

REGULARIZED LOW RANK APPROXIMATION OF WEIGHTED DATA SETS

SAPTARSHI DAS * AND ARNOLD NEUMAIER †

Abstract.

In this paper we propose a fast and accurate method for computing a regularized low rank approximation of a weighted data set. Unlike the non-weighted case, the optimization problem posed to obtain a low rank approximation for weighted data may have local minima. To alleviate the problem with local minima, and consequently to obtain a meaningful solution, we use a priori information about the data set, and construct a regularized solution. In this paper, for illustration we use a priori information that the low rank approximants are smooth. How to use some other types of a priori information is also discussed. We use an iterative method to estimate left and right approximants. Within each iteration, we estimate the right (left) approximants by posing a constrained weighted least squares problem. Unfortunately, the variables of the quadratic optimization problem are not separable. We exploit the structure and sparsity of the associated matrices to design a tractable algorithm. The proposed method has a potential use in various applied problems, e.g. 3D image reconstruction, background correction in 2D chromatography, background correction of astronomical images. We illustrate the proposed method by reconstructing background of an astronomical image observed in the optical wavelength range. The algorithm is fast, i.e. the number of flops per iteration is of the order of the number of data points, and the memory requirement is negligible compared with the input data storage. We provide extensive numerical results to highlight different features of the method.

Key words. Low rank approximation, missing data, regularized approximation.

AMS subject classifications. 15A18, 65F30, 62H25, 62H35

1. Introduction.

1.1. Organization. This paper is organized as follows. In this section we present the motivation behind this work, our contributions, potential applications of the proposed method, and previous work done by other researchers. Our proposed method is described in detail in Section 2. Results of our experiments are presented in Section 3. Finally, our conclusions are presented in Section 4.

The algorithms in this paper are presented to demonstrate the step by step operations, and to make all the fundamental mathematical operations explicit. A direct implementation of the algorithms as presented might be sub-optimal in terms of memory. To avoid such problems one should make sure that matrix-transpose-vector-product, and matrix-vector-product-transpose routines are implemented in an memory efficient way.

A MATLAB implementation of the proposed algorithm, and the high resolution images presented in this paper can be found online at the link: [20].

1.2. Motivation. The following image processing problem is the main motivation behind this work.

APPLICATION 1.1. *Wide field astronomical images observed in the optical wavelength range have typically significant structures in the background of the image, varying smoothly over the field of view. Such structures, also known as gradients, occur mostly because of the light passing through the atmosphere, a halo of a bright object*

*Faculty of Mathematics, University of Vienna (saptarshi.das@univie.ac.at). Supported by the Federal Ministry of Science and Research, Austria, in the framework of the Austrian in-kind contribution to the European Southern Observatory (ESO).

†Faculty of Mathematics, University of Vienna. (arnold.neumaier@univie.ac.at)

in the vicinity of the field of view, or unwanted scattering in the light pass through the telescope. For a precise photometry of a celestial object, the observed image needs to be free from such unwanted structures in the background.

The state-of-the-art method to overcome the effect of the background structure, described in Application 1.1, works as follows. First, the bright objects are systematically identified from the image pixel intensity data. This information is used to create a mask (binary weights), to differentiate pixels into the pure background part and bright objects. Then, a low degree polynomial in two variables is fitted to the pixel intensity data corresponding to the sky background, using the least squares technique. Finally, the background is reconstructed as a polynomial, and subtracted from the observed image pixel intensity data. This procedure is implemented in several astronomical image reduction software packages, like IRAF: *Image Reduction and Analysis Facility* [12], (`imsurfit` routine); IRIS: *An Astronomical Image Processing Software* [3], (`remove gradient` procedure); some instrument specific data reduction pipelines of ESO also have this facility.

We observe the Runge phenomenon while using the basis of polynomials to model the background. In order to mitigate the Runge phenomenon, we have sought a better basis to represent the data. As a result, the image processing problem reduces to a low rank approximation problem with weighted data set. However, the low rank approximation problem with weighted data set admits multiple locally optimal solutions. In order to obtain a meaningful solution, we regularize the problem using a priori information regarding the smoothness of such backgrounds. A Comparison of the proposed method with the state-of-the-art method using the basis of polynomials is provided in Section 3.

Best low rank approximations of a data set (matrix) have several applications in the field of image processing, signal processing, data analysis, see [23, 10] for more examples. The problem of finding, for some natural number R , a rank R approximation of a matrix \mathbf{A} of size $M \times N$, which is the best in terms of Frobenius norm, is generally formulated as:

$$[\mathbf{u}_r, \sigma_r, \mathbf{v}_r]_{r=1}^R = \operatorname{argmin}_{\mathbf{u}_r, \sigma_r, \mathbf{v}_r} \left\| \mathbf{A} - \sum_{r=1}^R \sigma_r \mathbf{u}_r \mathbf{v}_r^* \right\|_F^2, \quad (1.1)$$

or equivalently,

$$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \operatorname{argmin}_{\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}} \left\| \mathbf{A} - \mathbf{U} \mathbf{\Sigma} \mathbf{V}^* \right\|_F^2, \quad (1.2)$$

where $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_R]$ is an $M \times R$ matrix, $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_R]$ is an $M \times R$ matrix, and $\mathbf{\Sigma}$ is a diagonal matrix of size $R \times R$ with the σ_r as diagonal entries. Here $\| \cdot \|_F$ denotes the Frobenius norm. The columns of \mathbf{U} and \mathbf{V} are normalized to have norm one, otherwise the matrix $\mathbf{\Sigma}$ can be adjusted by scaling either the columns of \mathbf{U} or \mathbf{V} . We call the columns of \mathbf{U} and \mathbf{V} the left and right approximants, respectively. They are also known as the left and right singular vectors, respectively, or principal components. We note that, typically $R \ll M, N$.

The solution to this problem is obtained by means of the Singular Value Decomposition (SVD) of the data matrix \mathbf{A} , see Theorem 5.8 in [24]. There are stable algorithms available for SVD, see [7, 24]. However, the standard algorithms for computing SVD have certain limitations, some of which are enumerated below.

1. The standard algorithms for SVD are not applicable if the data set is incomplete. The *netflix problem* data set [1] is one such example.

2. If each entry of the data matrix has an associated weight, the Frobenius norm is not an appropriate measure of closeness, and the standard algorithms for SVD are not applicable. See [22] for a data set with weighted data points.
3. The standard algorithms for SVD do not use any a priori information about the singular vectors. For example, it might be known a priori that the low rank approximation is smooth, like background of images. Another reasonable a priori assumption is that the approximants are sparse, see [21].
4. With the addition of new data points, the standard SVD does not have any fast mechanism to update the existing singular values and vectors. Subspace tracking in signal processing is one prominent application where the SVD must be updated as data arrives online, see [28].

In this paper, we address the first three deficiencies of the standard SVD algorithms. More precisely, we intend to compute a regularized low rank approximation of a weighted data matrix. We note that data sets with missing values are special case of data sets with weights, in which the weights have binary values (0 if missing, and 1 if available).

1.3. Contributions. For a natural number R , the problem of finding the best rank R approximation of a data set \mathbf{A} of size $M \times N$, with non-negative weights corresponding to each entry of \mathbf{A} provided in another matrix \mathbf{W} is posed as follows:

$$\begin{aligned}
[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] &= \operatorname{argmin}_{\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}} \|\mathbf{A} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_W^2 \\
&= \operatorname{argmin}_{\mathbf{u}_r, \mathbf{v}_r, \sigma_r} \|\mathbf{A} - \sum_{r=1}^R \sigma_r \mathbf{u}_r \mathbf{v}_r^*\|_W^2 \\
&= \operatorname{argmin}_{\mathbf{u}_r, \mathbf{v}_r, \sigma_r} \sum_{i=1}^M \sum_{j=1}^N \mathbf{W}(i, j) \left(\mathbf{A}(i, j) - \sum_{r=1}^R \sigma_r \mathbf{u}_r(i) \bar{\mathbf{v}}_r(j) \right)^2. \quad (1.3)
\end{aligned}$$

Here, $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_R]$ and $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_R]$ are $M \times R$ and $N \times R$ rank R matrices, respectively, and $\mathbf{\Sigma}$ is a diagonal matrix with $\mathbf{\Sigma}(i, i) = \sigma_i$. We note that the case of missing data is handled in the same framework by setting an entry of \mathbf{W} to 1 if the corresponding data point is available, or 0 if it is not available.

The problem posed in Equation (1.3) admits several locally optimal solutions, see [22]. In order to alleviate the problem of local minima, and to obtain a meaningful solution, we use a priori information regarding the approximants whenever such information is available.

In many cases, a priori knowledge about the data, or the approximants are known. For example, in the case of images, smoothness is frequently a very suitable a priori assumption. The classical SVD algorithm cannot take such a priori information into account. If a priori information about the approximants is known, then the problem of finding a low rank (say R) approximation is posed as follows:

$$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \operatorname{argmin}_{\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}} \left\{ \|\mathbf{A} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_W^2 + \frac{\alpha_u}{2} \|\mathbf{B}_u \mathbf{U}\|_F^2 + \frac{\alpha_v}{2} \|\mathbf{B}_v \mathbf{V}\|_F^2 \right\}, \quad (1.4)$$

where \mathbf{U} , and \mathbf{V} are $M \times R$ and $N \times R$ rank R matrices, respectively. Here $\alpha_u, \alpha_v \geq 0$ are parameters used to adjust the level of regularization, \mathbf{B}_u , and \mathbf{B}_v are suitable linear operators reflecting a priori information. In particular, if a priori information about the smoothness of the approximants is available, then \mathbf{B}_u , and \mathbf{B}_v can be set to a second order finite difference matrices. We note that, the problem posed in the

Equation (1.4) is limited to a certain type of a priori information. For example, the Equation (1.4) is improper if the a priori information is regarding the sparsity of the approximants.

