Erasmus University Rotterdam

Erasmus School of Economics

Master thesis

Econometrics & Management Science

Business Analytics and Quantitative Marketing

Additional information and probabilistic matrix factorization for large sparse binary datasets: a minimization by majorization approach

Thijs de Bruin
427882

Supervisor: Prof. Dr. Patrick Groenen

Second assessor: Prof. Dr. Dennis Fok

January 28, 2021

Contents

1 Introduction

Recommendations are an integral part of any successful online service or store. Companies such as Netflix use recommendations to provide their users with a personalized selection of content. The value of correct recommendations cannot be understated as the best recommendations can lead to a competitive advantage in an age where switching costs and brand loyalty are low. The Netflix prize, a competition where the online video streaming service Netflix challenged teams of researchers to beat their recommendation engine for a million dollar prize, highlights this importance. Today, many different recommender systems exist. Correct design and implementation of these systems depends on the type of target users and their context; the devices that they would use; the role of the recommendation within the application; the goal of the recommendation; and the data that is available (Ricci et al., 2011).

Content-based (CB) recommender systems use information, such as documents, item descriptions or other information about items the user previously interacted with, to model the users interests based on the features of the object rated by the user (Lops et al., 2011). Each system consists of three steps. In the first step, the available information is analyzed and features are extracted. Documents might be unstructured and some pre-processing might be needed to extract useful features. In the second step the user profile is constructed. In the context of video recommendations this step might use likes and dislikes in combination with the features extracted in the first step. These preferences for features are generalized to extract a user profile which contains the likes and dislikes of the user. The last part of a CB recommender system is the filtering step. In this step the user profile is exploited and items matching the profile are recommended.

Collaborative filtering (CF) uses the opinions of other people to recommend items. CF techniques are based on the idea that humans have been recommending items to each other since far before the rise of the internet. If enough of your friends advise you to watch the latest movie, you might decide to see it for yourself. You might even have that one friend who is known to recommend great movies opinion you value more highly (Schafer et al., 2007). The internet allows us move beyond simple word-of-mouth and compare our tastes and preferences with thousands, if not millions of other users. Generally, CF methods outperform CB methods, although CF methods suffer cold start problems which restrict CF methods to address new products and users (Koren et al., 2009).

Latent factor models are a type of CF method which assume that the interest in items by a set of users are based on a relatively small set of latent factors in both the users and the items. In the context of a movie these factors might by straightforward, such as genre of the movie, or more obscure, such as the feeling accompanying the movie. A very popular and widely successful realization of the latent factor model is Matrix factorization (MF). In fact, MF models were implemented by the BellKor's Pragmatic Chaos team to win the Netflix Prize. In it's most basic form, MF models try to find a low-rank approximation of user item matrix, containing all combinations of users and items. This low rank approximation consists of user and item vectors which are multiplied to fill the user-item matrix. Input in an MF model can range from explicit to implicit. The most convenient input data is high-quality explicit feedback, such as ratings on a scale of 1-5, as this directly reflects the users preferences.

Some MF methods use implicit feedback such as clicks and page views. Implicit feedback can be collected at a much larger and faster scale compared to explicit feedback, without a need for the user to provide explicit sentiment. An example of a MF method using implicit data is Logistic Matrix Factorization (LMF) Johnson (2014). Besides the traditional user and item vectors LMF implements user and item biases. These biases allow the distinction between different types of users and items. Some users might be more willing to click on any item than others.

MF methods have grown in popularity due to their ability to scale and the predictive accuracy Koren et al. (2009). The importance of the ability to scale cannot be understated as the amount of collected data keeps increasing in our increasingly data driven world. The Netflix prize dataset consisted of 100 million ratings from 480 thousand anonymous subscribers on nearly 18 thousand movie titles (Bennett et al., 2007). Storage of the dense user item matrix alone would require over 30 Gigabytes of memory[1]. This is only glimpse at the scale companies such as Netflix deal with as Netflix is expected to have 200 million paying customers in the third quarter of 2020. Another characteristic of these data sets is that they are often highly sparse. In the Netflix example only 100 million ratings were given, such that only 1.2% of the user item matrix is observed.

The sparsity of the user item matrix can be exploited to easily handle matrices of very large dimensions. Mazumder et al. (2010) show that the user item matrix can be decomposed in a sparse and low rank part in the context of so called nuclear-norm-regularized matrix approximation. Hastie et al. (2015) use this sparse plus low rank data structure to derive an alternating least squares method for maximum-margin matrix factorization (MMMF) called the `softImpute-ALS` algorithm. The use of the sparse plus low rank data structure has yet to be applied in a probabilistic context, such as LMF. Additionally, LMF methods have yet to be applied in a sparse binary context where the users are only shown a subset of the data, such as in a mailing campaign. In this context, we observe three types of feedback from the user: they see the item and click, they see the item and don't click or they don't see the item at all. de Bruin et al. (2020) makes a first attempt to fill this gap in the literature and introduces a probabilistic matrix factorization model using a minimization by majorization procedure. As we build on their work, a thorough understanding of this model is needed in this paper. Therefore we will discuss and derive the model in full in Section 3.3.

Combining various sources of information and/or predictions from different models is a strategy which is often used in recommender systems to their increase predictive power. de Bruin et al. (2020) employs this hybridization by combining the predictions from two different models which use different sources of information. In this thesis, we will look for ways to incorporate additional information directly into the

[1] 4 bytes × 480.000 × 18.000

probabilistic matrix factorization model discussed in their paper. Hence, our research question reads: *Can we incorporate additional information in a sparse binary probabilistic matrix factorization algorithm using the sparse plus low rank data structure*?

The remainder of the paper will be structured as follows. In Section 2, we will briefly discuss some of the related literature. In Section 3, we first describe the unrestricted model formulated in de Bruin et al. (2020). Afterwards, we introduce both fully restricted and partially restricted probabilistic MF models. In Section 4, we will perform a series of timing results to test our models ability to scale. In Section 5, we test the performance of our models using various data sets obtained from a large online tour operator, Sunweb. Finally, we conclude and discuss our findings in Section 6.

2 Related works

The term Logistic Matrix Factorization (LMF) was coined in Johnson (2014) and describes a probabilistic model for matrix factorization with implicit feedback. This collaborative filtering method is probabilistic in that it attempts to encode the probability of a user choosing to interact with an item using a statistical distribution. The authors propose the use of a logistic function to model the probability of a user choosing an item. Various other statistical functions have been used to model the problem structure in the context of collaborative filtering. Mnih and Teh (2012) provides a probabilistic collaborative filtering method based on implicit feedback data for item selection process of users. They model this behaviour using a normalized exponential function. Additionally, the normalization to the distribution is approximated using a tree structure, easing computation times. Alternatively, Gopalan et al. (2013) introduces a MF model which uses a Poisson distribution to model the user and item factors.

Logistic models are often used in the context of binary data. The probability of a click is modelled by a binomial distribution where the mean of this distribution is modelled by the logistic function. Groenen et al. (2003) implements an iterative majorization algorithm for a logistic bi-additive model. Majorization is a technique where a complex objective function is approximated by an easily optimizable, often convex, majorization function. We can draw multiple similarities between the design of the LMF discussed in Johnson (2014) and the logistic bi-additive model. In the context of MF, both models contain user and item biases as well as a set of low rank user and item vectors. Both models are probabilistic, but differ in the way in which they optimize the parameters of their model. Johnson (2014) optimizes their objective function by performing an alternating gradient ascent procedure, whereas Groenen et al. (2003) is able to implement a rather elegant iterative majorization algorithm to derive and solve a weighted least squares problem in each iteration.

MF and Singular Value Decomposition (SVD) are closely related. However, traditional SVD is undefined when dealing with sparse matrices. Early MF systems relied on filling a dense user item matrix. This method was both computationally burdensome and relied on the often inaccurate imputation of the missing observations Koren et al. (2009). More modern methods only use the observed data whilst avoiding overfitting through regularization using the nuclear norm (see (Mnih and Salakhutdinov, 2008), (Johnson, 2014) and (Hastie et al., 2015)). The nuclear norm is equal to the sum of the singular values of the user item matrix. Implementation of the nuclear norm in the objective function results in a compression of the rank of the low rank approximation.

3 Methodology

First, we define some basic notation used in the remainder of the methodology. We define $\boldsymbol{Y}$ as an $n_u \times n_i$ sparse matrix corresponding to a dataset containing some sort of sparse binary feedback. To provide some context, later in the paper we will work with a dataset containing the click behaviour of customers. In this case $y_{ui} = 1$ if the user u has clicked on item i and $y_{ui} = 0$ if the user u has seen item i but has not clicked on the item. A missing value corresponds to user u not having been exposed to item i. The probability π_{ui} of a response for a binary variable y_{ui} is often modelled with a binomial distribution. We model the items the user is most interested in by approximating the mean of the binomial distribution μ_{ui}.

3.1 Logistic Bi-Additive Model

Groenen et al. (2003) implements a logistic bi-additive model which is used for the analysis of binary data. de Bruin et al. (2020) used this model as inspiration for the model discussed in Section 3.3. The mean of the binomial distribution μ_{ui} is modelled using a logistic function

$$\mu_{ui} = \frac{e^{\gamma_{ui}}}{1 + e^{\gamma_{ui}}}. \tag{1}$$

The elements of this logistic function, γ_{ui}, are captured in an $n_u \times n_i$ matrix $\boldsymbol{\Gamma}$. $\boldsymbol{\Gamma}$ can in turn be dissected into different sources of variation:

- the main effect for the users in the $n_u \times 1$ vector $\boldsymbol{\alpha}$,
- the main effect for the items in the $n_i \times 1$ vector $\boldsymbol{\beta}$, and
- the bi-additive interaction effect between rows and columns (users and items) estimated by the rank f decomposition $\boldsymbol{C}\boldsymbol{D}'$ with $\boldsymbol{C} = (\boldsymbol{c}'_1, \boldsymbol{c}'_2, \ldots, \boldsymbol{c}'_{n_u})'$ of size $n_u \times f$ and $\boldsymbol{D} = (\boldsymbol{d}'_1, \boldsymbol{d}'_2, \ldots, \boldsymbol{d}'_{n_i})'$ of size $n_i \times f$.

As such, $\boldsymbol{\Gamma}$ can be written as

$$\boldsymbol{\Gamma} = \boldsymbol{\alpha}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}' + \boldsymbol{C}\boldsymbol{D}', \tag{2}$$

where $\boldsymbol{\beta}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are column-centered to maintain unique identification. $\boldsymbol{Y}$ is characterized by its sparsity. As such, we introduce a set Ψ which contains all observed pairs of user item combinations (u, i).

Using this Ψ the likelihood function becomes

$$\mathcal{L}(\boldsymbol{Y}|\boldsymbol{\Gamma}) = \prod_{(u,i)\in\Psi} \mu_{ui}^{y_{ui}}(1-\mu_{ui})^{(1-y_{ui})}. \tag{3}$$

If we take the natural logarithm of the likelihood function we find

$$\log\mathcal{L}(\boldsymbol{\Gamma}|\boldsymbol{Y}) = \sum_{(u,i)\in\Psi} y_{ui}\log(\mu_{ui}) + (1-y_{ui})\log(1-\mu_{ui}). \tag{4}$$

3.2 Nuclear norm

It is common to use a nuclear norm in the context of low rank matrix completion problems. The nuclear norm is defined as

$$\|\boldsymbol{Z}\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i(\boldsymbol{Z}). \tag{5}$$

Where $\sigma_i(\boldsymbol{Z})$ corresponds to the i'th largest singular value of the $m\times n$ matrix $\boldsymbol{Z}$. One example of regularization using a nuclear norm is described in Mazumder et al. (2010) where a minimization problem is introduced as

$$\min_{\boldsymbol{B}} \quad \frac{1}{2}\|\boldsymbol{E}-\boldsymbol{B}\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_*, \tag{6}$$

where the sparse matrix $\boldsymbol{E}_{m\times n}$ has rank r and $\lambda > 0$ is the regularization parameter. The solution is given by $\hat{\boldsymbol{B}} = \boldsymbol{S}_\lambda(\boldsymbol{E})$, where $\boldsymbol{S}_\lambda(\boldsymbol{E}) \equiv \boldsymbol{U}\boldsymbol{D}_\lambda\boldsymbol{V}'$ with $\boldsymbol{D}_\lambda = \mathrm{diag}[(d_1-\lambda)_+, ..., (d_r-\lambda)_+]$, $\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}'$ is the SVD of $\boldsymbol{E}$, $\boldsymbol{D} = \mathrm{diag}[d_1, ..., d_r]$ and $t_+ = \max(t, 0)$. The notation $\boldsymbol{S}_\lambda(\boldsymbol{E})$ refers to soft-thresholding. The solution demonstrates the compression of the rank and the working of the constraint as for larger values of λ, more singular values are reduced to zero, reducing the rank of the solution. Additionally, Mazumder et al. (2010) show that the nuclear norm is intimately related to the so-called maximum margin matrix factorization methods, MMMF in short. The criterion corresponding to MMMF reads

$$\min_{\boldsymbol{U},\boldsymbol{V}} \quad \frac{1}{2}\|\boldsymbol{E}-\boldsymbol{U}\boldsymbol{V}'\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\left(\|\boldsymbol{U}\|_{\mathrm{F}}^2 + \|\boldsymbol{V}\|_{\mathrm{F}}^2\right). \tag{7}$$

Mazumder et al. (2010) formally prove that for any matrix $\boldsymbol{B}$ it holds that

$$\min_{\boldsymbol{B}} \|\boldsymbol{B}\|_* = \min_{\boldsymbol{U},\boldsymbol{V}:\boldsymbol{B}=\boldsymbol{U}\boldsymbol{V}'} \frac{1}{2}\left(\|\boldsymbol{U}\|_{\mathrm{F}}^2 + \|\boldsymbol{V}\|_{\mathrm{F}}^2\right). \tag{8}$$

We can show the equality of these two criteria with a short and informal proof (not shown in Mazumder et al. (2010)). Given the minimization problem in (6) define the singular value decomposition of $\boldsymbol{U}\boldsymbol{V}'$ as $\boldsymbol{B} = \boldsymbol{P}\boldsymbol{\Phi}\boldsymbol{Q}$, with $\boldsymbol{U} = \boldsymbol{P}\boldsymbol{\Phi}^{\frac{1}{2}}$ and $\boldsymbol{V} = \boldsymbol{Q}\boldsymbol{\Phi}^{\frac{1}{2}}$. Using the properties of the singular value decomposition,

$\boldsymbol{P}'\boldsymbol{P} = \boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{I}$ and $\boldsymbol{\Phi}^{\frac{1}{2}}\boldsymbol{\Phi}^{\frac{1}{2}} = \boldsymbol{\Phi}$, we can rewrite

$$\begin{aligned}
&\min_{\boldsymbol{U},\boldsymbol{V}} \quad \frac{1}{2}\|\boldsymbol{E} - \boldsymbol{U}\boldsymbol{V}'\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\left(\|\boldsymbol{U}\|_{\mathrm{F}}^2 + \|\boldsymbol{V}\|_{\mathrm{F}}^2\right) \\
&= \min_{\boldsymbol{U},\boldsymbol{V}} \quad \frac{1}{2}\|\boldsymbol{E} - \boldsymbol{U}\boldsymbol{V}'\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\left(\|\boldsymbol{P}\boldsymbol{\Phi}^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\boldsymbol{Q}\boldsymbol{\Phi}^{\frac{1}{2}}\|_{\mathrm{F}}^2\right) \\
&= \min_{\boldsymbol{B}} \quad \frac{1}{2}\|\boldsymbol{E} - \boldsymbol{B}\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\operatorname{tr}(\boldsymbol{\Phi}^{\frac{1}{2}}\boldsymbol{P}'\boldsymbol{P}\boldsymbol{\Phi}^{\frac{1}{2}}) + \frac{\lambda}{2}\operatorname{tr}(\boldsymbol{\Phi}^{\frac{1}{2}}\boldsymbol{Q}'\boldsymbol{Q}\boldsymbol{\Phi}^{\frac{1}{2}}) \\
&= \min_{\boldsymbol{B}} \quad \frac{1}{2}\|\boldsymbol{E} - \boldsymbol{B}\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\operatorname{tr}(\boldsymbol{\Phi}) + \frac{\lambda}{2}\operatorname{tr}(\boldsymbol{\Phi}) \\
&= \min_{\boldsymbol{B}} \quad \frac{1}{2}\|\boldsymbol{E} - \boldsymbol{B}\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_{*}. \qquad (9)
\end{aligned}$$

3.3 Minimization by majorization

3.3.1 Majorization

Minimization by majorization (MM) is a technique that is used to simplify complex optimization problems, by replacing the objective function with a majorizing function for which the minimum can be found more easily (de Leeuw and Lange, 2009). The definition of a majorizing function is as follows. A function $g : \mathcal{D}_g \to \mathcal{R}_g$ is considered to be a majorizing function for an objective function $f : \mathcal{D}_f \to \mathcal{R}_f$ at the point x_0 if

$$g(x_0; x_0) = f(x_0), \qquad (10)$$

$$g(x; x_0) \geq f(x), \qquad \forall x \in \mathcal{D}_f. \qquad (11)$$

Additionally, if f and g are both twice differentiable, the following properties should also apply:

$$g'(x_0; x_0) = f'(x_0), \qquad (12)$$

$$g''(x_0; x_0) \geq f''(x_0). \qquad (13)$$

Optimization (i.e. minimization) of f using MM is achieved iteratively by first finding a majorization function g at the current best solution $x^{(k)}$ and updating the solution with $x^{(k+1)}$, where $x^{(k+1)}$ minimizes g. Then, a new majorization function is created at the updated solution after which this is again minimized. This process is repeated until the solution has converged. Following from the properties of majorizing functions shown in (10) and (11), this process guarantees descent of the objective function, as illustrated by the 'sandwich inequality'

$$f(x^{(k+1)}) \leq g(x^{(k+1)}; x^{(k)}) \leq g(x^{(k)}; x^{(k)}) = f(x^{(k)}), \qquad (14)$$

which demonstrates that when g majorizes f at $x^{(k)}$, the x that minimizes g yields an objective value for f that is at least as small as the previous objective value.

de Bruin et al. (2020) implements logistic majorization based on the bi-additive model proposed in Groenen et al. (2003). Instead of maximizing the log likelihood, the authors minimize the negative log likelihood using convex quadratic majorizers. Rewriting the log likelihood function in equation (4) yields the following negative log likelihood

$$-\log \mathcal{L}(\boldsymbol{\Gamma}|\boldsymbol{Y}) = \sum_{(u,i)\in\Psi} y_{ui}\log(1+e^{-\gamma_{ui}}) + (1-y_{ui})(\gamma_{ui} + \log(1+e^{-\gamma_{ui}})). \tag{15}$$

The following terms are majorized

$$\begin{aligned} f_1(\gamma_{ui}) &= \log(1+e^{-\gamma_{ui}}) && \text{if} \quad y_{ui} = 1 \ \wedge \ (u,i)\in\Psi, && (16)\\ f_2(\gamma_{ui}) &= \gamma_{ui} + \log(1+e^{-\gamma_{ui}}) && \text{if} \quad y_{ui} = 0 \ \wedge \ (u,i)\in\Psi, && (17)\\ f_3(\gamma_{ui}) &= 0 && \text{if} \quad (u,i)\notin\Psi, && (18)\end{aligned}$$

using quadratic majorizating functions

$$g(\gamma_{ui};\gamma_{ui}^{(0)}) = a_{ui}\gamma_{ui}^2 - 2b_{ui}\gamma_{ui} + c_{ui}, \tag{19}$$

with $\gamma_{ui}^{(0)}$ indicating the current value of γ_{ui}. Then, the properties from (10) - (13) are used to find parameters a_{ui}, b_{ui}, and c_{ui} as follows. First, second derivatives of the separate terms f_k are found for all $k \in \{1,2,3\}$ and $g(\gamma_{ui};\gamma_{ui}^{(0)})$, given by

$$\begin{aligned} f_1''(\gamma_{ui}) = f_2''(\gamma_{ui}) &= \frac{e^{\gamma_{ui}}}{(1+e^{\gamma_{ui}})^2},\\ f_3''(\gamma_{ui}) &= 0,\\ g''(\gamma_{ui};\gamma_{ui}^{(0)}) &= 2a_{ui}. \end{aligned} \tag{20}$$

The second derivative of f_1 and f_2 has a maximum of $\frac{1}{4}$. Therefore, a_{ui} is set to $\frac{1}{8}$ such that the second derivative of $g(\gamma_{ui};\gamma_{ui}^{(0)})$ is always larger or equal to the second derivatives of f_k, satisfying property (13). All a_{ui} are now restricted to $\frac{1}{8}$ in order to simplify and ease computation. Next, (10) and (12) are used to derive the conditions for b_{ui} and c_{ui}, illustrated below for f_1,

$$f_1(\gamma_{ui}^{(0)}) = g(\gamma_{ui}^{(0)};\gamma_{ui}^{(0)}) = \frac{1}{8}(\gamma_{ui}^{(0)})^2 - 2b_{ui}\gamma_{ui}^{(0)} + c_{ui} \tag{21}$$

$$f_1'(\gamma_{ui}^{(0)}) = g'(\gamma_{ui}^{(0)};\gamma_{ui}^{(0)}) = \frac{1}{4}\gamma_{ui}^{(0)} - 2b_{ui}. \tag{22}$$

Solving these equations for b_{ui} and c_{ui}, we get

$$b_{ui} = \frac{1}{8}\gamma_{ui}^{(0)} - \frac{1}{2}f_1'(\gamma_{ui}^{(0)}), \tag{23}$$

$$c_{ui} = f_1(\gamma_{ui}^{(0)}) + \frac{1}{8}(\gamma_{ui}^{(0)})^2 - f_1'(\gamma_{ui}^{(0)})\gamma_{ui}^{(0)}. \tag{24}$$

