Linköping University Post Print

**Grassmann algorithms for low rank
approximation of matrices with missing values**

Lennart Simonsson and Lars Elden

N.B.: When citing this work, cite the original article.

The final publication is available at www.springerlink.com:

Lennart Simonsson and Lars Elden, Grassmann algorithms for low rank approximation of matrices with missing values, 2010, BIT NUMERICAL MATHEMATICS, (50), 1, 173-191.

<http://dx.doi.org/10.1007/s10543-010-0253-9>

Copyright: Springer Science Business Media

<http://www.springerlink.com/>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-54404>

Grassmann Algorithms for Low Rank Approximation of Matrices with Missing Values

Lennart Simonsson & Lars Eldén

June 18, 2008

1 Background and Introduction

This paper concerns the problem of low rank approximation of a given matrix where the rank d is given beforehand by the user. This amounts to computing a pair of subspaces that are the (numerical) range spaces of the matrix A and A^T . Just as in the eigenvalue problem there is an inherent nonuniqueness in the basis chosen for at least one of the subspaces. Our approach, following [?], is to use Newton's method on the Grassmann manifold.

In statistics and data analysis one often wants to compress data given in a matrix into a matrix of considerably lower rank. In the statistics literature, e.g. [?], this is known as principal component analysis where one uses a few vectors to build a linear model of the data matrix. Let $m \times n$ matrix A be given with singular value decomposition (SVD)

$$A = U\Sigma V^T. \quad (1)$$

Our first problem is that of approximating A by a matrix of rank d :

$$\min_{\text{rank}(B)=d} \|A - B\|_F. \quad (2)$$

It is well-known, see e.g. [?, Chapter 2], that the solution of this problem is

$$B = U_d \Sigma_d V_d^T, \quad (3)$$

where U_d and V_d are the first d columns of orthogonal matrices U and V , respectively, and Σ_d is the upper $d \times d$ block of the diagonal matrix Σ . We will use (??) as a model problem for optimization on a product manifold.

It frequently happens in practice that some of the measurement data in the matrix A are missing. The problem of finding a rank- d approximation of such a partially specified matrix is more difficult, and no closed solution exists. For this problem we measure the closeness of two matrices with missing data with a modified Frobenius norm just as in [?]. Thus we want to solve

$$\min_{\text{rank}(B)=d} \|A - B\|_{\text{mod}}, \quad (4)$$

where the norm is the Frobenius norm and the modification consists of deleting the missing elements from the sum. For this approach to be sound we have to assume that relatively few elements are unspecified.

A first ansatz that takes care of the rank- d assumption is to represent the unknown matrix as

$$B = XY^T,$$

where both X and Y have d columns, so that the problem becomes

$$\min_{X,Y} \|A - XY^T\|_{\text{mod}}. \quad (5)$$

The problem (??) is overparameterized: this is easily seen by putting $\tilde{X} = XE$, $\tilde{Y} = YE^{-T}$, which will leave $B = XY^T = \tilde{X}\tilde{Y}^T$ unchanged for any nonsingular $d \times d$ matrix E . Therefore the problem (??) will not have an isolated local minimum, which will severely degrade the performance of standard optimization algorithms. For instance, Newton's method will converge at a linear rate instead of quadratic. A possible remedy is to impose constraints on X and/or Y to get a unique solution. The most natural ones are to require X or Y to be orthogonal. The problem (??) with orthogonality constraints will be our second example of a product manifold problem.

In [?] it was shown how to enforce orthogonality of the iterates in the standard optimization procedures Conjugate Gradient (CG) and Newton. We will discuss the two problems mentioned above and develop optimization procedures, in particular Newton's method, on a product manifold. Our approach will resemble Section 9.4 of [?], where problems with more than one set of variables are treated. We will give some numerical experiments and compare with algorithms proposed in [?].

To give a flavor of Grassmann optimization we present the symmetric eigenvalue problem for an $n \times n$ matrix A originally treated in [?] and then followed up in e.g. [?],[?],[?]. The eigenspaces are stationary points of

$$F(X) = \text{tr}(X^T A X) \quad (6)$$

where following [?] X is orthogonal $n \times d$ ($X^T X = I$). (??) is a function on the Grassmann manifold because it is independent of the choice of basis spanned by the columns of X . For any orthogonal $d \times d$ matrix Q we have $F(XQ) = F(X)$.