Our main contribution in this paper is a fast algorithm to obtain a regularized low rank approximation of a data set with weighted (or missing) data points, i.e. a fast algorithm to solve the problem posed in Equation (1.4). We regularize the approximants using a priori information that they are smooth. Thus we use a roughness penalty in the cost function (1.4). Smoothness is a very suitable a priori assumption for a wide class of data, including images. How to use other types of a priori information in the same framework is also discussed. We note that using just the norm penalty is trivial, and leads to a shrinkage estimator.

As described in Section 2, and also noted in [16, 2], unlike the norm penalty, the roughness penalty, or other penalties, makes the variables of the problem inseparable. We exploit the structure and sparsity of the associated matrices to achieve a tractable algorithm. Moreover, with the proposed method, it is possible to regularize different singular vectors (approximants) with different severity of regularization.

Instead of regularizing by means of penalizing the approximants \mathbf{U} and \mathbf{V} as demonstrated in Equation (1.4), a possible alternative might be to penalize the approximation $\mathbf{U}\Sigma\mathbf{V}^*$. With penalty on the approximation, we have found that the variables become inseparable in a different way. Consequently, we are unable to devise a tractable algorithm for solving the low rank approximation problem using a penalty on the approximate $\mathbf{U}\Sigma\mathbf{V}^*$.

1.4. Applications. Weighted data sets are obtained whenever the data points are collected with different level of accuracies, or from different sources etc... Data sets with missing data points are special case of a weighted data set. Apart from Application 1.1, which is the main motivation for this work, potential applications of the proposed work includes the following.

APPLICATION 1.2. *Digital images obtained from two dimensional gas or liquid chromatography have a significant background varying smoothly over the whole image. For a precise analysis of the compounds of interest, the background should be removed. For details on how such backgrounds are produced, and the state of the art method for its correction, see [17, 11].*

APPLICATION 1.3. *Many problems in computer vision are posed as a low rank approximation problem, like 3D reconstruction, or face recognition. In case of 3D reconstruction from several 2D images observed at different angles, the problem of low rank approximation with missing data arises, see [8, 2] for further details. In this case, important a priori information about the smoothness of the final image is usually available.*

APPLICATION 1.4. *Spectral estimation has several applications, like the direction of arrival (DOA) estimation. Several methods for spectral estimation rely on a low rank approximation, e.g. the MUSIC algorithm. The spectral estimation problem with missing data arises frequently for a wide range of applications, see [25]. Frequently, a priori information about the spectral concentration of the signals is known.*

We illustrate our new approach using astronomical image data for Application 1.1. As we shall see in Section 3, the methods proposed in this paper considerably improve the quality of the reconstructed images, and also alleviate the problem with local minima.

1.5. Previous Work. In 1970, Christofferson considered the problem of computing a rank one ($R = 1$) approximation of a data set with missing values, see [4]. In

1974, Ruhe described a method to compute a low rank $R \geq 1$ approximation of a data set with missing values [19]. The method of Ruhe computes the low rank approximation in a recursive fashion, by repeatedly computing rank one approximation of the data set, and then updates the data set by subtracting the rank one approximation. Ruhe also described a way to handle the case of weighted data, by making it a special case of missing data. The approach of Ruhe for weighted data is counter intuitive. Moreover, this approach of handling weighted data enormously expands the dimension of the problem. Around the same time, in 1969, Wold, and Lyttkens described an iterative method, known as NIPALS, for computing low rank approximations of data with equal weights, see [27]. However, like power iteration methods, NIPALS is inadequate for data sets with close singular values, for a detailed analysis see [5]. NIPALS was initially intended as an alternative to SVD for principal components analysis (PCA) of unweighted data set. Motivated by NIPALS, Gabriel, and Zamir, proposed a method in 1979, to compute a low rank approximation of weighted data sets, see [6]. Unlike the method of Ruhe [19], the method by Gabriel, and Zamir [6] computes all the low rank approximants together. This approach has an advantage of preserving the orthogonality between vectors of the left approximants, and the same with right approximants. Below we outline the method by Gabriel, and Zamir, which is an algorithm to solve the problem posed in Equation (1.3). We call this method *bi-iterative singular value decomposition* with weighted data. The relevance of the name is reflected in the description of the method, and we call the routine `biSVD_weighted`.

For the purpose of reference, now we describe a method for estimating low rank approximation of a weighted data set, without any regularization, i.e. as formulated in Equation 1.3 The presentation is similar to the one in [6]. The method starts with an initial approximation \mathbf{U}_0 of the left approximants, $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_R)$. The matrix \mathbf{U}_0 can be any random isometric matrix, i.e., $\mathbf{U}_0^* \mathbf{U}_0 = \mathbf{I}_R$. In successive iterations, the right approximants $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_R)$, are updated using the last estimate of \mathbf{U} , and vice-versa. We note that, one can also begin with an initial approximation \mathbf{V}_0 of \mathbf{V} and start the iterations. In the i th step, for computing \mathbf{V}_i given \mathbf{U}_{i-1} , the problem posed in Equation (1.3) reduces to the following problem:

$$\begin{aligned} \tilde{\mathbf{Y}} &= \underset{\mathbf{Y}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{U}_{(i-1)} \mathbf{Y}^*\|_W^2. \\ &= \underset{\mathbf{Y}}{\operatorname{argmin}} \sum_{m=1}^M \sum_{n=1}^N \mathbf{W}(m, n) \left(\mathbf{A}(m, n) - \sum_{r=1}^R \mathbf{U}_{(i-1)}(m, r) \mathbf{Y}^*(r, n) \right)^2, \\ &= \underset{\mathbf{Y}}{\operatorname{argmin}} F(\mathbf{Y}). \end{aligned} \quad (1.5)$$

Next, the normal equations of the quadratic optimization problem posed in Equation (1.5) are obtained by setting $\frac{\partial F}{\partial \mathbf{Y}}$ to 0. By separating each of the N columns of \mathbf{A} , the normal equations can be written as:

$$\mathbf{U}_{(i-1)}^* \operatorname{diag}[\mathbf{W}(:, n)] \mathbf{U}_{(i-1)} \mathbf{Y}^*(:, n) = \mathbf{U}_{(i-1)}^* \operatorname{diag}[\mathbf{W}(:, n)] \mathbf{A}(:, n), \quad (1.6)$$

where $n = 1, \dots, N$. The notation $\mathbf{A}(:, n)$, and $\mathbf{W}(:, n)$ represent the n th column of \mathbf{A} , and the n th column of \mathbf{W} , respectively. Here, diag is used to represent the diagonal matrix formed by its vector argument. $\tilde{\mathbf{Y}}$ is obtained by solving for $\mathbf{Y}^*(:, n)$, $n = 1, \dots, N$, from Equation (1.6). Finally, we compute \mathbf{V}_i , by orthonormalizing $\tilde{\mathbf{Y}}$, and obtain Σ_i as the normalizing factor. One can do this in MATLAB by using a

standard SVD routine:

$$[\mathbf{V}_i, \boldsymbol{\Sigma}_i] = \text{svd}(\tilde{\mathbf{Y}}), \quad (1.7)$$

The above step requires an SVD of an $N \times R$ matrix, thus the complexity of the above step is $\mathcal{O}(R^2N)$ flops. Next, \mathbf{U}_i is obtained similarly, by regressing rows of \mathbf{A} onto \mathbf{V}_i , and orthonormalizing. This orthogonalization can be done efficiently using Householder reflections, Givens rotations. One can do this in MATLAB by using the `orth` or `qr` routine, and the computational complexity of this step is $\mathcal{O}(R^2M)$ flops. This way of solving bilinear systems by successively regressing columns and rows of data matrix is also known as *criss-cross* regression. Alternatively, one can also consider an SVD while computing \mathbf{U}_i and orthonormalization while computing \mathbf{V}_i . Since the method is based on alternating regression, both approaches produce identical results. The detailed algorithm to compute a low rank approximation with weighted data, i.e. an algorithm to solve the problem posed in Equation 1.3, is presented as Algorithm 1.

Algorithm 1 (Previous work) `biSVD_weighted`
– bi iterative SVD on weighted data set

Require: \mathbf{A} , \mathbf{W} , R , `maxi`
 $[M, N] = \text{size}(\mathbf{A})$
 $\mathbf{A}_w = \mathbf{W} \cdot \cdot \mathbf{A}$
 $\mathbf{U}_0 \leftarrow$ any isometric matrix with R columns;
for $i = 1$ to `maxi` **do**
 for $n = 1$ to N **do**
 $\mathbf{R} = \mathbf{U}_{(i-1)}^* \mathbf{A}_w(:, n)$
 $\mathbf{S} = \mathbf{U}_{(i-1)}^* [\text{diag}(\mathbf{W}(:, n))] \mathbf{U}_{(i-1)}$
 $\mathbf{Y}^*(:, n) = \mathbf{S} \setminus \mathbf{R}$ // matrix left division
 end for
 $[\mathbf{V}_i, \boldsymbol{\Sigma}_i] = \text{svd}(\mathbf{Y})$ // standard SVD (economy size)
 for $m = 1$ to M **do**
 $\mathbf{R} = \mathbf{V}_i^* \mathbf{A}_w(:, m)$
 $\mathbf{S} = \mathbf{V}_i^* [\text{diag}(\mathbf{W}(m, :))] \mathbf{V}_i$
 $\tilde{\mathbf{X}}(m, :) = \mathbf{S} \setminus \mathbf{R}$
 end for
 $[\mathbf{U}_i, \tilde{\mathbf{R}}] = \text{qr}(\tilde{\mathbf{X}})$ // standard QR decomposition (economy size)
end for
return \mathbf{U}_{maxi} $\boldsymbol{\Sigma}_{\text{maxi}}$ \mathbf{V}_{maxi}

Unlike the cost function for low rank approximation with equally weighted data (1.2), the cost function with weighted data (1.3) has local minima, see [22]. Thus regularization is beneficial to avoid local minima, and consequently obtain a meaningful result.

