Fast Regularized Low Rank Approximation of Weighted Data Sets

Saptarshi Das[a,*], **Arnold Neumaier**[a,1]

[a]*Department of Mathematics, University of Vienna, Nordbergstrasse 15, Vienna, Austria 1090.*

Abstract

In this paper we propose a fast and accurate algorithm for computing a regularized low rank approximation of large weighted data sets. 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, we use a priori information that the low rank approximants are smooth.

We use alternating optimization method to estimate left and right approximants. The variables of the quadratic optimization problem posed to optimize one set of variables (assuming the other set is known) are not separable, one needs to exploit the structure and sparsity of the associated matrices to get a tractable algorithm.

The proposed 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.

The proposed algorithm is applicable to various applied problems where a priori information regarding smoothness of approximants is available. We illustrate the proposed method by reconstructing background of an astronomical image observed in the optical wavelength range, and on a data set consisting of temperature measurements of 20 cities. We provide extensive numerical results to highlight different features of the method.

Keywords: Low rank approximation, missing data, regularized approximation.

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

*Corresponding author. saptarshi.das@univie.ac.at, Fax:+43 1 4277 50690 Phone: +43 1 4277 50731

[1]arnold.neumaier@univie.ac.at

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, the high resolution images presented in this paper, and the data sets used for the experiments can be found online at the link: [38].

1.2. Low Rank Approximation

Best low rank approximations of a data set (matrix) have several applications in the field of image processing, signal processing, data analysis, see [43, 21] 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:

$$[\mathbf{u}_r, \sigma_r, \mathbf{v}_r]_{r=1}^{R} = \underset{\mathbf{u}_r, \sigma_r, \mathbf{v}_r}{\operatorname{argmin}} \ \|\mathbf{A} - \sum_{r=1}^{R} \sigma_r \mathbf{u}_r \mathbf{v}_r^*\|_F^2, \tag{1}$$

or equivalently,

$$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \underset{\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}}{\operatorname{argmin}} \ \|\mathbf{A} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_F^2, \tag{2}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_R]$ is an $M \times R$ matrix, $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_R]$ is an $M \times R$ matrix, and Σ is a diagonal matrix of size $R \times R$ with the σ_r as diagonal entries. Here $\| \ \|_F$ denotes the Frobenius norm. The columns of $\mathbf{U}$ and $\mathbf{V}$ are normalized to have norm one, otherwise the matrix Σ 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 [44]. There are stable algorithms available for SVD, see [17, 44]. 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 [41] for a data set with weighted data points.
3. The standard algorithms for SVD do not use any a priori information about the singular values or singular vectors. For example, it might be known a priori that the low rank approximation is smooth, like background of images or weather data sets. Another reasonable a priori assumption is that the approximants are sparse, see [40].

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:

$$\begin{aligned}[\mathbf{U},\mathbf{\Sigma},\mathbf{V}] &= \underset{\mathbf{U},\mathbf{\Sigma},\mathbf{V}}{\operatorname{argmin}} \|\mathbf{A}-\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_W^2 \\ &= \underset{\mathbf{u}_r,\mathbf{v}_r,\sigma_r}{\operatorname{argmin}} \|\mathbf{A}-\sum_{r=1}^{R}\sigma_r\mathbf{u}_r\mathbf{v}_r^*\|_W^2 \\ &= \underset{\mathbf{u}_r,\mathbf{v}_r,\sigma_r}{\operatorname{argmin}} \sum_{i=1}^{M}\sum_{j=1}^{N}\mathbf{W}(i,j)\left(\mathbf{A}(i,j)-\sum_{r=1}^{R}\sigma_r\mathbf{u}_r(i)\bar{\mathbf{v}}_r(j)\right)^2. \end{aligned} \tag{3}$$

Here, $\mathbf{U}=[\mathbf{u}_1,\ldots,\mathbf{u}_R]$ and $\mathbf{V}=[\mathbf{v}_1,\ldots,\mathbf{v}_R]$ are $M\times R$ and $N\times R$ rank R matrices, respectively, and $\mathbf{\Sigma}$ is a diagonal matrix with $\mathbf{\Sigma}(i,i)=\sigma_i$.

The problem posed in Equation (3) admits several locally optimal solutions, see [41]. In order to alleviate the problem of local minima, and to obtain a meaningful solution, one should 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. In Section 3 we show that smoothness assumption regarding singular vectors of weather data is also very useful, because weather changes fairly smoothly with time. 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:

$$[\mathbf{U},\mathbf{\Sigma},\mathbf{V}]=\underset{\mathbf{U},\mathbf{\Sigma},\mathbf{V}}{\operatorname{argmin}}\left\{\|\mathbf{A}-\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_W^2+\frac{\alpha_u}{2}\|\mathbf{B}_u\mathbf{U}\|_F^2+\frac{\alpha_v}{2}\|\mathbf{B}_v\mathbf{V}\|_F^2\right\}, \tag{4}$$

where $\mathbf{U}$, and $\mathbf{V}$ are $M\times R$ and $N\times R$ rank R matrices, respectively. Here $\alpha_u,\alpha_v\geqslant 0$ 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 (4) is limited to a certain type of a priori information. For example, the Equation (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 (4). We regularize the approximants using a priori information that they are smooth. Thus we use a roughness penalty in the cost function (4). 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.

The formulation of regularized low rank approximation as presented in equation (4) first appeared in [28]. With respect to the work done in [28], our contribution is a fast algorithm to solve the problem posed in equation (4), and extensive tests that demonstrate the usefulness of a priori information regarding smoothness on the quality of low rank approximation.

As described in Section 2, and also noted in [34, 7, 28], 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 (4), a possible alternative might be to penalize the approximation $\mathbf{U\Sigma V}^*$. 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 $\mathbf{U\Sigma V}^*$. Also if the smoothness information is available only either for the left or for the right singular vectors, then using a penalty on the approximation $\mathbf{U\Sigma V}^*$ is improper. For example, on the weather data used for numerical demonstration in Section 3, one can only make a smoothness assumption regarding the left singular vectors.

1.4. Applications

The proposed method has potential use in various applied problems, some of those are listed below.

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 in the vicinity of the field of view, twilight 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.*

Out of several applied problems that can benefit from the proposed method, the image processing problem described above is the first application that motivate us to pursue this work.

In order to demonstrate the performance of the proposed algorithm, we use a astronomical image data as one of the test data sets, see Section 3. 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* [29], (`imsurfit` routine); IRIS:*An Astronomical Image Processing Software* [8], (`remove gradient` procedure); some instrument specific data reduction pipelines of ESO also have this facility. Apart from the approach mentioned above, other approaches include median filtering [42]. However, for median filtering, a good estimate is required for the size of the block over which the local median is computed. If the block size is smaller than half the size of the smallest object present, then the computed background will be effected by the object. On the other hand, if the size of the block is too large, then the local information about the background will be lost.

Using the basis of polynomials to model the background, we observe the Runge phenomenon. In order to mitigate the Runge phenomenon, one needs 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. A Comparison of the proposed method with the state-of-the-art method using the basis of polynomials is provided in Section 3.

In the context of image processing, the above problem is know as image inpainting. Several methods and algorithms for image inpainting are available. The inpainting algorithms proposed by Bertalmio et al. and Criminisi et al. are based on continuation of the isophote lines [3, 4, 12]. The inpainting algorithms proposed by Elad et al. is based on morphological component analysis

[14]. These method also requires a proper choice of dictionaries that can suitably represent different contents of the image. However, inpainting algorithms are suitable for recovering a visually pleasing image. Ma et al. demonstrate that image inpainting can be done via a matrix completion approach. In this approach, a low rank matrix is sought that agrees with the available samples and have small nuclear norm [27].

