AUTOMATIC RELEVANCE DETERMINATION IN NONNEGATIVE MATRIX FACTORIZATION BASED ON A ZERO-INFLATED COMPOUND POISSON-GAMMA DISTRIBUTION

Hiroyasu Abe* and Hiroshi Yadohisa†

ABSTRACT

In this paper, we consider the determination of the number of factors in nonnegative matrix factorization (NMF) for a zero-inflated data matrix. This zero-inflated case leads to poor approximation to the nonnegative data matrix. To address this problem, we use the zero-inflated compound Poisson-gamma distribution as the error distribution in NMF. In addition, we consider automatic relevance determination (ARD) for model order selection. Our simulation study shows that our method is better than the basic ARD method for zero-inflated data. We apply our proposed method to real-world purchasing data to determine the number of buying patterns.

1. Introduction

Nonnegative matrix factorization (NMF) is a technique used for approximating a given data matrix that consists of nonnegative elements by a product of two factor matrices consisting of nonnegative elements (Lee and Seung, 1999, 2001; Berry et al., 2007; Wang and Zhang, 2013). The rank of both factor matrices i.e., the number of factors of each, might be known prior to factorization. However, if no prior knowledge of rank is given, it must be estimated using available data. Though a large order model leads to good approximations, it may lead to overfitting, which is inappropriate from the prediction point of view; that is, the number of factors must be chosen considering this trade-off. Model order selection 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). In this paper, we consider automatic relevance determination (ARD), which is based on a Bayesian model, for the model order selection proposed by Tan and Févotte (2013). There are two advantages of the ARD method: first, there are few hyperparameters that must be decided in advance; second, its computation is relatively fast because the number of factors and the factor matrices are estimated simultaneously.

Another issue in NMF is related to error distribution. One of the commonly used distributions for NMF is the Tweedie distribution (Févotte and Idier, 2011; Nakano et al., 2010), which Tan and Févotte (2013) use in ARD. The Tweedie distribution has an index parameter $\beta$ that determines the shape of its density or probability function, and is a generalized distribution that includes normal ($\beta = 2$), Poisson ($\beta = 1$), gamma ($\beta \to 0$), and compound Poisson-gamma (CP) distributions ($0 < \beta < 1$) (Jorgensen, 1997; Dunn and Smyth, 2001;...

*Graduate School of Culture and Information Science, Doshisha University, 1-3, Tatara Miyakodani, Kyotanabe-shi, Kyoto-fu 610-0394, Japan E-mail: hiro00012@gmail.com
†Faculty of Culture and Information Science, Doshisha University, 1-3, Tatara Miyakodani, Kyotanabe-shi, Kyoto-fu 610-0394, Japan

Key words: model order selection; count data; EM algorithm; Tweedie distribution
Cichocki and Amari, 2010). Log-likelihood maximization under the Tweedie distribution is interpreted as the minimization of the $\beta$-divergence (Basu et al., 1998), where the parameter $\beta$ corresponds to that of the Tweedie distribution. $\beta$-divergence is a generalized divergence that includes Euclidean distance, Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951), and Itakura-Saito (IS) divergence (Itakura and Saito, 1968). Table 1 shows the relationship between the probability distributions and divergence. It is known that the index parameter $\beta$ affects the robustness of parameter estimation (Cichocki and Amari, 2010). Although there are several approaches to determining the value of $\beta$ (Zhang, 2013), it is a considerably difficult and complicated problem, and selecting an error distribution in NMF is often based on prior knowledge about the given data. For example, for count data, it is appropriate to use a Poisson distribution as the error distribution. Our interest for this study revolves around data that are derived from a gross summation of nonnegative values, examples of which include rainfall, insurance, and purchasing data. Such data are compatible with a CP distribution, which, as the name suggests, is a Poisson mixture of gamma distributions. That is, its random variable is the sum of $n$ identically gamma distributed random variables, where $n$ is a Poisson distributed random variable.

Table 1: The relationship among divergences, probability distributions, and values of $\beta$.

| Divergence               | Probability distribution | The value of $\beta$               |
|--------------------------|--------------------------|------------------------------------|
| Euclidean distance       | Normal                   | $\beta = 2$                        |
| KL divergence            | Poisson                  | $\beta = 1$                        |
| Itakura-Saito divergence | Gamma (exponential)      | $\beta \to 0$                      |
| $\beta$-divergence       | CP                       | $0 < \beta < 1$                    |
|                          |                          | $-\infty < \beta < \infty$         |
| $\beta$-divergence       | Tweedie                  | (In case of Tweedie: $\beta \in (-\infty, 1] \cup [2, \infty)$) |

When the nonnegative data matrix is given as a two-way table consisting of a count (like a contingency table) or a gross summation of the nonnegative values of a pair of objects in two sets, its matrix tends to contain many values of 0. This zero-inflated case leads to poor approximation to the nonnegative data matrix (Abe and Yadohisa, 2016, pp.6). For better approximation in the case of a zero-inflated data matrix, Abe and Yadohisa (2016) proposed a method that uses the zero-inflated compound Poisson-gamma (ZICP) distribution in NMF; the ZICP distribution is a combination of the CP distribution and the distribution that takes the value 0 with probability 1. The ZICP distribution is similar to the zero-inflated Poisson (ZIP) distribution (Lambert, 1992). In this study, we extend our proposed method to enable model order selection with an ARD approach. Further, we show a numerical example comparing the consistency of model order selection of the proposed ARD method with the basic ARD method for a zero-inflated nonnegative data matrix. Moreover, we apply our method to a point-of-sale (POS) dataset to detect the number of purchase patterns.
2. Notation

In this section, we introduce the notation used in this paper. We use bold uppercase letters, e.g., $M$, to denote a matrix, and the lowercase, e.g., $m_{ij}$, for its $i, j$ element. Further, we use $m_i$ and $m(j)$ as the $i$th row vector and $j$th column vector of $M$, respectively. We use the prime symbol to express a transpose matrix, e.g., $M'$ is the transpose of $M$. The Euclidean norm of a matrix or vector is represented as $\|M\| = \sqrt{\text{tr}(M'M)}$, where $\text{tr}(M)$ denotes the trace of $M$. We use $\odot$ as the Hadamard product. The element-wise quotient of two matrices is denoted by fraction notation; e.g., $M/N$ is the element-wise quotient of $M$ and $N$. $M^\beta$ is the element-wise $\beta$ power of the matrix $M$. Finally, $\mathbb{R}^{n \times p}_{+}$ is a set of $n \times p$ matrices consisting of nonnegative elements.

3. Automatic relevance determination in NMF based on zero-inflated compound Poisson-gamma distribution (ARDNMF-ZICP)

In this section, we describe the details of our proposed method. First, we define the likelihood and the prior of the parameters. Then, we present the objective function with respect to these parameters and update rules for parameters that should be estimated. Finally, we present the ARDNMF-ZICP algorithm.

