INVITED PAPER *Special Section on Award-winning Papers*

Learning of Nonnegative Matrix Factorization Models for Inconsistent Resolution Dataset Analysis

Masahiro KOHJIMA[†a)], **Tatsushi MATSUBAYASHI**[†], ***Nonmembers***, ***and*** **Hiroshi SAWADA**[†], ***Member***

SUMMARY Due to the need to protect personal information and the impracticality of exhaustive data collection, there is increasing need to deal with datasets with various levels of granularity, such as user-individual data and user-group data. In this study, we propose a new method for jointly analyzing multiple datasets with different granularity. The proposed method is a probabilistic model based on nonnegative matrix factorization, which is derived by introducing latent variables that indicate the high-resolution data underlying the low-resolution data. Experiments on purchase logs show that the proposed method has a better performance than the existing methods. Furthermore, by deriving an extension of the proposed method, we show that the proposed method is a new fundamental approach for analyzing datasets with different granularity.

key words: *inconsistent resolution dataset, probabilistic model, nonnegative matrix factorization, collective matrix factorization*

1. Introduction

Companies that have succeeded by using the power of data analysis are attracting attention and many more companies are accelerating efforts on data collection and analysis. Due to the difficulty of exhaustive data collection and the need to protect personal information, it is becoming more urgent to be able to analyze multiple datasets that have various levels of granularity, for example, a set of user-individual data such as "how many times an item is purchased by a user" and user-group data such as "how many times a shop is visited by users of the same age". Therefore, we consider the problem of inconsistent resolution dataset analysis, which is to analyze a combination of datasets with different granularity. High resolution datasets (such as user-individual data) capture the events that occurred in fine detail such as individual visits and purchases and said to have fine grain size. Low resolution datasets (user group data) offer less detail, i.e. coarser granularity.

We provide two examples that require inconsistent resolution dataset analysis. The first example is an analysis of data collected in the retail industry (Fig. 1 (a)). Currently, many retail shops collect information about users by issuing them with membership cards. However, since not all shoppers will have a membership card, exhaustively data collection, some purchase log entries do not have the identification information (ID) of the membership card; Instead they contain only information on the sex and age of the user as input by the shop staff from assessments of the appearance of the user at the sales point. Therefore, the collected data consists of user-individual data and user-group data. When the purchase log contains little member user data, inconsistent resolution dataset analysis can be useful by allowing use of the purchase data of non-member users. The second example is analysis of the combined datasets of different companies (Fig. 1 (b)). The social data provided by location information services e.g., Foursquare* and Yelp**, omits the data of individual users to protect personal information; only visit logs of user groups are disclosed, for example, how many "women" have visited a certain shop. Therefore, inconsistent resolution dataset analysis is required if we are to analyze a dataset created by combining user-individual data and the above social data.

In this study, we propose a new method for inconsistent resolution dataset analysis***. The proposed method is a probabilistic model based on nonnegative matrix factorization (NMF) [2]–[4]. First of all, to introduce the basic setting of inconsistent resolution dataset analysis, we focus on the situation where two assumptions are satisfied: (A1) common user set exists, (A2) data are independent and identically distributed. We use assumptions (A1) and (A2) and the NMF formulation to propose probabilistic nonnegative inconsistent resolution matrix factorization (*pNimf*) that can jointly analyze high and low resolution data. *pNimf* makes it possible to analyze data more accurately than the methods that use a single set of data. For example, applying *pNimf* to the purchase history of the members/non-members mentioned above, improves the accuracy of missing value complementation in the matrix, making it possible to more accurately predict the quantities purchased by members/non-members. In addition, it is possible to extract purchasing patterns that reflect the purchasing tendencies of both members and non-members.

pNimf is derived by considering the data generative process that covers the latent high resolution data that underlies the low resolution matrix. Latent high resolution data can be defined from assumption (A1) and relation between high resolution data and low resolution data can be deduced from assumption (A2). While it is not possible to assume that assumptions (A1) and (A2) hold for all prob-

Manuscript received September 27, 2018.
Manuscript revised December 20, 2018.
Manuscript publicized February 4, 2019.

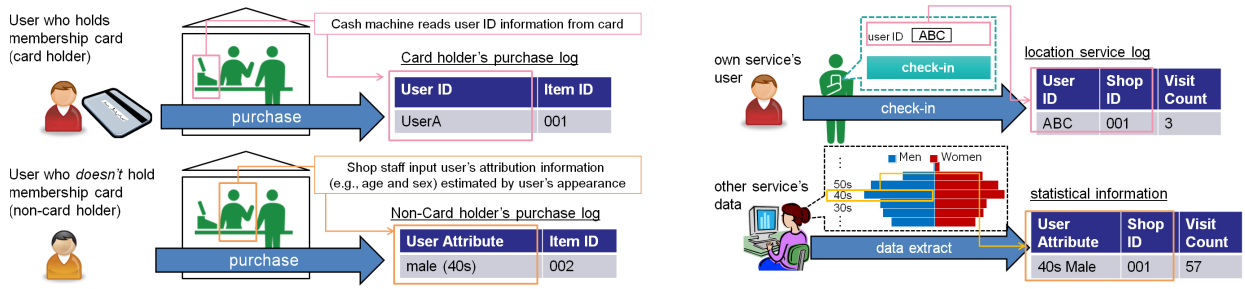
[†]The authors are with NTT Service Evolution Laboratories, Yokosuka-shi, 239–0847 Japan.
a) E-mail: kohjima.masahiro@lab.ntt.co.jp

DOI: 10.1587/transinf.2018AWI0002

*http://gnip.com/sources/foursquare/

**http://www.yelp.com/dataset_challenge

***An earlier version of this work was presented at international conference on information and knowledge management (CIKM) [1].



(a) members' and non-members' logs (b) own/other companies' logs

Fig. 1 Example of datasets requiring inconsistent resolution analysis.

lems, approaches that use the relationship described in this paper can be the basis for solving a lot of general problems. In this paper, we also show the situation that diverges from the above two assumptions, and an extended version of the proposed method is provided for cases that demand different assumptions.

The structure of this paper is as follows. §2 introduces related researches and §3 details the proposed method. Further analysis of *pNimf* is done in §4 and the experimental evaluation is shown in §5. §6 shows examples of an extension of the proposed method and §7 gives our conclusions.

2. Related Works

Nonnegative matrix factorization (NMF) [2], [3] is a method of factorizing an input matrix into a product of nonnegative matrices. It is known that NMF can be used for soft clustering and completion of missing values in a matrix by utilizing the result of factorization. Since NMF can deal with various loss functions, it can be applied to various types of data such as movie evaluation logs, document corpora, purchase logs and so on [4]. In addition, when generalized Kullback-Leibler divergence is used as the loss function, NMF is equivalent to a prominent technique in information retrieval, probabilistic latent semantic indexing (PLSI) [5], which is also the basis of latent dirichlet allocation (LDA) [6], [7]. Given these facts, NMF is considered to be one of the core technologies in machine learning, so we adopted it as the basis of *pNimf*.