Application 1.2. *Weather information is generally measured using distributed sensors over different locations. Failure of the sensors, or the network, leads to missing information in the data set. Weather information, like temperature, collected over a period of time from different weather stations admits a very low rank approximation, and for each station the data vary smoothly with the time. Algorithms based on matrix completion have been shown useful to estimate missing information for this type of data set, see [9].*

We also demonstrate the performance of the proposed algorithm on a weather data set, see Section 3. We used a weather data set that consists of temperature measurements of 20 cities observed every half an hour for a period of two days. A similar data set with temperature measurement was used in [9]. In [9] a rank 2 approximation of the temperature measurement is obtained by using the nuclear norm of the approximation for regularization. As we shall see in Section 3, smoothness of left approximants is a better a priori information for such data set compared to the boundedness of the nuclear norm.

Some other applications that can benefit from the proposed method includes:

Application 1.3. *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 [35, 24].*

Application 1.4. *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 [18, 7] for further details. In this case, important a priori information about the smoothness of the final image is usually available.*

Application 1.5. *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 [45]. Frequently, a priori information about the spectral concentration of the signals is known.*

1.5. Previous Work

In 1970, Christoffersson considered the problem of computing a rank one ($R = 1$) approximation of a data set with missing values, see [11]. In 1974, Ruhe described a method to compute a low rank ($R \geqslant 1$) approximation of a data set with missing values [37]. 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 [47]. However, like power iteration methods, NIPALS is inadequate for data sets with close singular values, for a detailed analysis see [15]. 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 [16]. Unlike the method of Ruhe [37], the method by Gabriel, and Zamir [16] 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.

For the purpose of reference, now we briefly describe the approach of Gabriel, and Zamir for estimating low rank approximation of a weighted data set as formulated in Equation 3. The presentation is similar to the one in [16]. The method starts with an initial approximation $\mathbf{U}_0$ of the left approximants, $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_R)$. 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 $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_R)$, 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 ith step, for computing $\mathbf{V}_i$ given $\mathbf{U}_{i-1}$, the problem posed in Equation (3) reduces to the following problem:

$$\begin{aligned}
\tilde{\mathbf{Y}} &= \underset{\mathbf{Y}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{U}_{(i-1)}\mathbf{Y}^*\|_W^2. \\
&= \underset{\mathbf{Y}}{\operatorname{argmin}} \sum_{m=1}^{M}\sum_{n=1}^{N} \mathbf{W}(m,n)\left(\mathbf{A}(m,n) - \sum_{r=1}^{R}\mathbf{U}_{(i-1)}(m,r)\mathbf{Y}^*(r,n)\right)^2, \\
&= \underset{\mathbf{Y}}{\operatorname{argmin}}\ F(\mathbf{Y}). \qquad (5)
\end{aligned}$$

The normal equations of the quadratic optimization problem posed in Equation (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:

$$\mathbf{U}_{(i-1)}^*\mathtt{diag}[\mathbf{W}(:,n)]\mathbf{U}_{(i-1)}\mathbf{Y}^*(:,n) = \mathbf{U}_{(i-1)}^*\mathtt{diag}[\mathbf{W}(:,n)]\mathbf{A}(:,n), \qquad (6)$$

where $n = 1, \ldots, N$. The notation $\mathbf{A}(:,n)$, and $\mathbf{W}(:,n)$ represent the nth column of $\mathbf{A}$, and the nth 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 $\mathbf{Y}^*(:,n)$, $n = 1, \ldots, N$, from Equation (6). Finally, we compute $\mathbf{V}_i$, by orthonormalizing $\tilde{\mathbf{Y}}$, and obtain $\mathbf{\Sigma}_i$ as the normalizing factor.

Next, within the ith iteration $\mathbf{U}_i$ is obtained similarly, by regressing rows of $\mathbf{A}$ onto $\mathbf{V}_i$, and orthonormalizing.

This way of solving bilinear systems by successively regressing columns and rows of data matrix is also known as *criss-cross* regression. This type of iterative procedure for minimizing the objective function jointly over all variables by alternating restricted minimization over disjoint subset (block) of variables is commonly known as alternating optimization. For further details related to convergence of alternating optimization, see [5, 6]. For some application of alternating optimization, see [22, 13, 23].

Unlike the cost function for low rank approximation with equally weighted data (2), the cost function with weighted data (3) has local minima, see [41]. 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 [20]. 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. In 2007, Raiko et al. proposed a method [33] in which the penalty term in the cost function (4) is proportional to the norm of the approximants. In the same paper, it is shown that regularizing with a priori information that the data points follow a normal distribution is equivalent to adding a norm penalty in the cost function. That is $\mathbf{B}_u$, and $\mathbf{B}_v$ are set to the identity. More precisely, Raiko et al. [33] solves the following optimization problem:

$$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \underset{\mathbf{U},\mathbf{\Sigma},\mathbf{V}}{\operatorname{argmin}} \left\{ \|\mathbf{A} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_W^2 + \frac{\alpha_u}{2}\|\mathbf{U}\|_F^2 + \frac{\alpha_v}{2}\|\mathbf{V}\|_F^2 \right\}. \tag{7}$$

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, [32], Buchanan et al., 2005 [7], for regularization. For a comprehensive discussion of the available methods to obtain a norm regularized low rank approximation in case of missing data, see the work by Ilin, and Raiko, 2010, [20], and references therein. However, regularization with norm penalty is trivial in the framework of alternating regression, because the unknown variables are separable at each stage of the iteration.

Regularization based on a priori information that there are only a few singular values is also investigated by researchers, see [10, 9, 27]. In this approach, the number of non-zero singular value, equivalently the rank, or the counting "norm" of the singular values is used for regularization. Objective function involving counting "norm" of the singular values leads to intractable algorithms for large data sets. In order to design a tractable algorithm, the counting "norm" of the singular value is replaces by its convex relaxation, which is the nuclear norm of the matrix, equivalently the sum of singular values. More precisely, the following problems is solved:

$$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \underset{\mathbf{U},\mathbf{\Sigma},\mathbf{V}}{\operatorname{argmin}} \left\{ \|\mathbf{A} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_W^2 + \lambda \ \operatorname{tr}(\mathbf{\Sigma}) \right\}, \tag{8}$$

here, $\operatorname{tr}(\mathbf{\Sigma})$ denotes the trace of the matrix $\mathbf{\Sigma}$, i.e. the sum of all the diagonal elements of the matrix $\mathbf{\Sigma}$.

Regularization with the a priori information that the approximants have few significant entries is also used in some applications [48, 40, 46]. 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. Alike regularization using the counting "norm" of the singular values, regularization using the counting "norm" of the approximants (singular vectors) also leads to intractable algorithms for large data sets. In order to design a tractable algorithm, the counting "norm" of the approximants is replaced by its convex relaxation, which is the ℓ_1 norm of the approximants. More precisely, within this framework, the following optimization problem is solved:

$$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \underset{\mathbf{U},\mathbf{\Sigma},\mathbf{V}}{\operatorname{argmin}} \left\{ \|\mathbf{A} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_F^2 + \frac{\alpha_u}{2}\|\mathbf{U}\|_1 + \frac{\alpha_v}{2}\|\mathbf{V}\|_1 \right\}, \tag{9}$$

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 [34], 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 [36]. 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. The problem of estimating smooth principal components for weighted data sets is first formulated by Markovsky and Niranjan 2010 [28]. An outline of a method similar to the one presented in this paper is also proposed in [28]. However, in [28] no algorithmic details or numerical results demonstrating the quality of the smooth principal components is presented.

In last few paragraphs, we try to summarize some related previous works on computing regularized low rank approximation using different a priori information. However, there are several other works on low rank approximation of data matrices where different criterion and constraints (other than the one described above) are used to obtain a regularized solution. For example, low rank approximation with non-negative (elementwise) factors is studied by Paatero and Tapper 1994 [30], Lee and Seung 1999 [25, 26], Berry et al. [2].