The function corresponding to (??) for the case of the SVD problem is

$$F(X, Y) = \text{tr}(X^T A Y) \quad (7)$$

where the stationary points are subsets of the U and V matrices. $F(X, Y)$ is no longer a function on a product of Grassmann manifolds because $F(XQ_1, YQ_2) \neq F(X, Y)$ in general. In [?] we tried to (implicitly) "preprocess" $F(X, Y)$ to turn it into a proper function on Grassmann manifolds. The argument was that $\max_{Q_1, Q_2} F(X, Y)$ will be a "Grassmann" function so that the minimization algorithms in [?] could be used. Similarly for the function (??) we used a dimension counting argument that the Newton method could be applied to

it as if it was a proper Grassmann function. In this paper we will use a less painful and more streamlined route to derive the equations in the Newton method. It consist of finding a vector field related to the function whose zeroes are stationary points of the function and then apply the Newton method from [?] to this problem.

Of particular interest is the problem when not all elements of A are available. For a matrix A with missing entries, there seem to be no standard way to approximate it with a matrix of rank d . There are model based criteria and algorithms that try to predict the missing elements of A with e.g. clustering. We will treat the approach of principal component analysis, that is to minimize (??). Let the positions where data are absent be a set $\mathcal{M} = \{(i, j) | a_{ij} \text{ is unknown} \}$. We are going to minimize a pseudo-norm

$$\|A - B\|_{\text{mod}}^2 = \sum_{(i,j) \notin \mathcal{M}} (a_{ij} - b_{ij})^2,$$

where we will set $B = XY^T$. This is a natural alteration of the Frobenius norm that has been proposed in e.g. [?]. $\|A - B\|_{\text{mod}}^2$ can be written in different ways. With a $mn \times mn$ diagonal weight matrix w where $W_{kk} = 0$ if $k = i(m - 1) + j$ for all $(i, j) \in \mathcal{M}$ and 1 otherwise, we have

$$\|A - B\|_{\text{mod}}^2 = (\text{vec}(A - B))^T W \text{vec}(A - B).$$

This problem is treated with Grassmann optimization methods in [?] but a critical assumption in the derivations therein is that W is symmetric positive definite and can thus not be applied directly to our semi-definite W here.

An outline of the paper follows. In section 2 we give a brief introduction to the Grassmann manifold and the concept of a product manifold. Section 3 applies the Newton-Grassmann algorithm to find the zero of a vector field on a product of manifolds. We begin by a rederivation of the correction equation in the JD-SVD algorithm from [?] and give the obvious generalization to a block version of it. We proceed with the least squares problem (??) as a starting point to solve the missing data problem (??) with Newton-Grassmann. Section 4 contains numerical demonstrations.

2 The Grassmann Manifold

We give a brief and somewhat informal introduction to concepts from differential geometry that will be needed. For a thorough account of differential geometry we recommend [?] or a textbook as [?]. A point on the Grassmann manifold $\text{Gr}(d, n)$ is a subspace of dimension d in $\mathbf{R}^n$. We follow [?] and let a point be represented by an orthogonal ($X^T X = I$) $n \times d$ matrix X . The choice of basis for a subspace is irrelevant so X and XQ , with $Q \in \text{O}(d)$ are the same point in $\text{Gr}(d, n)$. We will refer to X simply as a matrix instead of the strictly correct but cumbersome “subspace spanned by the columns of X ”.

For a general k -dimensional manifold M there are coordinate functions $\phi_1(X), \phi_2(X), \dots, \phi_k(X)$ that is a mapping of a neighbourhood of X to $\mathbf{R}^k$. The inverses of the ϕ_i constitute a parametrization (locally) of M . In principle an optimization problem on a manifold can be

formulated as an unconstrained optimization problem on $\mathbf{R}^k$ in terms of the parameters. Of course there may exist bounds on the parameters because they are only local in general but this turns out not to be the case for $\text{Gr}(d, n)$ to which we now turn. Given a basis for the orthogonal complement $X_0^\perp$ for an arbitrary “origin” X_0 the Grassmann manifold $\text{Gr}(d, n)$ can be (locally) parameterized by an $(n - d) \times d$ matrix B as

$$X = X_0 + X_0^\perp B. \quad (8)$$

These coordinates are used by e.g. Manton [?] and are appropriate for differentiating a function on $\text{Gr}(d, n)$. A drawback for large problems with (??) is the need to compute a basis for $X_0^\perp$, instead we are going to use a matrix Δ in $X_0^\perp$. Further, several of our problems do not directly involve a real-valued function on $\text{Gr}(d, n)$ but rather amount to finding the zeroes of a vector field. A vector field is a function that assigns a tangent vector to each point on a manifold.

The tangent space at X consist of all possible “directions” one can move in and infinitesimally remain on M . It consist of the derivatives $c'(0)$ of curves $c(t) = (\phi_1(t), \dots, \phi_k(t))$ through X , i.e. $c(0) = X$. For $\text{Gr}(d, n)$ [?] show that $\Delta \in \mathbf{R}^{n \times d}$ is a tangent vector at X if $X^T \Delta = 0$. We will occasionally refer to tangent vectors as simply tangents as the “vectors” are represented as matrices.

