Matrix Co-Factorization on Compressed Sensing

Jiho Yoo[1] and **Seungjin Choi**[1,2]
[1] Department of Computer Science
[2] Division of IT Convergence Engineering
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea
{zentasis, seungjin}@postech.ac.kr

Abstract

In this paper we address the problem of matrix factorization on compressively-sampled measurements which are obtained by random projections. While this approach improves the scalability of matrix factorization, its performance is not satisfactory. We present a matrix co-factorization method where compressed measurements and a small number of uncompressed measurements are jointly decomposed, sharing a factor matrix. We evaluate the performance of three matrix factorization methods in terms of Cramér-Rao bounds, including: (1) matrix factorization on uncompressed data (MF); (2) matrix factorization on compressed data (CS-MF); (3) matrix co-factorization on compressed and uncompressed data (CS-MCF). Numerical experiments demonstrate that CS-MCF improves the performance of CS-MF, emphasizing the useful behavior of exploiting side information (a small number of uncompressed measurements).

1 Introduction

Matrix factorization is a decomposition method where a data matrix $\boldsymbol{X} \in \mathbb{R}^{M \times N}$ is approximated as a product of two or more factor matrices which uncover the latent structure of the data. The 2-factor decomposition seeks latent factor matrices $\boldsymbol{U} \in \mathbb{R}^{M \times K}$ and $\boldsymbol{V} \in \mathbb{R}^{N \times K}$ which minimize the following objective function

$$\mathcal{J}_{\text{MF}} = \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^\top\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenious norm. When columns in $\boldsymbol{X}$ are treated as data points in M-dimensional space, $\boldsymbol{U}$ and $\boldsymbol{V}$ are referred to be as *basis matrix* and *encoding matrix*, respectively. Exemplary matrix factorization methods include principal component analysis (PCA) [Jolliffe, 2002], nonnegative matrix factorization (NMF) [Lee and Seung, 1999], probabilistic matrix factorization (PMF) [Srebro *et al.*, 2005], to name a few. Matrix factorization has established itself as a powerful technique in various applications such as collaborative prediction [Koren *et al.*, 2009], document clustering [Xu *et al.*, 2003; Yoo and Choi, 2010], music transcription [Smaragdis and Brown, 2003], and brain wave analysis [Lee and Choi, 2009]. In some of these applications such as collaborative prediction, the data matrix is sparse since most of entries are missing or unobserved. Thus, matrix factorization methods can be applied to a large scale problem without much extra care, since only observed elements are used for learning. On the other hand, data matrix is dense in some applications where spectrograms of music or brain wave data are analyzed. In such applications, it is not easy to handle large-scale data.

Compressed sensing is a new sensing/sampling paradigm which ensures near-optimal recovery of sparse signals from a small number of linear measurements [Donoho, 2006; Candes and Tao, 2006]. Compressed sensing is a promising approach to handling a large-scale dense data matrix since it is allowed to perform a processing (learning or data analysis) in the compressed domain (in which a compressively-sampled matrix is much smaller than the original input data matrix in size) rather than in the ambient space. Let $\boldsymbol{Y} \in \mathbb{R}^{P \times N}$ be the compressively-sampled data matrix which is obtained by $\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{X}$, in which $\boldsymbol{\Phi} \in \mathbb{R}^{P \times M}$ is a linear sensing matrix which is often given by a random projection. A direct application of the compressed sensing framework to matrix factorization, referred to as CS-MF, yields the following objective function

$$\mathcal{J}_{\text{CS-MF}} = \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{V}^\top\|_F^2. \tag{2}$$

CS-MF determines $\boldsymbol{U}$ and $\boldsymbol{V}$ which minimize (2), improving the scalability since $\boldsymbol{Y}$ and $\boldsymbol{\Phi}$ requires less space in storage, compared to $\boldsymbol{X}$. Recently, nonnegative matrix factorization (NMF) in the compressed domain, referred to as CS-NMF, was presented in [O'Grady and Rickard, 2008], where (2) is minimized with nonnegativity constraints imposed on factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ as well as the data matrix $\boldsymbol{X}$. In contrast to the standard compressed sensing framework where $\boldsymbol{U}$ is known in advance, CS-MF or CS-NMF estimates both $\boldsymbol{U}$ and $\boldsymbol{V}$ in the compressed-domain, which degrades the reconstruction performance.

Matrix co-factorization (MCF) is a technique that decompose two or more data matrices jointly, sharing some factor matrices in factorizations. In addition to the data matrix, side information matrices are also considered for the joint decomposition, including label information [Yu *et al.*, 2005; Lee *et al.*, 2010], link information [Zhu *et al.*, 2007], inter-subject variations [Lee and Choi, 2009], and spectrograms

of music [Yoo *et al.*, 2010]. MCF was also applied to the general entity-relationship models [Singh and Gordon, 2008; Yoo and Choi, 2009] to learn the characteristics of each entity from the relationships. In this paper we present a method of MCF on compressively-sampled data, referred to as CS-MCF, where we decompose the compressed data matrix and a small portion of uncompressed original data jointly in order to improve the performance of matrix factorization in the compressed domain. We compute the Cramér-Rao bound (CRB) for matrix factorization on uncompressed data (MF), CS-MF, and CS-MCF in the case of Gaussian likelihood, showing that CS-MCF indeed improves the performance over CS-MF in terms of the reconstruction quality. We consider alternating least squares (ALS) algorithms for MF, CS-MF, and CS-MCF. In addition, we also consider multiplicative updates, as in NMF, comparing NMF, CS-NMF (NMF in the compressed domain), and CS-NMCF (nonnegative matrix co-factorization in the compressed domain with small uncompressed information).

2 Matrix Co-Factorization on Compressively Sampled Data

Compressed sensing [Donoho, 2006; Candes and Tao, 2006] is a new framework which performs the measurement and compression simultaneously to reduce computation required in the measurement process. The fundamental theoretical results of compressed sensing is based on the sparsity of the representation of given data, which makes possible to reduce the required number of measurement less than the amount computed from the classical Shannon-Nyquist sampling theorem. Compressed sensing assumes that the basis which sparsely represents the given data is known, however, the appropriate representation basis for given data is usually not known in advance. In that cases we have to learn a representation matrix from a given data, however, the problem of learning representation basis from the compressed measurement has not been studied much.

Matrix factorization in the compressed domain considers a decomposition of the compressively-sampled data matrix $\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{X}$ where $\boldsymbol{\Phi} \in \mathbb{R}^{P\times M}$ is a random projection. Thus given $\boldsymbol{X}$ and $\boldsymbol{\Phi}$, CS-MF seeks $\boldsymbol{Y} \approx \boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{V}^\top$, determining factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ which minimizes (2). In contrast to the compressed sensing framework where the basis matrix $\boldsymbol{U}$ is known in advance, CS-MF should estimate both $\boldsymbol{U}$ and $\boldsymbol{V}$ given compressed data $\boldsymbol{Y}$. We believe that this explains the poor reconstruction performance of CS-MF in our experiments.