In recent years, collective matrix factorization (CMF) or multiple matrix factorization (MMF) techniques have been proposed for multiple dataset analysis [8]. A CMF/MMF extension of NMF called Nonnegative Multiple Matrix Factorization (NMMF) [9] has been described [9], [10]. These techniques combine multiple matrices and have been reported to offer a better performance than the techniques that use only a single matrix. However, these methods are not designed to handle datasets that have different resolutions. For a context different from CMF/MMF, Aimoto et al. proposed a method for combining information of aggregated data (corresponding to low resolution matrix in this paper) in matrix factorization [11]. However, this method is specialized for situations where the datasets with different granularity represent exactly the same data, making it unsuitable as a basic method for general inconsistent resolution dataset analysis.

3. Proposed Method

3.1 Formulation

In this section we focus on the problem of inconsistent resolution dataset analysis in situations where two assumptions are satisfied: (A1) - common user assumption, (A2) independently and identically distributed assumption. Before providing a mathematical representation of these assumptions, we give an intuitive explanation. A certain supermarket issued a members card in December to all users of the store. Clearly then the shop's sales records contain no personal details prior to December. As shown in Fig. 2, the purchase history for November consists of low resolution data, while that for December contain high resolution data. Note that user attribute information such as sex and age (est.) is recorded in the purchase history for November. In this example, assumption (A1) is that the set of all shop users in November and December are equal (whether or not they purchased any item). Assumption (A2) states that each user will make the same product purchases in November and December. We will explain using this example of purchase history analysis, and the symbol definitions follow this example. However, our research is not limited to this example, and more general circumstances are explained in §4.2.

Definition of Symbols: Let I, J and K represent the number of users, items, and attributes, respectively. We define the element of $\boldsymbol{X}$, x_{ij}, as the number of purchases of item j by user i in Dec. and the element of $\boldsymbol{Y}$, y_{kj}, as the number of purchases of item j by users with attribute k in Nov. Each $\boldsymbol{X}$ and $\boldsymbol{Y}$ are taken to be the high-resolution matrix and the low-resolution matrix, respectively. We also assume that user's attribute information is available. This assumption is natural because such data is required, for example, when the user creates the membership card. $\boldsymbol{V} = \{v_{ik}\}_{i,k=1}^{I,K}$, whose element $v_{ik} \in \{0, 1\}$ is set to 1 if the attribute of user i is k, otherwise 0.

Latent High Resolution Matrix: Next, we define the *latent high resolution matrix*, $\boldsymbol{Z}$. This matrix plays an important role in our model. We define $\boldsymbol{Z}$ as the matrix that corresponds to the high-resolution data in Nov., i.e. data

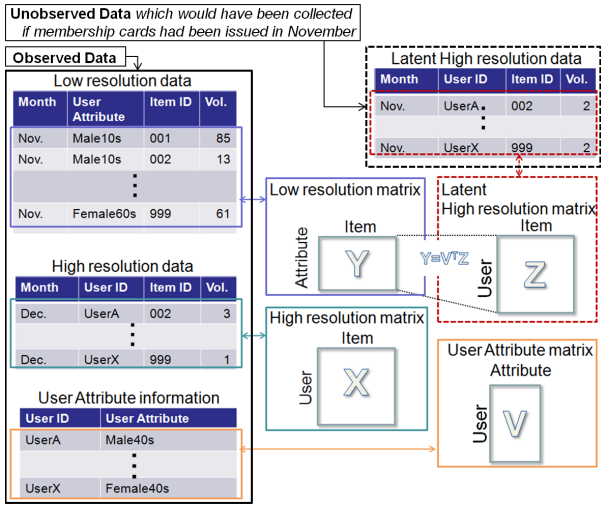


Fig. 2 Example of observed and unobserved data.

which would have collected if membership cards had been issued in Nov. Since only low-resolution data is collected in Nov., $\boldsymbol{Z}$ is the unobserved latent high resolution data which lies under the low resolution data. We are usually unable to know the set of users that exist behind $\boldsymbol{Z}$ and the number of rows of $\boldsymbol{Z}$ cannot be defined. To resolve this, we use (A1), which we formally define as follows: *user population of high-resolution data* $\boldsymbol{X}$ *and that of latent high-resolution data* $\boldsymbol{Z}$ *are identical*. (A1) allows us to define the number of the rows of $\boldsymbol{Z}$ as being identical to $\boldsymbol{X}$, I. Then, we define element z_{ij} as the number of purchases of item j by user i in Nov. Importantly, this definition yields a relation between $\boldsymbol{Y}$ and $\boldsymbol{Z}$, $\boldsymbol{Y} = \boldsymbol{V}^T\boldsymbol{Z}$. This comes from the fact that y_{kj} is equal to the summation of z_{ij} over user i with attribute k, i.e. $y_{kj} = \sum_i v_{ik} z_{ij}$.

3.2 Model

This subsection presents the proposed model. Let $\boldsymbol{A} := \{a_{ir}\}_{i,r=1}^{I,R}$ and $\boldsymbol{B} := \{b_{jr}\}_{j,r=1}^{J,R}$ be the user factor matrix and item factor matrix, respectively. R is the number of factors. Each vector of factor matrices $(a_{i1}, \cdots, a_{iR})$, $(b_{j1}, \cdots, b_{jR})$ is interpreted as the latent feature of user i and item j. We also define $\hat{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{B}^T$; its element is written as $\hat{x}_{ij} = \sum_r a_{ir} b_{jr}$. Since the Poisson distribution is frequently used to model count data such as purchase log and visit count, we adopt it for our model. NMF that uses Poisson models the probability of generating matrix $\boldsymbol{X}$ as

$$P(\boldsymbol{X}|\boldsymbol{A},\boldsymbol{B}) = \prod_{i,j=1}^{I,J} \mathcal{PO}(x_{ij}|\hat{x}_{ij}), \tag{1}$$

where $\mathcal{PO}$ is the Poisson probability distribution:

$$\mathcal{PO}(x_{ij}|\hat{x}_{ij}) = \exp\{-\hat{x}_{ij} + x_{ij}\log(\hat{x}_{ij}) - \log\Gamma(x_{ij}+1)\}.$$

Note that our model can be extended, in an analogous manner, to the case that other probability distributions such as Gaussian are adopted.

We derive the proposed method based on the data generative process summarized as follows: (i) define the probability distribution that generates both $\boldsymbol{X}$ and $\boldsymbol{Z}$. (ii) use (A2) *iid assumption*, which we formally define as follows: elements of $\boldsymbol{X}$ and $\boldsymbol{Z}$ that have the same indices, x_{ij} and z_{ij}, follow the identical probability distribution (in this case, Poisson dist. with parameter $\hat{x}_{ij}$ as in Eq. (1)) and they are mutually independent. (A2) helps to extract factors which are independent of month. (iii) use the relation between $\boldsymbol{Z}$ and $\boldsymbol{Y}$ ($y_{kj} = \sum_i v_{ik} z_{ij}$) explained in the previous section. Combining these parts, the joint distribution of $\boldsymbol{X}$, $\boldsymbol{Z}$, $\boldsymbol{Y}$ is written as

$$\begin{aligned} &P(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{B},\boldsymbol{V}) \\ &= \prod_{i,j} \mathcal{PO}(x_{ij}|\hat{x}_{ij})\mathcal{PO}(z_{ij}|\hat{x}_{ij}) \prod_{k,j} \delta(y_{kj} - \sum_i v_{ik} z_{ij}), \end{aligned} \tag{2}$$

where $\delta(\cdot)$ is the delta function. Figure 3 (a) shows a graphical model representation. By explicitly modeling the generation of *latent high-resolution matrix* $\boldsymbol{Z}$, we can naturally define the probability distribution of all matrices. However, since the size of $\boldsymbol{Z}$ is $I \times J$, which is considerable, it is desirable to work with more convenient probabilistic models.