1.6. Matrix Procrustes Problem

As we shall see in Section 2, in the framework of the proposed method, within each block of an iteration, we solve a problem (14) similar to the matrix Procrustes problem. Given two matrices $\mathbf{A}$ and $\mathbf{B}$, the matrix Procrustes problem [39] is posed as follows:

$$\underset{\mathbf{Q}:\mathbf{Q}^*\mathbf{Q}=\mathbf{I}}{\operatorname{argmin}} \|\mathbf{A}-\mathbf{B}\mathbf{Q}\|_F^2. \tag{10}$$

The above problem can be solved efficiently using singular value decomposition, see [39, 19]. For further variants of the matrix Procrustes problem, see [19].

Although, the problem that we solve (14) is similar to matrix Procrustes problem, there are certain differences between the problem that we solve (14) and the matrix Procrustes problem (10). First, in the proposed framework the regressors ($\mathbf{U}_{i-1}$ in (14)) are orthonormal. Second, a weighted norm is used instead of Frobenius norm. Third, depending on a priori information there are associated penalty terms. In Section 2, we describe a method to solve (14).

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 (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. Finally, 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 (4). However, for this case of non-weighted data set, we replace the weighted norm in the cost function of Equation (4) by the Frobenius norm. Denoting the rank by R, we consider the following problem:

$$[\mathbf{U},\mathbf{\Sigma},\mathbf{V}] = \underset{\mathbf{U},\Sigma,\mathbf{V}}{\operatorname{argmin}} \left\{ \|\mathbf{A}-\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_F^2 + \frac{\alpha_u}{2}\|\mathbf{B}_u\mathbf{U}\|_F^2 + \frac{\alpha_v}{2}\|\mathbf{B}_v\mathbf{V}\|_F^2 \right\}, \tag{11}$$

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

We consider the same iterative block coordinate (alternating optimization) approach used to solve the problem posed in (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, $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_R)$, 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 ith step, for computing $\mathbf{V}_i$ from $\mathbf{U}_{i-1}$, the problem is posed as follows:

$$\begin{aligned}
\tilde{\mathbf{Y}} &= \underset{\mathbf{Y}}{\operatorname{argmin}} \left\{ \|\mathbf{A} - \mathbf{U}_{(i-1)}\mathbf{Y}^*\|_F^2 + \frac{\alpha_v}{2}\|\mathbf{B}_v\mathbf{Y}\|_F^2 \right\}, \\
&= \underset{\mathbf{Y}}{\operatorname{argmin}} \left\{ \|\mathbf{U}_{(i-1)}^*\mathbf{A} - \mathbf{Y}^*\|_F^2 + \frac{\alpha_v}{2}\|\mathbf{B}_v\mathbf{Y}\|_F^2 \right\}, \\
&= \underset{\mathbf{Y}}{\operatorname{argmin}} \left\| \begin{pmatrix} \mathbf{A}^*\mathbf{U}_{(i-1)} \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{I} \\ \sqrt{\frac{\alpha_v}{2}}\mathbf{B}_v \end{pmatrix} \mathbf{Y} \right\|_F^2 . \qquad (12)
\end{aligned}$$

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 (12) 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:

$$\left(\mathbf{I} + \frac{\alpha_v}{2}\mathbf{B}_v^*\mathbf{B}_v\right)\mathbf{Y} = \mathbf{A}^*\mathbf{U}_{(i-1)}. \qquad (13)$$

We note that Equation (13) 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 (13). Finally, we compute $\mathbf{V}_i$ by orthonormalizing the estimate of $\tilde{\mathbf{Y}}$, and obtain $\mathbf{\Sigma}_i$ as the normalizing factor.

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.

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 (4). We consider the same approach of alternating optimization as we did for solving the problems posed in Equations (3) and (11), 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, $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_R)$, 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 ith step, for computing $\mathbf{V}_i$ given $\mathbf{U}_{i-1}$, the problem posed in Equation (4) reduces to the following problem:

$$\tilde{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmin}} \left\{ \left\| \mathbf{A} - \mathbf{U}_{(i-1)}\mathbf{Y}^* \right\|_W^2 + \frac{\alpha_v}{2} \left\| \mathbf{B}_v \mathbf{Y} \right\|_F^2 \right\}. \tag{14}$$

Unlike Equation (12), Equation (14) is not separable in general, because here the Frobenius norm is replaced by the weighted norm, and therefore the algebraic manipulations done in Equation (12) 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 [7] discuss the necessity of going beyond the regularization with a norm penalty. They also observe that the variables of Equation (14) are not separable, and therefore it is not easy to proceed with the alternating iterative method, see the section on discussion and conclusions in [7]. 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 (14) is expressed as follows:

$$\begin{aligned}
& \left\| \mathbf{A} - \mathbf{U}_{(i-1)}\mathbf{Y}^* \right\|_W^2 \\
= & \left\| \mathbf{W}_{\frac{1}{2}} \circ \mathbf{A} - \mathbf{W}_{\frac{1}{2}} \circ \left[\mathbf{U}_{(i-1)}\mathbf{Y}^* \right] \right\|_F^2 \\
= & \sum_{n=1}^{N} \left\| \left[diag\left(\mathbf{W}_{\frac{1}{2}}(:,q) \right) \right] \mathbf{A}(:,q) - \left[diag\left(\mathbf{W}_{\frac{1}{2}}(:,q) \right) \right] \left[\mathbf{U}_{(i-1)}\mathbf{Y}^*(:,q) \right] \right\|_2^2 \\
= & \left\| \left[diag\left(\mathbf{W}_{\frac{1}{2}}(:,:) \right) \right] \mathbf{A}(:,:) - \left[diag\left(\mathbf{W}_{\frac{1}{2}}(:,:) \right) \right] \left[\mathbf{I} \otimes \mathbf{U}_{(i-1)} \right] \left[\mathbf{Y}^*(:,:) \right] \right\|_2^2 \\
= & \left\| \mathbf{W}_d \vec{\mathbf{A}} - \mathbf{W}_d \left[\mathbf{I} \otimes \mathbf{U}_{(i-1)} \right] \vec{\mathbf{Y}}^* \right\|_2^2.
\end{aligned} \tag{15}$$

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 $n = 1, \ldots, N$, $m = 1, \ldots, M$, and $r = 1, \ldots, R$. 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:

$$\begin{aligned}
\frac{\alpha_v}{2} \|\mathbf{B}_v \mathbf{Y}\|_F^2 = \frac{\alpha_v}{2} \sum_{n=1}^{N} \sum_{r=1}^{R} \left| \sum_{k=1}^{N} \mathbf{B}_v(n,k)\mathbf{Y}(k,r) \right|^2 &= \frac{\alpha_v}{2} \sum_{n=1}^{N} \sum_{r=1}^{R} \left| \sum_{k=1}^{N} \overline{\mathbf{B}_v(n,k)} \ \overline{\mathbf{Y}(k,r)} \right|^2 \\
&= \sum_{t=1}^{RN} \left(\sum_{s=1}^{RN} \ddot{\mathbf{B}}_v(t,s) \vec{\mathbf{Y}}^*(s) \right)^2 \\
&= \left\| \ddot{\mathbf{B}}_v \vec{\mathbf{Y}}^* \right\|_2^2.
\end{aligned} \tag{16}$$