For a Riemannian manifold there is a metric (inner product) between tangent vectors $\langle \Delta_1, \Delta_2 \rangle$. We use the standard scalar product

$$\langle \Delta_1, \Delta_2 \rangle = \text{trace}(\Delta_1^T \Delta_2). \quad (9)$$

A function $F(X)$ on $\text{Gr}(d, n)$ is independent of the choice of basis so $F(XQ) = F(X)$. The gradient ∇F at X is defined as the tangent vector that satisfy $\langle \nabla F, \Delta \rangle = \frac{d}{dt} F(c(t))$ where $c(0) = X$ and $c'(0) = \Delta$. It is

$$\nabla F = (I - XX^T) \text{grad} F := \Pi_X \text{grad} F \quad (10)$$

where $\text{grad} F$ is the usual Euclidean gradient.

A vector field assigns a tangent vector $Z(X)$ to a point X . We are going to use Newton’s method to find X such that $Z(X) = 0$. To that end we need the covariant derivative (a.k.a. the connection) that is a directional derivative of a vector field that in Euclidean space is

$$\nabla_\Delta Y = \frac{d}{dt} Y(X + t\Delta).$$

evaluated at $t = 0$. On a manifold the line $X + t\Delta$ is replaced by a curve with $c(0) = X$ and $c'(0) = \Delta$. From [?] we have the connection

$$\nabla_\Delta Z = \Pi(\mathbf{D}Z(\Delta)). \quad (11)$$

where $\mathbf{D}Z(\Delta)$ denotes the first order variation of Z in the direction Δ . The Newton direction is then obtained from the equation

$$\nabla_{\Delta} Z = -Z. \quad (12)$$

Many of our functions involve the trace of a product of matrices. We will then obtain the differential from the expansion

$$F(X + \Delta) = F(X) + \langle \mathbf{D}F, \Delta \rangle \quad (13)$$

where from (??) $\langle \mathbf{D}F, \Delta \rangle = \langle \Delta, \mathbf{D}F \rangle = \text{trace}(\Delta_1^T \Delta_2)$. For later reference we state the following (for proof see e.g. [?])

$$\text{tr}(AB) = \text{tr}(BA), \quad (14)$$

$$\text{tr}(ABC) = \text{tr}(CAB) = \text{tr}(BCA), \quad (15)$$

$$\text{tr}(A^T B) = \text{tr}(B^T A) = \text{tr}(AB^T) = \text{tr}(BA^T), \quad (16)$$

where all involved matrix products are required to be square and the cyclic permutation property (??), that is a consequence of (??), generalizes to a product of any number of matrices.

To move in direction Δ we may use the geodesic curves, corresponding to straight lines, from [?]

$$X_{\Delta}(t) = XVC(t)V^T + US(t)V^T \quad (17)$$

where $\Delta = U\Sigma V^T$ is the compact SVD, $C(t) = \cos(t\Sigma)$, and $S(t) = \sin(t\Sigma)$. An alternative, resembling (??), is to use

$$X := \text{qf}(X + \Delta) \quad (18)$$

where qf is the matrix Q in the QR factorization. In [?] we demonstrate that (??) agree with (??) to third order.

The geodesics provide a special coordinate system, often called normal or geodesic coordinates, around X . For a point $\tilde{X}$ the coordinate is the tangent vector Δ that is reached by a geodesic $\tilde{X} = X_{\Delta}(1)$ (though strictly speaking Δ should be given in terms of a basis for $X^{\perp}$ as in (??) to complete the mapping to $\mathbf{R}^{d(n-d)}$). For a function on $\text{Gr}(d, n)$, or any manifold, the Hessian is defined as the connection applied to ∇F and in geodesic coordinates the connection simplifies to the identity making $\text{Gr}(d, n)$ “locally flat”. In [?] we often had a function on $\text{Gr}(d, n)$ and used a Taylor expansion of $F(X_{\Delta}(t))$ to identify the Hessian.

It will happen frequently that a function $F(X)$ is not a proper function on $\text{Gr}(d, n)$ i.e. $F(XQ) \neq F(X)$ may happen. Nevertheless Newton’s method can be used to find a stationary point $\text{grad}F(X) = 0$ by solving $\Pi \text{grad}F(X) = 0$ that is a vector field on $\text{Gr}(d, n)$ using the procedure above (??). This may require a motivation that we really have solved the optimization problem because we can only guarantee that $\text{grad}F(X) = XK$ for some $d \times d$ matrix K .