The key to practical implementation lies in a characteristic of Poisson distributions: the sum of Poisson-distributed random variables is also a Poisson-distributed random variable, i.e., closed under addition. In our model, z_{ij} represents Poisson-distributed random variables and y_{kj} is their summation. Thus, we can marginalize out $\boldsymbol{Z}$ from Eq. (2) which yields the following equation:

$$\begin{aligned} P(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{B},\boldsymbol{V}) &= \int P(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{B},\boldsymbol{V}) d\boldsymbol{Z} \\ &= \prod_{i,j} \mathcal{PO}(x_{ij}|\hat{x}_{ij}) \prod_{k,j} \mathcal{PO}(y_{kj}|\hat{y}_{kj}), \end{aligned} \tag{3}$$

$$\text{where} \quad \hat{y}_{kj} = \sum_{r=1}^{R} c_{kr} b_{jr} \quad \text{and} \quad c_{kr} = \sum_{i=1}^{I} v_{ik} a_{ir}. \tag{4}$$

Figure 3 (b) shows a graphical model representation. Considering that $\boldsymbol{C} := \{c_{kr}\}_{k,r=1}^{K,R}$ is the attribute latent factor matrix, Eq. (3) can be interpreted as factorizing the high-resolution matrix and low-resolution matrix simultaneously, while retaining the relation between factor matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ using $\boldsymbol{V}$ ($\boldsymbol{C} = \boldsymbol{V}^T\boldsymbol{A}$ as in Eq. (4)). Thus, we call this proposal probabilistic non-negative inconsistent-resolution matrix factorization (*pNimf*). Figure 4 shows the factorization form. Note that removing the linear equality relation between factor matrices, $\boldsymbol{C} = \boldsymbol{V}^T\boldsymbol{A}$, *pNimf* is reduced a CMF method (NMMF) [9]. Thus, *pNimf* can be seen as subsuming NMMF.

The optimization problem for estimating factor matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ is summarized as follows:

$$\begin{aligned} &\arg\max_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \ \mathcal{L}(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}) = \log p(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{B},\boldsymbol{V}), \\ &\text{s.t. } \boldsymbol{A} \geq 0, \boldsymbol{B} \geq 0, \boldsymbol{C} \geq 0, \boldsymbol{C} = \boldsymbol{V}^T\boldsymbol{A} \end{aligned} \tag{5}$$

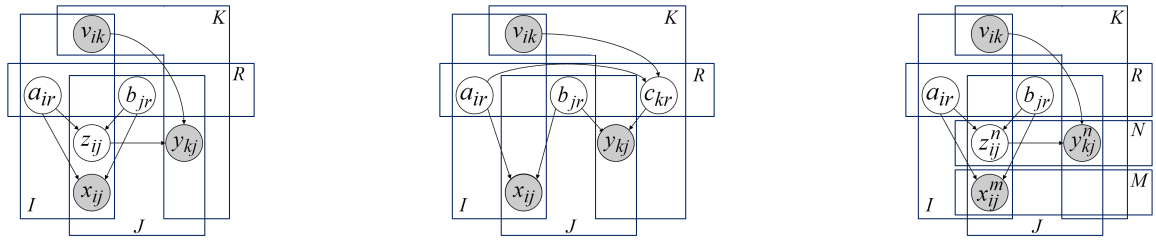


Fig. 3 Graphical models. Shaded nodes indicate observed variables. Figure (a) presents the original definition of the proposed model described in Eq. (2). By marginalizing out $\boldsymbol{Z}$, Fig. (b), which is given by Eq. (3), is obtained. Figure (c) represents the generalized model stated in §4.2.

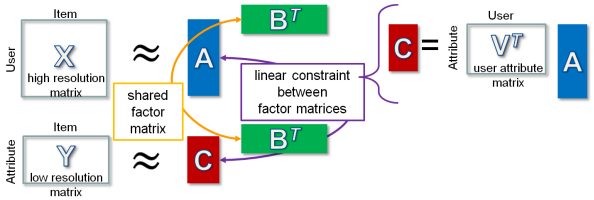


Fig. 4 Factorization form that corresponds to Fig. 3 (b).

where $\boldsymbol{A} \geq 0$ means that all elements of $\boldsymbol{A}$ are nonnegative. Note that for the above optimization problem, Eq. (5) is equivalent to

$$\begin{aligned} &\arg\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \{\mathcal{D}_{KL}(\boldsymbol{X}|\hat{\boldsymbol{X}}) + \mathcal{D}_{KL}(\boldsymbol{Y}|\hat{\boldsymbol{Y}})\}, \\ &\text{s.t. } \boldsymbol{A} \geq 0, \boldsymbol{B} \geq 0, \boldsymbol{C} \geq 0, \boldsymbol{C} = \boldsymbol{V}^T\boldsymbol{A} \end{aligned} \tag{6}$$

where $\mathcal{D}_{KL}$ is generalized KL divergence.

$$\mathcal{D}_{KL}(\boldsymbol{X}|\hat{\boldsymbol{X}}) = \sum_{i,j=1}^{I,J} x_{ij} \log \frac{x_{ij}}{\hat{x}_{ij}} - x_{ij} + \hat{x}_{ij}. \tag{7}$$

3.3 Algorithm

As shown in the next subsection, the following algorithm can be used to solve the optimization problem posed by Eq. (5).

$$a_{ir}^{\text{new}} \leftarrow a_{ir} \frac{\left(\sum_j \frac{x_{ij}}{\hat{x}_{ij}} b_{jr} + \sum_k \sum_j v_{ik} \frac{y_{kj}}{\hat{y}_{kj}} b_{jr}\right)}{\sum_j b_{jr} + \sum_k \sum_j v_{ik} b_{jr}}, \tag{8}$$

$$b_{jr}^{\text{new}} \leftarrow b_{jr} \frac{\left(\sum_i \frac{x_{ij}}{\hat{x}_{ij}} a_{ir} + \sum_k \frac{y_{kj}}{\hat{y}_{kj}} c_{kr}\right)}{\sum_i a_{ir} + \sum_k c_{kr}}, \tag{9}$$

$$c_{kr}^{\text{new}} \leftarrow \sum_i v_{ik} a_{ir}. \tag{10}$$

Update rules for $\boldsymbol{A}$, $\boldsymbol{B}$ are given in "multiplicative form". The right hand side of the update for $\boldsymbol{A}$ is (I) always nonnegative and (II) equals a_{ir} when $x_{ij} = \hat{x}_{ij}$ and $y_{kj} = \hat{y}_{kj}$. By iteratively updating the parameters following Eqs. (8)–(10) from their initial values, the algorithm converges to (local) minima; proof is provided in §4. Pseudo code of the method is shown in Algorithm 1. Note that an almost analogous algorithm is derived when matrix $\boldsymbol{X}$ and/or $\boldsymbol{Y}$ has missing values.