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

$$\ddot{\mathbf{B}}_v\left((n-1)R+r_a,(k-1)R+r_b\right)=\begin{cases}\sqrt{\frac{\alpha_v}{2}} & \overline{\mathbf{B}_v((q-1)+r_a,k)} \quad \text{if} \quad r_a=r_b,\\ 0 & \text{otherwise,}\end{cases} \tag{17}$$

where $n,k=1,\ldots,N$, and $r_a,r_b=1,\ldots,R$. With this formulation one can also consider different regularization for different approximants. In that case, for $r=1,\ldots,R$, 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:

$$\ddot{\mathbf{B}}_v\left((n-1)R+r_a,(k-1)R+r_b\right)=\begin{cases}\sqrt{\frac{\alpha_v^r}{2}} & \overline{\mathbf{B}_v^r((q-1)+r_a,k)} \quad \text{if} \quad r_a=r_b=r,\\ 0 & \text{otherwise.}\end{cases} \tag{18}$$

The optimization problem posed in Equation (14), is now expressed with the modified form of the cost functions, derived in Equation (15) and (16), as follows:

$$\begin{aligned}\tilde{\vec{\mathbf{Y}}} &= \underset{\vec{\mathbf{Y}}}{\operatorname{argmin}}\left\{\left\|\mathbf{W}_d\vec{\mathbf{A}}-\mathbf{W}_d\left[\mathbf{I}\otimes\mathbf{U}_{(i-1)}\right]\vec{\mathbf{Y}}^*\right\|_2^2+\left\|\ddot{\mathbf{B}}_v\vec{\mathbf{Y}}\right\|_2^2\right\}\\ &= \underset{\vec{\mathbf{Y}}}{\operatorname{argmin}}\left\|\begin{pmatrix}\mathbf{W}_d\vec{\mathbf{A}}\\ 0\end{pmatrix}-\begin{pmatrix}\mathbf{W}_d\left[\mathbf{I}\otimes\mathbf{U}_{(i-1)}\right]\\ \ddot{\mathbf{B}}_v\end{pmatrix}\vec{\mathbf{Y}}^*\right\|_2^2.\end{aligned} \tag{19}$$

We note that the above formulation also appears in the appendix of [28].

2.3. 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 to set proper boundary values while constructing $\mathbf{D}$. We note that $\mathbf{D}$ is a banded matrix, therefore by construction (18) $\ddot{\mathbf{D}}_u$ is also banded.

2.4. Sketch of the Algorithm

The structure and sparsity of the associated matrices in the least square problem derived in Equation (19), and using a roughness penalty, is pictorially illustrated in Figure 1. We note that with this formulation, one can adopt different regularization for different approximants as shown in Equation (18), while preserving sparsity.

At iteration i, our first objective is to compute $\mathbf{V}_i$, given $\mathbf{U}_{i-1}$. A system of normal equations corresponding to problem posed in (19) 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:

$$\left(\left[\mathbf{I}\otimes\mathbf{U}_{(i-1)}^*\right]\mathbf{W}_d^2\left[\mathbf{I}\otimes\mathbf{U}_{(i-1)}\right]+\ddot{\mathbf{B}}_v^*\ddot{\mathbf{B}}_v\right)\vec{\mathbf{Y}}^*=\left[\mathbf{I}\otimes\mathbf{U}_{(i-1)}^*\right]\mathbf{W}_d^2\vec{\mathbf{A}}. \tag{20}$$

The coefficient matrix of the normal equation (20) is pictorially represented in Figure 2, where the sparsity of the associated matrices are also illustrated. We note that the coefficient matrix of the normal Equations (20) 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.

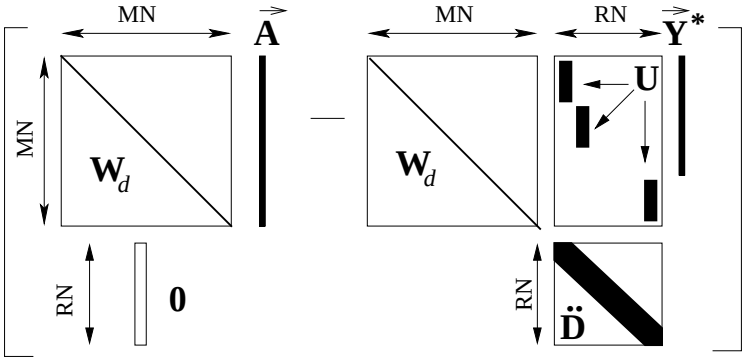


Figure 1: Pictorial representation of the cost function, illustrating the sparsity structure.

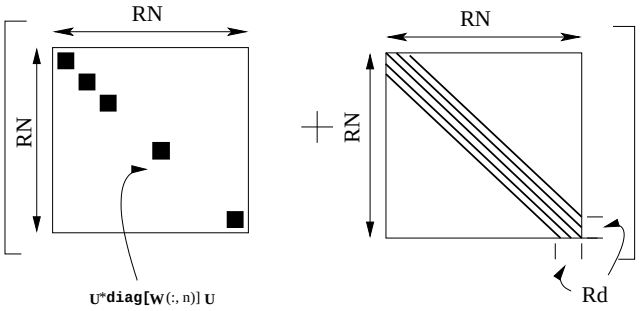


Figure 2: Pictorial representation of the coefficient matrix of the normal equation, illustrating the sparsity.

In order to compute orthonormal $\mathbf{V}_i$ in the this (i.e., ith) iteration, we first compute $\mathbf{Y}$ given $\mathbf{U}_{(i-1)}$, by solving (20) for $\vec{\mathbf{Y}}^*$, and rearranging the terms such that $\mathbf{Y}(n,r) = \left(\vec{\mathbf{Y}}^*(nR+r)\right)^*$. Hence, $\mathbf{Y}$ is optimal considering the weighted norm and the penalty term, i.e., $\mathbf{Y}$ is optimal for the problem posed in (14). Next, we consider an economy size SVD of $\mathbf{Y}$:

$$\mathbf{Y} = \mathbf{U}_y \mathbf{\Sigma}_y \mathbf{V}_y^*, \tag{21}$$

and we update the matrix $\mathbf{V}_i$ by setting it to $\mathbf{U}_y$, and we update the matrix $\mathbf{\Sigma}_i$ by setting it to $\mathbf{\Sigma}_y$. We note that at this point of the iteration the best estimate of the left approximant is the orthonormal matrix $\mathbf{U}_{i-1}\mathbf{V}_y^*$. At this stage of the iteration, the objective function is minimized along the direction of right approximants, and a new left approximant, a new right approximant and a new set of singular values are computed. However, in the second half of the iteration we discard the current estimate of the left approximate, i.e., $\mathbf{U}_{i-1}\mathbf{V}_y^*$, and estimate the left approximate from scratch using the latest right approximant $\mathbf{V}_i$. For computing the right approximant $\mathbf{U}_i$ from $\mathbf{V}_i$ we use the equation (19) by symmetrically replacing $\mathbf{U}_{i-1}$ by $\mathbf{V}_i$. In this second part of the iteration, $\mathbf{\Sigma}_i$ is already computed, thus in order to economize on computation, we can avoid the SVD and compute $\mathbf{U}_i$ just by orthonormalizing the result. 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 1, we call this routine `BIRSVD`. In case of equally weighted (non-weighted) data, the routine `BIRSVD` can be used efficiently by setting all the entries of the input weight matrix $\mathbf{W}$ to 1. Also, in case no regularization is required, the routine `BIRSVD` can be used efficiently by setting α_u and α_v to $\mathbf{0}$.

2.5. 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 (19) is an overdetermined least square problem with the following coefficient matrix:

$$\begin{pmatrix} \mathbf{W}_d \left[\mathbf{I} \otimes \mathbf{U}_{(i-1)}\right] \\ \ddot{\mathbf{B}}_v \end{pmatrix}. \tag{22}$$

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 $\ddot{\mathbf{B}}_v$ (17) 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 (22), the coefficient matrix and its Hermitian transpose can be applied to a vector with $\mathcal{O}(RM + RN))$ operations only. Hence an iterative method like LSQR [31] designed for overdetermined least square problems, can be used to solve Equation (19). With the regularization factor, the method works like damped LSQR, and the resulting semi convergence is very mild.