We will make use of the two product manifolds $\text{Gr}(d, m) \times \text{Gr}(d, n)$ and $\text{Gr}(d, m) \times \mathbf{R}^{n \times d}$. The tangent directions will in the first case be a pair

$$\Delta = \begin{pmatrix} \Delta_X \\ \Delta_Y \end{pmatrix} \quad (19)$$

where $X^T \Delta_X = Y^T \Delta_Y = 0$. The scalar product generalizes to

$$\langle \Delta^{(1)}, \Delta^{(2)} \rangle = \text{tr}((\Delta_X^{(1)})^T (\Delta_X^{(2)}) + (\Delta_Y^{(1)})^T (\Delta_Y^{(2)})). \quad (20)$$

To get the gradient of a function $F(X, Y)$ (??) changes to

$$\nabla F = \begin{pmatrix} (I - XX^T) \text{grad}_X F \\ (I - YY^T) \text{grad}_Y F \end{pmatrix} \quad (21)$$

where $\text{grad}_X F$ and $\text{grad}_Y F$ denote the “gradient” with respect to the X and Y variables. The projection in the connection (??) can be written as a matrix

$$\Pi = \begin{pmatrix} \Pi_X & 0 \\ 0 & \Pi_Y \end{pmatrix} \quad (22)$$

where $\Pi_X = I - XX^T$ and $\Pi_Y = I - YY^T$. Finally the geodesic is

$$\begin{pmatrix} X(t) \\ Y(t) \end{pmatrix} = \begin{pmatrix} XV_x \cos(\Sigma_x t) V_x^T + U_x \sin(\Sigma_x t) V_x^T \\ YV_y \cos(\Sigma_y t) V_y^T + U_y \sin(\Sigma_y t) V_y^T \end{pmatrix}, \quad (23)$$

where $\Delta_X = U_x \Sigma_x V_x^T$ and $\Delta_Y = U_y \Sigma_y V_y^T$ are “skinny” SVDs.

For the simpler second case Δ_Y is unconstrained and no projection on $Y^\perp$ is to be done, $\Pi_Y = I$ in (??) and (??). The geodesic also becomes

$$\begin{pmatrix} X(t) \\ Y(t) \end{pmatrix} = \begin{pmatrix} XV_x \cos(\Sigma_x t) V_x^T + U_x \sin(\Sigma_x t) V_x^T \\ Y + t \Delta_Y \end{pmatrix} \quad (24)$$

3 Newton Methods with a Pair of Manifolds

In this section we will first treat the model problem (??) without missing elements as a preliminary to the problem (??) in the form of (??). A recurring theme will be the realization of (??) for an appropriate vector field Z .

3.1 Trace maximization and JD-SVD

To start we will give a block generalization of the Jacobi-Davidson variant for the partial SVD from [?]. It is almost parallel to the development in [?] as the correction equation is derived from a Newton step to get a zero residual.

We are going to use the manifold $\text{Gr}(d, m) \times \text{Gr}(d, n)$ where now two points are identified if $X_1 = X_2 Q_1$ and $Y_1 = Y_2 Q_2$. Now with those changes of coordinates we no longer have invariance of

$$F(X, Y) = \text{tr}(X^T AY) \quad (25)$$

under basis change $X := XQ_1, Y := YQ_2$ in general. However the global optimum of (??) is attained when X and Y consist of the first (or last) d columns of the matrices in the U and V of the SVD. This is the famous Eckart-Young theorem [?]. As argued before we will get a correct Newton method by setting the gradient projected to $\text{Gr}(d, m) \times \text{Gr}(d, n) = 0$.

We now proceed to insert $(\tilde{X}, \tilde{Y}) = (X + \Delta_X, Y + \Delta_Y)$ into (??) and get from the linearity of the trace

$$F(\tilde{X}, \tilde{Y}) = \text{tr}(X^T AY) + \text{tr}(\Delta_X^T AY + X^T A \Delta_Y). \quad (26)$$

where we immediately see that $\text{grad}_X F = AY$. Transposing the term with Δ_Y in (??) that is allowed from (??) we obtain $\text{grad}_Y F = A^T X$. The vector field Z in the Newton equation (??) is then

$$Z = \begin{pmatrix} \Pi_X AY \\ \Pi_Y A^T X \end{pmatrix}.$$

For X and Y making $Z = 0$ we see that we get a pair of subspaces satisfying the invariance equations $AY = XH$ and $A^T X = YH^T$ that hold for matching columns of U and V in the SVD.