Now we explain matrix co-factorization in the compressed domain with exploiting partial uncompressed data. We partition the data matrix $\boldsymbol{X} \in \mathbb{R}^{M\times N}$ into two sub-matrices $\boldsymbol{X}^c \in \mathbb{R}^{M\times C}$ and $\boldsymbol{X}^u \in \mathbb{R}^{M\times(N-C)}$. Assuming that $C > (N-C)$, the sub-matrix $\boldsymbol{X}^c$ is compressed by a random projection $\boldsymbol{\Phi}$, leading to $\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{X}^c$ with the abuse of notation. CS-MCF considers a joint decomposition of the compressed data matrix $\boldsymbol{Y}$ and partial uncompressed data $\boldsymbol{X}^u$, minimizing the following objective function:

$$\mathcal{J}_{\text{CS-MCF}} = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{V}^\top\|_F^2 + \frac{\lambda}{2}\|\boldsymbol{X}^u - \boldsymbol{U}\boldsymbol{W}^\top\|_F^2, \quad (3)$$

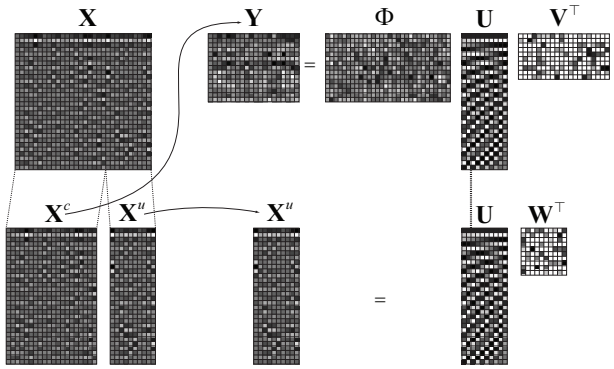


Figure 1: Pictorial illustration of CS-MCF. The input data matrix $\boldsymbol{X}$ is partitioned into $\boldsymbol{X}^c$ (to be compressed) and $\boldsymbol{X}^u$ (partial uncompressed data). Then $\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{X}^c$ and $\boldsymbol{X}^u$ are jointly decomposed, sharing the factor matrix $\boldsymbol{U}$, leading to $\boldsymbol{Y} \approx \boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{V}^\top$ and $\boldsymbol{X}^u \approx \boldsymbol{U}\boldsymbol{W}^\top$.

where the factor matrix $\boldsymbol{U} \in \mathbb{R}^{M\times K}$ is shared, capturing the common characteristics in representing both $\boldsymbol{Y}$ and $\boldsymbol{X}^u$, and $\boldsymbol{V} \in \mathbb{R}^{C\times K}$ and $\boldsymbol{W} \in \mathbb{R}^{(N-C)\times K}$ capture individual characteristics. The parameter λ controls the balance between two decomposition. Pictorial illustration of CS-MCF is given in Fig. 1.

2.1 ALS Updates

ALS updates for $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$ are easily derived by solving

$$\begin{aligned}\frac{\partial \mathcal{J}_{CS-MCF}}{\partial \boldsymbol{U}} &= 0,\\ \frac{\partial \mathcal{J}_{CS-MCF}}{\partial \boldsymbol{V}} &= 0,\\ \frac{\partial \mathcal{J}_{CS-MCF}}{\partial \boldsymbol{W}} &= 0,\end{aligned}$$

for $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$, respectively, and the corresponding gradients of (3) are calculated as

$$\begin{aligned}\frac{\partial \mathcal{J}_{CS-MCF}}{\partial \boldsymbol{U}} &= -\boldsymbol{\Phi}^\top\boldsymbol{Y}\boldsymbol{V} + \boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{V}^\top\boldsymbol{V}\\ &\quad -\lambda\boldsymbol{X}^u\boldsymbol{W} + \lambda\boldsymbol{U}\boldsymbol{W}^\top\boldsymbol{W},\\ \frac{\partial \mathcal{J}_{CS-MCF}}{\partial \boldsymbol{V}} &= -\boldsymbol{Y}^\top\boldsymbol{\Phi}\boldsymbol{U} + \boldsymbol{V}\boldsymbol{U}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U},\\ \frac{\partial \mathcal{J}_{CS-MCF}}{\partial \boldsymbol{W}} &= -\boldsymbol{X}^u\boldsymbol{U} + \boldsymbol{W}\boldsymbol{U}^\top\boldsymbol{U},\end{aligned}$$

leading to ALS updates

$$\begin{aligned}\text{vec}(\boldsymbol{U}) &\leftarrow ((\boldsymbol{V}^\top\boldsymbol{V}) \otimes (\boldsymbol{\Phi}^\top\boldsymbol{\Phi}) + \lambda(\boldsymbol{W}^\top\boldsymbol{W}) \otimes \boldsymbol{I}_M)^{-1}\\ &\quad \text{vec}(\boldsymbol{\Phi}^\top\boldsymbol{Y}\boldsymbol{V} + \lambda\boldsymbol{X}^u\boldsymbol{W}),\\ \boldsymbol{V} &\leftarrow \boldsymbol{Y}^\top\boldsymbol{\Phi}\boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U})^{-1},\\ \boldsymbol{W} &\leftarrow \boldsymbol{X}^{u\top}\boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{U})^{-1},\end{aligned}$$

where $\boldsymbol{I}_M$ is the identity matrix with size $M \times M$, $\text{vec}(\boldsymbol{U}) = [\boldsymbol{u}_1^\top, \ldots, \boldsymbol{u}_K^\top]^\top$ is the 'vec' operator which creates a column vector from the matrix $\boldsymbol{U}$ by stacking columns of $\boldsymbol{U}$, and $\otimes$ denotes the Kronecker product (tensor product).

Table 1: Algorithm description for ALS updates and multiplicative updates.

Model	Updates	Model	Updates
MF	$\boldsymbol{U} \leftarrow \boldsymbol{X}\boldsymbol{V}(\boldsymbol{V}^\top\boldsymbol{V})^{-1}$ $\boldsymbol{V} \leftarrow \boldsymbol{X}^\top\boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{U})^{-1}$	NMF	$\boldsymbol{U} \leftarrow \boldsymbol{U} \odot \frac{\boldsymbol{X}\boldsymbol{V}}{\boldsymbol{U}\boldsymbol{V}^\top\boldsymbol{V}}$ $\boldsymbol{V} \leftarrow \boldsymbol{V} \odot \frac{\boldsymbol{X}^\top\boldsymbol{U}}{\boldsymbol{V}\boldsymbol{U}^\top\boldsymbol{U}}$
CS-MF	$\mathrm{vec}(\boldsymbol{U}) \leftarrow ((\boldsymbol{V}^\top\boldsymbol{V}) \otimes (\boldsymbol{\Phi}^\top\boldsymbol{\Phi}))^{-1}\mathrm{vec}(\boldsymbol{\Phi}^\top\boldsymbol{Y}\boldsymbol{V})$ $\boldsymbol{V} \leftarrow \boldsymbol{Y}^\top\boldsymbol{\Phi}\boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U})^{-1}$	CS-NMF	$\boldsymbol{U} \leftarrow \boldsymbol{U} \odot \frac{\boldsymbol{\Phi}^\top\boldsymbol{Y}\boldsymbol{V}}{\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{V}^\top\boldsymbol{V}}$ $\boldsymbol{V} \leftarrow \boldsymbol{V} \odot \frac{\boldsymbol{Y}^\top\boldsymbol{\Phi}\boldsymbol{U}}{\boldsymbol{V}\boldsymbol{U}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U}}$
CS-MCF	$\mathrm{vec}(\boldsymbol{U}) \leftarrow ((\boldsymbol{V}^\top\boldsymbol{V}) \otimes (\boldsymbol{\Phi}^\top\boldsymbol{\Phi}) + \lambda(\boldsymbol{W}^\top\boldsymbol{W}) \otimes \boldsymbol{I}_M)^{-1}$ $\mathrm{vec}(\boldsymbol{\Phi}^\top\boldsymbol{Y}\boldsymbol{V} + \lambda\boldsymbol{X}^u\boldsymbol{W})$ $\boldsymbol{V} \leftarrow \boldsymbol{Y}^\top\boldsymbol{\Phi}\boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U})^{-1}$ $\boldsymbol{W} \leftarrow \boldsymbol{X}^{u\top}\boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{U})^{-1}$	CS-NMCF	$\boldsymbol{U} \leftarrow \boldsymbol{U} \odot \frac{\boldsymbol{\Phi}^\top\boldsymbol{Y}\boldsymbol{V}+\lambda\boldsymbol{X}^{u\top}\boldsymbol{W}}{\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{V}^\top\boldsymbol{V}+\lambda\boldsymbol{U}\boldsymbol{W}^\top\boldsymbol{W}}$ $\boldsymbol{V} \leftarrow \boldsymbol{V} \odot \frac{\boldsymbol{Y}^\top\boldsymbol{\Phi}\boldsymbol{U}}{\boldsymbol{V}\boldsymbol{U}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U}}$ $\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{X}^{u\top}\boldsymbol{U}}{\boldsymbol{W}\boldsymbol{U}^\top\boldsymbol{U}}$