Algorithm 1 `BIRSVD` – Bi Iterative Regularized SVD with weighted data

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

$[\mathtt{M}, \mathtt{N}] = \mathtt{size}(\mathbf{A})$

$\mathbf{A}_w = \mathbf{W} .* \mathbf{A}$

$\mathbf{U} \leftarrow$ `any isometric matrix with, N rows and R columns;`

Compute $\ddot{\mathbf{B}}_u$ from $\mathbf{B}_u$ `// Equation (17); in practice, access entries`

Compute $\ddot{\mathbf{B}}_v$ from $\mathbf{B}_v$ `// of` $\ddot{\mathbf{B}}_u$, $\ddot{\mathbf{B}}_u$ `directly from` $\mathbf{B}_v$, $\mathbf{B}_v$

for $i = 1$ to `maxi` **do**

  $\mathbf{R} = \mathtt{reshape}(\mathbf{U}^* \mathbf{A}_w, \mathtt{NR}, 1)$

  $\mathbf{L} = [\ ]$ `// set empty`

  for $n = 1$ to `N` **do**

    $\mathbf{L} = \mathtt{blkdiag}(\mathbf{L}, \mathbf{U}^*((\mathtt{repmat}(\mathbf{W}(:, n), 1, \mathtt{R}) .* \mathbf{U})))$

  end for

  $[\mathbf{L}_v, \mathbf{L}_v^*] = \mathtt{chol}(\mathbf{L} + \ddot{\mathbf{B}}_v)$

  $\mathbf{Y} = \mathbf{L}_v^* \setminus (\mathbf{L}_v \setminus \mathbf{R})$

  $\mathbf{Y} = \mathtt{reshape}(\mathbf{Y}, \mathtt{R}, \mathtt{N})$

  $[\mathbf{V}, \mathbf{\Sigma}] = \mathtt{svd}(\mathbf{Y}, 0)$ `// economy size SVD`

  $\mathbf{R} = \mathtt{reshape}(\mathbf{V}^* \mathbf{A}_w^*, \mathtt{MR}, 1)$

  $\mathbf{L} = [\ ]$ `// set empty`

  for $m = 1$ to M **do**

    $\mathbf{L} = \mathtt{blkdiag}(\mathbf{L}, \mathbf{V}^*((\mathtt{repmat}(\mathbf{W}^*(:, m), 1, \mathtt{R}) .* \mathbf{V})))$

  end for

  $[\mathbf{L}_u, \mathbf{L}_u^*] = \mathtt{chol}(\mathbf{L} + \ddot{\mathbf{B}}_u)$

  $\mathbf{X} = \mathbf{L}_v^* \setminus (\mathbf{L}_v \setminus \mathbf{R})$

  $\mathbf{X} = \mathtt{reshape}(\mathbf{X}, \mathtt{R}, \mathtt{M})$

  $[\mathbf{U}, \mathbf{R}] = \mathtt{qr}(\tilde{\mathbf{X}}, 0)$ `// economy size QR`

end for

return $\mathbf{U}\ \mathbf{\Sigma}\ \mathbf{V}$

`// reshape() - reorder entries with given dimensions`

`// blkdiag() - concatenated matrices block diagonally`

`// repmat() - make multiple copy of matrices`

2.6. 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 left approximants $\mathbf{U}$, and vice-versa. Here 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 (17) or (18), 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 (17) or (18). 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 (20).
- Computation of right hand side vector of Equation (20).
- Cholesky factorization of the coefficient matrix.
- The forward and the backward substitution to solve Equation (20).
- Finally, orthogonalizing $\mathbf{Y}$ to get an estimate $\mathbf{V}$ for this iteration.

The coefficient matrix of Equation (20) is presented pictorially in Figure 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, the coefficient matrix is banded, and also symmetric positive definite, see Equation (20). 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 (20) requires $2RNM$ complex flops. The forward and the backward substitution for solving Equation (20) 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 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 (20), 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 (20) 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. We compare the results obtained using the proposed `BIRSVD` algorithm with the results obtained using the `FPCA` algorithm developed by Ma et al. [27]. For FPCA algorithm we use the software accompanying [27]. We note that, although an outline of a method for computing smooth low rank approximation of weighted data sets is presented in [28], the software accompanying [28] do not have an implementation of the method.

We present results obtained from two particular data sets. The first data set is an image with missing pixel values. The second data set is based on temperature measurements of 20 cities over two days sampled every half an hour. These two type of data sets are selected because of the following complementary properties they have as a data set:

- Astronomical image data we considered for illustration in this section (2041×1649) is much larger than the weather data set we considered (96×20).
- Each pixel measurement in the astronomical image is available as 32 bit IEEE 754 floating point number, whereas the temperature measurement of the weather data is available only with 3 significant digits.
- The weather data is complete, and incomplete weather data set is generated by random sampling of specified amount of data points. Hence, with this data set one can compute the fidelity of the approximation over data points that were not used for approximation.

Regularized low rank approximation using `BIRSVD` algorithm were done on many other data sets, and the results obtained are in accordance with the presented ones.

3.1. Astronomical Image Background

3.1.1. Test Data: Wide Field Astronomical Image

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) to differentiate the pure background part of the image from the celestial objects. For astronomical images, such mask are easy to construct because of the following reasons:

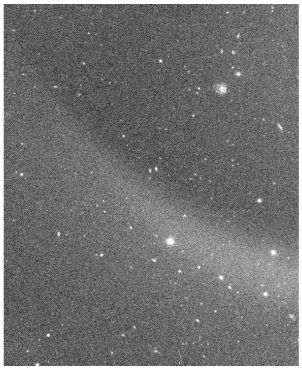
- celestial objects have much higher magnitude than the background, hence a rough separation of objects from the background can be done. Further, the object size is increased using morphological dilation in order to eliminate even the slightest halo of the objects from the backgrounds.
- precise astronomical catalogues with the position of the celestial objects are already available, hence reliable separation of the objects is possible.
- defective pixel maps with information regarding hot and cold pixels are readily available for astronomical imaging instruments.

After the pure background part is identified, 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 background 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.

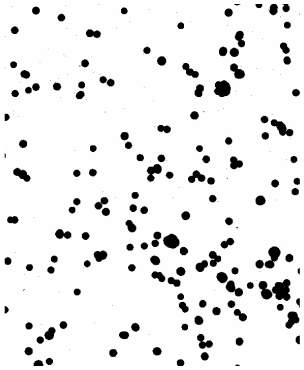
Instead of reconstructing the background pixel intensities for the whole image as a polynomial in two variables, we estimated the background pixel intensities for the whole image by a regularized low rank approximation of the pure background pixel intensity data. Since the pixels corresponding to the 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 [38]. The image is shown in Figure 3(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



(a) An astronomical image with a varying background



(b) A binary weight file differentiating between bright objects and pure sky background

Figure 3: Data set, and corresponding binary weights

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(a) is shown in Figure 3(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.

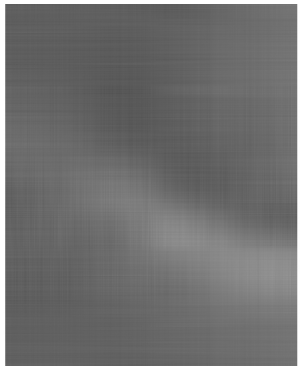
3.1.2. Reconstructed Backgrounds

For the purpose of comparison, we estimated the background by fitting tensor products of the Legendre polynomials to the pure background pixel intensities. Figure 4(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.