Extending the above for f_2 and f_3, and keeping in mind that $f_3(\gamma_{ui}) = f_3'(\gamma_{ui}) = 0$, the parameters of $g(\gamma_{ui}; \gamma_{ui}^{(0)})$ can be defined as

$$\begin{aligned}
a_{ui} &= \frac{1}{8}, \\
b_{ui} &= \begin{cases} \frac{1}{8}\gamma_{ui}^{(0)} - \frac{1}{2}f_1'(\gamma_{ui}^{(0)}) & \text{if } y_{ui} = 1 \ \wedge \ (u,i) \in \Psi, \\ \frac{1}{8}\gamma_{ui}^{(0)} - \frac{1}{2}f_2'(\gamma_{ui}^{(0)}) & \text{if } y_{ui} = 0 \ \wedge \ (u,i) \in \Psi, \\ \frac{1}{8}\gamma_{ui}^{(0)} & \text{if } (u,i) \notin \Psi, \end{cases} \\
c_{ui} &= \begin{cases} f_1(\gamma_{ui}^{(0)}) + \frac{1}{8}(\gamma_{ui}^{(0)})^2 - f_1'(\gamma_{ui}^{(0)})\gamma_{ui}^{(0)} & \text{if } y_{ui} = 1 \ \wedge \ (u,i) \in \Psi, \\ f_2(\gamma_{ui}^{(0)}) + \frac{1}{8}(\gamma_{ui}^{(0)})^2 - f_2'(\gamma_{ui}^{(0)})\gamma_{ui}^{(0)} & \text{if } y_{ui} = 0 \ \wedge \ (u,i) \in \Psi, \\ \frac{1}{8}(\gamma_{ui}^{(0)})^2 & \text{if } (u,i) \notin \Psi, \end{cases}
\end{aligned} \tag{25}$$

where the derivatives of f_1 and f_2 are defined as

$$f_1'(\gamma_{ui}) = -\frac{1}{1 + e^{\gamma_{ui}}}, \tag{26}$$

$$f_2'(\gamma_{ui}) = \frac{1}{1 + e^{-\gamma_{ui}}}. \tag{27}$$

(15) is then rewritten using (25) as

$$-\log \mathcal{L}(\mathbf{\Gamma}|\boldsymbol{Y}) \leq \sum_{(u,i)\in\Psi} \left(\frac{1}{8}\gamma_{ui}^2 - 2b_{ui}\gamma_{ui} + c_{ui} \right), \tag{28}$$

which can be rewritten as the weighted least-squares problem

$$\begin{aligned}
-\log \mathcal{L}(\mathbf{\Gamma}|\boldsymbol{Y}) &\leq \frac{1}{8} \sum_{(u,i)\in\Psi} (h_{ui} - \gamma_{ui})^2 + s \\
&= \frac{1}{8} \|\boldsymbol{H} - \mathbf{\Gamma}\|_{\mathrm{F}}^2 + s,
\end{aligned} \tag{29}$$

where

$$\begin{aligned}
h_{ui} &= 8b_{ui}, \\
s &= \sum_{(u,i)\in\Psi} \left(c_{ui} - 8b_{ui}^2 \right).
\end{aligned} \tag{30}$$

3.3.2 Parameter updates

Using the definitions of b_{ui} in (25), $\boldsymbol{H}$ can be reformulated using a sparse plus low-rank structure. Define the sparse matrix $\boldsymbol{Z_s}$ as

$$z_{ui} = \begin{cases} -4f_1'(\gamma_{ui}^{(0)}) & \text{if } y_{ui} = 1 \ \wedge \ (u,i) \in \Psi, \\ -4f_2'(\gamma_{ui}^{(0)}) & \text{if } y_{ui} = 0 \ \wedge \ (u,i) \in \Psi, \\ 0 & \text{if } (u,i) \notin \Psi. \end{cases} \tag{31}$$

Now, $\boldsymbol{H}$ can be written as

$$\begin{aligned} \boldsymbol{H} &= \boldsymbol{Z}_s + \boldsymbol{\Gamma}^{(0)} \\ &= (\boldsymbol{Z}_s) + \left(\boldsymbol{\alpha}^{(0)}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}^{(0)\prime} + \boldsymbol{C}^{(0)}\boldsymbol{D}^{(0)\prime}\right) \\ &= \text{sparse} + \text{low-rank}. \end{aligned} \tag{32}$$

By pre- and post-multiplying $\boldsymbol{H}$ with centering matrices $\boldsymbol{J}_{n_u}$ and $\boldsymbol{J}_{n_i}$, we get

$$\begin{aligned} \boldsymbol{J}_{n_u}\boldsymbol{H}\boldsymbol{J}_{n_i} &= (\boldsymbol{I}_{n_u} - n_u^{-1}\mathbf{1}\mathbf{1}')\boldsymbol{H}(\boldsymbol{I}_{n_i} - n_i^{-1}\mathbf{1}\mathbf{1}') \\ &= \boldsymbol{H} - n_u^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{H} - n_i^{-1}\boldsymbol{H}\mathbf{1}\mathbf{1}' + (n_u n_i)^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{H}\mathbf{1}\mathbf{1}' \\ &= \boldsymbol{H} - n_i^{-1}\boldsymbol{H}\mathbf{1}\mathbf{1}' - n_u^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{H}\boldsymbol{J}_{n_i}, \end{aligned} \tag{33}$$

which implies

$$\boldsymbol{H} = \boldsymbol{J}_{n_u}\boldsymbol{H}\boldsymbol{J}_{n_i} + n_i^{-1}\boldsymbol{H}\mathbf{1}\mathbf{1}' + n_u^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{H}\boldsymbol{J}_{n_i}. \tag{34}$$

By substituting $\boldsymbol{\Gamma}$ with equation (2), $\boldsymbol{H}$ with equation (34), and regrouping the terms in problem (29), we can majorize the log likelihood as

$$\begin{aligned} -\log\mathcal{L}(\boldsymbol{\Gamma}|\boldsymbol{Y}) &\leq \frac{1}{8}\|\boldsymbol{H} - \boldsymbol{\Gamma}\|_{\mathrm{F}}^2 + s \\ &= \frac{1}{8}\|\boldsymbol{H} - (\boldsymbol{\alpha}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}' + \boldsymbol{C}\boldsymbol{D}')\|_{\mathrm{F}}^2 + s \\ &= \frac{1}{8}\left\|(\boldsymbol{J}_{n_u}\boldsymbol{H}\boldsymbol{J}_{n_i} + n_i^{-1}\boldsymbol{H}\mathbf{1}\mathbf{1}' + n_u^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{H}\boldsymbol{J}_{n_i}) - (\boldsymbol{\alpha}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}' + \mathbf{C}\mathbf{D}')\right\|_{\mathrm{F}}^2 + s \\ &= \frac{1}{8}\left\|(n_i^{-1}\boldsymbol{H}\mathbf{1}\mathbf{1}' - \boldsymbol{\alpha}\mathbf{1}') + (n_u^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{H}\boldsymbol{J}_{n_i} - \mathbf{1}\boldsymbol{\beta}') + (\boldsymbol{J}_{n_u}\boldsymbol{H}\boldsymbol{J}_{n_i} - \boldsymbol{C}\boldsymbol{D}')\right\|_{\mathrm{F}}^2 + s. \end{aligned} \tag{35}$$

Now we can see the importance of column centering $\boldsymbol{\beta}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ for identification purposes. We have $\boldsymbol{\beta}' = \boldsymbol{\beta}'\boldsymbol{J}_{n_i}$ and $\boldsymbol{C}\boldsymbol{D}' = \boldsymbol{J}_{n_u}\boldsymbol{C}\boldsymbol{D}'\boldsymbol{J}_{n_i}$, such that

$$\begin{aligned}
-\log \mathcal{L}(\boldsymbol{\Gamma}|\boldsymbol{Y}) &\leq \frac{1}{8}\left\|(n_i^{-1}\boldsymbol{H}\mathbf{1} - \boldsymbol{\alpha})\mathbf{1}' + \mathbf{1}(n_u^{-1}\mathbf{1}'\boldsymbol{H} - \boldsymbol{\beta}')\boldsymbol{J}_{n_i} + (\boldsymbol{J}_{n_u}(\boldsymbol{H} - \boldsymbol{C}\boldsymbol{D}')\boldsymbol{J}_{n_i})\right\|_{\mathrm{F}}^2 + s \\
&= \frac{1}{8}\left(\left\|(n_i^{-1}\boldsymbol{H}\mathbf{1} - \boldsymbol{\alpha})\mathbf{1}'\right\|_{\mathrm{F}}^2 + \left\|\mathbf{1}(n_u^{-1}\mathbf{1}'\boldsymbol{H} - \boldsymbol{\beta}')\boldsymbol{J}_{n_i}\right\|_{\mathrm{F}}^2 + \|\boldsymbol{J}_{n_u}\left(\boldsymbol{H} - \boldsymbol{C}\boldsymbol{D}'\right)\boldsymbol{J}_{n_i}\|_{\mathrm{F}}^2\right) \\
&\quad + \frac{1}{4}\operatorname{tr}\left(\mathbf{1}\left(n_i^{-1}\boldsymbol{H}\mathbf{1} - \boldsymbol{\alpha}\right)'\mathbf{1}\left(n_u^{-1}\mathbf{1}'\boldsymbol{H} - \boldsymbol{\beta}'\right)\boldsymbol{J}_{n_i}\right) \\
&\quad + \frac{1}{4}\operatorname{tr}\left(\mathbf{1}\left(n_i^{-1}\boldsymbol{H}\mathbf{1} - \boldsymbol{\alpha}\right)'\boldsymbol{J}_{n_u}\left(\boldsymbol{H} - \boldsymbol{C}\boldsymbol{D}'\right)\boldsymbol{J}_{n_i}\right) \\
&\quad + \frac{1}{4}\operatorname{tr}\left(\boldsymbol{J}_{n_i}\left(n_u^{-1}\mathbf{1}'\boldsymbol{H} - \boldsymbol{\beta}'\right)'\mathbf{1}'\boldsymbol{J}_{n_u}\left(\boldsymbol{H} - \boldsymbol{C}\boldsymbol{D}'\right)\boldsymbol{J}_{n_i}\right) + s. \qquad (36)
\end{aligned}$$

Using the cyclic property of the trace and $\mathbf{1}'\boldsymbol{J}_{n_u} = \mathbf{0}$ and $\boldsymbol{J}_{n_i}\mathbf{1} = \mathbf{0}$, all traces are equal to zero, this leaves

$$\begin{aligned}
-\log \mathcal{L}(\boldsymbol{\Gamma}|\boldsymbol{Y}) &\leq \frac{1}{8}\left(\left\|\left(n_i^{-1}\boldsymbol{H}\mathbf{1} - \boldsymbol{\alpha}\right)\mathbf{1}'\right\|_{\mathrm{F}}^2 + \left\|\mathbf{1}\left(n_u^{-1}\mathbf{1}'\boldsymbol{H} - \boldsymbol{\beta}'\right)\boldsymbol{J}_{n_i}\right\|_{\mathrm{F}}^2 + \|\boldsymbol{J}_{n_u}\left(\boldsymbol{H} - \boldsymbol{C}\boldsymbol{D}'\right)\boldsymbol{J}_{n_i}\|_{\mathrm{F}}^2\right) \\
&\quad + s. \qquad (37)
\end{aligned}$$

This formulation allows to split the problems into three separate minimization problems. The majorization function in (37) is minimized with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ when

$$\boldsymbol{\alpha} = n_i^{-1}\boldsymbol{H}\mathbf{1}, \qquad (38)$$

$$\boldsymbol{\beta}' = n_u^{-1}\mathbf{1}'\boldsymbol{H}\boldsymbol{J}_{n_i}, \qquad (39)$$

as both $\|\left(n_i^{-1}\boldsymbol{H}\mathbf{1} - \boldsymbol{\alpha}\right)\mathbf{1}'\|_{\mathrm{F}}^2$ and $\|\mathbf{1}\left(n_u^{-1}\mathbf{1}'\boldsymbol{H} - \boldsymbol{\beta}'\right)\boldsymbol{J}_{n_i}\|_{\mathrm{F}}^2$ will then be zero. Next, estimates for $\boldsymbol{C}$ and $\boldsymbol{D}$ should be obtained to minimize the remainder of the majorization function. We minimize

$$\min_{\boldsymbol{C},\boldsymbol{D}} \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{C}\boldsymbol{D}'\|_{\mathrm{F}}^2 \qquad (40)$$

where $\widetilde{\boldsymbol{H}} = \boldsymbol{J}_{n_u}\boldsymbol{H}\boldsymbol{J}_{n_i}$.

Groenen et al. (2003) highlights the importance of imposing proper restrictions on γ_{ui}. Without these restrictions, the majorized solution may converge to the trivial solution of maximizing the log likelihood in equation (3): $\gamma_{ui} = \infty$ for $y_{ui} = 1$ and $\gamma_{ui} = -\infty$ for $y_{ui} = 0$. This should be avoided as it could lead to overfitting and would reduce the predictive power of our model. Therefore de Bruin et al. (2020) implements the nuclear norm, introduced in Section 3.2, to restrict parameter estimation of $\boldsymbol{C}$ and $\boldsymbol{D}$. As shown in the aforementioned section, nuclear norm regularization is equal to minimizing

$$\min_{\boldsymbol{C},\boldsymbol{D}} \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{C}\boldsymbol{D}'\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|\boldsymbol{C}\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|\boldsymbol{D}\|_{\mathrm{F}}^2, \qquad (41)$$

To minimize (41), the Rank-Restricted Soft SVD algorithm from Hastie et al. (2015) is used. This algorithm is optimized with respect to the sparse + low rank formulation of $\widetilde{\boldsymbol{H}}$. The algorithm alternates between fixing $\boldsymbol{C}$ and minimizing (41) with respect to $\boldsymbol{D}$ using a ridge regression, and vice versa. After each ridge regression, a low-rank approximation is obtained using singular value decomposition. This process is iterated until the convergence of $\hat{\boldsymbol{C}}\hat{\boldsymbol{D}}'$. More formally, the algorithm is given in Algorithm 1.

Algorithm 1 Rank-Restricted Soft SVD for finding $\boldsymbol{C}$ and $\boldsymbol{D}$

```
Initialize
    $\widetilde{\boldsymbol{H}} \leftarrow \boldsymbol{J}_{n_u}\boldsymbol{H}\boldsymbol{J}_{n_i}$
    $\boldsymbol{U} \leftarrow$ an $n_u \times f$ random matrix with orthonormal columns
    $\boldsymbol{Z} \leftarrow \boldsymbol{I}_f$
    $\boldsymbol{C} \leftarrow \boldsymbol{U}\boldsymbol{Z}$
repeat
    Given $\boldsymbol{C}$, solve for $\boldsymbol{D}$:
        $$\hat{\boldsymbol{D}} = \arg\min_{\boldsymbol{D}} \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{C}\boldsymbol{D}'\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|\boldsymbol{D}\|_{\mathrm{F}}^2 = \widetilde{\boldsymbol{H}}'\boldsymbol{U}\boldsymbol{Z}(\boldsymbol{Z}^2 + 4\lambda\boldsymbol{I})^{-1}$$
    Compute the SVD of $\hat{\boldsymbol{D}}\boldsymbol{Z} = \hat{\boldsymbol{V}}\hat{\boldsymbol{Z}}^2\boldsymbol{R}'$, and let $\boldsymbol{V} \leftarrow \hat{\boldsymbol{V}}$, $\boldsymbol{Z} \leftarrow \hat{\boldsymbol{Z}}$, and $\boldsymbol{D} \leftarrow \boldsymbol{V}\boldsymbol{Z}$
    Given $\boldsymbol{D}$, solve for $\boldsymbol{C}$:
        $$\hat{\boldsymbol{C}} = \arg\min_{\boldsymbol{C}} \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{C}\boldsymbol{D}'\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|\boldsymbol{C}\|_{\mathrm{F}}^2 = \widetilde{\boldsymbol{H}}\boldsymbol{V}\boldsymbol{Z}(\boldsymbol{Z}^2 + 4\lambda\boldsymbol{I})^{-1}$$
    Compute the SVD of $\hat{\boldsymbol{C}}\boldsymbol{Z} = \hat{\boldsymbol{U}}\hat{\boldsymbol{Z}}^2\boldsymbol{R}'$, and let $\boldsymbol{U} \leftarrow \hat{\boldsymbol{U}}$, $\boldsymbol{Z} \leftarrow \hat{\boldsymbol{Z}}$, and $\boldsymbol{C} \leftarrow \boldsymbol{U}\boldsymbol{Z}$
until $\boldsymbol{C}\boldsymbol{D}'$ has converged
Compute the SVD of $\widetilde{\boldsymbol{H}}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{Z}_\sigma\boldsymbol{R}'$, and let $\boldsymbol{S}_\lambda(\boldsymbol{Z}_\sigma) \leftarrow \mathrm{diag}[\max(\sigma_1 - \lambda, 0), \ldots, \max(\sigma_f - \lambda, 0)]$,
and $\boldsymbol{V} \leftarrow \boldsymbol{V}\boldsymbol{R}$
return $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{S}_\lambda(\boldsymbol{Z}_\sigma)^{1/2}$ and $\boldsymbol{D} = \boldsymbol{V}\boldsymbol{S}_\lambda(\boldsymbol{Z}_\sigma)^{1/2}$
```

Using the estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ from equations (38) and (39) and $\boldsymbol{C}$ and $\boldsymbol{D}$ from Algorithm 1, $\boldsymbol{H}$ can be re-estimated, which can then be used to find new estimates of the parameters of $\boldsymbol{\Gamma}$. These steps are iterated until the negative log likelihood in equation (15) has converged. The complete majorization algorithm is given below.

Algorithm 2 `Unrestricted`: Complete unrestricted minimization by majorization algorithm

```
initialize $\boldsymbol{\alpha}^{(0)}$, $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{C}^{(0)}$, $\boldsymbol{D}^{(0)}$
while $t = 0$ or $\left(\log L(\boldsymbol{\Gamma}^{(t-1)}|\boldsymbol{Y})\right)^{-1}(\log L(\boldsymbol{\Gamma}^{(t)}|\boldsymbol{Y}) - \log L(\boldsymbol{\Gamma}^{(t-1)}|\boldsymbol{Y})) \geq \epsilon$ do
    $t \leftarrow t + 1$
    Update $\boldsymbol{H}$: $h_{ui}^{(t)} \leftarrow 8b_{ui}^{(t-1)}$
    Update $\boldsymbol{\alpha}$: $\boldsymbol{\alpha}^{(t)} \leftarrow n_i^{-1}\boldsymbol{H}^{(t)}\mathbf{1}$
    Update $\boldsymbol{\beta}$: $\boldsymbol{\beta}^{(t)\prime} \leftarrow n_u^{-1}\mathbf{1}'\boldsymbol{H}^{(t)}\boldsymbol{J}_{n_i}$
    Update $\boldsymbol{C}^{(t)}$ and $\boldsymbol{D}^{(t)}$ using Algorithm 1
    Update $\gamma_{ui}$: $\gamma_{ui}^{(t)} \leftarrow \alpha_u^{(t)} + \beta_i^{(t)} + \boldsymbol{c}_u^{(t)\prime}\boldsymbol{d}_i^{(t)} \; \forall \, (u,i) \in \Psi$
    Compute $-\log L(\boldsymbol{\Gamma}^{(t)}|\boldsymbol{Y})$
end while
return $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{\beta}^{(t)}$, $\boldsymbol{C}^{(t)}$, $\boldsymbol{D}^{(t)}$
```

3.4 Restricted minimization by majorization

Our interest lies in incorporating available additional information into the method in the previous section. Consider the scenario in which we have additional information on both the users and the items. Define matrix $\boldsymbol{X}$ as the $n_u \times d_x$ matrix with n_u users and d_x variables. $\boldsymbol{G}$ is the $n_i \times d_g$ matrix with n_i items and d_g variables. Without loss of generality, we assume that both $\boldsymbol{X}$ and $\boldsymbol{G}$ are column centered. In Section 3.1, the bi-additive model is formulated as

$$\boldsymbol{\Gamma} = \boldsymbol{\alpha}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}' + \boldsymbol{C}\boldsymbol{D}'. \tag{42}$$

We now restrict the bi-additive interaction effect to be a linear combination of the $\boldsymbol{X}$ and $\boldsymbol{G}$, such that

$$\boldsymbol{\Gamma} = \boldsymbol{\alpha}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}' + \boldsymbol{X}\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}', \tag{43}$$

where $\boldsymbol{C}_r$ is the $d_x \times f$ weight matrix for the user characteristics and $\boldsymbol{D}_r$ is the $d_g \times f$ item characteristics weight matrix, where f is the specified number factors. Notice that $\boldsymbol{C} = \boldsymbol{X}\boldsymbol{C}_r$ and $\boldsymbol{D} = \boldsymbol{G}\boldsymbol{D}_r$, where both $\boldsymbol{X}$ and $\boldsymbol{G}$ are given. If we set $\boldsymbol{X} = \boldsymbol{I}_u$ and $\boldsymbol{G} = \boldsymbol{I}_i$, where $\boldsymbol{I}_u$ and $\boldsymbol{I}_i$ are identity matrices of rank n_u and n_i respectively, we have equality of both $\boldsymbol{C} = \boldsymbol{C}_r$ and $\boldsymbol{D} = \boldsymbol{D}_r$ leaving us with the unrestricted problem in (42).