2.2 Multiplicative Updates

As in NMF, we derive multiplicative updates for $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$ which iteratively minimize (3) with nonnegativity constraints imposed on factor matrices, given the nonnegative data matrix. Our derivation follows the technique used in [Yoo and Choi, 2008] where the gradient of an objective function $\mathcal{J}$ is decomposed as

$$\frac{\partial \mathcal{J}}{\partial \Theta} = \left[\frac{\partial \mathcal{J}}{\partial \Theta}\right]^+ - \left[\frac{\partial \mathcal{J}}{\partial \Theta}\right]^-,$$

where $\left[\frac{\partial \mathcal{L}}{\partial \Theta}\right]^+ > 0$ and $\left[\frac{\partial \mathcal{L}}{\partial \Theta}\right]^- > 0$, and then multiplicative updates for parameters Θ are given by

$$\Theta \leftarrow \Theta \odot \frac{\left[\frac{\partial \mathcal{J}}{\partial \Theta}\right]^-}{\left[\frac{\partial \mathcal{J}}{\partial \Theta}\right]^+},$$

where $\odot$ denotes Hadamard product (element-wise product) and the division is done in an element-wise manner. Applying this technique to the minimization of (3) yields

$$\begin{aligned}
\boldsymbol{U} &\leftarrow \boldsymbol{U} \odot \frac{\boldsymbol{\Phi}^\top\boldsymbol{Y}\boldsymbol{V} + \lambda\boldsymbol{X}^u\boldsymbol{W}}{\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{V}^\top\boldsymbol{V} + \lambda\boldsymbol{U}\boldsymbol{W}^\top\boldsymbol{W}}, \\
\boldsymbol{V} &\leftarrow \boldsymbol{V} \odot \frac{\boldsymbol{Y}^\top\boldsymbol{\Phi}\boldsymbol{U}}{\boldsymbol{V}\boldsymbol{U}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{U}}, \\
\boldsymbol{W} &\leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{X}^u\boldsymbol{U}}{\boldsymbol{W}\boldsymbol{U}^\top\boldsymbol{U}}.
\end{aligned}$$

ALS or multiplicative updating algorithms for MF and CS-MF are also derived in the same way, which are summarized in Table 1.

3 Cramér-Rao Bounds

We calculate the Cramér-Rao bound (CRB) to evaluate the performance of matrix factorization, CS-MF, and CS-MCF. CRB [Kay, 1993] provide the lower bound of the variance of unbiased estimators of a deterministic parameter in the following form,

$$\mathbb{E}\left\{(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top\right\} \geq \mathcal{I}^{-1}, \tag{4}$$

where $\boldsymbol{\theta}$ is the estimated parameter of the model, $\tilde{\boldsymbol{\theta}}$ is the true value of the parameter, $\boldsymbol{A} \geq \boldsymbol{B}$ denotes that $\boldsymbol{A} - \boldsymbol{B}$ is the positive semi-definite matrix, and $\mathcal{I}$ is the *Fisher information matrix*, each element of which is computed by

$$\mathcal{I}_{ij} = \mathbb{E}\left\{-\frac{\partial^2 \log p(\boldsymbol{x}|\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\right\}, \tag{5}$$

where $p(\boldsymbol{x}|\boldsymbol{\theta})$ is the likelihood. In the case of matrix factorization, we have factor matrices $\boldsymbol{U} \in \mathbb{R}^{M\times K}$ and $\boldsymbol{V} \in \mathbb{R}^{N\times K}$ as the parameters for the model, so $\boldsymbol{\theta}$ becomes

$$\begin{aligned}
\boldsymbol{\theta} &= \left[\mathrm{vec}(\boldsymbol{U})^\top, \mathrm{vec}(\boldsymbol{V})^\top\right]^\top \\
&= \left[\boldsymbol{u}_1^\top, \boldsymbol{u}_2^\top, \cdots, \boldsymbol{u}_M^\top, \boldsymbol{v}_1^\top, \boldsymbol{v}_2^\top, \cdots, \boldsymbol{v}_N^\top\right]^\top,
\end{aligned}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ represent the column vector consist of the elements in the i-th and j-th row of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. Then, the Fisher information matrix has the following representation,

$$\begin{aligned}
\mathcal{I} &= \begin{bmatrix} \mathcal{I}_U & \mathcal{I}_{UV} \\ \mathcal{I}_{UV}^\top & \mathcal{I}_V \end{bmatrix} \\
&= \begin{bmatrix}
\mathcal{I}_{\boldsymbol{u}_1} & \cdots & \mathcal{I}_{\boldsymbol{u}_1\boldsymbol{u}_M} & \mathcal{I}_{\boldsymbol{u}_1\boldsymbol{v}_1} & \cdots & \mathcal{I}_{\boldsymbol{u}_1\boldsymbol{v}_N} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\mathcal{I}_{\boldsymbol{u}_M\boldsymbol{u}_1} & \cdots & \mathcal{I}_{\boldsymbol{u}_M} & \mathcal{I}_{\boldsymbol{u}_M\boldsymbol{v}_1} & \cdots & \mathcal{I}_{\boldsymbol{u}_M\boldsymbol{v}_N} \\
\mathcal{I}_{\boldsymbol{v}_1\boldsymbol{u}_1} & \cdots & \mathcal{I}_{\boldsymbol{v}_1\boldsymbol{u}_M} & \mathcal{I}_{\boldsymbol{v}_1} & \cdots & \mathcal{I}_{\boldsymbol{v}_1\boldsymbol{v}_N} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\mathcal{I}_{\boldsymbol{v}_N\boldsymbol{u}_1} & \cdots & \mathcal{I}_{\boldsymbol{v}_1\boldsymbol{u}_M} & \mathcal{I}_{\boldsymbol{v}_N\boldsymbol{v}_1} & \cdots & \mathcal{I}_{\boldsymbol{v}_N}
\end{bmatrix}.
\end{aligned}$$