3.1. Log-likelihood

Let $Y \in \mathbb{R}_{+}^{n \times p}$ be a given nonnegative data matrix, and $F \in \mathbb{R}_{+}^{n \times k}$ and $A \in \mathbb{R}_{+}^{p \times k}$ be nonnegative factor matrices. $n$ and $p$ are the number of observations and number of measured features, respectively. $k$ is the number of factors or bases, and it follows the relation $k < \min(n, p)$. We assume that the rank of $F$ and $A$ is $k$. Our aim is to obtain estimates of $F$ and $A$ such that $Y \approx X := FA'$ where "\approx" is a term representing approximation. Here, we assume that all elements of $Y$, i.e., $y_{ij} (i = 1, \ldots, n; j = 1, \ldots, p)$, are conditionally independent ZICP distributed random variables given $F$ and $A$, that is:

$$
\begin{cases}
  y_{ij} \sim 0 & \text{with probability } w \\
  y_{ij} \sim CP(x_{ij}, \phi, \beta) & \text{with probability } 1 - w,
\end{cases}
$$

where $CP(x, \phi, \beta)$ is the CP distribution (Jorgensen, 1997, pp.140–144; Dunn and Smyth, 2001, pp.1–2; Cichocki and Amari, 2010, pp.1545), $w \in (0, 1)$ is a mixture ratio, and $x_{ij} := f_i^\prime a_j$. The probability density function of $CP(x, \phi, \beta)$ is as follows:

$$
g_{CP}(y|x, \phi, \beta) = \begin{cases}
  0 & \text{if } (y < 0) \\
  \exp\left(-\frac{x^\beta}{\phi \beta}\right) & \text{if } (y = 0) \\
  h(y, \phi, \beta) \exp\left\{\frac{1}{\phi} \left(\frac{y x^{\beta-1}}{\beta-1} - \frac{x^\beta}{\beta}\right)\right\} & \text{if } (y > 0)
\end{cases}
$$

The parameters $\phi \in (0, \infty)$ and $\beta \in (0, 1)$ are referred to as dispersion and index parameters, respectively. $h(y, \phi, \beta)$ is the normalizing constant of the CP distribution and is difficult to describe analytically. The mean and variance of the CP distributed random variable are $x$ and $\phi x^{2-\beta}$, respectively.

Then, the likelihood with respect to parameters $\theta = \{F, A, w\}$ is:
\[ L(\theta|Y) := g_Y(Y|\theta) \]
\[ = \prod_{i=1}^{n} \prod_{j=1}^{p} g_{y_{ij}}(y_{ij}|\theta) \]
\[ = \prod_{i=1}^{n} \prod_{j=1}^{p} \{wI(y_{ij} = 0) + (1 - w)g_{CP}(y_{ij}|x_{ij}, \phi, \beta)\}, \quad (1) \]

where \( I(\cdot) \) is an indicator function.

### 3.2. Prior

We assume that the parameters \( f_{im} \) \((i = 1, \ldots, n; m = 1, \ldots, k)\), \( a_{jm} \) \((j = 1, \ldots, p; m = 1, \ldots, k)\), and \( w \) are conditionally independent random variables given parameters \( \lambda = (\lambda_1, \ldots, \lambda_k) \). The prior distributions of these parameters are:

\[ f_{im} \sim HN(\lambda_m) \quad (2) \]
\[ a_{jm} \sim HN(\lambda_m) \quad (3) \]
\[ w \sim Beta(c, d) \quad (4) \]

where \( HN(\lambda) \) is the half-normal distribution (Daniel, 1959, pp.313) with the following probability density function:

\[ g_{HN}(y|\lambda) = \begin{cases} \sqrt{\frac{2}{\pi\lambda}} \exp\left(-\frac{y^2}{2\lambda}\right) & (y \geq 0) \\ 0 & (y < 0) \end{cases}, \]

and \( Beta(c, d) \) is the beta distribution (Bishop, 2006, pp.71) with the following probability density function:

\[ g_{Beta}(y|c, d) = \begin{cases} \frac{\Gamma(c + d)}{\Gamma(c)\Gamma(d)} y^{c-1}(1 - y)^{d-1} & (0 \leq y \leq 1) \\ 0 & (\text{others}) \end{cases}, \]

where \( \Gamma(\cdot) \) is the gamma function. From (2), (3), and (4), we can state that \( f_{im} \geq 0 \), \( a_{jm} \geq 0 \), and \( w \in (0, 1) \).

In addition, we assume that \( \lambda_m \) \((m = 1, \ldots, k)\) are identically and independently inverse-gamma distributed random variables; that is,

\[ \lambda_m \sim IG(u, v). \quad (5) \]

The probability density function of the inverse-gamma distribution (Bishop, 2006, pp.71) is defined as follows:

\[ g_{IG}(y|u, v) = \begin{cases} \frac{v^u}{\Gamma(u)} y^{-(u+1)} \exp\left(-\frac{v}{y}\right) & (y \geq 0) \\ 0 & (y < 0) \end{cases}. \]

It is noteworthy that the aforementioned priors for \( f_{im}, a_{jm}, \) and \( \lambda_m \) are the same as those defined by Tan and Févotte (2013). They call the ARD with this prior setting \( \ell_2 \)-ARD; the half-normal distribution assumption corresponds to the \( \ell_2 \) norm regularization.
The prior for \( w \) is a well-known natural conjugate prior for the Bernoulli distribution. As mentioned later, \( w \) is a parameter possessed by independent and identically distributed Bernoulli random variables \( z_{ij} \) \((i = 1, \ldots, n; j = 1, \ldots, p)\).

From (2), (3), (4), and (5), the prior of parameters \( \theta \) and \( \lambda \) is as follows:

\[
g_{\theta, \lambda}(\theta, \lambda) := n \prod_{i=1}^{n} p \prod_{j=1}^{p} k \prod_{m=1}^{k} \left( g_{HN}(f_{im} | \lambda_m) g_{HN}(a_{jm} | \lambda_m) g_{IG}(\lambda_m | u, v) g_{Beta}(w | c, d) \right) \tag{6}
\]

3.3. Objective function

Considering (1) and (6), the objective function for \( F, A, \phi, w, \) and \( \lambda \) is obtained as follows:

\[
Q_{\text{incomp}}(F, A, \phi, w, \lambda | \beta, u, v, c, d) := - \log \{ L(\theta | Y) g_{\theta, \lambda}(\theta, \lambda) \}
\]

\[
= - n \sum_{i=1}^{n} p \sum_{j=1}^{p} \log \left[ w I(y_{ij} = 0) + (1 - w) h(y_{ij}, \phi, \beta) \exp \left\{ \frac{1}{\phi} \left( \frac{y_{ij} x_{ij}^{\beta - 1}}{\beta - 1} - \frac{x_{ij}^{\beta}}{\beta} \right) \right\} \right]
+ k \sum_{m=1}^{k} \frac{1}{\lambda_m} \left( \frac{\| f_m \|^2}{2} + \frac{\| a_m \|^2}{2} + v \right) + \left( \frac{n + p}{2} + u + 1 \right) \sum_{m=1}^{k} \log(\lambda_m)
- (c - 1) \log(w) - (d - 1) \log(1 - w) + \text{const.} \tag{7}
\]

Here, "const." denotes terms that are independent of the parameters to be optimized.