A previous attempt to regularize a low rank approximation of a weighted data set used a priori information that the data points are from normal distribution [9]. In [15], it is shown that considering the normal density function for the data is equivalent to a norm penalty in the cost function. However, regularization by using a priori information that the data points are from normal distribution is not adequate, for several applications, like reconstruction of image backgrounds. Moreover, regulariza-

tion with norm penalty is trivial in the framework of alternating regression, because the unknown variables are separable at each stage of the iteration.

In 2007, Raiko et al. proposed a method [15], in which the penalty term in the cost function (1.4) is proportional to the norm of the approximants. That is \mathbf{B}_u , and \mathbf{B}_v are set to the identity. They compute the solution by the gradient descent method. In order to further accelerate the convergence, they multiply the gradient by the inverse of the Hessian matrix. Only the diagonal of the Hessian is used in order to keep the complexity low. Similar a priori information is also used by Paterek, 2007, [14], Buchanan et al., 2005 [2], for regularization. For a comprehensive discussion of the available methods to obtain a regularized low rank approximation in case of missing data, see the work by Ilin, and Raiko, 2010, [9], and references therein.

Regularization with the a priori information that the approximants have few significant entries is also used in some applications. Such a priori information can be used in the cost function by adding a penalty term proportional to the l_0 "norm" (number of non-zero entries) of the approximants. The resulting approximants are called sparse principal components. Hard thresholding is used by several researchers for sparse principal components, see [21, 26].

For data set with equal weights, a concise discussion of the problem and solutions using a roughness penalty is available in the book by Ramsay, and Silverman [16], 1998, chapter 7. Another approach to compute smooth approximants (principal components) for the case of equally weighted data is proposed by Rice, and Silverman in 1991, see [18]. This method has the advantage that different levels of smoothing can be used for different approximants. However, the complexity of this method turns out to be very high.

2. Proposed Methods. In this section, we describe our method for computing a regularized low rank approximation of a weighted data set. At first, we describe the method for non-weighted case, and develop a preliminary algorithm for regularized low rank approximation of non-weighted (i.e. equally weighted) data sets. Thereafter we describe how the weighted case is different from the non-weighted (i.e. equally weighted) case. Next, we describe our method for dealing with the weighted case, and develop an algorithm for regularized low rank approximation of weighted data sets. In the last subsection, we describe the computational complexity and memory usage of the proposed algorithm for regularized low rank approximation of weighted data sets.

2.1. Regularized Low Rank Approximations. The problem of regularized low rank approximation of a data set is posed using an equation similar to Equation (1.4). However, we replace the weighted norm in the cost function of Equation (1.4) by the Frobenius norm. Denoting the rank by R , we consider the following problem:

$$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \operatorname{argmin}_{\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}} \left\{ \|\mathbf{A} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_F^2 + \frac{\alpha_u}{2} \|\mathbf{B}_u \mathbf{U}\|_F^2 + \frac{\alpha_v}{2} \|\mathbf{B}_v \mathbf{V}\|_F^2 \right\}, \quad (2.1)$$

where \mathbf{U} and \mathbf{U} are $M \times R$ and $N \times R$, rank R matrices, respectively.

We consider the same approach used in Equation (1.3), i.e. we iteratively compute the left approximants \mathbf{U} from the right approximants \mathbf{V} , and vice-versa. We start with an initial approximation of the left approximants, $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_R)$, we call it \mathbf{U}_0 . The matrix \mathbf{U}_0 is an random isometric matrix, i.e., $\mathbf{U}_0^* \mathbf{U}_0 = \mathbf{I}_R$. We note that, one can also begin the iterations with an initial approximation \mathbf{V}_0 of \mathbf{V} . In the i th

step, for computing \mathbf{V}_i from \mathbf{U}_{i-1} , the problem is posed as follows:

$$\begin{aligned}\tilde{\mathbf{Y}} &= \underset{\mathbf{Y}}{\operatorname{argmin}} \left\{ \|\mathbf{A} - \mathbf{U}_{(i-1)} \mathbf{Y}^*\|_F^2 + \frac{\alpha_v}{2} \|\mathbf{B}_v \mathbf{Y}\|_F^2 \right\}, \\ &= \underset{\mathbf{Y}}{\operatorname{argmin}} \left\{ \|\mathbf{U}_{(i-1)}^* \mathbf{A} - \mathbf{Y}^*\|_F^2 + \frac{\alpha_v}{2} \|\mathbf{B}_v \mathbf{Y}\|_F^2 \right\}, \\ &= \underset{\mathbf{Y}}{\operatorname{argmin}} \left\| \begin{pmatrix} \mathbf{A}^* \mathbf{U}_{(i-1)} \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{I} \\ \sqrt{\frac{\alpha_v}{2}} \mathbf{B}_v \end{pmatrix} \mathbf{Y} \right\|_F^2.\end{aligned}\quad (2.2)$$

In the above derivations, we assume that $\mathbf{U}_{(i-1)}$ is an isometry. This assumption is justified because \mathbf{U}_0 is an isometry, and as described later, after each iteration we orthonormalize \mathbf{U}_i .

The normal equations of the optimization problem posed in Equation (2.2) are obtained by equating the partial derivatives of the objective function, with respect to \mathbf{Y} , to 0. An estimate of $\tilde{\mathbf{Y}}$ is obtained by solving the normal equation as follows:

$$\begin{aligned}\begin{pmatrix} \mathbf{I} \\ \sqrt{\frac{\alpha_v}{2}} \mathbf{B}_v \end{pmatrix}^* \begin{pmatrix} \mathbf{I} \\ \sqrt{\frac{\alpha_v}{2}} \mathbf{B}_v \end{pmatrix} \mathbf{Y} &= \begin{pmatrix} \mathbf{I} \\ \sqrt{\frac{\alpha_v}{2}} \mathbf{B}_v \end{pmatrix}^* \begin{pmatrix} \mathbf{A}^* \mathbf{U}_{(i-1)} \\ \mathbf{0} \end{pmatrix} \\ \left(\mathbf{I} + \frac{\alpha_v}{2} \mathbf{B}_v^* \mathbf{B}_v \right) \mathbf{Y} &= \mathbf{A}^* \mathbf{U}_{(i-1)}.\end{aligned}\quad (2.3)$$

We note that Equation (2.3) is a linear system of equations with positive definite coefficient matrix, and therefore the Cholesky decomposition can be employed to solve the system. In this case, the coefficient matrix is constant throughout the iterations, thus only one Cholesky factorization is required. Given a priori information that the norm of \mathbf{V} is bounded, one can set \mathbf{B}_v to the identity matrix \mathbf{I} . Making use of this a priori information is trivial, and finally results to a shrinkage estimator. With a priori information that the columns of \mathbf{V} are smooth, we use a roughness penalty by setting \mathbf{B}_v to a second order finite differential operator. A second order differential operator is approximated by a banded matrix. Thus, the coefficient matrix is also banded, and positive definite. Consequently, the Cholesky factorization is fast and the factors are banded. The banded Cholesky factors enables fast forward and backward substitutions required to solve Equation (2.3). Finally, we compute \mathbf{V}_i by orthonormalizing the estimate of $\tilde{\mathbf{Y}}$, and obtain Σ_i as the normalizing factor. The standard economy size SVD algorithm is applied for this purpose, at a computational complexity of $\mathcal{O}(R^2 N)$ flops.

Next, \mathbf{U}_i is obtained similarly, by regressing the rows of \mathbf{A} onto \mathbf{V}_i along with the associated penalty term, and orthonormalizing them at the end. This orthogonalization can be done efficiently using Householder reflections, Givens rotations, at a computational complexity of $\mathcal{O}(R^2 M)$ flops. The algorithm to compute a low rank approximation with non-weighted (i.e. equally weighted) data set, i.e. an algorithm for the problem posed in Equation (2.1), is presented in Algorithm 2, we call this routine `biSVD_regularized`. Alternatively, one can also consider an SVD while computing \mathbf{U}_i and just orthonormalization while computing \mathbf{V}_i . Since the method is based on alternating regression, both approaches produces identical results.

