}\}}{(\boldsymbol{A}^{(t)} \otimes \boldsymbol{F}^{(t)})^{\prime}\mathrm{vec}(\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t)})^{\beta-1}}\right]^{\frac{1}{2-\beta}}$
    **if** $t \leq \delta$ or $t \bmod \kappa = 0$ **then**
        $\phi^{(t)}$ is obtained as the optimal $\phi$ that optimizes $Q_{\phi}(\phi)$ given $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$ $\boldsymbol{A}^{(t)}$, $\boldsymbol{Z}^{*(t-1)}$, and $\beta$ using the BFGS quasi-Newton method with constraints $\phi > 0$
    **end if**
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $z_{ij}^{(t)} \leftarrow \begin{cases} \dfrac{w^{(t)}}{w^{(t)} + (1-w^{(t)})h(0,\phi^{(t)},\beta)\exp\{-(x_{ij}^{(t)})^{\beta}/(\phi^{(t)}\beta)\}} & \text{if } y_{ij} = 0 \\ 0 & \text{if } y_{ij} \neq 0 \end{cases}$
    $(i = 1,\ldots,n;\ j = 1,\ldots,p)$
    $\boldsymbol{Z}^{*(t)} \leftarrow \boldsymbol{E}_{n\times p} - \boldsymbol{Z}^{(t)}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log\left\{w^{(t)}I(y_{ij}=0) + (1-w^{(t)})f_{\mathrm{CP}}(y_{ij}|x_{ij}^{(t)},\phi^{(t)},\beta)\right\}$
**until** $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
**Output** $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$, $\boldsymbol{A}^{(\upsilon)}$, $\boldsymbol{Z}^{(t)}$, $w^{(t)}$, and $\phi^{(\nu)}$
```

Chapter 6

Two-factor orthogonal NMF

In this chapter, we present four two-factor orthogonal NMFs. The objective of these methods is to obtain estimators of $\boldsymbol{F} \in \mathbb{R}_{+}^{n \times k}$ and $\boldsymbol{A} \in \mathbb{R}_{+}^{p \times k}$ such that $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{A}'$ is approximated to a given data matrix $\boldsymbol{Y}$ with column orthogonality constraints on $\boldsymbol{F}$. Clusters R_m $(m = 1, \ldots, k)$ and $\mathcal{R}$, which are used in this chapter, represent the m-th row cluster and a set of row clusters, respectively, as defined in Section 3.2.

6.1 Normal distribution

In this section we present details of two-factor orthogonal NMF based on a normal distribution, named N2ONMF. This method was proposed by Pompili et al. (2014) and they named this method “Weighted Spherical k-means.”

Objective function

From (3.30), the objective function to be minimized with respect to $\boldsymbol{F}$, $\boldsymbol{A}$, and σ^2 is

$$Q(\boldsymbol{F}, \boldsymbol{A}, \sigma^2) = \frac{np}{2}\log\{\sigma^2\} + \frac{1}{2\sigma^2}\|\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{A}'\|^2 + \text{const}. \tag{6.1}$$

Hence, the optimization problem is as follows:

$$\begin{aligned}&\underset{\boldsymbol{F},\boldsymbol{A},\sigma^2}{\operatorname{argmin}}\{Q(\boldsymbol{F}, \boldsymbol{A}, \sigma^2)\}\\ &\text{subject to } \boldsymbol{F} \in \mathbb{R}_{+}^{n \times k}, \boldsymbol{A} \in \mathbb{R}_{+}^{p \times k}, \text{ and } \boldsymbol{f}'_{(m)}\boldsymbol{f}_{(u)} = 0 \ (m \neq u).\end{aligned} \tag{6.2}$$

The objective function (6.1) is invariant to changes in the length of each column vector of $\boldsymbol{A}$ because the following is satisfied:

$$\begin{aligned}Q(\boldsymbol{F}, \boldsymbol{A}, \sigma^2) &= \frac{np}{2}\log\{\sigma^2\} + \frac{1}{2\sigma^2}\|\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{D}_{\boldsymbol{A}}\boldsymbol{D}_{\boldsymbol{A}}^{-1}\boldsymbol{A}'\|^2 + \text{const}\\ &= \frac{np}{2}\log\{\sigma^2\} + \frac{1}{2\sigma^2}\|\boldsymbol{Y} - \boldsymbol{F}^{\star}\boldsymbol{A}^{\star\prime}\|^2 + \text{const},\end{aligned} \tag{6.3}$$

where $\boldsymbol{F}^{\star} = \boldsymbol{F}\boldsymbol{D}_{\boldsymbol{A}}$ and $\boldsymbol{A}^{\star} = \boldsymbol{A}\boldsymbol{D}_{\boldsymbol{A}}^{-1}$. Then, we have

$$\boldsymbol{f}^{\star\prime}_{(m)}\boldsymbol{f}^{\star}_{(u)} = 0 \ (m \neq u) \text{ and } \operatorname{diag}(\boldsymbol{A}^{\star\prime}\boldsymbol{A}^{\star}) = \boldsymbol{I}_k. \tag{6.4}$$

According to (6.3) and (6.4), the optimization problem of N2ONMF can be rewritten as

$$\begin{aligned}&\underset{\boldsymbol{F},\boldsymbol{A},\sigma^2}{\operatorname{argmin}}\, Q(\boldsymbol{F},\boldsymbol{A},\sigma^2)\\&\text{subject to } \boldsymbol{F}\in\mathbb{R}_+^{n\times k}, \boldsymbol{A}\in\mathbb{R}_+^{p\times k}, \boldsymbol{f}'_{(m)}\boldsymbol{f}_{(u)}=0\ (m\neq u),\ \text{and } \operatorname{diag}(\boldsymbol{A}'\boldsymbol{A})=\boldsymbol{I}_k.\end{aligned}\tag{6.5}$$

It is noted that $\boldsymbol{F}$ satisfies (3.7) under the condition in (6.5).

Update rules

We show the update rules of the parameters, $\boldsymbol{F}$ and $\boldsymbol{A}$. Note that the update rule of σ^2 is (3.31).

Update rule for $\boldsymbol{F}$

The discussion in section 3.2 enables us to divide the optimization problem of $\boldsymbol{F}$ into that of $\mathcal{R}$ and f_{im} ($i\in R_m;\ m=1,\dots,k$). Hence, the objective function with respect to $\boldsymbol{F}$ and $\boldsymbol{A}$ can be written as follows:

$$\begin{aligned}Q_{\boldsymbol{F},\boldsymbol{A}}(\boldsymbol{F},\boldsymbol{A})&=\sum_{i=1}^{n}\left\|\boldsymbol{y}_i-\sum_{m=1}^{k}f_{im}\boldsymbol{a}_{(m)}\right\|^2\\&=\sum_{m=1}^{k}\sum_{i\in R_m}\|\boldsymbol{y}_i-f_{im}\boldsymbol{a}_{(m)}\|^2\ (\because (3.7))\\&=\sum_{m=1}^{k}\sum_{i\in R_m}\{\boldsymbol{y}'_i\boldsymbol{y}_i-2f_{im}\boldsymbol{y}'_i\boldsymbol{a}_{(m)}+f_{im}^2\boldsymbol{a}'_{(m)}\boldsymbol{a}_{(m)}\}\ (\because \operatorname{diag}(\boldsymbol{A}'\boldsymbol{A})=\boldsymbol{I}_k)\\&=\sum_{m=1}^{k}\sum_{i\in R_m}\{-2f_{im}\boldsymbol{y}'_i\boldsymbol{a}_{(m)}+f_{im}^2\}+\text{const.}\end{aligned}\tag{6.6}$$

Hence, the minimizer of f_{im} ($i=1,\dots,n;\ m=1,\dots,k$) for (6.6) given $\mathcal{R}$ and $\boldsymbol{A}$ is

$$f_{im}=\begin{cases}\boldsymbol{y}'_i\boldsymbol{a}_{(m)} & \text{if } i\in R_m\\ 0 & \text{if } i\notin R_m\end{cases}\ (i=1,\dots,n;\ m=1,\dots,k).\tag{6.7}$$

Substituting (6.7) into (6.6) and rearranging the terms proportional to the parameters, we obtain

$$Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A})=\sum_{m=1}^{k}\sum_{i\in R_m}\Big\{-(\boldsymbol{y}'_i\boldsymbol{a}_{(m)})^2\Big\}.\tag{6.8}$$

Therefore, the problem of minimizing $Q_{\boldsymbol{F},\boldsymbol{A}}(\boldsymbol{F},\boldsymbol{A})$ is the same as the problem of minimizing $Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A})$. Given $\boldsymbol{A}$, the minimizers of $\mathcal{R}$ for $Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A})$ are derived by, e.g., a k-means algorithm such that

$$R_m=\left\{i\ \middle|\ \underset{u}{\operatorname{argmax}}(\boldsymbol{y}'_i\boldsymbol{a}_{(u)})=m\right\}\ (m=1,\dots,k).\tag{6.9}$$

Update rule for $\boldsymbol{A}$

Because (6.8) can be rewritten such that

$$Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A}) = -\sum_{m=1}^{k} \|\boldsymbol{Y}_m \boldsymbol{a}_{(m)}\|^2 \tag{6.10}$$

where $\boldsymbol{Y}_m$ $(m = 1, \ldots, k)$ is a $|R_m| \times p$ submatrix of $\boldsymbol{Y}$ consisting of the row vectors of R_m, the minimizer of $\boldsymbol{a}_{(m)}$, given $\mathcal{R}$, can be obtained as the first nonnegative singular vector of $\boldsymbol{Y}_m'$ as follows:

$$\boldsymbol{a}_{(m)} = \mathit{\Delta}(\boldsymbol{Y}_m')\ (m = 1, \ldots, k). \tag{6.11}$$

Algorithm

The N2ONMF algorithm, which is derived from (6.7), (6.9), (6.11), and (3.31), is presented in Algorithm 11.

Algorithm 11 N2ONMF Algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_+^{n\times p}$, $k \in \mathbb{N}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p\times k}$ $(\mathrm{diag}(\boldsymbol{A}^{(0)\prime}\boldsymbol{A}^{(0)}) = \boldsymbol{I}_k)$, $\tau > 0$, and $\upsilon \in \mathbb{N}$
$t \leftarrow 0$
$R_m^{(t)} \leftarrow \left\{ i \,\middle|\, \underset{u}{\mathrm{argmax}}(\boldsymbol{y}_i'\boldsymbol{a}_{(u)}^{(t)}) = m \right\}$ $(m = 1, \ldots, k)$
Set $\boldsymbol{Y}_m^{(t)}$ as the submatrix of $\boldsymbol{Y}$ consisting of the row vectors of $R_m^{(t)}$ for $m = 1, \ldots, k$
$f_{im}^{(t)} \leftarrow \begin{cases} \boldsymbol{y}_i'\boldsymbol{a}_{(m)}^{(t)} & \text{if } i \in R_m \\ 0 & \text{if } i \notin R_m \end{cases}$ $(i = 1, \ldots, n;\ m = 1, \ldots, k)$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
$(\sigma^{(t)})^2 \leftarrow \frac{1}{np}\|\boldsymbol{Y} - \boldsymbol{X}^{(t)}\|^2$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{N}}\left(y_{ij} \,\middle|\, x_{ij}^{(t)}, (\sigma^{(t)})^2\right)$
repeat
    $t \leftarrow t + 1$
    $\boldsymbol{a}_{(m)} \leftarrow \mathit{\Delta}(\boldsymbol{Y}_m^{(t-1)\prime})$ $(m = 1, \ldots, k)$
    $R_m^{(t)} \leftarrow \left\{ i \,\middle|\, \underset{r}{\mathrm{argmax}}(\boldsymbol{y}_i'\boldsymbol{a}_{(r)}^{(t)}) = m \right\}$ $(m = 1, \ldots, k)$
    Set $\boldsymbol{Y}_m^{(t)}$ as the submatrix of $\boldsymbol{Y}$ consisting of the row vectors of $R_m^{(t)}$ for $m = 1, \ldots, k$
    $f_{im}^{(t)} \leftarrow \begin{cases} \boldsymbol{y}_i'\boldsymbol{a}_{(m)}^{(t)} & \text{if } i \in R_m \\ 0 & \text{if } i \notin R_m \end{cases}$ $(i = 1, \ldots, n;\ m = 1, \ldots, k)$
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $(\sigma^{(t)})^2 \leftarrow \frac{1}{np}\|\boldsymbol{Y} - \boldsymbol{X}^{(t)}\|^2$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{N}}\left(y_{ij} \,\middle|\, x_{ij}^{(t)}, (\sigma^{(t)})^2\right)$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(t)}$, $\mathcal{R}^{(t)}$ and $(\sigma^{(t)})^2$
```

6.2 Poisson distribution

In this section we present details of two-factor orthogonal NMF based on a Poisson distribution, named P2ONMF.

Objective function

From (3.35), the objective function to be minimized with respect to $\boldsymbol{F}$ and $\boldsymbol{A}$ is

$$Q(\boldsymbol{F},\boldsymbol{A}) = -\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij}\log\left\{\sum_{m=1}^{k} f_{im}a_{jm}\right\} + \sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{m=1}^{k} f_{im}a_{jm} + \text{const}. \tag{6.12}$$

Hence, the optimization problem is as follows:

$$\begin{aligned}&\underset{\boldsymbol{F},\boldsymbol{A}}{\operatorname{argmin}}\{Q(\boldsymbol{F},\boldsymbol{A})\}\\ &\text{subject to } \boldsymbol{F}\in\mathbb{R}_{+}^{n\times k}, \boldsymbol{A}\in\mathbb{R}_{+}^{p\times k}, \text{ and } \boldsymbol{f}'_{(m)}\boldsymbol{f}_{(u)} = 0\ (m\neq u).\end{aligned} \tag{6.13}$$

Update rules

We show the update rules of the parameters, $\boldsymbol{F}$ and $\boldsymbol{A}$.

Update rule for $\boldsymbol{F}$

For N2ONMF, the optimization problem of $\boldsymbol{F}$ is divided into that of $\mathcal{R}$ and f_{im} $(i \in R_m;\ m = 1,\dots,k)$. The objective function with respect to $\boldsymbol{F}$ and $\boldsymbol{A}$ can be written as follows:

$$\begin{aligned}Q_{\boldsymbol{F},\boldsymbol{A}}(\boldsymbol{F},\boldsymbol{A}) &= \sum_{i=1}^{n}\sum_{j=1}^{p}\left(-y_{ij}\log\left\{\sum_{m=1}^{k} f_{im}a_{jm}\right\} + \sum_{m=1}^{k} f_{im}a_{jm}\right)\\ &= \sum_{m=1}^{k}\sum_{i\in R_m}\sum_{j=1}^{p}[-y_{ij}\log\{f_{im}a_{jm}\} + f_{im}a_{jm}]\ (\because (3.7)).\end{aligned} \tag{6.14}$$

From (6.14), the minimizer of f_{im} $(i = 1,\dots,n;\ m = 1,\dots,k)$ for (6.14) given $\mathcal{R}$ and $\boldsymbol{A}$ is

$$f_{im} = \begin{cases}\dfrac{\sum_{j=1}^{p} y_{ij}}{\sum_{j=1}^{p} a_{jm}} & \text{if } i\in R_m\\ 0 & \text{if } i\notin R_m\end{cases} \quad (i = 1,\dots,n;\ m = 1,\dots,k). \tag{6.15}$$

Substituting (6.15) into (6.14) and rearranging the terms proportional to the parameters, we obtain

$$Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A}) = -\sum_{m=1}^{k}\sum_{i\in R_m}\sum_{j=1}^{p} y_{ij}\log\left\{\frac{a_{jm}}{\sum_{s=1}^{p} a_{sm}}\right\}. \tag{6.16}$$

Therefore, the problem of minimizing $Q_{\boldsymbol{F},\boldsymbol{A}}(\boldsymbol{F},\boldsymbol{A})$ is the same as the problem of minimizing $Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A})$. Given $\boldsymbol{A}$, the minimizers of $\mathcal{R}$ for $Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A})$ are derived such that

$$R_m = \left\{ i \,\middle|\, \underset{u}{\operatorname{argmax}}\left\{\sum_{j=1}^{p} y_{ij}\log\left\{\frac{a_{ju}}{\sum_{s=1}^{p} a_{su}}\right\}\right\} = m\right\}\ (m = 1,\dots,k). \tag{6.17}$$

Update rule for $\boldsymbol{A}$

Note that (6.16) is rewritten as

$$Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A}) = \sum_{m=1}^{k}\sum_{i\in R_m}\sum_{j=1}^{p} y_{ij}\log\left\{\sum_{s=1}^{p} a_{sm}\right\} - \sum_{m=1}^{k}\sum_{i\in R_m}\sum_{j=1}^{p} y_{ij}\log\{a_{jm}\}. \tag{6.18}$$

It is difficult to directly obtain the minimizer of a_{jm} with respect to (6.18) because the summation of a_{jm} occurs in the log function. However, we can obtain the optimal a_{jm} using the auxiliary function method. Because the log function $f(x) = \log(x)$ is concave, we obtain the following auxiliary function from the inequality (4.23):

$$\begin{aligned} Q_{\boldsymbol{A}}^{\mathrm{aux}}(\boldsymbol{A}) = &\sum_{m=1}^{k}\sum_{i\in R_m}\sum_{j=1}^{p} y_{ij}\left\{\log\lambda_m + \frac{1}{\lambda_m}\left(\sum_{s=1}^{p} a_{sm} - \lambda_m\right)\right\} \\ &- \sum_{m=1}^{k}\sum_{i\in R_m}\sum_{j=1}^{p} y_{ij}\log\{a_{jm}\}, \end{aligned} \tag{6.19}$$

where $\lambda_m = \sum_{s=1}^{p} a_{sm}^{*}$ $(m = 1,\ldots,k)$. Therefore, we obtain the following update equation of a_{jm} as a minimizer with respect to (6.19):

$$a_{jm} = \frac{\sum_{s=1}^{p} a_{sm}^{*}\sum_{i\in R_m} y_{ij}}{\sum_{i\in R_m}\sum_{s=1}^{p} y_{is}} \quad (j = 1,\ldots,p;\ m = 1,\ldots,k). \tag{6.20}$$

Algorithm

The P2ONMF algorithm, which is based on (6.15), (6.17), and (6.20), is presented in Algorithm 12.

6.3 Compound Poisson-gamma distribution

In this section we present details of two-factor orthogonal NMF based on a compound Poisson-gamma distribution, named CP2ONMF.

Objective function

From (3.39), the objective function to be minimized with respect to $\boldsymbol{F}$, $\boldsymbol{A}$, and ϕ is

$$\begin{aligned} Q(\boldsymbol{F},\boldsymbol{A},\phi) = &-\sum_{i=1}^{n}\sum_{j=1}^{p}\log\{h(y_{ij},\phi,\beta)\} \\ &- \frac{1}{\phi}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\frac{y_{ij}(\sum_{m=1}^{k} f_{im}a_{jm})^{\beta-1}}{\beta-1} - \frac{(\sum_{m=1}^{k} f_{im}a_{jm})^{\beta}}{\beta}\right). \end{aligned} \tag{6.21}$$

Hence, the optimization problem is as follows:

$$\begin{aligned} &\underset{\boldsymbol{F},\boldsymbol{A},\phi}{\operatorname{argmin}}\{Q(\boldsymbol{F},\boldsymbol{A},\phi)\} \\ &\text{subject to } \boldsymbol{F}\in\mathbb{R}_{+}^{n\times k}, \boldsymbol{A}\in\mathbb{R}_{+}^{p\times k},\ \text{and } \boldsymbol{f}_{(m)}'\boldsymbol{f}_{(u)} = 0\ (m\neq u). \end{aligned} \tag{6.22}$$

Algorithm 12 P2ONMF Algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_+^{n\times p}$, $k \in \mathbb{N}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p\times k}$, $\tau > 0$, and $\upsilon \in \mathbb{N}$
$t \leftarrow 0$
$R_m^{(t)} \leftarrow \left\{ i \,\middle|\, \operatorname*{argmax}_u \left\{ \sum_{j=1}^p y_{ij} \log \left\{ \frac{a_{ju}^{(t)}}{\sum_{s=1}^p a_{su}^{(t)}} \right\} \right\} = m \right\}$ $(m = 1, \dots, k)$
$f_{im}^{(t)} \leftarrow \begin{cases} \frac{\sum_{j=1}^p y_{ij}}{\sum_{j=1}^p a_{jm}^{(t)}} & \text{if } i \in R_m^{(t)} \\ 0 & \text{if } i \notin R_m^{(t)} \end{cases}$ $(i = 1, \dots, n;\ m = 1, \dots, k)$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)} \boldsymbol{A}^{(t)\prime}$
$L^{(t)} \leftarrow \sum_{i=1}^n \sum_{j=1}^p \log f_{\mathrm{P}}\left(y_{ij} \middle| x_{ij}^{(t)}\right)$
repeat
    $t \leftarrow t + 1$
    $a_{jm}^{(t)} \leftarrow \frac{\sum_{s=1}^p a_{sm}^{*(t-1)} \sum_{i \in R_m^{(t-1)}} y_{ij}}{\sum_{i \in R_m^{(t-1)}} \sum_{s=1}^p y_{is}}$ $(j = 1, \dots, p;\ m = 1, \dots, k)$
    $R_m^{(t)} \leftarrow \left\{ i \,\middle|\, \operatorname*{argmax}_u \left\{ \sum_{j=1}^p y_{ij} \log \left\{ \frac{a_{ju}^{(t)}}{\sum_{s=1}^p a_{su}^{(t)}} \right\} \right\} = m \right\}$ $(m = 1, \dots, k)$
    $f_{im}^{(t)} \leftarrow \begin{cases} \frac{\sum_{j=1}^p y_{ij}}{\sum_{j=1}^p a_{jm}^{(t)}} & \text{if } i \in R_m^{(t)} \\ 0 & \text{if } i \notin R_m^{(t)} \end{cases}$ $(i = 1, \dots, n;\ m = 1, \dots, k)$
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)} \boldsymbol{A}^{(t)\prime}$
    $L^{(t)} \leftarrow \sum_{i=1}^n \sum_{j=1}^p \log f_{\mathrm{P}}\left(y_{ij} \middle| x_{ij}^{(t)}\right)$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(t)}$, and $\mathcal{R}^{(t)}$
```

Update rules

We show the update rules of the parameters, $\boldsymbol{F}$ and $\boldsymbol{A}$. ϕ is obtained as described in Section 3.3.

Update rule for $\boldsymbol{F}$

For N2ONMF and P2ONMF, the optimization problem of $\boldsymbol{F}$ is divided into that of $\mathcal{R}$ and f_{im} $(i \in R_m;\ m = 1, \dots, k)$. The objective function with respect to $\boldsymbol{F}$ and $\boldsymbol{A}$ can be written as follows:

$$\begin{aligned} Q_{\boldsymbol{F},\boldsymbol{A}}(\boldsymbol{F}, \boldsymbol{A}) &= \frac{1}{\beta} \sum_{i=1}^n \sum_{j=1}^p \left(\sum_{m=1}^k f_{im} a_{jm} \right)^{\beta} - \frac{1}{\beta - 1} \sum_{i=1}^n \sum_{j=1}^p y_{ij} \left(\sum_{m=1}^k f_{im} a_{jm} \right)^{\beta - 1} \\ &= \frac{1}{\beta} \sum_{m=1}^k \sum_{i \in R_m} f_{im}^{\beta} \sum_{j=1}^p a_{jm}^{\beta} - \frac{1}{\beta - 1} \sum_{m=1}^k \sum_{i \in R_m} f_{im}^{\beta - 1} \sum_{j=1}^p y_{ij} a_{jm}^{\beta - 1} \ (\because (3.7)). \end{aligned} \tag{6.23}$$

The minimizer of f_{im} $(i = 1, \ldots, n;\ m = 1, \ldots, k)$ for (6.23) given $\mathcal{R}$ and $\boldsymbol{A}$ is

$$f_{im} = \begin{cases} \dfrac{\sum_{j=1}^{p} y_{ij} a_{jm}^{\beta-1}}{\sum_{j=1}^{p} a_{jm}^{\beta}} & \text{if } i \in R_m \\ 0 & \text{if } i \notin R_m \end{cases} \quad (i = 1, \ldots, n;\ m = 1, \ldots, k). \tag{6.24}$$

Substituting (6.24) into (6.23) and rearranging the terms proportional to the parameters, we obtain

$$Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R}, \boldsymbol{A}) = -\frac{1}{\beta(\beta-1)} \sum_{m=1}^{k} \sum_{i \in R_m} \frac{\left\{ \sum_{j=1}^{p} y_{ij} a_{jm}^{\beta-1} \right\}^{\beta}}{\left\{ \sum_{j=1}^{p} a_{jm}^{\beta} \right\}^{\beta-1}}. \tag{6.25}$$

Therefore, the problem of minimizing $Q_{\boldsymbol{F},\boldsymbol{A}}(\boldsymbol{F}, \boldsymbol{A})$ is the same as the problem of minimizing $Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R}, \boldsymbol{A})$. Given $\boldsymbol{A}$, the minimizers of $\mathcal{R}$ for $Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R}, \boldsymbol{A})$ are derived such that

$$R_m = \left\{ i \,\middle|\, \underset{u}{\operatorname{argmin}} \left\{ \frac{\left\{ \sum_{j=1}^{p} y_{ij} a_{ju}^{\beta-1} \right\}^{\beta}}{\left\{ \sum_{j=1}^{p} a_{ju}^{\beta} \right\}^{\beta-1}} \right\} = m \right\} \ (m = 1, \ldots, k). \tag{6.26}$$

Update rule for $\boldsymbol{A}$

It is difficult to directly obtain the minimizer of a_{jm} with respect to (6.25) because the summation of a_{jm} occurs in the two power functions. However, we can obtain the optimal a_{jm} using the auxiliary function method. We can find that the following bivariate function

$$f(x, y) = -\frac{1}{\beta(\beta-1)} (y^{\beta}/x^{\beta-1}) \tag{6.27}$$

is in the (6.25). In fact, (6.27) is concave if $0 < \beta < 1$ as can easily be proven. Therefore, we have

$$f(x, y) \leq f(\lambda, \eta) + f_x(\lambda, \eta)(x - \lambda) + f_y(\lambda, \eta)(y - \eta) \tag{6.28}$$

for any λ and η with equality if and only if $x = \lambda$ and $y = \eta$. From this inequality, we obtain the following auxiliary function of (6.25) for a_{jm}:

$$\begin{aligned} Q_{\boldsymbol{A}}^{\text{aux}}(\boldsymbol{A}, \boldsymbol{A}^*) = &\sum_{m=1}^{k} \sum_{i \in R_m} \left\{ -\frac{1}{\beta(\beta-1)} \frac{\eta_{im}^{\beta}}{\lambda_m^{\beta-1}} + \frac{1}{\beta} \left(\frac{\eta_{im}}{\lambda_m} \right)^{\beta} \left(\sum_{j=1}^{p} a_{jm}^{\beta} - \lambda_m \right) \right. \\ &\left. + \frac{1}{1-\beta} \left(\frac{\eta_{im}}{\lambda_m} \right)^{\beta-1} \left(\sum_{j=1}^{p} y_{ij} a_{jm}^{\beta-1} - \eta_{im} \right) \right\}, \end{aligned} \tag{6.29}$$

where

$$\lambda_m = \sum_{s=1}^{p} a_{sm}^{*\beta} \ (m = 1, \ldots, k) \tag{6.30}$$

$$\text{and } \eta_{im} = \sum_{s=1}^{p} y_{is} a_{sm}^{*\beta-1} \ (i = 1, \ldots n;\ m = 1, \ldots, k). \tag{6.31}$$

It is clear that (2.2) and (2.3) also hold for (6.29). Hence, the minimizer of $\boldsymbol{A}$ with respect to (6.29) is at least the optimal $\boldsymbol{A}$, which is monotonically non-increasing for (6.25). Finally, we obtain the minimizer of a_{jm} with respect to (6.29) as follows:

$$a_{jm} = \frac{(\sum_{s=1}^{p} a_{sm}^{*\beta}) \sum_{i \in R_m} (\sum_{s=1}^{p} y_{is} a_{sm}^{*\beta-1})^{\beta-1} y_{ij}}{\sum_{i \in R_m} (\sum_{s=1}^{p} y_{is} a_{sm}^{*\beta-1})^{\beta}} \; (j = 1, \ldots, p; \; m = 1, \ldots, k). \qquad (6.32)$$

Algorithm

The CP2ONMF algorithm, which is based on (6.24), (6.26), (6.32), and the discussion in Section 3.3 about optimal ϕ, is presented in Algorithm 13.

6.4 Zero-inflated compound Poisson-gamma distribution

In this section we present details of the two-factor orthogonal NMF based on a zero-inflated compound Poisson-gamma distribution, named ZICP2ONMF. This method is an extended version of CP2ONMF and from another perspective, it is a restricted version of ZICP2NMF.

Objective function

From (3.44), the objective function is

$$\begin{aligned} & Q(\boldsymbol{F}, \boldsymbol{A}, w, \phi) \\ &= -\sum_{i=1}^{n} \sum_{j=1}^{p} \log \Bigg[w I(y_{ij} = 0) \\ & + (1-w) h(y_{ij}, \phi, \beta) \exp \left\{ \frac{1}{\phi} \left(\frac{y_{ij} (\sum_{m=1}^{k} f_{im} a_{jm})^{\beta-1}}{\beta - 1} - \frac{(\sum_{m=1}^{k} f_{im} a_{jm})^{\beta}}{\beta} \right) \right\} \Bigg]. \qquad (6.33) \end{aligned}$$

Hence, the optimization problem is as follows:

$$\begin{aligned} & \underset{\boldsymbol{F}, \boldsymbol{A}, w, \phi}{\operatorname{argmin}} \{ Q(\boldsymbol{F}, \boldsymbol{A}, w, \phi) \} \\ & \text{subject to } \boldsymbol{F} \in \mathbb{R}_{+}^{n \times k}, \boldsymbol{A} \in \mathbb{R}_{+}^{p \times k}, \text{ and } \boldsymbol{f}'_{(m)} \boldsymbol{f}_{(u)} = 0 \; (m \neq u). \qquad (6.34) \end{aligned}$$

However, the update rule of $\boldsymbol{F}$, $\boldsymbol{A}$, w, and ϕ is obtained as these optimizers, which minimize (3.51).

Update rules

We show the update rules of the parameters $\boldsymbol{F}$ and $\boldsymbol{A}$. The update rule of $\hat{\boldsymbol{Z}}$ and w is obtained as (3.52) and (3.55). ϕ is obtained as described in Section 3.3.

Algorithm 13 CP2ONMF Algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_{+}^{n \times p}$, $\beta \in (0,1)$, $k \in \mathbb{N}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_{+}^{p \times k}$, $\phi^{(0)} > 0$, $\tau > 0$, $\upsilon \in \mathbb{N}$, $\delta \in \mathbb{N}$, and $\kappa \in \mathbb{N}$
$t \leftarrow 0$
$R_m^{(t)} \leftarrow \left\{ i \,\middle|\, \underset{u}{\operatorname{argmin}} \left\{ \frac{\{\sum_{j=1}^{p} y_{ij} (a_{ju}^{(t)})^{\beta-1}\}^{\beta}}{\{\sum_{j=1}^{p} (a_{ju}^{(t)})^{\beta}\}^{\beta-1}} \right\} = m \right\} \; (m = 1, \ldots, k)$
$f_{im}^{(t)} \leftarrow \begin{cases} \frac{\sum_{j=1}^{p} y_{ij} (a_{jm}^{(t)})^{\beta-1}}{\sum_{j=1}^{p} (a_{jm}^{(t)})^{\beta}} & \text{if } i \in R_m^{(t)} \\ 0 & \text{if } i \notin R_m^{(t)} \end{cases} \; (i = 1, \ldots, n;\ m = 1, \ldots, k)$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)} \boldsymbol{A}^{(t)\prime}$
$L^{(t)} \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{p} \log f_{\mathrm{CP}} \left( y_{ij} \,\middle|\, x_{ij}^{(t)}, \phi^{(t)}, \beta \right)$
repeat
    $t \leftarrow t + 1$
    $a_{jm}^{(t)} \leftarrow \frac{\{\sum_{s=1}^{p} (a_{sm}^{(t-1)})^{\beta}\} \sum_{i \in R_m^{(t-1)}} \{\sum_{s=1}^{p} y_{is} (a_{sm}^{(t-1)})^{\beta-1}\}^{\beta-1} y_{ij}}{\sum_{i \in R_m^{(t-1)}} \{\sum_{s=1}^{p} y_{is} (a_{sm}^{(t-1)})^{\beta-1}\}^{\beta}}$
    $(j = 1, \ldots, p;\ m = 1, \ldots, k)$
    $R_m^{(t)} \leftarrow \left\{ i \,\middle|\, \underset{u}{\operatorname{argmin}} \left\{ \frac{\{\sum_{j=1}^{p} y_{ij} (a_{ju}^{(t)})^{\beta-1}\}^{\beta}}{\{\sum_{j=1}^{p} (a_{ju}^{(t)})^{\beta}\}^{\beta-1}} \right\} = m \right\} \; (m = 1, \ldots, k)$
    $f_{im}^{(t)} \leftarrow \begin{cases} \frac{\sum_{j=1}^{p} y_{ij} (a_{jm}^{(t)})^{\beta-1}}{\sum_{j=1}^{p} (a_{jm}^{(t)})^{\beta}} & \text{if } i \in R_m^{(t)} \\ 0 & \text{if } i \notin R_m^{(t)} \end{cases} \; (i = 1, \ldots, n;\ m = 1, \ldots, k)$
    if $t \leq \delta$ or $t \bmod \kappa = 0$ then
        $\phi^{(t)}$ is obtained as the optimal $\phi$ that optimizes $Q_{\phi}(\phi)$ given $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(t)}$, and
        $\beta$ using the BFGS quasi-Newton method with constraints $\phi > 0$
    end if
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)} \boldsymbol{A}^{(t)\prime}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{p} \log f_{\mathrm{CP}} \left( y_{ij} \,\middle|\, x_{ij}^{(t)}, \phi^{(t)}, \beta \right)$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(t)}$, $\mathcal{R}^{(t)}$, and $\phi^{(t)}$
```