We must obtain the optimal parameters that minimize this objective function. Since we cannot derive the global solution in this problem, we derive an update rule for each parameter to at least decrease the objective function given the other parameters, and develop an iterative algorithm to optimize the objective function, alternating between update rules.

We are not directly using this objective function to derive the update rules for \( F, A, \phi, \) and \( w \) because the log-likelihood term, the second line in (7), is difficult to differentiate with respect to these parameters. Instead of using the likelihood \( g(Y | \theta) \) in (1), we use the complete log-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

\[
z_{ij} = \begin{cases} 
1 & \text{if } y_{ij} \sim 0 \\
0 & \text{if } y_{ij} \sim CP(x_{ij}, \phi, \beta), 
\end{cases} \tag{8}
\]

and assume that \( z_{ij} \) is an identically and independently Bernoulli distributed random variable, that is,

\[
z_{ij} \sim Be(w). \tag{9}
\]

Here, the probability function of \( Be(w) \) is defined as follows:

\[
g_{Be}(y | w) = \begin{cases} 
w^y (1 - w)^{1-y} & (y = 0, 1) \\
0 & (\text{others}) \end{cases}
\]

We also define the conditional distribution of \( y_{ij} \) given \( z_{ij} \) from (8) as follows:

\[
g_{y_{ij} | z_{ij}, f_i, a_j}(y_{ij} | z_{ij}, f_i, a_j) = \{ I(y_{ij} = 0) \}^{z_{ij}} \{ g_{CP}(y_{ij} | x_{ij}, \phi, \beta) \}^{1 - z_{ij}}.
\]

---

[33]
From (9) and the above result, the complete likelihood is defined as the joint distribution of \( Y \) and \( Z \) as follows:

\[
L(\theta \mid Y, Z) = g_{Y, Z}(Y, Z \mid \theta)
\]

\[
= \prod_{i=1}^{n} \prod_{j=1}^{p} g_{y_{ij}, z_{ij}}(y_{ij}, z_{ij} \mid \theta)
\]

\[
= \prod_{i=1}^{n} \prod_{j=1}^{p} g_{y_{ij} \mid z_{ij}}(y_{ij} \mid z_{ij}, f_i, a_j) g_{Be}(z_{ij} \mid w)
\]

\[
= \prod_{i=1}^{n} \prod_{j=1}^{p} (wI(y_{ij} = 0))^{z_{ij}} \{ (1 - w) g_{CP}(y_{ij} \mid x_{ij}, \phi, \beta) \}^{1 - z_{ij}}.
\]

Then, the new objective function is defined using a logarithm of \( L(\theta \mid Y, Z) \), instead of the second line in (7), as follows:

\[
Q_{\text{comp}}(F, A, \phi, w, \lambda \mid \hat{Z}, \beta, u, v, c, d) := - \log \left\{ L(\theta \mid Y, \hat{Z}) g_{\theta, \lambda}(\theta, \lambda) \right\}
\]

\[
= - \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij} \log(w) + (1 - \hat{z}_{ij}) \left\{ \log(1 - w) + \log(h(y_{ij}, \phi, \beta)) + \frac{1}{\phi} \left( \frac{y_{ij} x_{ij}^{\beta - 1}}{\beta - 1} - \frac{x_{ij}^{\beta}}{\beta} \right) \right\}
\]

\[
+ \sum_{m=1}^{k} \frac{1}{\lambda_m} \left( \frac{\|f_{(m)}\|^2}{2} + \frac{\|a_{(m)}\|^2}{2} + v \right) + \left( \frac{n + p}{2} + u + 1 \right) \sum_{m=1}^{k} \log(\lambda_m)
\]

\[
- (c - 1) \log(w) - (d - 1) \log(1 - w) + \text{const},
\]

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

\[
\hat{z}_{ij} = E[z_{ij} \mid y_{ij}]
\]

\[
= \sum_{z_{ij} \in \{0, 1\}} z_{ij} g_{y_{ij}, z_{ij}}(y_{ij}, z_{ij} \mid \theta)
\]

\[
= \frac{wI(y_{ij} = 0)}{wI(y_{ij} = 0) + (1 - w) g_{CP}(y_{ij} \mid x_{ij}, \phi, \beta)}
\]

\[
= \begin{cases} 
  w + (1 - w) g_{CP}(0 \mid x_{ij}, \phi, \beta) & \text{if } y_{ij} = 0 \\
  0 & \text{if } y_{ij} \neq 0.
\end{cases}
\]

The above equation is known as the update rule of Estep in the EM algorithm. Note that \( 0 \leq \hat{z}_{ij} \leq 1 \).

3.4. Update rules

In this section, we present and discuss the update rules for \( F, A, \phi, w, \) and \( \lambda \).
Update rule for $F$

From (10), the objective function with respect to $F$ is as follows:

$$Q_F(F) = -\frac{1}{\phi} \sum_{i=1}^{n} \sum_{j=1}^{p} \left( \frac{z_{ij}^{*} y_{ij} (f'(a_j))^\beta - 1}{\beta - 1} - \frac{z_{ij}^{*} (f'(a_j))^\beta}{\beta} \right) + \frac{1}{2} \sum_{m} \frac{\|f_{(m)}\|^2}{2\lambda_m},$$  \hspace{1cm} (12)

where $\hat{z}_{ij}^{*} := 1 - \hat{z}_{ij}$. This objective function is similar to that of Tan and Févote (2013); however, the weight value in this case is $\hat{z}_{ij}^{*}$. Since $\hat{z}_{ij}$ is positive, we can use Tan and Févote’s auxiliary function technique and derive the update rule for $F$ as follows:

$$f_{im} = f_{im}^{*} \left\{ \begin{array}{l}
\sum_{j=1}^{p} \hat{z}_{ij}^{*} y_{ij} \left( \sum_{\ell=1}^{k} f_{i\ell}^{*} a_{j\ell} \right)^{\beta - 2} \\
\sum_{j=1}^{p} \hat{z}_{ij}^{*} \left( \sum_{\ell=1}^{k} f_{i\ell}^{*} a_{j\ell} \right)^{\beta - 1} a_{jm} + \frac{\phi f_{im}^{*}}{\lambda_m} \end{array} \right\}^{\frac{1}{\beta - 1}},$$  \hspace{1cm} (13)

where $f_{im}^{*}$ is the current value of $f_{im}$ in the algorithm. As shown in update rule (13), a small value of $z_{ij}^{*}$ reduces the impact of the $i, j$ element. The matrix form of this update rule is as follows:

$$F = F^{*} \odot \left\{ \begin{array}{l}
\hat{Z}^{*} \odot Y \odot (F^{*} A')^{\beta - 2} A' \\
\hat{Z}^{*} \odot (F^{*} A')^{\beta - 1} A + \frac{\phi F^{*}}{\Lambda F} \end{array} \right\}^{\frac{1}{\beta - 1}},$$

where $\Lambda F$ is an $n \times k$ matrix that contains $n$ row vectors $\lambda'$.