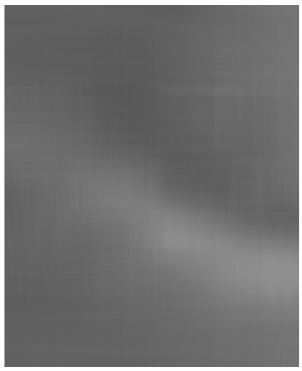
Next, also for the purpose of comparison, we estimated the background as a low rank approximation of the pixel intensity data, Figure 3(a), with weights shown in Figure 3(b) and no regularization. For this purpose, we used the routine `BIRSVD` without regularization, i.e., α_u, α_v are set to $\mathbf{0}$. The resulting background with a rank 4 approximation is shown in Figure 4(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, without regularization, the routine `BIRSVD` converges to a local minimum, see the next subsection for more detail.

Another possible approach to obtain the low rank approximation without any regularization

(a) A background estimated by least square fitting of tensor products of Legendre polynomials.



(b) A background estimated by low rank approximation of the data weighted data set



(c) A background estimated by a low rank approximation of the weighted data set, regularized with a second order difference of accuracy 2

(d) A background estimated by a low rank approximation of the weighted data set, regularized with a second order difference of accuracy 8

Figure 4: Sky background estimated using polynomials, and low rank approximations

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 [16]. 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 [38] 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(a), with weights shown in Figure 3(b). Figure 4(c) and 4(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 `BIRSVD` algorithm for this purpose, see Algorithm 1.

As the final step, we correct the given image, Figure 3(a), by subtracting the background from it. Figure 5 shows the results after a background correction. Figure 5(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 5(b) shows the background corrected image, where the background was computed as a rank 4 approximation of the background pixel intensities shown in Figure 4(b). Figure 5(c) and Figure 5(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 4(c), and accuracy 8 shown in Figure 4(d), respectively. Comparing Figure 5(a) with Figure 5(d), we notice a significant improvement achieved by the proposed method in correcting the background.

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 in this example can be found online at [38].

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.

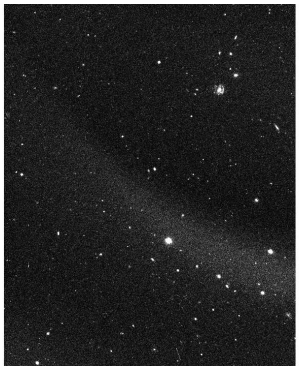
3.2. Alleviation of Local Minima

In this subsection we present our observations regarding the effectiveness of the proposed method in order to alleviate local minima. We test the proposed `BIRSVD` algorithm for regularized low rank approximation of weighted data, against non-regularized low rank approximation of weighted data [16], on the same data set $1,000$ times, but with a random starting point. We notice that each time the non-regularized algorithm converges to a different point, whereas the proposed Algorithm `BIRSVD` with regularization 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 1, we show the first 4 singular values obtained with different starting points by `BIRSVD` algorithm, without and with regularization. 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 if no regularization is done. On the other hand, `BIRSVD` algorithm with regularization 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 `BIRSVD` algorithm with regularization not only minimizes the error on the data, but also includes terms related to the roughness penalties. Especially for this particular data set, we observed that the unregularized method tends to fit some bright pixels corresponding to undetected faint objects. On the other hand, the regularized method was immune to such undetected faint objects.

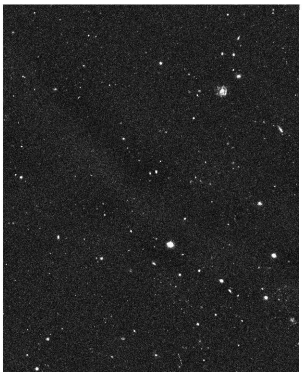
3.3. Speed of Convergence

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 6(a) shows the rate of convergence for different ranks of the approximation. The horizontal axis shows



(a) Background corrected with the polynomial based sky, Figure 4(a)

(b) Background corrected with weighted low rank approximation of the image, Figure 4(b)



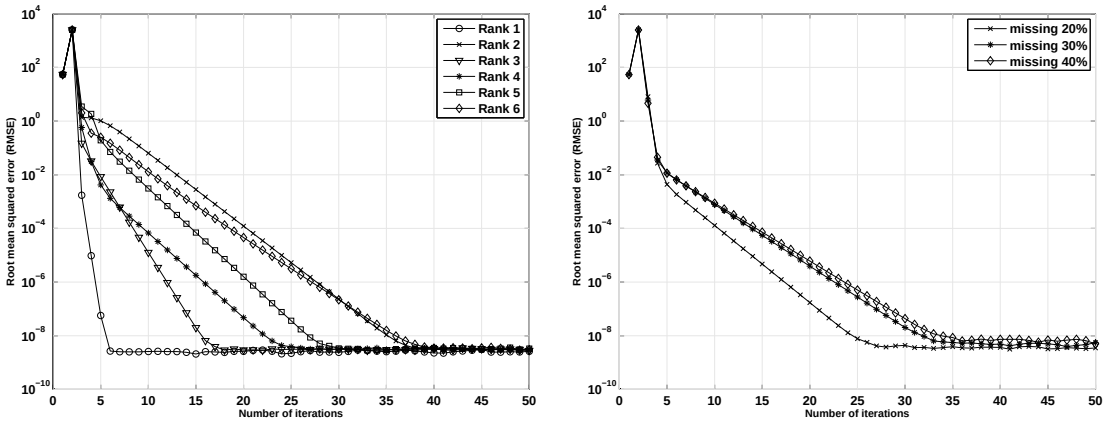
(c) Background corrected with regularized and weighted low rank approximation of the image, Figure 4(c)

(d) Background corrected with regularized and weighted low rank approximation of the image, Figure 4(d)

Figure 5: Background corrected images

Algorithm `BIRSVD` with no regularization					
Exp. no.	σ_1	σ_2	σ_3	σ_4	$\frac{\|\mathbf{A}-\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_W^2}{\|\mathbf{A}\|_W^2}$
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
Algorithm `BIRSVD` with regularization					
$1 - 1,000$	4.5040 E06	1.6688 E04	1.1915 E04	5.9132 E03	1.4981 E − 02

Table 1: Singular Values of Rank 4 approximation, and approximation error



(a) Rate of convergence for different rank approximations (b) Rate of convergence for different amount of missing values

Figure 6: Rate of convergence

the number of iterations, and the vertical axis shows the root mean squared error (RMSE) between successive iterations, i.e.

$$\text{RMSE}_i = \frac{1}{\sqrt{MN}} \left\| \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^* - \mathbf{U}_{(i-1)} \mathbf{\Sigma}_{(i-1)} \mathbf{V}_{(i-1)}^* \right\|_F \tag{23}$$

All curves in Figure 6(a) are generated with the same data set shown in Figure 3, 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 Rth 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 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 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 6(b). For all the curves in Figure 6(b), 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 2 do not justify using a rank higher than $R = 4$.

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 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

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 2: Slowness of convergence for different rank of 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 4.

3.4. Weather Data

3.4.1. Data: Temperature of 20 *Cities*

Our second data set consists of temperature measurements of 20 European cities, sampled every half an hour over two days (Aug. 16-17, 2011). Each column of the data set represents the temperature information of one of the 20 cities, thus, the dimension of this data set is 96×20. A similar data set is considered in [9]. Unlike the last example on astronomical image data, in this case, we knew all the 96×20 entries of this data set. For this data set, a rank 2 approximation contributes to 99.8% of the total matrix norm. Similar observation is reported in [9]. From the complete data set on temperature measurements, we sampled at random some entries to form a data set with specific amount of missing entries.