Changing the bi-additive modelling of $\boldsymbol{\Gamma}$ does not alter the steps involved in majorizing the likelihood function (4). In fact, we can majorize the likelihood function in exactly the same way as described in Section 3.3.1, yielding the majorized least squares problem:

$$\begin{aligned} -\log \mathcal{L}(\boldsymbol{\Gamma}|\boldsymbol{Y}) &\leq \frac{1}{8}\sum_{(u,i)\in\Psi}(h_{ui} - \gamma_{ui})^2 + s \\ &= \frac{1}{8}\|\boldsymbol{H} - \boldsymbol{\Gamma}\|_{\mathrm{F}}^2 + s, \end{aligned} \tag{44}$$

where $\boldsymbol{\Gamma} = \boldsymbol{\alpha}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}' + \boldsymbol{X}\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'$, s is a constant not dependent on $\boldsymbol{\Gamma}$ and Ψ is the set containing all observed user item pairs. We are again able to formulate $\boldsymbol{H}$ using a sparse plus low-rank structure as

$$\begin{aligned}\boldsymbol{H} &= \boldsymbol{Z}_s + \boldsymbol{\Gamma}^{(0)} \\ &= (\boldsymbol{Z}_s) + \left(\boldsymbol{\alpha}^{(0)}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}'^{(0)} + \boldsymbol{X}\boldsymbol{C}_r^{(0)}\boldsymbol{D}_r'^{(0)}\boldsymbol{G}'\right) \\ &= \text{sparse} + \text{low-rank},\end{aligned} \tag{45}$$

where $\boldsymbol{Z}_s$ is defined as in (31). To obtain updates we once again need to column center $\boldsymbol{\beta}$. Once column centered we can derive, similar to Section 3.3.2

$$\begin{aligned}-\log\mathcal{L}(\boldsymbol{\Gamma}|\boldsymbol{Y}) &\leq \frac{1}{8}\|\boldsymbol{H} - \boldsymbol{\Gamma}\|_{\mathrm{F}}^2 + s \\ &= \frac{1}{8}\left\|\boldsymbol{H} - (\boldsymbol{\alpha}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}' + \boldsymbol{X}\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}')\right\|_{\mathrm{F}}^2 + s \\ &= \frac{1}{8}\left\|(\boldsymbol{J}_{n_u}\boldsymbol{H}\boldsymbol{J}_{n_i} + n_i^{-1}\boldsymbol{H}\mathbf{1}\mathbf{1}' + n_u^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{H}\boldsymbol{J}_{n_i}) - (\boldsymbol{\alpha}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}' + \boldsymbol{X}\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}')\right\|_{\mathrm{F}}^2 + s \\ &= \frac{1}{8}\left\|(n_i^{-1}\boldsymbol{H}\mathbf{1}\mathbf{1}' - \boldsymbol{\alpha}\mathbf{1}') + (n_u^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{H}\boldsymbol{J}_{n_i} - \mathbf{1}\boldsymbol{\beta}') + (\boldsymbol{J}_{n_u}\boldsymbol{H}\boldsymbol{J}_{n_i} - \boldsymbol{X}\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}')\right\|_{\mathrm{F}}^2 + s \\ &= \frac{1}{8}\left(\left\|\left(n_i^{-1}\boldsymbol{H}\mathbf{1} - \boldsymbol{\alpha}\right)\mathbf{1}'\right\|_{\mathrm{F}}^2 + \left\|\mathbf{1}\left(n_u^{-1}\mathbf{1}'\boldsymbol{H} - \boldsymbol{\beta}'\right)\boldsymbol{J}_{n_i}\right\|_{\mathrm{F}}^2 + \|\boldsymbol{J}_{n_u}\left(\boldsymbol{H} - \boldsymbol{X}\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'\right)\boldsymbol{J}_{n_i}\|_{\mathrm{F}}^2\right) \\ &\quad + s,\end{aligned} \tag{46}$$

using the cyclic property of the trace and centering matrix property that $\mathbf{1}'\boldsymbol{J}_{n_u} = \mathbf{0}$ and $\boldsymbol{J}_{n_i}\mathbf{1} = \mathbf{0}$. This minimization problem can be split into three different minimization problems. The first two of which are minimized with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively, when

$$\boldsymbol{\alpha} = n_i^{-1}\boldsymbol{H}\mathbf{1}, \tag{47}$$

$$\boldsymbol{\beta}' = n_u^{-1}\mathbf{1}'\boldsymbol{H}\boldsymbol{J}_{n_i}, \tag{48}$$

as both $\|\left(n_i^{-1}\boldsymbol{H}\mathbf{1} - \boldsymbol{\alpha}\right)\mathbf{1}'\|_{\mathrm{F}}^2$ and $\|\mathbf{1}\left(n_u^{-1}\mathbf{1}'\boldsymbol{H} - \boldsymbol{\beta}'\right)\boldsymbol{J}_{n_i}\|_{\mathrm{F}}^2$ will then be zero. This leaves us with one final minimization problem

$$\frac{1}{8}\|\boldsymbol{J}_{n_u}\left(\boldsymbol{H} - \boldsymbol{X}\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'\right)\boldsymbol{J}_{n_i}\|_{\mathrm{F}}^2 = \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{X}\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'\|_{\mathrm{F}}^2, \tag{49}$$

3.4.1 Regularization

Rank restricted parameter regularization of $\boldsymbol{C}_r$ and $\boldsymbol{D}_r$ is required to avoid the overfitting problem as discussed in Groenen et al. (2003). To introduce the nuclear norm in (49) we need to introduce the concept of the Generalised Singular Value Decomposition (GSVD). In a GSVD, two positive definite square matrices $\boldsymbol{M}$ and $\boldsymbol{W}$ are used to express constraints imposed on the rows and columns of a given matrix $\boldsymbol{A}$ (Abdi, 2007). The matrix $\boldsymbol{A}$ is now decomposed as

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{\Phi}\boldsymbol{Q}', \tag{50}$$

with $\boldsymbol{P}'\boldsymbol{M}\boldsymbol{P} = \boldsymbol{Q}'\boldsymbol{W}\boldsymbol{Q} = \boldsymbol{I}$. Thus, the generalised singular vectors are orthogonal under the constraints imposed by both $\boldsymbol{M}$ and $\boldsymbol{W}$ (Abdi, 2007). To obtain $\boldsymbol{P}$ and $\boldsymbol{Q}$ we need to define a new matrix

$$\widetilde{\boldsymbol{A}} = \boldsymbol{M}^{\frac{1}{2}}\boldsymbol{A}\boldsymbol{W}^{\frac{1}{2}}. \tag{51}$$

Computing the standard Singular Value Decomposition of $\widetilde{\boldsymbol{A}}$ as

$$\widetilde{\boldsymbol{A}} = \widetilde{\boldsymbol{P}}\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{Q}}', \tag{52}$$

with $\widetilde{\boldsymbol{P}}'\widetilde{\boldsymbol{P}} = \widetilde{\boldsymbol{Q}}'\widetilde{\boldsymbol{Q}} = \boldsymbol{I}$. The generalised singular vectors are then calculated as

$$\boldsymbol{P} = \boldsymbol{M}^{-\frac{1}{2}}\widetilde{\boldsymbol{P}} \quad \text{and} \quad \boldsymbol{Q} = \boldsymbol{W}^{-\frac{1}{2}}\widetilde{\boldsymbol{Q}}. \tag{53}$$

The matrix of singular values of $\boldsymbol{A}$ is simply equal to the matrix of singular values $\widetilde{\boldsymbol{A}}$, or similarly $\boldsymbol{\Phi} = \widetilde{\boldsymbol{\Phi}}$. Proof of this statement follows by substitution and can be found in Abdi (2007). In our example, row constraints are set to $\boldsymbol{M} = (\boldsymbol{X}'\boldsymbol{X})$ and column constraints are set to $\boldsymbol{W} = (\boldsymbol{G}'\boldsymbol{G})$, which are positive definite matrices if all columns of $\boldsymbol{X}$ and $\boldsymbol{G}$ are linearly independent. We decompose matrix $\boldsymbol{B} = \boldsymbol{C}_r\boldsymbol{D}_r'$ by the GSVD $\boldsymbol{B} = \boldsymbol{C}_r\boldsymbol{D}_r' = \boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{V}'$, with $\boldsymbol{C}_r = \boldsymbol{U}\boldsymbol{\Phi}^{\frac{1}{2}}$ and $\boldsymbol{D}_r = \boldsymbol{V}\boldsymbol{\Phi}^{\frac{1}{2}}$. Now we can introduce a weighted nuclear norm in (49), similar to the introduction of a nuclear norm in (9), as

$$\begin{aligned}
&\min_{\boldsymbol{C}_r,\boldsymbol{D}_r} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{X}\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\left(\|\boldsymbol{X}\boldsymbol{C}_r\|_{\mathrm{F}}^2 + \|\boldsymbol{G}\boldsymbol{D}_r\|_{\mathrm{F}}^2\right) \\
&\quad = \min_{\boldsymbol{B}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{X}\boldsymbol{B}\boldsymbol{G}'\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\left(\|\boldsymbol{X}\boldsymbol{U}\boldsymbol{\Phi}^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\boldsymbol{G}\boldsymbol{V}\boldsymbol{\Phi}^{\frac{1}{2}}\|_{\mathrm{F}}^2\right) \\
&\quad = \min_{\boldsymbol{B}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{X}\boldsymbol{B}\boldsymbol{G}'\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\operatorname{tr}(\boldsymbol{\Phi}^{\frac{1}{2}}\boldsymbol{U}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{U}\boldsymbol{\Phi}^{\frac{1}{2}}) + \frac{\lambda}{2}\operatorname{tr}(\boldsymbol{\Phi}^{\frac{1}{2}}\boldsymbol{V}'\boldsymbol{G}'\boldsymbol{G}\boldsymbol{V}\boldsymbol{\Phi}^{\frac{1}{2}}) \\
&\quad = \min_{\boldsymbol{B}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{X}\boldsymbol{B}\boldsymbol{G}'\|_{\mathrm{F}}^2 + \lambda\operatorname{tr}(\boldsymbol{\Phi}) && (54) \\
&\quad = \min_{\boldsymbol{B}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{X}\boldsymbol{B}\boldsymbol{G}'\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_*. && (55)
\end{aligned}$$

We minimize (55) with a different approach compared to the Rank-Restricted Soft SVD algorithm in Section 3.3.2. We rewrite (54) using the cyclic property of the trace and the properties of the GSVD

$\boldsymbol{U}'(\boldsymbol{X}'\boldsymbol{X})\boldsymbol{U} = \boldsymbol{V}'(\boldsymbol{G}'\boldsymbol{G})\boldsymbol{V} = \boldsymbol{I}$ as

$$\begin{aligned}
&\min_{\boldsymbol{B}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{X}\boldsymbol{B}\boldsymbol{G}'\|_{\mathrm{F}}^2 + \lambda\,\mathrm{tr}\,(\boldsymbol{\Phi}) \\
&= \min_{\boldsymbol{U},\boldsymbol{\Phi},\boldsymbol{V}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}} - \boldsymbol{X}\boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{V}'\boldsymbol{G}'\|_{\mathrm{F}}^2 + \lambda\,\mathrm{tr}\,(\boldsymbol{\Phi}) \\
&= \min_{\boldsymbol{U},\boldsymbol{\Phi},\boldsymbol{V}} \quad \frac{1}{8}\left(\|\widetilde{\boldsymbol{H}}\|_{\mathrm{F}}^2 + \|\boldsymbol{X}\boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{V}'\boldsymbol{G}'\|_{\mathrm{F}}^2\right) - \frac{1}{4}\,\mathrm{tr}\,(\widetilde{\boldsymbol{H}}'\boldsymbol{X}\boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{V}'\boldsymbol{G}') + \lambda\,\mathrm{tr}\,(\boldsymbol{\Phi}) \\
&= \min_{\boldsymbol{U},\boldsymbol{\Phi},\boldsymbol{V}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}}\|_{\mathrm{F}}^2 + \frac{1}{8}\,\mathrm{tr}\,(\boldsymbol{G}\boldsymbol{V}\boldsymbol{\Phi}\boldsymbol{U}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{V}'\boldsymbol{G}') - \frac{1}{4}\,\mathrm{tr}\,(\widetilde{\boldsymbol{H}}'\boldsymbol{X}\boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{V}'\boldsymbol{G}') + \lambda\,\mathrm{tr}\,(\boldsymbol{\Phi}) \\
&= \min_{\boldsymbol{U},\boldsymbol{\Phi},\boldsymbol{V}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}}\|_{\mathrm{F}}^2 + \frac{1}{8}\,\mathrm{tr}\,(\boldsymbol{\Phi}\boldsymbol{V}'\boldsymbol{G}'\boldsymbol{G}\boldsymbol{V}\boldsymbol{\Phi}) - \frac{1}{4}\,\mathrm{tr}\,(\widetilde{\boldsymbol{H}}'\boldsymbol{X}\boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{V}'\boldsymbol{G}') + \lambda\,\mathrm{tr}\,(\boldsymbol{\Phi}) \\
&= \min_{\boldsymbol{U},\boldsymbol{\Phi},\boldsymbol{V}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}}\|_{\mathrm{F}}^2 + \frac{1}{8}\|\boldsymbol{\Phi}\|_{\mathrm{F}}^2 - \frac{1}{4}\,\mathrm{tr}\,(\widetilde{\boldsymbol{H}}'\boldsymbol{X}\boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{V}'\boldsymbol{G}') + \lambda\,\mathrm{tr}\,(\boldsymbol{\Phi}). \qquad (56)
\end{aligned}$$

We can find $\boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{V}'$ by (51)-(53). We substitute $\boldsymbol{B} = \boldsymbol{C}_r\boldsymbol{D}_r' = \boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{V}'$ for $\widetilde{\boldsymbol{B}} = (\boldsymbol{X}'\boldsymbol{X})^{\frac{1}{2}}\boldsymbol{B}(\boldsymbol{G}'\boldsymbol{G})^{\frac{1}{2}}$ in (56), which we decompose by the SVD $\widetilde{\boldsymbol{B}} = (\boldsymbol{X}'\boldsymbol{X})^{\frac{1}{2}}\boldsymbol{B}(\boldsymbol{G}'\boldsymbol{G})^{\frac{1}{2}} = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'$. As the matrix of singular values of $\boldsymbol{B}$ is equal to the matrix of singular values of $\widetilde{\boldsymbol{B}}$ we substitute $\boldsymbol{\Phi}$ by $\widetilde{\boldsymbol{\Phi}}$, this yields

$$\min_{\widetilde{\boldsymbol{U}},\widetilde{\boldsymbol{\Phi}},\widetilde{\boldsymbol{V}}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}}\|_{\mathrm{F}}^2 + \frac{1}{8}\|\widetilde{\boldsymbol{\Phi}}\|_{\mathrm{F}}^2 - \frac{1}{4}\,\mathrm{tr}\left(\widetilde{\boldsymbol{H}}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-\frac{1}{2}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'(\boldsymbol{G}'\boldsymbol{G})^{-\frac{1}{2}}\boldsymbol{G}'\right) + \lambda\,\mathrm{tr}\,(\widetilde{\boldsymbol{\Phi}}). \qquad (57)$$

In the remainder of this section we will substitute $\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-\frac{1}{2}} = \widetilde{\boldsymbol{X}}$ and $(\boldsymbol{G}'\boldsymbol{G})^{-\frac{1}{2}}\boldsymbol{G}' = \widetilde{\boldsymbol{G}}'$ to ease notation. We minimize (57) in an iterative procedure where we fix two of the three matrices $\widetilde{\boldsymbol{U}}$, $\widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{V}}$ and minimize (57) for the remaining matrix. Minimizing with respect to $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{V}}$ requires the introduction of a lower bound inequality derived by Kristof (1970). This inequality states that if $\boldsymbol{L}$ is a diagonal matrix with non negative entries, and $\boldsymbol{N}$ is orthogonal then it holds that

$$-\,\mathrm{tr}\,(\boldsymbol{N}\boldsymbol{L}) \geq -\,\mathrm{tr}\,(\boldsymbol{L}), \qquad (58)$$

with equality if and only if $\boldsymbol{N} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. We will use this inequality to obtain updates for $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{V}}$ similar to Borg and Groenen (2005) in their chapter about the Orthogonal Procrustean Problem. Minimizing with respect to $\widetilde{\boldsymbol{U}}$ we rewrite (57) as

$$\begin{aligned}
&\min_{\widetilde{\boldsymbol{U}}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}}\|_{\mathrm{F}}^2 - \frac{1}{4}\,\mathrm{tr}\left(\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\right) + \frac{1}{8}\|\widetilde{\boldsymbol{\Phi}}\|_{\mathrm{F}}^2 + \lambda\,\mathrm{tr}\,(\widetilde{\boldsymbol{\Phi}}) \\
&= \min_{\widetilde{\boldsymbol{U}}} \quad c - \frac{1}{4}\,\mathrm{tr}\left(\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\right) \\
&= \min_{\widetilde{\boldsymbol{U}}} \quad c - \frac{1}{4}\,\mathrm{tr}\,(\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}). \qquad (59)
\end{aligned}$$

Decomposing $\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}'}\widetilde{\boldsymbol{X}}$ by the SVD $\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}'}\widetilde{\boldsymbol{X}} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{Q}'$, we can rewrite (59) as

$$\begin{aligned}
&\min_{\widetilde{\boldsymbol{U}}} \quad c - \frac{1}{4}\,\mathrm{tr}\,(\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}'}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}) \\
&\qquad = \min_{\widetilde{\boldsymbol{U}}} \quad c - \frac{1}{4}\,\mathrm{tr}\,(\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{Q}'\widetilde{\boldsymbol{U}}) \\
&\qquad = \min_{\widetilde{\boldsymbol{U}}} \quad c - \frac{1}{4}\,\mathrm{tr}\,(\boldsymbol{Q}'\widetilde{\boldsymbol{U}}\boldsymbol{P}\boldsymbol{\Lambda}) \\
&\qquad \geq \min_{\widetilde{\boldsymbol{U}}} \quad c - \frac{1}{4}\,\mathrm{tr}\,(\boldsymbol{\Lambda}),
\end{aligned} \tag{60}$$

where c captures all constant terms. Since $\widetilde{\boldsymbol{U}}$ is orthogonal, so is $\boldsymbol{Q}'\widetilde{\boldsymbol{U}}\boldsymbol{P}$. The minimization problem is now in the form of (58), with $\boldsymbol{N} = \boldsymbol{Q}'\widetilde{\boldsymbol{U}}\boldsymbol{P}$ and $\boldsymbol{L} = \boldsymbol{\Lambda}$. From the Kristof inequality, we know that (60) is minimal if and only if $\boldsymbol{N} = \boldsymbol{Q}'\widetilde{\boldsymbol{U}}\boldsymbol{P} = \boldsymbol{I}$. Hence, we should set $\widetilde{\boldsymbol{U}} = \boldsymbol{Q}\boldsymbol{P}'$ such that $\boldsymbol{Q}'\boldsymbol{Q}\boldsymbol{P}'\boldsymbol{P} = \boldsymbol{I}$ due to the properties of the SVD. Due to the size of the involved matrices calculating $\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}'}\widetilde{\boldsymbol{X}}$ is rather expensive. To ease computation we derive

$$\begin{aligned}
&\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}'}\widetilde{\boldsymbol{X}} \\
&\quad = \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\big(\boldsymbol{J}_{n_u}\left(\boldsymbol{Z}_s + \boldsymbol{\alpha}^{(0)}\boldsymbol{1}' + \boldsymbol{1}\boldsymbol{\beta}'^{(0)} + \boldsymbol{X}\boldsymbol{C}_r^{(0)}\boldsymbol{D}_r'{}^{(0)}\boldsymbol{G}'\right)\boldsymbol{J}_{n_i}\big)'\widetilde{\boldsymbol{X}} \\
&\quad = \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\left(\boldsymbol{Z}_s + \boldsymbol{\alpha}^{(0)}\boldsymbol{1}' + \boldsymbol{1}\boldsymbol{\beta}'^{(0)} + \boldsymbol{X}\boldsymbol{C}_r^{(0)}\boldsymbol{D}_r'{}^{(0)}\boldsymbol{G}'\right)'\widetilde{\boldsymbol{X}} \\
&\quad = \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\boldsymbol{Z}_s'\widetilde{\boldsymbol{X}} + \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\boldsymbol{1}\boldsymbol{\alpha}'^{(0)}\widetilde{\boldsymbol{X}} \\
&\qquad + \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\boldsymbol{\beta}^{(0)}\boldsymbol{1}'\widetilde{\boldsymbol{X}} + \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\boldsymbol{G}\boldsymbol{D}_r^{(0)}\boldsymbol{C}_r'{}^{(0)}\boldsymbol{X}'\widetilde{\boldsymbol{X}} \\
&\quad = \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\boldsymbol{Z}_s'\widetilde{\boldsymbol{X}} + \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'(\boldsymbol{G}'\boldsymbol{G})^{\frac{1}{2}}\boldsymbol{D}_r^{(0)}\boldsymbol{C}_r'{}^{(0)}(\boldsymbol{X}'\boldsymbol{X})^{\frac{1}{2}},
\end{aligned} \tag{61}$$

where $\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\boldsymbol{1}\boldsymbol{\alpha}'^{(0)}\widetilde{\boldsymbol{X}}$ and $\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\boldsymbol{\beta}^{(0)}\boldsymbol{1}'\widetilde{\boldsymbol{X}}$ are both zero since $(\boldsymbol{G}'\boldsymbol{1}) = (\boldsymbol{1}'\boldsymbol{X}) = \boldsymbol{0}$ as both $\boldsymbol{X}$ and $\boldsymbol{G}$ are column centered. To further increase the efficiency of the algorithm, we can calculate matrices which do not change over each iteration, such as $(\boldsymbol{X}'\boldsymbol{X})^{\frac{1}{2}}$ and $(\boldsymbol{G}'\boldsymbol{G})^{\frac{1}{2}}$ this yields

$$\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}'}\widetilde{\boldsymbol{X}} = \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'(\widetilde{\boldsymbol{G}}'\boldsymbol{Z}_s'\widetilde{\boldsymbol{X}}) + \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\left((\boldsymbol{G}'\boldsymbol{G})^{\frac{1}{2}}\boldsymbol{D}_r^{(0)}\boldsymbol{C}_r'{}^{(0)}(\boldsymbol{X}'\boldsymbol{X})^{\frac{1}{2}})\right). \tag{62}$$