Update rule for $A$

The minimization of the objective function with respect to $A$ takes the same form as (12); in addition, we can obtain this update rule in a manner similar to that of $F$:

$$a_{jm} = a_{jm}^{*} \left\{ \begin{array}{l}
\sum_{i=1}^{n} \hat{z}_{ij}^{*} y_{ij} f_{im} \left( \sum_{\ell=1}^{k} f_{i\ell} a_{j\ell}^{*} \right)^{\beta - 2} \\
\sum_{i=1}^{n} \hat{z}_{ij}^{*} \left( \sum_{\ell=1}^{k} f_{i\ell} a_{j\ell}^{*} \right)^{\beta - 1} f_{im} + \frac{\phi a_{jm}^{*}}{\lambda_m} \end{array} \right\}^{\frac{1}{\beta - 1}},$$  \hspace{1cm} (14)

where $a_{jm}^{*}$ is the current value of $a_{jm}$ in the algorithm. The matrix form of this update rule is

$$A = A^{*} \odot \left\{ \begin{array}{l}
\hat{Z}^{*} \odot Y \odot (FA')^{\beta - 2} F' \\
\hat{Z}^{*} \odot (FA')^{\beta - 1} F' + \frac{\phi A^{*}}{\Lambda A} \end{array} \right\}^{\frac{1}{\beta - 1}},$$

where $\Lambda A$ is a $p \times k$ matrix that contains $p$ row vectors $\lambda'$.

Update rule for $\lambda$

From (10), the objective function with respect to $\lambda$ is as follows:
\[ Q_\lambda(\lambda) = \sum_{m=1}^{k} \frac{1}{\lambda_m} \left( \frac{\|f_{(m)}\|^2}{2} + \frac{\|a_{(m)}\|^2}{2} + v \right) + \left( \frac{n+p}{2} + u + 1 \right) \sum_{m=1}^{k} \log(\lambda_m). \]

This is of the same form as in Tan and Févotte (2013), and thus, the optimal \( \lambda_m \) can be obtained as follows:

\[ \lambda_m = \frac{\|f_{(m)}\|^2 + \|a_{(m)}\| + 2v}{n + p + 2(u + 1)}. \]  

(15)

Therefore, we can understand that \( \lambda \) represents the effectiveness of the \( m \)th factor. In Tan and Févotte (2013), \( \lambda_m \) is referred to as the relevance weight. If all elements of the \( m \)th vector of \( F \) and \( A \) become 0, that is, the \( m \)th factor has no meaning, \( \lambda \) reaches a minimum value as follows:

\[ \frac{2v}{n + p + 2(u + 1)}. \]  

(16)

**Update rule for \( w \)**

From (10), the objective function with respect to \( w \) is as follows:

\[ Q_w(w) = -\log(w) \left( \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij} + c - 1 \right) - \log(1 - w) \left( \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij}^* + d - 1 \right). \]

The partial derivative of \( Q_w(w) \) is as follows:

\[ \frac{\partial Q_w(w)}{\partial w} = -\frac{1}{w} \left( \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij} + c - 1 \right) + \frac{1}{1 - w} \left( \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij}^* + d - 1 \right). \]  

(17)

The local minima of \( Q_w(w) \) with respect to \( w \) is obtained by solving the equation \( \frac{\partial Q_w(w)}{\partial w} = 0 \) with respect to \( w \) as follows:

\[ \hat{w} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij} + c - 1}{np + c + d - 2}. \]  

(18)

Here, \( \hat{w} \) must lie on \((0,1)\) because the terms \( \log(w) \) and \( \log(1 - w) \) are contained in \( Q_w(w) \). Now, we derive sufficient conditions

(a) for the downward convexity of \( Q_w(w) \) in \( w \in (0,1) \)

(b) and for \( \hat{w} \) existing in \((0,1)\).

The second partial derivative of \( Q_w(w) \) is as follows:

\[ \frac{\partial^2 Q_w(w)}{\partial w^2} = \frac{1}{w^2} \left( \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij} + c - 1 \right) + \frac{1}{(1 - w)^2} \left( \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij}^* + d - 1 \right). \]

From this, \( \frac{\partial^2 Q_w(w)}{\partial w^2} > 0 \) for all \( w \in (0,1) \) is sufficient when \( \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij} + c - 1 > 0 \) and \( \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij}^* + d - 1 > 0 \). Since \( \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij} > 0 \) and \( \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij}^* > 0 \) from (11), (a)
ARD in NMF Based on a Zero-Inflated Compound Poisson-Gamma Distribution

is sufficient when \( c - 1 \geq 0 \) and \( d - 1 \geq 0 \). (b) is also sufficient when \( c - 1 \geq 0 \) and \( d - 1 \geq 0 \) because \( \hat{w} \) can be rewritten as follows:

\[
\hat{w} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij} + c - 1}{\sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij} + c - 1 + \sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}^*_{ij} + d - 1}.
\]

From the above discussion, (a) and (b) are satisfied when \( c \geq 1 \) and \( d \geq 1 \), and hence, \( \hat{w} \) has only one optimal \( w \). Therefore, the hyperparameters \( c \) and \( d \) should be set to \( c \geq 1 \) and \( d \geq 1 \), respectively. If \( c = 1 \) and \( d = 1 \), that is, we set the non-informative prior to \( w \), the solution of this \( w \) becomes the maximum likelihood estimator as follows:

\[
\hat{w} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{p} \hat{z}_{ij}}{np}.
\]

### Update rule for \( \phi \)

From (10), the objective function with respect to \( \phi \) is as follows:

\[
Q_\phi(\phi) = -\sum_{i=1}^{n} \sum_{j=1}^{p} (1 - \hat{z}_{ij}) \log \{ g_{\text{CP}}(y_{ij}|x_{ij}, \phi, \beta) \}
\]

The optimal \( \phi \) cannot be obtained analytically because of the \( h(y_{ij}, \phi, \beta) \) term. Hence, we use the BFGS quasi-Newton method (Byrd et al., 1995) with constraints \( \phi > 0 \) to obtain the optimal value of \( \phi \).