Update rule for $\boldsymbol{F}$

For N2ONMF, P2ONMF, and CP2ONMF, the optimization problem of $\boldsymbol{F}$ is divided into that of $\mathcal{R}$ and f_{im} $(i \in R_m;\ m = 1, \ldots, k)$. From (3.51), the objective function with respect to $\boldsymbol{F}$ and $\boldsymbol{A}$ can be written as follows:

$$\begin{aligned} Q_{\boldsymbol{F},\boldsymbol{A}}(\boldsymbol{F},\boldsymbol{A}) &= \frac{1}{\beta}\sum_{i=1}^{n}\sum_{j=1}^{p}\hat{z}_{ij}^{*}\left(\sum_{m=1}^{k} f_{im}a_{jm}\right)^{\beta} - \frac{1}{\beta-1}\sum_{i=1}^{n}\sum_{j=1}^{p}\hat{z}_{ij}^{*}y_{ij}\left(\sum_{m=1}^{k} f_{im}a_{jm}\right)^{\beta-1} \\ &= \frac{1}{\beta}\sum_{m=1}^{k}\sum_{i\in R_m} f_{im}^{\beta}\sum_{j=1}^{p}\hat{z}_{ij}^{*}a_{jm}^{\beta} - \frac{1}{\beta-1}\sum_{m=1}^{k}\sum_{i\in R_m} f_{im}^{\beta-1}\sum_{j=1}^{p}\hat{z}_{ij}^{*}y_{ij}a_{jm}^{\beta-1} \ (\because (3.7)). \end{aligned} \tag{6.35}$$

The minimizer of f_{im} $(i = 1, \dots, n;\ m = 1, \dots, k)$ for (6.23) given $\mathcal{R}$ and $\boldsymbol{A}$ is

$$f_{im} = \begin{cases} \dfrac{\sum_{j=1}^{p}\hat{z}_{ij}^{*}y_{ij}a_{jm}^{\beta-1}}{\sum_{j=1}^{p}\hat{z}_{ij}^{*}a_{jm}^{\beta}} & \text{if } i \in R_m \\ 0 & \text{if } i \notin R_m \end{cases} \quad (i = 1, \dots, n;\ m = 1, \dots, k). \tag{6.36}$$

Substituting (6.36) into (6.35) and rearranging the terms proportional to the parameters, we obtain

$$Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A}) = -\frac{1}{\beta(\beta-1)}\sum_{m=1}^{k}\sum_{i\in R_m}\frac{\left\{\sum_{j=1}^{p}\hat{z}_{ij}^{*}y_{ij}a_{jm}^{\beta-1}\right\}^{\beta}}{\left\{\sum_{j=1}^{p}\hat{z}_{ij}^{*}a_{jm}^{\beta}\right\}^{\beta-1}}. \tag{6.37}$$

Therefore, the problem of minimizing $Q_{\boldsymbol{F},\boldsymbol{A}}(\boldsymbol{F},\boldsymbol{A})$ is the same as the problem of minimizing $Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A})$. Given $\boldsymbol{A}$, the minimizers of $\mathcal{R}$ for $Q_{\mathcal{R},\boldsymbol{A}}(\mathcal{R},\boldsymbol{A})$ are derived such that

$$\hat{R}_m = \left\{ i \,\middle|\, \underset{u}{\operatorname{argmin}} \left\{ \frac{\left\{\sum_{j=1}^{p}\hat{z}_{ij}^{*}y_{ij}a_{ju}^{\beta-1}\right\}^{\beta}}{\left\{\sum_{j=1}^{p}\hat{z}_{ij}^{*}a_{ju}^{\beta}\right\}^{\beta-1}} \right\} = m \right\} \ (m = 1, \dots, k). \tag{6.38}$$

Update rule for $\boldsymbol{A}$

It is difficult to directly obtain the minimizer of a_{jm} with respect to (6.37) because the summation of a_{jm} is in the two power functions. However, we can derive the auxiliary function as in the case of CP2ONMF. The inequality (6.28) allows us to obtain the following auxiliary function of (6.37) for a_{jm}:

$$\begin{aligned} Q_{\boldsymbol{A}}^{\text{aux}}(\boldsymbol{A},\boldsymbol{A}^{*}) = \sum_{m=1}^{k}\sum_{i\in R_m}\Bigg\{ &-\frac{1}{\beta(\beta-1)}\frac{\eta_{im}^{\beta}}{\lambda_{im}^{\beta-1}} + \frac{1}{\beta}\left(\frac{\eta_{im}}{\lambda_{im}}\right)^{\beta}\left(\sum_{j=1}^{p}\hat{z}_{ij}^{*}a_{jm}^{\beta} - \lambda_{im}\right) \\ &+ \frac{1}{1-\beta}\left(\frac{\eta_{im}}{\lambda_{im}}\right)^{\beta-1}\left(\sum_{j=1}^{p}\hat{z}_{ij}^{*}y_{ij}a_{jm}^{\beta-1} - \eta_{im}\right)\Bigg\}, \end{aligned} \tag{6.39}$$

where

$$\lambda_{im} = \sum_{s=1}^{p}\hat{z}_{is}^{*}a_{sm}^{*\beta} \ (i = 1, \dots, n;\ m = 1, \dots, k) \tag{6.40}$$

$$\text{and } \eta_{im} = \sum_{s=1}^{p}\hat{z}_{is}^{*}y_{is}a_{sm}^{*\beta-1} \ (i = 1, \dots n;\ m = 1, \dots, k). \tag{6.41}$$

Finally, we obtain the minimizer of a_{jm} with respect to (6.39) as follows:

$$\hat{a}_{jm} = \frac{\sum_{i\in R_m}(\eta_{im}/\lambda_{im})^{\beta-1}\hat{z}_{ij}^{*}y_{ij}}{\sum_{i\in R_m}(\eta_{im}/\lambda_{im})^{\beta}\hat{z}_{ij}^{*}} \ (j = 1, \dots, p;\ m = 1, \dots, k). \tag{6.42}$$

Algorithm

The ZICP2ONMF algorithm, which is based on (3.52), (3.55), (6.36), (6.38), (6.42), and the discussion in Section 3.3 about optimal ϕ, is presented in Algorithm 14.

Algorithm 14 ZICP2ONMF Algorithm

```
**Input** $\boldsymbol{Y} \in \mathbb{R}_+^{n\times p}$, $\beta \in (0,1)$, $k \in \mathbb{N}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_+^{n\times k}$ $(\boldsymbol{f}_{(m)}^{(0)}\boldsymbol{f}_{(u)}^{(0)\prime} = 0;\ m \neq u)$, $\mathcal{R}^{(0)}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p\times k}$, $w^{(0)} \in (0,1)$, $\phi^{(0)} > 0$, $\tau > 0$, $\upsilon \in \mathbb{N}$, $\delta \in \mathbb{N}$, and $\kappa \in \mathbb{N}$
$t \leftarrow 0$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
$z_{ij}^{(t)} \leftarrow \begin{cases} \dfrac{w^{(t)}}{w^{(t)} + (1-w^{(t)})h(0,\phi^{(t)},\beta)\exp\{-(x_{ij}^{(t)})^\beta/(\phi^{(t)}\beta)\}\}} & \text{if } y_{ij} = 0 \\ 0 & \text{if } y_{ij} \neq 0. \end{cases}$
$(i = 1,\dots,n;\ j = 1,\dots,p)$
$\boldsymbol{Z}^{*(t)} \leftarrow \boldsymbol{E}_{n\times p} - \boldsymbol{Z}^{(t)}$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log\left\{w^{(t)}I(y_{ij}=0) + (1-w^{(t)})f_{\mathrm{CP}}(y_{ij}|x_{ij}^{(t)},\phi^{(t)},\beta)\right\}$
**repeat**
    $t \leftarrow t+1$
    $w^{(t)} \leftarrow \dfrac{\sum_{i=1}^{n}\sum_{j=1}^{p} z_{ij}^{(t-1)}}{np}$
    $\lambda_{im}^{(t)} \leftarrow \sum_{s=1}^{p} z_{is}^{*(t-1)}(a_{sm}^{(t-1)})^\beta$ $(i = 1,\dots,n;\ m = 1,\dots,k)$
    $\eta_{im}^{(t)} \leftarrow \sum_{s=1}^{p} z_{is}^{*(t-1)}y_{is}(a_{sm}^{(t-1)})^{\beta-1}$ $(i = 1,\dots n;\ m = 1,\dots,k)$
    $R_m^{(t)} \leftarrow \left\{ i \,\middle|\, \underset{u}{\operatorname{argmin}} \left\{ \dfrac{\left(\eta_{iu}^{(t)}\right)^\beta}{\left(\lambda_{iu}^{(t)}\right)^{\beta-1}} \right\} = m \right\}$ $(m = 1,\dots,k)$
    $a_{jm}^{(t)} \leftarrow \dfrac{\sum_{i\in R_m^{(t)}}(\eta_{im}^{(t)}/\lambda_{im}^{(t)})^{\beta-1}z_{ij}^{*(t-1)}y_{ij}}{\sum_{i\in R_m^{(t)}}(\eta_{im}^{(t)}/\lambda_{im}^{(t)})^{\beta}z_{ij}^{*(t-1)}}$ $(j = 1,\dots,p;\ m = 1,\dots,k)$
    $f_{im}^{(t)} \leftarrow \begin{cases} \dfrac{\sum_{j=1}^{p} z_{ij}^{*(t-1)}y_{ij}(a_{jm}^{(t)})^{\beta-1}}{\sum_{j=1}^{p} z_{ij}^{*(t-1)}(a_{jm}^{(t)})^{\beta}} & \text{if } i \in R_m^{(t)} \\ 0 & \text{if } i \notin R_m^{(t)} \end{cases}$ $(i = 1,\dots,n;\ m = 1,\dots,k)$
    **if** $t \leq \delta$ or $t \bmod \kappa = 0$ **then**
        $\phi^{(t)}$ is obtained as the optimal $\phi$ that optimizes $Q_\phi(\phi)$ given $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(t)}$, $\boldsymbol{Z}^{(t-1)}$, $w^{(t)}$, and $\beta$ using the BFGS quasi-Newton method with constraints $\phi > 0$
    **end if**
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $z_{ij}^{(t)} \leftarrow \begin{cases} \dfrac{w^{(t)}}{w^{(t)} + (1-w^{(t)})h(0,\phi^{(t)},\beta)\exp\{-(x_{ij}^{(t)})^\beta/(\phi^{(t)}\beta)\}\}} & \text{if } y_{ij} = 0 \\ 0 & \text{if } y_{ij} \neq 0. \end{cases}$
    $(i = 1,\dots,n;\ j = 1,\dots,p)$
    $\boldsymbol{Z}^{*(t)} \leftarrow \boldsymbol{E}_{n\times p} - \boldsymbol{Z}^{(t)}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log\left\{w^{(t)}I(y_{ij}=0) + (1-w^{(t)})f_{\mathrm{CP}}(y_{ij}|x_{ij}^{(t)},\phi^{(t)},\beta)\right\}$
**until** $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
**Output** $\boldsymbol{F}^{(t)}$, $\boldsymbol{A}^{(t)}$, $\mathcal{R}^{(t)}$, $\boldsymbol{Z}^{(t)}$, $w^{(t)}$, and $\phi^{(t)}$
```

Chapter 7

Three-factor orthogonal NMF

In this chapter, we present four three-factor orthogonal NMFs. The objective of these methods is to obtain estimators of $\boldsymbol{F} \in \mathbb{R}_+^{n\times k}$, $\boldsymbol{S} \in \mathbb{R}_+^{k\times \ell}$, and $\boldsymbol{A} \in \mathbb{R}_+^{p\times \ell}$ such that $\boldsymbol{X} := \boldsymbol{F}\boldsymbol{S}\boldsymbol{A}'$ is approximated to a given data matrix $\boldsymbol{Y}$ with column orthogonality constraints on $\boldsymbol{F}$ and $\boldsymbol{A}$. In this chapter, R_m, C_q, $\mathcal{R}$, and $\mathcal{C}$ are the same symbols defined in Section 3.2.

7.1 Normal distribution

In this section we present details of three-factor orthogonal NMF based on a normal distribution, named N3ONMF. This method is an extension of the method proposed by Pompili et al. (2014) to a three-factor model. Moreover, this is an improvement of methods proposed by Ding et al. (2006) and Yoo and Choi (2010b) described in Section 3.2.

Objective function

From (3.30), the objective function to be minimized with respect to $\boldsymbol{F}$, $\boldsymbol{S}$, $\boldsymbol{A}$, and σ^2 is

$$Q(\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}, \sigma^2) = \frac{np}{2}\log\{\sigma^2\} + \frac{1}{2\sigma^2}\|\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{S}\boldsymbol{A}'\|^2 + \text{const.} \tag{7.1}$$

Hence, the optimization problem of N3ONMF is

$$\begin{aligned} &\underset{\boldsymbol{F},\boldsymbol{S},\boldsymbol{A},\sigma^2}{\text{argmin}} \left\{Q(\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}, \sigma^2)\right\} \\ &\text{subject to } \boldsymbol{F} \in \mathbb{R}_+^{n\times k}, \boldsymbol{S} \in \mathbb{R}_+^{k\times \ell}, \boldsymbol{A} \in \mathbb{R}_+^{p\times \ell}, \boldsymbol{F}'\boldsymbol{F} = \boldsymbol{I}_k, \text{ and } \boldsymbol{A}'\boldsymbol{A} = \boldsymbol{I}_\ell. \end{aligned} \tag{7.2}$$

The objective function (7.1) is invariant for changes in the length of each row vector of $\boldsymbol{S}$ because the following is satisfied:

$$\begin{aligned} Q(\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}, \sigma^2) &= \frac{np}{2}\log\{\sigma^2\} + \frac{1}{2\sigma^2}\|\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{S}\boldsymbol{A}'\| \\ &= \frac{np}{2}\log\{\sigma^2\} + \frac{1}{2\sigma^2}\|\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{D}_{\boldsymbol{S}'}\boldsymbol{D}_{\boldsymbol{S}'}^{-1}\boldsymbol{S}\boldsymbol{A}'\|^2 \\ &= \frac{np}{2}\log\{\sigma^2\} + \frac{1}{2\sigma^2}\|\boldsymbol{Y} - \boldsymbol{F}^*\boldsymbol{S}^*\boldsymbol{A}'\|^2, \end{aligned} \tag{7.3}$$

where $\boldsymbol{F}^* = \boldsymbol{F}\boldsymbol{D}_{\boldsymbol{S}'}$ and $\boldsymbol{S}^* = \boldsymbol{D}_{\boldsymbol{S}'}^{-1}\boldsymbol{S}$. Consequently,

$$\operatorname{diag}\left(\boldsymbol{S}^*\boldsymbol{A}'\boldsymbol{A}\boldsymbol{S}^{*\prime}\right) = \operatorname{diag}\left(\boldsymbol{S}^*\boldsymbol{S}^{*\prime}\right) = \boldsymbol{I}_k \tag{7.4}$$

is also satisfied. According to (7.3) and (7.4), the optimization problem (7.2) is the same as

$$\begin{aligned} &\underset{\boldsymbol{F},\boldsymbol{S},\boldsymbol{A},\sigma^2}{\operatorname{argmin}} \left\{Q(\boldsymbol{F},\boldsymbol{S},\boldsymbol{A},\sigma^2)\right\} \\ &\text{subject to } \boldsymbol{F} \in \mathbb{R}_+^{n\times k},\ \boldsymbol{S} \in \mathbb{R}_+^{k\times q},\ \boldsymbol{A} \in \mathbb{R}_+^{p\times \ell}, \\ &\qquad \boldsymbol{f}'_{(m)}\boldsymbol{f}_{(r)} = 0\ (m \neq r),\ \operatorname{diag}\left(\boldsymbol{S}\boldsymbol{S}'\right) = \boldsymbol{I}_k,\ \boldsymbol{A}'\boldsymbol{A} = \boldsymbol{I}_\ell. \end{aligned} \tag{7.5}$$

It is noted that $\boldsymbol{F}$ satisfies (3.7) under the condition in (7.5).

Update rules

We show the update rules of the parameters, $\boldsymbol{F}$, $\boldsymbol{S}$, $\boldsymbol{A}$, and ϕ.

Update rule for $\boldsymbol{F}$

Based on the discussion in section 3.2, we can divide the optimization problem of $\boldsymbol{F}$ into that of $\mathcal{R}$ and f_{im} $(i \in R_m;\ m = 1, \ldots, k)$. Hence, the objective function with respect to $\boldsymbol{F}$ and $\boldsymbol{S}$ given that $\boldsymbol{A}$ can be written as follows:

$$\begin{aligned} Q_{\boldsymbol{F},\boldsymbol{S}}(\boldsymbol{F},\boldsymbol{S}) &= \sum_{i=1}^{n} \left\| \boldsymbol{y}_i - \sum_{m=1}^{k} f_{im}\boldsymbol{A}\boldsymbol{s}_m \right\|^2 \\ &= \sum_{m=1}^{k}\sum_{i\in R_m} \|\boldsymbol{y}_i - f_{im}\boldsymbol{A}\boldsymbol{s}_m\|^2 \quad (\because (3.7)) \\ &= \sum_{m=1}^{k}\sum_{i\in R_m} \left\{\boldsymbol{y}_i'\boldsymbol{y}_i - 2f_{im}\boldsymbol{y}_i'\boldsymbol{A}\boldsymbol{s}_m + f_{im}^2\boldsymbol{s}_m'\boldsymbol{A}'\boldsymbol{A}\boldsymbol{s}_m\right\} \\ &= \sum_{m=1}^{k}\sum_{i\in R_m} \left\{\boldsymbol{y}_i'\boldsymbol{y}_i - 2f_{im}\boldsymbol{y}_i'\boldsymbol{A}\boldsymbol{s}_m + f_{im}^2\right\} \quad (\because \operatorname{diag}\left(\boldsymbol{S}\boldsymbol{S}'\right) = \boldsymbol{I}_k). \end{aligned} \tag{7.6}$$

Hence, the minimizer of f_{im} given $\mathcal{R}$ for (7.6) is

$$f_{im} = \begin{cases} \boldsymbol{y}_i'\boldsymbol{A}\boldsymbol{s}_m & \text{if } i \in R_m \\ 0 & \text{if } i \notin R_m \end{cases} \quad (i = 1, \ldots, n;\ m = 1, \ldots, k). \tag{7.7}$$

Substituting (7.7) into (7.6) and rearranging the terms proportional to $\mathcal{R}$ and $\boldsymbol{S}$, we obtain

$$Q_{\mathcal{R},\boldsymbol{S}}(\mathcal{R},\boldsymbol{S}) = \sum_{m=1}^{k}\sum_{i\in R_m} \left\{-\left(\boldsymbol{y}_i'\boldsymbol{A}\boldsymbol{s}_m\right)^2\right\}. \tag{7.8}$$

Therefore, the problem of minimizing $Q_{\boldsymbol{F},\boldsymbol{S}}(\boldsymbol{F},\boldsymbol{S})$ is the same as the problem of minimizing $Q_{\mathcal{R},\boldsymbol{S}}(\mathcal{R},\boldsymbol{S})$. Given $\boldsymbol{S}$ and $\boldsymbol{A}$, the minimizers of $\mathcal{R}$ for $Q_{\mathcal{R},\boldsymbol{S}}(\mathcal{R},\boldsymbol{S})$ are derived by, e.g., a k-means algorithm such that

$$R_m = \left\{ i \;\middle|\; \underset{r}{\operatorname{argmax}} \left(\boldsymbol{y}_i'\boldsymbol{A}\boldsymbol{s}_r\right) = m \right\} \quad (m = 1, \ldots, k). \tag{7.9}$$

Update rule for $\boldsymbol{S}$

Because (7.8) can be rewritten such that

$$Q_{\mathcal{R},\boldsymbol{S}}(\mathcal{R},\boldsymbol{S}) = -\sum_{m=1}^{k} \|\boldsymbol{Y}_m \boldsymbol{A} \boldsymbol{s}_m\|^2, \tag{7.10}$$

where $\boldsymbol{Y}_m$ $(m = 1, \dots, k)$ is the same matrix defined in section 6.1, the minimizer of $\boldsymbol{S}$, given $\mathcal{R}$ and $\boldsymbol{A}$, can be obtained as the first nonnegative singular vector of $\boldsymbol{A}'\boldsymbol{Y}_m'$ as follows:

$$\boldsymbol{s}_m = \varDelta(\boldsymbol{A}'\boldsymbol{Y}_m') \quad (m = 1, \dots, k). \tag{7.11}$$

Update rule for $\boldsymbol{A}$

If we regard the approximation problem as $\boldsymbol{Y}' \approx \boldsymbol{A}\boldsymbol{S}'\boldsymbol{F}'$, the update rules of $\boldsymbol{A}$, $\mathcal{C}$, and $\boldsymbol{S}$ can be derived similarly to (7.7), (7.9), and (7.11) as follows:

$$a_{jq} = \begin{cases} \boldsymbol{y}_{(j)}'\boldsymbol{F}\boldsymbol{s}_{(q)} & (j \in C_q) \\ 0 & (j \notin C_q) \end{cases} \quad (j = 1, \dots p;\ q = 1, \dots, \ell), \tag{7.12}$$

$$C_q = \left\{ j \,\middle|\, \operatorname*{argmax}_{c} \left\{ \boldsymbol{y}_{(j)}'\boldsymbol{F}\boldsymbol{s}_{(c)} \right\} = q \right\} \quad (q = 1, \dots, \ell), \tag{7.13}$$

$$\boldsymbol{s}_{(q)} = \varDelta(\boldsymbol{F}'\boldsymbol{Y}_{(q)}) \quad (q = 1, \dots, \ell), \tag{7.14}$$

where $\boldsymbol{Y}_{(q)}$ $(q = 1, \dots, \ell)$ is an $n \times |C_q|$ submatrix of $\boldsymbol{Y}$ consisting of the column vectors of C_q.

Algorithm

The ZICP2ONMF algorithm, which is derived from (7.7), (7.9), (7.11), (7.12), (7.13), (7.14), and (3.31), is presented in Algorithm 15.

7.2 Poisson distribution

In this section we present details of the three-factor orthogonal NMF based on a Poisson distribution, named P3ONMF. This factorization is a modified version of N3ONMF described in Section 7.1, and it assumes that the data follow a Poisson distribution. Although a multiplicative updating algorithm for three-factor NMF under this assumption was previously proposed by Yoo and Choi (2009), the orthogonal constraint was not imposed thereon. In contrast, our algorithm is not based on a multiplicative updating algorithm; instead, ours is based on the weighted spherical k-means algorithm, and the orthogonality constraints are imposed on it.