Algorithm 1 probabilistic nonnegative inconsistent resolution matrix factorization (*pNimf*)

Input: $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{V}$: input data, R: rank of approximation
Output: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$: factor matrices
1: initialization for $\boldsymbol{A}, \boldsymbol{B}$ and set $\boldsymbol{C} = \boldsymbol{V}^T\boldsymbol{A}$.
2: **repeat**
3: Update $\boldsymbol{A}$ and $\boldsymbol{C}$ by Eqs. (8) and (10)
4: Update $\boldsymbol{B}$ by Eq. (9)
5: **until** a stopping condition is met

3.4 Algorithm Derivation

In this subsection, we derive the multiplicative update rules given by Eqs. (8), (9), and (10). We define the function $\mathcal{F}(\boldsymbol{A}, \boldsymbol{B})$, where constant terms of the objective function in Eq. (6) are removed and matrix $\boldsymbol{C}$ is replaced by $\boldsymbol{V}^T\boldsymbol{A}$ as follows:

$$\begin{aligned} \mathcal{F}(\boldsymbol{A}, \boldsymbol{B}) = &\sum_{i,j} \left\{ \left(\hat{x}_{ij} - x_{ij} \log(\hat{x}_{ij}) \right) \right\} \\ &+ \sum_{k,j} \left\{ \hat{y}_{kj} - y_{kj} \log(\hat{y}_{kj}) \right\}. \end{aligned} \tag{11}$$

We minimize $\mathcal{F}(\boldsymbol{A}, \boldsymbol{B})$ following the optimization scheme of majorization minimization (MM) [12], [13], similar to [3]. Let us define the auxiliary (majorizing) function $\mathcal{F}^+$ as

$$\begin{aligned} &\mathcal{F}^+(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{S}, \boldsymbol{T}) \\ &= \sum_{i,j} \left\{ \left(\hat{x}_{ij} - x_{ij} \sum_{r=1}^{R} s_{ijr} \log\left(\frac{a_{ir} b_{jr}}{s_{ijr}} \right) \right) \right\} \\ &\quad + \sum_{k,j} \left\{ \hat{y}_{kj} - y_{kj} \sum_{r=1}^{R} t_{kjr} \log\left(\frac{(\sum_i v_{ik} a_{ir}) b_{jr}}{t_{kjr}} \right) \right\}, \end{aligned} \tag{12}$$

where $\boldsymbol{S} = \{s_{ijr}\}$ and $\boldsymbol{T} = \{t_{ijr}\}$ are auxiliary variables satisfying $\sum_r s_{ijr} = 1$, $\sum_r t_{kjr} = 1$ $(\forall (k, j))$. It can be verified that auxiliary function $\mathcal{F}^+$ has the following two properties:

$$\begin{aligned} &1. \quad \mathcal{F}(\boldsymbol{A}, \boldsymbol{B}) \leq \mathcal{F}^+(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{S}, \boldsymbol{T}) \\ &2. \quad \mathcal{F}(\boldsymbol{A}, \boldsymbol{B}) = \min_{\boldsymbol{S},\boldsymbol{T}} \mathcal{F}^+(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{S}, \boldsymbol{T}) \end{aligned} \tag{13}$$

Note that the equality of Eq. (13) holds if and only if

$$s_{ijr} = \frac{a_{ir}b_{jr}}{\sum_{r'=1}^{R} a_{ir'}b_{jr'}}, \quad t_{kjr} = \frac{(\sum_i v_{ik}a_{ir})b_{jr}}{\sum_{r'=1}^{R}(\sum_i v_{ik}a_{ir'})b_{jr'}}. \quad (14)$$

Since the partial derivative of $\mathcal{F}^+$ w.r.t. $\boldsymbol{A}$ is given by

$$\frac{\partial \mathcal{F}^+}{\partial a_{ir}} = \sum_j b_{jr} + \sum_{k,j} v_{ik}b_{jr} - \sum_j \frac{x_{ij}s_{ijr}}{a_{ir}} - \sum_{j,k} \frac{v_{ik}y_{kj}t_{kjr}}{\sum_{i'} v_{i'k}a_{i'r}},$$

the necessary condition of the local minima, $\frac{\partial \mathcal{F}^+}{\partial a_{ir}} = 0$, can be simplified to

$$a_{ir} = \frac{\sum_j x_{ij}s_{ijr} + \sum_{j,k} \frac{v_{ik}a_{ir}y_{kj}t_{kjr}}{\sum_i v_{ik}a_{ir}}}{\sum_j b_{jr} + \sum_{k,j} v_{ik}b_{jr}}. \quad (15)$$

By substituting Eq. (14) into Eq. (15), we obtain the multiplicative update rules for $\boldsymbol{A}$ given by Eq. (8). We omit the derivation of the update rules for $\boldsymbol{B}$ since the derivation is exactly same as that of standard NMF. The update for $\boldsymbol{C}$ is given by the linear constraint.

4. Further Analysis

4.1 Theoretical Analysis

Here we confirm the convergence property of the algorithm.

Theorem *Objective function $\mathcal{F}(\boldsymbol{A}, \boldsymbol{B})$ is monotonically decreasing under the update by Eqs. (8), (9) and (10). The divergence is invariant if and only if $\boldsymbol{A}, \boldsymbol{B}$ are at a stationary point.*

This theorem indicates that the algorithm reaches a local minimum by update iteration. The theorem is proven by showing that $\mathcal{F}^+$ decreases with each optimization step. We need to prove the following two lemmas to prove the theorem.

Lemma 1 *$\mathcal{F}^+$ is a convex function w.r.t. $\boldsymbol{A}$ and thus $\boldsymbol{A}$ satisfying Eq. (15) is the global minimum if the other parameters are fixed.*

proof *Since $-\log(a_{ir})$ is convex and the sum of convex functions is convex, we need to show $-\log(\sum_i v_{ik}a_{ir})$ is convex. Since its Hessian is given by*

$$-\frac{\partial^2 \log(\sum_i v_{ik}a_{ir})}{\partial a_{ir}\partial a_{i'r'}} = \delta_{rr'}\frac{v_{ik}v_{i'k}}{(\sum_i a_{ir})^2},$$

where $\delta_{rr'} = 1$ if $r = r'$ and 0 otherwise, it can be expressed by, using a non-degenerate matrix W, W^TW. Therefore, Hessian is positive definite, and thus convex. □