### 3.5. Algorithm

Using the above update rules, the ARDNMF-ZICP algorithm is presented in Algorithm. The ‘mod’ in Algorithm denotes a modulo operation. Details of the input parameters are as follows:

- \( Y \): \( n \times p \) data matrix
- \( \beta \): index parameter
- \( u \): hyperparameter of the prior of \( \lambda \)
- \( c \): hyperparameter of the prior of \( w \)
- \( d \): hyperparameter of the prior of \( w \)
- \( \delta \): the number of iterations to update dispersion parameter \( \phi \) at the beginning of the algorithm
- \( \eta \): interval to update dispersion parameter \( \phi \)
- \( \tau_{\text{cut}} \): threshold to stop algorithm and to select factors
- \( k \): initial number of bases
- \( F^{(0)} \): initial left-side \( n \times k \) nonnegative factor matrix
- \( A^{(0)} \): initial right-side \( p \times k \) nonnegative factor matrix
- \( \phi^{(0)} \): initial dispersion parameter
- \( w^{(0)} \): initial mixture weight
Algorithm ARD-NMF-ZICP

1: Input $Y, \beta, u, c, d, \delta, \eta, \tau_{cut}, k, F^{(0)}, A^{(0)}, \phi^{(0)}, w^{(0)}$.
2: $\hat{y} \leftarrow \frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} y_{ij}$
3: $v \leftarrow \hat{y}(u - 1)\pi$
4: $\lambda_{m}^{(0)} \leftarrow \frac{\|f^{(0)}(m)\|^2/2 + \|a^{(0)}(q)\|^2/2 + v}{(n+p)/2 + u + 1} (m = 1, \ldots, k)$
5: $z_{ij}^{(0)} \leftarrow \begin{cases} \frac{w^{(0)} + (1 - w^{(0)})g_{CP}(0 | f^{(0)}_{i}a^{(0)}_{j}, \phi^{(0)} , \beta)}{w^{(0)}} & \text{if } y_{ij} = 0 \\ 0 & \text{if } y_{ij} \neq 0. \end{cases}$
6: $t \leftarrow 0$
7: repeat
8: $t \leftarrow t + 1$
9: $F^{(t)} \leftarrow F^{(t-1)} \odot \begin{cases} \left[ Z^{(t-1)} \odot Y \odot \left( F^{(t-1)} A^{(t-1)'} \right)^{\beta-2} \right] A^{(t-1)} & \frac{1}{\beta - \delta} \\ \left[ Z^{(t-1)} \odot \left( F^{(t-1)} A^{(t-1)'} \right)^{\beta-1} \right] A^{(t-1)} + \frac{\phi^{(t-1)} F^{(t-1)}}{A F^{(t-1)}} & \frac{1}{\beta - \delta} \end{cases}$
10: $A^{(t)} \leftarrow A^{(t-1)} \odot \begin{cases} \left[ Z^{(t-1)} \odot Y \odot \left( F^{(t)} A^{(t-1)'} \right)^{\beta-2} \right] F^{(t)} & \frac{1}{\beta - \delta} \\ \left[ Z^{(t-1)} \odot \left( F^{(t)} A^{(t-1)'} \right)^{\beta-1} \right] F^{(t)} + \frac{\phi^{(t-1)} A^{(t-1)}}{A F^{(t-1)}} & \frac{1}{\beta - \delta} \end{cases}$
11: $w^{(t-1)} \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{p} z_{ij}^{(t-1)} + c - 1$
12: $\lambda_{m}^{(t)} \leftarrow \frac{\|f^{(t)}_{(m)}\|^2 + \|a^{(t)}_{(q)}\|^2 + 2v}{n+p+2(u+1)} (m = 1, \ldots, k)$
13: if $t \leq \delta$ or $t \mod \eta = 0$ then
14: $\phi^{(t)}$ is obtained as the optimal $\phi$ that optimizes $Q_{\phi}(\phi)$ given $F^{(t)}, A^{(t)}, \beta$, and $Z^{(t)}$ using the BFGS quasi-Newton method with constraints $\phi > 0$.
15: end if
The meaningless vectors of $F$ and $A$ become zero vectors automatically as the algorithm runs. Update rules (13) and (14) indicate that values of the $m$th vector decrease when updated $\lambda_m$ is small. Additionally, update rule (15) indicates that $\lambda_m$ becomes smaller when values of the $m$th vectors are small. If all values of the $m$th vectors become 0, $\lambda_m$ reaches its minimum value (16). When the maximum relative difference between the current $\lambda^{(t)}_m$ and previous $\lambda^{(t-1)}_m$ decreases below a sufficiently small threshold $\tau_{cut}$, the algorithm stops.

At the end of the algorithm, the effective factors are determined to be all $m$th factors such that the estimated relevance weight $\lambda_m$ are larger than the minimum (16). This stopping rule, and the factor selection rule, are the same as in Tan and Févotte (2013). Note that we limit the number of times $\phi$ is updated in order to avoid a large computation time. $\phi$ represents the trade-off between the data fitting and the regularization term, as shown in (10). Hence, to determine the degree of this trade-off at the beginning of an iteration, we update $\phi$ for the first $\delta$ iterations; then, for the remainder of the iterations, we update it at every $\eta$th iteration. The hyperparameter $v$ is determined using the prior assumption and the law of large numbers.

Input hyperparameters include $\beta$, $u$, $c$, $d$, $\delta$, $\eta$, $\tau_{cut}$, and $k$. Though there are quite a few parameters here, $\beta$ is the most crucial for the algorithm, whereas the others are not as important or have a recommended value. $u$ should be a small value (e.g., $u = 5$) to be an informative prior for $\lambda$. In fact, if the value of $u$ is sufficiently smaller than $n + p$, the impact of updating $\lambda$ using (15) from $u$ becomes small. Tan and Févotte (2013) demonstrated good performance for true order determination with a small value of $u$ and overestimated with a large value of $u$. Since we assume a priori distribution for $w$ and as (4), if there is no a priori assumption for $w$, $c$ and $d$ must be set to $c = 1$ and $d = 1$. We can select the number of iterations $\delta$ such that changes in the value of $\phi$ become small within the first $\delta$ iterations. $\eta$ should be a relatively large number (e.g., $\eta = 500$) to reduce computation time. $\tau_{cut}$ should be an extremely small value (e.g., $\tau_{cut} = 10^{-7}$), as in the work of Tan and Févotte (2013). There is no explanation of the reasoning behind this $\tau_{cut}$ value in Tan and Févotte (2013). However, it seems to be a good value. If it is decreased, the computational time becomes large; if it is increased, the algorithm stops before it catches the true number of factors. $k$ should be a large value because it denotes the maximum number of factors to be assumed. Unfortunately, there is no procedure for estimating $\beta$ in NMF problems. Although Simşekli et al. (2013) estimates $\beta$ using linear search maximum likelihood in NMF, this procedure is not appropriate. Using different $\beta$ means using different measures