Minimizing (57) with respect to $\widetilde{\boldsymbol{V}}$ is done in similar fashion. This yields the update

$$\widetilde{\boldsymbol{V}} = \bar{\boldsymbol{P}}\bar{\boldsymbol{Q}}', \tag{63}$$

where we decompose $\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}'}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}$ by the SVD $\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}'}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}} = \bar{\boldsymbol{P}}\bar{\boldsymbol{\Lambda}}\bar{\boldsymbol{Q}}'$. Again, for efficiency reasons, we rewrite $\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}'}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}$ similar to (62) as

$$\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}'}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}} = (\widetilde{\boldsymbol{G}}'\boldsymbol{Z}_s'\widetilde{\boldsymbol{X}})\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}} + \left((\boldsymbol{G}'\boldsymbol{G})^{\frac{1}{2}}\boldsymbol{D}_r^{(0)}\boldsymbol{C}_r'{}^{(0)}(\boldsymbol{X}'\boldsymbol{X})^{\frac{1}{2}})\right)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}. \tag{64}$$

To minimize (57) with respect to $\widetilde{\boldsymbol{\Phi}}$ we derive

$$\begin{aligned}
&\min_{\widetilde{\boldsymbol{\Phi}}} \quad \frac{1}{8}\|\widetilde{\boldsymbol{H}}\|_{\mathrm{F}}^2 - \frac{1}{4}\operatorname{tr}(\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}') + \frac{1}{8}\|\widetilde{\boldsymbol{\Phi}}\|_{\mathrm{F}}^2 + \lambda\operatorname{tr}(\widetilde{\boldsymbol{\Phi}}) \\
&= \min_{\widetilde{\boldsymbol{\Phi}}} \quad c - \frac{1}{4}\operatorname{tr}(\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}) + \frac{1}{8}\|\widetilde{\boldsymbol{\Phi}}\|_{\mathrm{F}}^2 + \lambda\operatorname{tr}(\widetilde{\boldsymbol{\Phi}}) \\
&= \min_{\widetilde{\boldsymbol{\Phi}}} \quad c - \frac{1}{4}\sum_{i=1}^{r}\widetilde{f}_{ii}\widetilde{\phi}_{ii} + \frac{1}{8}\sum_{i=1}^{r}\widetilde{\phi}_{ii}^2 + \lambda\sum_{i=1}^{r}\widetilde{\phi}_{ii} \\
&= \min_{\widetilde{\boldsymbol{\Phi}}} \quad c + \sum_{i=1}^{r}\left[\left(\lambda - \frac{1}{4}\widetilde{f}_{ii}\right)\widetilde{\phi}_{ii} + \frac{1}{8}\widetilde{\phi}_{ii}^2\right], \qquad (65)
\end{aligned}$$

where $\widetilde{\boldsymbol{F}} = \widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}$ and $r = \min\{(f_x + n_u - n_x), (d_g + n_i - n_g)\}$ for the $(f_x{+}n_u{-}n_x)\times(d_g{+}n_i{-}n_g)$ matrix $\boldsymbol{B}$. Minimization problem (65) can be minimized by minimizing r separate convex minimization problems. We derive for each ϕ_{ii}, $i \in \{1, ..., r\}$

$$\begin{aligned}
&\min_{\widetilde{\phi}_{ii}} \quad c + \sum_{i=1}^{r}\left[\left(\lambda - \frac{1}{4}\widetilde{f}_{ii}\right)\widetilde{\phi}_{ii} + \frac{1}{8}\widetilde{\phi}_{ii}^2\right] \\
&\Leftrightarrow \frac{d}{d\widetilde{\phi}_{ii}}\left(c + \sum_{i=1}^{r}\left[\left(\lambda - \frac{1}{4}\widetilde{f}_{ii}\right)\widetilde{\phi}_{ii} + \frac{1}{8}\widetilde{\phi}_{ii}^2\right]\right) = 0 \\
&\Leftrightarrow \lambda - \frac{1}{4}\widetilde{f}_{ii} + \frac{1}{4}\widetilde{\phi}_{ii} = 0 \\
&\Leftrightarrow \widetilde{\phi}_{ii} = \widetilde{f}_{ii} - 4\lambda. \qquad (66)
\end{aligned}$$

Following Section 3.2, each ϕ_{ii} is regularized as

$$\widetilde{\phi}_{ii} = \max(\widetilde{f}_{ii} - 4\lambda, 0). \qquad (67)$$

$\widetilde{\boldsymbol{F}} = \widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}$ can once again be more efficiently calculated, similar to (62) and (64), as

$$\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}} = \widetilde{\boldsymbol{V}}'(\widetilde{\boldsymbol{G}}'\boldsymbol{Z}_s'\widetilde{\boldsymbol{X}})\widetilde{\boldsymbol{U}} + \widetilde{\boldsymbol{V}}'\left((\boldsymbol{G}'\boldsymbol{G})^{\frac{1}{2}}\boldsymbol{D}_r^{(0)}\boldsymbol{C}_r'^{(0)}(\boldsymbol{X}'\boldsymbol{X})^{\frac{1}{2}})\right)\widetilde{\boldsymbol{U}}. \qquad (68)$$

Now we find $\boldsymbol{U}$ and $\boldsymbol{V}$ using (53) by setting $\boldsymbol{U} = (\boldsymbol{X}'\boldsymbol{X})^{-\frac{1}{2}}\widetilde{\boldsymbol{U}}$ and $\boldsymbol{V} = (\boldsymbol{G}'\boldsymbol{G})^{-\frac{1}{2}}\widetilde{\boldsymbol{V}}$. We also set $\boldsymbol{\Phi} = \widetilde{\boldsymbol{\Phi}}$. Using these parameters we derive updates for $\boldsymbol{C_r}$ and $\boldsymbol{D_r}$ as $\boldsymbol{C_r} = \boldsymbol{U}\boldsymbol{\Phi}^{\frac{1}{2}}$ and $\boldsymbol{D_r} = \boldsymbol{V}\boldsymbol{\Phi}^{\frac{1}{2}}$. We denote the process of obtaining parameter updates more formally in Algorithm 3.

Algorithm 3 Updating of $\boldsymbol{C}_r$ and $\boldsymbol{D}_r$

```
Initialize
    $\boldsymbol{Z}_s \leftarrow \boldsymbol{Z}_s$
    $\widetilde{\boldsymbol{V}} \leftarrow$ a $d_x \times r$ random matrix with orthonormal columns
    $\widetilde{\boldsymbol{\Phi}} \leftarrow \boldsymbol{I}_r$
    $\boldsymbol{X} \leftarrow \boldsymbol{X}$
    $\boldsymbol{G} \leftarrow \boldsymbol{G}$
repeat
    Given $\widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{V}}$, solve for $\widetilde{\boldsymbol{U}}$:
        Compute the SVD of $\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}} = \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'(\widetilde{\boldsymbol{G}}'\boldsymbol{Z}_s'\widetilde{\boldsymbol{X}}) + \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'\left((\boldsymbol{G}'\boldsymbol{G})^{\frac{1}{2}}\boldsymbol{D}_r^{(0)}\boldsymbol{C}_r'^{(0)}(\boldsymbol{X}'\boldsymbol{X})^{\frac{1}{2}})\right) = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{Q}'$,
        and let $\widetilde{\boldsymbol{U}} \leftarrow \boldsymbol{Q}\boldsymbol{P}'$
    Given $\widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{U}}$, solve for $\widetilde{\boldsymbol{V}}$
        Compute the SVD of $\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}} = (\widetilde{\boldsymbol{G}}'\boldsymbol{Z}_s'\widetilde{\boldsymbol{X}})\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}} + \left((\boldsymbol{G}'\boldsymbol{G})^{\frac{1}{2}}\boldsymbol{D}_r^{(0)}\boldsymbol{C}_r'^{(0)}(\boldsymbol{X}'\boldsymbol{X})^{\frac{1}{2}})\right)\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}} = \bar{\boldsymbol{P}}\bar{\boldsymbol{\Lambda}}\bar{\boldsymbol{Q}}'$,
        and let $\widetilde{\boldsymbol{V}} \leftarrow \bar{\boldsymbol{P}}\bar{\boldsymbol{Q}}'$
    Given $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{V}}$, solve for $\widetilde{\boldsymbol{\Phi}}$
        Compute $\widetilde{\boldsymbol{F}} = \widetilde{\boldsymbol{V}}'(\widetilde{\boldsymbol{G}}'\boldsymbol{Z}_s'\widetilde{\boldsymbol{X}})\widetilde{\boldsymbol{U}} + \widetilde{\boldsymbol{V}}'\left((\boldsymbol{G}'\boldsymbol{G})^{\frac{1}{2}}\boldsymbol{D}_r^{(0)}\boldsymbol{C}_r'^{(0)}(\boldsymbol{X}'\boldsymbol{X})^{\frac{1}{2}})\right)\widetilde{\boldsymbol{U}}$,
        set $\phi_{ii} \leftarrow \max(f_{ii} - 4\lambda, 0)\ \forall\ i \in \{1, ..., r\}$
until $\widetilde{\boldsymbol{U}}, \widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{V}}$ have converged
return $\boldsymbol{C}_r = (\boldsymbol{X}'\boldsymbol{X})^{-\frac{1}{2}}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}^{1/2}$ and $\boldsymbol{D}_r = (\boldsymbol{G}'\boldsymbol{G})^{-\frac{1}{2}}\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{\Phi}}^{1/2}$
```

Using the estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ from equations (47) and (48) and $\boldsymbol{C}_r$ and $\boldsymbol{D}$ from Algorithm 3, we can re-estimate $\boldsymbol{H}$ under a restricted solution, which can then be used to find new estimates of the parameters of $\boldsymbol{\Gamma}$. We iterate these steps until the negative log likelihood in equation (15) has converged. The complete restricted minimization by majorization algorithm is given below.

Algorithm 4 `Fully restricted`: Complete restricted minimization by majorization algorithm

```
initialize $\boldsymbol{\alpha}^{(0)}$, $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{C}_r^{(0)}$, $\boldsymbol{D}_r^{(0)}$
Column center $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{X}$, and $\boldsymbol{G}$
while $t = 0$ or $(\log L(\boldsymbol{\Gamma}^{(t)}|\boldsymbol{Y}) - \log L(\boldsymbol{\Gamma}^{(t-1)}|\boldsymbol{Y}))/\log L(\boldsymbol{\Gamma}^{(t-1)}|\boldsymbol{Y}) \geq \epsilon$ do
    $t \leftarrow t + 1$
    Update $\boldsymbol{H}$: $h_{ui}^{(t)} \leftarrow 8b_{ui}^{(t-1)}$
    Update $\boldsymbol{\alpha}$: $\boldsymbol{\alpha}^{(t)} \leftarrow n_i^{-1}\boldsymbol{H}^{(t)}\mathbf{1}$
    Update $\boldsymbol{\beta}$: $\boldsymbol{\beta}'^{(t)} \leftarrow n_u^{-1}\mathbf{1}'\boldsymbol{H}^{(t)}\boldsymbol{J}_{n_i}$
    Update $\boldsymbol{C}_r^{(t)}$ and $\boldsymbol{D}_r^{(t)}$ using Algorithm 3
    Update $\gamma_{ui}$: $\gamma_{ui}^{(t)} \leftarrow \alpha_u^{(t)} + \beta_i^{(t)} + \boldsymbol{x}_u'\boldsymbol{C}^{(t)}\boldsymbol{D}^{(t)\prime}\boldsymbol{g}_i \;\forall\, (u,i) \in \Psi$
    Compute $-\log L(\boldsymbol{\Gamma}^{(t)}|\boldsymbol{Y})$
end while
return $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{\beta}^{(t)}$, $\boldsymbol{C}_r^{(t)}$, $\boldsymbol{D}_r^{(t)}$
```

3.5 Partially restricted minimization by majorization

In the previous section we restricted all users and items. In practice, there could be a large part of the user and items for which we do not observe additional information. We should therefore also consider a model for which we only restrict the bi-additive interaction effect of a subset of users and items. This is achieved by reformatting the matrices $\boldsymbol{X}$ and $\boldsymbol{G}$ from Section 3.4 as

$$\boldsymbol{X} = \left[\begin{array}{c|c} \boldsymbol{X}^* & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{I}_x \end{array}\right] \text{ and } \boldsymbol{G} = \left[\begin{array}{c|c} \boldsymbol{G}^* & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{I}_g \end{array}\right]. \tag{69}$$

Where the $n_x \times d_x$ matrix $\boldsymbol{X}$ and the $n_g \times d_g$ matrix $\boldsymbol{G}$ correspond to extra information on the subset of users and items. The matrices $\boldsymbol{I}_x$ and $\boldsymbol{I}_g$ correspond to rank $n_u - n_x$ and $n_i - n_g$ identity matrices, respectively. As we restrict a subset of the bi-additive interaction effect to be a linear combination of the columns of $\boldsymbol{X}$ and $\boldsymbol{G}$, $\boldsymbol{C}_r$ and $\boldsymbol{D}_r$ are always smaller in terms of dimensions compared to the unrestricted solution, requiring less parameters to be estimated. However, if the number of unrestricted users or items is large the size of $\boldsymbol{I}_x$ and $\boldsymbol{I}_g$ increases. In turn, $\boldsymbol{X}$ and $\boldsymbol{G}$ become harder to store and calculating inverses become computationally infeasible. To avoid these issues we formulate the partially bi-additive model as

$$\boldsymbol{\Gamma} = \boldsymbol{\alpha}_{pr}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_{pr}' + \boldsymbol{X}\boldsymbol{C}_{pr}\boldsymbol{D}_{pr}'\boldsymbol{G}'. \tag{70}$$

Now we introduce a block structure to the parameters corresponding to the users and items for which we do and do not have additional information, defining $\boldsymbol{\alpha}_{pr}$ and $\boldsymbol{\beta}_{pr}'$ as

$$\boldsymbol{\alpha}_{pr} = \left[\begin{array}{c} \boldsymbol{\alpha}_1 \\ \hline \boldsymbol{\alpha}_2 \end{array}\right] \text{ and } \boldsymbol{\beta}_{pr}' = \left[\begin{array}{c|c} \boldsymbol{\beta}_1' & \boldsymbol{\beta}_2' \end{array}\right], \tag{71}$$

where $\boldsymbol{\alpha}_{pr}$ and $\boldsymbol{\beta}_{pr}$ are the respective $n_u \times 1$ and $1 \times n_i$ matrices containing two vectors of which the first corresponds to the restricted users and items and the second corresponds to the unrestricted users and items. Furthermore, we define $\boldsymbol{X}$ and $\boldsymbol{G}$ as in (69). Lastly, we define $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$ as

$$\boldsymbol{C}_{pr} = \left[\begin{array}{c} \boldsymbol{C}_r \\ \hline \boldsymbol{C} \end{array}\right] \text{ and } \boldsymbol{D}_{pr} = \left[\begin{array}{c} \boldsymbol{D}_r \\ \hline \boldsymbol{D} \end{array}\right], \tag{72}$$

where $\boldsymbol{C}_r$ and $\boldsymbol{D}_r$ correspond to restricted bi-additive terms whereas $\boldsymbol{C}$ and $\boldsymbol{D}$ correspond to unrestricted bi-additive terms. Majorization of the likelihood function and the formulation of $\boldsymbol{H}_{pr}$ is identical to the previous sections. We formulate $\boldsymbol{H}_{pr}$ as

$$\begin{aligned} \boldsymbol{H}_{pr} &= \boldsymbol{Z}_s + \boldsymbol{\Gamma}^{(0)} \\ &= (\boldsymbol{Z}_s) + \left(\boldsymbol{\alpha}_{pr}^{(0)}\boldsymbol{1}' + \boldsymbol{1}\boldsymbol{\beta}_{pr}'^{(0)} + \boldsymbol{X}\boldsymbol{C}_{pr}^{(0)}\boldsymbol{D}_{pr}'^{(0)}\boldsymbol{G}'\right) \\ &= \text{sparse} + \text{low-rank}, \end{aligned} \tag{73}$$

where $\boldsymbol{Z}_s$ is defined as in (31). We can define $\boldsymbol{H}_{pr}$ similar to the block structure observed in the parameters in (71)-(72) as

$$\boldsymbol{H}_{pr} = \left[\begin{array}{c|c} \boldsymbol{H}_{1,1} & \boldsymbol{H}_{1,2} \\ \hline \boldsymbol{H}_{2,1} & \boldsymbol{H}_{2,2} \end{array}\right], \tag{74}$$

where $\boldsymbol{H}_{1,1}$ is the $n_x \times n_g$ block corresponding to the users and items for which additional information is observed and $\boldsymbol{H}_{2,2}$ is the $(n_u - n_x) \times (n_i - n_g)$ block corresponding to the users and items for which no additional information is observed, the remaining two blocks correspond to the user and item combinations for which we only observe either a restricted user or item, not both. We rename the indices $n_o = (n_u - n_x)$ and $n_p = (n_i - n_g)$ as we will be often using them in this section. For identification purposes we column center $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{X}^*, \boldsymbol{G}^*, \boldsymbol{C}$ and $\boldsymbol{D}$. It should be noted that this method is more restrictive on $\boldsymbol{\Gamma}$ as opposed to the methods discussed in previous sections. We now center both $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ whereas before we only required the entire $\boldsymbol{\beta}$ to be centered as a whole. Similarly we now require $\boldsymbol{X}^*, \boldsymbol{G}^*, \boldsymbol{C}$ and $\boldsymbol{D}$ to be centered independently whereas before we required the entire bi-additive interaction effect to be column centered.

Now we can rewrite the majorized likelihood as

$$\begin{aligned}
&-\log \mathcal{L}(\boldsymbol{\Gamma}|\boldsymbol{Y}) \\
&\quad\leq \frac{1}{8}\|\boldsymbol{H}_{pr} - \boldsymbol{\Gamma}\|_{\mathrm{F}}^2 + s \\
&\quad= \frac{1}{8}\left\|\boldsymbol{H}_{pr} - (\boldsymbol{\alpha}_{pr}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_{pr}' + \boldsymbol{X}\boldsymbol{C}_{pr}\boldsymbol{D}_{pr}'\boldsymbol{G}')\right\|_{\mathrm{F}}^2 + s \\
&\quad= \frac{1}{8}\left\|\left[\begin{array}{c|c} \boldsymbol{H}_{1,1} & \boldsymbol{H}_{1,2} \\ \hline \boldsymbol{H}_{2,1} & \boldsymbol{H}_{2,2} \end{array}\right] - \left(\left[\begin{array}{c} \boldsymbol{\alpha}_1 \\ \hline \boldsymbol{\alpha}_2 \end{array}\right]\mathbf{1}' + \mathbf{1}\left[\begin{array}{c|c} \boldsymbol{\beta}_1' & \boldsymbol{\beta}_2' \end{array}\right]\right.\right. \\
&\qquad\left.\left. + \left[\begin{array}{c|c} \boldsymbol{X}^* & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{I}_x \end{array}\right]\left[\begin{array}{c} \boldsymbol{C}_r \\ \hline \boldsymbol{C} \end{array}\right]\left[\begin{array}{c|c} \boldsymbol{D}_r' & \boldsymbol{D}' \end{array}\right]\left[\begin{array}{c|c} \boldsymbol{G}^* & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{I}_g \end{array}\right]\right)\right\|_{\mathrm{F}}^2 + s \\
&\quad= \frac{1}{8}\left\|\left[\begin{array}{c|c} \boldsymbol{H}_{1,1} - (\boldsymbol{\alpha}_1\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_1' + \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*) & \boldsymbol{H}_{1,2} - (\boldsymbol{\alpha}_1\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_2' + \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}') \\ \hline \boldsymbol{H}_{2,1} - (\boldsymbol{\alpha}_2\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_1' + \boldsymbol{C}\boldsymbol{D}_r'\boldsymbol{G}'^*) & \boldsymbol{H}_{2,2} - (\boldsymbol{\alpha}_2\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_2' + \boldsymbol{C}\boldsymbol{D}') \end{array}\right]\right\|_F^2 + s. \qquad (75)
\end{aligned}$$

If we define each block in (75) as $\boldsymbol{\Omega}_{a,b}$ with the indices a, b corresponding to the position in the block structure we can rewrite

$$\begin{aligned}
&-\log \mathcal{L}(\boldsymbol{\Gamma}|\boldsymbol{Y}) \\
&\quad\leq \frac{1}{8}\left\|\left[\begin{array}{c|c} \boldsymbol{H}_{1,1} - (\boldsymbol{\alpha}_1\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_1' + \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*) & \boldsymbol{H}_{1,2} - (\boldsymbol{\alpha}_1\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_2' + \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}') \\ \hline \boldsymbol{H}_{2,1} - (\boldsymbol{\alpha}_2\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_1' + \boldsymbol{C}\boldsymbol{D}_r'\boldsymbol{G}'^*) & \boldsymbol{H}_{2,2} - (\boldsymbol{\alpha}_2\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_2' + \boldsymbol{C}\boldsymbol{D}') \end{array}\right]\right\|_F^2 + s \\
&\quad= \frac{1}{8}\left\|\left[\begin{array}{c|c} \boldsymbol{\Omega}_{1,1} & \boldsymbol{\Omega}_{1,2} \\ \hline \boldsymbol{\Omega}_{2,1} & \boldsymbol{\Omega}_{2,2} \end{array}\right]\right\|_F^2 + s \\
&\quad= \frac{1}{8}\left(\|\boldsymbol{\Omega}_{1,1}\|_F^2 + \|\boldsymbol{\Omega}_{1,2}\|_F^2 + \|\boldsymbol{\Omega}_{2,1}\|_F^2 + \|\boldsymbol{\Omega}_{2,2}\|_F^2\right) + s. \qquad (76)
\end{aligned}$$