For the temperature measurement data set, we can only make a priori information regarding smoothness of the left approximants, i.e. $\mathbf{U}$. This is because, the temperature measurement vary fairly smoothly over time in each station. On the other hand, no such smoothness assumption is realistic for the right singular vectors $\mathbf{V}$. Hence, in order to compute a regularized low rank approximation, we solved the following optimization problem:

$$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \underset{\mathbf{U},\mathbf{\Sigma},\mathbf{V}}{\operatorname{argmin}} \left\{ \|\mathbf{A} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\|_W^2 + \frac{\alpha_u}{2}\|\mathbf{B}_u\mathbf{U}\|_F^2 \right\}. \tag{24}$$

The above equation is equivalent to setting α_v to zero in Equation (4).

3.4.2. Out of Sample Fidelity

We computed a regularized rank 2 approximation of the temperature measurement data set using the proposed `BIRSVD` algorithm, where we regularize using a priori information that the left approximants, i.e., column of $\mathbf{U}$, are smooth. For the purpose of comparison, we also computed another regularized low rank approximation of the temperature measurement data set using the method proposed in [27], where regularization is done by penalizing the nuclear norm of the approximating matrix. The method presented in [27] is called Fixed Point Continuation with Approximate

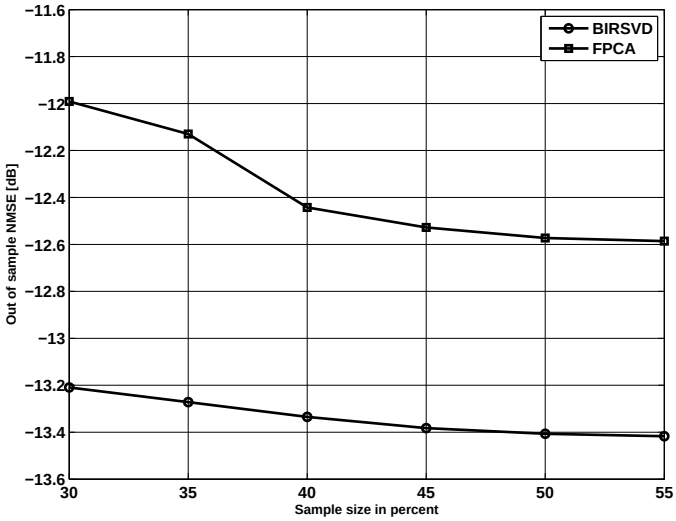


Figure 7: Fidelity of the reconstruction on out of sample measurements

SVD (`FPCA`). In [9], the low rank approximation of a weather data set was obtained by using the `FPCA` algorithm. An implementation of the `FPCA` algorithm available in the following link is used:

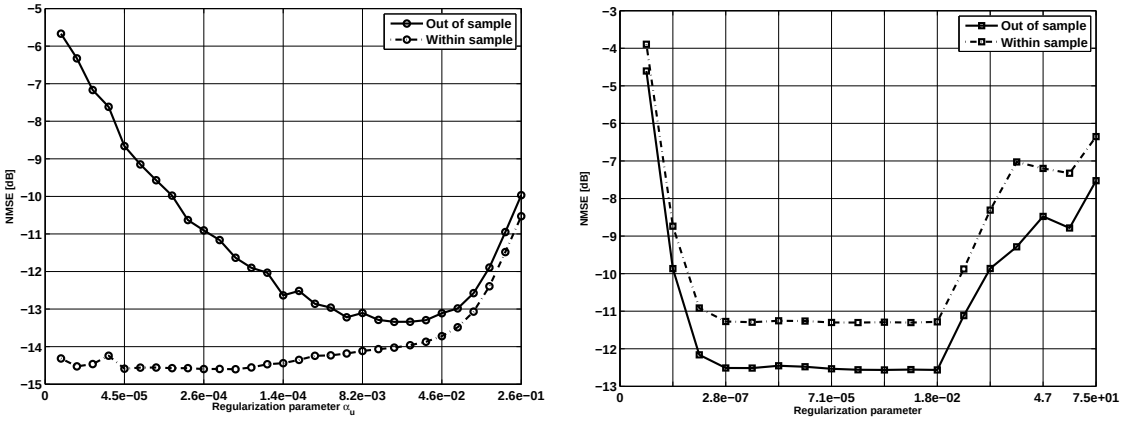
`http://www.columbia.edu/~sm2756/FPCA.htm`

The regularization parameter α_u for the proposed `BIRSVD` algorithm was set fixed to .006 for all the experiments varying over different amounts of missing information. On the other hand, as implemented in the `FPCA` software, the optimal regularization parameter for the `FPCA` algorithms was selected by inspecting the dimension of the data set, and the amount of missing information.

In Fig. 7 we report the dependence of the fidelity of the reconstruction on the amount of missing measurements. Since, we have the complete data set, we computed the out of sample fidelity of the reconstruction. The fidelity is measured in terms of the normalized mean square error (NMSE) over all the samples that were not considered in the reconstruction process. In Fig. 7 we present the out of sample NMSE obtained by the proposed `BIRSVD` algorithm, and the `FPCA` algorithm, for different amount of samples. Each of the reported NMSE is computed by averaging the results obtained over 100 random sampling patterns. The NMSE of a rank 2 approximation of the complete data set is -13.84 dB, and this number sets the floor of the reported NMSE. We note that the proposed `BIRSVD` algorithm consistently outperforms the `FPCA` algorithm. This is because, for the temperature measurement data set, smoothness of the left singular vector is a much better a priori information than boundedness of the nuclear norm.

3.5. Dependence on the Regularization Parameter

A desirable property of an algorithm that employ regularization is its insensitivity over a broad range of the regularization parameter. In Fig. 8(a) we demonstrate the dependence of the out of sample NMSE, and the within sample NMSE on the regularization parameter α_u as used in the



(a) BIRSVD (b) FPCA

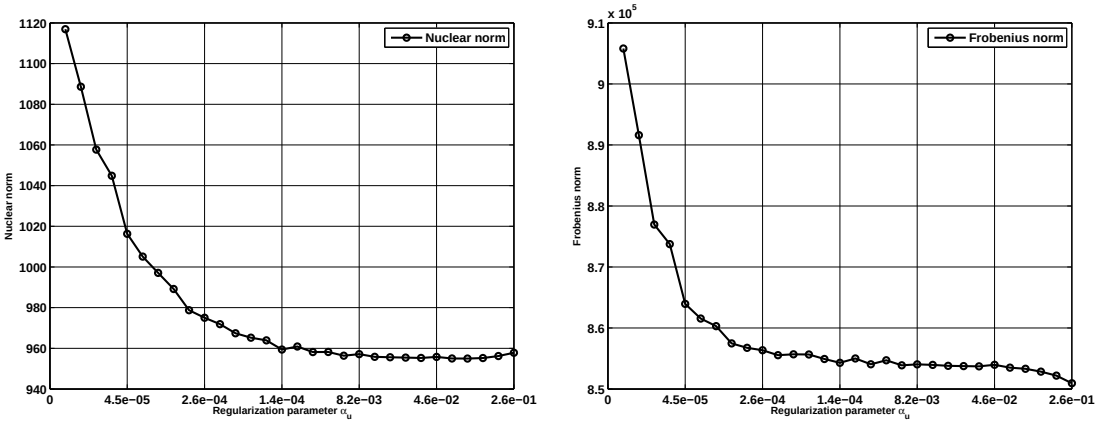
Figure 8: Dependence on regularization parameter

proposed `BIRSVD` algorithm. Each of the reported NMSE is computed by averaging the results obtained over 100 random sampling patterns that samples 30% of the whole data. Form Fig. 8(a) we notice that the out of sample NMSE is between -13 dB and -14 dB when the regularization parameter α_u changes from ca. $2.7e-04$ to ca. $2.7e-02$. Thus the performance of the proposed `BIRSVD` algorithm is insensitive over a wide range of the regularization parameter α_u. In Fig. 8(b) we present the dependence of the within sample NMSE, and the out of sample NMSE on the regularization parameter of the `FPCA` algorithm. In this figure, we note that the `FPCA` algorithm is insensitive to a wide range of the regularization parameter. This is a very pleasing feature of `FPCA` algorithm. However, comparing to Fig. 8(a) with Fig. 8(b) we notice that the out of sample NMSE obtained using the proposed `BIRSVD` algorithm is better then the out of sample NMSE obtained using the `FPCA` algorithm by 2 dB over a considerable range of regularization parameters.

