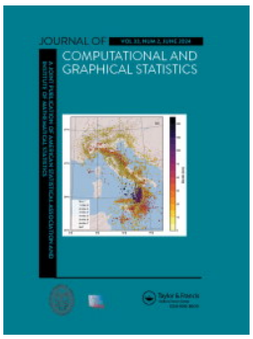

Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ucgs20

MFAI: A Scalable Bayesian Matrix Factorization Approach to Leveraging Auxiliary Information

Zhiwei Wang, Fa Zhang, Cong Zheng, Xianghong Hu, Mingxuan Cai & Can Yang

To cite this article: Zhiwei Wang, Fa Zhang, Cong Zheng, Xianghong Hu, Mingxuan Cai & Can Yang (25 Mar 2024): MFAI: A Scalable Bayesian Matrix Factorization Approach to Leveraging Auxiliary Information, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2024.2319160

To link to this article: https://doi.org/10.1080/10618600.2024.2319160

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2024, VOL. 00, NO. 0, 1–11
https://doi.org/10.1080/10618600.2024.2319160

MFAI: A Scalable Bayesian Matrix Factorization Approach to Leveraging Auxiliary Information

Zhiwei Wang[a], Fa Zhang[a], Cong Zheng[a], Xianghong Hu[a], Mingxuan Cai[b], and Can Yang[a]

[a]Department of Mathematics, The Hong Kong University of Science and Technology, Kowloon, Hong Kong; [b]Department of Biostatistics, City University of Hong Kong, Kowloon, Hong Kong

ABSTRACT

In various practical situations, matrix factorization methods suffer from poor data quality, such as high data sparsity and low signal-to-noise ratio (SNR). Here, we consider a matrix factorization problem by using auxiliary information, which is massively available in real-world applications, to overcome the challenges caused by poor data quality. Unlike existing methods that mainly rely on simple linear models to combine auxiliary information with the main data matrix, we propose to integrate gradient boosted trees in the probabilistic matrix factorization framework to effectively leverage auxiliary information (MFAI). Thus, MFAI naturally inherits several salient features of gradient boosted trees, such as the capability of flexibly modeling nonlinear relationships and robustness to irrelevant features and missing values in auxiliary information. The parameters in MFAI can be automatically determined under the empirical Bayes framework, making it adaptive to the utilization of auxiliary information and immune to overfitting. Moreover, MFAI is computationally efficient and scalable to large datasets by exploiting variational inference. We demonstrate the advantages of MFAI through comprehensive numerical results from simulation studies and real data analyses. Our approach is implemented in the R package *mfair* available at *https://github.com/YangLabHKUST/mfair*. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received March 2023
Accepted February 2024

KEYWORDS
Empirical Bayes; Gradient boosting; Human brain gene expression; Low-rank; Matrix completion; Movie rating

1. Introduction

Matrix factorization (Srebro, Rennie, and Jaakkola 2004; Salakhutdinov and Mnih 2007) is widely used when handling large-scale data. It has become an important topic in the fields of applied mathematics, statistics, and machine learning because of its broad applications. For example, as motivated by the Netflix Prize, matrix factorization has emerged as an effective method to infer the unobserved entries, commonly referred to as the matrix completion problem (Candès and Recht 2009; Mazumder, Hastie, and Tibshirani 2010; Ilin and Raiko 2010). Matrix factorization can also help uncover the underlying structures of datasets from diverse research topics, such as background modeling in moving object detection (Zhou, Yang, and Yu 2012; Zhou et al. 2014), dimension reduction and adjustment for confounding variations (Yang et al. 2013; Lin et al. 2016).

Although existing matrix factorization methods have been used in various applications, major challenges remain due to low-quality data in practice. First, the observed matrix can be very sparse for the matrix completion problem. Second, the observed matrix can be quite noisy, and matrix factorization in low signal-to-noise ratio (SNR) settings tends to overfit easily. Effective extraction of signals in the low SNR setting becomes critical for the success of matrix factorization. A promising way to overcome the above challenges is to leverage auxiliary information (Singh and Gordon 2008; Kula 2015; Aktukmak, Yilmaz, and Uysal 2019; Yilmaz, Aktukmak, and Hero 2021), which is massively available in real-world applications (Velten et al. 2022; Shang and Zhou 2022). To date, there have been a number of studies on matrix factorization with auxiliary information. These methods can be roughly grouped into two categories: regularized methods and Bayesian methods. For regularized methods, they often assume some shared structures between auxiliary information and the main matrix such that auxiliary information can be incorporated to regularize the factorization of the main matrix. For Bayesian methods, they often build a probabilistic model where auxiliary information is incorporated through a linear model.

Despite many efforts in the incorporation of auxiliary information, several main issues remain. First, existing methods rely on linear models to combine auxiliary information with the main matrix, which may limit its role because linear models are not flexible enough. A more flexible framework is highly desired to take full advantage of auxiliary information. Second, the computational costs of existing methods are often quite expensive, even though only linear models are used. For example, Bayesian methods often use sampling methods to approximate posterior distributions, which are too computationally expensive to scale up for large datasets. For some regularized methods, efficient implementation is also lacking due to the challenge of parallelization (Hubbard and Hegde 2017). Third, incorporating irrelevant information will

CONTACT Can Yang *macyang@ust.hk* Department of Mathematics, The Hong Kong University of Science and Technology, Kowloon, Hong Kong; Mingxuan Cai. *mingxcai@cityu.edu.hk* Department of Biostatistics, City University of Hong Kong, Kowloon, Hong Kong.
Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/JCGS*.

not improve but degrade the performance. Existing methods largely rely on parameter tuning to control the amount of auxiliary information incorporated. Although cross-validation can help with this, it will become very time-consuming when there are many tuning parameters. Statistical methods that can adaptively leverage auxiliary information are highly demanding.

In this article, we develop a scalable Bayesian **M**atrix **F**actorization approach to adaptively leveraging **A**uxiliary **I**nformation (MFAI). Specifically, MFAI is a unified probabilistic approach to integrating gradient boosted trees (Friedman 2001) with matrix factorization. Through innovations in the model and algorithm designs, MFAI has several unique advantages over existing matrix factorization methods. First, MFAI naturally inherits several salient features of gradient boosted trees, such as the capability of flexibly modeling nonlinear relationships, robustness to irrelevant features and missing values in predictors, and ranking the relative importance of auxiliary information, which offers more interpretable insights (Elith, Leathwick, and Hastie 2008; Sigrist 2022). Second, the parameters in MFAI can be automatically determined under the empirical Bayes framework, making it adaptive to the utilization of auxiliary information. Third, MFAI is computationally efficient and scalable to large datasets by exploiting variational inference (VI) (Bishop 2006; Blei, Kucukelbir, and McAuliffe 2017). Through comprehensive simulation experiments and real data studies, we demonstrate that MFAI can perform better in matrix factorization and completion tasks than the existing methods.

2. Methods

2.1. The MFAI Model

Given the main data matrix $\mathbf{Y} \in \mathbb{R}^{N\times M}$ of N samples and M features, we consider the following matrix factorization problem:

$$\mathbf{Y} = \mathbf{Z}\mathbf{W}^{\mathrm{T}} + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathbf{Z} \in \mathbb{R}^{N\times K}$ and $\mathbf{W} \in \mathbb{R}^{M\times K}$ are two matrices with $K \leq \min\{N, M\}$, and $\boldsymbol{\epsilon} \in \mathbb{R}^{N\times M}$ is a matrix of residual error terms. Here, we adopt the terminology of factor analysis and refer to $\mathbf{Z}$ as the "factors", $\mathbf{W}$ as the "loadings", and K as the number of factors. We can further expand the above formulation as the sum of the K factors

$$\mathbf{Y} = \sum_{k=1}^{K} \mathbf{Z}_{\cdot k}\mathbf{W}_{\cdot k}^{\mathrm{T}} + \boldsymbol{\epsilon}, \tag{2}$$

where $\mathbf{Z}_{\cdot k}$ and $\mathbf{W}_{\cdot k}$ are the kth column of $\mathbf{Z}$ and $\mathbf{W}$, respectively. To perform matrix factorization of $\mathbf{Y}$, we can use not only the main matrix but also some auxiliary information that may be helpful in identifying the factors. Specifically, we relate $\mathbf{Z}_{\cdot k}$ and auxiliary covariates $\mathbf{X}$ using the following probabilistic model:

$$\mathbf{Z}_{\cdot k} \sim \mathcal{N}_N\left(F_k(\mathbf{X}), \beta_k^{-1}\mathbf{I}_N\right),\ k = 1, \ldots, K, \tag{3}$$

where $F_k(\mathbf{X}) \in \mathbb{R}^{N\times 1}$ is the mean vector of the factor $\mathbf{Z}_{\cdot k}$, β_k is the precision, $\mathbf{I}_N \in \mathbb{R}^{N\times N}$ is an identity matrix, and $\mathcal{N}_N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the N-variate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Note that $F_k(\mathbf{X})$ is the row-wise evaluation of the unknown function $F_k : \mathbb{R}^C \to \mathbb{R}$, $F_k(\mathbf{X}) = (F_k(\mathbf{X}_{1\cdot}), \ldots, F_k(\mathbf{X}_{N\cdot}))^{\mathrm{T}}$, where $\mathbf{X}_{n\cdot} = (\mathbf{X}_{n1}, \ldots, \mathbf{X}_{nC})^{\mathrm{T}} \in \mathbb{R}^{C\times 1}$ is the nth row of $\mathbf{X}$ containing auxiliary information for the nth sample. In the MFAI model, we assume that $F_k(\cdot)$ in (3) is a nonlinear function represented by a tree ensemble,

$$F_k(\cdot) = \sum_{t=1}^{T_k} f_k^t(\cdot), \tag{4}$$

where $f_k^t(\cdot)$ is a regression tree (Breiman 1984), and T_k is the total number of trees. We then assign an independent Gaussian prior for the corresponding kth loading $\mathbf{W}_{\cdot k}$

$$\mathbf{W}_{\cdot k} \sim \mathcal{N}_M(0, \mathbf{I}_M), \tag{5}$$

which can push the variability to the factor $\mathbf{Z}_{\cdot k}$ side and partially help avoid the non-identifiability issue. Matrices $\mathbf{Z}$ and $\mathbf{W}$ here are often referred to as latent variables in the statistical machine learning literature. At last, we assume independent Gaussian error terms

$$\boldsymbol{\epsilon}_{nm} \sim \mathcal{N}(0, \tau^{-1}),\ n = 1, \ldots, N \text{ and } m = 1, \ldots, M, \tag{6}$$

where τ is the precision parameter.