Now, with $\|\boldsymbol{\Omega}_{1,1}\|_F^2$ as an example, each of the four norms can be rewritten in the following fashion

$$\begin{aligned}
&\|\boldsymbol{\Omega}_{1,1}\|_F^2 \\
&\quad= \left\|\boldsymbol{H}_{1,1} - (\boldsymbol{\alpha}_1\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_1' + \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*)\right\|_{\mathrm{F}}^2 \\
&\quad= \left\|(\boldsymbol{J}_{n_x}\boldsymbol{H}_{1,1}\boldsymbol{J}_{n_g} + n_g^{-1}\boldsymbol{H}_{1,1}\mathbf{1}\mathbf{1}' + n_x^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{H}_{1,1}\boldsymbol{J}_{n_g}) - (\boldsymbol{\alpha}_1\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}_1' + \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*)\right\|_{\mathrm{F}}^2 \\
&\quad= \left\|(n_g^{-1}\boldsymbol{H}_{1,1}\mathbf{1}\mathbf{1}' - \boldsymbol{\alpha}_1\mathbf{1}') + (n_x^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{H}_{1,1}\boldsymbol{J}_{n_g} - \mathbf{1}\boldsymbol{\beta}_1') + (\boldsymbol{J}_{n_x}\boldsymbol{H}_{1,1}\boldsymbol{J}_{n_g} - \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*)\right\|_{\mathrm{F}}^2 \\
&\quad= \left\|\left(n_g^{-1}\boldsymbol{H}_{1,1}\mathbf{1} - \boldsymbol{\alpha}_1\right)\mathbf{1}'\right\|_{\mathrm{F}}^2 + \left\|\mathbf{1}\left(n_x^{-1}\mathbf{1}'\boldsymbol{H}_{1,1} - \boldsymbol{\beta}_1'\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2 \\
&\qquad + \left\|\boldsymbol{J}_{n_x}\left(\boldsymbol{H}_{1,1} - \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2, \qquad (77)
\end{aligned}$$

where we use the cyclic property of the trace and the fact that $\boldsymbol{\beta_1}$ and $\boldsymbol{X}^*$ are column centered to drop the cross products. Now we can rewrite (76) as

$$\begin{aligned}
&-\log \mathcal{L}(\boldsymbol{\Gamma}|\boldsymbol{Y})\\
&\quad\leq \frac{1}{8}\left(\|\boldsymbol{\Omega}_{1,1}\|_F^2 + \|\boldsymbol{\Omega}_{1,2}\|_F^2 + \|\boldsymbol{\Omega}_{2,1}\|_F^2 + \|\boldsymbol{\Omega}_{2,2}\|_F^2\right) + s\\
&\quad= \frac{1}{8}\Big(\left\|\left(n_g^{-1}\boldsymbol{H}_{1,1}\mathbf{1} - \boldsymbol{\alpha}_1\right)\mathbf{1}'\right\|_{\mathrm{F}}^2 + \left\|\mathbf{1}\left(n_x^{-1}\mathbf{1}'\boldsymbol{H}_{1,1} - \boldsymbol{\beta}_1'\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2 + \left\|\boldsymbol{J}_{n_x}\left(\boldsymbol{H}_{1,1} - \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2\\
&\qquad+ \left\|\left(n_p^{-1}\boldsymbol{H}_{1,2}\mathbf{1} - \boldsymbol{\alpha}_1\right)\mathbf{1}'\right\|_{\mathrm{F}}^2 + \left\|\mathbf{1}\left(n_x^{-1}\mathbf{1}'\boldsymbol{H}_{1,2} - \boldsymbol{\beta}_2'\right)\boldsymbol{J}_{n_p}\right\|_{\mathrm{F}}^2\\
&\qquad+ \left\|\boldsymbol{J}_{n_x}\left(\boldsymbol{H}_{1,2} - \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}'\right)\boldsymbol{J}_{n_p}\right\|_{\mathrm{F}}^2 + \left\|\left(n_g^{-1}\boldsymbol{H}_{2,1}\mathbf{1} - \boldsymbol{\alpha}_2\right)\mathbf{1}'\right\|_{\mathrm{F}}^2\\
&\qquad+ \left\|\mathbf{1}\left(n_o^{-1}\mathbf{1}'\boldsymbol{H}_{2,1} - \boldsymbol{\beta}_1'\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2 + \left\|\boldsymbol{J}_{n_o}\left(\boldsymbol{H}_{2,1} - \boldsymbol{C}\boldsymbol{D}_r'\boldsymbol{G}'^*\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2\\
&\qquad+ \left\|\left(n_p^{-1}\boldsymbol{H}_{2,2}\mathbf{1} - \boldsymbol{\alpha}_2\right)\mathbf{1}'\right\|_{\mathrm{F}}^2 + \left\|\mathbf{1}\left(n_o^{-1}\mathbf{1}'\boldsymbol{H}_{2,2} - \boldsymbol{\beta}_2'\right)\boldsymbol{J}_{n_p}\right\|_{\mathrm{F}}^2\\
&\qquad+ \left\|\boldsymbol{J}_{n_o}\left(\boldsymbol{H}_{2,2} - \boldsymbol{C}\boldsymbol{D}'\right)\boldsymbol{J}_{n_p}\right\|_{\mathrm{F}}^2\Big) + s. \qquad (78)
\end{aligned}$$

We use (78) to derive parameter updates. First we minimize (78) with respects to $\boldsymbol{\alpha}_1$, yielding

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}_1} \ \frac{1}{8}\left(\left\|\left(n_g^{-1}\boldsymbol{H}_{1,1}\mathbf{1} - \boldsymbol{\alpha}_1\right)\mathbf{1}'\right\|_{\mathrm{F}}^2 + \left\|(n_p^{-1}\boldsymbol{H}_{1,2}\mathbf{1} - \boldsymbol{\alpha}_1)\mathbf{1}'\right\|_{\mathrm{F}}^2\right)\\
&\qquad= \min_{\boldsymbol{\alpha}_1} \frac{1}{8}\left(\left\|\left[\ n_g^{-1}\boldsymbol{H}_{1,1}\mathbf{1}\mathbf{1}' - \boldsymbol{\alpha}_1\mathbf{1}' \ \middle|\ n_p^{-1}\boldsymbol{H}_{1,2}\mathbf{1}\mathbf{1}' - \boldsymbol{\alpha}_1\mathbf{1}'\ \right]\right\|_F^2\right). \qquad (79)
\end{aligned}$$

Equation (79) is most intuitively minimized by considering each separate element of $\boldsymbol{\alpha}_1$. Minimizing (79) with respects to $\alpha_{1,q}$ for $q \in \{1, ..., n_x\}$, where $\boldsymbol{h}'_{(1,1),q}$ equals row q of $\boldsymbol{H}_{1,1}$ yields

$$\begin{aligned}
&\min_{\alpha_{1,q}} \ \left\|\left[\ n_g^{-1}\boldsymbol{H}_{1,1}\mathbf{1}\mathbf{1}' - \boldsymbol{\alpha}_1\mathbf{1}' \ \middle|\ n_p^{-1}\boldsymbol{H}_{1,2}\mathbf{1}\mathbf{1}' - \boldsymbol{\alpha}_1\mathbf{1}'\ \right]\right\|_F^2\\
&\qquad= \min_{\alpha_{1,q}} \left\|\left[\ n_g^{-1}\boldsymbol{h}'_{(1,1),q}\mathbf{1}\mathbf{1}' - \alpha_{1,q}\mathbf{1}' \ \middle|\ n_p^{-1}\boldsymbol{h}'_{(1,2),q}\mathbf{1}\mathbf{1}' - \alpha_{1,q}\mathbf{1}'\ \right]\right\|_F^2\\
&\qquad= \min_{\alpha_{1,q}} n_g(\bar{h}_{(1,1),q} - \alpha_{1,q})^2 + n_p(\bar{h}_{(1,2),q} - \alpha_{1,q})^2\\
&\qquad\Leftrightarrow \frac{d}{d\alpha_1}\left(n_g(\bar{h}_{(1,1),q} - \alpha_{1,q})^2 + n_p(\bar{h}_{(1,2),q} - \alpha_{1,q})^2\right) = 0\\
&\qquad\Leftrightarrow -2n_g(\bar{h}_{(1,1),q} - \alpha_{1,q}) - 2n_p(\bar{h}_{(1,2),q} - \alpha_{1,q}) = 0\\
&\qquad\Leftrightarrow \alpha_{1,q} = \frac{n_g\bar{h}_{(1,1),q} + n_p\bar{h}_{(1,2),q}}{n_g + n_p}. \qquad (80)
\end{aligned}$$

In other words, (79) is minimized if we set $\boldsymbol{\alpha}_1$ equal to the weighted row means of $\boldsymbol{H}_{1,1}$. Similar reasoning for $\boldsymbol{\alpha_2}, \boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$ yield the following updates

$$\boldsymbol{\alpha}_1 = \frac{n_g\bar{\boldsymbol{h}}_{1,1} + n_p\bar{\boldsymbol{h}}_{1,2}}{n_g + n_p}, \boldsymbol{\alpha}_2 = \frac{n_g\bar{\boldsymbol{h}}_{2,1} + n_p\bar{\boldsymbol{h}}_{1,1}}{n_g + n_p}, \qquad (81)$$

$$\boldsymbol{\beta}_1' = \frac{n_x\hat{\boldsymbol{h}}_{1,1} + n_o\hat{\boldsymbol{h}}_{2,1}}{n_x + n_o} \text{ and } \boldsymbol{\beta}_2' = \frac{n_x\hat{\boldsymbol{h}}_{1,2} + n_o\hat{\boldsymbol{h}}_{2,2}}{n_x + n_o}, \qquad (82)$$

where $\bar{\boldsymbol{h}}_{1,1} = n_g^{-1}\boldsymbol{H}_{1,1}\boldsymbol{1}$ and $\hat{\boldsymbol{h}}_{1,1} = n_x^{-1}\boldsymbol{1}'\boldsymbol{H}_{1,1}\boldsymbol{J}_{n_g}$. The remaining to be minimized terms in (78) can be written as

$$\begin{aligned}
&\frac{1}{8}\Big(\left\|\boldsymbol{J}_{n_x}\left(\boldsymbol{H}_{1,1} - \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2 + \left\|\boldsymbol{J}_{n_x}\left(\boldsymbol{H}_{1,2} - \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}'\right)\boldsymbol{J}_{n_p}\right\|_{\mathrm{F}}^2 \\
&\qquad + \left\|\boldsymbol{J}_{n_o}\left(\boldsymbol{H}_{2,1} - \boldsymbol{C}\boldsymbol{D}_r'\boldsymbol{G}'^*\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2 + \left\|\boldsymbol{J}_{n_o}\left(\boldsymbol{H}_{2,2} - \boldsymbol{C}\boldsymbol{D}'\right)\boldsymbol{J}_{n_p}\right\|_{\mathrm{F}}^2\Big) \\
&= \frac{1}{8}\left\|\left[\begin{array}{c|c} \boldsymbol{J}_{n_x} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{J}_{n_o}\end{array}\right]\left(\boldsymbol{H}_{pr} - \boldsymbol{X}\boldsymbol{C}_{pr}\boldsymbol{D}_{pr}'\boldsymbol{G}'\right)\left[\begin{array}{c|c} \boldsymbol{J}_{n_g} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{J}_{n_p}\end{array}\right]\right\|_F^2 \\
&= \frac{1}{8}\left\|\widetilde{\boldsymbol{H}}_{pr} - \boldsymbol{X}\boldsymbol{C}_{pr}\boldsymbol{D}_{pr}'\boldsymbol{G}'\right\|_F^2, \qquad (83)
\end{aligned}$$

as $\boldsymbol{X}^*, \boldsymbol{G}^*, \boldsymbol{C}$ and $\boldsymbol{D}$ are column centered.

3.5.1 Regularization

Again, rank restricted parameter regularization of $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$ is required to avoid the overfitting problem as discussed in Groenen et al. (2003). We can introduce the weighted nuclear norm similar to Section 3.4.1 as

$$\frac{1}{8}\left\|\widetilde{\boldsymbol{H}}_{pr} - \boldsymbol{X}\boldsymbol{C}_{pr}\boldsymbol{D}_{pr}'\boldsymbol{G}'\right\|_F^2 + \frac{\lambda}{2}\left(\|\boldsymbol{X}\boldsymbol{C}_{pr}\|_{\mathrm{F}}^2 + \|\boldsymbol{G}\boldsymbol{D}_{pr}\|_{\mathrm{F}}^2\right). \qquad (84)$$

Now, we can take two approaches to minimize (84). Firstly, we can implement the same methodology as used in Section 3.4 and update both $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$ at once. Secondly, we can take a block relaxation approach by fixing all but one of the matrices $\boldsymbol{C}_r$, $\boldsymbol{D}_r$, $\boldsymbol{C}$ and $\boldsymbol{D}$ and minimizing (84) with respect to the remaining matrix. In the first minimization approach we minimize (84) with respects to both $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$ and follow the same approach as described in Section 3.4.1, using Algorithm 3 to obtain updates for $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$. The block structure of $\boldsymbol{X}$ and $\boldsymbol{G}$ does allow us to rewrite of the matrices matrices in Algorithm 3. If we redefine $\widetilde{\boldsymbol{V}}$ and $\widetilde{\boldsymbol{U}}$ from Algorithm 3 as

$$\widetilde{\boldsymbol{V}}_{pr} = \left[\begin{array}{c} \widetilde{\boldsymbol{V}}_r \\ \hline \widetilde{\boldsymbol{V}} \end{array}\right] \text{ and } \widetilde{\boldsymbol{U}}_{pr} = \left[\begin{array}{c} \widetilde{\boldsymbol{U}}_r \\ \hline \widetilde{\boldsymbol{U}} \end{array}\right], \qquad (85)$$

where the $\widetilde{\boldsymbol{V}}_r$ and $\widetilde{\boldsymbol{U}}_r$ correspond to the restricted terms whereas $\widetilde{\boldsymbol{V}}$ and $\widetilde{\boldsymbol{U}}$ correspond to the unrestricted terms. We should also implement the block structure from this section in the other matrices in Algorithm 3. In Step 3 of the algorithm we have to calculate $\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}_{pr}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}_{pr}'\widetilde{\boldsymbol{X}}$, this matrix can be rewritten. For simplicity we will first start with $\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}}$

$$\begin{aligned}
&\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}} \\
&= \left[\begin{array}{c|c} \widetilde{\boldsymbol{G}}^* & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I}_g \end{array}\right] \left(\left[\begin{array}{c|c} \boldsymbol{J}_{n_x} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{J}_{n_o} \end{array}\right] \left[\begin{array}{c|c} \boldsymbol{H}_{1,1} & \boldsymbol{H}_{1,2} \\ \hline \boldsymbol{H}_{2,1} & \boldsymbol{H}_{2,2} \end{array}\right] \left[\begin{array}{c|c} \boldsymbol{J}_{n_g} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{J}_{n_p} \end{array}\right] \right)' \left[\begin{array}{c|c} \widetilde{\boldsymbol{X}}^* & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I}_g \end{array}\right] \\
&= \left[\begin{array}{c|c} \widetilde{\boldsymbol{G}}^* & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{J}_{n_p} \end{array}\right] \left[\begin{array}{c|c} \boldsymbol{H}'_{1,1} & \boldsymbol{H}'_{2,1} \\ \hline \boldsymbol{H}'_{1,2} & \boldsymbol{H}'_{2,2} \end{array}\right] \left[\begin{array}{c|c} \widetilde{\boldsymbol{X}}^* & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{J}_{n_o} \end{array}\right] \\
&= \left[\begin{array}{c|c} \widetilde{\boldsymbol{G}}^*\boldsymbol{H}'_{1,1}\widetilde{\boldsymbol{X}}^* & \widetilde{\boldsymbol{G}}^*\boldsymbol{H}'_{2,1}\boldsymbol{J}_{n_o} \\ \hline \boldsymbol{J}_{n_p}\boldsymbol{H}'_{1,2}\widetilde{\boldsymbol{X}}^* & \boldsymbol{J}_{n_p}\boldsymbol{H}'_{2,2}\boldsymbol{J}_{n_o} \end{array}\right] \\
&= \left[\begin{array}{c|c} \boldsymbol{\Xi}_{1,1} & \boldsymbol{\Xi}_{1,2} \\ \hline \boldsymbol{\Xi}_{2,1} & \boldsymbol{\Xi}_{2,2} \end{array}\right] \qquad (86)
\end{aligned}$$

where we again use the fact that $\boldsymbol{G}^*$ and $\boldsymbol{X}^*$ are column centered. Where we briefly redefine the blocks using $\boldsymbol{\Xi}$ for ease of derivation. We can then rewrite $\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'_{pr}\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}}$ as

$$\begin{aligned}
\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'_{pr}\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}} &= \widetilde{\boldsymbol{\Phi}} \left[\begin{array}{c|c} \widetilde{\boldsymbol{V}}'_r & \widetilde{\boldsymbol{V}}' \end{array}\right] \left[\begin{array}{c|c} \boldsymbol{\Xi}_{1,1} & \boldsymbol{\Xi}_{1,2} \\ \hline \boldsymbol{\Xi}_{2,1} & \boldsymbol{\Xi}_{2,2} \end{array}\right] \\
&= \widetilde{\boldsymbol{\Phi}} \left[\begin{array}{c|c} \widetilde{\boldsymbol{V}}'_r\boldsymbol{\Xi}_{1,1} + \widetilde{\boldsymbol{V}}'\boldsymbol{\Xi}_{2,1} & \widetilde{\boldsymbol{V}}'_r\boldsymbol{\Xi}_{1,2} + \widetilde{\boldsymbol{V}}'\boldsymbol{\Xi}_{2,2} \end{array}\right] \qquad (87)
\end{aligned}$$

We can simplify each block as

$$\begin{aligned}
\boldsymbol{\Xi}_{1,1} = \widetilde{\boldsymbol{G}}^*\boldsymbol{H}'_{1,1}\widetilde{\boldsymbol{X}}^* &= \widetilde{\boldsymbol{G}}^* \left(\boldsymbol{Z}'_{s,(1,1)} + \boldsymbol{1}\boldsymbol{\alpha}'_1 + \boldsymbol{\beta}_1\boldsymbol{1}' + \boldsymbol{G}^*\boldsymbol{D}_r\boldsymbol{C}'_r\boldsymbol{X}^{*\prime} \right) \widetilde{\boldsymbol{X}}^* \\
&= \widetilde{\boldsymbol{G}}^*\boldsymbol{Z}'_{s,(1,1)}\widetilde{\boldsymbol{X}}^* + \widetilde{\boldsymbol{G}}^*\boldsymbol{G}^*\boldsymbol{D}_r\boldsymbol{C}'_r\boldsymbol{X}^{*\prime}\widetilde{\boldsymbol{X}}^*, \qquad (88)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\Xi}_{1,2} = \widetilde{\boldsymbol{G}}^*\boldsymbol{H}'_{2,1}\boldsymbol{J}_{n_o} &= \widetilde{\boldsymbol{G}}^* \left(\boldsymbol{Z}'_{s,(2,1)} + \boldsymbol{1}\boldsymbol{\alpha}'_2 + \boldsymbol{\beta}_1\boldsymbol{1}' + \boldsymbol{G}^*\boldsymbol{D}_r\boldsymbol{C}' \right) \boldsymbol{J}_{n_o} \\
&= \widetilde{\boldsymbol{G}}^*\boldsymbol{Z}'_{s,(2,1)}\boldsymbol{J}_{n_o} + \widetilde{\boldsymbol{G}}^*\boldsymbol{G}^*\boldsymbol{D}_r\boldsymbol{C}', \qquad (89)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\Xi}_{2,1} = \boldsymbol{J}_{n_p}\boldsymbol{H}'_{1,2}\widetilde{\boldsymbol{X}}^* &= \boldsymbol{J}_{n_p} \left(\boldsymbol{Z}'_{s,(1,2)} + \boldsymbol{1}\boldsymbol{\alpha}'_1 + \boldsymbol{\beta}_2\boldsymbol{1}' + \boldsymbol{D}\boldsymbol{C}'_r\boldsymbol{X}^{*\prime} \right) \widetilde{\boldsymbol{X}}^* \\
&= \boldsymbol{J}_{n_p}\boldsymbol{Z}'_{s,(1,2)}\widetilde{\boldsymbol{X}}^* + \boldsymbol{D}\boldsymbol{C}'_r\boldsymbol{X}^{*\prime}\widetilde{\boldsymbol{X}}^*, \qquad (90)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\Xi}_{2,2} = \boldsymbol{J}_{n_p}\boldsymbol{H}'_{2,2}\boldsymbol{J}_{n_o} &= \boldsymbol{J}_{n_p} \left(\boldsymbol{Z}_{s,(2,2)} + \boldsymbol{1}\boldsymbol{\alpha}'_2 + \boldsymbol{\beta}_2\boldsymbol{1}' + \boldsymbol{D}_r\boldsymbol{C}'_r \right) \boldsymbol{J}_{n_o} \\
&= \boldsymbol{J}_{n_p}\boldsymbol{Z}'_{s,(2,2)}\boldsymbol{J}_{n_o} + \boldsymbol{D}\boldsymbol{C}', \qquad (91)
\end{aligned}$$

since both $\boldsymbol{D}$ and $\boldsymbol{C}$ are column centered and $\boldsymbol{J}_{n_p}\boldsymbol{1} = \boldsymbol{1}'\boldsymbol{J}_{n_o} = \boldsymbol{0}$. Similarly, we can use the block structure in both $\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}_{pr}\widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{F}} = \widetilde{\boldsymbol{V}}'_{pr}\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}_{pr}$ to find

$$\begin{aligned}\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}_{pr}\widetilde{\boldsymbol{\Phi}} &= \left[\begin{array}{c|c} \boldsymbol{\Xi}_{1,1} & \boldsymbol{\Xi}_{1,2} \\ \hline \boldsymbol{\Xi}_{2,1} & \boldsymbol{\Xi}_{2,2} \end{array}\right] \left[\begin{array}{c} \widetilde{\boldsymbol{U}}_r \\ \hline \widetilde{\boldsymbol{U}} \end{array}\right] \widetilde{\boldsymbol{\Phi}} \\ &= \left[\begin{array}{c} \boldsymbol{\Xi}_{1,1}\widetilde{\boldsymbol{U}}_r + \boldsymbol{\Xi}_{1,2}\widetilde{\boldsymbol{U}} \\ \hline \boldsymbol{\Xi}_{2,1}\widetilde{\boldsymbol{U}}_r + \boldsymbol{\Xi}_{2,2}\widetilde{\boldsymbol{U}} \end{array}\right] \widetilde{\boldsymbol{\Phi}}, \end{aligned} \tag{92}$$

and

$$\begin{aligned}\widetilde{\boldsymbol{F}} = \widetilde{\boldsymbol{V}}'\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}} &= \left[\begin{array}{c|c} \widetilde{\boldsymbol{V}}'_r & \widetilde{\boldsymbol{V}}' \end{array}\right] \left[\begin{array}{c|c} \boldsymbol{\Xi}_{1,1} & \boldsymbol{\Xi}_{1,2} \\ \hline \boldsymbol{\Xi}_{2,1} & \boldsymbol{\Xi}_{2,2} \end{array}\right] \left[\begin{array}{c} \widetilde{\boldsymbol{U}}_r \\ \hline \widetilde{\boldsymbol{U}} \end{array}\right] \\ &= \left[\widetilde{\boldsymbol{V}}'_r\boldsymbol{\Xi}_{1,1}\widetilde{\boldsymbol{U}}_r + \widetilde{\boldsymbol{V}}'_r\boldsymbol{\Xi}_{1,2}\widetilde{\boldsymbol{U}} + \widetilde{\boldsymbol{V}}'\boldsymbol{\Xi}_{2,1}\widetilde{\boldsymbol{U}}_r + \widetilde{\boldsymbol{V}}'\boldsymbol{\Xi}_{2,2}\widetilde{\boldsymbol{U}}\right]. \end{aligned} \tag{93}$$

Algorithm 5 summarizes these changes.