To obtain the connection $\nabla_\Delta Z$ we then gather the first order terms in Δ_X and Δ_Y in

$$\begin{pmatrix} \Pi_X(I - \tilde{X}\tilde{X}^T)A\tilde{Y} \\ \Pi_Y(I - \tilde{Y}\tilde{Y}^T)A^T\tilde{X} \end{pmatrix} \quad (27)$$

As $\Pi_X X = 0$ and $\Pi_X \Delta_X = \Delta_X$ we have that $\Pi_X(I - \tilde{X}\tilde{X}^T) = \Pi_X - \Delta_X X^T$ to first order because the $\Delta_X \Delta_X^T$ in $\tilde{X}\tilde{X}^T$ is dropped. With this we quickly identify the first half of the connection as

$$\Pi_X A \Delta_Y - \Delta_X X^T AY$$

as the second order term $\Delta_X X^T A \Delta_Y$ is discarded. A completely analogous calculation yields the second component in (??) so that the Newton equation becomes

$$\begin{cases} -\Delta_X X^T AY + \Pi_X A \Delta_Y &= -\Pi_X AY \\ -\Delta_Y Y^T A^T X + \Pi_Y A^T \Delta_X &= -\Pi_Y A^T X \end{cases} \quad (28)$$

This is a system of Sylvester equations corresponding to the one for the Newton-Grassmann algorithm applied to the symmetric eigenvalue problem [?, ?]. It is also the “obvious” block generalization of the correction equation in the JDSVD [?].

Algorithm 1: Product Newton-Grassmann for trace maximization

1. Compute $H = X^T AY$ and the pair of residuals
 $R_1 = \Pi_X AY = AY - XH, R_2 = \Pi_Y A^T X = A^T X - YH^T$.

2. Solve the system of equations with algorithm 2

$$\begin{cases} -\Delta_X H + \Pi_X A \Delta_Y &= -R_1, \\ \Pi_Y A^T \Delta_X - \Delta_Y H^T &= -R_2, \end{cases} \quad (29)$$

for the correction direction $\Delta = (\Delta_x, \Delta_y)$.

3. Update (X, Y) in the direction of Δ .

4. Deflate converged columns of X and Y and repeat.

Following [?] we should make the change of variables $X := XQ_1$, $Y := YQ_2$ so that H in (??) becomes diagonal. This is accomplished via the SVD of H and will produce d independent linear equations for each pair of columns of Δ_X and Δ_Y . As $\Pi_X \Delta_X = \Delta_X$ and $\Pi_Y \Delta_Y = \Delta_Y$ we get the decoupled equations for column i , $i = 1, 2, \dots, d$:

$$\Pi \begin{pmatrix} -\tau_i I & A \\ A^T & -\tau_i I \end{pmatrix} \Pi \begin{pmatrix} \Delta_X(:, i) \\ \Delta_Y(:, i) \end{pmatrix} = - \begin{pmatrix} AY(:, i) - \tau_i X(:, i) \\ A^T X(:, i) - \tau_i Y(:, i) \end{pmatrix}, \quad (30)$$

where

$$\Pi = \begin{pmatrix} \Pi_X & 0 \\ 0 & \Pi_Y \end{pmatrix}.$$

For each i this is the correction equation in the JDSVD algorithm 4.1 in [?] with the difference that there X and Y are one-dimensional so that $\Pi_x = I - xx^T$ and $\Pi_y = I - yy^T$. With a slight reformulation of [?, Lemma 7.1] that gives an augmented system equivalent to (??) we get the following

Algorithm 2: Solution of the correction equation (??)

1. Compute the SVD of $H = X^T A Y = Q_1 \text{diag}(\tau_1, \dots, \tau_d) Q_2^T$.
2. Update $X := XQ_1$ and $Y := YQ_2$.
3. Solve the system for $i = 1, 2, \dots, d$

$$\begin{pmatrix} -\tau_i I & A & X & 0 \\ A^T & -\tau_i I & 0 & Y \\ X^T & 0 & 0 & 0 \\ 0 & Y^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta_X(:, i) \\ \Delta_Y(:, i) \\ l_1 \\ l_2 \end{pmatrix} = \begin{pmatrix} \tau_i X(:, i) - AY(:, i) \\ \tau_i Y(:, i) - A^T X(:, i) \\ 0 \\ 0 \end{pmatrix} \quad (31)$$

where l_1 and l_2 are Lagrange multiplier vectors.

We will only solve the correction equation exactly to demonstrate the local cubic convergence on moderately sized problems but we point out that for a large sparse matrix A , the system (??) is solved iteratively. If we have a preconditioner

$$P \approx \begin{pmatrix} -\tau_i I & A \\ A^T & -\tau_i I \end{pmatrix}$$

then in each iteration of a preconditioned solver like GMRES (??) is solved with the approximation above just as for the eigenvalue problem [?],[?],[?].