Objective function

From (3.35), the objective function to be minimized with respect to $\boldsymbol{F}$, $\boldsymbol{S}$, and $\boldsymbol{A}$ is

$$Q(\boldsymbol{F},\boldsymbol{S},\boldsymbol{A}) = -\sum_{i=1}^{n}\sum_{j=1}^{p} y_{ij} \log\left\{ \sum_{m=1}^{k}\sum_{q=1}^{\ell} f_{im}s_{mq}a_{jq} \right\} + \sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij} + \text{const.} \tag{7.15}$$

Algorithm 15 N3ONMF Algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_{+}^{n\times p}$, $k \in \mathbb{N}$, $\ell \in \mathbb{N}$, $\boldsymbol{S}^{(0)} \in \mathbb{R}_{+}^{k\times \ell}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_{+}^{p\times \ell}$ $(\mathrm{diag}(\boldsymbol{A}^{(0)\prime}\boldsymbol{A}^{(0)}) = \boldsymbol{I}_{\ell})$, $\tau > 0$, and $\upsilon \in \mathbb{N}$
$t \leftarrow 0$
$L^{(t)} \leftarrow -\infty$
repeat
    $t \leftarrow t+1$
    $R_m^{(t)} \leftarrow \left\{ i \,\middle|\, \underset{r}{\mathrm{argmax}} \left\{ \boldsymbol{y}_i' \boldsymbol{A}^{(t-1)} \boldsymbol{s}_r^{(t-1)} \right\} = m \right\}$ $(m = 1,\ldots,k)$
    Set $\boldsymbol{Y}_m^{(t)}$ as the submatrix of $\boldsymbol{Y}$ consisting of the row vectors of $R_m^{(t)}$ for $m = 1,\ldots,k$
    $\boldsymbol{s}_m^{*(t)} \leftarrow \Delta(\boldsymbol{A}^{(t-1)\prime} \boldsymbol{Y}_m^{(t)\prime})$ $(m = 1,\ldots,k)$
    $f_{im}^{*(t)} \leftarrow \begin{cases} \boldsymbol{y}_i' \boldsymbol{A}^{(t-1)} \boldsymbol{s}_m^{*(t)} & (i \in R_m^{(t)}) \\ 0 & (i \notin R_m^{(t)}) \end{cases}$ $(i = 1,\ldots,n;\ m = 1,\ldots,k)$
    $\boldsymbol{F}^{(t)} \leftarrow \boldsymbol{F}^{*(t)} \boldsymbol{D}_{\boldsymbol{F}^{*(t)}}^{-1}$
    $\boldsymbol{S}^{\dagger(t)} \leftarrow \boldsymbol{D}_{\boldsymbol{F}^{*(t)}} \boldsymbol{S}^{*(t)}$
    $C_q^{(t)} \leftarrow \left\{ j \,\middle|\, \underset{c}{\mathrm{argmax}} \left\{ \boldsymbol{y}_{(j)}' \boldsymbol{F}^{(t)} \boldsymbol{s}_{(c)}^{\dagger(t)} \right\} = q \right\}$ $(q = 1,\ldots,\ell)$
    Set $\boldsymbol{Y}_{(q)}^{(t)}$ as the submatrix of $\boldsymbol{Y}$ consisting of the column vectors of $C_q^{(t)}$ for $q = 1,\ldots,\ell$
    $\boldsymbol{s}_{(q)}^{\star(t)} \leftarrow \Delta(\boldsymbol{F}^{(t)\prime} \boldsymbol{Y}_{(q)}^{(t)})$ $(q = 1,\ldots,\ell)$
    $a_{jq}^{*(t)} \leftarrow \begin{cases} \boldsymbol{y}_{(j)}' \boldsymbol{F}^{(t)} \boldsymbol{s}_{(q)}^{\star(t)} & (j \in C_q^{(t)}) \\ 0 & (j \notin C_q^{(t)}) \end{cases}$ $(j = 1,\ldots p;\ q = 1,\ldots,\ell)$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{*(t)} \boldsymbol{D}_{\boldsymbol{A}^{*(t)}}^{-1}$
    $\boldsymbol{S}^{(t)} \leftarrow \boldsymbol{S}^{\star(t)} \boldsymbol{D}_{\boldsymbol{A}^{*(t)}}$
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)} \boldsymbol{S}^{(t)} \boldsymbol{A}^{(t)\prime}$
    $(\sigma^{(t)})^2 \leftarrow \frac{1}{np} \|\boldsymbol{Y} - \boldsymbol{X}^{(t)}\|^2$
    $L^{(t)} \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{p} \log f_{\mathrm{N}}\left( y_{ij} \,\middle|\, x_{ij}^{(t)}, (\sigma^{(t)})^2 \right)$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$, $\boldsymbol{A}^{(t)}$, $\mathcal{R}^{(t)}$, $\mathcal{C}^{(t)}$, and $(\sigma^{(t)})^2$
```

Hence, the optimization problem of P3ONMF is

$$\underset{\boldsymbol{F},\boldsymbol{S},\boldsymbol{A}}{\mathrm{argmin}} \{Q(\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A})\}$$
$$\text{subject to } \boldsymbol{F} \in \mathbb{R}_{+}^{n\times k}, \boldsymbol{S} \in \mathbb{R}_{+}^{k\times q}, \boldsymbol{A} \in \mathbb{R}_{+}^{p\times \ell}, \boldsymbol{f}_{(m)}' \boldsymbol{f}_{(r)} = 0 \ (m \neq r), \boldsymbol{a}_{(q)}' \boldsymbol{a}_{(c)} = 0 \ (q \neq c). \tag{7.16}$$

The constrained condition has been slightly changed from (7.2). It is noted that under these conditions, we have (3.7) and (3.8) for the factor matrices $\boldsymbol{F}$ and $\boldsymbol{A}$, respectively.

Update rules

We show the update rules of the parameters, $\boldsymbol{F}$, $\boldsymbol{S}$, and $\boldsymbol{A}$.

Update rule for $\boldsymbol{F}$

If we treat the $\boldsymbol{AS}'$ as the right-hand factor matrix in two-factor ONMF, the form of the objective function with respect to $\boldsymbol{F}$ is the same as (6.14). Moreover, based on the discussion in section 3.2, we can divide the optimization problem of $\boldsymbol{F}$ into that of $\mathcal{R}$ and f_{im} $(i \in R_m;\ m = 1, \dots, k)$. Hence, we can obtain their update rule in the same form as (6.15) and (6.17) as follows:

$$f_{im} = \begin{cases} \dfrac{\sum_{j=1}^{p} y_{ij}}{\sum_{j=1}^{p} [\boldsymbol{SA}']_{mj}} & \text{if } i \in R_m \\ 0 & \text{if } i \notin R_m \end{cases} \quad (i = 1, \dots, n;\ m = 1, \dots, k), \tag{7.17}$$

$$R_m = \left\{ i \,\middle|\, \underset{r}{\operatorname{argmax}} \left\{ \sum_{j=1}^{p} y_{ij} \log \left\{ \frac{[\boldsymbol{SA}']_{rj}}{\sum_{\gamma=1}^{p} [\boldsymbol{SA}']_{r\gamma}} \right\} \right\} = m \right\}. \tag{7.18}$$

Update rule for $\boldsymbol{S}$

Considering $\boldsymbol{A}$ in (6.18) as $\boldsymbol{AS}'$, the objective function with respect to $\boldsymbol{S}$ can be written as follows:

$$\begin{aligned} Q_{\boldsymbol{S}}(\boldsymbol{S}) &= \sum_{m=1}^{k} \sum_{i \in R_m} \sum_{j=1}^{p} y_{ij} \log \left\{ \sum_{\gamma=1}^{p} \sum_{q=1}^{\ell} s_{mq} a_{\gamma q} \right\} - \sum_{m=1}^{k} \sum_{i \in R_m} \sum_{j=1}^{p} y_{ij} \log \left\{ \sum_{q=1}^{\ell} s_{mq} a_{\gamma q} \right\} \\ &= \sum_{m=1}^{k} \sum_{i \in R_m} \sum_{j=1}^{p} y_{ij} \log \left\{ \sum_{\gamma=1}^{p} \sum_{q=1}^{\ell} s_{mq} a_{\gamma q} \right\} - \sum_{m=1}^{k} \sum_{i \in R_m} \sum_{q=1}^{\ell} \sum_{j \in C_q} y_{ij} \log \left\{ s_{mq} a_{jq} \right\}. \end{aligned} \tag{7.19}$$

Here, we use (3.8). It is difficult to directly obtain the minimizer of s_{mq} for (7.19) because the summation of s_{mq} occurs in the log function. However, we can obtain the optimal s_{mq} using the auxiliary function method. Because the log function $f(x) = \log(x)$ is concave, we can obtain the following auxiliary function of (7.19) from the inequality (4.23):

$$\begin{aligned} & Q_{\boldsymbol{S}}^{\text{aux}}(\boldsymbol{S}, \boldsymbol{S}^*) \\ &= \sum_{m=1}^{k} \sum_{i \in R_m} \sum_{j=1}^{p} y_{ij} \left(\log \lambda_m + \frac{1}{\lambda_m} \left(\sum_{\gamma=1}^{p} \sum_{q=1}^{\ell} s_{mq} a_{\gamma q} - \lambda_m \right) \right) \\ &- \sum_{m=1}^{k} \sum_{i \in R_m} \sum_{q=1}^{\ell} \sum_{j \in C_q} y_{ij} \log \left\{ s_{mq} a_{jq} \right\}, \end{aligned} \tag{7.20}$$

where $\lambda_m = \sum_{q=1}^{\ell} \sum_{j=1}^{p} s_{mq}^* a_{jq}$ $(m = 1, \dots, k)$ and s_{mq}^* is the current s_{mq}. We obtain the following update equation of s_{mq} as a minimizer for $Q_{\boldsymbol{S}}^{\text{aux}}(\boldsymbol{S}, \boldsymbol{S}^*)$:

$$s_{mq} = \frac{\lambda_m \sum_{i \in R_m} \sum_{j \in C_q} y_{ij}}{\left(\sum_{i \in R_m} \sum_{j=1}^{p} y_{ij} \right) \left(\sum_{j=1}^{p} a_{jq} \right)}. \tag{7.21}$$

Update rule for $\boldsymbol{A}$

If we regard the model as $\boldsymbol{Y}' \approx \boldsymbol{A}\boldsymbol{S}'\boldsymbol{F}'$, the update rules of $\mathcal{C}$, $\boldsymbol{S}$, and $\boldsymbol{A}$ can be derived similarly to (7.17), (7.18), and (7.21) as follows:

$$a_{jq} = \begin{cases} \dfrac{\sum_{i=1}^{n} y_{ij}}{\sum_{i=1}^{n} [\boldsymbol{F}\boldsymbol{S}]_{iq}} & (j \in C_q) \\ 0 & (j \notin C_q) \end{cases} \quad (j = 1, \ldots p;\ q = 1, \ldots, \ell), \tag{7.22}$$

$$C_q = \left\{ j \,\middle|\, \underset{c}{\operatorname{argmax}} \left\{ \sum_{i=1}^{n} y_{ij} \log \left\{ \frac{[\boldsymbol{F}\boldsymbol{S}]_{ic}}{\sum_{v=1}^{n} [\boldsymbol{F}\boldsymbol{S}]_{vc}} \right\} \right\} = q \right\} \quad (q = 1, \ldots, \ell), \tag{7.23}$$

$$s_{mq} = \frac{\lambda_q^* \sum_{i \in R_m} \sum_{j \in C_q} y_{ij}}{\left(\sum_{j \in C_q} \sum_{i=1}^{n} y_{ij} \right) \left(\sum_{i=1}^{n} f_{im} \right)} \quad (m = 1, \ldots, k;\ q = 1, \ldots, \ell), \tag{7.24}$$

$$\text{where} \quad \lambda_q^* = \sum_{i=1}^{n} \sum_{m=1}^{k} s_{mq}^* f_{im} \quad (q = 1, \ldots, \ell).$$

Algorithm

The P3ONMF algorithm, which is based on (7.17), (7.18), (7.21), (7.22), (7.23), and (7.24), is presented in Algorithm 16.

7.3 Compound Poisson-gamma distribution

In this section we present details of three-factor orthogonal NMF based on a compound Poisson-gamma distribution, named CP3ONMF. It is a modified version of N3ONMF and P3ONMF described in Sections 7.1 and 7.2, and it assumes that the data follow a compound Poisson-gamma distribution.

Objective function

From (3.39), the objective function to be minimized with respect to $\boldsymbol{F}$, $\boldsymbol{S}$, $\boldsymbol{A}$, and ϕ is

$$\begin{aligned} & Q(\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}, \phi) \\ = & -\sum_{i=1}^{n} \sum_{j=1}^{p} \log\{h(y_{ij}, \phi, \beta)\} \\ & - \frac{1}{\phi} \sum_{i=1}^{n} \sum_{j=1}^{p} \left(\frac{y_{ij} (\sum_{m=1}^{k} \sum_{q=1}^{\ell} f_{im} s_{mq} a_{jq})^{\beta - 1}}{\beta - 1} - \frac{(\sum_{m=1}^{k} \sum_{q=1}^{\ell} f_{im} s_{mq} a_{jq})^{\beta}}{\beta} \right). \end{aligned} \tag{7.25}$$

Hence, the optimization problem of CP3ONMF is

$$\begin{aligned} & \underset{\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}, \phi}{\operatorname{argmin}} \{ Q(\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}, \phi) \} \\ & \text{subject to } \boldsymbol{F} \in \mathbb{R}_+^{n \times k}, \boldsymbol{S} \in \mathbb{R}_+^{k \times q}, \boldsymbol{A} \in \mathbb{R}_+^{p \times \ell}, \boldsymbol{f}_{(m)}' \boldsymbol{f}_{(r)} = 0 \ (m \neq r), \boldsymbol{a}_{(q)}' \boldsymbol{a}_{(c)} = 0 \ (q \neq c). \end{aligned} \tag{7.26}$$

Algorithm 16 P3ONMF

```
**Input** $\boldsymbol{Y} \in \mathbb{R}_+^{n\times p}$, $k \in \mathbb{N}$, $\ell \in \mathbb{N}$, $\boldsymbol{S}^{(0)} \in \mathbb{R}_+^{k\times\ell}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p\times\ell}$ ($\boldsymbol{a}_{(q)}^{(0)\prime}\boldsymbol{a}_{(c)}^{(0)} = 0\ (q \neq c)$), $\mathcal{C}^{(0)}$, $\tau > 0$, and $\upsilon \in \mathbb{N}$
$t \leftarrow 0$
$L^{(t)} \leftarrow -\infty$
**repeat**
    $t \leftarrow t + 1$
    $R_m^{(t)} \leftarrow \left\{ i \,\middle|\, \operatorname*{argmax}_{r} \left\{ \sum_{j=1}^{p} y_{ij} \log \left\{ \frac{[\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime}]_{rj}}{\sum_{\gamma=1}^{p}[\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime}]_{r\gamma}} \right\} \right\} = m \right\}$
    $(m = 1, \dots, k)$
    $\lambda_m^{(t)} \leftarrow \sum_{j=1}^{p} \sum_{q=1}^{\ell} s_{mq}^{(t-1)} a_{jq}^{(t-1)}$
    $s_{mq}^{*(t)} \leftarrow \frac{\lambda_m^{(t)} \sum_{i \in R_m^{(t)}} \sum_{j \in C_q^{(t-1)}} y_{ij}}{\left(\sum_{i \in R_m^{(t)}} \sum_{j=1}^{p} y_{ij}\right)\left(\sum_{j=1}^{p} a_{jq}^{(t-1)}\right)} \quad (m = 1, \dots, k;\ q = 1, \dots, \ell)$
    $f_{im}^{(t)} \leftarrow \begin{cases} \frac{\sum_{j=1}^{p} y_{ij}}{\sum_{j=1}^{p}[\boldsymbol{S}^{*(t)}\boldsymbol{A}^{(t-1)\prime}]_{mj}} & (i \in R_m^{(t)}) \\ 0 & (i \notin R_m^{(t)}) \end{cases} \quad (i = 1, \dots, n;\ m = 1, \dots, k)$
    $C_q \leftarrow \left\{ j \,\middle|\, \operatorname*{argmax}_{c} \left\{ \sum_{i=1}^{n} y_{ij} \log \left\{ \frac{[\boldsymbol{F}^{(t)}\boldsymbol{S}^{*(t)}]_{ic}}{\sum_{\upsilon=1}^{n}[\boldsymbol{F}^{(t)}\boldsymbol{S}^{*(t)}]_{\upsilon c}} \right\} \right\} = q \right\} \quad (q = 1, \dots, \ell)$
    $\lambda_q^{*(t)} \leftarrow \sum_{i=1}^{n} \sum_{m=1}^{k} s_{mq}^{*(t)} f_{im}^{(t)} \quad (q = 1, \dots, \ell)$
    $s_{mq}^{(t)} \leftarrow \frac{\lambda_q^{*(t)} \sum_{i \in R_m^{(t)}} \sum_{j \in C_q^{(t)}} y_{ij}}{\left(\sum_{j \in C_q^{(t)}} \sum_{i=1}^{n} y_{ij}\right)\left(\sum_{i=1}^{n} f_{im}^{(t)}\right)} \quad (m = 1, \dots, k;\ q = 1, \dots, \ell)$
    $a_{jq}^{(t)} \leftarrow \begin{cases} \frac{\sum_{i=1}^{n} y_{ij}}{\sum_{i=1}^{n}[\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}]_{iq}} & (j \in C_q^{(t)}) \\ 0 & (j \notin C_q^{(t)}) \end{cases} \quad (j = 1, \dots p;\ q = 1, \dots, \ell)$
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{p} \log f_{\mathrm{P}}\left(y_{ij} \,\middle|\, x_{ij}^{(t)}\right)$
**until** $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
**Output** $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$, $\boldsymbol{A}^{(t)}$, $\mathcal{R}^{(t)}$, and $\mathcal{C}^{(t)}$
```

Update rules

We show the update rules of the parameters, $\boldsymbol{F}$, $\boldsymbol{S}$, and $\boldsymbol{A}$. ϕ is obtained as described in Section 3.3.

Update rule for $\boldsymbol{F}$

For P3ONMF, if we treat the $\boldsymbol{AS}'$ as the right hand factor matrix in two-factor ONMF, the form of the objective function with respect to $\boldsymbol{F}$ is the same as (6.23). Moreover, based on the discussion in section 3.2, we can divide the optimization problem of $\boldsymbol{F}$ into that of $\mathcal{R}$ and f_{im} $(i \in R_m;\ m = 1, \dots, k)$. Hence, we can obtain their update rule in the same form as (6.24) and (6.26) as follows:

$$f_{im} = \begin{cases} \dfrac{\sum_{j=1}^{p} y_{ij} [\boldsymbol{S}\boldsymbol{A}']_{mj}^{\beta-1}}{\sum_{j=1}^{p} [\boldsymbol{S}\boldsymbol{A}']_{mj}^{\beta}} & \text{if } i \in R_m \\ 0 & \text{if } i \notin R_m \end{cases} \quad (i = 1, \dots, n;\ m = 1, \dots, k), \tag{7.27}$$

$$R_m = \left\{ i \,\middle|\, \underset{r}{\operatorname{argmin}} \left\{ \frac{\left\{ \sum_{j=1}^{p} y_{ij} [\boldsymbol{S}\boldsymbol{A}']_{rj}^{\beta-1} \right\}^{\beta}}{\left\{ \sum_{j=1}^{p} [\boldsymbol{S}\boldsymbol{A}']_{rj}^{\beta} \right\}^{\beta-1}} \right\} = m \right\}. \tag{7.28}$$

Update rule for $\boldsymbol{S}$

Considering $\boldsymbol{A}$ in (6.25) as $\boldsymbol{A}\boldsymbol{S}'$, the objective function with respect to $\boldsymbol{S}$ can be written as follows:

$$\begin{aligned} Q_{\boldsymbol{S}}(\boldsymbol{S}) &= -\frac{1}{\beta(\beta-1)} \sum_{m=1}^{k} \sum_{i \in R_m} \frac{\left\{ \sum_{j=1}^{p} y_{ij} (\sum_{q=1}^{\ell} s_{mq} a_{jq})^{\beta-1} \right\}^{\beta}}{\left\{ \sum_{j=1}^{p} (\sum_{q=1}^{\ell} s_{mq} a_{jq})^{\beta} \right\}^{\beta-1}} \\ &= -\frac{1}{\beta(\beta-1)} \sum_{m=1}^{k} \sum_{i \in R_m} \frac{\left\{ \sum_{q=1}^{\ell} \sum_{j \in C_q} y_{ij} s_{mq}^{\beta-1} a_{jq}^{\beta-1} \right\}^{\beta}}{\left\{ \sum_{q=1}^{\ell} \sum_{j \in C_q} s_{mq}^{\beta} a_{jq}^{\beta} \right\}^{\beta-1}}. \end{aligned} \tag{7.29}$$

Here, we use (3.8). It is difficult to directly obtain the minimizer of s_{mq} for (7.29) because the summation of s_{mq} occurs in the two power functions. However, we can obtain the optimal s_{mq} using the auxiliary function method in a manner similar to CP2ONMF. We find that (7.29) contains the function (6.27). Hence, the inequality (6.28) enables us to obtain the following auxiliary function of (7.29):

$$\begin{aligned} &Q_{\boldsymbol{S}}^{\text{aux}}(\boldsymbol{S}, \boldsymbol{S}^*) \\ &= \sum_{m=1}^{k} \sum_{i \in R_m} \left\{ -\frac{1}{\beta(\beta-1)} \frac{\eta_{im}^{\beta}}{\lambda_m^{\beta-1}} + \frac{1}{\beta} \left(\frac{\eta_{im}}{\lambda_m} \right)^{\beta} \left(\sum_{q=1}^{\ell} \sum_{j \in C_q} s_{mq}^{\beta} a_{jq}^{\beta} - \lambda_m \right) \right. \\ &\left. + \frac{1}{1-\beta} \left(\frac{\eta_{im}}{\lambda_m} \right)^{\beta-1} \left(\sum_{q=1}^{\ell} \sum_{j \in C_q} y_{ij} s_{mq}^{\beta-1} a_{jq}^{\beta-1} - \eta_{im} \right) \right\}, \end{aligned} \tag{7.30}$$

where

$$\lambda_m = \sum_{q=1}^{\ell} \sum_{j \in C_q} s_{mq}^{*\beta} a_{jq}^{\beta} \ (m = 1, \dots, k) \tag{7.31}$$

$$\text{and } \eta_{im} = \sum_{q=1}^{\ell} \sum_{j \in C_q} y_{ij} s_{mq}^{*\beta-1} a_{jq}^{\beta-1} \ (i = 1, \dots n;\ m = 1, \dots, k). \tag{7.32}$$

We obtain the following update equation of s_{mq} as a minimizer for $Q_{\boldsymbol{S}}^{\text{aux}}(\boldsymbol{S}, \boldsymbol{S}^*)$:

$$s_{mq} = \frac{\lambda_m \sum_{i \in R_m} \sum_{j \in C_q} \eta_{im}^{\beta-1} y_{ij} a_{jq}^{\beta-1}}{\sum_{i \in R_m} \sum_{j \in C_q} \eta_{im}^{\beta} a_{jq}^{\beta}}. \tag{7.33}$$

Update rule for $\boldsymbol{A}$

If we regard the model as $\boldsymbol{Y}' \approx \boldsymbol{A}\boldsymbol{S}'\boldsymbol{F}'$, the update rules of $\mathcal{C}$, $\boldsymbol{S}$, and $\boldsymbol{A}$ can be derived similarly to (7.27), (7.28), and (7.33) as follows:

$$a_{jq} = \begin{cases} \dfrac{\sum_{i=1}^{n} y_{ij} [\boldsymbol{F}\boldsymbol{S}]_{iq}^{\beta-1}}{\sum_{i=1}^{n} [\boldsymbol{F}\boldsymbol{S}]_{iq}^{\beta}} & (j \in C_q) \\ 0 & (j \notin C_q) \end{cases} \quad (j = 1, \ldots p;\ q = 1, \ldots, \ell), \tag{7.34}$$

$$C_q = \left\{ j \,\middle|\, \underset{c}{\operatorname{argmin}} \left\{ \frac{\left\{ \sum_{i=1}^{n} y_{ij} [\boldsymbol{F}\boldsymbol{S}]_{ic}^{\beta-1} \right\}^{\beta}}{\left\{ \sum_{i=1}^{n} [\boldsymbol{F}\boldsymbol{S}]_{ic}^{\beta} \right\}^{\beta-1}} \right\} = q \right\} \quad (q = 1, \ldots, \ell), \tag{7.35}$$

$$s_{mq} = \frac{\lambda_q^* \sum_{i \in R_m} \sum_{j \in C_q} (\eta_{jq}^*)^{\beta-1} y_{ij} f_{im}^{\beta-1}}{\sum_{i \in R_m} \sum_{j \in C_q} (\eta_{jq}^*)^{\beta} f_{im}^{\beta}} \quad (m = 1, \ldots, k;\ q = 1, \ldots, \ell), \tag{7.36}$$

where

$$\lambda_q^* = \sum_{m=1}^{k} \sum_{i \in R_m} s_{mq}^{*\beta} f_{im}^{\beta} \ (q = 1, \ldots, \ell), \tag{7.37}$$

$$\eta_{jq}^* = \sum_{m=1}^{k} \sum_{i \in R_m} y_{ij} s_{mq}^{*\beta-1} f_{im}^{\beta-1} \ (j = 1, \ldots p;\ q = 1, \ldots, \ell). \tag{7.38}$$

Algorithm

The CP3ONMF algorithm, which is derived from (7.27), (7.28), (7.33), (7.34), (7.35), (7.36), and the discussion in Section 3.3 about optimal ϕ, is presented in Algorithm 17.

7.4 Zero-inflated compound Poisson-gamma distribution

In this section we present details of three-factor orthogonal NMF based on a zero-inflated compound Poisson-gamma distribution, named ZICP3ONMF. This NMF is a modified version of ZICP2ONMF, CP3ONMF, and ZICP3NMF.

Objective function

From (3.44), the objective function is

$$\begin{aligned} Q(\boldsymbol{F}, \boldsymbol{S}, \boldsymbol{A}, w, \phi) = -\sum_{i=1}^{n} \sum_{j=1}^{p} \log \Bigg[& wI(y_{ij} = 0) \\ & + (1-w) h(y_{ij}, \phi, \beta) \exp \left\{ \frac{1}{\phi} \left(\frac{y_{ij} (\sum_{m=1}^{k} \sum_{q=1}^{\ell} f_{im} s_{mq} a_{jq})^{\beta-1}}{\beta - 1} \right. \right. \\ & \left. \left. - \frac{(\sum_{m=1}^{k} \sum_{q=1}^{\ell} f_{im} s_{mq} a_{jq})^{\beta}}{\beta} \right) \right\} \Bigg]. \end{aligned} \tag{7.39}$$

Algorithm 17 CP3ONMF algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_{+}^{n\times p}$, $\beta \in (0,1)$, $k \in \mathbb{N}$, $\ell \in \mathbb{N}$, $\boldsymbol{S}^{(0)} \in \mathbb{R}_{+}^{k\times \ell}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_{+}^{p\times \ell}$ $(\boldsymbol{a}_{(q)}^{(0)\prime}\boldsymbol{a}_{(c)}^{(0)} = 0\ (q \neq c))$, $\mathcal{C}^{(0)}$, $\phi^{(0)} > 0$, $\tau > 0$, $\upsilon \in \mathbb{N}$, $\delta \in \mathbb{N}$, and $\kappa \in \mathbb{N}$
$L^{(0)} \leftarrow -\infty$
$t \leftarrow 0$
repeat
    $t \leftarrow t+1$
    $R_m^{(t)} \leftarrow \left\{ i \,\middle|\, \mathop{\mathrm{argmin}}_{r} \left\{ \frac{\left\{\sum_{j=1}^{p} y_{ij}[\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime}]_{rj}^{\beta-1}\right\}^{\beta}}{\left\{\sum_{j=1}^{p}[\boldsymbol{S}^{(t-1)}\boldsymbol{A}^{(t-1)\prime}]_{rj}^{\beta}\right\}^{\beta-1}} \right\} = m \right\} \quad (m = 1,\dots,k)$
    $\lambda_m^{(t)} \leftarrow \sum_{q=1}^{\ell}\sum_{j\in C_q^{(t-1)}} (s_{mq}^{(t-1)})^{\beta}(a_{jq}^{(t-1)})^{\beta} \quad (\ m = 1,\dots,k)$
    $\eta_{im}^{(t)} \leftarrow \sum_{q=1}^{\ell}\sum_{j\in C_q^{(t-1)}} y_{ij}(s_{mq}^{(t-1)})^{\beta-1}(a_{jq}^{(t-1)})^{\beta-1} \quad (i = 1,\dots,n;\ m = 1,\dots,k)$
    $s_{mq}^{*(t)} \leftarrow \frac{\lambda_m^{(t)}\sum_{i\in R_m^{(t)}}\sum_{j\in C_q^{(t-1)}}(\eta_{im}^{(t)})^{\beta-1}y_{ij}(a_{jq}^{(t-1)})^{\beta-1}}{\sum_{i\in R_m^{(t)}}\sum_{j\in C_q^{(t-1)}}(\eta_{im}^{(t)})^{\beta}(a_{jq}^{(t-1)})^{\beta}} \quad (m = 1,\dots,k;\ q = 1,\dots,\ell)$
    $f_{im}^{(t)} \leftarrow \begin{cases} \frac{\sum_{j=1}^{p} y_{ij}[\boldsymbol{S}^{*(t)}\boldsymbol{A}^{(t-1)\prime}]_{mj}^{\beta-1}}{\sum_{j=1}^{p}[\boldsymbol{S}^{*(t)}\boldsymbol{A}^{(t-1)\prime}]_{mj}^{\beta}} & (i \in R_m) \\ 0 & (i \notin R_m) \end{cases}$
    $C_q^{(t)} \leftarrow \left\{ j \,\middle|\, \mathop{\mathrm{argmin}}_{c} \left\{ \frac{\left\{\sum_{i=1}^{n} y_{ij}[\boldsymbol{F}^{(t)}\boldsymbol{S}^{*(t)}]_{ic}^{\beta-1}\right\}^{\beta}}{\left\{\sum_{i=1}^{n}[\boldsymbol{F}^{(t)}\boldsymbol{S}^{*(t)}]_{ic}^{\beta}\right\}^{\beta-1}} \right\} = q \right\} \quad (q = 1,\dots,\ell)$
    $\lambda_q^{*(t)} \leftarrow \sum_{m=1}^{k}\sum_{i\in R_m^{(t)}} (s_{mq}^{*(t)})^{\beta}(f_{im}^{(t)})^{\beta} \quad (q = 1,\dots,\ell)$
    $\eta_{jq}^{*(t)} \leftarrow \sum_{m=1}^{k}\sum_{i\in R_m^{(t)}} y_{ij}(s_{mq}^{*(t)})^{\beta-1}(f_{im}^{(t)})^{\beta-1} \quad (j = 1,\dots,p;\ q = 1,\dots,\ell)$
    $s_{mq}^{(t)} \leftarrow \frac{\lambda_q^{*(t)}\sum_{i\in R_m^{(t)}}\sum_{j\in C_q^{(t)}}(\eta_{jq}^{*(t)})^{\beta-1}y_{ij}(f_{im}^{(t)})^{\beta-1}}{\sum_{i\in R_m^{(t)}}\sum_{j\in C_q^{(t)}}(\eta_{jq}^{*(t)})^{\beta}(f_{im}^{(t)})^{\beta}} \quad (m = 1,\dots,k;\ q = 1,\dots,\ell)$
    $a_{jq}^{(t)} \leftarrow \begin{cases} \frac{\sum_{i=1}^{n} y_{ij}[\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}]_{iq}^{\beta-1}}{\sum_{i=1}^{n}[\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}]_{iq}^{\beta}} & (j \in C_q) \\ 0 & (j \notin C_q) \end{cases}$
    if $t \leq \delta$ or $t \bmod \kappa = 0$ then
        $\phi^{(t)}$ is obtained as the optimal $\phi$ that optimizes $Q_{\phi}(\phi)$ given $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$, $\boldsymbol{A}^{(t)}$, and $\beta$ using the BFGS quasi-Newton method with constraints $\phi > 0$
    end if
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_{\mathrm{CP}}\left(y_{ij} \,\middle|\, x_{ij}^{(t)}, \phi^{(t)}, \beta\right)$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$, $\boldsymbol{A}^{(t)}$, $\mathcal{R}^{(t)}$, $\mathcal{C}^{(t)}$, $\phi^{(t)}$
```

Hence, the optimization problem is as follows:

$$\begin{aligned} &\mathop{\mathrm{argmin}}_{\boldsymbol{F},\boldsymbol{S},\boldsymbol{A},w,\phi} \{Q(\boldsymbol{F},\boldsymbol{S},\boldsymbol{A},w,\phi)\} \\ &\text{subject to } \boldsymbol{F} \in \mathbb{R}_{+}^{n\times k}, \boldsymbol{S} \in \mathbb{R}_{+}^{k\times q}, \boldsymbol{A} \in \mathbb{R}_{+}^{p\times \ell}, \boldsymbol{f}_{(m)}'\boldsymbol{f}_{(r)} = 0\ (m \neq r), \boldsymbol{a}_{(q)}'\boldsymbol{a}_{(c)} = 0\ (q \neq c). \end{aligned} \tag{7.40}$$

However, the update rule of $\boldsymbol{F}$, $\boldsymbol{S}$, $\boldsymbol{A}$, w, and ϕ is obtained as these optimizers, which minimize (3.51).