2.2. Regularized Low Rank Approximations with Weighted Data. In this section we describe a method to obtain a regularized low rank approximation with weighted data. That is a method for the problem posed in equation (1.4). We consider the same approach of alternating regression as we did for solving the problems posed in Equations (1.3) and (2.1), i.e. we iteratively compute the left

Algorithm 2 (Preliminary algorithm)

biSVD_regularized – bi-iterative SVD for regularized singular vectors

Require: \mathbf{A} , \mathbf{B}_u , \mathbf{B}_v , R , maxi $\mathbf{U}_0 \leftarrow$ any isometric matrix with, R columns; $[\mathbf{L}_u, \mathbf{L}_u^*] = \text{chol}(\mathbf{I} + \mathbf{B}_u^* \mathbf{B}_u)$ // $\text{chol}()$ -- Cholesky Decomposition $[\mathbf{L}_v, \mathbf{L}_v^*] = \text{chol}(\mathbf{I} + \mathbf{B}_v^* \mathbf{B}_v)$ **for** $i = 1$ to maxi **do** $\mathbf{Y} = \mathbf{L}_u^* \setminus (\mathbf{L}_u \setminus (\mathbf{A}^* \mathbf{U}_{(i-1)}))$ $[\mathbf{V}_i, \mathbf{\Sigma}_i] = \text{svd}(\mathbf{Y})$ // economy size SVD $\mathbf{X} = \mathbf{L}_v^* \setminus (\mathbf{L}_v \setminus (\mathbf{A} \mathbf{V}_i))$ $[\mathbf{U}_i, \tilde{\mathbf{R}}] = \text{qr}(\mathbf{X})$ // economy size QR**end for****return** \mathbf{U}_{maxi} $\mathbf{\Sigma}_{\text{maxi}}$ \mathbf{V}_{maxi}

approximants \mathbf{U} from the right approximants \mathbf{V} , and vice-versa. We start with an initial approximation of the left approximants, $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_R)$, we call it \mathbf{U}_0 . The matrix \mathbf{U}_0 can be any random isometric matrix, i.e., $\mathbf{U}_0^* \mathbf{U}_0 = \mathbf{I}_R$. We note that, one can also start the iterations with an initial approximation \mathbf{V}_0 of \mathbf{V} . In the i th step, for computing \mathbf{V}_i given \mathbf{U}_{i-1} , the problem posed in Equation (1.4) reduces to the following problem:

$$\tilde{\mathbf{Y}} = \underset{\mathbf{Y}}{\text{argmin}} \left\{ \|\mathbf{A} - \mathbf{U}_{(i-1)} \mathbf{Y}^*\|_W^2 + \frac{\alpha_v}{2} \|\mathbf{B}_v \mathbf{Y}\|_F^2 \right\}. \quad (2.4)$$

Unlike Equation (2.2), Equation (2.4) is not separable in general, because here the Frobenius norm is replaced by the weighted norm, and therefore the algebraic manipulations done in Equation (2.2) are not applicable in this case. However, we note that, in case of a norm penalty, \mathbf{B}_v is set to identity, and therefore the variables are still separable, which leads to a linear least square problem. Buchanan and Fitzgibbon [2] discuss the necessity of going beyond the regularization with a norm penalty. They also observe that the variables of Equation (2.4) are not separable, and therefore it is not easy to proceed with the alternating iterative method, see the section on discussion and conclusions in [2]. However in order to formulate a tractable algorithm to solve for the above problem, we exploit the structure and sparsity of the associated matrices. First, we derive a solution of the above equation with \mathbf{B}_v as any matrix suitable for the purpose of regularization. Later we demonstrate in particular how to use the roughness penalty operator with low complexity.

The first part of the cost function in Equation (2.4) is expressed as follows:

$$\begin{aligned} & \|\mathbf{A} - \mathbf{U}_{(i-1)} \mathbf{Y}^*\|_W^2 \\ &= \left\| \mathbf{W}_{\frac{1}{2}} \circ \mathbf{A} - \mathbf{W}_{\frac{1}{2}} \circ [\mathbf{U}_{(i-1)} \mathbf{Y}^*] \right\|_F^2 \\ &= \sum_{n=1}^N \left\| \left[\text{diag} \left(\mathbf{W}_{\frac{1}{2}}(:, q) \right) \right] \mathbf{A}(:, q) - \left[\text{diag} \left(\mathbf{W}_{\frac{1}{2}}(:, q) \right) \right] [\mathbf{U}_{(i-1)} \mathbf{Y}^*(:, q)] \right\|_2^2 \\ &= \left\| \left[\text{diag} \left(\mathbf{W}_{\frac{1}{2}}(:, :) \right) \right] \mathbf{A}(:, :) - \left[\text{diag} \left(\mathbf{W}_{\frac{1}{2}}(:, :) \right) \right] [\mathbf{I} \otimes \mathbf{U}_{(i-1)}] [\mathbf{Y}^*(:, :)] \right\|_2^2 \\ &= \left\| \mathbf{W}_d \vec{\mathbf{A}} - \mathbf{W}_d [\mathbf{I} \otimes \mathbf{U}_{(i-1)}] \vec{\mathbf{Y}}^* \right\|_2^2. \end{aligned} \quad (2.5)$$

Here $\mathbf{W}_{\frac{1}{2}}$ is element-wise square root of the non-negative weight matrix \mathbf{W} . The matrix \mathbf{W}_d is a diagonal matrix of size $MN \times MN$ such that $\mathbf{W}_d(nM+m, nM+m) = \mathbf{W}_{\frac{1}{2}}(m, n)$, $\vec{\mathbf{Y}}^*$ is a vectorised form of \mathbf{Y}^* such that $\vec{\mathbf{Y}}^*(nR+r) = \mathbf{Y}^*(n, r)$, and $\vec{\mathbf{A}}$ is a vectorised form of \mathbf{A} such that $\vec{\mathbf{A}}^*(nM+m) = \mathbf{A}^*(m, n)$, for $n = 1, \dots, N$, $m = 1, \dots, M$, and $r = 1, \dots, R$. The operator *diag* gives a diagonal matrix with its vector argument in the diagonal.

The second part of the cost function is expressed as follows:

$$\begin{aligned}
& \frac{\alpha_v}{2} \|\mathbf{B}_v \mathbf{Y}\|_F^2 \\
&= \frac{\alpha_v}{2} \sum_{n=1}^N \sum_{r=1}^R \left| \sum_{k=1}^N \mathbf{B}_v(n, k) \mathbf{Y}(k, r) \right|^2 \\
&= \frac{\alpha_v}{2} \sum_{n=1}^N \sum_{r=1}^R \left| \sum_{k=1}^N \overline{\mathbf{B}_v(n, k)} \overline{\mathbf{Y}(k, r)} \right|^2 \\
&= \sum_{t=1}^{RN} \left(\sum_{s=1}^{RN} \ddot{\mathbf{B}}_v(t, s) \vec{\mathbf{Y}}^*(s) \right)^2 \\
&= \|\ddot{\mathbf{B}}_v \vec{\mathbf{Y}}^*\|_2^2.
\end{aligned} \tag{2.6}$$

Here $\ddot{\mathbf{B}}_v$ is defined in terms of \mathbf{B}_v as follows:

$$\ddot{\mathbf{B}}_v((n-1)R+r_a, (k-1)R+r_b) = \begin{cases} \sqrt{\frac{\alpha_v}{2}} \overline{\mathbf{B}_v((q-1)+r_a, k)} & \text{if } r_a = r_b, \\ 0 & \text{otherwise,} \end{cases} \tag{2.7}$$

where $n, k = 1, \dots, N$, and $r_a, r_b = 1, \dots, R$. With this formulation one can also consider different regularization for different approximants. In that case, for $r = 1, \dots, R$, one can use $\mathbf{B}_u^r, \mathbf{B}_v^r$, and α_u^r, α_v^r , instead of $\mathbf{B}_u, \mathbf{B}_v$, and α_u, α_v , respectively, by expressing $\ddot{\mathbf{B}}_v$ (and similarly $\ddot{\mathbf{B}}_u$) as follows:

$$\ddot{\mathbf{B}}_v((n-1)R+r_a, (k-1)R+r_b) = \begin{cases} \sqrt{\frac{\alpha_v^r}{2}} \overline{\mathbf{B}_v^r((q-1)+r_a, k)} & \text{if } r_a = r_b = r, \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

The optimization problem posed in Equation (2.4), is now expressed with the modified form of the cost functions, derived in Equation (2.5) and (2.6), as follows:

$$\begin{aligned}
\tilde{\vec{\mathbf{Y}}} &= \underset{\vec{\mathbf{Y}}}{\operatorname{argmin}} \left\{ \left\| \mathbf{W}_d \vec{\mathbf{A}} - \mathbf{W}_d [\mathbf{I} \otimes \mathbf{U}_{(i-1)}] \vec{\mathbf{Y}}^* \right\|_2^2 + \|\ddot{\mathbf{B}}_v \vec{\mathbf{Y}}\|_2^2 \right\} \\
&= \underset{\vec{\mathbf{Y}}}{\operatorname{argmin}} \left\| \begin{pmatrix} \mathbf{W}_d \vec{\mathbf{A}} \\ 0 \end{pmatrix} - \begin{pmatrix} \mathbf{W}_d [\mathbf{I} \otimes \mathbf{U}_{(i-1)}] \\ \ddot{\mathbf{B}}_v \end{pmatrix} \vec{\mathbf{Y}}^* \right\|_2^2.
\end{aligned} \tag{2.9}$$

2.2.1. Smooth approximants. With a priori information that low rank approximants are smooth, a roughness penalty term in the cost function is an appropriate choice. Such a penalty is constructed by setting \mathbf{B}_v (and \mathbf{B}_u) to a second order finite difference matrices with different order of accuracies. We denote the second order finite difference matrix by \mathbf{D} . We note that depending on the problem one have

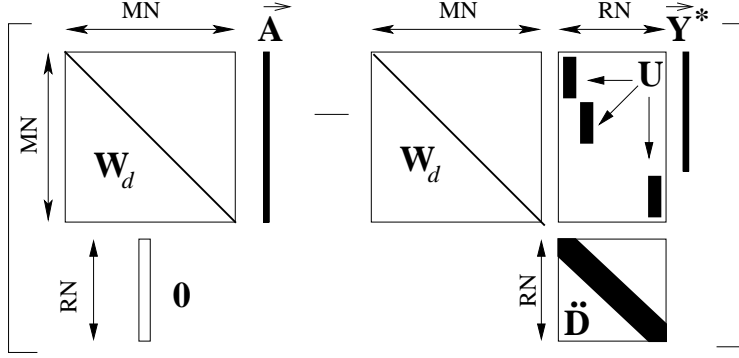


FIG. 2.1. Pictorial representation of the cost function, illustrating the sparsity structure.