Let $\boldsymbol{\Theta} = \{\tau, \boldsymbol{\beta}\} = \{\tau; \beta_1, \ldots, \beta_K\}$ be the collection of model parameters and $\mathbf{F}(\cdot) = \{F_1(\cdot), \ldots, F_K(\cdot)\}$ be the collection of K unknown functions. Combining (2), (3), (5), (6), we can write down the joint probabilistic model as

$$\begin{aligned}
&\Pr(\mathbf{Y}, \mathbf{Z}, \mathbf{W} \mid \boldsymbol{\Theta}, \mathbf{F}(\cdot)) \\
&= \Pr(\mathbf{Y} \mid \mathbf{Z}, \mathbf{W}; \tau) \Pr(\mathbf{Z} \mid \boldsymbol{\beta}, \mathbf{F}(\cdot)) \Pr(\mathbf{W}) \\
&= \Pr(\mathbf{Y} \mid \mathbf{Z}, \mathbf{W}; \tau) \prod_{k=1}^{K} \Pr(\mathbf{Z}_{\cdot k} \mid \beta_k, F_k(\cdot)) \prod_{k=1}^{K} \Pr(\mathbf{W}_{\cdot k}).
\end{aligned} \tag{7}$$

As an empirical Bayes approach, we can adaptively estimate $\boldsymbol{\Theta}$ and $\mathbf{F}(\cdot)$ by optimizing the log marginal likelihood

$$\begin{aligned}
\left(\widehat{\boldsymbol{\Theta}}, \widehat{\mathbf{F}}(\cdot)\right) &= \underset{\boldsymbol{\Theta}, \mathbf{F}(\cdot)}{\arg\max}\ \log \Pr(\mathbf{Y} \mid \boldsymbol{\Theta}, \mathbf{F}(\cdot)) \\
&= \underset{\boldsymbol{\Theta}, \mathbf{F}(\cdot)}{\arg\max}\ \log \int \Pr(\mathbf{Y}, \mathbf{Z}, \mathbf{W} \mid \boldsymbol{\Theta}, \mathbf{F}(\cdot))\, d\mathbf{Z}\, d\mathbf{W}.
\end{aligned} \tag{8}$$

Then, we can infer the latent factors and loadings using the posterior probability

$$\Pr\left(\mathbf{Z}, \mathbf{W} \mid \mathbf{Y}; \widehat{\boldsymbol{\Theta}}, \widehat{\mathbf{F}}(\cdot)\right) = \frac{\Pr\left(\mathbf{Y}, \mathbf{Z}, \mathbf{W} \mid \widehat{\boldsymbol{\Theta}}, \widehat{\mathbf{F}}(\cdot)\right)}{\Pr\left(\mathbf{Y} \mid \widehat{\boldsymbol{\Theta}}, \widehat{\mathbf{F}}(\cdot)\right)}. \tag{9}$$

2.2. Fitting the MFAI Model

We begin our algorithm design with the single-factor case, that is, $K = 1$, and extend our algorithm to the multi-factor case in Section 2.2.4. To further lighten the notation, we use $\mathbf{z} \in \mathbb{R}^{N\times 1}$ and $\mathbf{w} \in \mathbb{R}^{M\times 1}$ instead of $\mathbf{Z}_{\cdot 1}$ and $\mathbf{W}_{\cdot 1}$.

2.2.1. Approximate Bayesian Inference

The Bayesian inference using (8) and (9) is intractable since the marginal likelihood $\Pr(\mathbf{Y} \mid \boldsymbol{\Theta}, F(\cdot))$ cannot be computed by marginalizing all latent variables. To tackle the Bayesian inference problem, there are two main methods: Markov Chain Monte Carlo (MCMC) (Neal 1993), which is a sampling-based approach, and variational inference (VI) (Bishop 2006; Blei, Kucukelbir, and McAuliffe 2017), which is an approximation-based approach. The advantage of the sampling-based methods is that they produce exact results asymptotically. In practice, however, they are often too computationally expensive for large-scale problems. Here, we propose a variational expectation-maximization (EM) algorithm to perform approximate Bayesian inference (see Appendix Section A.1 for details). To apply variational approximation, we first define $q(\mathbf{z}, \mathbf{w})$ as an approximated distribution of posterior $\Pr(\mathbf{z}, \mathbf{w} \mid \mathbf{Y}; \boldsymbol{\Theta}, F(\cdot))$. Then, we obtain the evidence lower bound (ELBO) of the logarithm of the marginal likelihood using Jensen's inequality

$$\begin{aligned}
&\log \Pr(\mathbf{Y} \mid \boldsymbol{\Theta}, F(\cdot)) \\
&\geq \int q(\mathbf{z}, \mathbf{w}) \log \frac{\Pr(\mathbf{Y}, \mathbf{z}, \mathbf{w} \mid \boldsymbol{\Theta}, F(\cdot))}{q(\mathbf{z}, \mathbf{w})} \, d\mathbf{z} \, d\mathbf{w} \\
&= \mathbb{E}_q \left[\log \Pr(\mathbf{Y}, \mathbf{z}, \mathbf{w} \mid \boldsymbol{\Theta}, F(\cdot))\right] - \mathbb{E}_q \left[\log q(\mathbf{z}, \mathbf{w})\right] \\
&\triangleq \text{ELBO}\left(q; \boldsymbol{\Theta}, F(\cdot)\right),
\end{aligned} \tag{10}$$

where the equality holds if and only if $q(\mathbf{z}, \mathbf{w})$ is the exact posterior $\Pr(\mathbf{z}, \mathbf{w} \mid \mathbf{Y}; \boldsymbol{\Theta}, F(\cdot))$. Instead of maximizing the logarithm of the marginal likelihood, we can iteratively maximize the ELBO with respect to the variational approximate posterior q, the model parameters $\boldsymbol{\Theta}$, and the function $F(\cdot)$

$$\left(\widehat{q}, \widehat{\boldsymbol{\Theta}}, \widehat{F}(\cdot)\right) = \underset{q; \boldsymbol{\Theta}, F(\cdot)}{\arg\max} \, \text{ELBO}\left(q; \boldsymbol{\Theta}, F(\cdot)\right). \tag{11}$$

Using the terminology in the EM algorithm, maximizing ELBO with respect to q is known as the E-step, and maximizing ELBO with respect to $\boldsymbol{\Theta}$ and $F(\cdot)$ is known as the M-step. To approximate the posterior distribution, we consider the following mean-field factorization of $q(\mathbf{z}, \mathbf{w})$:

$$q(\mathbf{z}, \mathbf{w}) = q(\mathbf{z}) \, q(\mathbf{w}). \tag{12}$$

Without further assumptions, we show that (with details in Appendix Section A.1.1) the optimal solutions of $q(\mathbf{z})$ and $q(\mathbf{w})$ in the E-step are given as two Gaussian distributions

$$q(\mathbf{z}) = \mathcal{N}_N\left(\mathbf{z} \mid \boldsymbol{\mu}, a^2 \mathbf{I}_N\right), \; q(\mathbf{w}) = \mathcal{N}_M\left(\mathbf{w} \mid \boldsymbol{\nu}, b^2 \mathbf{I}_M\right), \tag{13}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{N \times 1}$ and $\boldsymbol{\nu} \in \mathbb{R}^{M \times 1}$ are posterior mean vectors, a^2 and b^2 are posterior variances. Now suppose that we are at the tth step of the iteration, and we have obtained $\left\{\boldsymbol{\mu}^{(t-1)}, a^{2(t-1)}; \boldsymbol{\nu}^{(t-1)}, b^{2(t-1)}\right\}$, $\boldsymbol{\Theta}^{(t-1)} = \left\{\tau^{(t-1)}, \beta^{(t-1)}\right\}$, and $F^{(t-1)}(\cdot)$ at the $(t-1)$th step. To maximize ELBO in the tth E-step, we can update variational parameters as

$$\begin{aligned}
a^{2(t)} &= \frac{1}{\beta^{(t-1)} + \tau^{(t-1)}\left(\left\|\boldsymbol{\nu}^{(t-1)}\right\|_2^2 + M b^{2(t-1)}\right)}, \\
\boldsymbol{\mu}^{(t)} &= a^{2(t)}\left(\beta^{(t-1)} F^{(t-1)}(\mathbf{X}) + \tau^{(t-1)} \mathbf{Y} \boldsymbol{\nu}^{(t-1)}\right), \\
b^{2(t)} &= \frac{1}{1 + \tau^{(t-1)}\left(\left\|\boldsymbol{\mu}^{(t)}\right\|_2^2 + N a^{2(t)}\right)}, \\
\boldsymbol{\nu}^{(t)} &= b^{2(t)} \tau^{(t-1)} \mathbf{Y}^{\mathrm{T}} \boldsymbol{\mu}^{(t)},
\end{aligned} \tag{14}$$

where $\|\cdot\|_2$ for a vector denotes the Euclidean norm. Under the variational approximation framework, $q^{(t)}(\mathbf{z})$ and $q^{(t)}(\mathbf{w})$ can be viewed as the approximation to the posteriors $\Pr\left(\mathbf{z} \mid \mathbf{Y}; \boldsymbol{\Theta}^{(t-1)}, F^{(t-1)}(\cdot)\right)$ and $\Pr\left(\mathbf{w} \mid \mathbf{Y}; \boldsymbol{\Theta}^{(t-1)}, F^{(t-1)}(\cdot)\right)$. Naturally, we can infer $\mathbf{z}$ and $\mathbf{w}$ using the posterior mean $\boldsymbol{\mu}^{(t)}$ and $\boldsymbol{\nu}^{(t)}$ after the convergence of the algorithm. With $q^{(t)}$ updated in the tth E-step, the ELBO can be written as