| Algorithm | ARDNMF-ZICP (continued) |
|-----------|--------------------------|
| 16: $z_{ij}^{(t)} \leftarrow \begin{cases} w^{(t)} & \text{if } y_{ij} = 0 \\ w^{(t)} + (1 - w^{(t)})g_{CP}(0|\mathbf{f}_i^{(t)}\mathbf{a}_j^{(t)},\phi^{(t)},\beta) & \text{if } y_{ij} \neq 0. \end{cases}$ |
| 17: $\tau^{(t)} \leftarrow \max_m \left| \frac{\lambda^{(t)}_m - \lambda^{(t-1)}_m}{\lambda^{(t-1)}_m} \right|$ |
| 18: until $\tau^{(t)} < \tau_{cut}$ |
| 19: Select factors as $M = \{m | m \in \{1, \ldots, k\}$ and $\frac{\lambda^{(t)}_m - 2p/\{n + p + 2(u+1)\} > \tau_{cut} \}$ |
| 20: Set $\hat{F}$ and $\hat{A}$ containing $m$th $(m \in M)$ column vectors of $F^{(t)}$ and $A^{(t)}$, respectively. |
| 21: Output $\hat{F}, \hat{A}, \mathbf{Z}^{(t)}, \phi^{(t)}$, and $w^{(t)}$ |
of divergence between the data and model, e.g., a normal distribution \((\beta = 2)\) corresponds to a Euclidean distribution; a Poisson distribution \((\beta = 1)\) corresponds to KL-divergence; and an exponential distribution \((\beta \to 0)\) corresponds to Itakura-Saito divergence, as shown in Table 1. Models estimated using different measures of divergence should not compared.

In this algorithm, the sequence of the objective function values is monotonically non-increasing, as shown in Figure 1. However, the estimates obtained by this algorithm can be local optima. Hence, it is recommended to run the algorithm with some random starts and select estimates with the least objective function value.

Fig. 1: Plot of the sequence of the objective function values in iterations for an example of the simulation study described in Section 4. on conditions as follows: \(n = 100, \beta = 0.8, r = 40\). Note that the maximum of the horizontal axis is set to 2000 in this plot. The actual number of iterations is more than 10000.

4. Simulation study

In this section, we describe a simulation study to confirm our model order selection accuracy. We generate a synthetic zero-inflated data matrix, execute Tan and Févotte’s ARD method as well as ours using this matrix, and confirm the estimated model order. Details of the procedure are as follows:
1. Generating synthetic data matrix $Y$.

1.1. Parameters $n$, $p$, $\bar{k}$, $\bar{u}$, $\bar{v}$, $\bar{\phi}$, $\beta$, and $r$ are set for generating the synthetic data matrix. Here, the $k$ and $\phi$ are the true number of factors and the true dispersion parameter, respectively.

1.2. $\bar{\lambda}_m (m = 1, \ldots, \bar{k})$ are generated from $IG(\bar{u}, \bar{v})$.

1.3. $\bar{f}_{im}$ and $\bar{a}_{jm} (i = 1, \ldots, n; j = 1, \ldots, p; m = 1, \ldots, \bar{k})$ are generated from $HN(\bar{\lambda}_m)$.

1.4. A noiseless data matrix is calculated as $X = \bar{F}\bar{A}'$.

1.5. Each element of data matrix $y^*_{ij}$ is generated from $CP(x_{ij}, \bar{\phi}, \beta)$.

1.6. $Y$ is generated such that $r\%$ of elements in $Y^*$ are converted to 0.

2. Executing our proposed method and Tan and Févotte’s method.

2.1. Hyperparameters $u$, $c$, $d$, $\delta$, $\eta$, $\tau_{\text{cut}}$, and $k$ are set for algorithm execution.

2.2. Initial input $\tilde{F}$, $\tilde{A}$, $\tilde{\phi}$, $\tilde{w}$, is set such that $\tilde{f}_{im} \sim U(0, 1)$, $\tilde{a}_{jm} \sim U(0, 1)$, $\tilde{\phi} = 0.1$, and $\tilde{w} = 0.5$. $U(0, 1)$ is a uniform distribution with range (0, 1).

2.3. Tan and Févotte’s and our ARD methods are executed given $Y$, $\beta$, $u$, $c$, $d$, $\delta$, $\eta$, $\tau_{\text{cut}}$, $k$, $\tilde{F}$, $\tilde{A}$, $\tilde{\phi}$, and $\tilde{w}$ and the estimated model orders, $\hat{k}_{\text{Tan}}$ and $\hat{k}_{\text{ours}}$, are obtained.

The parameter settings are as follows:

- $n = \{100, 500\}$,
- $p = 50$,
- $\bar{k} = 5$,
- $\bar{u} = 50$, $\bar{v} = \pi(\bar{u} - 1)10/(2\bar{k})$,
- $\bar{\phi} = 1$,
- $\beta = \{0.2, 0.5, 0.8\}$,
- $r = \{0, 20, 40, 60\}$,
- $u = 2$, $c = d = 1$,
- $\delta = 5$,
- $\eta = 100$,
- $\tau_{\text{cut}} = 10^{-7}$,
- $k = 10$.

It is important to note that $\bar{v}$ is set such that the expected value of an element in the data matrix is 10. Design variables $n$, $\beta$, and $r$ seem to affect the model order estimation; therefore, the number of experiment conditions is $2 \times 3 \times 4 = 24$. We generate 50 synthetic data matrices for each condition. We check whether the estimated numbers of factors, $\hat{k}_{\text{Tan}}$ and $\hat{k}_{\text{ours}}$, are equal to the true number of factors, $\bar{k} = 5$. Figure 2 shows box plots of the 50 estimated numbers of factors for each condition.

First, we find that the estimated model order of Tan and Févotte’s method is less than 5 when $\beta = 0.2$, even if $r = 0$. This result indicates that Tan and Févotte’s ARD method fails to obtain the true number of factors when $\beta = 0.2$. Although we do not understand the reason, it may be a crucial problem of Tan and Févotte’s ARD method and its analysis is a component of our future work. In contrast, when $\beta = 0.5$ and $\beta = 0.8$, both Tan and Févotte’s and our ARD method can estimate the true model order 5. For a zero-inflated
matrix (i.e., \(r = 20, r = 40,\) and \(r = 60\)), we find that Tan and Févotte’s ARD method estimates the model order to be less than 5. Of course, this is because many elements in the data matrix are forcibly converted to 0. Our ARD method estimates the order nearer to 5 than Tan and Févotte’s ARD method. Although the orders estimated when \(\beta = 0.2\) and \(\beta = 0.5\) are smaller than 5, the orders estimated when \(\beta = 0.8\) are near to 5. However, the order tends to be small when the data matrix is small and contains many 0s, a seemingly obvious result. For a small and highly zero-inflated matrix, matrix information decreases and even our method cannot catch the true factors. In summary, our ARD method, based on a zero-inflated CP distribution, performs better than that of Tan and Févotte, especially in estimating the true order for relatively large matrices that are not highly zero-inflated; however, when \(\beta\) or the data matrix are small, or the matrix is highly zero-inflated, even our proposed ARD method does not estimate a correct order.