where each part is computed for the corresponding parameter, for example,

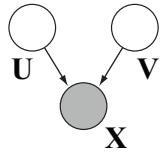
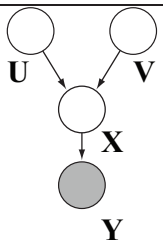
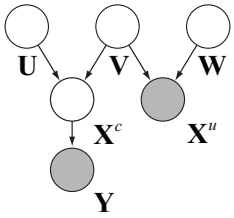
$$\left[\mathcal{I}_{\boldsymbol{u}_i}\right]_{k,k'} = \mathbb{E}\left\{-\frac{\partial^2 \log p(\boldsymbol{X}|\boldsymbol{U},\boldsymbol{V})}{\partial u_{ik}\partial u_{ik'}}\right\}, \tag{6}$$

$$\left[\mathcal{I}_{\boldsymbol{u}_i\boldsymbol{u}_{i'}}\right]_{k,k'} = \mathbb{E}\left\{-\frac{\partial^2 \log p(\boldsymbol{X}|\boldsymbol{U},\boldsymbol{V})}{\partial u_{ik}\partial u_{i'k'}}\right\}, \tag{7}$$

where $[\boldsymbol{A}]_{i,j}$ represents the (i,j)-th element in the matrix $\boldsymbol{A}$. The Fisher information matrix of CS-MCF has a different parametrization with the matrix $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{W}$, which is

$$\mathcal{I} = \begin{bmatrix} \mathcal{I}_U & \mathcal{I}_{UV} & \mathcal{I}_{UW} \\ \mathcal{I}_{UV}^\top & \mathcal{I}_V & \mathcal{I}_{VW} \\ \mathcal{I}_{UW}^\top & \mathcal{I}_{VW}^\top & \mathcal{I}_W \end{bmatrix},$$

Table 2: Graphical model, likelihood, Fisher information matrix (FIM) and Cramér-Rao lower bound for matrix factorization (MF), matrix factorization for compressed measurements (CS-MF), and matrix co-factorization for compressed measurements (CS-MCF).

	MF	**CS-MF**	**CS-MCF**
Graphical Model			
Likelihood	$p(\boldsymbol{X}\|\boldsymbol{U},\boldsymbol{V})$ $= \prod_j \mathcal{N}(\boldsymbol{x}_j\|\boldsymbol{U}\boldsymbol{v}_j,\boldsymbol{\Sigma})$	$p(\boldsymbol{Y}\|\boldsymbol{U},\boldsymbol{V})$ $= \prod_j \mathcal{N}(\boldsymbol{y}_j\|\boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{v}_j,\boldsymbol{\Sigma})$	$p(\boldsymbol{Y},\boldsymbol{X}^u\|\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})$ $= \prod_j \mathcal{N}(\boldsymbol{y}_j\|\boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{v}_j,\boldsymbol{\Sigma}_b)$ $\prod_k \mathcal{N}(\boldsymbol{x}^u_k\|\boldsymbol{U}\boldsymbol{w}_k,\boldsymbol{\Sigma}_a)$
FIM $\mathcal{I}_U$ $\mathcal{I}_V$ $\mathcal{I}_{UV}$ $\mathcal{I}_W$ $\mathcal{I}_{UW}$ $\mathcal{I}_{VW}$	$(\boldsymbol{V}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{V})\otimes\boldsymbol{I}_M$ $(\boldsymbol{U}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{U})\otimes\boldsymbol{I}_N$ $\mathrm{vec}(\boldsymbol{U}\boldsymbol{\Sigma}^{-1})\mathrm{vec}(\boldsymbol{V})^\top$	$(\boldsymbol{V}^\top\boldsymbol{V})\otimes(\boldsymbol{\Phi}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi})$ $(\boldsymbol{U}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{U})\otimes\boldsymbol{I}_N$ $\mathrm{vec}(\boldsymbol{U}\boldsymbol{\Phi}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi})\mathrm{vec}(\boldsymbol{V})^\top$	$(\boldsymbol{V}^\top\boldsymbol{V})\otimes(\boldsymbol{\Phi}^\top\boldsymbol{\Sigma}_a^{-1}\boldsymbol{\Phi})+(\boldsymbol{W}^\top\boldsymbol{\Sigma}_b^{-1}\boldsymbol{W})\otimes\boldsymbol{I}_M$ $(\boldsymbol{U}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Sigma}_a^{-1}\boldsymbol{\Phi}\boldsymbol{U})\otimes\boldsymbol{I}_C$ $\mathrm{vec}(\boldsymbol{U}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Sigma}_a^{-1}\boldsymbol{\Phi})\mathrm{vec}(\boldsymbol{V})^\top$ $(\boldsymbol{U}^\top\boldsymbol{\Sigma}_b^{-1}\boldsymbol{U})\otimes\boldsymbol{I}_{N-C}$ $\mathrm{vec}(\boldsymbol{U}\boldsymbol{\Sigma}_b^{-1})\mathrm{vec}(\boldsymbol{W})^\top$ $\mathbf{0}$
CRB ($\boldsymbol{U}$ known)	$[\mathcal{I}_V]^{-1}$	$[\mathcal{I}_V]^{-1}$	$\begin{bmatrix}\mathcal{I}_V & \mathcal{I}_{VW}\\ \mathcal{I}_{VW}^\top & \mathcal{I}_W\end{bmatrix}^{-1}$
CRB ($\boldsymbol{U}$ unknown)	$\begin{bmatrix}\mathcal{I}_U & \mathcal{I}_{UV}\\ \mathcal{I}_{UV}^\top & \mathcal{I}_V\end{bmatrix}^{-1}$	$\begin{bmatrix}\mathcal{I}_U & \mathcal{I}_{UV}\\ \mathcal{I}_{UV}^\top & \mathcal{I}_V\end{bmatrix}^{-1}$	$\begin{bmatrix}\mathcal{I}_U & \mathcal{I}_{UV} & \mathcal{I}_{UW}\\ \mathcal{I}_{UV}^\top & \mathcal{I}_V & \mathcal{I}_{VW}\\ \mathcal{I}_{UW}^\top & \mathcal{I}_{VW}^\top & \mathcal{I}_W\end{bmatrix}^{-1}$

and each element of the parts is computed in the similar way to (6) and (7).

The CRB for matrix factorization, CS-MF and CS-MCF are computed in the following subsections. We use the maximum likelihood estimator (MLE), which is known to be asymptotically unbiased and efficient under mild regularity conditions [Kay, 1993]. The analysis is based on these asymptotic properties, assuming that there are sufficient number of samples. The models, computed Fisher information matrices and the form of CRB are summarized in Table 2.