$$\begin{aligned}
&\text{ELBO}\left(q^{(t)}\left(\mathbf{z}, \mathbf{w} \mid \boldsymbol{\mu}^{(t)}, a^{2(t)}; \boldsymbol{\nu}^{(t)}, b^{2(t)}\right); \tau, \beta, F(\cdot)\right) \\
&= \mathbb{E}_{q^{(t)}(\mathbf{z}, \mathbf{w})}\left[\log \Pr(\mathbf{Y}, \mathbf{z}, \mathbf{w} \mid \boldsymbol{\Theta}, F(\cdot))\right] - \mathbb{E}_{q^{(t)}(\mathbf{z}, \mathbf{w})}\left[\log q^{(t)}(\mathbf{z}, \mathbf{w})\right] \\
&= -\frac{\tau}{2}\left(\left\|\mathbf{Y}^{(t)} - \boldsymbol{\mu}^{(t)} \boldsymbol{\nu}^{(t)\mathrm{T}}\right\|_{\mathrm{F}}^2 + \left\|\left(\boldsymbol{\mu}^{(t)2} + a^{2(t)}\right)\left(\boldsymbol{\nu}^{(t)2} + b^{2(t)}\right)^{\mathrm{T}}\right.\right. \\
&\quad \left.\left. - \boldsymbol{\mu}^{(t)2}\left(\boldsymbol{\nu}^{(t)2}\right)^{\mathrm{T}}\right\|_{1,1}\right) + \frac{NM}{2}\log\tau + \frac{N}{2}\log\beta \\
&\quad - \frac{\beta}{2}\left(\left\|\boldsymbol{\mu}^{(t)} - F(\mathbf{X})\right\|_2^2 + N a^{2(t)}\right) + \text{const.}
\end{aligned} \tag{15}$$

Here, for a matrix, $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, and $\|\cdot\|_{1,1}$ denotes the entry-wise matrix norm. We first fix $F^{(t-1)}(\cdot)$ and consider optimizing $\boldsymbol{\Theta}$. By setting the derivatives with respect to $\boldsymbol{\Theta}$ to zero, we obtain the update equations

$$\begin{aligned}
\tau^{(t)} &= \frac{NM}{\left\|\mathbf{Y} - \boldsymbol{\mu}^{(t)} \boldsymbol{\nu}^{(t)\mathrm{T}}\right\|_{\mathrm{F}}^2 + \left\|\left(\boldsymbol{\mu}^{(t)2} + \text{diag}\left(\mathbf{A}^{(t)}\right)\right)\left(\boldsymbol{\nu}^{(t)2} + \text{diag}\left(\mathbf{B}^{(t)}\right)\right)^{\mathrm{T}} - \boldsymbol{\mu}^{(t)2}\left(\boldsymbol{\nu}^{(t)2}\right)^{\mathrm{T}}\right\|_{1,1}}, \\
\beta^{(t)} &= \frac{N}{\left\|\boldsymbol{\mu}^{(t)} - F^{(t-1)}(\mathbf{X})\right\|_2^2 + N a^{2(t)}},
\end{aligned} \tag{16}$$

where $\boldsymbol{\mu}^2 \in \mathbb{R}^{N \times 1}$ denotes Hadamard product $\boldsymbol{\mu}^2 = \boldsymbol{\mu} \odot \boldsymbol{\mu}$, and $\text{diag}(\mathbf{A}) \in \mathbb{R}^{N \times 1}$ denotes a vector containing all the entries on the main diagonal of $\mathbf{A}$. To maximize ELBO with respect to $F(\cdot)$, we propose to update $F(\cdot)$ in a stage-wise manner by sequentially adding $f^{(t)}(\cdot)$ to the current estimate $F^{(t-1)}(\cdot)$

$$\begin{aligned}
f^{(t)}(\cdot) &= \underset{f(\cdot)}{\arg\min} \left\|\boldsymbol{\mu}^{(t)} - F^{(t-1)}(\mathbf{X}) - f(\mathbf{X})\right\|_2^2, \\
F^{(t)}(\cdot) &= F^{(t-1)}(\cdot) + s \cdot f^{(t)}(\cdot),
\end{aligned} \tag{17}$$

where $f^{(t)}(\cdot)$ is a single regression tree to approximate the functional gradient, and $0 < s < 1$ is the so-called shrinkage parameter or learning rate, whose default value is set relatively small ($s = 0.1$) in our implementation to avoid overfitting. Clearly, information in the auxiliary matrix is gradually incorporated to modulate the prior of $\mathbf{z}$. To summarize, the proposed algorithm is a variational EM algorithm and its convergence is naturally guaranteed. The novelty of the proposed algorithm comes from the way we update function $F(\cdot)$, where we combine the gradient boosting strategy (17) into the iterations of our variational EM algorithm. We denote the proposed algorithm to fit the single-factor model as MFAI_SF and summarize it in Algorithm 1.

To assess the scalability of MFAI, we analyze the computational complexity of each step in the proposed algorithm. We begin with the computations of the approximate posterior mean and variance in the E-step, which are given by (14). The updates for both $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ rely on matrix-vector multiplications, resulting

Algorithm 1: Fitting the Single-Factor MFAI Model

Data: main data matrix $\mathbf{Y}$ and auxiliary matrix $\mathbf{X}$
Result: estimate of the latent variables $\widehat{\mathbf{z}} = \boldsymbol{\mu}$ and $\widehat{\mathbf{w}} = \boldsymbol{\nu}$

initialize $q^{(0)}$, $\boldsymbol{\Theta}^{(0)}$, and $F^{(0)}(\cdot) = 0$;
$t \leftarrow 0$;
repeat
  $t \leftarrow t + 1$;
  $\boldsymbol{\mu}^{(t)}, a^{2(t)}; \boldsymbol{\nu}^{(t)}, b^{2(t)} \leftarrow \underset{\boldsymbol{\mu}, a^2; \boldsymbol{\nu}, b^2}{\arg\max}\ \text{ELBO}\left(q\left(\boldsymbol{\mu}, a^2; \boldsymbol{\nu}, b^2\right); \tau^{(t-1)}, \beta^{(t-1)}; F^{(t-1)}(\cdot)\right)$;
  $\tau^{(t)}, \beta^{(t)} \leftarrow \underset{\tau, \beta}{\arg\max}\ \text{ELBO}\left(q\left(\boldsymbol{\mu}^{(t)}, a^{2(t)}; \boldsymbol{\nu}^{(t)}, b^{2(t)}\right); \tau, \beta; F^{(t-1)}(\cdot)\right)$;
  $f^{(t)}(\cdot) \leftarrow \underset{f(\cdot)}{\arg\max}\ \text{ELBO}\left(q\left(\boldsymbol{\mu}^{(t)}, a^{2(t)}; \boldsymbol{\nu}^{(t)}, b^{2(t)}\right); \tau^{(t)}, \beta^{(t)}; F^{(t-1)}(\cdot) + f(\cdot)\right)$;
  $F^{(t)}(\cdot) \leftarrow F^{(t-1)}(\cdot) + s \cdot f^{(t)}(\cdot)$;
until *convergence criterion satisfied*;
return $\boldsymbol{\mu}^{(t)}, a^{2(t)}; \boldsymbol{\nu}^{(t)}, b^{2(t)}; \tau^{(t)}, \beta^{(t)}; F^{(t)}(\cdot)$.

in a computational complexity of $\mathcal{O}(NM)$. The updates for a^2 and b^2 primarily involve the calculation of the ℓ^2-norm and require $\mathcal{O}(M)$ and $\mathcal{O}(N)$ computations, respectively. Moving on to the M-step, we need to update the model parameters $\boldsymbol{\Theta}$ as in (16) and fit a single regression tree as in (17). The update for τ entails matrix multiplications and matrix norm calculations, resulting in a total computational complexity of $\mathcal{O}(NM)$. The update for β also requires the calculation of the ℓ^2-norm, resulting in the computational complexity of $\mathcal{O}(N)$. To fit a single regression tree, we need to sort the data for each node and each auxiliary feature, which takes $\mathcal{O}(N \log N)$ computations. Following this, we traverse the data points to find the best threshold, which takes $\mathcal{O}(N)$ computations. Considering all C auxiliary features, the total computational complexity would be of $\mathcal{O}(CN \log N)$. These complexity analyses provide insights into the computational requirements of the MFAI algorithm and its scalability.

2.2.2. Missing Data

One important feature of MFAI is its ability to handle missing data, either in the main matrix $\mathbf{Y}$ or in the auxiliary matrix $\mathbf{X}$. To handle $\mathbf{Y}$ with missing entries, we first make the typical assumption that they are missing at random (MAR) (Little and Rubin 1987), that is, given the observed data, the missingness does not depend on the unobserved data or latent variables. Then, we can consider the following probabilistic model only for the observed entries $\mathbf{Y}^{\text{obs}}$:

$$\Pr\left(\mathbf{Y}^{\text{obs}} \mid \mathbf{z}, \mathbf{w}; \tau\right) = \prod_{(n,m) \in \Omega^{\text{obs}}} \Pr\left(\mathbf{Y}_{nm} \mid \mathbf{z}, \mathbf{w}; \tau\right), \tag{18}$$

where Ω^{obs} is the collection of the indices of the observed entries of $\mathbf{Y}$. The detailed algorithm within the approximate Bayesian inference framework is similar to that of the complete data case and is included in Appendix Section A.2. As for the missing data in the auxiliary matrix, it is clear that only the update steps involving the auxiliary matrix $\mathbf{X}$ need to be reconsidered. using the *rpart* package (Therneau and Atkinson 2022), any observation with value for the dependent variable (i.e., $\boldsymbol{\mu}_n$) and at least one independent variable (i.e., one of $\{\mathbf{X}_{n1}, \ldots, \mathbf{X}_{nC}\}$) will participate in the modeling. For each split, the observation with the missing split variable will be split based on the best surrogate variable; if that's missing, then by the next best, and so on.