Algorithm 5 Updating of $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$ using the SVD approach

```
Initialize
    $\boldsymbol{Z}_s \leftarrow \boldsymbol{Z}_s$
    $\widetilde{\boldsymbol{V}}_{pr} \leftarrow$ a $(d_x + n_o) \times r$ random matrix with orthonormal columns
    $\widetilde{\boldsymbol{\Phi}} \leftarrow \boldsymbol{I}_r$
    $\boldsymbol{X}^* \leftarrow \boldsymbol{X}^*$
    $\boldsymbol{G}^* \leftarrow \boldsymbol{G}^*$
repeat
    Given $\widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{V}}_{pr}$, solve for $\widetilde{\boldsymbol{U}}_{pr}$:
        Compute the SVD of $\widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'_{pr}\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}} = \widetilde{\boldsymbol{\Phi}} \left[ \widetilde{\boldsymbol{V}}'_r\boldsymbol{\Xi}_{1,1} + \widetilde{\boldsymbol{V}}'\boldsymbol{\Xi}_{2,1} \;\middle|\; \widetilde{\boldsymbol{V}}'_r\boldsymbol{\Xi}_{1,2} + \widetilde{\boldsymbol{V}}'\boldsymbol{\Xi}_{2,2} \right] = \boldsymbol{P\Lambda Q}'$,
        and let $\widetilde{\boldsymbol{U}}_{pr} \leftarrow \boldsymbol{QP}'$
    Given $\widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{U}}_{pr}$, solve for $\widetilde{\boldsymbol{V}}_{pr}$
        Compute the SVD of $\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}_{pr}\widetilde{\boldsymbol{\Phi}} = \left[\begin{array}{c} \boldsymbol{\Xi}_{1,1}\widetilde{\boldsymbol{U}}_r + \boldsymbol{\Xi}_{1,2}\widetilde{\boldsymbol{U}} \\ \hline \boldsymbol{\Xi}_{2,1}\widetilde{\boldsymbol{U}}_r + \boldsymbol{\Xi}_{2,2}\widetilde{\boldsymbol{U}} \end{array}\right] \widetilde{\boldsymbol{\Phi}} = \bar{\boldsymbol{P}}\bar{\boldsymbol{\Lambda}}\bar{\boldsymbol{Q}}'$,
        and let $\widetilde{\boldsymbol{V}}_{pr} \leftarrow \bar{\boldsymbol{P}}\bar{\boldsymbol{Q}}'$
    Given $\widetilde{\boldsymbol{U}}_{pr}$ and $\widetilde{\boldsymbol{V}}_{pr}$, solve for $\widetilde{\boldsymbol{\Phi}}$
        Compute $\widetilde{\boldsymbol{F}} = \left[\widetilde{\boldsymbol{V}}'_r\boldsymbol{\Xi}_{1,1}\widetilde{\boldsymbol{U}}_r + \widetilde{\boldsymbol{V}}'_r\boldsymbol{\Xi}_{1,2}\widetilde{\boldsymbol{U}} + \widetilde{\boldsymbol{V}}'\boldsymbol{\Xi}_{2,1}\widetilde{\boldsymbol{U}}_r + \widetilde{\boldsymbol{V}}'\boldsymbol{\Xi}_{2,2}\widetilde{\boldsymbol{U}}\right]$,
        set $\phi_{ii} \leftarrow \max\left(f_{ii} - 4\lambda, 0\right) \; \forall \; i \in \{1, ..., r\}$
until $\widetilde{\boldsymbol{U}}, \widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{V}}$ have converged
return $\boldsymbol{C}_{pr} = \left[\begin{array}{c} \boldsymbol{C}_r \\ \hline \boldsymbol{C} \end{array}\right] = \left[\begin{array}{c} (\boldsymbol{X}^{*\prime}\boldsymbol{X}^*)^{-\frac{1}{2}}\widetilde{\boldsymbol{U}}_r\widetilde{\boldsymbol{\Phi}}^{1/2} \\ \hline \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}^{1/2} \end{array}\right]$ and $\boldsymbol{D}_{pr} = \left[\begin{array}{c} \boldsymbol{D}_r \\ \hline \boldsymbol{D} \end{array}\right] = \left[\begin{array}{c} (\boldsymbol{G}^{*\prime}\boldsymbol{G}^*)^{-\frac{1}{2}}\widetilde{\boldsymbol{V}}_r\widetilde{\boldsymbol{\Phi}}^{1/2} \\ \hline \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{\Phi}}^{1/2} \end{array}\right]$
```

However, we see that this method requires the SVD of a both a $r \times (d_x + n_o)$ and a $(d_g + n_p) \times r$ matrix. If we lack additional information of a large part of the users or the items, n_o or n_p become large and the SVD may slow down. We can implement another computational trick to replace the SVD using an eigendecomposition. Suppose we define $\boldsymbol{M}$ as an n matrix where $n >> r$ and decompose $\boldsymbol{M}$ by the SVD $\boldsymbol{M} = \boldsymbol{P\Lambda Q}'$. We can then calculate both $\boldsymbol{Q}$ and $\boldsymbol{\Lambda}$ by the eigendecomposition of $\boldsymbol{M}'\boldsymbol{M} = \boldsymbol{Q\Lambda}^2\boldsymbol{Q}'$, where $\boldsymbol{M}'\boldsymbol{M}$ is an $r \times r$ matrix. Now using this $\boldsymbol{Q}$ and $\boldsymbol{\Lambda}$ we can derive

$$\begin{aligned} \boldsymbol{M} = \boldsymbol{P\Lambda Q}' &\Leftrightarrow \boldsymbol{MQ} = \boldsymbol{P\Lambda} \\ &\Leftrightarrow \boldsymbol{MQ\Lambda}^{-1} = \boldsymbol{P}, \end{aligned} \tag{94}$$

to retrieve $\boldsymbol{P}$. Similarly, we can calculate the eigendecomposition of $\boldsymbol{MM}' = \boldsymbol{P\Lambda}^2\boldsymbol{P}'$ and retrieve $\boldsymbol{Q}$

similar to (94). If we define $\boldsymbol{F}_1 = \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'_{pr}\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}}$ and $\boldsymbol{F}_2 = \widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}_{pr}\widetilde{\boldsymbol{\Phi}}$ we can summarize this method in Algorithm 6 as

Algorithm 6 Updating of $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$ using an eigendecomposition

```
1: Initialize
       $\boldsymbol{Z}_s \leftarrow \boldsymbol{Z}_s$
       $\widetilde{\boldsymbol{V}}_{pr} \leftarrow$ a $(d_x + n_o) \times r$ random matrix with orthonormal columns
       $\widetilde{\boldsymbol{\Phi}} \leftarrow \boldsymbol{I}_r$
       $\boldsymbol{X}^* \leftarrow \boldsymbol{X}^*$
       $\boldsymbol{G}^* \leftarrow \boldsymbol{G}^*$
2: repeat
3:     Given $\widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{V}}_{pr}$, solve for $\widetilde{\boldsymbol{U}}_{pr}$

       If the SVD of $\boldsymbol{F}_1 = \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{V}}'_{pr}\widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}} = \boldsymbol{P\Lambda Q}'$, calculate the eigendecomposition $\boldsymbol{F}_1\boldsymbol{F}_1' = \boldsymbol{P\Lambda}^2\boldsymbol{P}'$.
       Now set $\boldsymbol{Q}' = \boldsymbol{\Phi}^{-1}\boldsymbol{P}'\boldsymbol{F}_1$ and let $\widetilde{\boldsymbol{U}}_{pr} \leftarrow \boldsymbol{QP}'$

4:     Given $\widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{U}}_{pr}$, solve for $\widetilde{\boldsymbol{V}}_{pr}$

       If the SVD of $\boldsymbol{F}_2 = \widetilde{\boldsymbol{G}}'\widetilde{\boldsymbol{H}}'_{pr}\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}_{pr}\widetilde{\boldsymbol{\Phi}} = \bar{\boldsymbol{P}}\bar{\boldsymbol{\Lambda}}\bar{\boldsymbol{Q}}'$, calculate the eigendecomposition $\boldsymbol{F}_2'\boldsymbol{F}_2 = \bar{\boldsymbol{Q}}\bar{\boldsymbol{\Lambda}}^2\bar{\boldsymbol{Q}}'$.
       Now set $\bar{\boldsymbol{P}} = \boldsymbol{F}_2\bar{\boldsymbol{Q}}\bar{\boldsymbol{\Phi}}^{-1}$ and let $\widetilde{\boldsymbol{V}}_{pr} \leftarrow \bar{\boldsymbol{P}}\bar{\boldsymbol{Q}}'$

5:     Given $\widetilde{\boldsymbol{U}}_{pr}$ and $\widetilde{\boldsymbol{V}}_{pr}$, solve for $\widetilde{\boldsymbol{\Phi}}$

           Compute $\widetilde{\boldsymbol{F}} = \left[\widetilde{\boldsymbol{V}}_r'\Xi_{1,1}\widetilde{\boldsymbol{U}}_r + \widetilde{\boldsymbol{V}}_r'\Xi_{1,2}\widetilde{\boldsymbol{U}} + \widetilde{\boldsymbol{V}}'\Xi_{2,1}\widetilde{\boldsymbol{U}}_r + \widetilde{\boldsymbol{V}}'\Xi_{2,2}\widetilde{\boldsymbol{U}}\right]$,
           set $\phi_{ii} \leftarrow \max(f_{ii} - 4\lambda, 0)\ \forall\ i \in \{1, ..., r\}$

6: until $\widetilde{\boldsymbol{U}}, \widetilde{\boldsymbol{\Phi}}$ and $\widetilde{\boldsymbol{V}}$ have converged
7: return $\boldsymbol{C}_{pr} = \left[\frac{\boldsymbol{C}_r}{\boldsymbol{C}}\right] = \left[\frac{(\boldsymbol{X}^{*\prime}\boldsymbol{X}^*)^{-\frac{1}{2}}\widetilde{\boldsymbol{U}}_r\widetilde{\boldsymbol{\Phi}}^{1/2}}{\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Phi}}^{1/2}}\right]$ and $\boldsymbol{D}_{pr} = \left[\frac{\boldsymbol{D}_r}{\boldsymbol{D}}\right] = \left[\frac{(\boldsymbol{G}^{*\prime}\boldsymbol{G}^*)^{-\frac{1}{2}}\widetilde{\boldsymbol{V}}_r\widetilde{\boldsymbol{\Phi}}^{1/2}}{\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{\Phi}}^{1/2}}\right]$
```

The block relaxation approach involves minimizing (84) using a more traditional approach. We rewrite (84) to find

$$
\begin{aligned}
&\frac{1}{8}\left\|\widetilde{\boldsymbol{H}}_{pr} - \boldsymbol{X}\boldsymbol{C}_{pr}\boldsymbol{D}_{pr}'\boldsymbol{G}'\right\|_F^2 + \frac{\lambda}{2}\left(\|\boldsymbol{X}\boldsymbol{C}_{pr}\|_{\mathrm{F}}^2 + \|\boldsymbol{G}\boldsymbol{D}_{pr}\|_{\mathrm{F}}^2\right) \\
&\quad = \frac{1}{8}\Big(\left\|\boldsymbol{J}_{n_x}\left(\boldsymbol{H}_{1,1} - \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2 + \left\|\boldsymbol{J}_{n_x}\left(\boldsymbol{H}_{1,2} - \boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}'\right)\boldsymbol{J}_{n_p}\right\|_{\mathrm{F}}^2 \\
&\qquad + \left\|\boldsymbol{J}_{n_o}\left(\boldsymbol{H}_{2,1} - \boldsymbol{C}\boldsymbol{D}_r'\boldsymbol{G}'^*\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2 + \left\|\boldsymbol{J}_{n_o}\left(\boldsymbol{H}_{2,2} - \boldsymbol{C}\boldsymbol{D}'\right)\boldsymbol{J}_{n_p}\right\|_{\mathrm{F}}^2\Big) \\
&\qquad + \frac{\lambda}{2}\left(\|\boldsymbol{X}^*\boldsymbol{C}_r\|_{\mathrm{F}}^2 + \|\boldsymbol{C}\|_{\mathrm{F}}^2 + \|\boldsymbol{G}^*\boldsymbol{D}_r\|_{\mathrm{F}}^2 + \|\boldsymbol{D}\|_{\mathrm{F}}^2\right).
\end{aligned}
\tag{95}
$$

We now fix the other parameters and minimize (95) with respects to $\boldsymbol{C}_r$, this yields the minimization problem

$$\begin{aligned}
&\min_{\boldsymbol{C}_r} \; \frac{1}{8}\Big(\left\|\boldsymbol{J}_{n_x}\left(\boldsymbol{H}_{1,1}-\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*\right)\boldsymbol{J}_{n_g}\right\|_{\mathrm{F}}^2+\left\|\boldsymbol{J}_{n_x}\left(\boldsymbol{H}_{1,2}-\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}'\right)\boldsymbol{J}_{n_p}\right\|_{\mathrm{F}}^2\Big)+\frac{\lambda}{2}\|\boldsymbol{X}^*\boldsymbol{C}_r\|_{\mathrm{F}}^2\\
&= \min_{\boldsymbol{C}_r} \; \frac{1}{8}\big(\|\widetilde{\boldsymbol{H}}_{1,1}-\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_r'\boldsymbol{G}'^*\|_F^2+\|\widetilde{\boldsymbol{H}}_{1,2}-\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}'\|_F^2\big)+\frac{\lambda}{2}\|\boldsymbol{X}^*\boldsymbol{C}_r\|_{\mathrm{F}}^2\\
&= \min_{\boldsymbol{C}_r} \; \frac{1}{8}\|\widetilde{\boldsymbol{H}}_{C_r}-\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_{pr}'\boldsymbol{G}'\|_F^2+\frac{\lambda}{2}\|\boldsymbol{X}^*\boldsymbol{C}_r\|_{\mathrm{F}}^2\\
&= \min_{\boldsymbol{C}_r} \; \frac{1}{8}\|\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_{pr}'\boldsymbol{G}'\|_F^2-\frac{1}{4}\operatorname{tr}(\widetilde{\boldsymbol{H}}_{C_r}'\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_{pr}'\boldsymbol{G}')+\frac{\lambda}{2}\|\boldsymbol{X}^*\boldsymbol{C}_r\|_{\mathrm{F}}^2,
\end{aligned} \tag{96}$$

where $\widetilde{\boldsymbol{H}}_{C_r}=\left[\begin{array}{c|c}\widetilde{\boldsymbol{H}}_{1,1}' & \widetilde{\boldsymbol{H}}_{1,2}'\end{array}\right]'$ and dropping the parts not containing $\boldsymbol{C}_r$. We can continue by writing

$$\begin{aligned}
&\min_{\boldsymbol{C}_r} \; \frac{1}{8}\|\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_{pr}'\boldsymbol{G}'\|_F^2-\frac{1}{4}\operatorname{tr}(\widetilde{\boldsymbol{H}}_{C_r}'\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_{pr}'\boldsymbol{G}')+\frac{\lambda}{2}\|\boldsymbol{X}^*\boldsymbol{C}_r\|_{\mathrm{F}}^2\\
&\Leftrightarrow \frac{d}{d\boldsymbol{C}_r}\left(\frac{1}{8}\operatorname{tr}\left(\boldsymbol{G}\boldsymbol{D}_{pr}\boldsymbol{C}_r'\boldsymbol{X}^{*\prime}\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_{pr}'\boldsymbol{G}'\right)-\frac{1}{4}\operatorname{tr}(\widetilde{\boldsymbol{H}}_{C_r}'\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_{pr}'\boldsymbol{G}')+\operatorname{tr}\left(\boldsymbol{C}_r'\boldsymbol{X}^{*\prime}\boldsymbol{X}^*\boldsymbol{C}_r\right)\right)=0\\
&\Leftrightarrow \frac{1}{4}\boldsymbol{X}^{*\prime}\boldsymbol{X}^*\boldsymbol{C}_r\boldsymbol{D}_{pr}'\boldsymbol{G}'\boldsymbol{G}\boldsymbol{D}_{pr}-\frac{1}{4}\boldsymbol{X}^{*\prime}\widetilde{\boldsymbol{H}}_{C_r}\boldsymbol{G}\boldsymbol{D}_{pr}+\lambda\boldsymbol{X}^{*\prime}\boldsymbol{X}^*\boldsymbol{C}_r=0\\
&\Leftrightarrow \boldsymbol{X}^{*\prime}\boldsymbol{X}^*\boldsymbol{C}_r\left(\boldsymbol{D}_{pr}'\boldsymbol{G}'\boldsymbol{G}\boldsymbol{D}_{pr}+4\lambda\boldsymbol{I}\right)=\boldsymbol{X}^{*\prime}\widetilde{\boldsymbol{H}}_{C_r}\boldsymbol{G}\boldsymbol{D}_{pr}\\
&\Leftrightarrow \boldsymbol{X}^{*\prime}\boldsymbol{X}^*\boldsymbol{C}_r\left(\boldsymbol{D}_{pr}'\boldsymbol{G}'\boldsymbol{G}\boldsymbol{D}_{pr}+4\lambda\boldsymbol{I}\right)=\boldsymbol{X}^{*\prime}(\boldsymbol{Z}_{s,C_r}+\boldsymbol{X}^*\boldsymbol{C}_r^{(0)}\boldsymbol{D}_{pr}^{(0)\prime}\boldsymbol{G}')\boldsymbol{G}\boldsymbol{D}_{pr}\\
&\Leftrightarrow \boldsymbol{C}_r=(\boldsymbol{X}^{*\prime}\boldsymbol{X}^*)^{-1}\boldsymbol{X}^{*\prime}(\boldsymbol{Z}_{s,C_r}+\boldsymbol{X}^*\boldsymbol{C}_r^{(0)}\boldsymbol{D}_{pr}^{(0)\prime}\boldsymbol{G}')\boldsymbol{G}\boldsymbol{D}_{pr}\left(\boldsymbol{D}_{pr}'\boldsymbol{G}'\boldsymbol{G}\boldsymbol{D}_{pr}+4\lambda\boldsymbol{I}\right)^{-1},
\end{aligned} \tag{97}$$

where $\boldsymbol{Z}_{s,C_r}$ corresponds to the $n_x \times n_i$ block in $\boldsymbol{Z}_s$. Similar updates can be derived for $\boldsymbol{D}_r$, $\boldsymbol{C}$ and $\boldsymbol{D}$, these updates read

$$\boldsymbol{D}_r=(\boldsymbol{G}^{*\prime}\boldsymbol{G}^*)^{-1}\boldsymbol{G}^{*\prime}(\boldsymbol{Z}_{s,D_r}'+\boldsymbol{G}^*\boldsymbol{D}_r^{(0)}\boldsymbol{C}_{pr}^{(0)\prime}\boldsymbol{X}')\boldsymbol{X}\boldsymbol{C}_{pr}\left(\boldsymbol{C}_{pr}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{C}_{pr}+4\lambda\boldsymbol{I}\right)^{-1}, \tag{98}$$

$$\boldsymbol{C}=\boldsymbol{J}_{n_o}(\boldsymbol{Z}_{s,C}+\boldsymbol{\alpha}_2\boldsymbol{1}'+\boldsymbol{C}^{(0)}\boldsymbol{D}_{pr}^{(0)\prime}\boldsymbol{G}')\boldsymbol{G}\boldsymbol{D}_{pr}\left(\boldsymbol{D}_{pr}'\boldsymbol{G}'\boldsymbol{G}\boldsymbol{D}_{pr}+4\lambda\boldsymbol{I}\right)^{-1}, \tag{99}$$

$$\boldsymbol{D}=\boldsymbol{J}_{n_p}(\boldsymbol{Z}_{s,D}'+\boldsymbol{\beta}_2\boldsymbol{1}'+\boldsymbol{D}^{(0)}\boldsymbol{C}_{pr}^{(0)\prime}\boldsymbol{X}')\boldsymbol{X}\boldsymbol{C}_{pr}\left(\boldsymbol{C}_{pr}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{C}_{pr}+4\lambda\boldsymbol{I}\right)^{-1}. \tag{100}$$

We can summarize this for finding updating $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$ as well as the complete minimization by majorization algorithm below.