Update rules

We show the update rules of the parameters, $\boldsymbol{F}$, $\boldsymbol{S}$, and $\boldsymbol{A}$. The update rule of $\hat{\boldsymbol{Z}}$ and w is obtained as (3.52) and (3.55). ϕ is obtained as described in Section 3.3.

Update rule for $\boldsymbol{F}$

For P3ONMF and CP3ONMF, if we treat the $\boldsymbol{AS}'$ as the right-hand factor matrix in two-factor ONMF, the form of the objective function with respect to $\boldsymbol{F}$ is the same as (6.35). Moreover, from the discussion in section 3.2, we can divide the optimization problem of $\boldsymbol{F}$ into that of $\mathcal{R}$ and f_{im} ($i \in R_m$; $m = 1, \dots, k$). Hence, we can obtain their update rule in the same form as (6.36) and (6.38) as follows:

$$\hat{f}_{im} = \begin{cases} \dfrac{\sum_{j=1}^{p} \hat{z}_{ij}^{*} y_{ij} [\boldsymbol{SA}']_{mj}^{\beta-1}}{\sum_{j=1}^{p} \hat{z}_{ij}^{*} [\boldsymbol{SA}']_{mj}^{\beta}} & \text{if } i \in R_m \\ 0 & \text{if } i \notin R_m \end{cases} \quad (i = 1, \dots, n;\ m = 1, \dots, k), \tag{7.41}$$

$$\hat{R}_m = \left\{ i \,\middle|\, \underset{r}{\operatorname{argmin}} \left\{ \frac{\left\{\sum_{j=1}^{p} \hat{z}_{ij}^{*} y_{ij} [\boldsymbol{SA}']_{rj}^{\beta-1}\right\}^{\beta}}{\left\{\sum_{j=1}^{p} \hat{z}_{ij}^{*} [\boldsymbol{SA}']_{rj}^{\beta}\right\}^{\beta-1}} \right\} = m \right\} \ (m = 1, \dots, k). \tag{7.42}$$

Update rule for $\boldsymbol{S}$

Considering $\boldsymbol{A}$ in (6.37) as $\boldsymbol{AS}'$, the objective function with respect to $\boldsymbol{S}$ can be written as follows:

$$\begin{aligned} Q_{\boldsymbol{S}}(\boldsymbol{S}) &= -\frac{1}{\beta(\beta-1)} \sum_{m=1}^{k} \sum_{i \in R_m} \frac{\left\{ \sum_{j=1}^{p} \hat{z}_{ij}^{*} y_{ij} (\sum_{q=1}^{\ell} s_{mq} a_{jq})^{\beta-1} \right\}^{\beta}}{\left\{ \sum_{j=1}^{p} \hat{z}_{ij}^{*} (\sum_{q=1}^{\ell} s_{mq} a_{jq})^{\beta} \right\}^{\beta-1}} \\ &= -\frac{1}{\beta(\beta-1)} \sum_{m=1}^{k} \sum_{i \in R_m} \frac{\left\{ \sum_{q=1}^{\ell} \sum_{j \in C_q} \hat{z}_{ij}^{*} y_{ij} s_{mq}^{\beta-1} a_{jq}^{\beta-1} \right\}^{\beta}}{\left\{ \sum_{q=1}^{\ell} \sum_{j \in C_q} \hat{z}_{ij}^{*} s_{mq}^{\beta} a_{jq}^{\beta} \right\}^{\beta-1}}. \end{aligned} \tag{7.43}$$

Here, we use (3.8). It is difficult to directly obtain the minimizer of s_{mq} for (7.43) because the summation of s_{mq} occurs in the two power functions. However, we can obtain the optimal s_{mq} using the auxiliary function method in a manner similar to ZICP2ONMF. We find that (7.43) contains the function (6.27). Hence, the inequality (6.28) enables us to obtain the following auxiliary function of (7.43):

$$\begin{aligned} &Q_{\boldsymbol{S}}^{\text{aux}}(\boldsymbol{S}, \boldsymbol{S}^{*}) \\ &= \sum_{m=1}^{k} \sum_{i \in R_m} \left\{ -\frac{1}{\beta(\beta-1)} \frac{\eta_{im}^{\beta}}{\lambda_{im}^{\beta-1}} + \frac{1}{\beta} \left(\frac{\eta_{im}}{\lambda_{im}} \right)^{\beta} \left(\sum_{q=1}^{\ell} \sum_{j \in C_q} \hat{z}_{ij}^{*} s_{mq}^{\beta} a_{jq}^{\beta} - \lambda_{im} \right) \right. \\ &\left. + \frac{1}{1-\beta} \left(\frac{\eta_{im}}{\lambda_{im}} \right)^{\beta-1} \left(\sum_{q=1}^{\ell} \sum_{j \in C_q} \hat{z}_{ij}^{*} y_{ij} s_{mq}^{\beta-1} a_{jq}^{\beta-1} - \eta_{im} \right) \right\}, \end{aligned} \tag{7.44}$$

where

$$\lambda_{im} = \sum_{q=1}^{\ell} \sum_{j \in C_q} \hat{z}_{ij}^* s_{mq}^{*\beta} a_{jq}^{\beta} \ (i = 1, \dots n;\ m = 1, \dots, k) \tag{7.45}$$

$$\text{and } \eta_{im} = \sum_{q=1}^{\ell} \sum_{j \in C_q} \hat{z}_{ij}^* y_{ij} s_{mq}^{*\beta-1} a_{jq}^{\beta-1} \ (i = 1, \dots n;\ m = 1, \dots, k). \tag{7.46}$$

We obtain the following update equation of s_{mq} as a minimizer for $Q_{\boldsymbol{S}}^{\mathrm{aux}}(\boldsymbol{S}, \boldsymbol{S}^*)$:

$$s_{mq} = \frac{\sum_{i \in R_m} \sum_{j \in C_q} (\eta_{im}/\lambda_{im})^{\beta-1} \hat{z}_{ij}^* y_{ij} a_{jq}^{\beta-1}}{\sum_{i \in R_m} \sum_{j \in C_q} (\eta_{im}/\lambda_{im})^{\beta} \hat{z}_{ij}^* a_{jq}^{\beta}}. \tag{7.47}$$

Update rule for $\boldsymbol{A}$

If we regard the model as $\boldsymbol{Y}' \approx \boldsymbol{A}\boldsymbol{S}'\boldsymbol{F}'$, the update rules of $\mathcal{C}$, $\boldsymbol{S}$, and $\boldsymbol{A}$ can be derived similarly to (7.41), (7.42), and (7.47) as follows:

$$a_{jq} = \begin{cases} \dfrac{\sum_{i=1}^{n} \hat{z}_{ij}^* y_{ij} [\boldsymbol{FS}]_{iq}^{\beta-1}}{\sum_{i=1}^{n} \hat{z}_{ij}^* [\boldsymbol{FS}]_{iq}^{\beta}} & (j \in C_q) \\ 0 & (j \notin C_q) \end{cases} \ (j = 1, \dots p;\ q = 1, \dots, \ell), \tag{7.48}$$

$$C_q = \left\{ j \,\middle|\, \underset{c}{\operatorname{argmin}} \left\{ \frac{\left\{ \sum_{i=1}^{n} \hat{z}_{ij}^* y_{ij} [\boldsymbol{FS}]_{ic}^{\beta-1} \right\}^{\beta}}{\left\{ \sum_{i=1}^{n} \hat{z}_{ij}^* [\boldsymbol{FS}]_{ic}^{\beta} \right\}^{\beta-1}} \right\} = q \right\} \ (q = 1, \dots, \ell), \tag{7.49}$$

$$s_{mq} = \frac{\sum_{i \in R_m} \sum_{j \in C_q} (\eta_{jq}^*/\lambda_{jq}^*)^{\beta-1} \hat{z}_{ij}^* y_{ij} f_{im}^{\beta-1}}{\sum_{i \in R_m} \sum_{j \in C_q} (\eta_{jq}^*/\lambda_{jq}^*)^{\beta} \hat{z}_{ij}^* f_{im}^{\beta}} \ (m = 1, \dots, k;\ q = 1, \dots, \ell), \tag{7.50}$$

where

$$\lambda_{jq}^* = \sum_{m=1}^{k} \sum_{i \in R_m} \hat{z}_{ij}^* s_{mq}^{*\beta} f_{im}^{\beta} \ (j = 1, \dots, p;\ q = 1, \dots, \ell), \tag{7.51}$$

$$\eta_{jq}^* = \sum_{m=1}^{k} \sum_{i \in R_m} \hat{z}_{ij}^* y_{ij} s_{mq}^{*\beta-1} f_{im}^{\beta-1} \ (j = 1, \dots p;\ q = 1, \dots, \ell). \tag{7.52}$$

Algorithm

The ZICP3ONMF algorithm, which is derived from (3.52), (3.55), (7.41), (7.42), (7.47), (7.48), (7.49), (7.50), and the discussion in Section 3.3 about optimal ϕ, is presented in Algorithm 18.

Algorithm 18 ZICP3ONMF Algorithm

```
Input $\boldsymbol{Y} \in \mathbb{R}_+^{n\times p}$, $\beta \in (0,1)$, $k \in \mathbb{N}$, $\ell \in \mathbb{N}$, $\boldsymbol{F}^{(0)} \in \mathbb{R}_+^{n\times k}$ ($\boldsymbol{f}_{(m)}^{(0)\prime}\boldsymbol{f}_{(r)}^{(0)} = 0$ $(m \neq r)$), $\boldsymbol{S}^{(0)} \in \mathbb{R}_+^{k\times \ell}$, $\boldsymbol{A}^{(0)} \in \mathbb{R}_+^{p\times \ell}$ ($\boldsymbol{a}_{(q)}^{(0)\prime}\boldsymbol{a}_{(c)}^{(0)} = 0$ $(q \neq c)$), $\mathcal{R}^{(0)}$, $\mathcal{C}^{(0)}$, $w^{(0)} \in (0,1)$, $\phi^{(0)} > 0$, $\tau > 0$, $\upsilon \in \mathbb{N}$, $\delta \in \mathbb{N}$, and $\kappa \in \mathbb{N}$
$t \leftarrow 0$
$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
$z_{ij}^{(t)} \leftarrow \begin{cases} \dfrac{w^{(t)}}{w^{(t)} + (1-w^{(t)})h(0,\phi^{(t)},\beta)\exp\{-(x_{ij}^{(t)})^{\beta}/(\phi^{(t)}\beta)\}} & \text{if } y_{ij} = 0 \\ 0 & \text{if } y_{ij} \neq 0 \end{cases}$
$(i = 1,\dots,n;\ j = 1,\dots,p)$
$\boldsymbol{Z}^{*(t)} \leftarrow \boldsymbol{E}_{n\times p} - \boldsymbol{Z}^{(t)}$
$L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log\left\{w^{(t)}I(y_{ij}=0) + (1-w^{(t)})f_{\mathrm{CP}}(y_{ij}|x_{ij}^{(t)},\phi^{(t)},\beta)\right\}$
repeat
    $t \leftarrow t+1$
    $w^{(t)} \leftarrow \dfrac{\sum_{i=1}^{n}\sum_{j=1}^{p} z_{ij}^{(t-1)}}{np}$
    $\lambda_{im}^{(t)} \leftarrow \sum_{q=1}^{\ell}\sum_{j\in C_q^{(t-1)}} z_{ij}^{*(t-1)}(s_{mq}^{(t-1)})^{\beta}(a_{jq}^{(t-1)})^{\beta}$ $(i = 1,\dots n;\ m = 1,\dots,k)$
    $\eta_{im}^{(t)} \leftarrow \sum_{q=1}^{\ell}\sum_{j\in C_q^{(t-1)}} z_{ij}^{*(t-1)}y_{ij}(s_{mq}^{(t-1)})^{\beta-1}(a_{jq}^{(t-1)})^{\beta-1}$ $(i = 1,\dots n;\ m = 1,\dots,k)$
    $R_m^{(t)} \leftarrow \left\{i \,\middle|\, \operatorname*{argmin}_r \left\{\dfrac{(\eta_{ir}^{(t)})^{\beta}}{(\lambda_{ir}^{(t)})^{\beta-1}}\right\} = m\right\}$ $(m = 1,\dots,k)$
    $s_{mq}^{*(t)} \leftarrow \dfrac{\sum_{i\in R_m^{(t)}}\sum_{j\in C_q^{(t-1)}}(\eta_{im}^{(t)}/\lambda_{im}^{(t)})^{\beta-1}z_{ij}^{*(t-1)}y_{ij}(a_{jq}^{(t-1)})^{\beta-1}}{\sum_{i\in R_m^{(t)}}\sum_{j\in C_q^{(t-1)}}(\eta_{im}^{(t)}/\lambda_{im}^{(t)})^{\beta}z_{ij}^{*(t-1)}(a_{jq}^{(t-1)})^{\beta}}$
    $(m = 1,\dots,k;\ q = 1,\dots,\ell)$
    $f_{im}^{(t)} \leftarrow \begin{cases} \dfrac{\sum_{j=1}^{p} z_{ij}^{*(t-1)}y_{ij}[\boldsymbol{S}^{*(t)}\boldsymbol{A}^{(t-1)\prime}]_{mj}^{\beta-1}}{\sum_{j=1}^{p} z_{ij}^{*(t-1)}[\boldsymbol{S}^{*(t)}\boldsymbol{A}^{(t-1)\prime}]_{mj}^{\beta}} & \text{if } i \in R_m^{(t)} \\ 0 & \text{if } i \notin R_m^{(t)} \end{cases}$ $(i = 1,\dots,n;\ m = 1,\dots,k)$
    $\lambda_{jq}^{*(t)} \leftarrow \sum_{m=1}^{k}\sum_{i\in R_m^{(t)}} z_{ij}^{*(t-1)}(s_{mq}^{*(t)})^{\beta}(f_{im}^{(t)})^{\beta}$ $(j = 1,\dots,p;\ q = 1,\dots,\ell)$
    $\eta_{jq}^{*(t)} \leftarrow \sum_{m=1}^{k}\sum_{i\in R_m^{(t)}} z_{ij}^{*(t-1)}y_{ij}(s_{mq}^{*(t)})^{\beta-1}(f_{im}^{(t)})^{\beta-1}$ $(j = 1,\dots p;\ q = 1,\dots,\ell)$
    $C_q^{(t)} \leftarrow \left\{j \,\middle|\, \operatorname*{argmin}_c \left\{\dfrac{(\eta_{jc}^{*(t)})^{\beta}}{(\lambda_{jc}^{*(t)})^{\beta-1}}\right\} = q\right\}$ $(q = 1,\dots,\ell)$
    $s_{mq}^{(t)} \leftarrow \dfrac{\sum_{i\in R_m^{(t)}}\sum_{j\in C_q^{(t)}}(\eta_{jq}^{*(t)}/\lambda_{jq}^{*(t)})^{\beta-1}z_{ij}^{*(t-1)}y_{ij}(f_{im}^{(t)})^{\beta-1}}{\sum_{i\in R_m^{(t)}}\sum_{j\in C_q^{(t)}}(\eta_{jq}^{*(t)}/\lambda_{jq}^{*(t)})^{\beta}z_{ij}^{*(t-1)}(f_{im}^{(t)})^{\beta}}$
    $(m = 1,\dots,k;\ q = 1,\dots,\ell)$
    $a_{jq} \leftarrow \begin{cases} \dfrac{\sum_{i=1}^{n} z_{ij}^{*(t-1)}y_{ij}[\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}]_{iq}^{\beta-1}}{\sum_{i=1}^{n} z_{ij}^{*(t-1)}[\boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}]_{iq}^{\beta}} & (j \in C_q^{(t)}) \\ 0 & (j \notin C_q^{(t)}) \end{cases}$ $(j = 1,\dots p;\ q = 1,\dots,\ell)$
```

Algorithm 18 ZICP3ONMF Algorithm (continued)

```
    if $t \leq \delta$ or $t \bmod \kappa = 0$ then
        $\phi^{(t)}$ is obtained as the optimal $\phi$ that optimizes $Q_\phi(\phi)$ given $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$,
        $\boldsymbol{A}^{(t)}$, $\boldsymbol{Z}^{(t-1)}$, $w^{(t)}$, and $\beta$ using the BFGS quasi-Newton method with
        constraints $\phi > 0$
    end if
    $\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{F}^{(t)}\boldsymbol{S}^{(t)}\boldsymbol{A}^{(t)\prime}$
    $z_{ij}^{(t)} \leftarrow \begin{cases} \dfrac{w^{(t)}}{w^{(t)} + (1-w^{(t)})h(0,\phi^{(t)},\beta)\exp\{-(x_{ij}^{(t)})^\beta/(\phi^{(t)}\beta)\}} & \text{if } y_{ij} = 0 \\ 0 & \text{if } y_{ij} \neq 0. \end{cases}$
    $(i = 1,\ldots,n;\ j = 1,\ldots,p)$
    $\boldsymbol{Z}^{*(t)} \leftarrow \boldsymbol{E}_{n\times p} - \boldsymbol{Z}^{(t)}$
    $L^{(t)} \leftarrow \sum_{i=1}^{n}\sum_{j=1}^{p} \log\left\{w^{(t)}I(y_{ij}=0) + (1-w^{(t)})f_{\mathrm{CP}}(y_{ij}|x_{ij}^{(t)},\phi^{(t)},\beta)\right\}$
until $L^{(t)} - L^{(t-1)} < \tau$ or $t = \upsilon$
Output $\boldsymbol{F}^{(t)}$, $\boldsymbol{S}^{(t)}$, $\boldsymbol{A}^{(t)}$, $\mathcal{R}^{(t)}$, $\mathcal{C}^{(t)}$, $\boldsymbol{Z}^{(t)}$, $w^{(t)}$, and $\phi^{(t)}$
```

Chapter 8

Numerical studies

In this chapter, we present selected simulation studies. Section 8.1 concerns three-factor orthogonal NMF; we show the effectiveness of N3ONMF in terms of its estimation accuracy and the robustness of CP3ONMF. In Section 8.2, we present a worse approximation to the given data by orthogonal NMFs; the poor performance is trade-off against the simple structure of the factor matrix. In Section 8.3, we demonstrate a accurate approximation of NMF based on a zero-inflated model for a zero-inflated data matrix.

8.1 Accuracy of estimates of three-factor orthogonal NMF

In this section, we describe two simulation studies relating to three-factor orthogonal NMF. The first study compares N3ONMF with previous three-factor orthogonal NMFs, Ding et al. (2006) and Yoo and Choi (2010b), in terms of estimation accuracy. The N3ONMF is our proposed method, and it forms a foundation for the other three-factor orthogonal NMFs. Therefore, a comparison with previous three-factor orthogonal NMFs is needed. The second study analyzes the characteristics of the estimates given by N3ONMF, P3ONMF, and CP3ONMF. An NMF based on a non-normal distribution, that is, the Poisson or CP distribution, can be more robust to outliers than an NMF based on a normal distribution. In this simulation study, we demonstrate its robustness in terms of three-factor orthogonal NMF. Although we could use the other NMF methods for checking the robustness, for example, two-factor non-orthogonal or orthogonal NMF or three-factor non-orthogonal NMF, we choose three-factor orthogonal NMF because all these methods are proposed by us and because of space limitations.

8.1.1 Estimation accuracy of N3ONMF

In this study, we conduct a simulation study to examine the estimation accuracy of N3ONMF. Additionally, we compare N3ONMF with previous three-factor orthogonal NMF techniques proposed by Ding et al. (2006) and Yoo and Choi (2010b). In this section, we refer to them as Ding et al's method and Yoo and Choi's method, respectively. Ding et al.'s method and Yoo and Choi's method have some estimation difficulties, and it is expected that N3ONMF, which uses the same model as they do but has a different algorithm, will perform better. In this simulation, we use synthetic data with a clear model structure, and this data may be far from the real world data. However, this simulation enables us to understand the advantages of N3ONMF.

First, we generate synthetic data using true $\tilde{\boldsymbol{F}}$, $\tilde{\boldsymbol{S}}$, $\tilde{\boldsymbol{A}}$, $\tilde{\mathcal{R}} = \{\tilde{R}_1, \ldots, \tilde{R}_k\}$, and $\tilde{\mathcal{C}} = \{\tilde{C}_1, \ldots, \tilde{C}_\ell\}$ as described later; second, we apply these three three-factor orthogonal NMFs to the synthetic data and obtain the estimated $\hat{\boldsymbol{F}}$, $\hat{\boldsymbol{S}}$, $\hat{\boldsymbol{A}}$, $\hat{\mathcal{R}} = \{\hat{R}_1, \ldots, \hat{R}_k\}$, and $\hat{\mathcal{C}} = \{\hat{C}_1, \ldots, \hat{C}_\ell\}$ for each of the methods. Finally, we measure the closeness between the true and estimated parameters by ARI$(\tilde{\mathcal{R}}, \hat{\mathcal{R}})$, ARI$(\tilde{\mathcal{C}}, \hat{\mathcal{C}})$, $\|\tilde{\boldsymbol{F}} - \hat{\boldsymbol{F}}\|/(nk)$, $\|\tilde{\boldsymbol{A}} - \hat{\boldsymbol{A}}\|/(p\ell)$, and $\|\tilde{\boldsymbol{S}} - \hat{\boldsymbol{S}}\|/(k\ell)$. Here, ARI$(\cdot, \cdot)$ is the adjusted Rand index (ARI) (Hubert and Arabie, 1985), which is a similarity measure between two partitions of objects. If the partitions are completely the same, ARI is 1; if they are different, ARI is close to 0.

The synthetic data is generated as follows. First, we determine the true row clusters $\tilde{\mathcal{R}}$ randomly; we then generate $\tilde{\boldsymbol{F}}^* \in \mathbb{R}^{n \times k}$ as follows:

$$\begin{cases} \tilde{f}^*_{im} \sim Ex(\mu^{1/3}) & (i \in \tilde{R}_m) \\ \tilde{f}^*_{im} = 0 & (i \notin \tilde{R}_m) \end{cases}$$
$$(i = 1, \ldots, n;\ m = 1, \ldots, k), \tag{8.1}$$

where $Ex(x)$ is an exponential distribution with an expected value x and μ represents the mean value of each element of the synthetic data matrix. The value of μ is determined in advance of the simulation. We use exponential random variables because some researchers, e.g., Schmidt et al. (2009) and Tan and Févotte (2013), set the exponential distribution as a prior of the factor matrix elements. Next, the norm of each column of $\tilde{\boldsymbol{F}}^*$ is converted to 1 as follows:

$$\tilde{\boldsymbol{F}} = \tilde{\boldsymbol{F}}^* \boldsymbol{D}^{-1}_{\tilde{\boldsymbol{F}}^*}. \tag{8.2}$$

The true $\tilde{\mathcal{C}}$ and $\tilde{\boldsymbol{A}} \in \mathbb{R}^{p \times \ell}$ are generated in the same manner as $\tilde{\mathcal{R}}$ and $\tilde{\boldsymbol{F}}$. We set $\tilde{\mathcal{C}}$ randomly and generate $\tilde{\boldsymbol{A}}^* \in \mathbb{R}^{p \times \ell}$ as follows:

$$\begin{cases} \tilde{a}^*_{jq} \sim Ex(\mu^{1/3}) & (j \in \tilde{C}_q) \\ \tilde{a}^*_{jq} = 0 & (j \notin \tilde{C}_q) \end{cases}$$
$$(j = 1, \ldots, p;\ q = 1, \ldots, \ell). \tag{8.3}$$

The norm of each column of $\tilde{\boldsymbol{A}}^*$ is then converted to 1 as follows:

$$\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{A}}^* \boldsymbol{D}^{-1}_{\tilde{\boldsymbol{A}}^*}. \tag{8.4}$$

Each element of true $\tilde{\boldsymbol{S}}$ is generated as follows: First, we generate $\tilde{\boldsymbol{S}}^*$ from

$$\tilde{s}^*_{mq} \sim Ex(\mu^{1/3}) \quad (m = 1, \ldots, k;\ q = 1, \ldots, \ell), \tag{8.5}$$

then we calculate

$$\tilde{\boldsymbol{S}} = \boldsymbol{D}_{\tilde{\boldsymbol{F}}^*} \tilde{\boldsymbol{S}}^* \boldsymbol{D}_{\tilde{\boldsymbol{A}}^*}. \tag{8.6}$$

Finally, we obtain the synthetic data matrix by a normal distribution such that

$$y_{ij} \sim N(\tilde{x}_{ij}, \sigma) \; (i = 1, \dots, n; \; j = 1, \dots, p), \tag{8.7}$$

where

$$\begin{aligned} \tilde{x}_{ij} &= \sum_{i=1}^{n} \sum_{j=1}^{p} \tilde{f}_{im} \tilde{s}_{mq} \tilde{a}_{jq} && (8.8) \\ &= \tilde{f}_{im} \tilde{s}_{mq} \tilde{a}_{jq} \quad (i = 1, \dots, n; \; j = 1, \dots, p; \; \tilde{R}_m \ni i; \; \tilde{C}_q \ni j). && (8.9) \end{aligned}$$

From the above, the expected value of each elements of synthetic data is as follows:

$$\begin{aligned} &E[y_{ij}] = E[\tilde{x}_{ij}] = E[\tilde{f}_{im} \tilde{s}_{mq} \tilde{a}_{jq}] = E[\tilde{f}_{im}] E[\tilde{s}_{mq}] E[\tilde{a}_{jq}] = (\mu^{1/3})^3 = \mu \\ &(i = 1, \dots, n; \; j = 1, \dots, p; \; \tilde{R}_m \ni i; \; \tilde{C}_q \ni j). \end{aligned} \tag{8.10}$$

From this, μ undoubtedly represents the expected value of y_{ij}. If $y_{ij} < 0$, then the element is converted to zero.

The parameters for generating the synthetic data are set as follows:

- $(n,p,k,\ell) = \{(100,60,5,3),(100,100,5,5),(1000,600,5,3),(1000,1000,5,5)\}$
- $\sigma = \{1, 2, 4\}$
- $\mu = 10$
- $\nu = 1000$ (maximum number of iterative cycles)

It is noteworthy that the true numbers of row and column clusters, and the estimated ones, are the same as k and ℓ, respectively.

We generate 100 synthetic data matrices for each $4 \times 3 = 12$ conditions; from among the candidates of the estimates given by 20 executions of each of the methods, we select the estimates for which the objective function value is minimized. The convergence is determined using (3.11) and the convergence threshold is set as $\tau = 10^{-7}np$ for all the methods. The results are shown in Fig. 8.1 through Fig. 8.5.

Fig. 8.1 shows the boxplots of ARI($\tilde{\mathcal{R}}, \hat{\mathcal{R}}$) obtained by the three methods for the 12 situations. Note that each ARI decreases as the variance σ increases in every situation and increase as the matrix become larger. In all situations, N3ONMF is the highest, followed by Yoo and Choi's method and Ding et al.'s method in that order. For rectangular matrices such as those for which $(n, p) = (100, 60)$ or $(n, p) = (1000, 600)$, both the methods of Ding et al. and Yoo and Choi obtain small values. These results indicate that N3ONMF appears to perform more accurately than the methods of Ding et al. and Yoo and Choi in terms of row cluster detection. Fig. 8.2 shows the boxplots of ARI($\tilde{\mathcal{C}}, \hat{\mathcal{C}}$). The results are similar to those in Fig. 8.1 for all square matrix situations. For rectangular data matrices, Yoo and Choi's method obtains larger ARI values for column clusters than for row clusters in Fig. 8.1. This result suggests that Yoo and Choi's method can accurately detect the smaller side clusters (in our simulation, this is the column side). However, in all situations, N3ONMF provides the most optimal clustering accuracy. Fig. 8.3 and Fig.