2.2.3. Ranking the Importance of Auxiliary Covariates

In a single tree, the importance of a variable is given by the total goodness of all the splits, either as a primary or a surrogate variable. The higher the importance value, the more the variable contributes to improving the model. By inheriting the merit of the regression trees, the model given by MFAI can be used to rank the importance of auxiliary covariates. Suppose the variable importance of the cth covariate (i.e., $\mathbf{X}_{\cdot c}$) in the tth tree (i.e., $f^t(\cdot)$) is $\mathcal{I}_{tc}$, then the total importance score is given by

$$\mathcal{I}_c = \sum_{t=1}^{T} \mathcal{I}_{tc}, \tag{19}$$

where T is the total number of trees contained in the model.

2.2.4. The Multi-Factor MFAI Model

We now extend the single-factor approach to fit the multi-factor model following Wang and Stephens (2021). To do so, we introduce variational approximations $\{q(\mathbf{Z}_{\cdot k}), q(\mathbf{W}_{\cdot k})\}$ for $k = 1, \ldots, K$, and then optimize ELBO$(q(\mathbf{Z}_{\cdot 1}, \mathbf{W}_{\cdot 1}), \ldots, q(\mathbf{Z}_{\cdot K}, \mathbf{W}_{\cdot K}); \tau, \beta_1, \ldots, \beta_K, F_1(\cdot), \ldots, F_K(\cdot))$. Similar to the single-factor case, the optimization can be done by iteratively updating parameters relating to a single factor while keeping others fixed. The updates of a single pair $\{\mathbf{Z}_{\cdot k}, \mathbf{W}_{\cdot k}\}$ are essentially identical to those for fitting the single-factor model, except that $\mathbf{Y}$ is replaced with the residuals obtained by removing the estimated effects of the other $K - 1$ pairs

$$\mathbf{R}^k = \mathbf{Y} - \sum_{k' \neq k} \mathbf{Z}_{\cdot k'} \mathbf{W}_{\cdot k'}^{\mathrm{T}}. \tag{20}$$

It is worth mentioning that by doing so, we implicitly assume the full factorization of q as

$$q(\mathbf{Z}, \mathbf{W}) = \prod_{k=1}^{K} \mathbf{Z}_{\cdot k} \prod_{k=1}^{K} \mathbf{W}_{\cdot k}, \tag{21}$$

which enjoys a fast computation speed at the cost of a slight decrease in accuracy. We implement two algorithms for fitting the K-factor MFAI model: the greedy algorithm and the backfitting algorithm. The greedy algorithm starts by fitting the single-factor model and then adds factors $k = 2, \ldots, K$, one at a time. The backfitting algorithm (Breiman and Friedman 1985) iteratively refines the estimates for each factor given the estimates for the other factors. In our MFAI framework, we choose to use the greedy algorithm first to provide rough estimates as the initialization for the backfitting algorithm. The two algorithms are both detailed in Appendix Section B.

A practical issue with matrix factorization is how to select the number of factors K. Taking advantage of the additive model and the stage-wise manner to fit the K-factor model sequentially, MFAI can automatically determine K with a little modification to the algorithm. We first set the maximum value of the number of factors $K_{\max}$ and perform the greedy algorithm with K replaced by $K_{\max}$. In this process, if we find the kth factor/loading combination $\boldsymbol{\mu}_k \boldsymbol{v}_k^{\mathrm{T}}$ is very close to zero for one specific $k \in \{1, \ldots, K_{\max}\}$, then we stop the procedure and only use the first $k-1$ factors as the final estimates. The modified algorithm is summarized in Appendix Section B. It is worth emphasizing that our approach is actually very similar to the automatic relevancy determination (ARD) (Babacan et al. 2012), whose key idea is that if the data are consistent with a small absolute value, then the prior precision will be estimated to be large, which results in the shrinkage of the corresponding factor/loading combinations toward zero and hence reduces the rank of the estimate.

3. Numerical Experiments

In this section, we gauge the performance of MFAI in comparison with alternative methods using both simulations and real data analyses. We choose the compared methods based on two considerations. First, they are scalable to large datasets. Second, their software are documented and maintained well. Based on the above criteria, we include EBMF (Wang and Stephens 2021), hardImpute, softImpute (Mazumder, Hastie, and Tibshirani 2010; Hastie et al. 2015), and CMF (Singh and Gordon 2008) in comparison. We note that the Bayesian methods (MFAI and EBMF) are self-tuning. The softImpute has a single tuning parameter λ to control the nuclear norm penalty, which is chosen by cross-validation. We apply CMF with the default settings of R package *cmfrec* (Cortes 2018). In the spirit of reproducibility, the source code and R scripts used to generate the results of our numerical experiments are made publicly available at *https://github.com/YangLabHKUST/mfair*.

3.1. Simulation Studies

3.1.1. Imputation Accuracy

The simulation datasets were generated as follows. For all settings, we fixed $N = 1000$, $M = 1000$, $C = 3$, and $K = 3$. The auxiliary matrix $\mathbf{X} \in \mathbb{R}^{1000\times 3}$ was generated from uniform distribution $\mathbf{X}_{nc} \overset{\text{iid}}{\sim} \mathcal{U}(-10, 10)$. Then we generated latent factors $\mathbf{Z} \in \mathbb{R}^{1000\times 3}$ via functions, $F_1(\mathbf{x}) = \frac{1}{2}x_1 - x_2$, $F_2(\mathbf{x}) = \frac{1}{10}x_1^2 - \frac{1}{10}x_2^2 + \frac{1}{5}x_1x_2$, and $F_3(\mathbf{x}) = 5\sin\left(\frac{1}{100}x_3^3\right)$. We defined the proportion of variance explained (PVE) by $F_k(\mathbf{X})$ as $\mathrm{PVE}_k = \frac{\mathrm{var}(F_k(\mathbf{X}))}{\mathrm{var}(F_k(\mathbf{X}))+\beta_k^{-1}}$, and controlled $\mathrm{PVE}_k = 0.95$ for $k \in \{1, 2, 3\}$. The latent loading matrix $\mathbf{W} \in \mathbb{R}^{1000\times 3}$ was generated from normal distribution $\mathbf{W}_{mk} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$. With $\mathbf{Z}$ and $\mathbf{W}$, we obtained the true value $\mathbf{Y}^{\text{true}} = \mathbf{Z}\mathbf{W}^{\mathrm{T}}$. At last, we added random noises $\mathbf{Y} = \mathbf{Y}^{\text{true}} + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon}_{nm} \sim N(0, \tau^{-1})$. Then the PVE by the factors is defined as $\mathrm{PVE} = \frac{\mathrm{var}(\mathbf{Y}^{\text{true}})}{\mathrm{var}(\mathbf{Y}^{\text{true}})+\tau^{-1}}$. To mimic the real data with a partially observed main matrix, we randomly masked a subset of entries of $\mathbf{Y}$ and denoted their index set as Ω^{miss} and the remaining with index set Ω^{obs}. The missing ratio then can be computed as $\frac{|\Omega^{\text{miss}}|}{NM}$. To evaluate the imputation accuracy, we used half of the entries in Ω^{obs} as the training data and the remaining entries as the test data with index set Ω^{train} and Ω^{test}, respectively. Then, we applied matrix factorization methods to $\mathbf{Y}^{\text{train}}$ and obtained $\widehat{\mathbf{Y}}$. The imputation accuracy can be measured by root-mean-square error (RMSE) on Ω^{test}

$$\mathrm{RMSE}\left(\widehat{\mathbf{Y}}, \mathbf{Y}\right) = \sqrt{\frac{\sum_{(n,m)\in\Omega^{\text{test}}}\left(\widehat{\mathbf{Y}}_{nm} - \mathbf{Y}_{nm}\right)^2}{|\Omega^{\text{test}}|}}. \tag{22}$$

We designed two sets of experiments to test the methods in a wide range of data quality settings. In Experiment 1, we fixed the missing ratio, $\frac{|\Omega^{\text{miss}}|}{NM} = 0.5$ and varied $\mathrm{PVE} \in \{0.1, 0.5, 0.9\}$. In Experiment 2, we fixed the $\mathrm{PVE} = 0.5$ and varied missing ratio $\in \{0, 0.5, 0.9\}$. We specified the number of factors for all methods to be the true value $K = 3$ and repeated the simulations 50 times for each setting. The greedy algorithm of our MFAI with default parameter settings took about 1 min, and the continuing backfitting algorithm took a few more seconds for each experiment using four CPU cores of Intel(R) Xeon(R) Gold 6230N CPU @ 2.30GHz processor on a Linux computing platform.

We summarized the relative RMSE of alternative methods to MFAI in the first row of Figure 1 for Experiment 1. Our MFAI method with backfitting achieved the best accuracy in all settings. When the signal was strong, the RMSE of EBMF, hardImpute, and softImpute were slightly higher than MFAI, while the advantage of MFAI became more evident as the PVE decreased due to its ability to incorporate auxiliary information. Although CMF can also do that, it generally performed poorly in this simulation setting because it can use only the linear model. In Experiment 2, as shown in the second row of Figure 1, the relative performance of alternative methods to MFAI highly depends on the data sparsity. When the main matrix $\mathbf{Y}$ was highly sparse, there was little room for improvement if only $\mathbf{Y}$ was available. Overall, MFAI can significantly outperform other approaches when the data quality is poor, such as weak signal and high sparsity. When the data quality is relatively good, it still retains its competitiveness and achieves slight but steady gains.