Algorithm 7 Updating of $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$ using block relaxation

```
**Initialize**
    $\boldsymbol{Z}_s \leftarrow \boldsymbol{Z}_s$
    $\boldsymbol{C}_r \leftarrow \boldsymbol{C}_r$
    $\boldsymbol{C} \leftarrow \boldsymbol{C}$
    $\boldsymbol{D}_r \leftarrow \boldsymbol{D}_r$
    $\boldsymbol{D} \leftarrow \boldsymbol{D}$
    $\boldsymbol{X} \leftarrow \left[\begin{array}{c|c} \boldsymbol{X}^* & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I}_x \end{array}\right]$
    $\boldsymbol{G} \leftarrow \left[\begin{array}{c|c} \boldsymbol{G}^* & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I}_g \end{array}\right]$
**repeat**
    Given $\boldsymbol{D}_r$, $\boldsymbol{C}$ and $\boldsymbol{D}$, solve for $\boldsymbol{C}_r$:
        set $\boldsymbol{C}_r = (\boldsymbol{X}^{*\prime}\boldsymbol{X}^*)^{-1}\boldsymbol{X}^{*\prime}(\boldsymbol{Z}_{s,C_r} + \boldsymbol{X}^*\boldsymbol{C}_r^{(0)}\boldsymbol{D}_{pr}^{(0)\prime}\boldsymbol{G}^\prime)\boldsymbol{G}\boldsymbol{D}_{pr}\left(\boldsymbol{D}_{pr}^\prime\boldsymbol{G}^\prime\boldsymbol{G}\boldsymbol{D}_{pr} + 4\lambda\boldsymbol{I}\right)^{-1}$
    Given $\boldsymbol{C}_r$, $\boldsymbol{C}$ and $\boldsymbol{D}$, solve for $\boldsymbol{D}_r$:
        set $\boldsymbol{D}_r = (\boldsymbol{G}^{*\prime}\boldsymbol{G}^*)^{-1}\boldsymbol{G}^{*\prime}(\boldsymbol{Z}_{s,D_r}^\prime + \boldsymbol{G}^*\boldsymbol{D}_r^{(0)}\boldsymbol{C}_{pr}^{(0)\prime}\boldsymbol{X}^\prime)\boldsymbol{X}\boldsymbol{C}_{pr}\left(\boldsymbol{C}_{pr}^\prime\boldsymbol{X}^\prime\boldsymbol{X}\boldsymbol{C}_{pr} + 4\lambda\boldsymbol{I}\right)^{-1}$
    Given $\boldsymbol{C}_r$, $\boldsymbol{D}_r$ and $\boldsymbol{D}$, solve for $\boldsymbol{C}$:
        $\boldsymbol{C} = \boldsymbol{J}_{n_o}(\boldsymbol{Z}_{s,C} + \boldsymbol{\alpha}_2\boldsymbol{1}^\prime + \boldsymbol{C}^{(0)}\boldsymbol{D}_{pr}^{(0)\prime}\boldsymbol{G}^\prime)\boldsymbol{G}\boldsymbol{D}_{pr}\left(\boldsymbol{D}_{pr}^\prime\boldsymbol{G}^\prime\boldsymbol{G}\boldsymbol{D}_{pr} + 4\lambda\boldsymbol{I}\right)^{-1}$
    Given $\boldsymbol{C}_r$, $\boldsymbol{D}_r$ and $\boldsymbol{C}$, solve for $\boldsymbol{D}$:
        $\boldsymbol{D} = \boldsymbol{J}_{n_p}(\boldsymbol{Z}_{s,D}^\prime + \boldsymbol{\beta}_2\boldsymbol{1}^\prime + \boldsymbol{D}^{(0)}\boldsymbol{C}_{pr}^{(0)\prime}\boldsymbol{X}^\prime)\boldsymbol{X}\boldsymbol{C}_{pr}\left(\boldsymbol{C}_{pr}^\prime\boldsymbol{X}^\prime\boldsymbol{X}\boldsymbol{C}_{pr} + 4\lambda\boldsymbol{I}\right)^{-1}$
**until** $\boldsymbol{C}_r$, $\boldsymbol{D}_r$, $\boldsymbol{C}$ and $\boldsymbol{D}$ have converged
**return** $\boldsymbol{C}_{pr} = \left[\begin{array}{c} \boldsymbol{C}_r \\ \hline \boldsymbol{C} \end{array}\right]$ and $\boldsymbol{D}_{pr} = \left[\begin{array}{c} \boldsymbol{D}_r \\ \hline \boldsymbol{D} \end{array}\right]$
```

Algorithm 8 `Partially restricted`: Complete partially restricted minimization by majorization algorithm

```
initialize $\boldsymbol{\alpha}_{pr}^{(0)}$, $\boldsymbol{\beta}_{pr}^{(0)}$, $\boldsymbol{C}_{pr}^{(0)}$, $\boldsymbol{D}_{pr}^{(0)}$
Column center $\boldsymbol{\beta}_1^{(0)}$, $\boldsymbol{\beta}_2^{(0)}$, $\boldsymbol{X}^*$, $\boldsymbol{G}^*$, $\boldsymbol{C}^{(0)}$, and $\boldsymbol{D}^{(0)}$
while $t = 0$ or $(\log L(\boldsymbol{\Gamma}^{(t)}|\boldsymbol{Y}) - \log L(\boldsymbol{\Gamma}^{(t-1)}|\boldsymbol{Y}))/\log L(\boldsymbol{\Gamma}^{(t-1)}|\boldsymbol{Y}) \geq \epsilon$ do
    $t \leftarrow t + 1$
    Update $\boldsymbol{H}_{pr}$: $h_{ui}^{(t)} \leftarrow 8b_{ui}^{(t-1)}$
    Update $\boldsymbol{\alpha}_1$: $\boldsymbol{\alpha}_1^{(t)} \leftarrow (n_g + n_p)^{-1}(n_g\bar{\boldsymbol{h}}_{1,1} + n_p\bar{\boldsymbol{h}}_{1,2})$
    Update $\boldsymbol{\alpha}_2$: $\boldsymbol{\alpha}_2^{(t)} \leftarrow (n_g + n_p)^{-1}(n_g\bar{\boldsymbol{h}}_{2,1} + n_p\bar{\boldsymbol{h}}_{2,2})$
    Update $\boldsymbol{\beta}_1$: $\boldsymbol{\beta}_1'^{(t)} \leftarrow (n_x + n_o)^{-1}(n_x\hat{\boldsymbol{h}}_{1,1} + n_o\hat{\boldsymbol{h}}_{2,1})$
    Update $\boldsymbol{\beta}_2$: $\boldsymbol{\beta}_2'^{(t)} \leftarrow (n_x + n_o)^{-1}(n_x\hat{\boldsymbol{h}}_{1,2} + n_o\hat{\boldsymbol{h}}_{2,2})$
    Update $\boldsymbol{C}_{pr}^{(t)}$ and $\boldsymbol{D}_{pr}^{(t)}$ using Algorithm 5, 6 or 7
    Update $\gamma_{ui}$: $\gamma_{ui}^{(t)} \leftarrow \alpha_u^{(t)} + \beta_i^{(t)} + \boldsymbol{x}_u'\boldsymbol{C}^{(t)}\boldsymbol{D}^{(t)\prime}\boldsymbol{g}_i \;\forall\; (u,i) \in \Psi$
    Compute $-\log L(\boldsymbol{\Gamma}^{(t)}|\boldsymbol{Y})$
end while
return $\boldsymbol{\alpha}_{pr}^{(t)}$, $\boldsymbol{\beta}_{pr}^{(t)}$, $\boldsymbol{C}_{pr}^{(t)}$, $\boldsymbol{D}_{pr}^{(t)}$
```

3.6 Baseline models

The models discussed in in this paper will be tested against four different baseline models. We will implement the same baseline models as discussed in de Bruin et al. (2020). These baseline models serve as elementary approximations for the probability of a user clicking on a certain item. Despite their elementary nature, the baseline models turn out to be relatively hard to beat in practice. We estimate the following four baseline models

1. Majority rule: the mode response to all observed items in the response dataset, i.e. zero for all predictions.
2. Global average click rate: $(\sum_{(u,i)\in\Psi} y_{ui})^{-1}|\Psi|$, where $|\Psi|$ is the cardinality of the set of observed user items combinations.
3. Average click rate per user: $(\sum_{(i)\in\Psi_u} y_{ui})^{-1}|\Psi_u|$, where $|\Psi_u|$ is the cardinality of the set of items seen by user u.
4. Average click rate per item: $(\sum_{u\in\Psi_i} y_{ui})^{-1}|\Psi_i|$, where $|\Psi_u|$ is the cardinality of the set of users who have seen item i.

3.7 Evaluation metrics

Appropriate evaluation metrics are important when evaluating the performance of the models. In this section we will discuss and contrast various performance measures. We define the observations in our test set as the set $\mathcal{G}$.

3.7.1 Root Mean Squared Error

The implicit feedback used in this paper is binary. A common performance measure used in this setting is the Root Mean Squared Error (RMSE). The RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{G}|} \sum_{(u,i)\in\mathcal{G}} (y_{ui} - \hat{\mu}_{ui})^2}. \tag{101}$$

The RMSE rewards the correct classification of both 0 and 1. As such, the RMSE is a good performance metric when considering binary datasets. We will use data from a large online tour operator to evaluate the performance of our models in Section 5. One of the datasets available contains implicit feedback data, we will refer to this dataset as the Sunweb dataset. This data set, and most implicit datasets alike, is very imbalanced. In the Sunweb dataset the global clickrate is only 2.19%. Simple baseline models, such as majority rule classification, might be expected to perform relatively well under these circumstances in terms of RMSE. Additionally, implicit datasets are known to be troublesome in encoding negative feedback (Johnson, 2014). The implicit feedback in the Sunweb dataset contains the clicks on items in opened emails from various email campaigns, where users are only shown a subset of all available items (those contained in the opened emails). In this context we only observe an indication that the user saw the item but did not click on the item. Not clicking the item can be due to various reasons besides him not liking the item. As the promotional emails contained multiple items we could for example think of a situation where the user would be interested in multiple items in the email but he is so satisfied with the first item he clicked on that he simply stops his search. Because of this, we might be more interested in correct classification of the clicks only.

3.7.2 Mean Percentage Ranking

We consider a second evaluation metric, which is more commonly used in the literature dealing with implicit feedback; the Mean Percentage Ranking (MPR) (see Hu et al. (2008) and Johnson (2014)). The MPR is a recall based evaluation metric which evaluates the user's satisfaction with an ordered list of recommended items. For each user in the test set we generated a ranked list of items sorted by preference. Let rank_{ui} denote the respective percentile rank of item i for user u. Thus, $\text{rank}_{ui} = 0\%$ reflects the highest ranked items for user u. Similarly, $\text{rank}_{ui} = 100\%$ reflects the least preferred item for user u. If we define y^t_{ui} as the implicit feedback of user u on item i in the test set the MPR is defined as

$$MPR = \frac{\sum_{(u,i)\in\mathcal{G}} y^t_{ui}\text{rank}_{ui}}{\sum_{(u,i)\in\mathcal{G}} y^t_{ui}}. \tag{102}$$

Lower values of the MPR are more desirable as they reflect that the user clicked on the items high on the predicted list. It should be noted that randomly produced recommendations have an expected MPR of 50%. Notice that as we evaluate the MPR per user, we cannot use the first three baseline models (majority rule, global average click rate and average click rate per user). These three models produce the same predictions for all items per user. The resulting rankings and MPR statistic is therefore identical and meaningless. For this reason we will still cover the RMSE, as this allows us to compare our models with all baseline models.

3.7.3 Precision-recall plots

As a last means of analysing the performance of the various methods we will estimate precision-recall (PRC) plots. PRC plots are shown to be very informative in the context of unbalanced binary data (Saito and Rehmsmeier, 2015). PRC curves plot the precision = $(\text{TP} + \text{FP})^{-1}(\text{TP})$ where TP are True Positives, on the y-axis against the recall = $(\text{TP} + \text{FN})^{-1}(\text{TP})$ for a gradually changing threshold value. This threshold value indicates the value above which a probabilistic prediction is deemed a click. A common issue with unbalanced datasets is that they predict a large number of true negatives. These true negatives then drown out the true positives in metrics such as the false positive rate = $(\text{TN} + \text{FP})^{-1}(\text{FP})$ used in for example ROC curves. Notice that the metrics used in the PRC curves do not use the true negatives, as such, they are only concerned with correct prediction of the minority class, the positive cases.

3.8 Additional implementation details

3.8.1 Model selection

All models discussed in Section 3 require optimized hyperparameters to maximize performance. The most important parameter which needs to be optimized is the value of λ. The second parameter is the number of factors our models use. One important characteristic of our models is the models ability to drop redundant factors if they do not help in increasing the objective function (due to implementation of the nuclear norm). As such, we do not need to optimize them to optimize the value of the objective function. Although we do not need to optimize the number of factors, we can use the factors to add additional constraints to the models to increase predictive power. To make our parameter selection as robust as possible we implement 5-fold cross validation. In 5-fold cross validation a training set is randomly divided into five different data sets. In each iteration, the model is trained using four of these sets and the performance is evaluated on the remaining dataset. Afterwards the results are averaged over these different folds. This method is known to yield more robust results compared to simply evaluating performance of the whole training set at once. We obtain both the training and the test set by splitting the full dataset at random using a 80/20 split for the training and test sets, respectively. The test set will only be used to evaluate the final performance of the models in Section 5.

3.8.2 Warm starts

The sparse plus low rank data structure also allows us to easily incorporate warm starts. This implementation is especially helpful during the cross validation of the models. Recall that the `Unrestricted` model described in Section 3.3 is formulated as $\boldsymbol{\Gamma} = \boldsymbol{\alpha}\mathbf{1}' + \mathbf{1}\boldsymbol{\beta}' + \boldsymbol{C}\boldsymbol{D}'$. For each fold in the cross validation procedure we estimate the model parameters corresponding to the first set of hyperparameters, in the model above these would be $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{C}$ and $\boldsymbol{D}$. These converged parameters are used as an initial estimates for the next set of hyperparameters. As we implemented a majorization approach in all of our methods, which guarantees decrease of the objective function in each iteration, the objective function either decreases further than the previous set of hyperparameters or fails to formulate an additional decrease and stops. The first iteration was performed using a very large value for λ. With a very large value of lambda, essentially all factors in $\boldsymbol{C}$ and $\boldsymbol{D}$ are reduced to zero. We slowly lowered the value for λ, releasing more factors in the process.

4 Timing experiments

In this section we will compare and contrast the models introduced in Section 3 in terms of their speed. As these models are novel in the type of data to which they are applied, we have no other known models to compare the performance of our models to. One of the most important characteristics of our models is their ability to scale. Therefore, we compare our models in two different settings and compare how they scale in these circumstances.

4.1 Simulation setup

Each γ_{ui} is simulated using the data generating process

$$\gamma_{ui} = \alpha_u + \beta_i + \boldsymbol{x}_u'\boldsymbol{C}\boldsymbol{D}'\boldsymbol{g}_i + \varepsilon_{ui}, \tag{103}$$

where ε_{ui} is a random error with mean zero. The probabilities for each user item combination are calculated using a simple logistic function

$$p_{ui} = \frac{e^{\gamma_{ui}}}{1 + e^{\gamma_{ui}}}. \tag{104}$$

Afterwards we transform this probability to implicit feedback with the simple rule

$$y_{ui} = \begin{cases} 1, & \text{if } p_{ui} \geq 0.5 \\ 0, & \text{otherwise.} \end{cases} \tag{105}$$

A distribution of the probabilities can be found in Figure 1. We can see that the probabilities reflect a small number of items in which the users are interested and a much larger fraction of items in which the users are not interested.

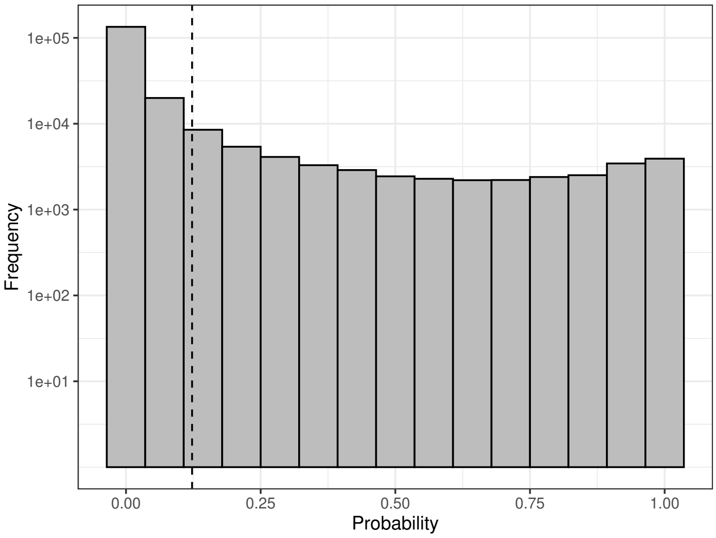


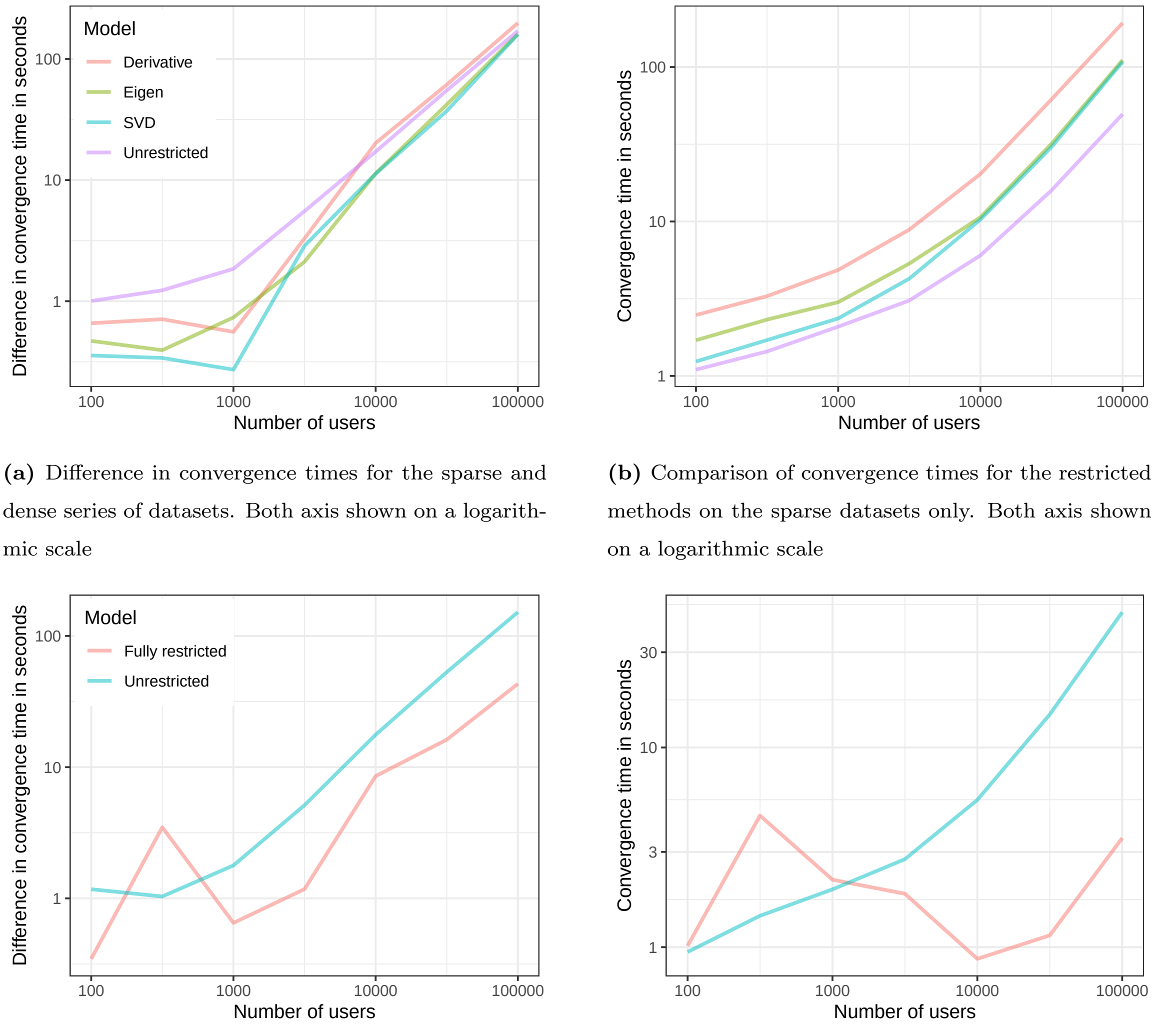
Figure 1: Frequency bar chart on a logarithmic scale of the probabilities with the mean as a dotted line

Five different models are compared, of which the last three all represent the `Partially restricted` model:

1. `Unrestricted` - Algorithm 2
2. `Fully restricted` - Algorithm 4
3. `SVD` - Algorithm 8 using Algorithm 5 to update the $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$
4. `Eigen` - Algorithm 8 using Algorithm 6 to update the $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$
5. `Derivative` - Algorithm 8 using Algorithm 7 to update the $\boldsymbol{C}_{pr}$ and $\boldsymbol{D}_{pr}$

4.2 Timing experiments

In the following timing experiments we will investigate the speed and scalability of our algorithms. The methods proposed in Section 3 all implement a sparse plus low rank data structure. This data structure allows us to efficiently work with sparse binary data sets. The Sunweb dataset has three classes for example, unobserved, no click, and click. More traditional implicit recommender systems would only consider the click, and consider the remaining observations as no click. Therefore, we generate two series of datasets, varying in size. In the first series we set the sparsity of the user item matrix (δ) to 5%. This implies that only 5% of the user item matrix is filled with feedback data which is similar to the Sunweb dataset, where only 6.2% of the data is observed. In the second series we set the sparsity to 100%. We will refer to these series of datasets as the sparse and dense series, respectively. To make our results more robust we average the timing results over five different runs. Figure 2 shows the average timing results on the two different sets of data. The remaining parameters of the simulation, the size in user (u) and items (i), factors of the data generating process (factors), factors used in the methods (f), percentage of the users and items we have additional information for (ζ) and the value of the restrictive parameter λ are fixed across the models. A model is said to have converged once the change in it's likelihood over the iteration falls under a certain threshold ϵ. Note that we only compare `Unrestricted` with `Fully restricted` in the case in which we have additional information on all users and items ($\zeta = 1$).