Since the solution of the matrix factorization is not uniquely determined, the direct comparison between the estimated parameters and the ground-truth values is meaningless. Instead of directly comparing the parameters themselves, we use the expected reconstruction error

$$\begin{aligned}\mathcal{E}_{ij} &= \mathbb{E}\left\{(\boldsymbol{X}_{ij}-\widetilde{\boldsymbol{X}}_{ij})^2\right\}\\ &= \mathbb{E}\left\{(\boldsymbol{u}_i^\top\boldsymbol{v}_j-\widetilde{\boldsymbol{u}}_i^\top\widetilde{\boldsymbol{v}}_j)^2\right\},\end{aligned}$$

where $\widetilde{\boldsymbol{X}}_{ij}$, $\widetilde{\boldsymbol{u}}_i$, and $\widetilde{\boldsymbol{v}}_j$ are the ground-truth values of the parameters, and $\boldsymbol{X}_{ij}$ is the element of the reconstructed input matrix from the estimated parameters $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$. Although the factor matrices is not uniquely determined, the reconstruction error $\mathcal{E}_{ij}$ is the same for all the possible decompositions. The lower bound for the reconstruction error is represented by using the variances of the estimators, such as

$$\begin{aligned}\mathcal{E}_{ij} &= \mathbb{E}\left\{(\boldsymbol{u}_i^\top\boldsymbol{v}_j-\widetilde{\boldsymbol{u}}_i^\top\boldsymbol{v}_j+\widetilde{\boldsymbol{u}}_i^\top\boldsymbol{v}_j-\widetilde{\boldsymbol{u}}_i^\top\widetilde{\boldsymbol{v}}_j)^2\right\}\\ &= \mathbb{E}\left\{\boldsymbol{v}_j^\top(\boldsymbol{u}_i-\widetilde{\boldsymbol{u}}_i)(\boldsymbol{u}_i-\widetilde{\boldsymbol{u}}_i)^\top\boldsymbol{v}_j\right\}\\ &\quad+\mathbb{E}\left\{\widetilde{\boldsymbol{u}}_i^\top(\boldsymbol{v}_j-\widetilde{\boldsymbol{v}}_j)(\boldsymbol{v}_j-\widetilde{\boldsymbol{v}}_j)^\top\widetilde{\boldsymbol{u}}_i\right\}\\ &\quad+2\mathbb{E}\left\{\boldsymbol{v}_j^\top(\boldsymbol{u}_i-\widetilde{\boldsymbol{u}}_i)(\boldsymbol{v}_j-\widetilde{\boldsymbol{v}}_j)^\top\widetilde{\boldsymbol{u}}_i\right\}.\end{aligned}$$

If we assume that the estimator is unbiased, the last term of the above equation becomes zero, and the remaining terms are lower-bounded by the computed CRB,

$$\mathcal{E}_{ij} \geq \mathbb{E}\left\{\boldsymbol{v}_j^\top\left[\mathcal{I}^{-1}\right]_{\boldsymbol{u}_i}\boldsymbol{v}_j\right\}+\widetilde{\boldsymbol{u}}_i^\top\left[\mathcal{I}^{-1}\right]_{\boldsymbol{v}_j}\widetilde{\boldsymbol{u}}_i \quad (8)$$

where $\left[\mathcal{I}^{-1}\right]_{\boldsymbol{u}_i}$ represents the part of the inverse Fisher information matrix corresponding to the position of $\mathcal{I}_{\boldsymbol{u}_i}$. Then,

the error bound becomes

$$\mathcal{E}_{ij} \geq \widetilde{\boldsymbol{v}}_j^\top \left[\mathcal{I}^{-1}\right]_{\boldsymbol{u}_i} \widetilde{\boldsymbol{v}}_j + \mathrm{Tr}\left\{\left[\mathcal{I}^{-1}\right]_{\boldsymbol{u}_i}\left[\mathcal{I}^{-1}\right]_{\boldsymbol{v}_j}\right\} + \widetilde{\boldsymbol{u}}_i^\top \left[\mathcal{I}^{-1}\right]_{\boldsymbol{v}_j} \widetilde{\boldsymbol{u}}_i. \quad (9)$$

Note that if we assume that the basis matrix $\boldsymbol{U}$ is known in advance, we only have to deal with the parameter $\boldsymbol{V}$. With the computation similar to the above, the lower bound of the reconstruction error is computed in the following form,

$$\mathcal{E}_{ij} \geq \widetilde{\boldsymbol{u}}_i^\top \left[\mathcal{I}^{-1}\right]_{\boldsymbol{v}_j} \widetilde{\boldsymbol{u}}_i^\top, \quad (10)$$

which is smaller than the case when the basis is unknown in the amount of the terms corresponding to $\boldsymbol{U}$, which is $\widetilde{\boldsymbol{v}}_j^\top \left[\mathcal{I}^{-1}\right]_{\boldsymbol{u}_i} \widetilde{\boldsymbol{v}}_j + \mathrm{Tr}\left\{\left[\mathcal{I}^{-1}\right]_{\boldsymbol{u}_i}\left[\mathcal{I}^{-1}\right]_{\boldsymbol{v}_j}\right\}$.

3.1 Cramér-Rao Bounds for Matrix Factorization

The Fisher information matrix is computed from the likelihood of the model. If we assume that the noise is additive and distributed as the Gaussian distribution, the model is written by

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top + \boldsymbol{N},$$

where $\boldsymbol{N}_{ij} \sim \mathcal{N}(0, \sigma^2)$, and $\mathcal{N}(\mu, \sigma^2)$ is the Gaussian distribution with mean μ and variance σ^2. Then, the likelihood of the model becomes

$$p(\boldsymbol{X}|\boldsymbol{U}, \boldsymbol{V}) = \prod_j \mathcal{N}(\boldsymbol{x}_j|\boldsymbol{U}\boldsymbol{v}_j, \sigma^2\boldsymbol{I}_K).$$

Calculation of $\mathcal{I}_V$
The derivative of the log-likelihood with respect to $\boldsymbol{v}_j$ is computed by

$$\frac{\partial \log p(\boldsymbol{X}|\boldsymbol{U}, \boldsymbol{V})}{\partial \boldsymbol{v}_j} = \boldsymbol{U}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x}_j - \boldsymbol{U}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{U}\boldsymbol{v}_j,$$

where $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}_M$. If we differentiate above again with respect to $\boldsymbol{v}_{j'}$, where $j' \neq j$, the result becomes zero. If $j' = j$, then,

$$\frac{\partial^2 \log p(\boldsymbol{X}|\boldsymbol{U}, \boldsymbol{V})}{\partial \boldsymbol{v}_j \partial \boldsymbol{v}_j} = -\boldsymbol{U}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{U}.$$

Therefore, the part of Fisher information matrix for $\boldsymbol{V}$ is computed by

$$\mathcal{I}_V = \mathbb{E}\left\{(\boldsymbol{U}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{U}) \otimes \boldsymbol{I}_N\right\} = (\boldsymbol{U}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{U}) \otimes \boldsymbol{I}_N.$$

Calculation of $\mathcal{I}_U$
In this case, we use the likelihood of the model having the following form,

$$p(\boldsymbol{X}|\boldsymbol{U}, \boldsymbol{V}) = \prod_i \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{V}\boldsymbol{u}_i, \sigma^2\boldsymbol{I}_K),$$

where $\boldsymbol{x}_i$ represents the i-th row of the input matrix $\boldsymbol{X}$. The first derivative with respect to $\boldsymbol{u}_i$ is computed as

$$\frac{\partial \log p(\boldsymbol{X}|\boldsymbol{U}, \boldsymbol{V})}{\partial \boldsymbol{u}_i} = \boldsymbol{V}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x}_i - \boldsymbol{V}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{V}\boldsymbol{u}_i. \quad (11)$$

Again, the second derivative becomes zero if $i' \neq i$, and if $i' = i$,

$$\frac{\partial^2 \log p(\boldsymbol{X}|\boldsymbol{U}, \boldsymbol{V})}{\partial \boldsymbol{u}_i \partial \boldsymbol{u}_{i'}} = -\boldsymbol{V}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{V}.$$