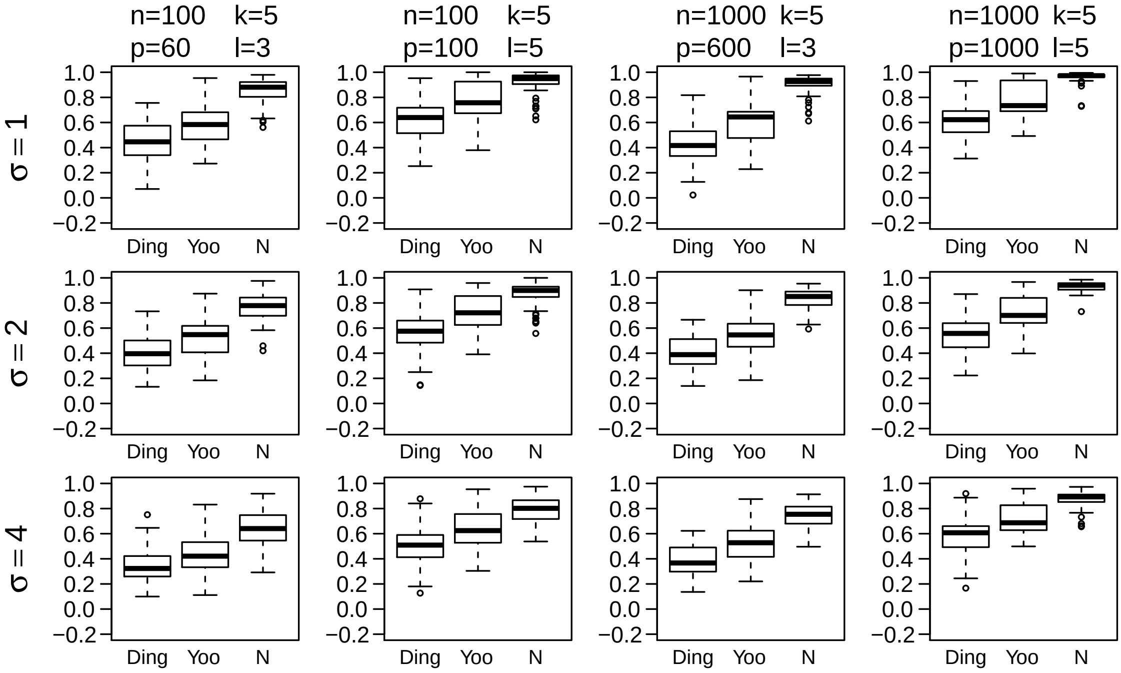


Figure 8.1: Boxplots of ARI($\tilde{\mathcal{R}}, \hat{\mathcal{R}}$) obtained by three three-factor orthogonal NMFs for 12 conditions. The "N" below each of the boxplots indicates N3ONMF.

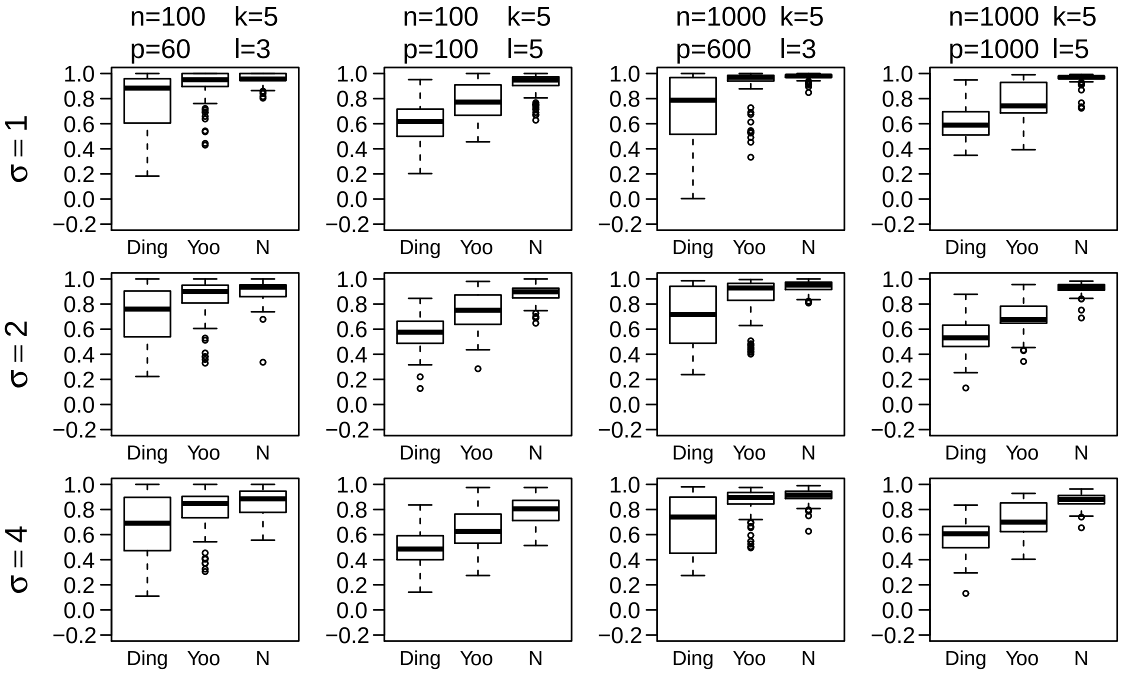


Figure 8.2: Boxplots of ARI($\tilde{\mathcal{C}}, \hat{\mathcal{C}}$) obtained by three three-factor orthogonal NMFs for 12 conditions. The "N" below each of the boxplots indicates N3ONMF.

8.4 show the boxplots of $\|\tilde{\boldsymbol{F}} - \hat{\boldsymbol{F}}\|/(nk)$ and $\|\tilde{\boldsymbol{A}} - \hat{\boldsymbol{A}}\|/(p\ell)$, respectively. The ranges of the vertical axes are not same because the magnitude of $\hat{f}_{im}$ and $\hat{a}_{jq}$ differs depending on n and p. Both results in Fig. 8.3 and Fig. 8.4 are similar to the ARI results in Fig. 8.1 and

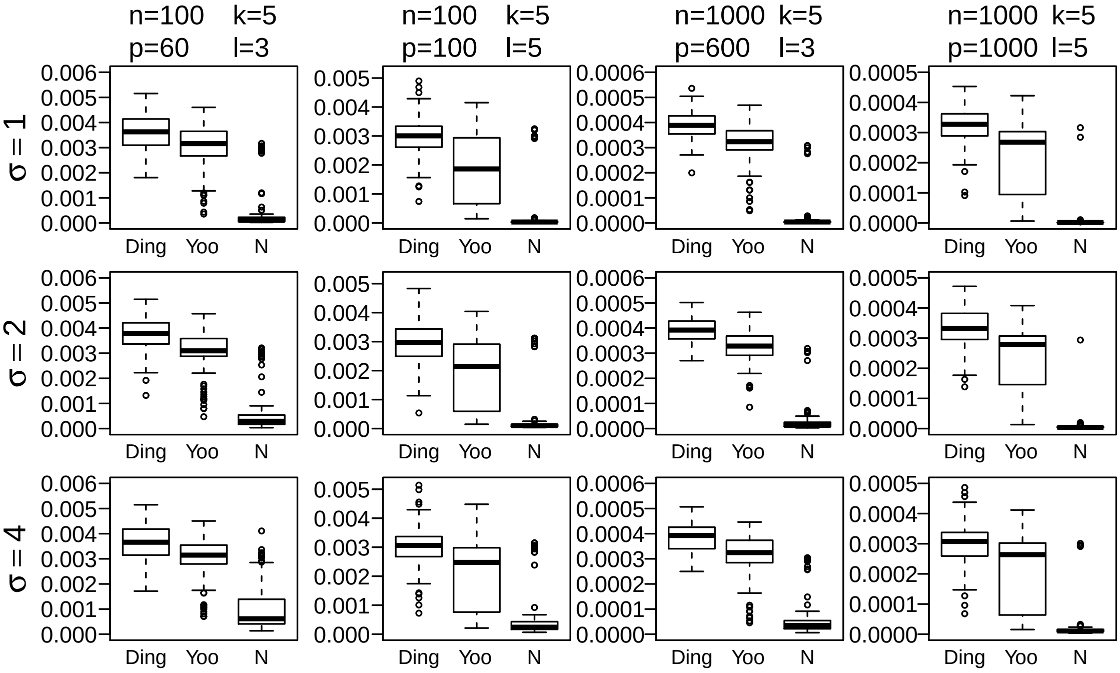


Figure 8.3: Boxplots of $\|\tilde{\boldsymbol{F}} - \hat{\boldsymbol{F}}\|/(nk)$ obtained by three three-factor orthogonal NMFs for 12 conditions. The "N" below each of the boxplots indicates N3ONMF.

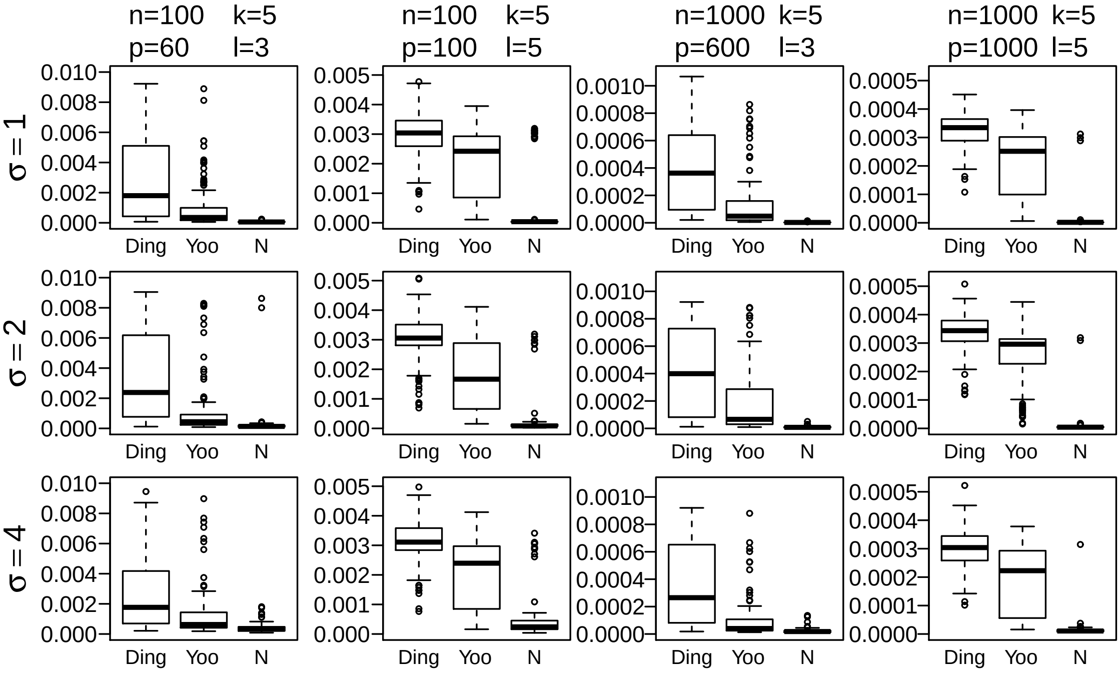


Figure 8.4: Boxplots of $\|\tilde{\boldsymbol{A}} - \hat{\boldsymbol{A}}\|/(p\ell)$ obtained by three three-factor orthogonal NMFs for 12 conditions. The "N" below each of the boxplots indicates N3ONMF.

Fig. 8.2. However, the variance of $\|\tilde{\boldsymbol{F}} - \hat{\boldsymbol{F}}\|/(nk)$ generated by N3ONMF is large when $\sigma = 4$ and $(n, p) = (100, 60)$. This reflects the fact that when any misclassifications of $\mathcal{R}$ occur, all errors of f_{im} are significant in N3ONMF, owing to the perfect orthogonality of

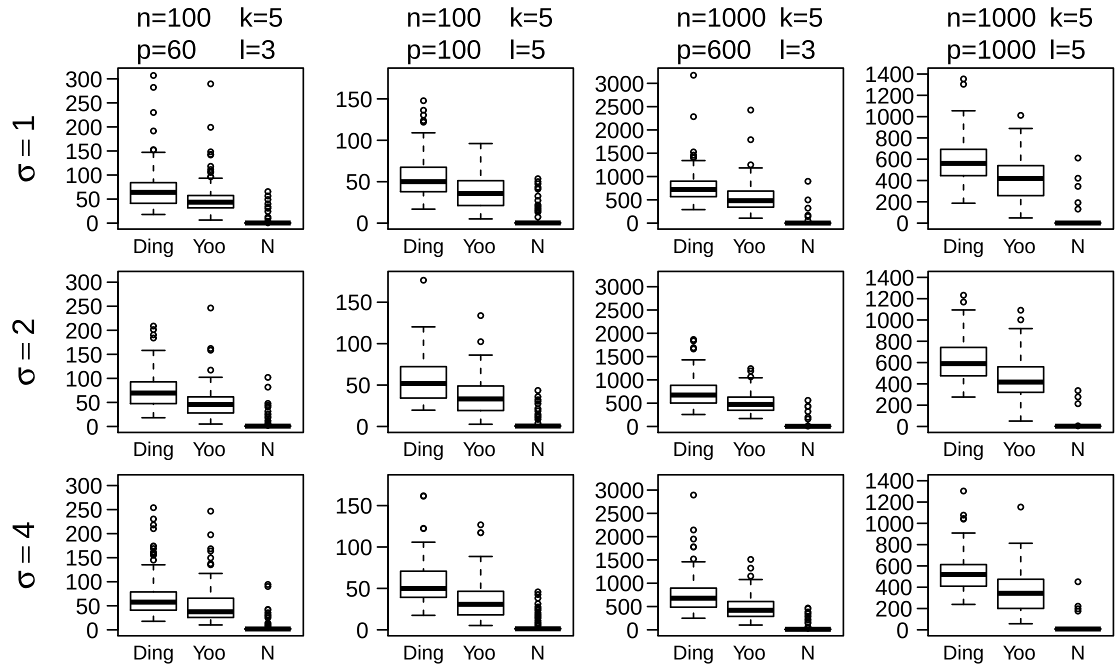


Figure 8.5: Boxplots of $\|\tilde{\boldsymbol{S}} - \hat{\boldsymbol{S}}\|/(k\ell)$ obtained by three three-factor orthogonal NMFs for 12 conditions. The "N" below each of the boxplots indicates N3ONMF.

$\boldsymbol{F}$ such as that represented by (3.7). This can be considered a drawback of N3ONMF. Fig. 8.5 shows the boxplots of $\|\tilde{\boldsymbol{S}} - \hat{\boldsymbol{S}}\|^2$. In all situations, the values obtained by N3ONMF are smaller than those obtained by the other two methods. This simulation study enables us to conclude that N3ONMF provides more effective estimation consistency than the other two methods.

8.1.2 Robustness of CP3ONMF

We conducted another simulation study to demonstrate the characteristics of the estimates provided by N3ONMF, P3ONMF, and CP3ONMF. As mentioned in Chapter 7, it is assumed that y_{ij} follows normal, Poisson, and compound Poisson distributions, respectively, in these three methods. These distributions belong to the Tweedie family, which is described by (3.15), and the value of β determines the distribution: it is normal if $\beta = 2$, Poisson if $\beta = 1$, and compound Poisson if $\beta \in (0, 1)$. The index parameter β is related to the robustness of parameter estimation as described in Section 3.3. We examined these characteristics in three-factor orthogonal NMF by measuring the estimation accuracy of N3ONMF, P3ONMF, and CP3ONMF for synthetic data matrices generated using normal, Poisson, and compound Poisson distributions of data. The accuracy was calculated using the ARI between true clusters and estimated clusters of row and column objects.

We now explain how to generate a synthetic data matrix. First, we generate $\tilde{\mathcal{R}}$, $\tilde{\boldsymbol{F}}^*$, $\tilde{\mathcal{C}}$, $\tilde{\boldsymbol{A}}^*$, and $\tilde{\boldsymbol{S}}^*$ as in Section 8.1.1. Then, we generate each element of the synthetic data matrix $\boldsymbol{Y}$ as a random number from $y_{ij} \sim TW(x_{ij}, \phi, \tilde{\beta})$, where the mean x_{ij} is the corresponding element of $\boldsymbol{X} = \tilde{\boldsymbol{F}}^* \tilde{\boldsymbol{S}}^* \tilde{\boldsymbol{A}}^{*\prime}$. It is noted that $TW(x_{ij}, \phi, \tilde{\beta})$ is normal if $\tilde{\beta} = 2$

and Poisson if $\tilde{\beta} = 1$. When $\tilde{\beta} = 2$, negative values of y_{ij} can be generated, in which case, y_{ij} is converted to zero.

The parameters for generating synthetic data are set as follows:

- $(n, p, k, \ell) = (100, 100, 5, 5)$
- $\phi = 2$
- $\tilde{\beta} = \{2, 1, 0.8, 0.5, 0.2\}$
- $\mu = 10$
- $\nu = 1000$

Fig. 8.6 shows plots of the probability density functions of the Tweedie distribution for $\beta = \{2, 1, 0.8, 0.5, 0.2\}$, where $\mu = 10$ and $\phi = 2$.

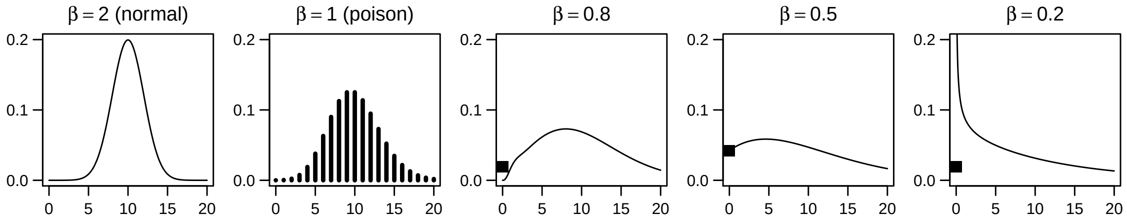


Figure 8.6: Probability density functions of the Tweedie distribution for $\beta = \{2, 1, 0.8, 0.5, 0.2\}$. The black square represents the probability at $y = 0$.

Here it should be noted that the true numbers of row and column clusters, and the estimated ones, are the same as k and ℓ, respectively. We generate 100 synthetic data matrices for each of five conditions. Then, from among the candidate estimations given by 20 executions, we select the estimates, $\hat{\mathcal{R}}$ and $\hat{\mathcal{C}}$, for which the objective function value is minimized. We then calculate ARI$(\tilde{\mathcal{R}}, \hat{\mathcal{R}})$ and ARI$(\tilde{\mathcal{C}}, \hat{\mathcal{C}})$ of each of the methods. The convergence is determined using (3.11) and the convergence threshold is set as $\tau = 10^{-7}np$ for all the methods.

We execute CP3ONMF for three cases: $\beta = \{0.2, 0.5, 0.8\}$ and refer to the procedures as CP3ONMF-2, CP3ONMF-5, and CP3ONMF-8, respectively. The results are shown in Figs. 8.7 and Fig. 8.8.

The two figures appear to be similar. When $\tilde{\beta} = 2$ (normal), N3ONMF is the most accurate, followed by P3ONMF, CP3ONMF-8, CP3ONMF-5, and CP3ONMF-2, in that order. When $\tilde{\beta} = 0.5$, N3ONMF is the least accurate; when $\tilde{\beta} = 0.2$, the accuracy deteriorates in the order of N3ONMF, P3ONMF, and CP3ONMF-8. Because more extreme outliers are generated from a compound Poisson distribution with small $\tilde{\beta}$ values, these results imply that N3ONMF, P3ONMF, and CP3ONMF procedures with relatively larger β values do not fit a data matrix containing some outliers. This does not mean that a CP3ONMF procedure with a small β value is the most accurate under all circumstances. In fact, its performance may be worse for a data matrix with a normal error, as shown in the case of $\tilde{\beta} = 2$ in Fig. 8.7 and Fig. 8.8. However, it may be preferable to use

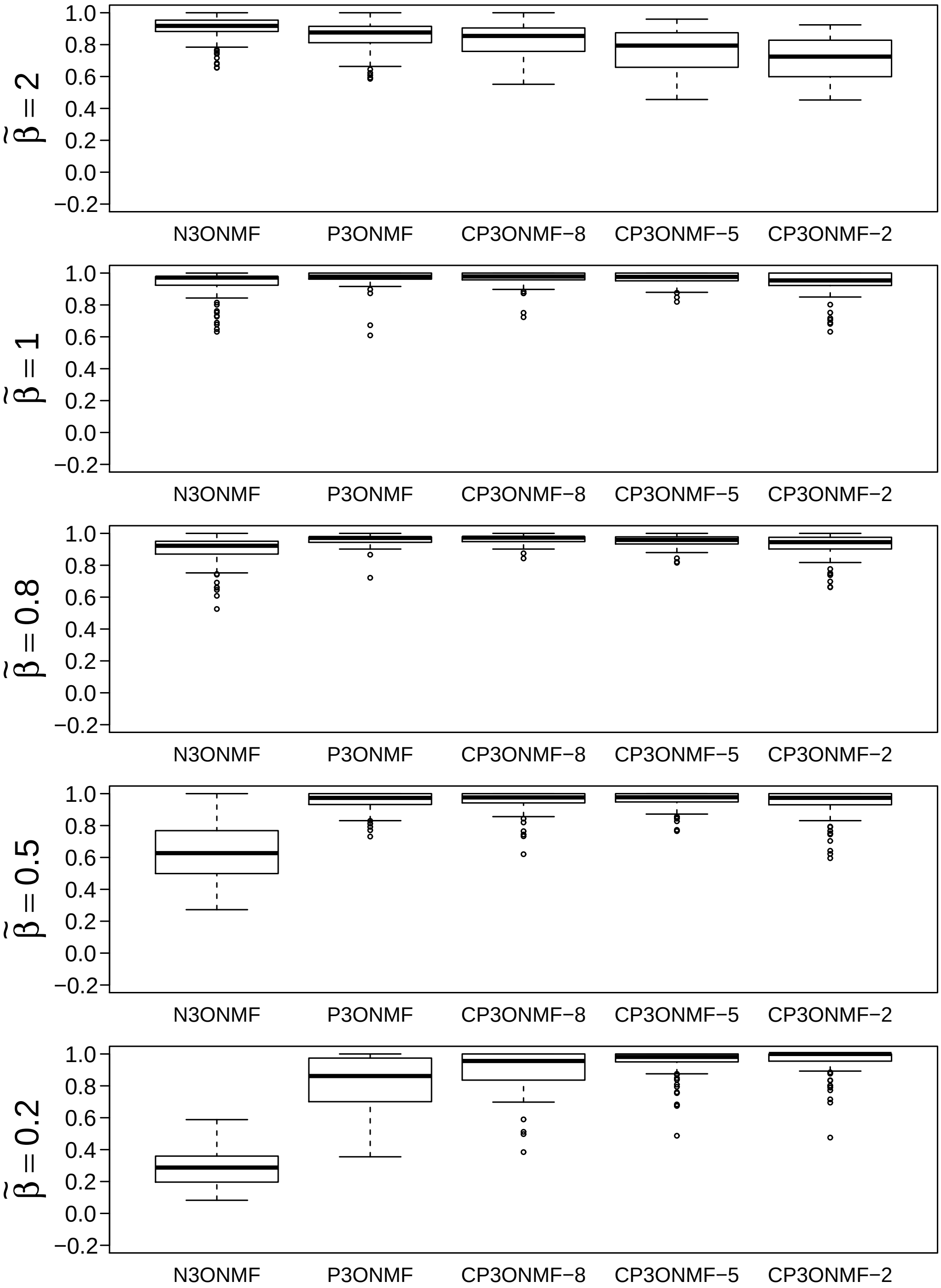


Figure 8.7: ARI($\tilde{\mathcal{R}}, \hat{\mathcal{R}}$) obtained by five three-factor orthogonal NMFs for five conditions

CP3ONMF because for small β values, CP3ONMF is less inaccurate than N3ONMF for a data matrix containing some outliers.

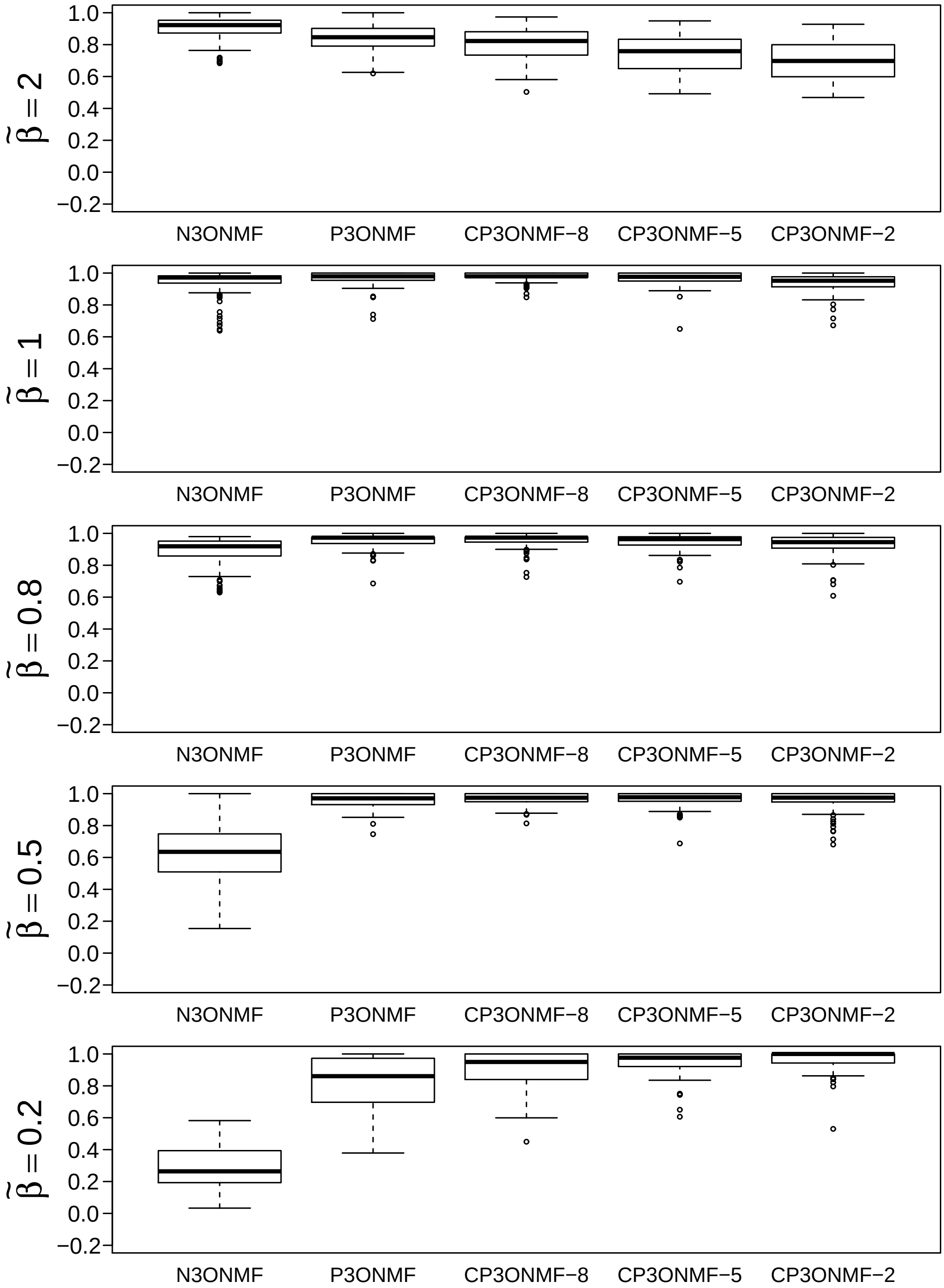


Figure 8.8: ARI($\tilde{\mathcal{C}}, \hat{\mathcal{C}}$) obtained by five three-factor orthogonal NMFs for five conditions

8.2 Approximation of NMF with and without orthogonal constraint

In this section we demonstrate the drawback of an orthogonal constraint by using a numerical example. As mentioned in Section 3.2, an orthogonal constraint simplifies a factor matrix, thereby facilitating interpretation of the result. However, a factor matrix with a simplified structure leads to a poor approximation to the $\boldsymbol{Y}$ by $\boldsymbol{X}$. We demonstrated this drawback by generating a few synthetic nonnegative data matrices and calculated the approximation accuracy of the non-orthogonal NMF and orthogonal NMF. We then compared the approximation accuracy of these two types of NMF for two- and three-factor NMF. Although we can use an NMF based on a normal, Poisson, or CP distribution for this comparison, we only use the CP distribution, which is the focus of this thesis, because of space limitations. Now, we explain how to simulate the two-factor NMF. First, we generate $\tilde{\boldsymbol{F}}$ such that

$$\tilde{f}_{im} \sim Ex\left(\left(\frac{\mu}{k_0}\right)^{1/2}\right) \; (i = 1, \ldots, n;\; m = 1, \ldots, k_0), \tag{8.11}$$

where $Ex(x)$ is an exponential distribution with an expected value x, μ is the expected value of the element of synthetic data $\boldsymbol{Y}$, and k_0 is the number of factors. $\tilde{\boldsymbol{A}} \in \mathbb{R}_+^{p \times k_0}$ is also generated in the same manner as $\tilde{\boldsymbol{F}}$. Then, we obtain a noiseless data matrix $\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{F}}\tilde{\boldsymbol{A}}'$, after which we obtain the synthetic data $\boldsymbol{Y}$ by the following CP distribution:

$$y_{ij} \sim CP(\tilde{x}_{ij}, \phi_0, \beta_0) \; (i = 1, \ldots, n;\; j = 1, \ldots, p). \tag{8.12}$$

Finally, we execute CP2NMF and CP2ONMF to $\boldsymbol{Y}$, obtain the estimated $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{F}}$ by the two methods, and calculate $d_\beta(\boldsymbol{Y}, \tilde{\boldsymbol{F}}\tilde{\boldsymbol{A}}')/(np)$. The simulation for three-factor NMF is as follows: first, we generate $\boldsymbol{F}$ such that

$$\tilde{f}_{im} \sim Ex\left(\left(\frac{\mu}{k_0 \ell_0}\right)^{1/3}\right) \; (i = 1, \ldots, n;\; m = 1, \ldots, k), \tag{8.13}$$

where ℓ_0 is the number of column factors. $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{S}}$ are also generated in the same manner as $\tilde{\boldsymbol{F}}$. Then, we obtain a noiseless data matrix $\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{F}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{A}}'$, after which we obtain the synthetic data $\boldsymbol{Y}$ by the following CP random number:

$$y_{ij} \sim CP(\tilde{x}_{ij}, \phi_0, \beta_0) \; (i = 1, \ldots, n;\; j = 1, \ldots, p). \tag{8.14}$$

Finally, we execute CP3NMF and CP3ONMF to $\boldsymbol{Y}$, obtain the estimated $\hat{\boldsymbol{F}}$, $\hat{\boldsymbol{S}}$, and $\hat{\boldsymbol{A}}$ by the two methods, and calculate $d_\beta(\boldsymbol{Y}, \tilde{\boldsymbol{F}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{A}}')/(np)$. We generate 50 synthetic data values for two- and three-factor NMF, and allow each of the four NMFs to execute 100 times per one synthetic data value. The parameter settings for generating synthetic data and for these algorithms are as follows:

- $n = 100$, $p = 60$, $k_0 = 5$, $\ell_0 = 3$
- $\mu = 10$
- $\phi_0 = 1$, $\beta_0 = 0.5$
- $\tau = 10^{-2}$, $\nu = 1000$
- $\delta = 50$, $\kappa = 100$

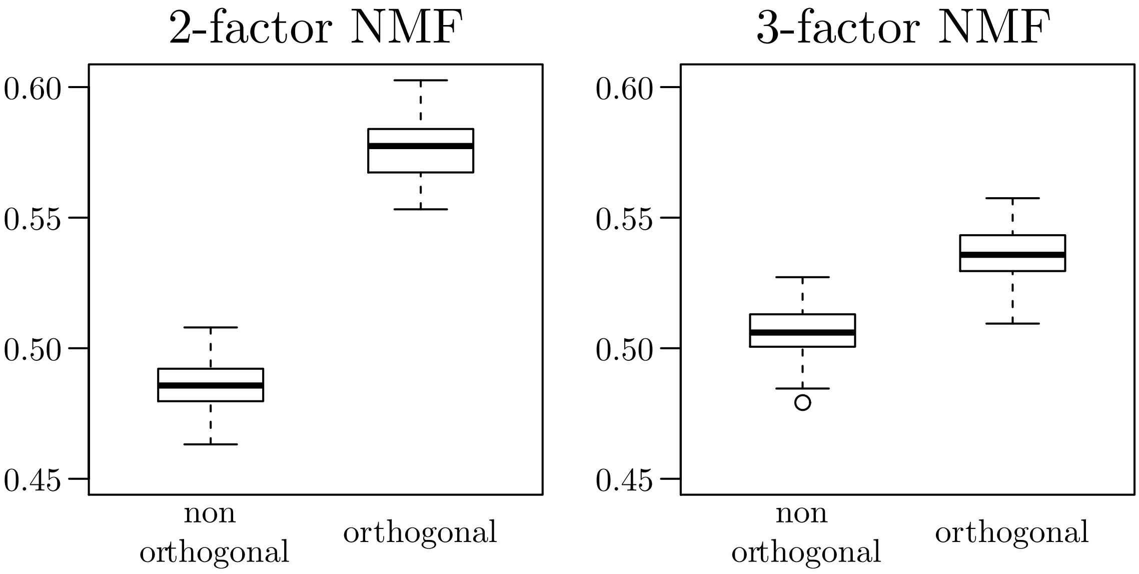


Figure 8.9: Beta divergence per one element between synthetic data $\boldsymbol{Y}$ and the approximation matrix $\tilde{\boldsymbol{X}}$ for two- and three-factor NMF with and without an orthogonal constraint.