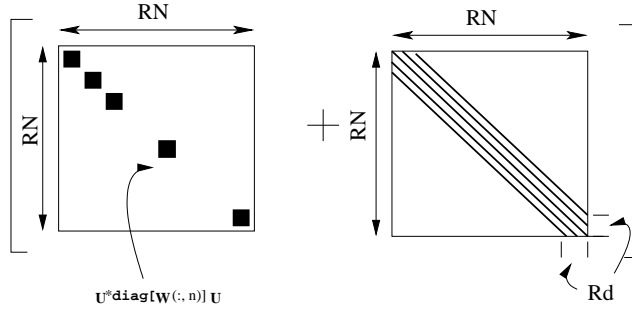


FIG. 2.2. Pictorial representation of the coefficient matrix of the normal equation, illustrating the sparsity.

to set proper boundary values while constructing \mathbf{D} . We note that \mathbf{D} is a banded matrix, which makes \mathbf{B}_v and $\ddot{\mathbf{B}}_v$ also banded.

The structure and sparsity of the associated matrices in the least square problem derived in Equation (2.9) is pictorially illustrated in Figure 2.1. We note that with this formulation, one can adopt different regularization for different approximants as shown in Equation (2.8), while preserving sparsity.

Next, a system of normal equations corresponding to Equation (2.9) is formulated by setting to zero the derivatives of the cost function with respect to $\vec{\mathbf{Y}}^*$. With a straightforward algebraic manipulation the normal equations are obtained as follows:

$$\left(\left[\mathbf{I} \otimes \mathbf{U}_{(i-1)}^* \right] \mathbf{W}_d^2 \left[\mathbf{I} \otimes \mathbf{U}_{(i-1)} \right] + \ddot{\mathbf{B}}_v^* \ddot{\mathbf{B}}_v \right) \vec{\mathbf{Y}}^* = \left[\mathbf{I} \otimes \mathbf{U}_{(i-1)}^* \right] \mathbf{W}_d^2 \vec{\mathbf{A}}. \quad (2.10)$$

The coefficient matrix of the normal equation (2.10) is pictorially represented in Figure 2.2, where the sparsity of the associated matrices are also illustrated. We note that the coefficient matrix of the normal Equations (2.10) is positive definite, therefore we can employ the Cholesky factorization for solving it. In the next subsection we discuss in detail the computational complexity of this method.

With the estimate of $\vec{\mathbf{Y}}^*$ obtained from Equation (2.10), the estimate of $\tilde{\mathbf{Y}}$ is obtained by rearranging and taking the conjugate transpose of the estimate of $\vec{\mathbf{Y}}^*$. Finally, we compute \mathbf{V}_i by orthonormalizing the estimate of $\tilde{\mathbf{Y}}$, and obtain Σ_i as the normalizing factor. The standard economy size SVD algorithm is used for this purpose. The above step requires an SVD of an $M \times R$ matrix, thus the complexity

of the above step is $\mathcal{O}(R^2N)$ flops.

In the next half of this iteration, \mathbf{U}_i is obtained similarly, by a regularized weighted regression of the rows of \mathbf{A} onto \mathbf{V}_i and orthonormalizing. This orthonormalization can be done efficiently using Householder reflections, or Givens rotations. One can do this in MATLAB by using the `orth` or `qr` routine, and the computational complexity of this step is $\mathcal{O}(R^2M)$ flops. The algorithm to compute regularized low rank approximation with weighted data is presented in Algorithm 3, we call this routine BIRSVD.

Algorithm 3 (Final Algorithm)

BIRSVD – Bi Iterative Regularized SVD with weighted data

Require: $\mathbf{A}, \mathbf{B}_u, \mathbf{B}_v, \mathbf{W}, R, \text{maxi}$

```

[M, N] = size(A)
A_w = W .* A
U_0 ← any isometric matrix with, N rows and R columns;
Compute B_u from B_u // Equation (2.7); in practice, access entries
Compute B_v from B_v // of B_u, B_u directly from B_v, B_v
for i = 1 to maxi do
    R = reshape(U* A_w, NR, 1)
    L = [ ] // set empty
    for n = 1 to N do
        L = blkdiag(L, U_{i-1}^* ((repmat(W(:, n), 1, R) .* U_{i-1})))
    end for
    [L_v, L_v^*] = chol(L + B_v)
    Y = L_v^* \ (L_v \ R)
    Y = reshape(Y, R, N)
    [V_i, Sigma_i] = svd(Y) // economy size SVD
    R = reshape(V^* A_w, MR, 1)
    L = [ ] // set empty
    for m = 1 to M do
        L = blkdiag(L, V_i^* ((repmat(W^*(:, m), 1, R) .* V_i)))
    end for
    [L_v, L_v^*] = chol(L + B_u)
    X = L_v^* \ (L_v \ R)
    X = reshape(X, R, M)
    [U_i, R] = qr(X) // economy size QR
end for
return U_{maxi} Sigma_{maxi} V_{maxi}
// reshape() - reorder entries with given dimensions
// blkdiag() - concatenated matrices block diagonally
// repmat() - make multiple copy of matrices

```

We note that the routine BIRSVD is equivalent to the routine `biSVD_weighted` in case no regularization is required. Also, in case all the data points are equally weighted, the routine BIRSVD is equivalent to the routine `biSVD_regularized`.

2.2.2. Other a priori information. Apart from regularization using a priori information regarding smoothness of the approximants, other types of a priori information about the approximants can also be used within the proposed framework whenever suitable penalty matrices \mathbf{B}_u and \mathbf{B}_v are available. Unlike the roughness

penalty matrices, the penalty matrices \mathbf{B}_u and \mathbf{B}_v associated with other such a priori information might be dense, e.g. when they correspond to a band pass filters. We note that Equation (2.9) is an overdetermined least square problem with the following coefficient matrix:

$$\begin{pmatrix} \mathbf{W}_d [\mathbf{I} \otimes \mathbf{U}_{(i-1)}] \\ \ddot{\mathbf{B}}_v \end{pmatrix}. \quad (2.11)$$

Typically, the penalty matrices \mathbf{B}_v associated with a certain a priori information are described only by a few parameters. Thus a matrix-vector multiplication with the penalty matrix and its Hermitian transpose should be achieved at a very low complexity. Consequently, the structure of $\ddot{\mathbf{B}}_v$, (2.7) suggests that one can perform a fast matrix vector multiplication with $\ddot{\mathbf{B}}_v$, and its Hermitian transpose as well. Thus, in spite of the size of the coefficient matrix (2.11), the coefficient matrix and its Hermitian transpose can be applied to a vector with $\mathcal{O}(RM + RN)$ operations only. Thus an iterative method like LSQR [13] designed for overdetermined least square problems, can be used to solve Equation (2.9). With the regularization factor, the method works like damped LSQR, and the resulting semi convergence is very mild.

2.3. Computational Complexity and Memory Usage. In this subsection we discuss in detail the computational complexity and the memory usage of the proposed algorithm BIRSVD for computing a regularized low rank approximation of a weighted data set. In particular, for regularization we consider a priori assumption that the low rank approximants are smooth, and thus we use a second order finite difference matrix with appropriate boundary values, in the penalty term.

The proposed method is iterative, therefore we estimate the work per iteration. In Section 3, we present the number of iterations required for the different cases, and thus an estimate of the total computational complexity is easily obtained. Within each iteration, we compute the right approximants \mathbf{V} from the last estimated approximants \mathbf{U} , and vice-versa. We discuss in detail the computational complexity of computing the right approximants \mathbf{V} in each iteration. The computational complexity per iteration of estimating the left approximants \mathbf{U} is thus obtained by swapping the number of rows, M , with number of columns, N , and swapping the parameters related to the penalty term in case the penalty terms are different for left and right approximants.

We assume that the following are given: a matrix \mathbf{A} of size $M \times N$, a corresponding weight matrix \mathbf{W} of size $M \times N$ with non negative entries. The matrix for the penalty term, a second order finite difference operator of accuracy d , call it \mathbf{D} . We note that the penalty matrix \mathbf{D} can be different for the left and the right approximants. For this algorithm we need the interleaved version of \mathbf{D} , i.e. one corresponding to $\ddot{\mathbf{B}}$ in Equation (2.7) or (2.8), we call it $\ddot{\mathbf{D}}$. In practice $\ddot{\mathbf{D}}$ should not be constructed explicitly, the entries of $\ddot{\mathbf{D}}$ should be accessed directly from \mathbf{D} using Equation (2.7) or (2.8). Also the rank of approximation, R , is provided as an input to the algorithm.

We measure the computational complexity in terms of the number of complex floating-point operations (flops). Each flop consists of one floating point complex multiplication and addition to the current value of the memory location, where the product is stored. We ignore the cost of copying numbers from one memory location to other, but such details are taken care while discussing the memory usage.

The computation of \mathbf{V} in each iteration requires the following steps:

- Computation of the coefficient matrix of Equation (2.10).
- Computation of right hand side vector of Equation (2.10).