3.1.2. Robustness

To exhibit MFAI's ability to distinguish useful auxiliary covariates from irrelevant ones, we rank the importance of the covariates in Figure 2, which has been defined in Section 2.2.3. We already have the auxiliary matrix $\mathbf{X} \in \mathbb{R}^{1000\times 3}$ and the main matrix $\mathbf{Y} \in \mathbb{R}^{1000\times 1000}$ with $\mathrm{PVE} = 0.5$, which were generated as described in Section 3.1.1. To introduce irrelevant

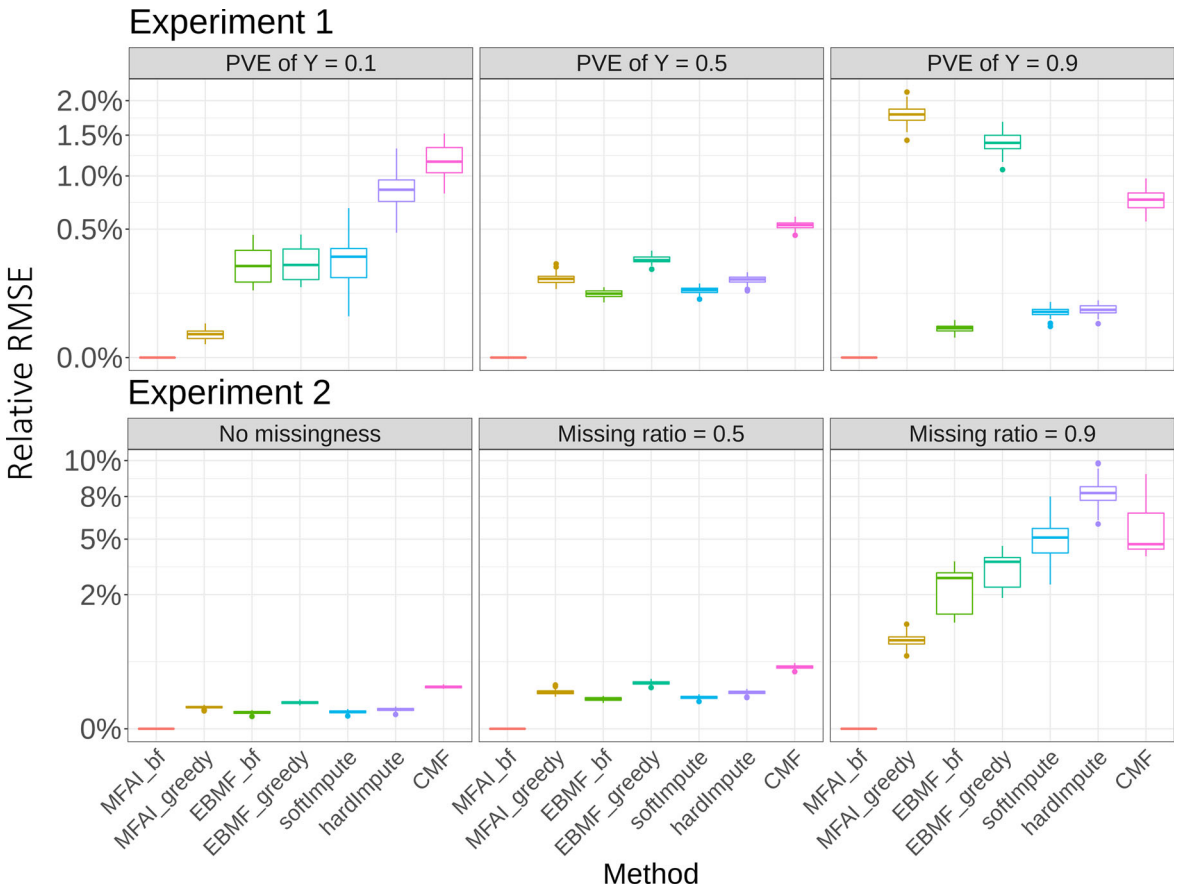

Figure 1. Boxplots comparing the imputation accuracy of different methods. Accuracy is measured by the difference in each method's RMSE from the MFAI's RMSE, then divided by the MFAI's RMSE, with smaller values indicating higher accuracy. The *y* axis is plotted on the square-root scale to avoid the plots being dominated by methods performed poorly.

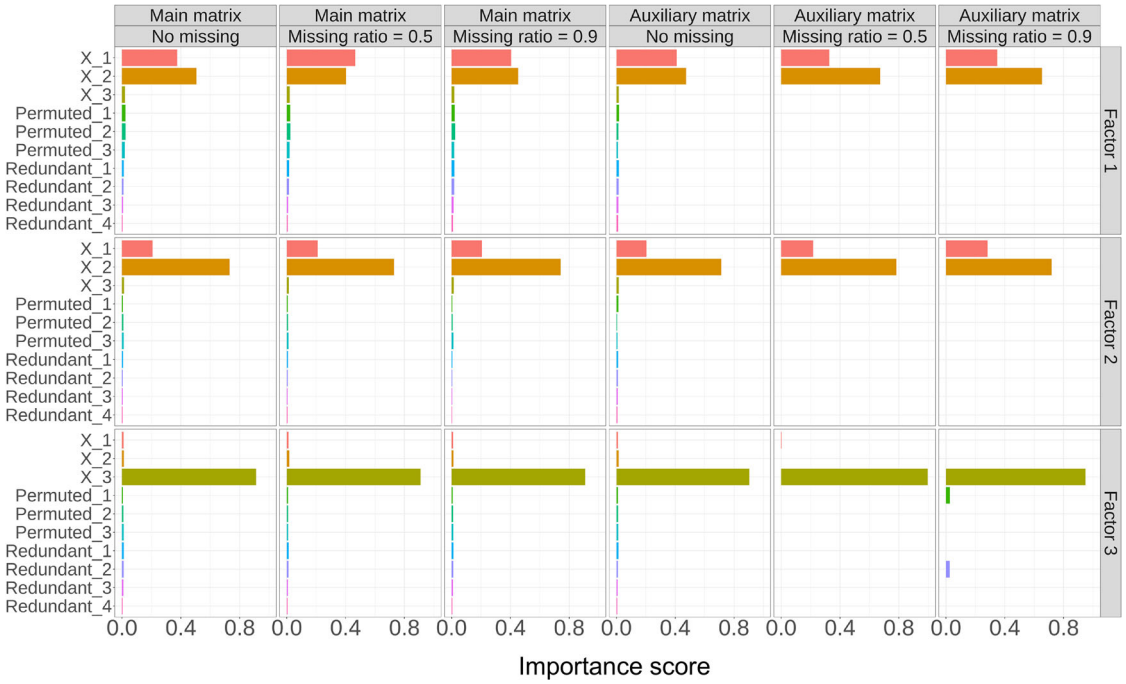

Figure 2. Barplots for the importance scores of the auxiliary covariates in Factor 1–3. The importance scores in each factor have been rescaled to have a sum of one.

covariates, we included three covariates by permuting the rows of $\mathbf{X}$ and four additional redundant variables from the uniform distribution $\mathcal{U}(-10, 10)$, denoted as $\mathbf{X}^{\text{pmt}} \in \mathbb{R}^{1000\times 3}$ and $\mathbf{X}^{\text{rdd}} \in \mathbb{R}^{1000\times 4}$, respectively. At last, we combined them column-wise and got $\mathbf{X}^{\text{all}} = \left[\mathbf{X}, \mathbf{X}^{\text{pmt}}, \mathbf{X}^{\text{rdd}}\right] \in \mathbb{R}^{1000\times 10}$. We applied MFAI to $\mathbf{Y}$ and $\mathbf{X}^{\text{all}}$ in different situations and visualized

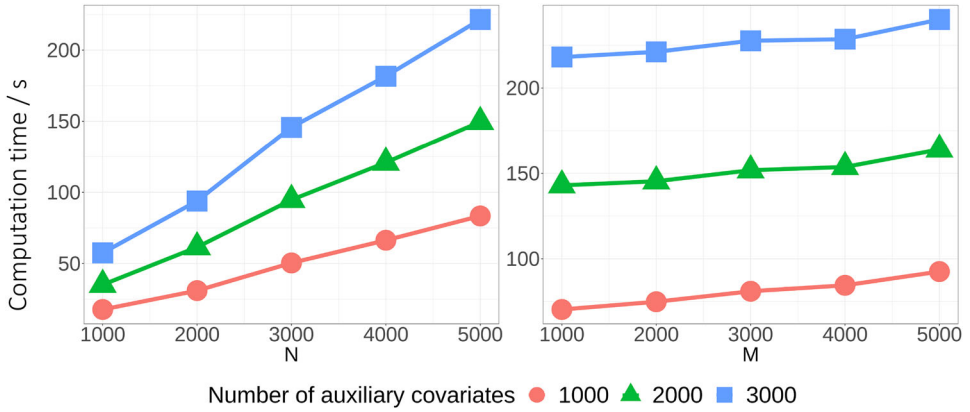

Figure 3. Lineplots for the computation timings against data size.

the importance scores of all the auxiliary covariates in the top three factors. In the left panel (first three columns), we masked the main matrix $\mathbf{Y}$ randomly and varied the missing ratio. In the right panel (next three columns), we fixed the missing ratio of $\mathbf{Y}$ at 0.5, and further masked the auxiliary matrix $\mathbf{X}^{\text{all}}$ randomly at the different missing levels. Figure 2 shows that those unimportant auxiliary covariates get nearly zero importance scores under all data sparsity settings, which indicates that MFAI can effectively distinguish those useful auxiliary covariates, even though the datasets were highly sparse.

3.1.3. Computational Efficiency

Finally, we show the computational efficiency of MFAI (Figure 3). We first fixed the sample size $N = 5000$ and varied the number of features $M \in \{1000, 2000, 3000, 4000, 5000\}$ (the left panel), and then fixed $M = 5000$ and varied $N \in \{1000, 2000, 3000, 4000, 5000\}$ (the right panel). We applied the single-factor MFAI and fixed the number of iteration steps as the same value of 20. Furthermore, the experiments were repeated with different numbers of auxiliary covariates, $C \in \{1000, 2000, 3000\}$, indicated by different colors.