Fig. 8.9 shows the result. The results show that the β-divergence of orthogonal NMF is larger than that of non-orthogonal NMF for both two- and three-factor NMF. This means that the orthogonal constraint adversely affects the approximation of NMF. This simulation enables us to conclude that NMF with an orthogonal constraint should be used considering the trade-off between its easy-to-understand estimates and its under fitting.

8.3 Approximation of NMF to zero-inflated matrix

In this section, we describe the numerical example that was used to test the NMF with the zero-inflated model in terms of approximation accuracy to the zero-inflated matrix. The following procedure was used in this simulation study. First, we generate the noiseless nonnegative matrix $\tilde{\boldsymbol{X}} \in \mathbb{R}^{n \times p}$ from the assumption of two- and three-factor NMF, and two- and three-factor orthogonal NMF as described later. Then, we obtain the non-zero-inflated nonnegative matrix $\boldsymbol{Y}^*$ by the following CP distribution:

$$y_{ij} \sim CP(\tilde{x}_{ij}, \phi_0, \beta_0) \ (i = 1, \dots, n; \ j = 1, \dots, p). \tag{8.15}$$

Next, we generate an artificial nonnegative data matrix $\boldsymbol{Y}$ such that $100w_0\%$ elements of $\boldsymbol{Y}^*$ are converted to zero. Finally, we execute the corresponding NMF with and without the zero-inflated model and calculate the degree of approximation according to the β-divergence for each estimation result. For the non-zero-inflated NMF, we calculate $d_\beta(\boldsymbol{Y}, \hat{\boldsymbol{X}})$, whereas for the zero-inflated NMF, we calculate $d_\beta(\boldsymbol{Y}, (\boldsymbol{E} - \hat{\boldsymbol{Z}}) \odot \hat{\boldsymbol{X}})$, where $\hat{\boldsymbol{X}}$ and $\hat{\boldsymbol{Z}}$ are the estimated approximation matrix and estimated latent variable matrix, respectively. The $\tilde{\boldsymbol{X}}$ is generated as follows.

Case of two-factor NMF

First, we generate $\tilde{\boldsymbol{F}} \in \mathbb{R}^{n \times k_0}$ such that

$$\tilde{f}_{im} \sim Ex\left(\left(\frac{\mu}{k_0}\right)^{1/2}\right) \; (i = 1, \ldots, n;\; m = 1, \ldots, k_0), \tag{8.16}$$

where k_0 is the number of factors. $\tilde{\boldsymbol{A}} \in \mathbb{R}_+^{p \times k_0}$ is also generated in the same manner as $\tilde{\boldsymbol{F}}$. In this way we obtain the noiseless data matrix $\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{F}}\tilde{\boldsymbol{A}}'$.

Case of three-factor NMF

First, we generate $\tilde{\boldsymbol{F}} \in \mathbb{R}^{n \times k_0}$ such that

$$\tilde{f}_{im} \sim Ex\left(\left(\frac{\mu}{k_0 \ell_0}\right)^{1/3}\right) \; (i = 1, \ldots, n;\; m = 1, \ldots, k_0), \tag{8.17}$$

where ℓ_0 is the number of column factors. $\tilde{\boldsymbol{A}} \in \mathbb{R}^{p \times \ell_0}$ and $\tilde{\boldsymbol{S}} \in \mathbb{R}^{k_0 \times \ell_0}$ are also generated in the same manner as $\tilde{\boldsymbol{F}}$. Then, we obtain the noiseless data matrix $\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{F}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{A}}'$.

Case of two-factor orthogonal NMF

First, we determine the true row clusters $\tilde{\mathcal{R}}$ randomly; we then generate true $\tilde{\boldsymbol{F}} \in \mathbb{R}^{n \times k_0}$ as follows:

$$\tilde{f}_{im} \sim \begin{cases} Ex(\mu^{1/2}) & (i \in \tilde{R}_m) \\ 0 & (i \notin \tilde{R}_m) \end{cases} \; (i = 1, \ldots, n;\; m = 1, \ldots, k_0). \tag{8.18}$$

Then, each element of true $\tilde{\boldsymbol{A}} \in \mathbb{R}^{p \times k_0}$ is generated as follows:

$$\tilde{a}_{jm} \sim Ex(\mu^{1/2}) \; (j = 1, \ldots, p;\; m = 1, \ldots, k_0). \tag{8.19}$$

Finally, we obtain the noiseless data matrix $\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{F}}\tilde{\boldsymbol{A}}'$.

Case of three-factor orthogonal NMF

First, we determine the true row clusters $\tilde{\mathcal{R}}$ randomly; we then generate true $\tilde{\boldsymbol{F}} \in \mathbb{R}^{n \times k_0}$ as follows:

$$\tilde{f}_{im} \sim \begin{cases} Ex(\mu^{1/3}) & (i \in \tilde{R}_m) \\ 0 & (i \notin \tilde{R}_m) \end{cases} \; (i = 1, \ldots, n;\; m = 1, \ldots, k_0). \tag{8.20}$$

Then, the true $\tilde{\mathcal{C}}$ and $\tilde{\boldsymbol{A}} \in \mathbb{R}^{p \times \ell_0}$ are generated in the same manner as $\tilde{\mathcal{R}}$ and $\tilde{\boldsymbol{F}}$. After that, each element of true $\tilde{\boldsymbol{S}} \in \mathbb{R}^{k_0 \times \ell_0}$ is generated as follows:

$$\tilde{s}_{mq} \sim Ex(\mu^{1/3}) \; (m = 1, \ldots, k_0;\; q = 1, \ldots, \ell_0). \tag{8.21}$$

Finally, we obtain the noiseless data matrix $\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{F}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{A}}'$.

We generate 50 synthetic data values for the combination of $w_0 = \{0.2, 0.4, 0.6\}$ and the four types of NMFs and perform 100 executions of each type of NMF per one synthetic data value. The parameter settings for generating the synthetic data and for these algorithms are as follows:

- $n = 100$, $p = 60$, $k_0 = 5$, $\ell_0 = 3$
- $\mu = 10$
- $\phi_0 = 1$, $\beta_0 = 0.5$
- $\tau = 10^{-2}$, $\nu = 1000$
- $\delta = 50$, $\kappa = 100$

Fig. 8.10, 8.11, and 8.12 show boxplots of the β-divergence for each of the corresponding situations. Fig. 8.10 relates to all elements of $\boldsymbol{Y}$, Fig. 8.11 only to the non-zero elements of $\boldsymbol{Y}$, and Fig. 8.12 relates only to the zero elements of $\boldsymbol{Y}$. As shown in Fig. 8.10, the β divergences for the zero-inflated NMF are smaller than those for the non-zero-inflated NMF for all four types of NMFs and w_0. This means that approximation using zero-inflated NMF is more accurate than that for non-zero-inflated NMF for any number of zero elements. Fig. 8.11 shows that β divergence values for the zero-inflated NMF are better than those for the non-zero-inflated NMF under all conditions, as does Fig. 8.10. This is a noteworthy result because it suggests that the use of zero-inflated NMF improves approximation not only for zero elements, but also for non-zero elements. Non-zero elements are generated from the CP distribution, which represents the factorization model. Hence, its result indicates that the factorized matrices estimated using zero-inflated NMF are better than those obtained using non-zero-inflated NMF, even in terms of approximation to non-zero elements. Fig. 8.12 shows that the proposed NMF model is even better than the basic model for zero elements.

Figs. 8.10 and 8.11 also indicate that the β divergence for non-zero-inflated NMF increased with the ratio of zero elements (w_0). This shows that a greater number of zero elements result in a less accurate approximation in non-zero-inflated NMF. On the other hand, the β divergence for the zero-inflated NMF model either remained constant or decreased slightly as w_0 increased. This is because the proposed NMF model does not take into account most zero elements, and it is easy to approximate fewer non-zero elements using the proposed NMF model.

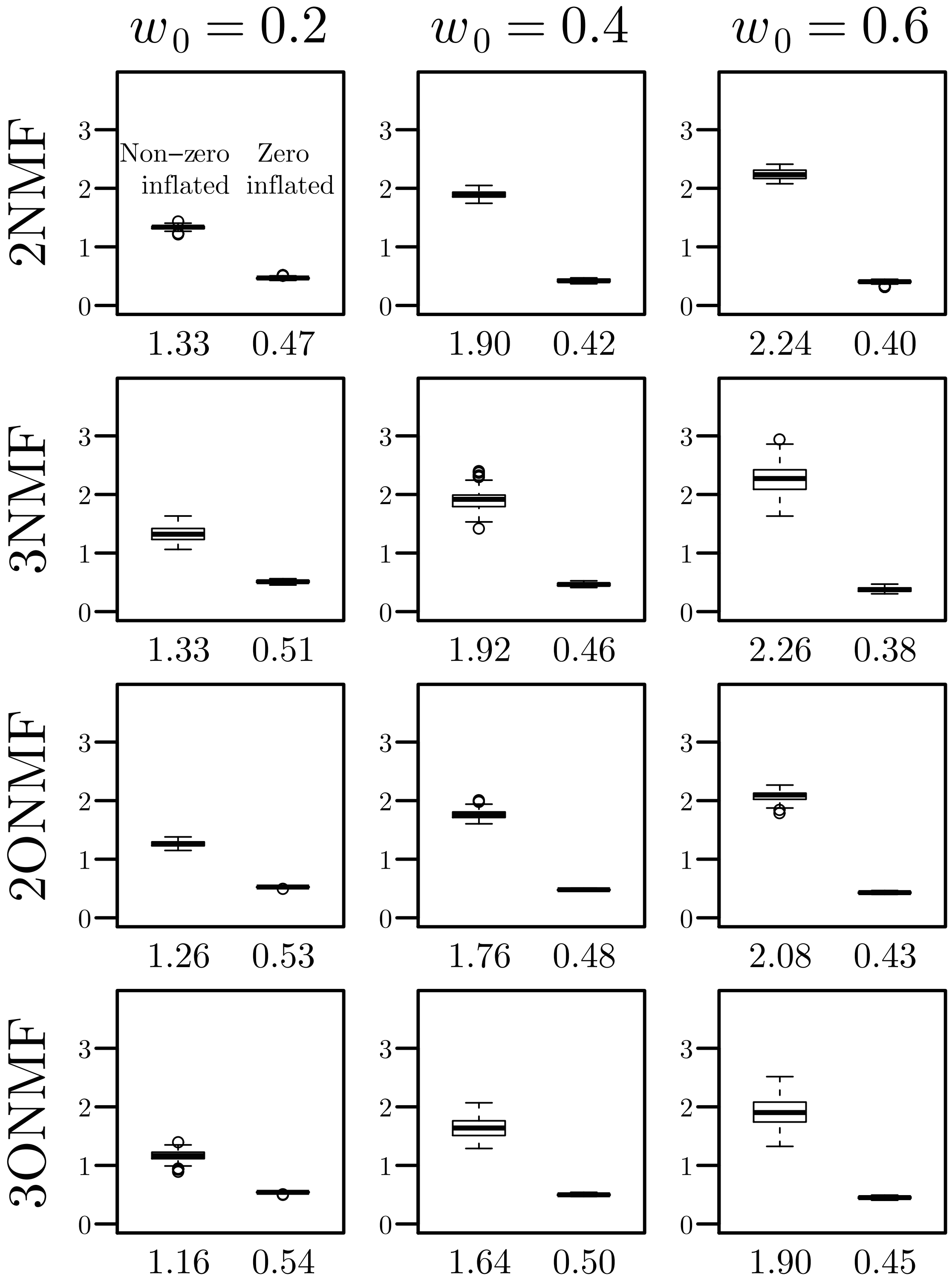


Figure 8.10: β divergence per element between synthetic data matrix $\boldsymbol{Y}$ and approximation matrix $\tilde{\boldsymbol{X}}$ by the non-zero-inflated (left boxplot) and the zero-inflated NMF model (right boxplot). Values at the bottom of each box are mean values.

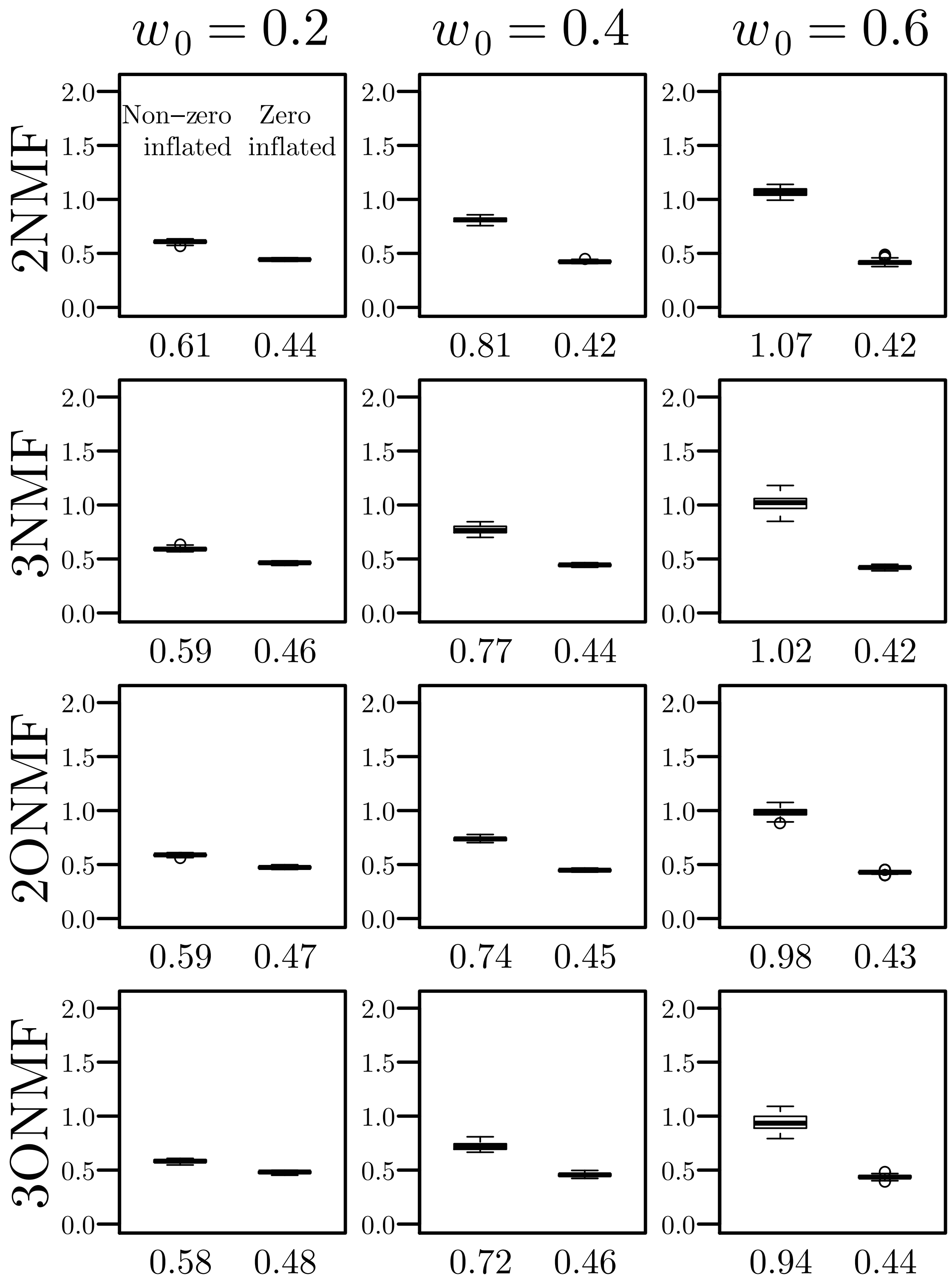


Figure 8.11: β divergence per element for only non-zero elements of synthetic data $\boldsymbol{Y}$ between $\boldsymbol{Y}$ and approximation matrix $\tilde{\boldsymbol{X}}$ by the non-zero-inflated (left boxplot) and the zero-inflated NMF model (right boxplot). Values at the bottom of each box are mean values.

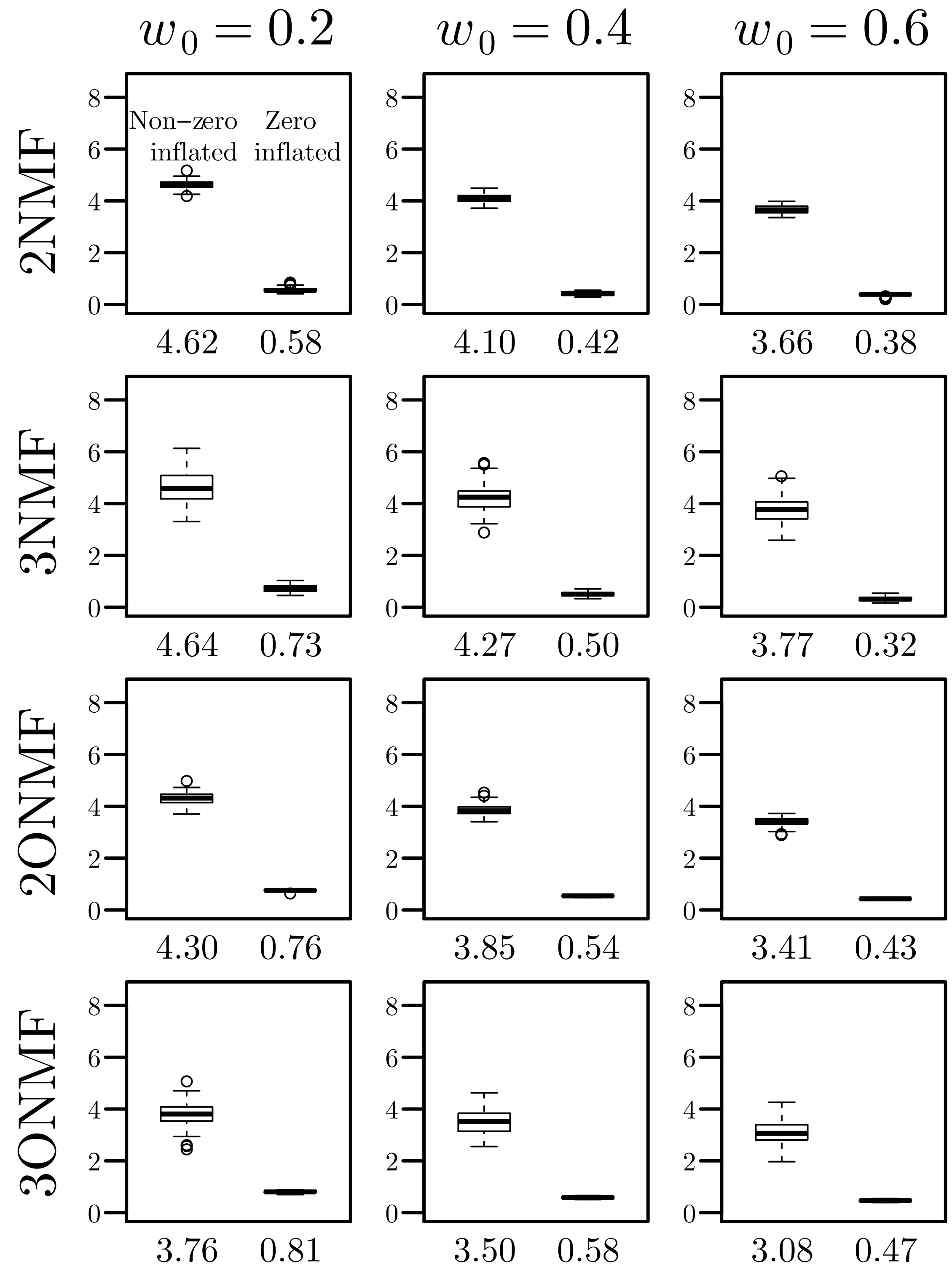


Figure 8.12: β divergence per element for only zero elements of synthetic data $\boldsymbol{Y}$ between $\boldsymbol{Y}$ and approximation matrix $\tilde{\boldsymbol{X}}$ by the non-zero-inflated (left boxplot) and the zero-inflated NMF model (right boxplot). Values at the bottom of each box are mean values.

Chapter 9

Applications

9.1 Document and term data

In this section, we describe an application involving a matrix containing document-term data to enable us to compare the clustering accuracy and computational time of N3ONMF with those of previous three-factor orthogonal NMFs, Ding et al. (2006) and Yoo and Choi (2010b). There are two reasons to compare these three methods: first, the three-factor NMF is compatible with the document-term clustering described in Section 3.1, second, these two three-factor orthogonal NMFs have some problems, as described in Section 3.2, and we are interested in its performance in a real data application. We do not use the other three-factor ONMF, P3ONMF, and CP3ONMF methods, in this application for the following reason. We use a document-term matrix that is converted using TF-IDF, which strongly weights terms in a few documents. The entries of these terms have large positive values. Hence, if we use a robust NMF, which implies that an NMF based on the Euclidean distance is not used, the effect of the weighted entries disappears and interpretable clusters cannot be obtained. Therefore, N3ONMF is appropriate in this application.

The data matrices we used were obtained from the open data CLUTO[1] website. The selected data matrices and statistics are listed in Table 9.1. The list of datasets in Table 9.1 are ordered by the number of elements. The *tr11*, *tr12*, *tr23*, *tr31*, *tr41*, and *tr45* datasets are derived from the TREC[2] collections. The true categories of the documents in *tr31* and *tr41* datasets are obtained by particular queries. The *re0* and *re1* datasets are from the Reuters-21578 text categorization test collection, distribution 1.0[3]. The *fbis* data set is from the Foreign Broadcast Information Service data of TREC-5. The *hitech* is a dataset of San Jose Mercury Newspaper articles and contains documents about computers, electronics, health, medicine, research, and technology. The *k1a*, *k1b*, and *wap* datasets were used for the WebACE project (Boley et al., 1999) and contain web pages in various subject directories of Yahoo![4]. Datasets *k1a* and *k1b* contain the same documents, but the true labels are different.

[1] `http://glaros.dtc.umn.edu/gkhome/views/cluto`

[2] `http://trec.nist.gov/`

[3] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

[4] `http://www.yahoo.com/`

Table 9.1: Statistics of selected text-word datasets in CLUTO. The mean, median, standard deviation (sd), minimum (min), and maximum (max) are calculated using the TF-IDF conversion.

data	documents	terms	classes	elements	nze	rnze	total-words	mean	median	sd	min	max
tr23	204	5831	6	1189524	78405	6.59	492911	0.004519	0.001060	0.012134	0.000007	0.531199
tr12	313	5799	8	1815087	84075	4.63	301180	0.007950	0.003572	0.014209	0.000001	0.372309
tr11	414	6424	9	2659536	114543	4.31	424511	0.007513	0.003291	0.014138	0.000001	0.458349
re0	1504	2886	13	4340544	77808	1.79	128671	0.050260	0.029622	0.068221	0.001719	2.260826
fbis	2463	2000	17	4926000	393386	7.99	1063914	0.012368	0.006438	0.020295	0.000008	0.574453
tr45	690	8261	10	5700090	193605	3.40	646537	0.007874	0.004378	0.014447	0.000000	0.803549
re1	1657	3758	25	6227006	87328	1.40	142680	0.060789	0.038921	0.072659	0.001609	1.498762
tr41	878	7453	10	6543734	170631	2.61	355524	0.012986	0.007958	0.019662	0.000018	1.333268
tr31	927	10127	7	9387729	247976	2.64	890507	0.008819	0.004770	0.018323	0.000009	2.055266
wap	1560	8440	20	13166400	189282	1.44	286987	0.026687	0.018241	0.030248	0.000239	0.825758
k1a	2340	21819	20	51056460	302992	0.59	454619	0.026748	0.018151	0.030898	0.000224	0.763670
k1b	2340	21819	6	51056460	302992	0.59	454619	0.026748	0.018151	0.030898	0.000224	0.763670
hitech	2301	22498	6	51767898	346881	0.67	549664	0.021766	0.013777	0.030802	0.000502	0.861557

We conducted term frequency-inverse document frequency (tf-idf) conversion for all data matrices. Before we started, we set the number of document clusters equal to the number of document classes provided, and the number of word clusters was set to 10 for all the data matrices. The convergence was determined using (3.11) and the convergence threshold was set as $\tau = 10^{-7}\|\boldsymbol{Y}\|$ for all the methods. The clustering accuracy was measured using the ARI between the given clusters and estimated clusters of the documents. It should be noted that some clusters occasionally become empty during iterative process to update N3ONMF. In this case, we restarted the update iteration from another initial parameter. Hence, we calculated the computational time of N3ONMF from the beginning of the first trial iteration to the end of the final trial iteration in which non-empty clusters are obtained. From among the candidates of the estimates given by 10 executions of each of the three methods, we select the best estimates for which the objective function value is minimized. The results are listed in Table 9.2.

Table 9.2: ARI between given clusters and estimated clusters of the documents and computational time generated by three methods for each CLUTO dataset.

	ARI			computational time (s)		
data	Ding	Yoo	N3ONMF	Ding	Yoo	N3ONMF
tr23	0.26	**0.30**	0.07	341	31	**6**
tr12	**0.55**	0.36	0.52	123	50	**3**
tr11	0.58	**0.71**	0.52	190	44	**6**
re0	**0.15**	0.07	0.10	420	29	**5**
fbis	0.28	0.35	**0.36**	301	128	**5**
tr45	0.11	0.22	**0.52**	1208	281	**22**
re1	0.08	0.07	**0.11**	243	63	**15**
tr41	0.23	0.41	**0.57**	991	288	**9**
tr31	0.06	0.15	**0.59**	2158	334	**16**
wap	0.39	0.33	**0.39**	3133	738	**27**
k1a	**0.37**	0.32	0.31	4244	5021	**138**
k1b	0.50	0.53	**0.74**	3629	2211	**147**
hitech	**0.19**	0.15	0.17	29012	6267	**77**

Although the performance of N3ONMF is less accurate for a relatively small data matrix, its performance improves in terms of estimating clusters for a relatively large data matrix. Moreover, the computational time of N3ONMF is extremely short in all cases. In fact, the N3ONMF has the fastest and deepest convergence, as shown in Fig. 9.1. These results imply that N3ONMF may be a superior method for estimating document clusters, because of its higher accuracy and computational efficiency.