Lemma 2 *The objective $\mathcal{F}^+(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{S}, \boldsymbol{T})$ is minimized w.r.t. $\boldsymbol{S}$ and $\boldsymbol{T}$ when $\boldsymbol{S}$ and $\boldsymbol{T}$ equals Eq. (14) and $\mathcal{F}(\boldsymbol{A}, \boldsymbol{B}) = \min_{\boldsymbol{S},\boldsymbol{T}} \mathcal{L}^+(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{S}, \boldsymbol{T})$ holds.*

proof *By applying Jensen's inequality to the term in Eq. (11),*

$$-\log(\hat{x}_{ij}) \le -\sum_r s_{ijr}\log(\frac{a_{ir}b_{jr}}{s_{ijr}}),$$
$$-\log(\hat{y}_{kj}) \le -\sum_r t_{kjr}\log(\frac{c_{kr}b_{jr}}{t_{kjr}})$$

holds, and since Eq. (14) is the equality condition, this concludes the proof. □

The theorem follows from the application of the above lemmas.

proof *Let us denote the parameter and the auxiliary variables that satisfy $\mathcal{F}(\boldsymbol{A}, \boldsymbol{B}) = \mathcal{F}^+(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{S}, \boldsymbol{T})$ as $\boldsymbol{A}^{old}$, $\boldsymbol{B}^{old}$, $\boldsymbol{S}^{old}$, $\boldsymbol{T}^{old}$. We also denote $\boldsymbol{A}$ after the first step of the MM given by Eq. (15) as $\boldsymbol{A}^{new}$, and $\boldsymbol{S}$ and $\boldsymbol{T}$ after the second step given by Eq. (14) as $\boldsymbol{S}^{new}$ and $\boldsymbol{T}^{new}$, respectively. From lemma 1 and lemma 2,*

$$\mathcal{F}^+(\boldsymbol{A}^{new}, \boldsymbol{S}^{old}, \boldsymbol{T}^{old}) \le \mathcal{F}^+(\boldsymbol{A}, \boldsymbol{S}^{old}, \boldsymbol{T}^{old}) \ (\forall \boldsymbol{A}),$$
$$\mathcal{F}^+(\boldsymbol{A}^{new}, \boldsymbol{S}^{new}, \boldsymbol{T}^{new}) \le \mathcal{F}^+(\boldsymbol{A}^{new}, \boldsymbol{S}, \boldsymbol{T}) \ (\forall \boldsymbol{S}, \boldsymbol{T}).$$

Note that we omit the notation of $\boldsymbol{B}$. Since $\mathcal{F}(\boldsymbol{A}^{old}) = \mathcal{F}^+(\boldsymbol{A}^{old}, \boldsymbol{S}^{old}, \boldsymbol{T}^{old})$ and $\mathcal{F}(\boldsymbol{A}^{new}) = \mathcal{L}^+(\boldsymbol{A}^{new}, \boldsymbol{S}^{new}, \boldsymbol{T}^{new})$, $\mathcal{F}(\boldsymbol{A}^{new}) \le \mathcal{F}(\boldsymbol{A}^{old})$ holds. Since proof for the update of $\boldsymbol{B}$ is analogous, this completes the proof. □

4.2 Generalization of Algorithms

We now explain the more general scenario that *pNimf* can be applied to. In §3.1, we gave the example in which both high-resolution data and low-resolution data are one month purchase logs. However, as long as assumptions (A1) and (A2) are satisfied, *pNimf* could be applied to any problem with theoretical support. Moreover, *pNimf* can deal with multiple high-resolution and low-resolution data by generalizing the data generative process. Let M be the number of high-resolution data entries and $\boldsymbol{X}^m = \{x_{ij}^m\}$ is the m-th high-resolution matrix. Similarly, let N be the number of low-resolution data entries and $\boldsymbol{Y}^n = \{y_{kj}^n\}$ is the n-th low-resolution matrix. Each m (or n) does not need correspond to a period of time, e.g. day, week and month (unlike the previous example) and it may instead be an indicator of location such as prefecture and country in which the data was collected. By the extending data generative process represented by Fig. 3 (a) to Fig. 3 (c), the estimation procedure is obtained by slight modification of the update rules given by Eqs. (8) and (9) as follows:

$$a_{ir}^{\text{new}} \leftarrow a_{ir}\frac{\left(M\sum_j \frac{\bar{x}_{ij}}{\hat{x}_{ij}}b_{jr} + N\sum_k\sum_j v_{ik}\frac{\bar{y}_{kj}}{\hat{y}_{kj}}b_{jr}\right)}{M\sum_j b_{jr} + N\sum_k\sum_j v_{ik}b_{jr}}, \quad (16)$$

$$b_{jr}^{\text{new}} \leftarrow b_{jr}\frac{\left(M\sum_i \frac{\bar{x}_{ij}}{\hat{x}_{ij}}a_{ir} + N\sum_k \frac{\bar{y}_{kj}}{\hat{y}_{kj}}c_{kr}\right)}{M\sum_i a_{ir} + N\sum_k c_{kr}}, \quad (17)$$

where, $\bar{x}_{ij} = \frac{1}{M}\sum_{m=1}^{M} x_{ij}^m$, $\bar{y}_{kj} = \frac{1}{N}\sum_{n=1}^{N} y_{kj}^n$.

4.3 Choice of Distribution/Divergence

This section discusses the validity of our algorithm derivation when a different distribution or loss function is used. As we have seen in §3.2, the "closed under the summation" property of the Poisson distribution is the key to our derivation. Thus, our derivation is valid if we chose a distribution that also has this property, e.g., a Gaussian distribution. For example, when a Gaussian distribution is adopted for modeling the probability of generating matrix $\boldsymbol{X}$ as in

$$P(\boldsymbol{X}|\boldsymbol{A},\boldsymbol{B}) = \prod_{i,j=1}^{I,J} \mathcal{N}(x_{ij}|\hat{x}_{ij},\sigma^2), \tag{18}$$

a derivation analogous to that in §3.2 leads to the following optimization problem:

$$\begin{aligned} &\arg\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \{\mathcal{D}_{EU}(\boldsymbol{X}|\hat{\boldsymbol{X}};\mathbf{1}_I) + \mathcal{D}_{EU}(\boldsymbol{Y}|\hat{\boldsymbol{Y}};\boldsymbol{\alpha})\}, \\ &\text{s.t. } \boldsymbol{A} \geq 0, \boldsymbol{B} \geq 0, \boldsymbol{C} \geq 0, \boldsymbol{C} = \boldsymbol{V}^T\boldsymbol{A}, \end{aligned} \tag{19}$$

where σ is the standard deviation, $\mathbf{1}_I$ is a size I vector with all-ones and $\boldsymbol{\alpha} = \{\alpha_k\}_{k=1}^K (\alpha_k \geq 0)$ is a weight parameter, and $\mathcal{D}_{EU}$ is the (weighted) Euclidean distance:

$$\mathcal{D}_{KL}(\boldsymbol{X}|\hat{\boldsymbol{X}};\alpha) = \sum_{i,j=1}^{I,J} \alpha_i (x_{ij} - \hat{x}_{ij})^2. \tag{20}$$

5. Experiment

5.1 Setting

We evaluate the performance of our method using synthetic data and real purchase log data.