Therefore, Fisher information matrix for $\boldsymbol{U}$ is written by

$$\mathcal{I}_U = \mathbb{E}\left\{(\boldsymbol{V}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{V}) \otimes \boldsymbol{I}_M\right\} = (\boldsymbol{V}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{V}) \otimes \boldsymbol{I}_M.$$

Calculation of $\mathcal{I}_{UV}$
To compute $\mathcal{I}_{UV}$, we differentiate (11) with respect to $\boldsymbol{v}_j$, which becomes

$$\frac{\partial^2 \log p(\boldsymbol{X}|\boldsymbol{U}, \boldsymbol{V})}{\partial \boldsymbol{u}_i \partial \boldsymbol{v}_j} = \frac{1}{\sigma^2}\left(\boldsymbol{x}_{ij}\boldsymbol{I}_K - \boldsymbol{u}_i^\top \boldsymbol{v}_j \boldsymbol{I}_K - \boldsymbol{v}_j \boldsymbol{u}_i^\top\right).$$

Since $\mathbb{E}\{\boldsymbol{x}_{ij}\boldsymbol{I}_K\} = \boldsymbol{u}_i^\top \boldsymbol{v}_j \boldsymbol{I}_K$, the first two terms of the above equation vanishes when we take the expectation. Therefore, the Fisher information matrix for $\boldsymbol{U}$ and $\boldsymbol{V}$ is written by

$$\mathcal{I}_{UV} = \mathrm{vec}(\boldsymbol{U}\boldsymbol{\Sigma}^{-1})\mathrm{vec}(\boldsymbol{V})^\top.$$

3.2 Cramér-Rao Bound for CS-MF

We model the CS-MF with the following two-step procedure,

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{U}\boldsymbol{V}^\top + \boldsymbol{N}_a, \\ \boldsymbol{Y} &= \boldsymbol{\Phi}\boldsymbol{X} + \boldsymbol{N}_b, \end{aligned}$$

where $[\boldsymbol{N}_a]_{i,j} \sim \mathcal{N}(0, \sigma_a^2)$ is the representation error and $[\boldsymbol{N}_b]_{i,j} \sim \mathcal{N}(0, \sigma_b^2)$ is the measurement error. The likelihood is written by

$$\begin{aligned} p(\boldsymbol{Y}|\boldsymbol{U}, \boldsymbol{V}) &= \int p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X}|\boldsymbol{U}, \boldsymbol{V})d\boldsymbol{X} \\ &= \prod_j \mathcal{N}(\boldsymbol{y}_j|\boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{v}_j, \boldsymbol{\Sigma}), \end{aligned}$$

where $\boldsymbol{\Sigma} = \sigma_a^2\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \sigma_b^2\boldsymbol{I}_P$, where $\boldsymbol{\Phi} \in \mathbb{R}^{P \times M}$.

Calculation of $\mathcal{I}_V$
The derivative of the log-likelihood with respect to $\boldsymbol{v}_j$ is computed by

$$\frac{\partial \log p(\boldsymbol{Y}|\boldsymbol{U}, \boldsymbol{V})}{\partial \boldsymbol{v}_j} = \boldsymbol{U}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{y}_j - \boldsymbol{U}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{U}\boldsymbol{v}_j.$$

Differentiate again with $\boldsymbol{v}_{j'}$ becomes zero when $j' \neq j$, and if $j' = j$, then

$$\frac{\partial^2 \log p(\boldsymbol{Y}|\boldsymbol{U}, \boldsymbol{V})}{\partial \boldsymbol{v}_j \partial \boldsymbol{v}_j} = -\boldsymbol{U}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{U}.$$

Therefore, the Fisher information matrix for $\boldsymbol{V}$ is computed by

$$\begin{aligned} \mathcal{I}_V &= \mathbb{E}\left\{\boldsymbol{U}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{U} \otimes \boldsymbol{I}_N\right\} \\ &= \boldsymbol{U}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{U} \otimes \boldsymbol{I}_N. \end{aligned}$$

Calculation of $\mathcal{I}_U$

The derivative with respect to $\boldsymbol{u}_i$ is written by

$$\frac{\partial \log p(\boldsymbol{Y}|\boldsymbol{U},\boldsymbol{V})}{\partial \boldsymbol{u}_i} = \sum_j \left(\boldsymbol{v}_j \boldsymbol{y}_j^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\phi}_i - \boldsymbol{v}_j \boldsymbol{\phi}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi} \boldsymbol{U} \boldsymbol{v}_j \right).$$

The derivative of above equation with respect to $\boldsymbol{u}_{i'}$ is simplified by

$$\frac{\partial^2 \log p(\boldsymbol{Y}|\boldsymbol{U},\boldsymbol{V})}{\partial \boldsymbol{u}_i \partial \boldsymbol{u}_{i'}} = -(\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi})_{ii'} \boldsymbol{V}^\top \boldsymbol{V},$$

which leads the Fisher information matrix for $\boldsymbol{U}$ to be

$$\mathcal{I}_U = (\boldsymbol{V}^\top \boldsymbol{V}) \otimes (\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}).$$

Calculation of $\mathcal{I}_{UV}$

If we differentiate (12) with respect to $\boldsymbol{v}_j$, we obtain

$$\frac{\partial^2 \log p(\boldsymbol{Y}|\boldsymbol{U},\boldsymbol{V})}{\partial \boldsymbol{u}_{i^*} \partial \boldsymbol{v}_{j^+}} = \boldsymbol{\phi}_{i^*}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{y}_{j^+} \boldsymbol{I}_K - \boldsymbol{\phi}_{i^*}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi} \boldsymbol{U} \boldsymbol{v}_{j^+} \boldsymbol{I}_K - \boldsymbol{U}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\phi}_{i^*} \boldsymbol{v}_{j^+}^\top.$$

The first two terms disappear when we take the expectation, so the Fisher information matrix for $\boldsymbol{U}$ and $\boldsymbol{V}$ becomes

$$\mathcal{I}_{UV} = \text{vec}(\boldsymbol{U}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}) \text{vec}(\boldsymbol{V})^\top.$$

The theoretical bound of CS-MF is larger than that of matrix factorization. The main difference of the computed Fisher information matrix is from the change of the term $\boldsymbol{\Sigma}^{-1}$ of matrix factorization to the term $\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}$. Note that the $\boldsymbol{\Sigma}^{-1}$ in the matrix factorization is

$$\boldsymbol{\Sigma}^{-1} = \sigma_a^{-2} \boldsymbol{I}_M. \tag{12}$$

However, $\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}$ in the CS-MF is written by

$$\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi} = \boldsymbol{\Phi}^\top (\sigma_a^2 \boldsymbol{\Phi} \boldsymbol{\Phi}^\top + \sigma_b^2 \boldsymbol{I}_P)^{-1} \boldsymbol{\Phi}.$$

This is usually smaller than the corresponding term in the matrix factorization. For example, if we assume that $\sigma_b^2 = 0$ and each element of sensing matrix is drawn independently from $\mathcal{N}(0,1)$, we approximate the term using $\boldsymbol{\Phi}^\top \boldsymbol{\Phi} \approx P \boldsymbol{I}_M$ and $\boldsymbol{\Phi} \boldsymbol{\Phi}^\top \approx M \boldsymbol{I}_P$, such that

$$\begin{aligned} \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi} &\approx \boldsymbol{\Phi}^\top (\sigma_a^2 M \boldsymbol{I}_P)^{-1} \boldsymbol{\Phi} \\ &\approx \frac{1}{M} \sigma_a^{-2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \\ &\approx \frac{P}{M} \sigma_a^{-2} \boldsymbol{I}_M, \end{aligned}$$

which is P/M times smaller than that part of matrix factorization (12). Since the CRB is calculated as the inverse of the Fisher information matrix, CS-MF has larger CRB than that of the matrix factorization.