3.2 A Least Squares Approach

We now proceed to solve (??) with a parametrization $B = XY^T$ where $X \in \text{Gr}(d, m)$ and $Y \in \mathbf{R}^{n \times d}$ have full rank d . Spelled out

$$\min \frac{1}{2} \|A - XY^T\|_F^2 = \min F(X, Y). \quad (32)$$

In [?] we used a dimension counting argument that the proper product manifold to place the problem on is $\text{Gr}(d, m) \times \mathbf{R}^{n \times d}$. As in the previous section we will project the Euclidean gradient on the manifold. We insert $(\tilde{X}, \tilde{Y}) = (X + \Delta_X, Y + \Delta_Y)$ into $F(\tilde{X}, \tilde{Y}) = \frac{1}{2} \text{tr}((A^T - \tilde{Y}\tilde{X}^T)(A - \tilde{X}\tilde{Y}^T))$. Collecting the first order terms we obtain

$$\frac{1}{2} \text{tr}(-Y\Delta_X^T A - A^T \Delta_X Y^T + \Delta_Y Y^T + Y\Delta_Y^T - \Delta_Y X^T A - A^T X \Delta_Y^T)$$

and after rearranging some terms with (??), (??) end up with

$$\text{tr}(-\Delta_X^T A Y + \Delta_Y^T (Y - A^T X))$$

from which we identify $\text{grad}_X F = -AY$ and $\text{grad}_Y F = Y - A^T X$. Our vector field Z in the Newton equation is then

$$Z = \begin{pmatrix} -\Pi_X A Y \\ Y - A^T X \end{pmatrix}.$$

Apart from the minus sign we see that the first equation is identical to (??) and thus reproduces the first Newton equation. The second equation without projection becomes with $(\tilde{X}, \tilde{Y})$ inserted to get the “Y-component” of the connection $-A^T \Delta_X + \Delta_Y$. Spelled out

$$\Delta_X H - \Pi_X A \Delta_Y = AY - XH, \quad (33)$$

$$-A^T \Delta_X + \Delta_Y = A^T X - Y. \quad (34)$$

where $H = X^T A Y$ as before. With $\Delta_Y = A^T(X + \Delta_X) - Y$ from (??) inserted into (??) we get the same type of correction equation in Δ_X as for Newton-Grassmann for the eigenvalue problem with the matrix AA^T . This is seen from the second term of (??)

$$-\Pi_X A \Delta_Y = -\Pi_X A A^T (X + \Delta_X) + \Pi_X A Y,$$

where $\Pi_X A Y$ equals the right hand side and is cancelled.. It is solved in a similar way as in [?, chap 7], [?] to get the following

Algorithm 3: Newton step for minimizing $\|A - XY^T\|$

1. Compute the SVD of $H = X^T A Y = Q_1 \text{diag}(\tau_1, \dots, \tau_d) Q_2^T$.
2. Update $X := X Q_1$ and $Y := Y Q_2$.

3. Solve the correction equation

$$\Delta_X H - \Pi_X A A^T \Delta_X = \Pi_X A A^T X =: R. \quad (35)$$

In the new variables this becomes

$$\begin{pmatrix} \tau_i I - A A^T & X \\ X^T & 0 \end{pmatrix} \begin{pmatrix} \Delta_X(:, i) \\ l \end{pmatrix} = \begin{pmatrix} R(:, i) \\ 0 \end{pmatrix} \quad i = 1, \dots, d.$$

4. Update X as in (??) with $t = 1$ and $Y := Y + \Delta_Y = A^T(X + \Delta_X)$.

The best choice here would be to take instead $Y = A^T X$ that would turn (??) into the correction equation for the eigenvalue problem for $A A^T$ that gives cubic convergence. With a different Y we get quadratic convergence only. Thus algorithm 3 is not the best choice for the case when all elements in A are known. The situation changes in case of missing elements of A however. In that case we do not have access to any multiplication by the matrix A and cannot generalize the objective function (??). Then we have to use a modification of (??).

3.3 Least Squares with Missing Data

We now turn to the problem of minimizing

$$\|A - B\|_{\text{mod}}^2 = \sum_{(i,j) \notin \mathcal{M}} (a_{ij} - b_{ij})^2,$$

where $\mathcal{M}$ is the set of indices for the missing elements. As for the full matrix case we set $B = X Y^T$ where $X \in \text{Gr}(d, m)$ and $Y \in \mathbf{R}^{n \times d}$. The development in the previous section for the function (??) will be reused.