Synthetic data: We constructed matrices with sizes of $I = 100$, $J = 100$, $K = 10$ using the probabilistic model given by Eq. (2). We prepared $\boldsymbol{V}$ whose elements $v_{ik} = 1$ if k is equal to the quotient of i/K and $v_{ik} = 0$ otherwise. Matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are generated by Gamma distribution and high/low resolution matrixes $\boldsymbol{X}$ and $\boldsymbol{Y}$ were prepared with different levels of sparsity.

Real purchase log data: We use consumer panel research data "SCI" provided by Intage Inc. as the real purchase log data. We use purchase logs of daily necessities (such as milk, coffee and snacks) from 2013.1.1 to 2013.12.31 in Japan. Thus, we can expect that (A2) is satisfied since these items likely to be purchased each month repeatedly. SCI includes user's attribute information such as age, sex and job. We construct Two-month data and Four-month data as follows. Two-month data is constructed using the log entries of Nov. and Dec. as in Fig. 2. We use only the logs of active users who have a purchase entry in each month to satisfy (A1) and items that appear more than ten times. The size and sparseness of $\boldsymbol{X}^{\text{Dec}}$ and $\boldsymbol{Y}^{\text{Nov}}$ are $I = 1589$, $J = 3164$, $K = 34$, 99.15% and 54.4%, respectively. We repeat this procedure for the logs of Jan. and Feb. Size and sparseness are almost similar to those of Nov. and Dec. Four-month data is also prepared in an analogous manner. For Four-month data, we used the logs of Sep, Oct, Nov and Dec and made high-resolution matrix $\boldsymbol{X}^{\text{Dec}}$ and low-resolution matrices $\boldsymbol{Y}^{\text{Nov}}$, $\boldsymbol{Y}^{\text{Oct}}$ and $\boldsymbol{Y}^{\text{Sep}}$. The resulting size was $I = 1288$, $J = 4842$, $K = 34$.

Evaluation Measure: In our experiments, we used a test set log likelihood to evaluate performance. We split the elements of matrix $\boldsymbol{X}$ into a training dataset and a test dataset and computed the log likelihood of the elements in the test. Test data were treated as missing values in the training phase. Log likelihood of the test data set is defined as $\frac{1}{|\mathcal{T}|}\sum_{(i,j)\in\mathcal{T}} \log \mathcal{PO}(x_{ij}|\hat{x}_{ij})$, where $\mathcal{T}$ is the set of element indexes in the test data and $|\cdot|$ indicates the number of elements in the set. We prepared 10 pairs of training and test datasets by randomly extracting 5% of non-zero elements as the test data.

Baseline Methods: For comparison, we considered the following methods. (1) NMF [2], traditional method which uses only high-resolution matrix $\boldsymbol{X}$. (2) NMMF [9], an NMF-based state-of-the-art CMF method that uses both $\boldsymbol{X}$ and $\boldsymbol{Y}$. The weight parameter of NMMF is chosen from the candidates $\alpha = 0.1, 0.5, 1.0$. We report the result for $\alpha = 1.0$ since it yielded the best result among the candidates.

5.2 Results

Table 1 shows the results for the synthetic data. Although the three methods have comparable performance when the sparseness of $\boldsymbol{X}$ is 50% and $R = 5$ and 10, NMMF and *pNimf* outperform NMF when the sparseness is 90%, and *pNimf* is superior to NMMF when the sparseness is 99%. This indicates that proposed method has better performance when the input matrix is very sparse.

Next, Table 2 shows the results using Two-month data. It confirms that *pNimf* and NMMF outperform NMF regardless of the number of factors for all datasets. This result indicates that the use of low-resolution data improves the performance. Moreover, the performance of *pNimf* is superior to that of NMMF in all settings. It seems that the linear relation between factor matrices using user's attribute information supports *pNimf* in handling the difference in resolution

Table 1 Results from synthetic data: test log likelihood for $\boldsymbol{X}$ determined with different sparseness values. Average and standard deviation are shown. Larger values are better. Scores and standard deviation are divided by ten in the 99% sparseness setting.

Sparseness	R	NMF	NMMF	*pNimf*
$\boldsymbol{X}$: 50% $\boldsymbol{Y}$: 10%	5	-2.77(±0.17)	-2.71(±0.10)	**-2.66**(±0.09)
	10	-2.72(±0.18)	-2.54(±0.09)	**-2.42**(±0.04)
	20	-16.7(±20.1)	-3.11(±0.18)	**-3.09**(±0.21)
$\boldsymbol{X}$: 90% $\boldsymbol{Y}$: 40%	5	-6.71(±2.13)	-4.11(±0.97)	**-3.48**(±0.69)
	10	-5.26(±0.92)	-3.28(±0.45)	**-2.96**(±0.23)
	20	-21.5(±4.38)	-7.57(±1.28)	**-5.72**(±0.48)
$\boldsymbol{X}$: 99% $\boldsymbol{Y}$: 80%	5	-5.44(±1.82)	-3.62(±1.73)	**-2.04**(±1.12)
	10	-6.64(±3.52)	-7.05(±2.62)	**-4.59**(±1.67)
	20	-17.0(±32.7)	-7.43(±3.37)	**-6.64**(±3.13)

Table 2 Result of two-month data: test set log-likelihood with various numbers of factors, R. Average and standard deviations are presented. Larger values are better.

Data	R	NMF	NMMF	*pNimf*
SCI	10	-13.3±0.63	-7.89±0.38	**-7.84**±0.17
$\boldsymbol{X}^{\text{Feb}}$&	20	-18.7±1.95	-9.14±0.48	**-8.81**±0.20
$\boldsymbol{Y}^{\text{Jan}}$	50	-32.7±2.15	-11.8±0.48	**-10.4**±0.33
SCI	10	-13.9±0.99	-8.24±0.59	**-7.83**±0.23
$\boldsymbol{X}^{\text{Dec}}$&	20	-14.4±0.73	-8.95±0.37	**-8.68**±0.27
$\boldsymbol{Y}^{\text{Nov}}$	50	-32.59±2.17	-12.32±0.50	**-10.45**±0.30

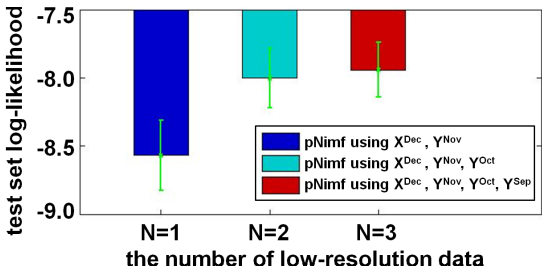


Fig. 5 Results from four-month data: test set log-likelihood for various numbers of low-resolution data entries, N.

and thus achieving better factorization results.[†]

We also evaluated the performance achieved while varying the number of low-resolution data entries, N, using the Four-month data. We set the number of factors to ten and compared it to *pNimf* using a different number of low-resolution data entries, N. Figure 5 shows the results. As the number of low-resolution data, N, increases, the performance of *pNimf* improves. Since *pNimf* can deal with multiple low-resolution data entries by the generalization provided by the data generative process, it works well even if the high-resolution and low-resolution data have different sizes.