In addition, we compare the closeness between true parameters and estimated parameters by using the two methods. We calculate the mean square error (MSE) between estimated parameter \(\hat{F}\) and true parameter \(\bar{F}\). Here, we convert the length of column vectors of \(\hat{A}\) and \(A\) to 1 and also convert these of \(\bar{F}\) and \(\bar{F}\), e.g., \(X = FA' = FDAD^{-1}A' = F^*A'\), where \(D_A\) is a \(k \times k\) diagonal matrix in which diagonal elements are \(\|a_{(1)}\|, \ldots, \|a_{(k)}\|\).
When the estimated model order is not equal to the true order, we calculate its MSE by adding the zero column vectors to the short factor matrix; if $\hat{k} = 3$, we add 2 zero vectors to $\hat{F}$, else if $\hat{k} = 6$, we add a zero vector to $\hat{F}$. Figure 3 and 4 show MSE of $\hat{F}$ and $\hat{A}$, respectively. We find that the MSE from our method is relatively small for a zero-inflated nonnegative matrix when $\beta = 0.5$ and $\beta = 0.8$. This indicates that our method has greater accuracy when estimating the factor matrices in that situation. However, when $\beta = 0.2$, our method shows a large MSE in some situations. As described above, the model order is also worse when $\beta = 0.2$. The cause of this problem in $\beta = 0.2$ will be investigated in our future work. Figure 5 shows the estimates of $\phi$. In the zero-inflated situation, the estimates of Tan and Févotte’s ARD method are large; on the other hand, $\phi$ can be estimated as the true value of 1 by our method. Figure 6 shows the estimates of $w$. Of course, in Tan and Févotte’s ARD method, $w$ is not estimated, and hence, we set the value as 0. In our method, $w$ can be estimated to smaller than $r/100$. This implies that some elements that are forcibly converted to 0 are generated from the factor model, e.g., $CP(x_{ij}, \phi, \beta)$.
Fig. 4: Box plots of the 50 calculated MSE of $\hat{A}$ for each condition. The left and right box plots in each plot area are values obtained by Tan and Févotte’s method and our method, respectively. Values below each box plot indicate mean of the values.
Fig. 5: Box plots of the 50 estimated $\phi$ for each condition. The left and right box plots in each plot area are estimates by Tan and Févotte’s method and our method, respectively. Values below each box plot indicate mean values of the estimated $\phi$. 
Fig. 6: Box plots of the 50 estimated $w$ for each condition. The left and right box plots in each plot area are estimates by Tan and Févotte’s method and our method, respectively. Values below each box plot indicate mean values of the estimated $w$. 
We also measured the computational time (Figure 7). It is obvious that our method requires more time than Tan and Féotte’s method does, owing to the $Z$ and $w$ updating steps. Our method has the longest computational time—about 800 seconds—when $r = 0$, $\beta = 0.8$, and $n = 500$, though it seems to be faster than other model order determination methods, including a Bayesian method using MCMC to calculate model evidence (Schmidt et al., 2009) and a cross-validation method (Owen and Perry, 2009). We find that both Tan and Féotte’s and our methods are very fast when they fail to estimate true number of factors.

Fig. 7: Box plots of the 50 computational times (sec) for each condition. The left and right box plots in each plot area are estimates by Tan and Féotte’s method and our method, respectively. Values below each box plot indicate mean values of the computational time.

5. Application

We apply our ARD method to the POS data of a Japanese grocery store for June 2014, including customer ID information. We create a matrix that includes customer spending

---

1"i-codePOS Data" provided by IDS Co., Ltd. in the 2015 Data Analysis Competition hosted by the Joint Association Study Group of Management Science.
in monetary units (rows) in various product categories (columns) via data cleansing. The statistics for the original data set and the data set after cleansing are shown in Table 2.

Table 2: Statistics for the original and cleansed data sets from the POS data

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

Using this data set, we estimate the number of factors and the factor matrix that represents buying patterns. We set the hyperparameters as follows:

\[
\begin{align*}
\beta &= 0.8 \\
u &= 2, \\
c &= d = 1 \\
\delta &= 20, \\
\eta &= 100 \\
\tau_{\text{cut}} &= 10^{-7} \\
k &= 20.
\end{align*}
\]

We chose \( \beta = 0.8 \) for the following reason: purchase price is similar to that the number of items purchased in that it is assumed to be Poisson (\( \beta = 1 \)) distributed, and \( \beta = 0.8 \) is near to \( \beta = 1 \). We use 20 random starts and select the best estimates that minimize \( Q_{\text{incomp}} \). Results are shown in Table 3.

The number of factors is estimated as 3 by both methods. We denote the \( m \)th factor of Tan and Févotte’s and our methods as \( \text{TF}_m \) and \( \text{OF}_m \), respectively. An interpretation of the factors estimated by Tan and Févotte’s method follows:

**TF1:** Buying alcoholic beverages and foods that accompany alcoholic beverages.

**TF2:** Buying fruit, vegetables, mizumono (natto, konjac, tofu, etc.), milk products, meats, and seasonings, which are basic foods in Japan.

**TF3:** Buying pre-prepared deli foods, lunch boxes, refreshing beverages, confectioneries, and breads.

Factors of our method are similar to those of Tan and Févotte’s method, but the first and third factors are a little more complicated.

**OF1:** Buying alcoholic beverages and items in TF3: especially rice deli foods, refreshing beverages, dry confectioneries, breads, and lunch boxes.

**OF2:** Buying items in TF2.

**OF3:** Buying all items in TF2 and TF3 except for meats, seasonings, lunch boxes, rice deli foods, and refreshing beverages.

It is assumed that such complicated factors are due to the effect of the zero-inflated model: two factor matrices (\( F \) and \( A \)), are estimated considering some values of \( y_{ij} = 0 \) as non-zero. For example, “Lunch box” and “Beer” are not in the same factor in Tan and Févotte’s
Table 3: A estimated by Tan and Févote’s (left) and our (right) ARD methods. All columns are normalized to 1. Only product categories with values greater than 0.2 are shown.