- Cholesky factorization of the coefficient matrix.
- The forward and the backward substitution to solve Equation (2.10).
- Finally, orthogonalizing \mathbf{Y} to get an estimate \mathbf{V} for this iteration.

The coefficient matrix of Equation (2.10) is presented pictorially in Figure 2.2. The creation of the first part of the coefficient matrix requires $2R^2MN$ complex flops. The entries of the second part of the coefficient are already available, and thus the cost of computation to obtain the coefficient matrix is $2R^2MN$ flops. As is evident from Figure 2.2, the coefficient matrix is banded, and also symmetric positive definite, see Equation (2.10). Therefore, we perform a banded Cholesky decomposition. The cost of computation of the decomposition is $(R+d)R^2N/3$ complex flops. Computation of the right hand side vector of Equation (2.10) requires $2RNM$ complex flops. The forward and the backward substitution for solving Equation (2.10) require $4dR^2M$ complex flops. And finally an economy size SVD to obtain \mathbf{V} and $\mathbf{\Sigma}$ from \mathbf{Y} requires $(4R^2N + 4R^3)$ complex flops.

In most practical problems, one is interested in a very small number of principal approximants R , compared to the total number of data points MN , i.e. $R \ll MN$. Thus on the whole, the proposed algorithms have a computational complexity of $\mathcal{O}(NM)$, i.e. linear in the number of data points.

For estimation of the memory requirements we assume that the entries of data set are complex doubles, and the weights are real doubles. For storing the coefficient matrix of the normal Equation (2.10), we require $(R^2 + 2d - 1)N$ complex doubles. The Cholesky factorization fills in the Rd lower diagonals, thus require ca. RdN complex doubles. However, the Cholesky factor can be overwritten on the locations of the coefficient matrix. Thus ca. RdN complex doubles are required to obtain the Cholesky factor. The right hand side of the normal Equations (2.10) requires RN complex doubles. The results of the forward and backward substitutions can be overwritten on the locations for the output \mathbf{V} . The orthonormalization using the standard SVD algorithm requires RN complex doubles.

For different data types of the entries of the data set and the corresponding weights, the complexity of the proposed method is scalable in an ad-hoc manner. For example, if for a certain problem, the entries of the data set and the corresponding weights are presented as real floats, then the complexity is one-fourth of what is required in case the entries of the data sets are presented as complex doubles and the corresponding weights are presented as doubles.

3. Numerical Results. In this section we illustrate numerically the performance of the proposed method to compute a regularized low rank approximation of a weighted data set.

3.1. Test Data. For illustration, we consider an astronomical image observed with a wide field imager. As discussed in the introductory section, wide field astronomical images observed at the optical wavelength range have significant structures in the background of the image. These structures are smooth, and vary significantly over the whole image. Typically, such astronomical images has two main components: the celestial objects, and the background. To illustrate the proposed method, we considered the task of estimation of the smoothly varying background and correction of the astronomical image. This correction is required for an accurate quantitative analysis of the celestial objects, involving photometry and astrometry.

As discussed in the introductory section, the state-of-the-art method for correcting image background works as follows. First, pixels corresponding to the bright objects are systematically identified, thereby creating a mask (which we will use as a weight)

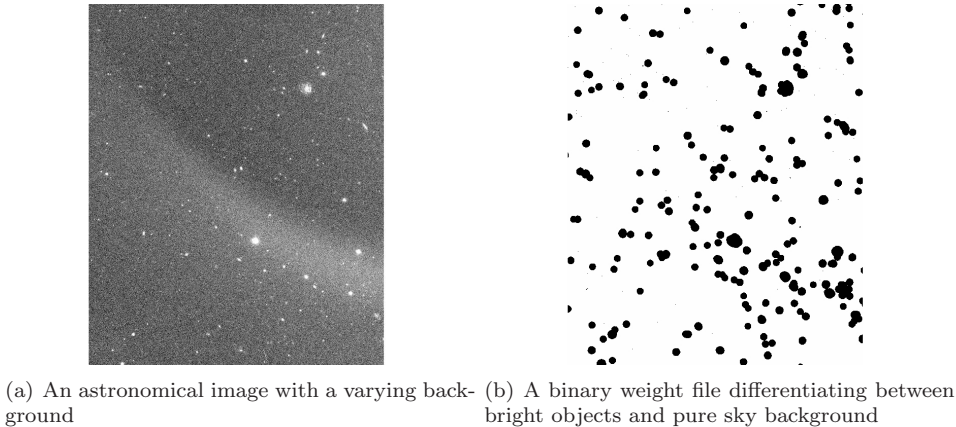


FIG. 3.1. *Data set, and corresponding binary weights*

to differentiate the pure background part of the image from the celestial objects. Then, a polynomial in two variables of a low degree is fitted to the pure background pixel intensity data, using the least squares technique. Finally, the sky pixel intensity data for the whole image is reconstructed as a linear combination of polynomials, and subtracted from the whole observed image pixel intensities. This procedure is implemented in several packages for astronomical image processing, see Section 1. Another popular method for background computation is median filtering, but this method fails at the edge pixels. This drawback of median filtering makes it inadequate for generating mosaic with multiple images.

Instead of reconstructing the background pixel intensities for the whole image as a polynomial in two variables, we estimated the background pixel intensities for the whole image by a regularized low rank approximation of the pure background pixel intensity data. Since the pixels corresponding to the bright objects are not used in estimating the background, the data set had missing values. This regime is described by a special weighted data set, where the weights are binary.

In particular, we considered an R-band image of the *Great Observatories Origins Deep Survey* (GOODS), using the VIMOS instrument mounted at the Very Large Telescope (VLT-U3) at European Southern Observatories (ESO) Cerro Paranal Observatory, Chile. The images of this survey program are available for public use, see <http://archive.eso.org>. The image used in this paper can also be found as a part of accompanying software [20]. The image is shown in Figure 3.1(a). Note the significant background, which in this case is due to unwanted scattering of light within the telescope. Such significant background hinders accurate photometry of the celestial objects.

As a first step, we identified the pixels that belong to the celestial objects present in the field of view, and thereby create a binary weight (mask) matrix for the data set. Pixels corresponding to celestial objects were assigned the weight 0, and pixels corresponding to background are assigned the weight 1. The procedure used to identify the pixels corresponding to celestial objects is as follows: first a robust mean of the background pixel intensities is estimated. For this case, the sample median of the pixel intensity data is used, so that the estimated mean of the background pixel intensities is not effected by the bright pixels corresponding to the celestial objects. Next, a robust estimate of the deviation of the background pixel intensities is computed from

the inter-quartile range. All the pixels whose intensity exceeded the estimated mean by κ times the estimated deviations are marked as pixels corresponding to celestial object. To make sure that even a weak halo of the bright objects is eliminated, we did a morphological dilation to grow the number of identified bright object pixels. Mask corresponding to the image in Figure 3.1(a) is shown in Figure 3.1(b). For creating the binary mask there are two parameters, first the factor κ , and second the radius of the kernel used for morphological dilation. For this example, we used $\kappa = 5$, and set the radius of the kernel for morphological dilation to 10 pixels.

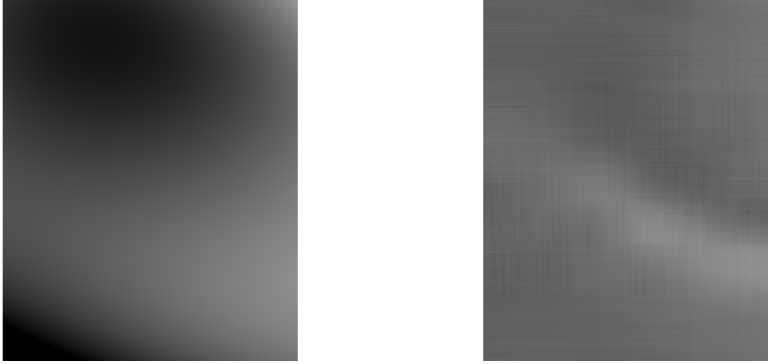
3.2. Test Results. For the purpose of comparison, we estimated the background by fitting tensor products of the Legendre polynomials to the pure background pixel intensities. Figure 3.2(a) shows such a background computed using tensor products of the Legendre polynomials whose degrees sum up to less than four. The background is smooth, but as evident from the figure, the reconstructed background grows (or decays) very fast towards the edges. This happens because of the well known Runge phenomenon.

Next, also for the purpose of comparison, we estimated the background as a low rank approximation of the pixel intensity data, Figure 3.1(a), with weights shown in Figure 3.1(b) and no regularization. We used the routine `biSVD_weighted` for this purpose, see Algorithm 1. The resulting background with a rank 4 approximation is shown in Figure 3.2(b). From the figure, it is evident that problem of the fast growth (or decay) around the edges is resolved. The background features are well captured. However, the background is not smooth locally, and vertical and horizontal criss-cross stripes are clearly visible. Moreover, this routine converges to a local minimum, see the next subsection for more detail.