3.3 Cramér-Rao Bound for CS-MCF

The relations in CS-MCF are modeled as in the previous section, such that

$$\begin{aligned} \boldsymbol{X}^c &= \boldsymbol{U} \boldsymbol{V}^\top + \boldsymbol{N}_a, \\ \boldsymbol{Y} &= \boldsymbol{\Phi} \boldsymbol{X}^c + \boldsymbol{N}_b, \\ \boldsymbol{X}^u &= \boldsymbol{U} \boldsymbol{W}^\top + \boldsymbol{N}_c, \end{aligned}$$

where $[\boldsymbol{N}_a]_{i,j} \sim \mathcal{N}(0, \sigma_a^2)$, $[\boldsymbol{N}_c]_{i,j} \sim \mathcal{N}(0, \sigma_a^2)$ and $[\boldsymbol{N}_b]_{i,j} \sim \mathcal{N}(0, \sigma_b^2)$. In this case we have three parameters $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$, so the Fisher information matrix is computed as

$$\mathcal{I} = \begin{bmatrix} \mathcal{I}_U & \mathcal{J}_{UV} & \mathcal{I}_{UW} \\ \mathcal{I}_{UV}^\top & \mathcal{I}_V & \mathcal{I}_{VW} \\ \mathcal{I}_{UW}^\top & \mathcal{I}_{VW}^\top & \mathcal{I}_W \end{bmatrix}.$$

The likelihood is

$$\begin{aligned} & p(\boldsymbol{Y}, \boldsymbol{X}^u | \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \\ &= p(\boldsymbol{Y}|\boldsymbol{U},\boldsymbol{V}) p(\boldsymbol{X}^u|\boldsymbol{U},\boldsymbol{W}) \\ &= \prod_j \mathcal{N}(\boldsymbol{y}_j | \boldsymbol{\Phi} \boldsymbol{U} \boldsymbol{v}_j, \boldsymbol{\Sigma}_b) \prod_l \mathcal{N}(\boldsymbol{x}_l^u | \boldsymbol{U} \boldsymbol{w}_l, \boldsymbol{\Sigma}_a), \end{aligned}$$

where $\boldsymbol{\Sigma}_b = \sigma_a^2 \boldsymbol{\Phi} \boldsymbol{\Phi}^\top + \sigma_b^2 \boldsymbol{I}_P$ and $\boldsymbol{\Sigma}_a = \sigma_a^2 \boldsymbol{I}_M$.

Calculation of $\mathcal{I}_U$

Differentiating the log-likelihood with respect to $\boldsymbol{u}_i$ leads

$$\begin{aligned} & \frac{\partial \log p(\boldsymbol{Y}, \boldsymbol{X}^u | \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})}{\partial \boldsymbol{u}_i} \\ = & \sum_j \left(\boldsymbol{v}_j \boldsymbol{y}_j^\top \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\phi}_i - \boldsymbol{v}_j \boldsymbol{\phi}_i^\top \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\Phi} \boldsymbol{U} \boldsymbol{v}_j \right) \\ & + \boldsymbol{W}^\top \boldsymbol{\Sigma}_a^{-1} \boldsymbol{x}_i - \boldsymbol{W}^\top \boldsymbol{\Sigma}_a^{-1} \boldsymbol{W} \boldsymbol{u}_i. \end{aligned} \tag{13}$$

Differentiating again with respect to $\boldsymbol{u}_{i'}$ yields

$$\begin{aligned} & \frac{\partial^2 \log p(\boldsymbol{Y}, \boldsymbol{X}^u | \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})}{\partial \boldsymbol{u}_i \partial \boldsymbol{u}_{i'}} \\ &= -(\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\Phi})_{ii'} \boldsymbol{V}^\top \boldsymbol{V} - \boldsymbol{W}^\top \boldsymbol{\Sigma}_a^{-1} \boldsymbol{W}, \end{aligned}$$

where $i' = i$, and yields

$$\frac{\partial^2 \log p(\boldsymbol{Y}, \boldsymbol{X}^u | \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})}{\partial \boldsymbol{u}_i \partial \boldsymbol{u}_{i'}} = -(\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\Phi})_{ii'} \boldsymbol{V}^\top \boldsymbol{V},$$

where $i' \neq i$. Therefore, the Fisher information matrix for $\boldsymbol{U}$ becomes

$$\mathcal{I}_U = (\boldsymbol{V}^\top \boldsymbol{V}) \otimes (\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\Phi}) + (\boldsymbol{W}^\top \boldsymbol{\Sigma}_a^{-1} \boldsymbol{W}) \otimes \boldsymbol{I}_M.$$

Calculation of $\mathcal{I}_V$, $\mathcal{I}_W$, $\mathcal{I}_{UV}$, and $\mathcal{I}_{VW}$

$\mathcal{I}_V$, $\mathcal{I}_W$, $\mathcal{I}_{UV}$, and $\mathcal{I}_{VW}$ are computed in the similar way as in matrix factorization and CS-MF, in the following forms,

$$\begin{aligned} \mathcal{I}_V &= (\boldsymbol{U}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\Phi} \boldsymbol{U}) \otimes \boldsymbol{I}_C, \\ \mathcal{I}_W &= (\boldsymbol{U}^\top \boldsymbol{\Sigma}_a^{-1} \boldsymbol{U}) \otimes \boldsymbol{I}_{N-C}, \\ \mathcal{I}_{UV} &= \text{vec}(\boldsymbol{U}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\Phi}) \text{vec}(\boldsymbol{V})^\top, \\ \mathcal{I}_{UW} &= \text{vec}(\boldsymbol{U} \boldsymbol{\Sigma}_a^{-1}) \text{vec}(\boldsymbol{W})^\top. \end{aligned}$$

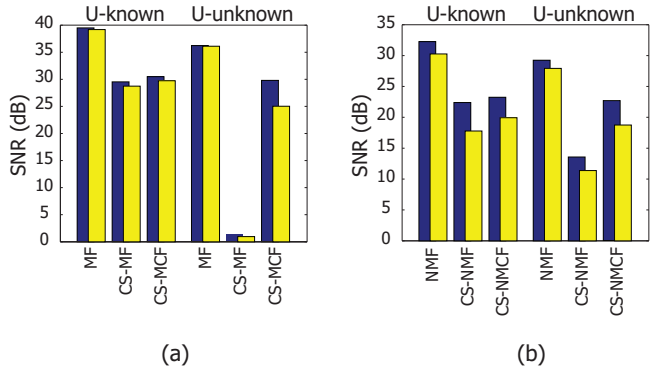


Figure 2: SNR computed from the CRB (dark) and SNR achieved from the algorithms (light), for the (a) ALS algorithms with Gaussian data (b) Multiplicative algorithms with nonnegative data. In each graph, left three are for the case of known representation $\boldsymbol{U}$, and right three are for the case of unknown $\boldsymbol{U}$.