In Fig. 9 we demonstrate how the nuclear norm and the Frobenius norm of the reconstructed matrix obtained using the proposed `BIRSVD` algorithm changes with the regularization parameter α_u. The experimental setup in this case is identical to the one used in the last section to demonstrate the dependence of the NMSE on the regularization parameter. From Fig. 8(a) and Fig. 9, we notice that over the range of regularization parameters α_u where we obtain consistent values of out of sample NMSE we also obtain consistent values of for the Frobenius norm and the nuclear norm. From the same figures, we also notice that a smaller values of the Frobenius norm or the nuclear norm does not guarantee a smaller value for the out of sample NMSE.

4. Conclusions

In this paper we proposed a fast method to compute a regularized low rank approximation of a weighted data set. Data sets with missing values were 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 regarding the smoothness of the approximants. To obtain smooth approximants, we used roughness penalty term in the cost function of the optimization problem. Unfortunately, this penalty term makes the cost



(a) Nuclear norm vs regularization parameter

(b) Frobenius norm vs regularization parameter

Figure 9: Nuclear norm and Frobenius norm as function of regularization parameter

function non separable. Nevertheless, we designed a tractable algorithm by 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 on two different type of data sets where a smoothness assumption can be made on the low rank approximants. The first data set was a wide field astronomical image, and the task was to reconstruct the smooth background of the image. 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. The second data set was on half-hourly temperature measurements from 20 different cities over two days. From this data set we sampled specific amount of data points to create a data set with missing value. Since the complete data was available, we were able to test the fidelity of proposed regularized approximation on the missing data. We compared the results obtained using the proposed `BIRSVD` algorithm with the results obtained using the `FPCA` algorithm, where regularization is done by penalizing the nuclear norm of the reconstructed matrix. The proposed `BIRSVD` algorithm produced a significant better result compared to the `FPCA` algorithm. However, in case reasonable a priori assumption regarding the approximants, like smoothness, is not available, then the `FPCA` is adequate.

References

[1] J. Bennett and S. Lanning, *The NETFLIX prize*, Proceedings of KDD Cup and Workshop, vol. 2007, Citeseer, 2007.

[2] M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, and R.J. Plemmons, *Algorithms and applications for approximate nonnegative matrix factorization*, Computational Statistics & Data Analysis **52** (2007), no. 1, 155–173.

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, Proceedings of the 27th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[4] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, *Simultaneous structure and texture image inpainting*, Image Processing, IEEE Transactions on **12** (2003), no. 8, 882–889.

[5] J. Bezdek and R. Hathaway, *Some notes on alternating optimization*, Advances in Soft ComputingAFSS 2002 (2002), 187–195.

[6] J.C. Bezdek and R.J. Hathaway, *Convergence of alternating optimization*, Neural, Parallel & Scientific Computations **11** (2003), no. 4, 351–368.

[7] 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.

[8] C. Buil, *IRIS: Astronomical Image-Processing Software*, Digital Astrophotography: The State of the Art (2005), 79–88.

[9] E.J. Candes and Y. Plan, *Matrix completion with noise*, Proceedings of the IEEE **98** (2010), no. 6, 925–936.

[10] E.J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Foundations of Computational Mathematics **9** (2009), no. 6, 717–772.

[11] A. Christoffersson, *The one component model with incomplete data, PhD thesis*, Uppsala University, Institute of Statistics, 1970.

[12] A. Criminisi, P. Pérez, and K. Toyama, *Region filling and object removal by exemplar-based image inpainting*, Image Processing, IEEE Transactions on **13** (2004), no. 9, 1200–1212.

[13] J. De Leeuw, *Block relaxation algorithms in statistics*, Information systems and data analysis (1994), 308–324.

[14] M. Elad, J.L. Starck, P. Querre, and D.L. Donoho, *Simultaneous cartoon and texture image inpainting using morphological component analysis (mca)*, Applied and Computational Harmonic Analysis **19** (2005), no. 3, 340–358.

[15] 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.

[16] K.R. Gabriel and S. Zamir, *Lower rank approximation of matrices by least squares with any choice of weights*, Technometrics (1979), 489–498.

[17] G.H. Golub and C.F. Van Loan, *Matrix computations*, Johns Hopkins Univ Pr, 1996.

[18] R. Hartley and F. Schaffalitzky, *PowerFactorization: 3D reconstruction with missing or uncertain data*, Australia-Japan advanced workshop on computer vision, Citeseer, 2003.

[19] N. Higham and P. Papadimitriou, *Matrix procrustes problems*, Tech. report, Tech. Rep., Department of Mathematics, University of Manchester, 1994.

[20] 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.

[21] I.T. Jolliffe, *Principal component analysis*, Springer verlag, 2002.

[22] H.A.L. Kiers, *An alternating least squares algorithm for parafac2 and three-way dedicom*, Computational statistics & data analysis **16** (1993), no. 1, 103–118.

[23] ———, *Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems*, Computational statistics & data analysis **41** (2002), no. 1, 157–170.

[24] 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.

[25] D.D. Lee and H.S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature **401** (1999), no. 6755, 788–791.

[26] ———, *Algorithms for non-negative matrix factorization*, Advances in neural information processing systems **13** (2001).

[27] S. Ma, D. Goldfarb, and L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Mathematical Programming **128** (2011), no. 1, 321–353.

[28] I. Markovsky and M. Niranjan, *Approximate low-rank factorization with structured factors*, Computational Statistics & Data Analysis **54** (2010), no. 12, 3411–3420.

[29] P. Massey, *A User's Guide to CCD Reductions with IRAF*, NOAO Laboratory **13** (1992), 14–15.

[30] P. Paatero and U. Tapper, *Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics **5** (1994), no. 2, 111–126.

[31] 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.

[32] A. Paterek, *Improving regularized singular value decomposition for collaborative filtering*, Proceedings of KDD Cup and Workshop, vol. 2007, Citeseer, 2007.

[33] 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.

[34] J.O. Ramsay and B.W. Silverman, *Functional data analysis*, Statistics and Computing **8** (1998), no. 4, 401–403.

[35] 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.

[36] 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.

[37] 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).

[38] Das S. and Neumaier N., *BIRSVD: Bi-Iterative Regularized Singular Value Decomposition*, `http://www.mat.univie.ac.at/~neum/software/birsvd`.

[39] P.H. Schönemann, *A generalized solution of the orthogonal procrustes problem*, Psychometrika **31** (1966), no. 1, 1–10.

[40] 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.

[41] N. Srebro and T. Jaakkola, *Weighted low-rank approximations*, MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-, vol. 20, 2003, p. 720.

[42] J.L. Starck and F. Murtagh, *Astronomical image and data analysis*, Springer, 2006.

[43] G. Strang, *Introduction to linear algebra*, Wellesley Cambridge Pr, 2003.

[44] L.N. Trefethen and D. Bau, *Numerical linear algebra*, Society for Industrial Mathematics, 1997.

[45] 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.

[46] 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.

[47] H. Wold and E. Lyttkens, *Nonlinear iterative partial least squares (NIPALS) estimation procedures*, Bull. Inst. Int. Stat. **43** (1969), 29–51.

[48] H. Zou, T. Hastie, and R. Tibshirani, *Sparse principal component analysis*, Journal of computational and graphical statistics **15** (2006), no. 2, 265–286.