Our criterion is only reasonable if relatively few elements are missing, that is $\#\mathcal{M} \ll mn$. We suggest to see the pseudonorm as a small modification of the Frobenius norm. If $\hat{A}$ is defined by inserting zeroes on the positions of the unknown data

$$\hat{a}_{ij} = \begin{cases} a_{ij} & \text{if } (i, j) \notin \mathcal{M}, \\ 0 & \text{if } (i, j) \in \mathcal{M}, \end{cases} \quad (36)$$

we have yet another way of writing the pseudonorm

$$\|A - B\|_{\text{mod}} = \|\hat{A} - B\|_F^2 - \sum_{(i,j) \in \mathcal{M}} b_{ij}^2. \quad (37)$$

To extract element b_{ij} we multiply $e_i^T B e_j$ where e_i and e_j are standard basis vectors in $\mathbf{R}^m$ and $\mathbf{R}^n$ respectively. We want a trace expression to facilitate the gradient computation. Let E_i^m denote an $m \times m$ matrix with all zeroes except position (i, i) that is 1 and E_j^n an $n \times n$ matrix with $E_{jj} = 1$. We then have $b_{ij}^2 = \|E_i^m B E_j^n\|_F^2$.

We can now formulate our objective function in terms of Frobenius norms used in (??)

$$F(X, Y) = \frac{1}{2} \|A - XY^T\|_{\text{mod}}^2 = \frac{1}{2} \|\hat{A} - XY^T\|_F^2 - \frac{1}{2} \sum_{(i,j) \in \mathcal{M}} \|E_i^m XY^T E_j^n\|_F^2, \quad (38)$$

with $\hat{A}$ from (??). The gradient for the first term of (??) was obtained above so we now turn to a term in the second sum

$$G^{ij}(X, Y) = \|E_i^m XY^T E_j^n\|_F^2 = \text{tr}(E_j^n Y X^T E_i^m E_i^m XY^T E_j^n), \quad (39)$$

We move the first E_j^n last and use $(E_i^m)^2 = E_i^m$ and $(E_j^n)^2 = E_j^n$ in (??) before inserting $(\tilde{X}, \tilde{Y}) = (X + \Delta_X, Y + \Delta_Y)$. The first order term in the expansion is then readily obtained by replacing each X and Y with Δ_X and Δ_Y . The result is

$$\frac{1}{2} \text{tr}(\Delta_Y X^T E_i^m XY^T E_j^n + Y \Delta_X^T E_i^m XY^T E_j^n + X^T E_i^m \Delta_X Y^T E_j^n + Y X^T E_i^m X \Delta_Y^T E_j^n),$$

that is rewritten in the form $\text{tr}(\Delta_X^T \text{grad} G_X^{ij}) + \text{tr}(\Delta_Y^T \text{grad} G_Y^{ij})$ with (??)- (??) and the symmetry of the E matrices. The result is

$$\text{tr}(\Delta_X^T E_i^m XY^T E_j^n Y) + \text{tr}(\Delta_Y^T E_j^n Y X^T E_i^m X).$$

Our vector field Z on $\text{Gr}(d, m) \times \mathbf{R}^{n \times d}$ then becomes

$$Z = \begin{pmatrix} \Pi_X \text{grad} F_X \\ \text{grad} F_Y \end{pmatrix} = \begin{pmatrix} \Pi_X \hat{A} Y - \sum_{(i,j) \in \mathcal{M}} \Pi_X E_i^m XY^T E_j^n Y \\ Y - \hat{A}^T X - \sum_{(i,j) \in \mathcal{M}} E_j^n Y X^T E_i^m X \end{pmatrix} \quad (40)$$

where for later reference we set, somewhat sloppily, $\nabla G_X = \sum_{(i,j) \in \mathcal{M}} \Pi_X E_i^m XY^T E_j^n Y$ and $\nabla G_Y = \sum_{(i,j) \in \mathcal{M}} E_j^n Y X^T E_i^m X$. Those terms in the last sum is the only difference from the connection above. Again using $\Pi_X(I - \tilde{X} \tilde{X}^T) = \Pi_X - \Delta_X X^T$ we see that the first order terms in $\Pi_X(I - \tilde{X} \tilde{X}^T) \sum_{(i,j) \in \mathcal{M}} E_i^m \tilde{X} \tilde{Y}^T E_j^n \tilde{Y}$ is a sum of

$$G_{XX}(\Delta_X) := \sum_{(i,j) \in \mathcal{M}} \Pi_X E_i^m \Delta_X Y^T E_j^n Y - \Delta_X X^T E_i^m XY^T E_j^n Y, \quad (41)$$

$$G_{XY}(\Delta_Y) := \sum_{(i,j) \in \mathcal{M}} \Pi_X (E_i^m X \Delta_Y^T E_j^n Y + E_i^m XY^T E_j^n \Delta_Y). \quad (42)$$

For the second component of the connection we see readily from the first order terms in $\sum_{(i,j) \in \mathcal{M}} E_j^n \tilde{Y} \tilde{X}^T E_i^m \tilde{X}$ that it consist of

$$G_{YX}(\Delta_X) := \sum_{(i,j) \in \mathcal{M}} E_j^n Y \Delta_X^T E_i^m X + E_j^n Y X^T E_i^m \Delta_X, \quad (43)$$

$$G_{YY}(\Delta_Y) := \sum_{(i,j) \in \mathcal{M}} E_j^n \Delta_Y X^T E_i^m X. \quad (44)$$