3.2. Real Data Analyses

3.2.1. Data Description and Methods Setup

The two real datasets used in this section are as follows: The *MovieLens 100K data* is extensively used to evaluate recommender system performance (Harper and Konstan 2015). The main matrix $\mathbf{Y} \in \mathbb{R}^{1682\times 943}$ contains 100K observed ratings (0–5 stars), where each row represents a movie and each column represents a user. The auxiliary matrix $\mathbf{X} \in \mathbb{R}^{1682\times 18}$ is a binary matrix indicating movie genre information with 18 genres in total. The *human brain gene expression data* is a fully observed matrix of bulk gene expression from the human brain transcriptome (HBT) project (Kang et al. 2011). We used the measurements in the neocortex areas as the main matrix $\mathbf{Y} \in \mathbb{R}^{886\times 17{,}568}$, with rows representing tissue samples and columns representing genes. Recent studies highlight the importance of region and age in global gene expression differences; therefore, we extracted brain region and time period information, resulting in an auxiliary data frame $\mathbf{X} \in \mathbb{R}^{886\times 2}$. More details can be found in Appendix Section C.

Both Bayesian methods, MFAI and EBMF can automatically estimate K, and we set $K_{\max} = 20$ for MovieLens 100K data and $K_{\max} = 150$ for human brain gene expression data which are sufficiently large. For softImpute, hardImpute, and CMF, we specified K based on the values inferred by MFAI. We first compared the imputation accuracy in terms of the RMSE and then showed that MFAI could illuminate the logic of how the auxiliary information relates to the main data matrix.

3.2.2. Imputation Accuracy

In this section, we examined the imputation performance of the compared methods (Figure 4). First, we randomly split the observed entries Ω^{obs} into a training set Ω^{train} and a test set Ω^{test}. Then, we applied matrix factorization methods to the training data and predicted the entries in the held-out set. The imputation accuracy of the held-out entries was measured by RMSE (22). We considered different values of the "training ratio" which is defined as $\frac{|\Omega^{\text{train}}|}{|\Omega^{\text{obs}}|}$. We repeated the experiments 50 times for each setting of the training ratio. MFAI used only around 2 min to analyze MovieLens 100K data with inferred $K = 9$ and around 150 min to analyze human brain gene expression data with inferred $K = 95$, using four CPU cores of Intel(R) Xeon(R) Gold 6230N CPU @ 2.30GHz processor on a Linux computing platform. By contrast, EBMF, another Bayesian method that cannot incorporate auxiliary information, used around 1 min to analyze MovieLens 100K data and around 130 min to analyze human brain gene expression data using the same computing resources, suggesting that MFAI can leverage auxiliary information with only minor computational overhead.

We summarized the results in Figure 4. For the MovieLens 100K data, MFAI and CMF outperformed other methods by incorporating the movie genre information, suggesting the movie genre provides useful information to predict user ratings. MFAI gained greater improvement from the movie genre information than CMF because the gradient boosted tree offers a more flexible structure than the linear model in CMF. The auxiliary information of the human brain gene expression data comes from two different sources: regions and time periods, where regions are represented as categorical variables and time periods are represented as numerical variables. MFAI also achieved the best performance because the tree structure in MFAI is very good at handling mixed data types (i.e., categorical and

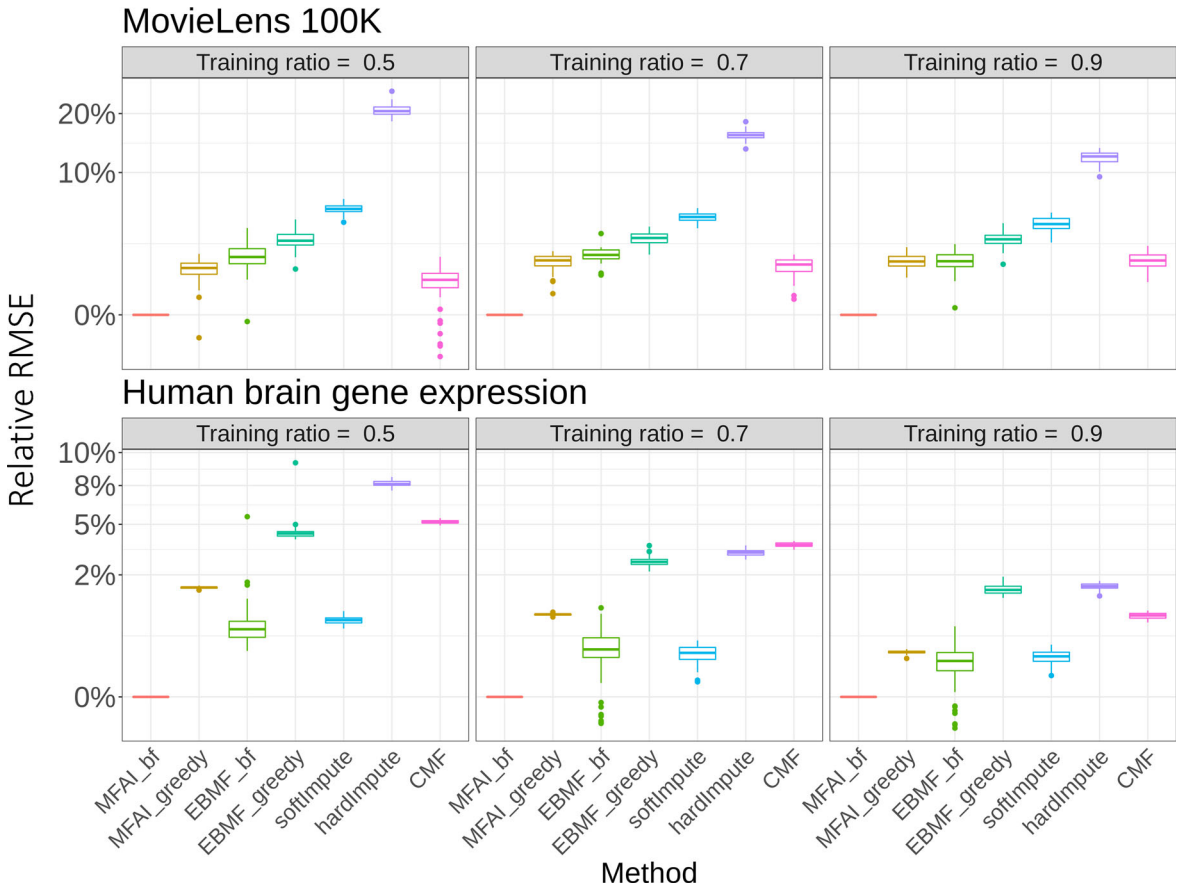

Figure 4. Boxplots comparing the imputation accuracy of different methods. Accuracy is measured by the difference in each method's RMSE from the MFAI's RMSE, then divided by the MFAI's RMSE, with smaller values indicating higher accuracy. The *y* axis is plotted on the square-root scale to avoid the plots being dominated by methods performed poorly.

numerical variables). In contrast, CMF did not perform well in this dataset, which may be attributed to the fact that the linear models are often not good at handling mixed data types and capturing possible spatial-temporal interaction effects in the gene expression data. These evidence suggests that MFAI can effectively leverage auxiliary information to improve imputation accuracy in highly sparse datasets.

3.2.3. Enrichment of Movie Genres in MovieLens 100K Data Analysis

In this section, we use MovieLens 100K data to illustrate the ability of MFAI to identify important variables in auxiliary information through decision trees (Figure 5). As a negative control, we constructed a permuted genre matrix $\mathbf{X}^{\text{pmt}} \in \mathbb{R}^{1682\times 18}$, where the cth column $\mathbf{X}^{\text{pmt}}_{\cdot c}$ was obtained by permuting the entries of $\mathbf{X}_{\cdot c}$ for $c = 1, \ldots, 18$. We applied MFAI to the whole MovieLens 100K data with three different auxiliary matrices $\mathbf{X}$, $\mathbf{X}^{\text{pmt}}$, and $\mathbf{X}^{\text{both}} = \left[\mathbf{X}, \mathbf{X}^{\text{pmt}}\right] \in \mathbb{R}^{1682\times 36}$. Figure 5 visualizes the importance scores of auxiliary covariates in the top three factors. The top left panel shows the importance scores obtained with $\mathbf{X}$, which indicates the relevance of true movie genres to user ratings. In Factor 1, "Drama" is the leading genre; "Action" and "Children's" are the two major genres in Factor 2; "Musical", "Children's", and "Comedy" influence Factor 3. When using the permuted matrix $\mathbf{X}^{\text{pmt}}$ as input (bottom left panel), MFAI correctly assigned low importance scores to all permuted genres and avoided incorporating irrelevant auxiliary information. Finally, in the presence of both true and permuted movie genres (right panel), MFAI successfully distinguished useful movie genres from irrelevant ones. We can also observe that the importance scores obtained using $\mathbf{X}^{\text{both}}$ are highly consistent with those obtained using $\mathbf{X}$ and $\mathbf{X}^{\text{pmt}}$ as separate inputs, indicating the stability and robustness of MFAI.

3.2.4. Spatial and Temporal Dynamics of Gene Regulation Among Tissues

The spatial and temporal patterns of gene regulation during brain development have attracted a great deal of attention in the neuroscience community. The aforementioned human brain gene expression data has been analyzed by several statistical methods (Lin et al. 2015, 2017). By modeling the relationship between the spatial-temporal information and the gene expression matrix via nonlinear functions $F_k(\cdot)$, MFAI can offer biological insights into the heterogeneity in temporal dynamics across different brain regions and the evolution of spatial patterns over multiple time periods. Following Hawrylycz et al. (2015), we selected genes with consistent spatial patterns across individuals using differential stability (DS), which was defined as the tendency for a gene to exhibit reproducible differential expression relationships across brain structures (see Appendix Section C.2.2 for details). As inputs of MFAI, we included 2000 genes with the highest DS, resulting in the new $\mathbf{Y} \in \mathbb{R}^{886\times 2000}$, and used the same $\mathbf{X} \in \mathbb{R}^{886\times 2}$ with spatial and temporal information.