6. Extensions

In the previous section, we focused on inconsistent resolution dataset analysis with two assumptions (A1) and (A2). The point to note here is that since the relationship that can be introduced between the high resolution matrix and the low resolution matrix can change depending on the problem setting, the proposed probabilistic model may need some modification for some problems in inconsistent resolution dataset analysis. However, we think that many types of inconsistent resolution dataset analysis can be solved by extending the model. Accordingly, we show examples of how new methods can be developed by extending the proposed model.

6.1 Formulation

Here we consider inconsistent resolution dataset analysis for the case where the member/non-member purchase logs are created by processes different from those indicated by § 3.1. In the example of § 3.1, since we assumed that membership cards were issued from December, we were able to satisfy (A1) common user set assumption that users in November and users in December were the same. However, for data collected at shops that can be used by non-member users, such as convenience stores, assumption (A1) no longer holds since the members and the non-members always exist together and represent different user groups. An explanation for this is made below using Fig. 6.

Figure 6 shows examples of possible situations when members and non-member users are different. The difference between Fig. 6 (a) and (b) is whether the population for each attribute in members and non-members is almost the same or very different. The proposed method indicated by § 3 has some validity in the case of (a), it loses validity in the case of (b). This is because assumptions (A1) and (A2) imply that "the total purchase amount of each item for each attribute is the almost same for members and non-members" and it is generally appropriate for (a), whereas it is clearly inappropriate in the setting of (b).

As a new assumption, we consider the approach that introduces a new assumption (A3), the attribute purchase quantity proportionality assumption, that is, "member purchase history is roughly proportional to non-member purchase history". Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be a high resolution matrix representing members' log, and a low resolution matrix representing non-members' log, respectively. Since the members' purchase log of attribute k is $\sum_{i=1}^{I} v_{ik}\boldsymbol{x}_i$, and the non-members' purchase log is $\boldsymbol{y}_k$, the proportional relation of assumption (A3) is represented by the equation $\boldsymbol{y}_k \propto \sum_{i=1}^{I} v_{ik}\boldsymbol{x}_i$.

If there are a certain number of members with attribute k, it is considered quite natural to make this assumption. Therefore, we consider a factorization form that holds this proportional relation on between $\hat{\boldsymbol{X}}$, $\hat{\boldsymbol{Y}}$. By defining the diagonal matrix $\boldsymbol{D} := diag(\{d_k\}_{k=1}^{K})$ whose elements d_k represent the proportionality constant of attribute k, the following equation holds given the proportional relationship:

$$\hat{\boldsymbol{Y}} = \boldsymbol{D}\boldsymbol{V}^T\hat{\boldsymbol{X}}. \tag{21}$$

Using factorization form $\hat{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{B}^T$ for $\hat{\boldsymbol{X}}$ and substituting it into Eq. (21), the factorization form for $\boldsymbol{Y}$ becomes

$$\hat{\boldsymbol{Y}} = \boldsymbol{D}\boldsymbol{C}\boldsymbol{B}^T, \quad \boldsymbol{C} = \boldsymbol{V}^T\boldsymbol{A}. \tag{22}$$

Summarizing the above yields the following probabilistic model:

$$p(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{V}) = \prod_{i,j} \mathcal{PO}(x_{ij}|\hat{x}_{ij}) \prod_{k,j} \mathcal{PO}(\beta y_{kj}|\beta\hat{y}_{kj}),$$

$$\text{where} \quad \hat{y}_{kj} = \sum_{r=1}^{R} d_k c_{kr} b_{jr} \quad \text{and} \quad c_{kr} = \sum_{i=1}^{I} v_{ik} a_{ir},$$

where β is a weight parameter that controls the contribution

[†]These results are consistent with the synthetic data results since the sparseness of Two-month data is ($\boldsymbol{X}$: 99.15%, $\boldsymbol{Y}$: 54.4%) and this corresponds to that between ($\boldsymbol{X}$: 99%, $\boldsymbol{Y}$: 80%) and ($\boldsymbol{X}$: 90%, $\boldsymbol{Y}$: 40%).

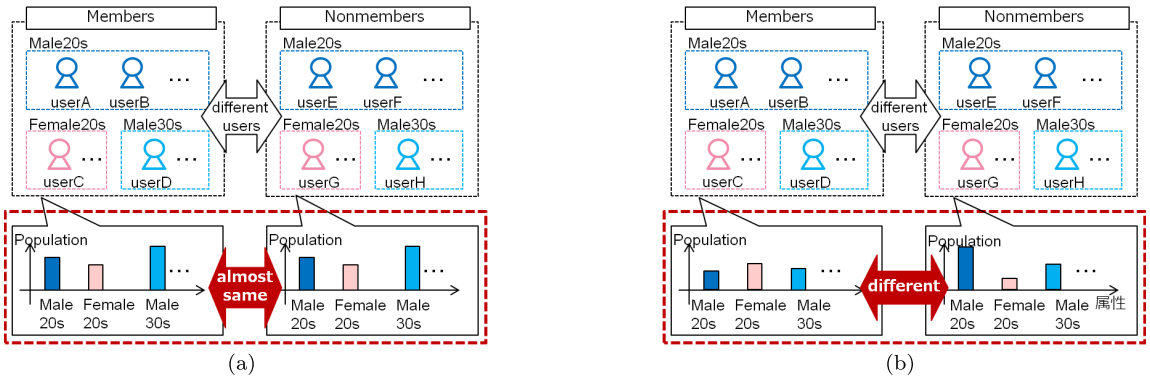


Fig. 6 An example where (a) members and non-members have almost the same population and (b) members and non-members have different populations. Proposed method shown in § 3 can be applied to the case of (a). However, it is not appropriate for the case of (b) since the members and non-members have greatly different purchase volumes.

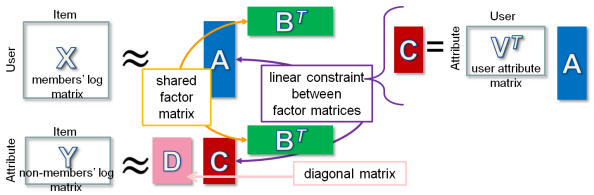


Fig. 7 Factorization form of the extended method.