(a) Difference in convergence times for the sparse and dense series of datasets. Both axis shown on a logarithmic scale

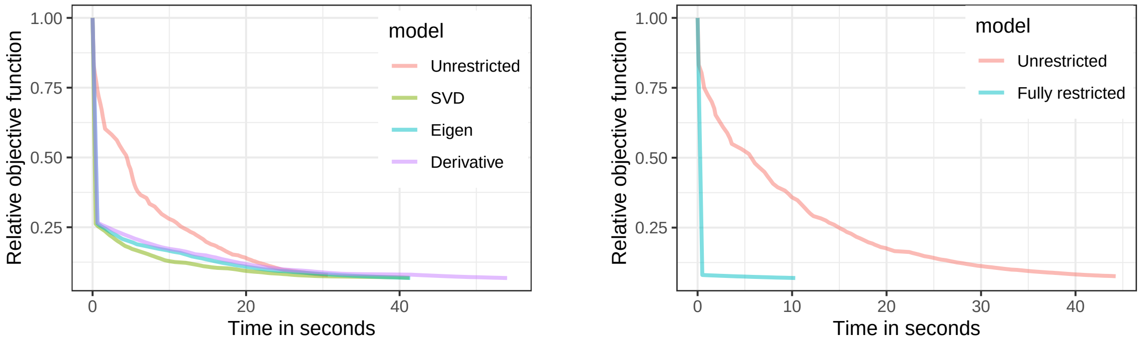
(b) Comparison of convergence times for the restricted methods on the sparse datasets only. Both axis shown on a logarithmic scale

(c) Difference in convergence times for the sparse and dense series of datasets. Both axis shown on a logarithmic scale

(d) Comparison of convergence times for the fully restricted methods on the sparse datasets only. Both axis shown on a logarithmic scale

Figure 2: Four timing experiments. The following parameters are fixed across the simulations: $i = 100$, factors$= 5$, $f = 5$ and $\lambda = 0.5$.

Panel (a) from Figure 2 demonstrates the scalability of our models compared to the binary models. We observe that generally, an increase in the number of users corresponds to an increase in the difference in convergence times for all methods. In this experiment, the size of the dataset varies from 100 to 100.000 users with 100 items. The size of these datasets is tiny compared to the size of many recommender problems in practice. Thus, the fact that we can already demonstrate significantly lower convergence times for the sparse binary setting is an important finding. Panel (b) plots the averaged convergence times for all partially restricted methods on the sparse datasets against one another. The panel demonstrates that for all dataset sizes, the `Derivative` model required the most amount of time to converge. When we compare `SVD` and `Eigen` we do not observe any significant difference in convergence times for the larger datasets and we can conclude that for larger datasets, there does not seem to be an added benefit in terms of convergence time due to using an eigen decomposition. Panel (c) and (d) from Figure 2 show similar experiments applied to the fully restricted methods, when all additional data is available. Panel (c) shows that again, the fully restricted method also follows the pattern observed with the partially restricted methods. Interestingly, the computation time in panel (d) seems to break the pattern observed in panel (b) for `Fully restricted`. The convergence time is influenced by both the number of iterations needed and the time per iteration. On closer inspection it seems that the number of iterations vary significantly for the `Fully restricted`. The data generating process of the simulation is perfectly aligned with the restricted models. As such, when more additional information is available, the restricted methods are able to make a greater initial jump towards convergence. Figure 3 shows this characteristic for two different amounts of additional information. Since the fully restricted method has all additional information available it is able to converge after only a handful of iterations for the large datasets, resulting in small convergence times.



(a) Example of convergence for 75% available additional information

(b) Example of convergence when all additional information is available

Figure 3: Two examples of objective function convergence for varying amounts of available additional information

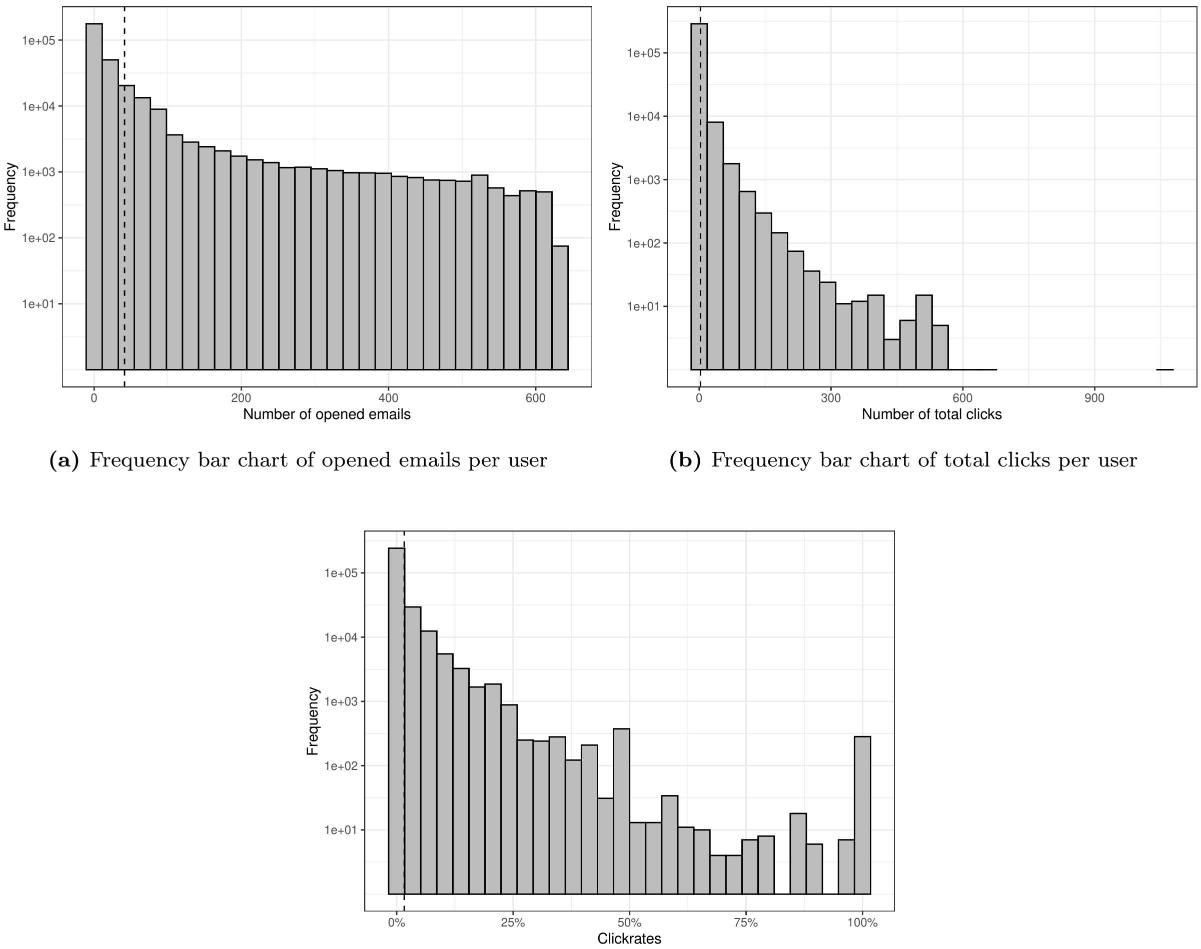
5 Empirical application

In this section we will apply both the unrestricted and the restricted methods on the Sunweb dataset. As `SVD`, `Eigen` and `Derivative` all minimize the same objective function and only differ in their optimization approach it makes little sense to estimate all three models. Therefore, we only estimate the `SVD` in the partially restricted method class as it performed the best in terms of speed in Section 4.

5.1 Data description

First we will analyse the available datasets. In total we have three datasets available. The first dataset, which we will refer to as the click dataset, contains the data on the click behaviour of the customers of a large online tour operator, Sunweb. Sunweb sends out weekly emails containing items which might be of interest to the customers. The click dataset collects the responses of these customers to the given items and proxies whether the customer is interested in the item, by clicking on the item, or is not interested in the item, by not clicking on the item. It is important to note that the dataset only contains opened emails. Our second and third datasets contain additional information on both the users and the items. As such, we will refer to these datasets as the user and item datasets, respectively. Our click dataset contains 40.216.265 observations with 12.347.588 unique mails send. The dataset contains 299.248 unique users and 2.199 unique items. This means that our feedback is highly sparse, with only 6.1% of all user-item combinations being observed. The click-through rate of the individual emails is 6.2% implying that for only a small fraction of emails the user is directed to the Sunweb website. On average each email contained 3.26 items. If we consider the number of total items and clicks we find a global click rate[2] of 2.19%. Figure 4 shows frequency bar charts of various statistics relating to the users. Panel (a) shows that on average, each user in our sample opened roughly 41 emails, with a median of 6. Some users opened more than 600 emails in roughly a two year period. Panel (b) shows that on average, each user recorded 2.95 clicks, which is tiny in comparison to the number of opened emails, with a median of 0. In fact, the vast majority of users (70.43%) did not click on a single item. We do observe some outliers with over 600 clicks. These individuals might be simply clicking on every item or simply like browsing for holidays. If we look at the clickrates in panel (c) we find that on average each individual has a clickrate of 1.62%. logically, a large group of users has a clickrate of 0%. We also observe a small group with a clickrate of 100%. Our models are unable to calculate predictions for these groups as they lack variation in the response variable. We will therefore predict zeros for all items for users with a clickrate equal to zero, and ones for all items for users with a clickrate equal to one.

[2] $(\text{Total clicks}) * (\text{Total observations})^{-1}$



(a) Frequency bar chart of opened emails per user

(b) Frequency bar chart of total clicks per user

(c) Frequency bar chart of clickrates per user

Figure 4: Frequency bar charts on a logarithmic scale of various statistics with the means as a dotted lines

An overview of the variables available in the item dataset can be found in Table 1. These characteristics are available for 1.713 out of the 2.199 unique items. The mean of the discounted price is 506.40 euros whilst the original price has a mean of 700.2 euros. This means that on average, the items have a 27.7% discount. The average star rating of the items is 3.87. The most popular holiday destination in the email items is Greece.

Table 1: Overview of item characteristics.

REVIEW_RATING	The rating of the accommodation on a scale of 1-10.
DURATION	Length of the holiday.
PRICE	Current price of the item.
PRICE_ORIGINAL	Original price of the item.
STAR_RATING	Star rating of the accommodation.
CHILDREN	Whether there are beds for children in the room
ACCO_FULL_WIDTH	Whether the layout in the email was full or half width.
item_POSITION	Position of the item in the email.
DESTINATION	Destination of the holiday.
YEAR	Year when the item was send.
QUARTER	Quarter when the item was send.
MAX_PERSON	Maximum amount of adults.
DEP_MONTH	Month of departure.
MEALPLAN	Meal plan of the item (all inclusive, full pension etc).

Unfortunately we do not have much user information available. The available user information is extracted from the website behaviour of the users on the Sunweb website. The raw data contains the website browsing behaviour from 71.705 anonymized users in our click dataset. The data contains information on the time of the sessions, which holiday destinations the user was browsing for and the layer of the ordering process. We assume that these layers reflect the interest of the user in the item as navigating through the layers requires more time and clicks. For example, we can infer that the user is more interested in the item if he is already browsing dates and prices compared to the situation where he is simply viewing the item. We can rank all available layers corresponding to the stage in the ordering process. We refer to these layers as webpages, with the most interest shown in the "deepest" webpage. Each user may have browsed the website on various occasions. The final variables we extracted from this dataset are shown in Table 2. The average user visited the Sunweb website on 16.24 different occasions.

Table 2: Overview of user characteristics.

MOST_RECENT_WEBPAGE	Deepest webpage in the most recent session.
MOST_RECENT_COUNTRY	Destination of the most interesting item in the most recent session.
MOST_INTEREST_WEBPAGE	Deepest webpage over all sessions
MOST_INTEREST_COUNTRY	Destination corresponding to the most interest.
MOST_RECENT_YEAR	Year of the most recent session
MOST_RECENT_QUARTER	Quarter of the most recent session
SESSIONS	Number of total sessions of the user.

5.2 Results

First we need to optimize the hyperparameters of both `Unrestricted` and `SVD`. We implemented the cross-validation procedure discussed in Section 3.8 on $\lambda = \{250, 100, 50, 25, 20, 15, 10, 8, 6, 5, 4, 3, 2, 1, 0.5\}$. Five-fold cross-validation yields $\lambda = 25$ for `Unrestricted`, for which 5 factors were retained. For `SVD` we find $\lambda = 8$, and only retain 2 factors.

Table 3: Test RMSE of baseline and novel models

	Majority rule	Global average	User average	Item average	Unrestricted	SVD
RMSE	0.1481	0.1465	0.1403	0.1455	0.1372	0.1389

Table 3 shows the RMSE results of `Unrestriced`, `SVD` and the baseline models. As we can see, all baseline models perform moderately well compared to our methods. Although we should keep in mind that small improvements in terms of RMSE can have huge value for businesses. In the Netflix prize, the winning team managed to beat Netflix's RMSE by only 10% and earned one million dollars. The best baseline method is the average click rate per user with a RMSE of 0.1403. Our models manage to beat the baseline predictions by 2.2 and 1.0 percent for `Unrestriced` and `SVD` respectively.

Table 4: Test MPR of baseline and novel models

	Item average	Unrestricted	SVD
MPR	0.3468	0.3085	0.3371

Table 4 shows the MPR results of `Unrestriced`, `SVD` and the average click rate per item. MPR results for the other baseline models are meaningless, as described in Section 3.7. All models beat the random prediction threshold of 0.5. Again, both our models also manage to beat the baseline model with `Unrestriced` performing the best, followed by `SVD`.

Lastly, we look at the results from the PRC plot shown in Figure 5 as well as the AUC values in Table 4. As precision can be interpreted as the models ability to correctly predict the click while recall refers to the percentage of total clicks correctly classified. If we view the plot from left to right, the prediction threshold, the value above which a prediction is classified as a click, is moved from zero to one. In the top left of the figure we thus simply predict everything as a click (as zero is the threshold) in that case we see that we have a precision of one, we predicted every click correctly. However our recall rate is close to zero since we classified all no clicks wrongly as well. A perfect algorithm hugs the top right and has an Area Under the Curve (AUC) of 1. The AUC is often used to quantify the results of the PRC plot. In both Figure 5 and Table 4 we dropped the results from the majority rule and the global average click rate baseline models as they did not add any additional information to our analysis. Once again, both our models outperform the baseline models, with `Unrestricted` performing the best.

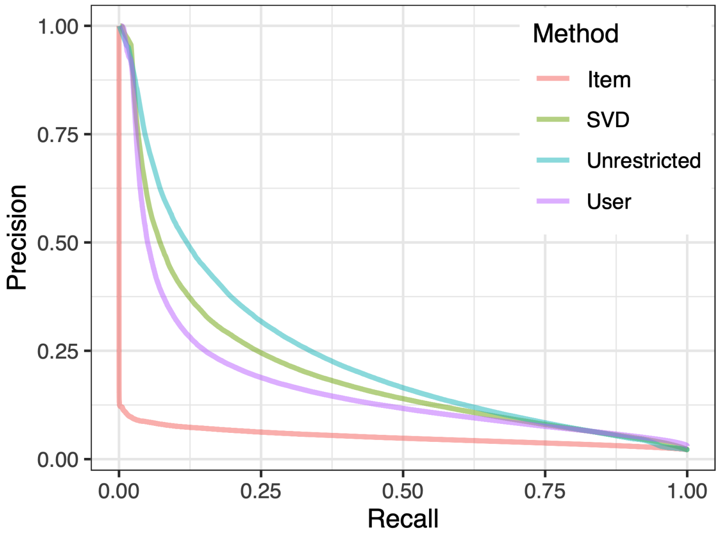


Figure 5: Precision-recall plots of four selected methods

Table 5: Area Under the Curve (AUC) of four selected methods

	Item average	User average	Unrestricted	SVD
AUC	0.051	0.170	0.239	0.201

`Unrestricted` was able to outperform `SVD` in all three cases. Nonetheless, we should be careful in concluding that the `Unrestricted` is a better method than `SVD`. Firstly, we cross validated our results over a set of roughly 20 values of λ and 5 different factors. Whilst `Unrestricted` performed best in this subset of parameters further parameter tuning might lead to increased performance of `SVD`. Secondly, as we saw in the simulation example, all restricted methods were able decrease their objective functions significantly when more additional data was available. In this dataset, we only had additional information on roughly 25% of users. Additionally, this data was unfortunately not very informative as it consisted of a hand full of dummy variables. More and better additional information about the users themselves could also lead to increases in performance. Nonetheless, we can conclude that both `Unrestricted` and `SVD` were able to outperform the baseline models.

6 Conclusion and discussion

In this paper, we tried to answer the research question: *Can we incorporate additional information in a sparse binary probabilistic matrix factorization algorithm using the sparse plus low rank data structure?* In total we derived and tested three models: `Unrestricted`, `Fully restricted` and `Partially restricted`. The first model was introduced in de Bruin et al. (2020) and exploits one source of information, in this paper we used click data from a mailing list of a large online tour operator. The second model exploits three sources of information: click data, and additional information on all users and items. `Partially restricted` differs from `Fully restricted` in that the model only requires additional information on a subset of users and items. All models are based on the principle of majorization using simple convex functions. This results in a few useful characteristics of our models. First, all models guarantee a decrease in the objective function in each iteration. This means that the path to convergence is smooth and dealing with convergence issues is rare when setting an adequate convergence parameter. This characteristic also allows for implementation of warm starts, where parameter estimates of a different set of hyperparameters are used as a starting point in the next set of hyperparameters. These warm starts drastically increasing the speed of the cross validation procedure.

We compared the speed and scalability of our sparse binary algorithms to a more traditional binary setting in a simulation study. In a series of timing experiments we showed that all three models generally require significantly less computation time to converge for our binary, highly sparse, datasets compared to the binary dataset. The difference in convergence times also seems to increase with the size of the dataset. This is an important results as it shows the power of our models, in terms of efficiency, as most real-life datasets are highly sparse.

In an empirical application we applied both the `Unrestricted` and `Partially restricted` models on data from a large online tour operator. We found that beating the baseline models was a difficult task. In terms of RMSE, we found that `Unrestricted` and `Partially restricted` managed to beat the best performing baseline by 2.2 and 1.0 percent respectively. Additionally, we considered the mean percentage ranking (MPR), an evaluation metric commonly used when dealing with implicit feedback. With MPR values of 0.3085 and 0.3371 both models significantly beat the random prediction threshold of 0.5 as well as the baseline of 0.3468. Based on the precision-recall plot we also conclude that both models beat the baseline. We cannot conclude that either `Unrestricted` or `Partially restricted` is the best model as both models are dependent on hyperparameter tuning and available data. We can, however, conclude that both models outperform the baseline models.

One of the main limitations of our model, and many collaborative recommender systems alike, is the cold start problem where predictions cannot be formulated for new users and items. In this paper, we chose to simply use non-personalized predictions by implementing the global click rate. Other approaches exist, such as using a content-based filtering recommender for the new users and items. Implementation of these content-based filtering techniques would be essential if one would wish to deal with these cold start problems.

We have shown that our models perform well on large datasets although implementation on industry sized datasets might require more advanced computing techniques such as parallelization. Other latent factor recommender systems implement sharding techniques to partition the problem into smaller pieces and run them in parallel (Johnson (2014), Das et al. (2007)). Additional research is required to investigate possible implementation of both the content-based recommender systems and parallelization as implementation of these techniques could yield better results, both in terms of performance and speed.

References

Abdi, H. (2007). Singular value decomposition (svd) and generalized singular value decomposition. *Encyclopedia of measurement and statistics*, pages 907–912.

Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York.

Borg, I. and Groenen, P. J. F. (2005). *Modern multidimensional scaling: Theory and applications.* Springer Science & Business Media.

Das, A. S., Datar, M., Garg, A., and Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280.

de Bruin, T., Huliselan, C., Lin, S., and Váradi, M. (2020). Logistic matrix factorization with minimization by majorization: a sparse plus low-rank approach. *Seminar, Erasmus University Rotterdam.*

de Leeuw, J. and Lange, K. (2009). Sharp quadratic majorization in one dimension. *Computational statistics & data analysis*, 53(7):2471–2484.

Gopalan, P., Hofman, J. M., and Blei, D. M. (2013). Scalable recommendation with poisson factorization. *arXiv preprint arXiv:1311.1704.*

Groenen, P. J. F., Giaquinto, P., and Kiers, H. A. L. (2003). Weighted majorization algorithms for weighted least squares decomposition models. Technical report.

Hastie, T., Mazumder, R., Lee, J. D., and Zadeh, R. (2015). Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee.

Johnson, C. C. (2014). Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems*, 27.

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

Kristof, W. (1970). A theorem on the trace of certain matrix products and some applications. *Journal of Mathematical Psychology*, 7(3):515–530.

Lops, P., De Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322.

Mnih, A. and Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264.

Mnih, A. and Teh, Y. (2012). Learning label trees for probabilistic modelling of implicit feedback. *Advances in Neural Information Processing Systems*, 25:2816–2824.

Ricci, F., Kantor, P. B., Rokach, L., and Shapira, B. (2011). Recommender systems handbook.

Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.