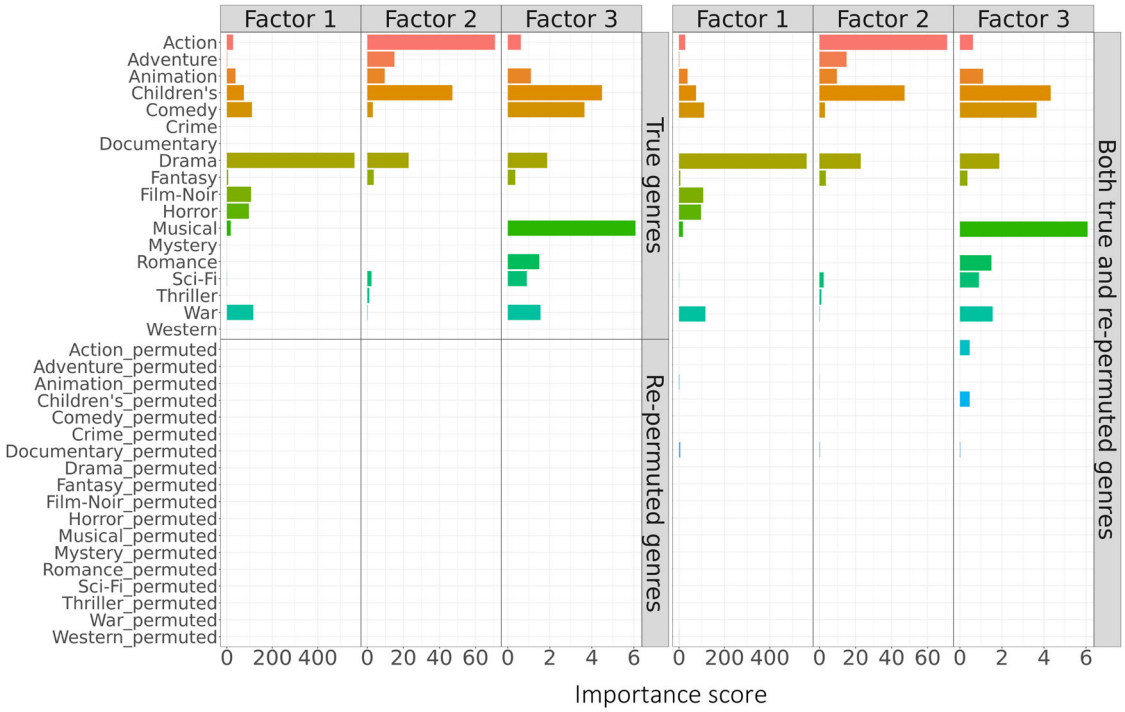

Figure 5. Barplots for the importance scores of the auxiliary covariates in Factor 1–3.

Table 1. Gene enrichment analysis on Loading 1–3.

	Biological process	*P*-value with Bonferroni correction
Loading 1	Axon development	1.97×10^{-2}
	Neuron development	1.82×10^{-3}
	Neuron differentiation	8.03×10^{-5}
Loading 2	Regulation of biological quality	2.25×10^{-10}
	Potassium ion transmembrane transport	6.25×10^{-5}
	Regulation of transport	2.84×10^{-7}
	Signaling	5.35×10^{-5}
	Cell communication	1.02×10^{-4}
Loading 3	Regulation of cell junction assembly	4.04×10^{-3}
	Cell adhesion	8.07×10^{-4}
	Cell junction assembly	3.84×10^{-4}
	Nervous system development	5.58×10^{-5}

To gain insights, the dynamic patterns of the top three factors across different neocortex areas and time periods, represented by fitted functions $\{F_1(\cdot), F_2(\cdot), F_3(\cdot)\}$, are given in Figure 6(A). Each factor has been normalized to have the ℓ^2-norm equal one. It is obvious that fitted functions not only capture the nonlinearity across different time periods but also implicate spatial-temporal interactions. Overall, all three factors show stronger temporal differences compared to spatial differences within the neocortex areas. The temporal trajectories of all three factors show clear signs of prenatal development (from Period 3 to Period 7). From infancy (Period 8 and afterward), Factor 2 exhibits increasing influence, while Factor 3 exhibits decreasing influence in magnitude. Then, all three factors maintain steady levels until late adulthood. All the non-V1C neocortex areas show particularly pronounced correlations and consistency during development. Factor 1 and Factor 3 in V1C showed distinctive signals throughout development and adulthood, compared to other neocortex areas.

Figure 6(B) is the heatmap of the top three inferred gene loadings $[\mathbf{W}_{\cdot 1}, \mathbf{W}_{\cdot 2}, \mathbf{W}_{\cdot 3}] \in \mathbb{R}^{2000\times 3}$. To understand them better, we conducted the gene set enrichment analysis based on Gene Ontology (*http://geneontology.org/*). Specifically, we first calculated the relative weight of the kth loading for the mth gene by $\frac{|\mathbf{W}_{mk}|}{\sum_{k'=1}^{3}|\mathbf{W}_{mk'}|}$, and then selected the top 300 weighted genes in each loading to form the gene sets. The enriched biological processes with corresponding p-values after Bonferroni correction are summarized in Table 1. Loading 1 relates to axon and neuron development, consistent with its status as the leading factor in the neocortex and relatively high signal level across all periods, as shown in Figure 6(A). Loading 2 is enriched in signaling (Luebke et al. 2004) and cell communication (López-Otín et al. 2013), which are aging-related processes. Combining Figure 6(A) and (B), the enrichment of Loading 2 in the ion transport provides evidence that the interstitial ion is a key regulator of state-dependent neural activity (Rasmussen et al. 2020). Loading 3 is mainly enriched in the cell junction, which plays an important role during the development of the mammalian brain. In the mammalian central nervous system (CNS), coupling of neurons by gap junctions (i.e., electrical synapses) and the expression of the neuronal gap junction protein, connexin 36 (Cx36), transiently increases during early postnatal development, then subsequently declines and remains low in adulthood, confined to specific subsets of neurons (Belousov and Fontes 2013). This trend is highly consistent with the temporal pattern of Factor 3 shown in Figure 6(A), reaching a brief high magnitude around birth and quickly falling back.

4. Discussion

The auxiliary information is particularly useful to improve matrix factorization when the observed matrix is noisy and

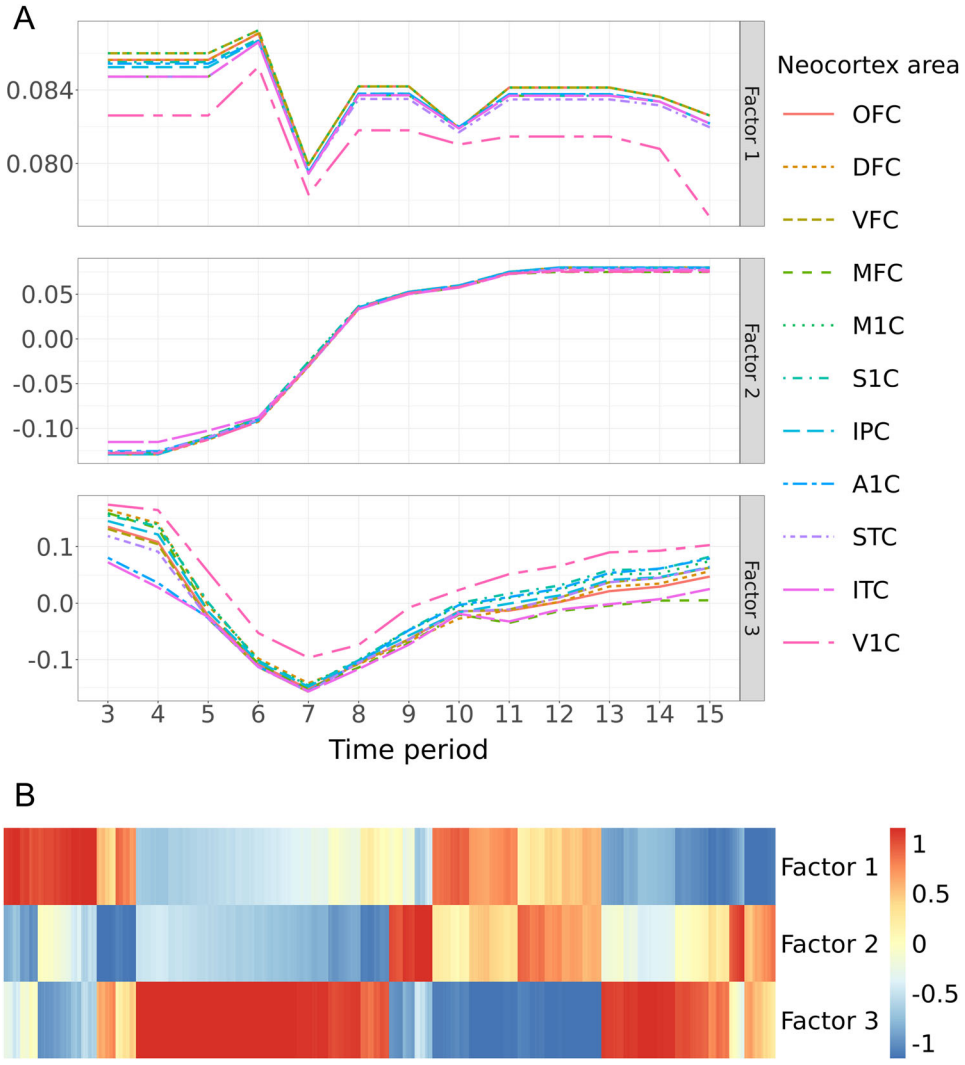

Figure 6. Spatial-temporal dynamic patterns. (A) shows the normalized factor levels across different neocortex areas and time periods. (B) is the heatmap of the corresponding normalized loadings.

sparse. In this article, we propose a scalable Bayesian matrix factorization approach named MFAI to leverage auxiliary information. By integrating the gradient boosted trees with probabilistic matrix factorization, MFAI enables nonlinear modeling of auxiliary covariates and allows the model parameters to be automatically estimated under the empirical Bayes framework, making MFAI adaptive to the complicated connections between the main matrix and auxiliary information. Besides, MFAI naturally inherits several salient features of gradient boosted trees, such as robustness to irrelevant features, immunity to missing values in predictors, and the ability to distinguish useful covariates in the auxiliary information. With our innovations in the model and algorithm designs, our *mfair* software is effective, stable, and scalable to large datasets. Through comprehensive experiments, we showed that MFAI is statistically accurate, especially in the scenario of high sparsity and weak signal strength.