| Product classifications | Tan and Févote’s ARD | Our ARD |
|-------------------------|----------------------|---------|
|                         | TF1 | TF2 | TF3 | OF1 | OF2 | OF3 |
| Beer                    | 0.61 | 0.00 | 0.06 | 0.42 | 0.00 | 0.00 |
| Wine                    | 0.37 | 0.00 | 0.00 | 0.18 | 0.00 | 0.00 |
| Sake                    | 0.29 | 0.00 | 0.00 | 0.13 | 0.00 | 0.00 |
| Sashimi platter         | 0.27 | 0.00 | 0.00 | 0.00 | 0.00 | 0.06 |
| Wagyu beef              | 0.23 | 0.08 | 0.00 | 0.00 | 0.16 | 0.04 |
| Rice                    | 0.00 | 0.13 | 0.00 | 0.35 | 0.00 | 0.00 |
| Fruit/vegetable         | 0.01 | 0.37 | 0.04 | 0.03 | 0.36 | 0.20 |
| Mizumono                | 0.05 | 0.26 | 0.07 | 0.06 | 0.25 | 0.16 |
| Milk product            | 0.05 | 0.24 | 0.14 | 0.14 | 0.25 | 0.16 |
| Imported fruit          | 0.00 | 0.23 | 0.10 | 0.06 | 0.21 | 0.19 |
| Vegetable-related       | 0.00 | 0.23 | 0.00 | 0.00 | 0.20 | 0.15 |
| Fruits in season        | 0.00 | 0.22 | 0.11 | 0.01 | 0.14 | 0.29 |
| Japan-raised pork       | 0.03 | 0.22 | 0.00 | 0.00 | 0.24 | 0.11 |
| Processed meat          | 0.06 | 0.21 | 0.05 | 0.04 | 0.23 | 0.12 |
| Basic seasoning         | 0.04 | 0.18 | 0.02 | 0.03 | 0.23 | 0.06 |
| Processing seasoning    | 0.06 | 0.19 | 0.04 | 0.05 | 0.22 | 0.08 |
| Salad deli              | 0.02 | 0.03 | 0.32 | 0.19 | 0.00 | 0.21 |
| Sushi                   | 0.06 | 0.11 | 0.32 | 0.15 | 0.00 | 0.37 |
| Fried food deli         | 0.00 | 0.06 | 0.28 | 0.15 | 0.00 | 0.24 |
| Japanese-style deli     | 0.00 | 0.01 | 0.27 | 0.08 | 0.00 | 0.19 |
| Rice deli               | 0.00 | 0.00 | 0.25 | 0.27 | 0.00 | 0.06 |
| Refreshing beverage     | 0.04 | 0.05 | 0.25 | 0.30 | 0.05 | 0.05 |
| Fresh Japanese sweets   | 0.00 | 0.12 | 0.24 | 0.15 | 0.05 | 0.24 |
| Bread                   | 0.00 | 0.17 | 0.24 | 0.21 | 0.14 | 0.19 |
| Dry confectionery       | 0.01 | 0.13 | 0.23 | 0.20 | 0.11 | 0.17 |
| Lunch box               | 0.00 | 0.00 | 0.21 | 0.31 | 0.00 | 0.00 |

estimates. This means that there are few customers buying both items. However, in our method, customers buying “Lunch box” or “Beer” but not both are regarded as customers buying both, because some zero values in data matrix \( Y \) are disregarded. In other words, some elements of \( y_{ij} = 0 \) in the “Lunch box” or “Beer” columns are assumed not to be generated from the distribution \( y_{ij} \sim \text{CP}(x_{ij}, \phi, \beta) \) but from \( y_{ij} \sim 0 \), 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 \( z_{ij} \approx 1 \), information from the \( i, j \) elements becomes weak in the \( f_{im} \) and \( a_{jm} \) update rules (see (13) and (14)). This result can be interpreted realistically as follows: customers who bought “Lunch box” items but not “Beer” would have bought “Beer” if the customers had conformed to the estimated buying model, but the customers did not actually buy “Beer” due to other causes. From the dataset, the number of customers who bought both items is 206 (Table 4). This is the 3322nd highest pattern in \( 114C_2 = 6441 \) product category patterns. This means that such customers exist, but that there are relatively few. From Figure 8, which shows the scores of 4 OF1 buying patterns, the customers buying both or either item have high scores. This indicates that our method
considers the customers buying either item to be the same as the customers buying both. Moreover, Figure 9 shows that the customers who only bought a “Lunch box” have larger $z_{ij}$ values for “Beer” than those who bought neither. This result suggests that the customers who did not buy “Beer” but who bought a “Lunch box” made these purchases not because of OF1 factors, but due to something else. From Figure 10, the cause of customers buying no “Lunch box” who bought “Beer” is also not from OF1 factors.

Table 4: The number of customers by 4 buying pattern

|                | Frequency | Proportion (%) |
|----------------|-----------|----------------|
| Both           | 206       | 2.8            |
| Lunch box only | 971       | 13.2           |
| Beer only      | 1053      | 14.3           |
| Neither        | 5118      | 69.7           |
| Total          | 7348      | 100.0          |

Fig. 8: Box plots of the 7348 customer’s scores of the OF1 by 4 buying pattern.
Fig. 9: Box plots of the 7348 customer’s values of the $z_{ij}$ of the “Beer” by 4 buying pattern.
Fig. 10: Box plots of the 7348 customer’s values of the $z_{ij}$ of the “Lunch box” by 4 buying pattern.
6. Conclusion

In this paper, we present a new ARD method to determine the model order of NMF using ZICP. Our simulation study performs better than the ordinary Tan and Févotte’s model order selection method in cases of zero-inflated data, but shows the limitation of our method: a true model order must be estimated differently when the data matrix in question is small, contains a large number of 0s, or $\beta$ has a specific value. Moreover, the application to POS shows the difference in estimated factor matrices between Tan and Févotte’s method and our ARD method.

Our future work will focus on model order selection for nonnegative matrix tri-factorization (NMTF), which is a method to decompose a data matrix $Y$ to three factor matrices (Cichocki et al., 2007). In NMTF, we have to determine two factor numbers for the row and column objects of $Y$.

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.

Basu, A., Harris, I. R., Hjort, N. L., and Jones, M. C. (1998). Robust and efficient estimation by minimising a density power divergence. Biometrika, 85, 549–559.

Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P. and Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. Computational statistics & data analysis, 52, 155–173.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Byrd, R. H., Lu, P., Nocedal, J. and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing, 16, 1190–1208.

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

Cichocki, A., Zdunek, R., Choi, S., Plemmons, R. and Amari, S. I. (2007). Non-negative tensor factorization using alpha and beta divergences. In 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07, 3, III–1393.

Daniel, C. (1959). Use of half-normal plots in interpreting factorial two-level experiments. Technometrics, 1, 311-341.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society. Series B (methodological), 1–38.

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

Févotte, C. and Idier, J. (2011). Algorithms for nonnegative matrix factorization with the $\beta$-divergence. Neural Computation, 23, 2421–2456.

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

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

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. The annals of mathematical statistics, 22, 79–86.
Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. Technometrics, 34, 1-14.

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

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

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

Nakano, M., Kameoka, H., Le Roux, J., Kitano, Y., Ono, N. and Sagayama, S. (2010). Convergence-guaranteed multiplicative algorithms for nonnegative matrix factorization with β-divergence. In Proceedings of the 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 283–288.

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

Schmidt, M. N., Winther, O., Hansen, L. K. (2009). Bayesian non-negative matrix factorization. In International Conference on Independent Component Analysis and Signal Separation, 540–547.

Şimşekli, U., Cemgil, A., Yilmaz, Y. K. (2013). Learning the beta-divergence in Tweedie compound Poisson matrix factorization models. In Proceedings of the 30th International Conference on Machine Learning (ICML-13), 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, 1592–1605.

Wang, Y. X. and Zhang, Y. J. (2013). Nonnegative matrix factorization: A comprehensive review. IEEE Transactions on Knowledge and Data Engineering, 25, 1336–1353.

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

(Received: August 16, 2016, Accepted: January 10, 2017)