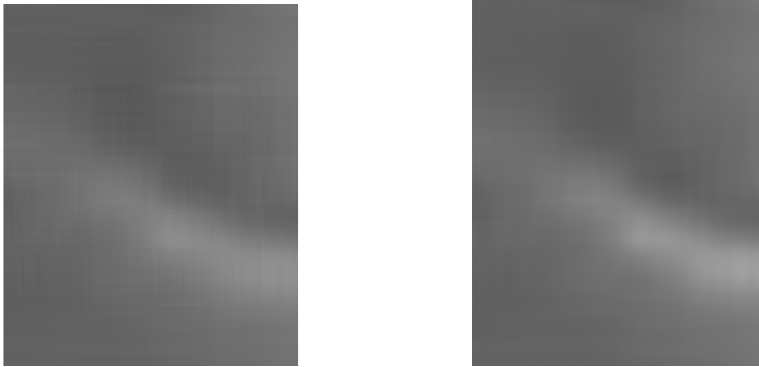
Another possible approach to obtain the low rank approximation without any regularization is by iteratively imputing the missing values from the approximants of the last iteration. This approach require an initial approximation of the missing values, and is known to be suboptimal, see [6]. We do not show the results obtained with this approach. However, an implementation of this approach is available as a part of the accompanying software [20] for the purpose of comparison only.

Furthermore, to alleviate the non-smoothness in the reconstructed background, we estimated the background as a regularized low rank approximation of the pixel intensity data, Figure 3.1(a), with weights shown in Figure 3.1(b). Figure 3.2(c) and 3.2(d) shows sky backgrounds estimated as a rank 4 approximation, where the low rank approximants are regularized with second order differences of accuracies 2 and 8, respectively. We use the proposed routine `BIRSVD` for this purpose, see Algorithm 3.

As the final step, we correct the given image, Figure 3.1(a), by subtracting the background from it. Figure 3.3 shows the results after a background correction. Figure 3.3(a) shows the background corrected image with the polynomial background. We notice that a significant amount of scattered light is still present in the image. Figure 3.3(b) shows the background corrected image, where the background was computed as a rank 4 approximation of the background pixel intensities shown in Figure 3.2(b). Figure 3.3(c) and Figure 3.3(d) show the background corrected images, where the background were computed as a rank 4 approximation of the background pixel intensities, and regularized with second order finite difference operators implemented with accuracy 2, shown in Figure 3.2(c), and accuracy 8 shown in Figure 3.2(d), respectively. Comparing Figure 3.3(a) with Figure 3.3(d), we notice a significant improvement achieved by the proposed method in correcting the background.



(a) A background estimated by least square fit- (b) A background estimated by low rank ap-
 of tensor products of Legendre polynomials. proximation of the data weighted data set



(c) A background estimated by a low rank ap- (d) A background estimated by a low rank ap-
 proximation of the weighted data set, regular- proximation of the weighted data set, regular-
 ized with a second order difference of accuracy 2 ized with a second order difference of accuracy 8

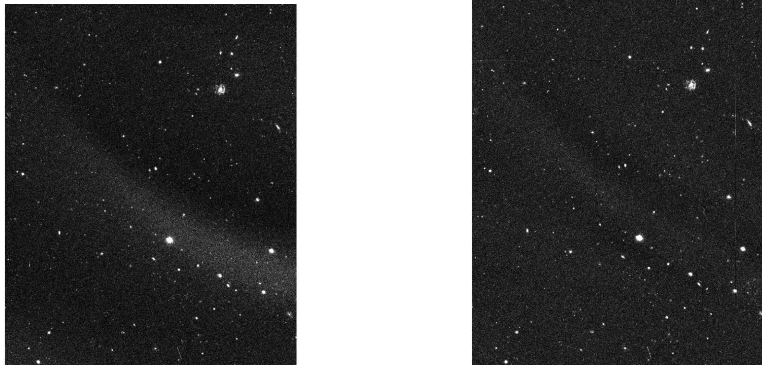
FIG. 3.2. *Sky background estimated using polynomials, and low rank approximations*

In this paper we deduce the effectiveness of the proposed method from the visual perception only. A further systematic quantitative analysis like relative photometry, and comparison of the flux of known celestial objects from a standard star catalogue is out of the scope of this paper, and it will be published in an astronomical journal.

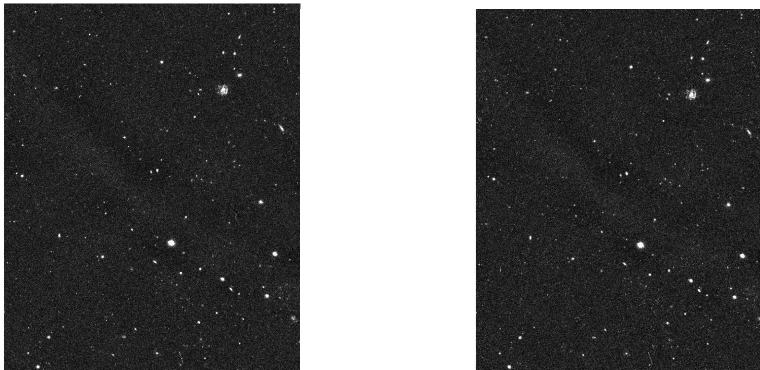
The high resolution images used illustration of the proposed method can be found online at [20].

3.3. Alleviation of Local Minima. In this subsection we present our observations regarding the effectiveness of the proposed method in order to alleviate local minima. We test the proposed method BIRSVD for regularized low rank approximation of weighted data, i.e. Algorithm 3 against the previously known method `biSVD_weighted` for low rank approximation of weighted data, i.e. Algorithm 1, on the same data set 1,000 times, but with a random starting point. We notice that each time Algorithm 1 converges to a different point, whereas the proposed Algorithm 3 consistently converges to the same point. We note that our experiment with 1,000 different starting points is in no way exhaustive. However, the results of this experiment at least conform with our expectation.

In Table 3.1, we show the first 4 singular values obtained with different start-



(a) Background corrected with the polynomial based sky, Figure 3.2(a) (b) Background corrected with weighted low rank approximation of the image, Figure 3.2(b)



(c) Background corrected with regularized and weighted low rank approximation of the image, Figure 3.2(c) (d) Background corrected with regularized and weighted low rank approximation of the image, Figure 3.2(d)

FIG. 3.3. *Background corrected images*

ing points by Algorithm 1 and Algorithm 3. Although we made experiment with 1,000 runs with different starting points, we only show the results of 4 such runs to demonstrate the deficiency of Algorithm 1. On the other hand, Algorithm 3 converges consistently in these 1,000 runs to the same result. The normalized error of the regularized approach is larger than that of the non-normalized one. This is expected, because Algorithm 3 not only minimizes the error on the data, but also includes terms related to the roughness penalties. Especially for this particular data set, the unregularized method, i.e. Algorithm 1, tends to fit some bright pixels corresponding to undetected faint objects. On the other hand, the regularized method, i.e. Algorithm 3 is immune to such undetected faint objects.

3.4. Speed of Convergence. The proposed method is iterative, hence in this paragraph we discuss its rate of convergence. The rank of the approximation is an important factor for the rate of convergence. Figure 3.4 shows the rate of convergence for different ranks of the approximation. The horizontal axis shows the number of iterations, and the vertical axis shows the root mean squared error (RMSE) between

biSVD_weighted, Algorithm 1 (No Regularization)					
Exp. no.	σ_1	σ_2	σ_3	σ_4	$\frac{\ \mathbf{A}-\mathbf{U}\Sigma\mathbf{V}^*\ _W^2}{\ \mathbf{A}\ _W^2}$
1	4.5041 E06	9.0680 E04	1.6715 E04	1.2160 E04	1.4939 E - 02
2	4.5041 E06	8.6671 E04	1.6689 E04	1.2150 E04	1.4941 E - 02
3	4.5037 E06	8.9940 E04	1.6717 E04	1.2152 E04	1.4940 E - 02
4	4.5040 E06	1.7032 E04	1.2170 E04	7.4412 E03	1.4945 E - 02
biSVD_weighted_regularized, Algorithm 3					
1 - 1,000	4.5040 E06	1.6688 E04	1.1915 E04	5.9132 E03	1.4981 E - 02

TABLE 3.1
Singular Values of Rank 4 approximation, and approximation error

successive iterations, i.e.

$$\text{RMSE}_i = \frac{1}{\sqrt{MN}} \left\| \mathbf{U}_i \Sigma_i \mathbf{V}_i^* - \mathbf{U}_{(i-1)} \Sigma_{(i-1)} \mathbf{V}_{(i-1)}^* \right\|_F \quad (3.1)$$

All curves in Figure 3.4 are generated with the same data set shown in Figure 3.1, with the binary weights generated by attributing ca. 20% of the pixel intensity values as part of bright objects. As expected with power iteration type methods, the rate of convergence for rank R approximation is relatively slower whenever the R th singular value is relatively close to $(R + 1)$ th singular value. For the exemplary data set, we define the slowness of convergence S_R , for a rank R approximation, as the difference of the number of iteration to reach an RMSE of 1 E-08 and the number of iteration to reach an RMSE of 1 E-01. For different rank approximations, Table 3.2 shows the singular values, σ_R ; the relative distance of the singular values, in terms of σ_R/σ_{R+1} ; and the slowness of convergence, S_R . In Table 3.2, we notice that the rank 6 and rank 2 approximations converge slower than others. This is explained by the fact that the 6th and the 2nd singular values are very close to their neighbors. The rates of convergence of the proposed method with different amounts of missing values are shown in Figure 3.5. For all the curves in Figure 3.5, a rank 4 approximation is considered. In conclusion, we observe a fast linear rate of convergence for several practical values of the rank of approximation, and the amount of missing values.

For this exemplary data set, we found that a rank 4 approximation was sufficient to model the varying background pixel intensities. The results obtained with rank 4 to rank 9 approximations are almost identical. The results obtained with rank 10 and onwards started modeling unwanted features in the background. Moreover, the singular values as shown in Table 3.2 do not justify using a rank higher than $R = 4$.