We now show the estimates given by N3ONMF using *k1a* as an example to demonstrate how to interpret N3ONMF results. The *k1a* dataset consists of Web news documents

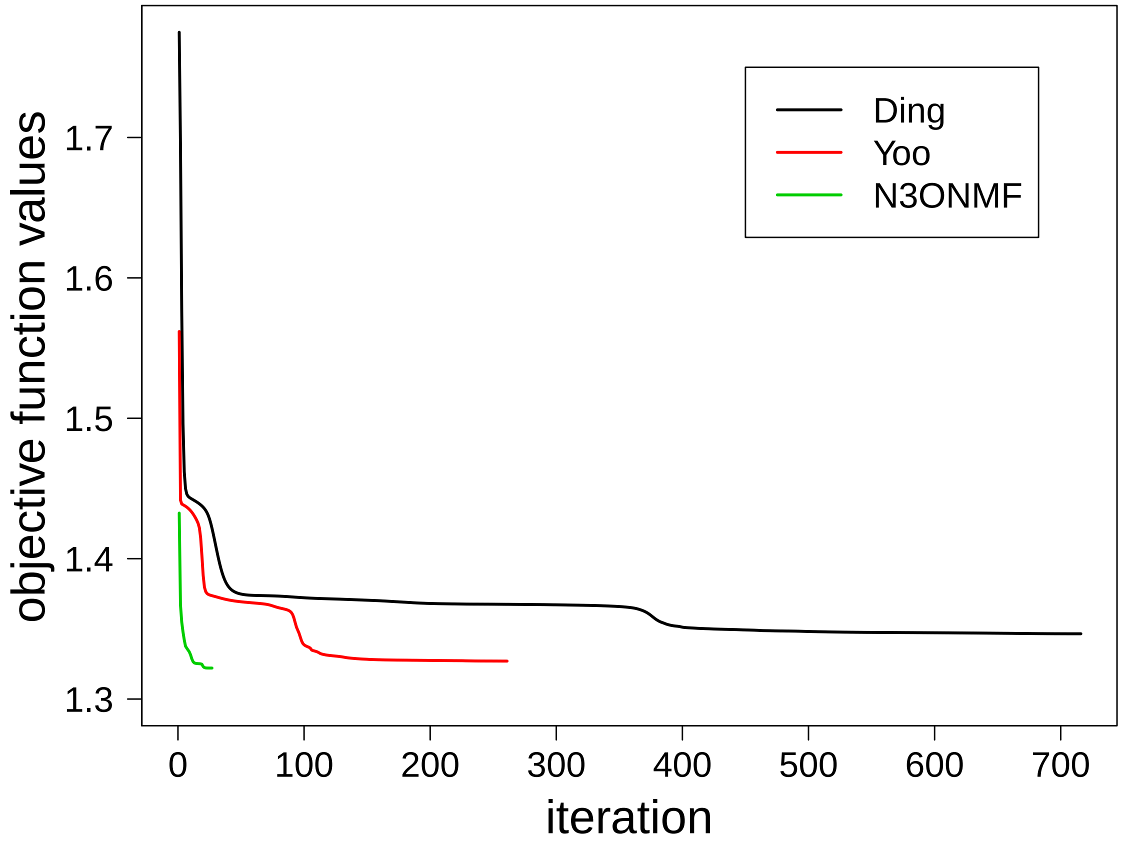


Figure 9.1: Plot of the sequence of the objective function values in iterations about "tr23" dataset for three NMF.

obtained from the Reuters news service in October, 1997 (Boley, 1998). In *k1a*, the documents are labeled by six categories, "business," "entertainment," "health," "politics," "sports," and "tech," in advance. Table 9.3 is a cross-tabulation of the number of documents between the given clusters and estimated clusters. We label the document clusters as DC 1 to DC 6. As shown in Table 9.3, "health," "sports," and "entertainment" doc-

Table 9.3: Cross-tabulation of the number of documents between the given and estimated clusters of the *k1a* dataset.

	DC 1	DC 2	DC 3	DC 4	DC 5	DC 6
business	1	141	0	0	0	0
entertainment	3	116	1189	47	30	4
health	492	2	0	0	0	0
politics	0	110	4	0	0	0
sports	0	0	4	1	0	136
tech	0	60	0	0	0	0

uments are clustered well. However, “business,” “politics,” and “tech” documents are contained in DC 2. The interpretation of each cluster is as follows.

DC 1 “Health” documents cluster.

DC 2 “Business,” “politics,” “tech,” and some “entertainment” documents cluster.

DC 3 First “entertainment” documents cluster.

DC 4 Second “entertainment” documents cluster.

DC 5 Third “entertainment” documents cluster.

DC 6 “Sports” documents cluster.

Table 9.4 presents the estimated factor matrix of the words. We label the word clusters as WC 1 to WC 10.

Table 9.4: Words and their factor matrix values for 10 clusters. Only words of rank more than or equal to 10 in each cluster are shown.

WC 1		WC 2		WC 3		WC 4		WC 5	
film	0.28	million	0.32	emmi	0.74	week	0.48	stock	0.28
box	0.19	rate	0.20	win	0.31	bestsell	0.45	internet	0.24
tv	0.18	deal	0.17	drama	0.25	weekli	0.35	compani	0.20
top	0.16	network	0.16	comedi	0.22	hardcov	0.33	dow	0.20
hollywood	0.15	am	0.15	actor	0.16	publish	0.28	clinton	0.17
cb	0.14	mondai	0.15	award	0.16	paperback	0.25	microsoft	0.16
star	0.14	set	0.14	franz	0.15	fiction	0.14	percent	0.16
offic	0.14	tuesdai	0.13	sundai	0.14	mass	0.11	comput	0.15
fox	0.12	cable	0.13	actress	0.14	random	0.10	house	0.14
diana	0.12	includ	0.12	gillian	0.13	edition	0.07	busi	0.14

WC 6		WC 7		WC 8		WC 9		WC 10	
cell	0.24	start	0.23	report	0.20	market	0.21	game	0.35
cancer	0.24	plai	0.21	accord	0.19	sale	0.18	season	0.20
risk	0.23	octob	0.20	people	0.16	quote	0.17	blackhawk	0.18
studi	0.22	record	0.19	american	0.16	presid	0.16	pippen	0.17
research	0.19	sign	0.19	develop	0.15	share	0.16	coach	0.16
patient	0.18	wednesdai	0.19	york	0.14	expect	0.15	marlin	0.16
women	0.17	hit	0.19	author	0.14	onlin	0.15	surgeri	0.15
diseas	0.17	run	0.18	death	0.14	earn	0.14	indian	0.14
heart	0.16	chicago	0.16	lead	0.13	unit	0.14	nomo	0.14
drug	0.13	five	0.14	famili	0.12	plan	0.14	oriol	0.14

Although a few clusters are ambiguous, we can find a meaning for most clusters. The interpretation of each word cluster is as follows.

WC 1 Words about cinema or television.

WC 2 Words about money ("million" and "rate") or date and time ("am," "mondai," and "tuesdai").

WC 3 Words about the Emmy awards. "gillian" and "franz" seem to be about Gillian Anderson and Dennis Franz. They won the outstanding lead actress and actor in a drama series of 49th Ammy award in 1997.

WC 4 Words about music CD sales ("week," "bestsell," "weekli," and "publish").

WC 5 Words about politics ("clinton," and "house"), economics ("stock," "compani," "dow," and "percent"), and technology ("internet," "microsoft," and "comput").

WC 6 Words about healthcare.

WC 7 Words about sports ("start" and "plai").

WC 8 Words about investigation ("report," "accord," and "author").

WC 9 Words about market ("market," "sale," "quote," and "share")

WC 10 Words about sports.

We can also grasp the relationship between the estimated document and word clusters using center factor matrix $\boldsymbol{S}$. Table 9.5 shows the values of its factor matrix. We find that DC 1 and DC 6 are well characterized by words in WC 6 and WC 10, respectively. This means that the words in WC 6 and WC 10 are effective for filtering documents

Table 9.5: Center factor matrix $\boldsymbol{S}$, which shows the relationship between the document and word clusters.

	DC 1	DC 2	DC 3	DC 4	DC 5	DC 6
WC 1	0.08	0.10	1.80	0.12	0.07	0.06
WC 2	0.24	0.49	0.97	0.08	0.03	0.14
WC 3	0.01	0.02	0.25	1.99	0.00	0.08
WC 4	0.06	0.05	0.15	0.02	1.75	0.03
WC 5	0.06	1.53	0.16	0.01	0.02	0.03
WC 6	1.98	0.07	0.10	0.01	0.01	0.02
WC 7	0.15	0.26	0.49	0.05	0.01	0.57
WC 8	0.79	0.33	0.40	0.03	0.03	0.10
WC 9	0.16	0.80	0.45	0.03	0.04	0.07
WC 10	0.05	0.04	0.13	0.01	0.00	1.45

about "health" and "sports." In contrast, DC 2 has a strong values at WC 5 and WC 9, which contain words connected with the subjects of politics, economics, the market, and technology. This indicates that these words are aggregated in two clusters and hence the documents about "business," "politics," and "tech" are not separated into different clusters. DC 2 also has some documents about "entertainment," and these documents may be about topics near to "business," "politics," and "tech." Although the other clusters, DC 3, DC 4, and DC 5 have documents labeled as "entertainment," they are related to different word clusters from each other. We can interpret that DC 3 is about entertainment in cinema or television, as DC 3 is related to WC 1, which contains words about the cinema or television industry. DC 4 is related to WC 3, which has words about the Emmy awards. We guess that these documents are about the 49th Primetime Emmy Awards held in September, 1997. DC 5 is related to WC 4, which consists of words about music CD sales. These results indicate that the documents labeled as "entertainment" are divided into four groups.

9.2 Point of sale data

In this section we describe the application of selected NMFs to point-of-sale data collected by a Japanese grocery store in June 2014, including customer ID information.[5] This application aims to observe the effects of CP distribution, the zero-inflated model, and an orthogonal constraint. We created a matrix that includes customer spending in monetary units (rows) in the various product categories (columns) using the following data cleansing steps:

Step 1: We removed those product categories for which the cumulative sum of sales is less than JPY 1,000,000. This is the same as removing the product categories for which the total sales is less than JPY 74,682.

Step 2: We removed those customers for whom the cumulative relative sum of customer spending was less than 30%. This is the same as removing the customers who spent less than JPY 5,719.

Step 3: We removed those customers for whom the number of product categories for their customer purchases was less than or equal to five.

The statistics for the original data set and the data set after cleansing are provided in Table 9.6.

Summaries of the cleansed data are shown in Fig. 9.2, Fig. 9.3, and Table 9.7. From Fig. 9.2, there are some peaks in the total sales. This indicates that some groups of customers may exist in this store. In Fig. 9.3, many customers purchased items from 15 to 30 categories.

[5] "i-codePOS Data" provided by IDS Co., Ltd. in the 2015 Data Analysis Competition hosted by the Joint Association Study Group of Management Science.

Table 9.6: Statistics for the original dataset and the dataset after cleansing based on the point-of-sale data.

	Original	After cleansing
Customers	33,456	7,348
Product categories	146	114
Proportion of zero elements	0.928	0.774
Total sales (JPY)	165,169,493	114,143,984

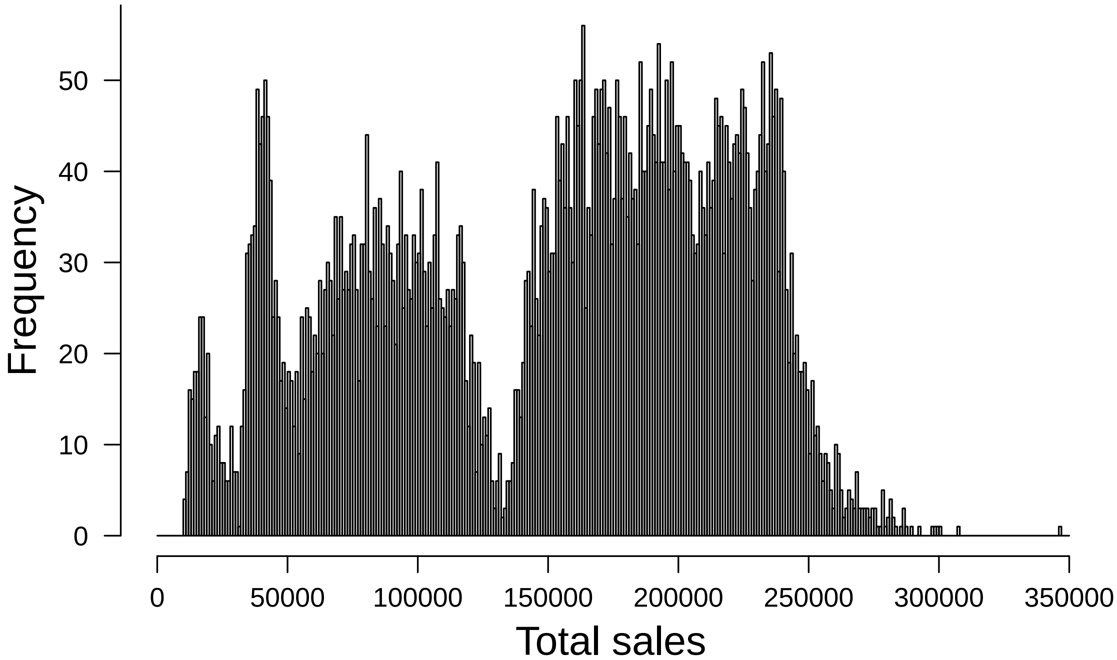


Figure 9.2: Histogram of customers' total sales.

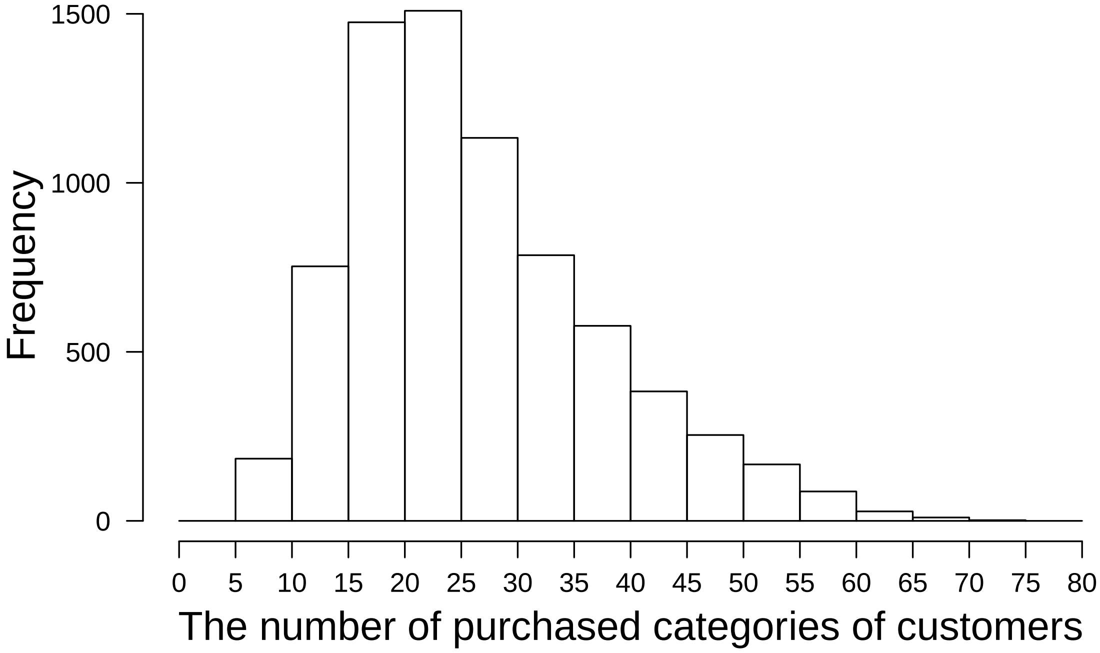


Figure 9.3: Histogram of the number of customers' purchasing categories.

Table 9.7: Summary of each category. The mean, standard deviation (sd), and quantiles are calculated using non-zero values.

category	total sales	propotion of purchasing customer	mean	sd	0%	25%	50%	75%	100%
Fruit vegetable	4416681	0.74	815	941	41	278	536	1,022	26,892
Milk product	3435222	0.61	768	843	60	272	511	959	12,930
Mizumono	3338322	0.74	611	649	49	206	407	768	10,184
Sushi	3214221	0.36	1,226	1,211	113	453	836	1,608	15,333
Bread	3161700	0.60	713	768	68	236	464	885	9,468
Imported fruit	3029955	0.57	729	826	54	213	466	921	18,396
Fruits in season	3027482	0.41	1,007	1,165	5	340	646	1,222	20,130
Milk beverage	2771834	0.55	686	799	64	205	418	829	14,421
Dry confectionery	2708673	0.53	698	865	20	216	438	845	20,860
Japanese cake	2673054	0.57	639	752	46	216	416	779	18,804
Processed meat	2575018	0.51	688	626	90	290	497	867	9,198
Vegetable-related	2516517	0.46	741	1,057	59	216	410	829	18,091
Japanese-grown Pork	2433614	0.40	821	874	111	339	522	968	10,074
Processing seasoning	2337306	0.53	603	553	31	243	422	766	6,245
Salad deli	2310696	0.48	652	740	47	206	410	810	10,012
Fried food deli	2282793	0.47	667	654	54	258	452	841	6,506
Basic seasoning	2132177	0.47	612	573	49	243	422	781	8,035
Japanese pickle	2060151	0.49	577	571	50	213	408	711	7,385
Refreshing beverage	1987744	0.44	613	880	70	162	320	724	16,678
Leaf vegetable	1933815	0.61	428	524	21	148	275	515	9,048

category	total sales	propotion of purchasing customer	mean	sd	0%	25%	50%	75%	100%
Bread deli	1839351	0.42	601	713	3	207	388	713	10,668
Beer	1777593	0.17	1,412	1,788	113	403	744	1,582	16,357
Precooked foods deli	1735755	0.40	593	621	50	257	391	705	9,918
Japanese-style deli	1708979	0.38	618	746	68	203	406	726	11,948
Noodle	1639796	0.43	518	515	24	210	357	639	5,786
Sashimi	1628552	0.24	917	815	153	429	628	1,084	9,806
Chicken egg	1546783	0.48	441	463	65	204	314	531	8,645
Fillet	1533978	0.22	954	964	129	433	628	1,137	14,999
Root vegetable	1529668	0.52	399	464	21	138	270	483	10,044
Unclassified product	1519899	0.43	484	559	2	178	306	576	8,501
Paste	1501783	0.44	460	451	70	194	306	570	6,574
Rice deli	1488056	0.28	729	758	4	264	481	922	9,641
Stem vegetable	1443297	0.49	405	640	1	138	269	480	19,576
Rice	1366755	0.10	1,794	1,298	131	1,010	1,382	2,167	9,505
Wagyu beef	1292351	0.11	1,559	1,796	168	657	978	1,834	26,886
Boxed lunch	1190990	0.16	1,012	1,069	204	410	711	1,226	17,361
Grilled deli	1114096	0.26	573	629	91	238	359	709	10,949
Jelly and pudding	1057971	0.30	479	525	60	192	307	577	7,009
Whole fish	1051023	0.21	697	832	95	321	431	824	21,161
Gifts and brand-name confectionery	1046684	0.06	2,213	5,607	108	432	866	1,782	61,128

category	total sales	propotion of purchasing customer	mean	sd	0%	25%	50%	75%	100%
Mushroom	1039779	0.46	309	295	51	105	213	391	3,450
Small fish	969584	0.27	489	435	106	223	325	578	6,331
Japanese-grown chicken	959667	0.25	528	477	60	224	385	639	5,344
Ground meat	919924	0.25	502	471	83	191	345	627	4,232
Vegetable and fruit beverage	918702	0.26	486	565	75	152	298	572	5,697
Boiled beans and food boiled in soy sauce	901114	0.28	441	458	59	203	321	520	5,928
Tasty beverage	843213	0.19	599	543	102	210	419	717	4,931
Imported pork	836716	0.18	622	584	136	279	392	739	7,521
Chinese-style deli	815469	0.23	492	403	95	219	378	606	4,574
Garnishing served with raw fish	805662	0.37	297	367	32	105	200	340	6,471
Dry marine food	791254	0.22	481	468	95	243	307	572	6,001
Japanese-grown beef	738639	0.14	735	585	114	406	518	885	5,425
Book	736845	0.14	727	674	145	362	490	841	7,040
Snack deli	700271	0.21	461	420	81	210	324	552	3,825
Wine	673515	0.06	1,555	2,024	203	584	1,014	1,788	29,115
Instant soup	610094	0.20	425	418	91	194	307	489	3,797
Marine processed food	606417	0.17	480	430	108	286	321	548	4,530
Noodle deli	579056	0.14	568	536	84	275	399	641	5,267
Salt curing	571958	0.12	661	582	111	373	452	785	6,210
Deep-frozen food	545833	0.13	591	625	70	204	408	663	5,764

category	total sales	propotion of purchasing customer	mean	sd	0%	25%	50%	75%	100%
Ice cream	526085	0.11	670	1,059	65	255	396	767	21,095
Flower	524425	0.11	648	667	64	259	410	780	5,646
Instant noodle	518564	0.18	384	323	81	182	276	473	2,480
Distilled spirit (Shochu)	504090	0.04	1,946	1,887	144	927	1,219	2,413	15,871
Dry food	489481	0.14	462	379	75	285	321	543	5,291
Liqueur	488895	0.10	671	999	91	162	337	781	11,855
Sake	485252	0.05	1,260	1,452	102	432	862	1,382	14,541
Been	482392	0.18	367	390	54	138	243	429	5,240
Brand chicken	466575	0.11	601	484	138	317	459	703	3,988
Cooking oil	459002	0.12	536	464	142	286	379	685	5,778
Seared fish	440511	0.10	583	424	205	395	417	648	4,522
Seasoned meat	440088	0.12	515	402	100	255	370	691	3,784
Shellfish	410673	0.12	478	373	100	321	321	599	3,581
Sashimi platter	409876	0.07	818	705	153	431	614	978	6,480
Other deli	399103	0.10	535	552	90	257	372	619	6,720
Agricultural dry food	384078	0.17	302	292	78	131	204	361	2,604
Cooked rice seasoning	363612	0.16	318	345	90	131	215	384	4,516
Seaweed	362121	0.16	307	318	73	172	204	338	5,184
Brand poke	361865	0.08	582	373	129	367	434	732	3,825
Fish pickled in salt	353913	0.10	496	384	95	306	380	601	3,331

category	total sales	propotion of purchasing customer	mean	sd	0%	25%	50%	75%	100%
Spread and dipping sauce	347716	0.10	465	388	95	210	382	620	3,597
Boiled fish	339134	0.10	468	394	116	238	388	497	3,780
Dry mixture	324723	0.15	299	303	90	135	215	346	5,510
Cut fruit	324119	0.08	569	523	86	278	378	656	4,781
Dry fruit	321234	0.08	521	601	95	204	310	552	4,756
Fish egg	320130	0.08	546	347	185	308	410	631	2,444
Germinating vegetable	303447	0.33	124	184	20	41	82	124	3,654
Chinese style semi-finished deli	292243	0.12	343	277	95	203	255	410	2,564
Condiment	282643	0.15	262	188	90	145	204	311	1,728
Snack food eaten while drinking	279174	0.10	398	485	101	203	276	432	8,970
Cut vegetable	250409	0.14	251	275	52	105	138	276	3,484
Precooked rice foods	238728	0.06	578	757	3	277	394	578	9,444
Western liquor	230108	0.01	2,585	2,695	810	1,276	1,624	2,861	16,122
Spitchcock deli (Kabayaki)	215863	0.02	1,910	1,287	376	1,062	1,385	2,231	10,285
Western style semi-finished deli	192909	0.05	569	400	156	302	472	679	3,417
Boiled vegetables	182965	0.08	320	267	101	152	206	364	2,155
Processed agricultural foods	149231	0.07	278	418	91	124	170	286	5,702
Material for confectionery	148016	0.05	381	354	102	178	294	410	3,406
Cereal	140346	0.03	610	429	192	311	451	821	2,831
Organ meat	138425	0.04	449	456	100	224	308	521	5,240

category	total sales	propotion of purchasing customer	mean	sd	0%	25%	50%	75%	100%
Marine raw food	135991	0.04	498	310	245	306	312	601	2,162
Western style semi-finished deli	132642	0.05	336	294	65	182	222	410	3,189
Skin dough	127049	0.05	325	336	95	162	256	351	3,321
Non-alcoholic beverage	126554	0.04	428	516	86	118	231	551	4,148
Farming raw diet	122532	0.04	464	290	176	287	360	517	2,162
Grilled semi-finished deli	107149	0.03	567	386	205	360	411	650	2,470
Frozen vegetable	90408	0.03	362	453	170	204	204	408	4,696
Imported beef	89933	0.01	1,084	970	297	562	835	1,105	6,394
Other items	82050	0.02	506	800	129	204	321	506	8,318
Snack semi-finished deli	78479	0.03	365	270	69	282	306	358	2,573
Deli platter	73874	0.03	373	325	141	203	203	418	2,437
Beef-related	62363	0.01	605	474	181	292	409	747	2,349
Items in other counter	49822	0.40	17	23	5	5	10	20	545
Vegetable processed food	48927	0.02	333	446	32	105	278	324	4,536

Using this data set, we obtained a factor matrix of product classification from N2NMF, CP2NMF, ZICP2NMF, and CP2ONMF. We compared the estimated factor matrix, between N2NMF and CP2NMF, to confirm the effect of CP distribution; CP2NMF and ZICP2NMF, to confirm the effect of the zero-inflated model; and CP2NMF and CP2ONMF, to confirm the effect of the orthogonal constraint. From among the 20 estimate candidates, we selected the estimator for the purpose of maximizing the objective function value. The parameter settings for the algorithm of these methods are as follows:

- $k = 5$
- $\beta = 0.5$
- $\tau = 10^{-2}$, $\nu = 1000$
- $\delta = 20$, $\kappa = 100$.

The number of clusters k is commonly determined in various appropriate ways by using information criteria, cross-validation, or a Bayesian method. However, for this application we select $k = 5$ to enable us to easily verify the characteristics of the estimators obtained by each NMF, and because of space limitations. The estimators of the factor matrix for product categories $\boldsymbol{A}$ obtained by the four NMFs are provided in Table 9.8, 9.9, 9.10, and 9.11. All factor matrices for product categories $\boldsymbol{A}$ are standardized such that the length of each column vector is 1. The table only includes product classifications with values greater than 0.2.