Calculation of $\mathcal{I}_{VW}$

Fisher information matrix for $\boldsymbol{V}$ and $\boldsymbol{W}$, $\mathcal{I}_{VW}$, becomes zero matrix of dimension $CK \times (N-C)K$ because the term $\boldsymbol{W}$ banishes in the derivative with respect to $\boldsymbol{v}_j$.

The Fisher information matrix of CS-MCF is the combination of the Fisher information matrix of matrix factorization and Fisher information matrix of CS-MF. The parameter $\boldsymbol{V}$ in matrix factorization and CS-MF is divided into $\boldsymbol{V}$ and $\boldsymbol{W}$ in CS-MCF. The $\mathcal{I}_V$ and $\mathcal{I}_{UV}$ in CS-MCF follows the form of the corresponding parts of CS-MF, but the $\mathcal{I}_W$ and $\mathcal{I}_{UW}$ follows the form of the corresponding parts of matrix factorization. Moreover, $\mathcal{I}_U$ is the direct combination of the $\mathcal{I}_U$ of matrix factorization and $\mathcal{I}_U$ of CS-MF. In the previous section, we showed that in general the CRB of matrix factorization is smaller than the CRB of CS-MF, so this kind of combination leads smaller CRB than that of CS-MF.

4 Numerical Experiments

For the numerical experiments, we computed the bounds of the reconstruction performances using the equation (9), and compared them with the actual reconstruction performances of the algorithms listed in the Table 1. We also computed the bounds for the cases where we assume that the parameter $\boldsymbol{U}$ is known in advance (10). In that cases, the actual reconstruction performances were measured based on the modified algorithms, where the parameter $\boldsymbol{U}$ is fixed with the ground-truth values.

To compute the CRB, we have to know the ground-truth value of the parameters $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$, which makes it difficult to use the real-world data where the ground-truth factor matrices are unknown. Therefore, we used synthetic data for the experiments. The factor matrices $\boldsymbol{U} \in \Re^{100\times 10}$, $\boldsymbol{V} \in \Re^{100\times 10}$ and the sensing matrix $\boldsymbol{\Phi} \in \Re^{20\times 100}$ are randomly generated, and the input matrices were drawn from them with additive Gaussian noise. For the ALS-based algorithms, each element of the factor matrices and the sensing matrix is sampled from the Gaussian distribution with mean 0 and variance 1. For the multiplicative update algorithms which have the nonnegativity constraints, each element is sampled from a uniform distribution between 0 and 1. For the co-factorizations, 20 out of 100 columns were chosen for the uncompressed prior.

As a measure of the reconstruction performances, we used signal-to-noise ratio (SNR)

$$\text{SNR} = 10\log_{10}\frac{\|\widetilde{\boldsymbol{X}}\|_F^2}{\|\widetilde{\boldsymbol{X}} - \boldsymbol{U}\boldsymbol{V}^\top\|_F^2},$$

where $\widetilde{\boldsymbol{X}}$ is the true value of the input matrix, $\boldsymbol{U}$ and $\boldsymbol{V}$ are the factor matrices obtained by running the algorithms. The SNRs were averaged over the 100 different trials (Figure 2). The bounds were better for the cases where the matrix $\boldsymbol{U}$ is known in advance, than the cases where the matrix $\boldsymbol{U}$ is unknown. This indicates that the estimation of both $\boldsymbol{U}$ and $\boldsymbol{V}$ is more difficult than the estimation of $\boldsymbol{V}$ only. The bounds of CS-MF and CS-NMF were seriously degraded compared to the bounds of MF and NMF, as well as the actual performances, especially for the cases that the matrix $\boldsymbol{U}$ is unknown. However, the co-factorization based methods showed much improved bounds, as well as the performances, compared to CS-MF and CS-NMF. The use of uncompressed prior as the side information actually helps to improve the reconstruction performance in the learning from the compressed data.

As an illustrative example, we ran the algorithms for the compressively sampled data on the cameraman image (Figure 3). We divided the image using 8-by-8 patches to build an input matrix, and compress each column into the size of 49. 25 % of the data is randomly selected to be used for the uncompressed prior. CS-MCF brought much better reconstruction result than the CS-MF.

5 Conclusions

We have presented a method for matrix co-factorization in the compressed domain where compressively-sampled data and partial uncompressed data were jointly decomposed, sharing a factor matrix. We calculated CRBs for three matrix factorization methods (MF, CS-MF, CS-MCF) in the case of Gaussian likelihood, showing that CS-MCF improves the reconstruction performance over CS-MF. Numerical experiments on image confirmed the better performance of CS-MCF over CS-MF.

Acknowledgments: This work was supported by NIPA-MSRA Creative IT/SW Research Project, the Converging Research Center Program funded by Korea MEST (No. 2010K001171), NIPA ITRC Support Program (NIPA-2011-C1090-1131-0009), and NRF WCU Program (R31-2010-000-10100-0).

References

[Candes and Tao, 2006] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

Figure 3: Reconstruction example of an image: (a) original image (b) the reconstruction result of CS-MF (c) the reconstruction result of CS-MCF (d) the reconstruction result of CS-NMF (e) the reconstruction result of CS-NMCF

[Donoho, 2006] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[Jolliffe, 2002] I. T. Jolliffe. *Principal Component Analysis*. Spinger-Verlag, 2 edition, 2002.

[Kay, 1993] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.

[Koren *et al.*, 2009] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[Lee and Choi, 2009] H. Lee and S. Choi. Group nonnegative matrix factorization for EEG classification. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Clearwater Beach, Florida, 2009.

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

[Lee *et al.*, 2010] H. Lee, J. Yoo, and S. Choi. Semi-supervised nonnegative matrix factorization. *IEEE Signal Processing Letters*, 17(1):4–7, 2010.

[O'Grady and Rickard, 2008] P. D. O'Grady and S. T. Rickard. Non-negative matrix factorisation of compressibely sampled non-negative signals. In *Proceedings of the 8th IMA International Conference on Mathematics in Signal Processing*, Cirencester, UK, 2008.

[Singh and Gordon, 2008] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Las Vegas, Nevada, 2008.

[Smaragdis and Brown, 2003] P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, New Paltz, NY, 2003.

[Srebro *et al.*, 2005] N. Srebro, J. D. M. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 17. MIT Press, 2005.

[Xu *et al.*, 2003] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, Toronto, Canada, 2003.

[Yoo and Choi, 2008] J. Yoo and S. Choi. Orthogonal nonnegative matrix factorization: Multiplicative updates on Stiefel manifolds. In *Proceedings of the 9th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, Daejeon, Korea, 2008.

[Yoo and Choi, 2009] J. Yoo and S. Choi. Weighted nonnegative matrix co-tri-factorization for collaborative prediction. In *Proceedings of the 1st Asian Conference on Machine Learning (ACML)*, Nanjing, China, 2009.

[Yoo and Choi, 2010] J. Yoo and S. Choi. Orthogonal nonnegative matrix-tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds. *Information Processing and Management*, 46(5):559–570, 2010.

[Yoo *et al.*, 2010] J. Yoo, M. Kim, K. Kang, and S. Choi. Nonnegative matrix partial co-factorization for drum source separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas, 2010.

[Yu *et al.*, 2005] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, Salvador, Brazil, 2005.

[Zhu *et al.*, 2007] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, Amsterdam, The Netherlands, 2007.