We tested the proposed method on several other images obtained with the VIMOS instrument, and the WFI instrument. The instrument WFI (Wide Field Imager), is mounted at the 2.5m MPG/ESO telescope at La Silla, Chile. The proposed method obtained much better backgrounds compared to the backgrounds obtained with the polynomial fit. We also tested the performance of the proposed method on simulated images. In cases where the background was simulated as tensor products of polynomials, with additive noise, the proposed method reliably identified the polynomials as left and right approximants.

For further accelerating the convergence, one can use mean corrected data. A weighted mean can be easily computed from the data points and the corresponding

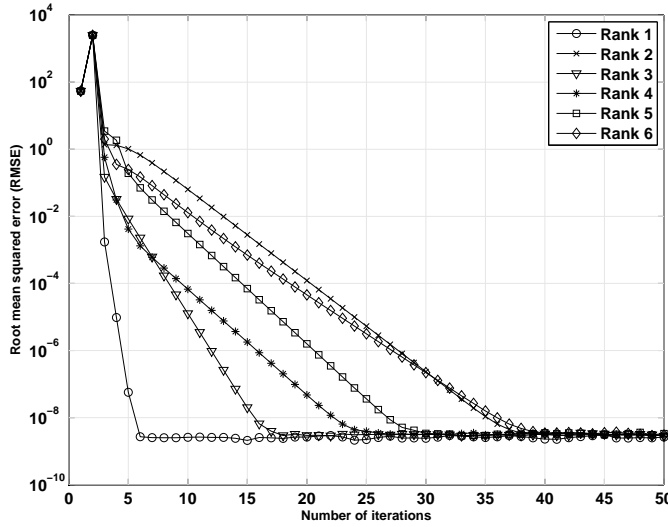


FIG. 3.4. Rate of convergence for different rank approximations

rank (R)	σ_R	σ_R/σ_{R+1}	S_R
1	4.50 E06	287.2	04
2	1.57 E04	1.311	26
3	1.20 E04	1.972	13
4	6.07 E03	1.466	20
5	4.14 E03	1.327	22
6	3.12 E03	1.061	31
7	2.94 E03		

TABLE 3.2
Slowness of convergence for different rank of approximation

weights. Finally, if the low rank approximation is required, then the weighted mean should be adjusted. We note, that for several purposes, the low rank approximants are sufficient, and the reconstructed low rank approximation of the data set is not necessary. To illustrate the effectiveness of the method in general, mean correction was not done for the exemplary data set. However, with the exemplary data set, we noticed the accelerated convergence achieved with a mean correction. Anyway, the pattern of convergence for different rank approximation remains the same, and the reconstructed backgrounds were visually indistinguishable from those presented in Figure 3.2.

4. Conclusions. In this paper we proposed a method to compute a regularized low rank approximation of a weighted data set. Data sets with missing values are treated as a special case, where the weights are set to 1 if the corresponding data point is present, or to 0 if the corresponding data point is missing or unreliable. To obtain a meaningful solution, we used a priori information about the approximants. In particular, we considered the data set to be smooth, and thus we sought smooth low rank approximants. To obtain smooth approximants, we used roughness penalty

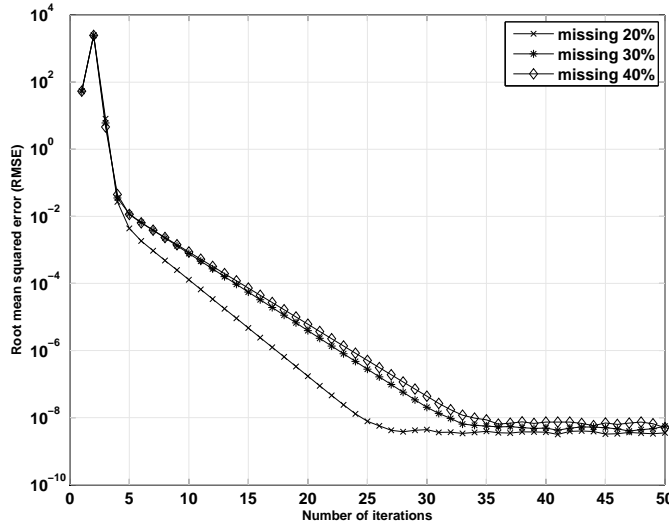


FIG. 3.5. Rate of convergence for different amount of missing values

term in the cost function of the optimization problem. Unfortunately, this penalty term made the variables of the cost function non separable. To design a tractable algorithm, we exploited the structure and sparsity of the associated matrices. The proposed algorithm has computational complexity linear in the number of data points, and the required memory is negligible compared to that needed for the storage of the given data. We tested the algorithm by reconstructing background of astronomical images. We compared the proposed method with the state-of-the-art method based on fitting tensor products of Legendre polynomials to the background pixel intensities. The background obtained with the proposed method has several desired features, compared to the state-of-the-art method. Experimentally, we found the proposed method to have a fast linear rate of convergence.

REFERENCES

- [1] J. Bennett and S. Lanning, *The netflix prize*, Proceedings of KDD Cup and Workshop, vol. 2007, Citeseer, 2007.
- [2] AM Buchanan and AW Fitzgibbon, *Damped Newton algorithms for matrix factorization with missing data*, Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2, IEEE, 2005, pp. 316–322.
- [3] C. Buil, *IRIS: Astronomical Image-Processing Software*, Digital Astrophotography: The State of the Art (2005), 79–88.
- [4] A. Christofferson, *The one component model with incomplete data*, PhD thesis, Uppsala University, Institute of Statistics, 1970.
- [5] Lars Elden, *Partial least-squares vs. lanczos bidiagonalization-i: analysis of a projection method for multiple regression*, Computational Statistics & Data Analysis **46** (2004), no. 1, 11 – 31.
- [6] K.R. Gabriel and S. Zamir, *Lower rank approximation of matrices by least squares with any choice of weights*, Technometrics (1979), 489–498.
- [7] G.H. Golub and C.F. Van Loan, *Matrix computations*, Johns Hopkins Univ Pr, 1996.
- [8] R. Hartley and F. Schaffalitzky, *PowerFactorization: 3D reconstruction with missing or uncertain data*, Australia-Japan advanced workshop on computer vision, Citeseer, 2003.
- [9] A. Ilin and T. Raiko, *Practical approaches to principal component analysis in the presence of missing values*, The Journal of Machine Learning Research **11** (2010), 1957–2000.

- [10] I.T. Jolliffe, *Principal component analysis*, Springer verlag, 2002.
- [11] J. Kuligowski, G. Quints, S Garrigues, and M. D. Guardia, *New background correction approach based on polynomial regressions for on-line liquid chromatography-fourier transform infrared spectrometry*, Journal of Chromatography A **1216** (2009), no. 15, 3122 – 3130.
- [12] P. Massey, *A User's Guide to CCD Reductions with IRAF*, NOAO Laboratory **13** (1992), 14–15.
- [13] C.C. Paige and M.A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Transactions on Mathematical Software (TOMS) **8** (1982), no. 1, 43–71.
- [14] A. Paterek, *Improving regularized singular value decomposition for collaborative filtering*, Proceedings of KDD Cup and Workshop, vol. 2007, Citeseer, 2007.
- [15] T. Raiko, A. Ilin, and J. Karhunen, *Principal component analysis for large scale problems with lots of missing values*, Machine Learning: ECML 2007 (2007), 691–698.
- [16] J.O. Ramsay and B.W. Silverman, *Functional data analysis*, Statistics and Computing **8** (1998), no. 4, 401–403.
- [17] S.E. Reichenbach, M. Ni, D. Zhang, and E.B. Ledford, *Image background removal in comprehensive two-dimensional gas chromatography*, Journal of Chromatography A **985** (2003), no. 1-2, 47–56.
- [18] J.A. Rice and B.W. Silverman, *Estimating the mean and covariance structure nonparametrically when the data are curves*, Journal of the Royal Statistical Society. Series B (Methodological) **53** (1991), no. 1, 233–243.
- [19] A. Ruhe, *Numerical computation of principal components when several observations are missing*, Dept. Inform. Processing, Umea Univ., Umea, Sweden, Tech. Rep. UMINF-48-74 (1974).
- [20] Das S. and Neumaier N., *BIRSVD: Bi-Iterative Regularized Singular Value Decomposition*, http://homepage.univie.ac.at/saptarshi.das/low_rank/low_rank_app.html.
- [21] H. Shen and J.Z. Huang, *Sparse principal component analysis via regularized low rank matrix approximation*, Journal of multivariate analysis **99** (2008), no. 6, 1015–1034.
- [22] N. Srebro and T. Jaakkola, *Weighted low-rank approximations*, MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-, vol. 20, 2003, p. 720.
- [23] G. Strang, *Introduction to linear algebra*, Wellesley Cambridge Pr, 2003.
- [24] L.N. Trefethen and D. Bau, *Numerical linear algebra*, Society for Industrial Mathematics, 1997.
- [25] Y. Wang, J. Li, and P. Stoica, *Spectral analysis of signals: the missing data case*, Synthesis Lectures on Signal Processing Series **1** (2006), no. 1, 1–102.
- [26] D.M. Witten, R. Tibshirani, and T. Hastie, *A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis*, Biostatistics **10** (2009), no. 3, 515.
- [27] H. Wold and E. Lyttkens, *Nonlinear iterative partial least squares (NIPALS) estimation procedures*, Bull. Inst. Int. Stat. **43** (1969), 29–51.
- [28] B. Yang, *Projection approximation subspace tracking*, IEEE Transactions on Signal Processing **43** (1995), no. 1, 95–107.