Algorithm 2 extended model of *pNimf*

Input: $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{V}$: input data, R: rank of approximation
Output: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$: factor matrices

1: initialization for $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{D}$ and set $\boldsymbol{C} = \boldsymbol{V}^T \boldsymbol{A}$.
2: **repeat**
3: Update $\boldsymbol{A}$ and $\boldsymbol{C}$ by Eqs. (23) and (25)
4: Update $\boldsymbol{B}$ by Eq. (24)
5: Update $\boldsymbol{D}$ by Eq. (26)
6: **until** a stopping condition is met

of non-member data. Figure 7 shows the factorization form of this model. The difference from the factorization form shown in Fig. 4 of §3 is the existence of diagonal matrix $\boldsymbol{D}$ and thus the former is an extended factorization form. Parameter update rules for a_{ir}, b_{jr}, c_{kr}, d_k are derived as follows:

$$a_{ir}^{\text{new}} \leftarrow a_{ir} \frac{\left(\sum_j \frac{x_{ij}}{\hat{x}_{ij}} b_{jr} + \beta \sum_k d_k v_{ik} \sum_j \frac{y_{kj}}{\hat{y}_{kj}} b_{jr}\right)}{\sum_j b_{jr} + \beta \sum_k d_k v_{ik} \sum_j b_{jr}}, \tag{23}$$

$$b_{jr}^{\text{new}} \leftarrow b_{jr} \frac{\left(\sum_i \frac{x_{ij}}{\hat{x}_{ij}} a_{ir} + \beta \sum_k \frac{y_{kj}}{\hat{y}_{kj}} d_k c_{kr}\right)}{\sum_i a_{ir} + \beta \sum_k d_k c_{kr}}, \tag{24}$$

$$c_{kr}^{\text{new}} \leftarrow \sum_i v_{ik} a_{ir}, \tag{25}$$

$$d_k^{\text{new}} \leftarrow \frac{\sum_j y_{kj}}{\sum_r c_{kr} (\sum_j b_{jr})}. \tag{26}$$

Pseudo code is shown in Algorithm 2.

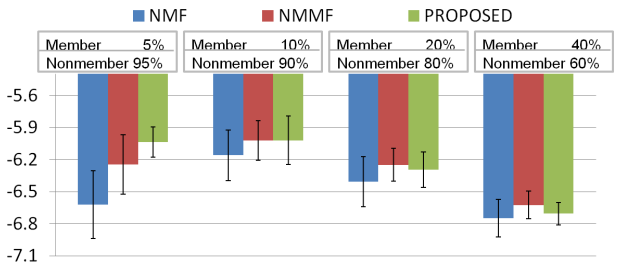


Fig. 8 Test set log-likelihood for various ratios of members and non-members. Blue, red and green histograms represent the results of NMF, NMMF and proposed method, respectively.

6.2 Experiment

We again use consumer panel research data "SCI" provided by Intage Inc. as the real purchase log data. We constructed members' log matrix $\boldsymbol{X}$ by randomly extracting 5%, 10%, 20%, 40% users and non-members' log matrix $\boldsymbol{Y}$ by using the remaining users' log. We used the set of users and items that appeared more than 50 and 30 times in the log; users and items had sizes of 5000 and 7000, respectively.

We used exactly the same evaluation measure and baseline methods as used in §5. In order to treat members' log and non-members' log equally, we set their ratio (members/non-members) as a weight parameter, i.e., $\beta = 0.05, 0.11, 0.25$, and 0.66.

Figure 8 shows the results. Regardless of the members' ratio, proposed method and NMMF are superior to NMF. This indicates that the use of the non-members' log data contributed to improving the performance. Moreover, it is confirmed that the improvement is small when the member's ratio is 40%, i.e., the number of members is large, and that the improvement is large when the member's ratios are 5%, 10%, and 20%, i.e., the number of members is small. Comparing the proposed method with NMMF, the performances are comparable when the member's ratios are 10%, 20%, and 40%, but the proposed method outperforms NMMF when the ratio is 5%. This indicates that the use of the proportional relation between the members' and non-

members' log data raises the performance when the number of members is very small.

7. Conclusion

In this paper, we proposed a new method for inconsistent resolution dataset analysis. By considering the data generative process using the latent high resolution matrix, we proposed a new probabilistic model *pNimf*. Furthermore, we also showed that extended variant of *pNimf* is constructed under the setting requiring different assumptions. These results show that the proposed method can be a fundamental approach for inconsistent resolution dataset analysis.

The remaining research topics include further expansion of the model by, for example, introducing seasonality. We also need to analyze how the difference of the grain sizes (difference in the number of lines) between the high and low resolution matrices affects the degree of performance improvement.

References

[1] M. Kohjima, T. Matsubayashi, and H. Sawada, "Probabilistic non-negative inconsistent-resolution matrices factorization," In Proceedings of the 24th ACM international conference on Information and Knowledge Management, pp.1855–1858, 2015.
[2] D.D. Lee and H.S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol.401, no.6755, pp.788–791, 1999.
[3] D.D. Lee and H.S. Seung, Algorithms for non-negative matrix factorization, In Advances in neural information processing systems, pp.556–562, 2001.
[4] A. Cichocki, R. Zdunek, A.H. Phan, and S. Amari, Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation, John Wiley & Sons, 2009.
[5] T. Hofmann, "Probabilistic latent semantic indexing," In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp.50–57, 1999.
[6] D.M. Blei, A.Y. Ng, and M.I. Jordan, Latent dirichlet allocation, the Journal of machine Learning research, vol.3, pp.993–1022, 2003.
[7] C. Ding, T. Li, and W. Peng, "On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing," Computational Statistics & Data Analysis, vol.52, no.8, pp.3913–3927, 2008.
[8] A.P. Singh and G.J. Gordon, "Relational learning via collective matrix factorization," In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.650–658, 2008.
[9] K. Takeuchi, K. Ishiguro, A. Kimura, and H. Sawada, Non-negative multiple matrix factorization, In Proceedings of the 23rd international joint conference on Artificial Intelligence, pp.1713–1720, 2013.
[10] H. Lee and S. Choi, Group nonnegative matrix factorization for EEG classification, In International Conference on Artificial Intelligence and Statistics, pp.320–327, 2009.
[11] Y. Aimoto and H. Kashima, "Matrix factorization with aggregated observations," In Advances in Knowledge Discovery and Data Mining, pp.521–532, Springer, 2013.
[12] D.R. Hunter and K. Lange, "A tutorial on mm algorithms," The American Statistician, vol.58, no.1, pp.30–37, 2004.
[13] J. De Leeuw, "Block-relaxation algorithms in statistics," In Information systems and data analysis, pp.308–324, Springer, 1994.

Masahiro Kohjima received the B.E. and M.E. degrees in computer science from Tokyo Institute of Technology, Tokyo, Japan, in 2009 and 2012, respectively. He joined NTT Corporation in 2012 and is currently working at NTT Service Evolution Laboratories. His research interests lie in the area of machine learning with emphasis on probabilistic models, Bayesian methods and reinforcement learning.

Tatsushi Matsubayashi received the B.S. degree in Physics from Kyoto University, Kyoto, Japan, in 2000. He received the M.S and Ph.D. degrees in Astrophysics from Tokyo Institute of Technology, Tokyo, Japan, in 2002 and 2006, respectively. He joined NTT Corporation in 2005 and is currently working at NTT Service Evolution Laboratories. His research interests lie in the area of high-performance computing, information visualization, and machine learning. He is a member of the IPSJ.

Hiroshi Sawada received the B.E., M.E. and Ph.D. degrees in information science from Kyoto University, Kyoto, Japan, in 1991, 1993 and 2001, respectively. He joined NTT Corporation in 1993. He is now an executive manager at the NTT Communication Science Laboratories, Kyoto, Japan. His research interests include statistical signal processing, audio source separation, array signal processing, machine learning, latent variable model, graph-based data structure, and computer architecture. He is an IEEE Fellow, an IEICE Senior Member, and a member of the ASJ.