Table 9.8: Factor matrix $\boldsymbol{A}$ produced by N2NMF.

product classifications	N2NMF				
	1	2	3	4	5
Gifts and brand-name confectionery	1.00	0.00	0.00	0.00	0.00
Sushi	0.00	0.95	0.00	0.00	0.00
Beer	0.00	0.00	0.97	0.00	0.00
Fruit vegetable	0.01	0.00	0.01	0.47	0.00
Vegetable-related	0.00	0.00	0.00	0.29	0.00
Fruits in season	0.00	0.17	0.00	0.28	0.00
Mizumono	0.00	0.02	0.04	0.25	0.10
Imported fruit	0.00	0.03	0.00	0.24	0.08
Japanese-grown Pork	0.00	0.01	0.02	0.23	0.03
Leaf vegetable	0.00	0.01	0.02	0.20	0.00
Bread	0.01	0.00	0.01	0.05	0.38
Dry confectionery	0.01	0.00	0.00	0.03	0.35
Fresh Japanese sweets	0.01	0.05	0.00	0.03	0.33
Salad deli	0.00	0.08	0.01	0.00	0.30
Refreshing beverage	0.01	0.00	0.04	0.00	0.29
Fried food deli	0.00	0.09	0.03	0.01	0.26
Japanese-style deli	0.00	0.09	0.00	0.00	0.21
Milk product	0.01	0.00	0.02	0.20	0.20

Table 9.9: Factor matrix $\boldsymbol{A}$ produced by CP2NMF.

product classifications	CP2NMF				
	1	2	3	4	5
Rice	0.62	0.00	0.00	0.00	0.00
Beer	0.49	0.00	0.00	0.09	0.09
Sashimi	0.31	0.04	0.00	0.18	0.00
Wine	0.24	0.00	0.00	0.00	0.00
Sushi	0.12	0.54	0.01	0.00	0.04
Salad deli	0.08	0.29	0.00	0.10	0.03
Rice deli	0.00	0.26	0.00	0.00	0.00
Fried food deli	0.10	0.25	0.04	0.05	0.06
Fresh Japanese sweets	0.03	0.24	0.15	0.08	0.03
Bread	0.03	0.23	0.13	0.17	0.18
Lunch box	0.00	0.20	0.00	0.00	0.00
Fruit vegetable	0.04	0.03	0.42	0.27	0.28
Mizumono	0.07	0.06	0.28	0.20	0.20
Imported fruit	0.03	0.10	0.26	0.22	0.07
Vegetable-related	0.04	0.03	0.25	0.15	0.08
Fillet	0.05	0.00	0.23	0.00	0.00
Fruits in season	0.12	0.12	0.23	0.19	0.01
Milk product	0.06	0.10	0.16	0.42	0.16
Wagyu beef	0.00	0.00	0.00	0.33	0.00
Milk beverage	0.04	0.12	0.13	0.26	0.12
Dry confectionery	0.05	0.19	0.08	0.21	0.11
Japanese-grown Pork	0.03	0.00	0.17	0.11	0.37
Processed meat	0.05	0.02	0.17	0.16	0.31
Processing seasoning	0.08	0.00	0.16	0.17	0.26
Deep-frozen food	0.00	0.00	0.00	0.00	0.22
Imported pork	0.00	0.00	0.05	0.00	0.22

Table 9.10: Factor matrix $\boldsymbol{A}$ produced by ZICP2NMF.

product classifications	ZICP2NMF				
	1	2	3	4	5
Beer	0.62	0.01	0.00	0.00	0.00
Milk product	0.23	0.16	0.22	0.10	0.08
Processing seasoning	0.23	0.04	0.14	0.09	0.05
Noodle	0.21	0.00	0.11	0.00	0.07
Sushi	0.00	0.35	0.10	0.28	0.12
Rice	0.00	0.35	0.01	0.00	0.00
Refreshing beverage	0.08	0.33	0.00	0.05	0.05
Lunch box	0.00	0.33	0.00	0.00	0.00
Dry confectionery	0.08	0.27	0.10	0.04	0.15
Bread	0.09	0.27	0.14	0.09	0.16
Rice deli	0.00	0.26	0.00	0.06	0.09
Fresh Japanese sweets	0.05	0.23	0.11	0.08	0.18
Fruit vegetable	0.23	0.03	0.39	0.18	0.07
Vegetable-related	0.06	0.00	0.31	0.04	0.03
Mizumono	0.16	0.04	0.28	0.09	0.11
Imported fruit	0.15	0.09	0.26	0.06	0.06
Fruits in season	0.03	0.05	0.25	0.21	0.13
Fillet	0.00	0.00	0.23	0.00	0.00
Japanese-grown Pork	0.14	0.00	0.21	0.13	0.02
Wagyu beef	0.01	0.00	0.00	0.56	0.00
Japanese-grown beef	0.00	0.00	0.00	0.40	0.00
Wine	0.00	0.00	0.00	0.26	0.00
Fried food deli	0.03	0.11	0.03	0.06	0.39
Japanese-style deli	0.00	0.06	0.01	0.02	0.38
Salad deli	0.02	0.17	0.00	0.07	0.37
Grilled deli	0.00	0.02	0.00	0.00	0.33
Gifts and brand-name confectionery	0.00	0.00	0.00	0.00	0.23

Table 9.11: Factor matrix $\boldsymbol{A}$ produced by CP2ONMF.

product classifications	CP2ONMF				
	1	2	3	4	5
Beer	0.59	0.00	0.00	0.00	0.00
Fillet	0.59	0.00	0.00	0.00	0.00
Small fish	0.46	0.00	0.00	0.00	0.00
Sushi	0.00	0.57	0.00	0.00	0.00
Fried food deli	0.00	0.41	0.00	0.00	0.00
Refreshing beverage	0.00	0.37	0.00	0.00	0.00
Precooked foods deli	0.00	0.32	0.00	0.00	0.00
Japanese-style deli	0.00	0.30	0.00	0.00	0.00
Rice deli	0.00	0.25	0.00	0.00	0.00
Fruit vegetable	0.00	0.00	0.33	0.00	0.00
Milk product	0.00	0.00	0.27	0.00	0.00
Mizumono	0.00	0.00	0.26	0.00	0.00
Bread	0.00	0.00	0.25	0.00	0.00
Imported fruit	0.00	0.00	0.24	0.00	0.00
Fruits in season	0.00	0.00	0.23	0.00	0.00
Dry confectionery	0.00	0.00	0.22	0.00	0.00
Fresh Japanese sweets	0.00	0.00	0.22	0.00	0.00
Milk beverage	0.00	0.00	0.21	0.00	0.00
Cooked beans and tsukudani	0.00	0.00	0.00	0.56	0.00
Japanese-grown chicken	0.00	0.00	0.00	0.53	0.00
Sliced fish for sashimi	0.00	0.00	0.00	0.35	0.00
Wine	0.00	0.00	0.00	0.23	0.00
Ice cream	0.00	0.00	0.00	0.23	0.00
Brand chicken	0.00	0.00	0.00	0.21	0.00
Agricultural dry food	0.00	0.00	0.00	0.21	0.00
Wagyu beef	0.00	0.00	0.00	0.00	0.52
Tasty beverage	0.00	0.00	0.00	0.00	0.47
Snack deli	0.00	0.00	0.00	0.00	0.45
Instant soup	0.00	0.00	0.00	0.00	0.37
Gifts and brand-name confectionery	0.00	0.00	0.00	0.00	0.27
Cooked rice seasoning	0.00	0.00	0.00	0.00	0.22

The interpretations of the estimated factors are as follows. We indicate the m-th factor of N2NMF, CP2NMF, ZICP2NMF, and CP2ONMF as N2_m, CP2_m, ZICP2_m, and CP2O$_m$, respectively.

Interpretation of the N2NMF result

N2_1: Buying gifts and brand-name confectionery.

N2_2: Buying sushi.

N2_3: Buying beer.

N2_4: Buying vegetables, fruits, and mizumono (e.g., natto, konjac, and tofu.

N2_5: Buying bread, confectionery, beverages, and deli foods, which are ready-to-eat foods.

Interpretation of the CP2NMF result

CP2_1: Buying rice, beer, and sashimi.

CP2_2: Buying deli and the other ready-to-eat foods.

CP2_3: Buying vegetables, fruits, and mizumono. This factor is similar to N2_4.

CP2_4: Buying items made out of milk and wagyu beef.

CP2_5: Buying meat (mainly pork) and processing seasonings.

Interpretation of the ZICP2NMF result

ZICP2_1: Buying mainly beer.

ZICP2_2: Buying refreshing beverages and somethings like a complete meal, e.g., sushi, lunch boxes, bread, and rice deli.

ZICP2_3: Buying vegetables, fruits, and mizumono. This factor is similar to N2_4 and CP2_3.

ZICP2_4: Buying beef and wine.

ZICP2_5: Buying deli foods such as side dishes.

Interpretation of the CP2ONMF result

CP2O$_1$: Buying beer and fish.

CP2O$_2$: Buying sushi and other deli foods such as side dishes.

CP2O$_3$: Buying items like those of N2_4 and CP2_3, e.g., vegetables, fruits, and mizumono, items made out of milk, and items like confectionery.

$\mathrm{CP2O}_4$: Buying chicken, cooked beans, and tsukudani.

$\mathrm{CP2O}_5$: Buying wagyu beef and refreshing beverages .

Summaries of the estimators of the factor matrix for customers $\boldsymbol{F}$ are provided in Table 9.12. $\boldsymbol{F}$ represents something like the each customer's amount of spending for the product categories in each factor.

Table 9.12: Summaries of the estimators of factor matrix $\boldsymbol{F}$ for customers.

		1	2	3	4	5
N2NMF	mean	146.6	494.2	281.9	1,286.2	971.4
	sd	1,521.9	949.7	922.5	1,500.4	1,024.8
	min	0.0	0.0	0.0	0.0	0.0
	25%	0.0	0.0	0.0	409.9	313.3
	median	0.2	88.1	7.6	862.9	706.1
	75%	7.9	576.1	110.0	1,639.6	1,287.4
	max	61,131.2	14,899.3	16,169.6	28,182.1	15,333.0
CP2NMF	mean	333.6	802.5	800.9	490.3	363.9
	sd	709.6	1,126.1	1,074.7	748.6	599.4
	min	0.0	0.0	0.0	0.0	0.0
	25%	0.0	0.0	0.0	0.0	0.0
	median	0.0	459.3	487.7	213.4	97.7
	75%	398.8	1,147.3	1,097.0	694.6	517.7
	max	12,174.8	15,265.5	12,881.1	9,420.6	8,368.9
ZICP2NMF	mean	463.0	585.6	1,001.1	320.0	504.0
	sd	790.3	884.5	1,277.1	575.5	837.9
	min	0.0	0.0	0.0	0.0	0.0
	25%	0.0	0.0	7.3	0.0	0.0
	median	129.7	243.5	667.8	0.0	201.7
	75%	651.7	856.6	1,363.4	441.4	694.8
	max	13,270.2	13,932.4	18,526.9	9,841.8	13,575.6
CP2ONMF	mean	387.4	824.7	1,785.1	289.4	303.7
	sd	661.4	979.7	1,508.0	438.1	739.6
	min	0.0	0.0	54.3	0.0	0.0
	25%	0.0	265.5	860.0	0.0	0.0
	median	156.5	543.4	1,273.9	156.8	122.0
	75%	474.5	1,039.0	2,153.7	385.2	345.4
	max	11,543.5	18,006.8	18,493.1	12,251.9	24,325.0

The results of $\boldsymbol{A}$ show that all methods extract two factors: buying basic Japanese foods, e.g., fruits, vegetables, mizumonoes, and buying ready-to-eat foods, e.g., some deli foods, breads, lunch boxes, and beverages. Table 9.12 indicates that the amount paid for these two factors' items is large. In contrast, the details of the other factors are different for each NMF method. For example, in the results of ZICP2NMF (Table 9.10), the two ready-to-eat food factors are estimated: a complete meal and side dishes. However, other methods provide one factor including these foods.

Below, we discuss the characteristics of each NMF by comparing them in pairs.

N2NMF vs CP2NMF

The factor matrix of N2NMF in Fig. 9.8 indicates that three bases are estimated with only one extremely strong value ("Gifts and brand-name confectionery," "Sushi," and "Beer"). In contrast, the result obtained by CP2NMF does not reflect any such bases, all of which have middle-ranging values for various product classifications. Thus, this result shows the effect of robust estimation using CP distribution. The basis for which we obtained only one extreme value is estimated when the data contains outliers. Fig. 9.4 plots of the number of customers whose proportion of money spent in a single category is more than $r\%$ for each category. This figures shows that there are a relatively large

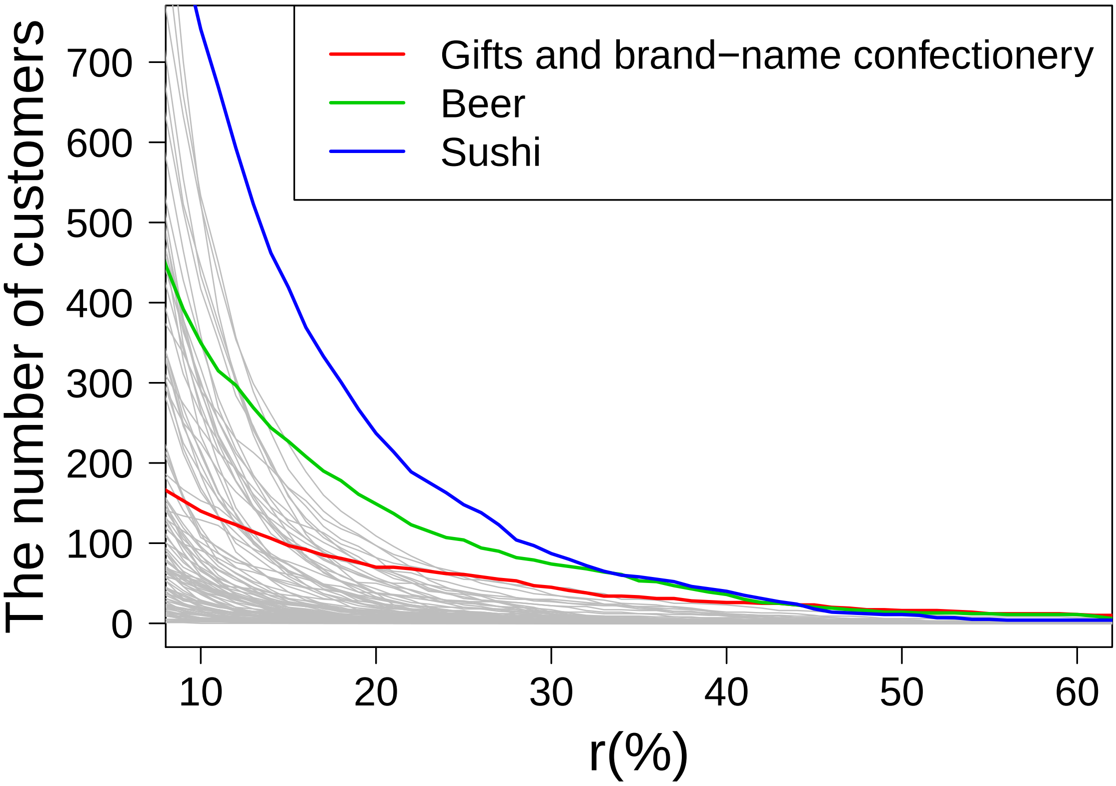


Figure 9.4: Number of customers whose proportion of money spent in a single category is more than $r\%$ for each category. For example, there are 87 customers who spent more than 30% of his/her total spending on "Sushi" items.

number of customers who spent money on “Sushi,” “Beer,” or “Gifts and brand-name confectionery” at a high rate, e.g, more than $r = 30\%$. Moreover, Table 9.7 shows that these categories all have high mean values for money spent but have a low proportion of purchasing customers. For “Gifts and brand-name confectionery,” the maximum money spent is the highest of all categories and is a very large value (JPY 61,128). These results suggest that the large values of a few samples strongly affect the estimates of factor matrices in NMFs using a normal distribution.

However, for the CP2NMF, such a extreme basis is difficult to estimate, even if there were outliers, because the penalty for large data values is weaker than for small data values in terms of β-divergence, as seen in Section 3.3. From the point of view of interpretation, a basis with only one extreme value is unavailing because the aim of NMF is to capture the co-occurrence relation. Such a basis indicates that there is no co-occurrence with an item that has such an extreme value.

CP2NMF vs ZICP2NMF

Fig. 9.10 displays the effect of the zero-inflated model: factor matrices are estimated considering some values of $y_{ij} = 0$ as non-zero. For example, “Wine” and “Beef,” (e.g., “Wagyu beef” and “Japanese-grown beef”) are not in the same factor in the estimates of CP2NMF. This means that few customers buy both of these items. However, in ZICP2NMF, customers buying either “Wine” or “Beef” but not both are regarded as customers buying both, because some of the zero values in data matrix $\boldsymbol{Y}$ are disregarded. In other words, some elements of $y_{ij} = 0$ in the “Wine” or “Beef” columns are assumed not to be generated from the distribution $y_{ij} \sim CP(x_{ij}, \phi, \beta)$ but from $y_{ij} \sim 0$ instead, and hence the values of $y_{ij} = 0$ are disregarded when the factor matrices, which are parameters of the CP distribution, are estimated. In fact, if $z_{ij} = 1$, information from the i, j elements becomes weak in the f_{im} and a_{jm} update rules (see (4.35) and (4.37)). A realistic interpretation of this result would be the following: customers who bought “Wine” items but not “Beef” would have bought “Beef” if the customers had conformed to the estimated buying model, but the customers did not actually buy “Beef” for other reasons. Therefore, the advantage of NMF with a zero-inflated model is that it can be usefully applied to a recommender system: “Beef” items are recommended to customers who bought “Wine” items but not “Beef.”

CP2NMF vs CP2ONMF

The effect of the orthogonal constraint is obvious in Fig. 9.11: each of the product classifications has only one non-zero value among the five bases. Although the results obtained for CP2ONMF are simple and easy to comprehend owing to this effect, it is difficult to interpret each of the bases. None of the bases match Japanese food culture except for the 2nd and 3rd bases: the 2nd basis indicates purchases of delicatessen and drink items, (e.g., “Sushi,” “Fried food deli,” “Precooked foods deli,” “Japanese-style deli,” and “Refreshing beverage”), whereas the 3rd basis indicates purchases of basic foods in Japan (e.g., “Fruit & vegetables,” “Milk product,” and “Mizumono”)

Summary of the four methods

We cannot determine which of the four methods is the best. However, we can give suggestions as to which method is better in some situations. If we wish to obtain factors that are affected by extremely large values of small samples such as “Gifts and brand-name confectionery,” “Sushi,” and “Beer,” it is best to use N2NMF. On the other hand, if we consider these values to be outliers, we should use CP2NMF. For zero-inflation, if we place importance on an approximation between the data and factorized matrices, we should use ZICP2NMF. If we want to simplify the result, it is best to use CP2ONMF.

The result obtained from the point-of-sale data show that the factors estimated by ZICP2NMF seem to be better from a Japanese food culture’s point of view because there are many meaningful factors: a complete meal, side dishes, basic cooking ingredients, or foods in Japan, and beef and wine. However, CP2ONMF seems to perform worse on this data because some factors are ambiguous. As described in Section 3.2, CP2ONMF does not approximate the data as well as other methods. These ambiguous factors could have been estimated because of the bad approximation.

Chapter 10

Conclusions

In this paper, we described properties, derivations of updating rules, algorithms, and examples of the usage of various NMFs for exploratory data analysis using a nonnegative data matrix. Nonnegative data matrices are widely and readily available in many academic and business fields, and NMF has been a very useful technique for generating knowledge from these matrices. However, analysis of these data with NMF encounters some difficulties when the nonnegative matrix contains many zero values and has some outliers. The presence of many zero values leads to a poor approximation to the matrix by NMF, whereas outliers result in the estimated factor bases becoming invaded by them. We addressed these problems by considering the assumption of a divergence, which is an error criteria between a given data matrix and its approximation matrix. Various divergences have previously been proposed to ensure robust NMF, e.g., α-divergence and β-divergence. Our research focused on NMF with β-divergence because it allows the use of a zero-inflated model. The assumption on which β-divergence is based is equivalent to the Tweedie distribution assumption, and Tweedie distribution with $\beta \in (0, 1)$, that is, CP distribution, corresponds well to the zero-inflated model because the distribution has mass at the zero-like Poisson distribution, which is also used in combination to the zero-inflated model. In addition, CP distribution can be given as the distribution followed by the sum of gamma-distributed random variables when the number of these variables is Poisson distributed; this generating model is suitable for data consisting of the sum of nonnegative values. Based on the above idea, we proposed the two-factor NMF based on ZICP distribution (ZICP2NMF). We used a simulation study involving three-factor orthogonal NMF to show that CP distribution is robust against outliers by applying our approach to point-of-sale data. Moreover, we used another simulation study to present the goodness of fit of ZICP2NMF to a zero-inflated nonnegative data matrix and presented the characteristics of ZICP2NMF by application to point-of-sale data. We also considered the use of an orthogonal constraint for the simple interpretation of the factor matrices of NMF. The combination of orthogonality and a nonnegativity constraint leads to factor matrices with a simple structure at the risk of poor approximation accuracy as shown in the simulation study about NMF with and without an orthogonal constraint. We discussed a simple structure property of such a constraint factor matrix, and from this property, we derived a new updating rule for nonnegative factor matrices with an orthogonal constraint in NMF based on Poisson and CP distribution with reference to the work of Pompili et al. (2014). Previous algorithms of orthogonal NMF were problematic in terms of the estimation of orthogonal factor matrices: no non-increasing property of the sequence of the objective function values and no orthogonality property of the estimated factor matrices with an orthogonal constraint. On the other hand, the orthogonal NMFs presented in this paper do not have these problems. Our simulation study and application to document-term data demonstrated improved accuracy for the new orthogonal NMF compared to existing NMF. Moreover, most previous orthogonal NMFs are based on a normal distribution and orthogonal NMFs with CP distribution do not exist. Hence, orthogonal NMF based on CP distribution, as proposed in this paper, namely CP2ONMF, is a valuable contribution. Of course, CP2ONMF is extended to NMF with a zero-inflated model: we introduced it as ZICP2ONMF. These ideas described above are applicable to three-factor NMF (see CP3NMF, CP3ONMF, ZICP3NMF, and ZICP3ONMF) in which the data matrix is decomposed into three types of factors: factors of row objects, factors of column objects, and factors that represent the relationship between the factors of row and column objects. Three-factor NMF is also referred to as a bi-clustering method because different factors are assumed for the row and column objects. ZICP3NMF, CP3ONMF, and ZICP3ONMF are also valuable because previous three-factor NMFs cannot be extended to using CP distribution, an orthogonal constraint, and a zero-inflated model due to the highly complicated derivation of the updating algorithm of previous three-factor NMFs.

The drawbacks of our proposed methods are as follows. NMF with orthogonal constraints may lead to a bad approximation of the data matrix. If the approximation is worse than that of non-orthogonal NMF, it is best to interpret the result of the non-orthogonal NMF. It is a future task to develop a method to determine which is better. NMF based on the CP distribution has a problem with the hyperparameter settings. Parameter β of the CP distribution affects the shape of the distribution, and hence we should not determine it using the maximum likelihood method. It seems that there might be a better solution using the marginal likelihood, also referred to as evidence in Bayesian statistics . Zero-inflated NMF also has a drawbacks: the impact of the zero elements on the estimation of the factor matrices is weak. If many zero-elements in the data matrix are truly zero, in other words, if they are generated from the factorization model $y \sim CP(x, \phi, \beta)$ in (3.40), we cannot obtain a true estimate of the factor matrices. We should compare the zero-inflated NMF with non-zero-inflated NMF results using a measure of approximation such as log likelihood.

One of the open problems is a model order determination: the way of determining the number of factors in NMF using a given data matrix. If prior knowledge exists about the number of factors in a given data matrix, a model order determination is not necessary. However, if no prior knowledge of the rank is available, it must be estimated using available data. This estimation is performed to simplify the calculation. Although a large order model leads to good approximations, it is difficult to interpret the estimated factor matrices. Model order determination is one of the main challenges in NMF, and several methods have been proposed for handling it in past work (Owen and Perry, 2009; Schmidt et al., 2009; Tan and Févotte, 2013). The development of a model order determination method in NMF based on a zero-inflated model or with an orthogonal constraint presents a new challenge. Especially, an extension to a Bayesian model seems to be useful because we can check the appropriate model order as well as the significance of each of the elements of factor matrices. Another open problem is an extension to tensor factorization. NMFs introduced in this paper can be easily extended to multi-array nonnegative data in the same fashion as in Cichocki et al. (2007) or Cichocki et al. (2009). Nonnegative tensor data can be easily obtained from data resulting from human behavior such as point-of-sale data, and such tensor data may contain a large number of zero values due to the shortage of combinations among the three types of objects representing each of the arrays of the tensor. Hence, a zero-inflated model would be available for such tensor data.

Acknowledgements

I am deeply grateful to Prof. H. Yadohisa for his guidance and persistent help. Without his encouragement, this thesis would not have materialized. Prof. K. Kawasaki, Prof. K. Adachi, Prof. T. Yano, and Prof. M. Jin give insightful comments and suggestions, which have helped us significantly improve the thesis. I am grateful to IDS Co., Ltd. and the Joint Association Study Group of Management Science for providing the data used in our application. I would like to thank Yadohisa's laboratory members, especially J. Tsuchida and M. Takagishi for useful discussions. Finally, I would like to express the appreciation to my family.

References

Abe, H. and Yadohisa, H. (2016). A non-negative matrix factorization model based on the zero-inflated Tweedie distribution. *Computational Statistics*. doi:`10.1007/s00180-016-0689-8`.

Badea, L. (2008). Extracting gene expression profiles common to colon and pancreatic adenocarcinoma using simultaneous nonnegative matrix factorization. *Pacific Symposium on Biocomputing*. **290**. (13). Citeseer, pp. 279–290.

Banerjee, A. et al. (2003). Generative model-based clustering of directional data. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 19–28.

Basu, A. et al. (1998). Robust and efficient estimation by minimising a density power divergence. *Biometrika* **85** (3), pp. 549–559.

Boley, D. (1998). *Hierarchical taxonomies using divisive partitioning*. Tech. rep. Technical Report TR-98-012, Department of Computer Science, University of Minnesota, Minneapolis.

Boley, D. et al. (1999). Document categorization and query generation on the world wide web using webace. *Artificial Intelligence Review* **13** (5-6), pp. 365–391.

Brunet, J.-P. et al. (2004). Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the national academy of sciences* **101** (12), pp. 4164–4169.

Byrd, R. H. et al. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* **16** (5), pp. 1190–1208.

Chen, G. et al. (2009). Collaborative filtering using orthogonal nonnegative matrix tri-factorization. *Information Processing & Management* **45** (3), pp. 368–379.

Chernoff, H. (1952). A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pp. 493–507.

Choi, S. (2008). Algorithms for orthogonal nonnegative matrix factorization. *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, pp. 1828–1832.

Cichocki, A. and Amari, S. (2010). Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. *Entropy* **12** (6), pp. 1532–1568.

Cichocki, A. et al. (2007). Non-negative tensor factorization using alpha and beta divergences. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*. **3**. IEEE, pp. III–1393.

Cichocki, A. et al. (2008). Non-negative matrix factorization with α-divergence. *Pattern Recognition Letters* **29** (9), pp. 1433–1440.

Cichocki, A. et al. (2009). *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation.* John Wiley & Sons.

Costa, G. and Ortale, R. (2014). XML Document Co-clustering via Non-negative Matrix Tri-factorization. *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on.* IEEE, pp. 607–614.

Dempster, A. P. et al. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38.

Ding, C. et al. (2006). Orthogonal nonnegative matrix tri-factorizations for clustering. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, pp. 126–135.

Dunn, P. K. and Smyth, G. K. (2001). Tweedie family densities: methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling, Odense, Denmark*, pp. 2–6.

Févotte, C. and Idier, J. (2011). Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation* **23** (9), pp. 2421–2456.

Févotte, C. et al. (2009). Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. *Neural computation* **21** (3), pp. 793–830.

Gaujoux, R. and Seoighe, C. (2010). A flexible R package for nonnegative matrix factorization. *BMC bioinformatics* **11** (1), p. 1.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification* **2** (1), pp. 193–218.

Itakura, F. and Saito, S. (1968). Analysis synthesis telephony based on the maximum likelihood method. *Proceedings of the 6th International Congress on Acoustics.* **17**. pp. C17–C20, pp. C17–C20.

Jorgensen, B. (1997). *The theory of dispersion models.* CRC Press.

Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. PhD thesis. Masterfs thesis, Dept. of Mathematics, Univ. of Chicago.

Kim, H. and Park, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics* **23** (12), pp. 1495–1502.

Kim, Y.-D. and Choi, S. (2007). Nonnegative tucker decomposition. *2007 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, pp. 1–8.

Kim, Y.-D. et al. (2008). Nonnegative Tucker decomposition with alpha-divergence. *2008 IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, pp. 1829–1832.

Kim, Y. et al. (2011). Principal network analysis: identification of subnetworks representing major dynamics using gene expression data. *Bioinformatics* **27** (3), pp. 391–398.

Kimura, K. et al. (2014). A Fast Hierarchical Alternating Least Squares Algorithm for Orthogonal Nonnegative Matrix Factorization. *ACML.*

Kompass, R. (2007). A generalized divergence measure for nonnegative matrix factorization. *Neural computation* **19** (3), pp. 780–791.

Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probabilities*. University of California Press, pp. 481–492.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics* **22** (1), pp. 79–86.

Lagrange, J. L. de (1788). Méchanique analitique. *Paris: Desaint, 1788; 512 p.; in 8.; DCC. 4.403* **1**.

Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* **34** (1), pp. 1–14.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature* **401** (6755), pp. 788–791.

Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, pp. 556–562.

Li, Z. et al. (2010). Nonnegative matrix factorization on orthogonal subspace. *Pattern Recognition Letters* **31** (9), pp. 905–911.

Mauthner, T. et al. (2010). Efficient object detection using orthogonal NMF descriptor hierarchies. *Pattern Recognition*. Springer, pp. 212–221.

McLachlan, G. and Krishnan, T. (2007). *The EM algorithm and extensions*. Vol. 382. John Wiley & Sons.

Mirzal, A. (2014). A convergent algorithm for orthogonal nonnegative matrix factorization. *Journal of Computational and Applied Mathematics* **260**, pp. 149–166.

Nakano, M. et al. (2010). Convergence-guaranteed multiplicative algorithms for nonnegative matrix factorization with β-divergence. *choice* **10**, p. 1.

Owen, A. B. and Perry, P. O. (2009). Bi-cross-validation of the SVD and the nonnegative matrix factorization. *The annals of applied statistics*, pp. 564–594.

Pascual-Montano, A. et al. (2006). Nonsmooth nonnegative matrix factorization (nsNMF). *IEEE transactions on pattern analysis and machine intelligence* **28** (3), pp. 403–415.

Pompili, F. et al. (2014). Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing* **141**, pp. 15–25.

Schmidt, M. N. et al. (2009). Bayesian non-negative matrix factorization. *International Conference on Independent Component Analysis and Signal Separation*. Springer, pp. 540–547.

Simchowitz, M. (2013). Zero-Inflated Poisson Factorization for Recommendation Systems. `https://www.academia.edu/6256225/Zero-Inflated_Poisson_Factorization_for_Recommendation_Systems`.

Şimşekli, U. et al. (2013). Learning the beta-Divergence in Tweedie Compound Poisson Matrix Factorization Models. *ICML (3)*, pp. 1409–1417.

Tan, V. Y. and Févotte, C. (2013). Automatic relevance determination in nonnegative matrix factorization with the β-divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35** (7), pp. 1592–1605.

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika* **31** (3), pp. 279–311.

Wang, F. et al. (2016). Orthogonal Nonnegative Matrix Factorization Based Local Hidden Markov Model for Multimode Process Monitoring. *Chinese Journal of Chemical Engineering*.

Wang, G. et al. (2006). LS-NMF: a modified non-negative matrix factorization algorithm utilizing uncertainty estimates. *BMC bioinformatics* **7** (1), p. 1.

Yoo, J. and Choi, S. (2010a). Nonnegative matrix factorization with orthogonality constraints. *Journal of computing science and engineering* **4** (2), pp. 97–109.

Yoo, J. and Choi, S. (2008). Orthogonal nonnegative matrix factorization: Multiplicative updates on Stiefel manifolds. *Intelligent Data Engineering and Automated Learning–IDEAL 2008*. Springer, pp. 140–147.

Yoo, J. and Choi, S. (2009). Probabilistic matrix tri-factorization. *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on.* IEEE, pp. 1553–1556.

Yoo, J. and Choi, S. (2010b). Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds. *Information processing & management* **46** (5), pp. 559–570.

Zhang, J. et al. (2008). Pattern expression nonnegative matrix factorization: algorithm and applications to blind source separation. *Computational intelligence and neuroscience* **2008**.

Zhang, Y. (2013). Likelihood-based and bayesian methods for tweedie compound poisson linear mixed models. *Statistics and Computing* **23** (6), pp. 743–757.