Taking advantage of probabilistic modeling, the MFAI can be modified easily to further introduce other properties, such as sparsity through spike and slab prior distribution over factors and loadings. Another potential extension is to incorporate auxiliary information not only of the samples to characterize the factors but also of the features to help identify the loadings.

Supplementary Materials

Appendices: The online appendix file contains the detailed derivation, algorithms, and data description (Appendices.pdf).

R Package: The R-package *mfair* contains the codes used in fitting the MFAI model and analyzing the results (mfair_1.0.0.tar.gz).

Implementations of Compared Methods:

flashr https://github.com/stephenslab/flashr

cmfrec https://cran.r-project.org/package=cmfrec

softImpute https://cran.r-project.org/package=softImpute

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

This work is supported in part by Hong Kong Research Grant Council Grants 16307818, 16301419, 16308120, and 16307221; Hong Kong Innovation and Technology Fund Grant PRP/029/19FX; The Hong Kong University of Science and Technology Startup Grants R9405 and Z0428 from the Big Data Institute; and City University of Hong Kong Startup Grant 7200746. The computation tasks for this work were performed using the X-GPU cluster supported by the Research Grants Council Collaborative Research Fund Grant C6021-19EF.

ORCID

Zhiwei Wang http://orcid.org/0000-0001-7682-0070
Xianghong Hu http://orcid.org/0000-0001-5260-3936
Mingxuan Cai http://orcid.org/0000-0003-4011-8292
Can Yang http://orcid.org/0000-0002-4407-3055

References

Aktukmak, M., Yilmaz, Y., and Uysal, I. (2019), "A Probabilistic Framework to Incorporate Mixed-Data Type Features: Matrix Factorization with Multimodal Side Information," *Neurocomputing*, 367, 164–175. [1]

Babacan, S. D., Luessi, M., Molina, R., and Katsaggelos, A. K. (2012), "Sparse Bayesian Methods for Low-Rank Matrix Estimation," *IEEE Transactions on Signal Processing*, 60, 3964–3977. [5]

Belousov, A. B., and Fontes, J. D. (2013), "Neuronal Gap Junctions: Making and Breaking Connections during Development and Injury," *Trends in Neurosciences*, 36, 227–236. [9]

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, New York: Springer. [2,3]

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017), "Variational Inference: A Review for Statisticians," *Journal of the American Statistical Association*, 112, 859–877. [2,3]

Breiman, L. (1984), *Classification and Regression Trees*, Franklin: Wadsworth International Group. [2]

Breiman, L., and Friedman, J. H. (1985), "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, 80, 580–598. [5]

Candès, E. J., and Recht, B. (2009), "Exact Matrix Completion via Convex Optimization," *Foundations of Computational Mathematics*, 9, 717–772. [1]

Cortes, D. (2018), "Cold-Start Recommendations in Collective Matrix Factorization," arXiv preprint arXiv:1809.00366. [5]

Elith, J., Leathwick, J. R., and Hastie, T. (2008), "A Working Guide to Boosted Regression Trees," *Journal of Animal Ecology*, 77, 802–813. [2]

Friedman, J. H. (2001), "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, 29, 1189–1232. [2]

Harper, F. M., and Konstan, J. A. (2015), "The MovieLens Datasets: History and Context," *ACM Transactions on Interactive Intelligent Systems*, 5, 1–19. [7]

Hastie, T., Mazumder, R., Lee, J. D., and Zadeh, R. (2015), "Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares," *Journal of Machine Learning Research*, 16, 3367–3402. [5]

Hawrylycz, M., Miller, J. A., Menon, V., Feng, D., Dolbeare, T., Guillozet-Bongaarts, A. L., Jegga, A. G., Aronow, B. J., Lee, C.-K., Bernard, A., et al. (2015), "Canonical Genetic Signatures of the Adult Human Brain," *Nature Neuroscience*, 18, 1832–1844. [8]

Hubbard, C., and Hegde, C. (2017), "Parallel Computing Heuristics for Low-Rank Matrix Completion," in *2017 IEEE Global Conference on Signal and Information Processing*, pp. 764–768. IEEE. [1]

Ilin, A., and Raiko, T. (2010), "Practical Approaches to Principal Component Analysis in the Presence of Missing Values," *The Journal of Machine Learning Research*, 11, 1957–2000. [1]

Kang, H. J., awasawa, Y. I., Cheng, F., Zhu, Y., Xu, X., Li, M., Sousa, A. M., Pletikos, M., Meyer, K. A., Sedmak, G., et al. (2011), "Spatio-Temporal Transcriptome of the Human Brain," *Nature*, 478, 483–489. [7]

Kula, M. (2015), "Metadata Embeddings for User and Item Cold-Start Recommendations," in *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems* (Vol. 1448), pp. 14–21. [1]

Lin, Z., Sanders, S. J., Li, M., Sestan, N., State, M. W., and Zhao, H. (2015), "A Markov Random Field-based Approach to Characterizing Human Brain Development Using Spatial–Temporal Transcriptome Data," *The Annals of Applied Statistics*, 9, 429–451. [8]

Lin, Z., Wang, T., Yang, C., and Zhao, H. (2017), "On Joint Estimation of Gaussian Graphical Models for Spatial and Temporal Data," *Biometrics*, 73, 769–779. [8]

Lin, Z., Yang, C., Zhu, Y., Duchi, J., Fu, Y., Wang, Y., Jiang, B., Zamanighomi, M., Xu, X., Li, M., et al. (2016), "Simultaneous Dimension Reduction and Adjustment for Confounding Variation," *Proceedings of the National Academy of Sciences*, 113, 14662–14667. [1]

Little, R. J., and Rubin, D. B. (1987), *Statistical Analysis with Missing Data*, Hoboken, NJ: Wiley. [4]

López-Otín, C., Blasco, M. A., Partridge, L., Serrano, M., and Kroemer, G. (2013), "The Hallmarks of Aging," *Cell*, 153, 1194–1217. [9]

Luebke, J., Chang, Y.-M., Moore, T., and Rosene, D. (2004), "Normal Aging Results in Decreased Synaptic Excitation and Increased Synaptic Inhibition of Layer 2/3 Pramidal Cells in the Monkey Prefrontal Cortex," *Neuroscience*, 125, 277–288. [9]

Mazumder, R., Hastie, T., and Tibshirani, R. (2010), "Spectral Regularization Algorithms for Learning Large Incomplete Matrices," *Journal of Machine Learning Research*, 11, 2287–2322. [1,5]

Neal, R. M. (1993), "Probabilistic Inference Using Markov Chain Monte Carlo Methods," Technical Report, Department of Computer Science, University of Toronto. [3]

Rasmussen, R., O'Donnell, J., Ding, F., and Nedergaard, M. (2020), "Interstitial Ions: A Key Regulator of State-Dependent Neural Activity?" *Progress in Neurobiology*, 193, 101802. [9]

Salakhutdinov, R., and Mnih, A. (2007), "Probabilistic Matrix Factorization," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 1257–1264. [1]

Shang, L., and Zhou, X. (2022), "Spatially Aware Dimension Reduction for Spatial Transcriptomics," *Nature Communications*, 13, 7203. [1]

Sigrist, F. (2022), "Gaussian Pocess Boosting," *Journal of Machine Learning Research*, 23, 1–46. [2]

Singh, A. P., and Gordon, G. J. (2008), "Relational Learning via Collective Matrix Factorization," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 650–658. [1,5]

Srebro, N., Rennie, J. D., and Jaakkola, T. S. (2004), "Maximum-Margin Matrix Factorization," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, pp. 1329–1336. [1]

Therneau, T., and Atkinson, B. (2022), *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1.19. [4]

Velten, B., Braunger, J. M., Argelaguet, R., Arnol, D., Wirbel, J., Bredikhin, D., Zeller, G., and Stegle, O. (2022), "Identifying Temporal and Spatial Patterns of Variation from Multimodal Data Using MEFISTO," *Nature Methods*, 19, 179–186. [1]

Wang, W., and Stephens, M. (2021), "Empirical Bayes Matrix Factorization," *Journal of Machine Learning Research*, 22, 1–40. [4,5]

Yang, C., Wang, L., Zhang, S., and Zhao, H. (2013), "Accounting for Non-genetic Factors by Low-Rank Representation and Sparse Regression for eQTL Mapping," *Bioinformatics*, 29, 1026–1034. [1]

Yilmaz, Y., Aktukmak, M., and Hero, A. O. (2021), "Multimodal Data Fusion in High-Dimensional Heterogeneous Datasets via Generative Models," *IEEE Transactions on Signal Processing*, 69, 5175–5188. [1]

Zhou, X., Yang, C., and Yu, W. (2012), "Moving Object Detection by Detecting Contiguous Outliers in the Low-Rank Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 597–610. [1]

Zhou, X., Yang, C., Zhao, H., and Yu, W. (2014), "Low-Rank Modeling and its Applications in Image Analysis," *ACM Computing Surveys*, 47, 1–33. [1]