We can now collect the Newton equations into

$$\Delta_X \hat{H} - G_{XX}(\Delta_X) - \Pi \hat{A} \Delta_Y - G_{XY}(\Delta_Y) = \hat{A}Y - X\hat{H} - \nabla G_X, \quad (45)$$

$$-\hat{A}^T \Delta_X - G_{YX}(\Delta_X) + \Delta_Y - G_{YY}(\Delta_Y) = \hat{A}^T X - Y - \nabla G_Y, \quad (46)$$

with gradients from (??), $G_{XX} - G_{YY}$ from (??)-(??) and where $\hat{H} = X^T \hat{A}Y$.

The Newton-equations (??) and (??) are a linear system of equations and can be solved by rewriting it as a matrix-vector system with Δ_X and Δ_Y in vec-form. The system of equations can no longer be decoupled into d equations for each pair of columns by making $\hat{H}$ diagonal. We give the details leading to (??) in an appendix.

4 Numerical Experiments

We are now going to demonstrate the algorithms in previous sections on an idealized matrix that is constructed like in [?]

$$A(\rho) = A_{r4} + \rho E \quad (47)$$

where A_{r4} is a rank 4 50×40 matrix with nonzero singular values 100,90,80,70 and E is a random matrix with elements from $N(0,1)$. The parameter ρ in (??) is used to adjust the gap between σ_4 and σ_5 of $A(\rho)$ and consequently the size of the residual $\|A - XY^T\|$. We are going to use $\rho = 0.1$ for a “nice” problem and $\rho = 3$ for a slightly more challenging one.

Our first example illustrates the cubic convergence of algorithm 1. In Figure ?? we applied it to $A(3)$ where the starting approximations X_0, Y_0 are a random perturbation of the 4 first singular vectors to give a subspace angle approximately equal to 0.5. The tripling of the accuracy is seen after the second iteration before full machine accuracy is attained.

Figure 1: $\|\nabla F_X\|_F$ (o) and $\|\nabla F_Y\|_F$ (+) for algorithm 1 on $A(3)$

Next we show the quadratic convergence of algorithm 3. We used the same X_0 and then $Y_0 = A^T X_0$ to get a fairly well scaled Y_0 . Our experience is that $\|Y_0\| \approx \|A\|$ is critical to

find the correct subspaces of singular vectors. If $\|Y_0\| \ll \|A\|$ it is very likely that we will converge to singular vectors corresponding to vectors corresponding to smaller singular values even if the angle between Y_0 and the exact Y is small. Figure ?? shows the local quadratic convergence expected from a Newton method. We also compare the alternative to update $X := \text{qf}(X + \Delta_X)$, where qf means the Q-factor in a compact QR decomposition, instead of the exact Grassmann geodesic update that do not affect the local convergence. However for a relatively crude starting guess $\|\Delta_X\|$ is typically large enough at the first Newton steps to make a significant difference between the two X : s and the latter update seems to have more problems to enter the quadratic convergence mode. This is more pronounced in the case of missing elements to which we now turn.

Figure 2: $\|\nabla F_X\|_F$ (o) and $\|\nabla F_Y\|_F$ (+) for algorithm 3 on $A(3)$ with different updating of X .

We first use the Newton algorithm on $A(0.1)$ and $A(3)$ with 20 (= 1%) of the elements removed. As starting guess we used for both problems $X_0 = U(:, 1 : 4)$ and $Y_0 = V(:, 1 : 4)\Sigma(1 : 4, 1 : 4)$ where $\hat{A} = U\Sigma V^T$ is the SVD of the matrix with zeroes inserted in missing positions. In Figure ?? we see quadratic convergence and very similar behaviour because the starting values are good in both cases. In the later figures we will display the norm $\|\nabla F\|_F = \sqrt{\|\nabla F_X\|_F^2 + \|\nabla F_Y\|_F^2}$

We now try to use an alternative starting approximation following [?] where a missing element in position (i, j) is estimated with the mean value of the known elements in row i or column j depending on which sample variance is the lowest. This is a primitive form of “imputation”, i.e. estimation of missing elements. The starting values are then obtained from the SVD of this matrix. Also we compare exact geodesic update of X with $X := \text{qf}(X + \Delta_X)$. In Figure ?? we see that the performance from the starting values are indistinguishable for both matrices with the difference being the method to update X as in Fig. ??.

For a more challenging problem we now remove 400 (= 20%) of the elements in $A(3)$ and $A(0.1)$ and repeat the experiment. We note in Figure ?? that geodesic update again

Figure 3: $\|\nabla F_X\|_F$ (left) $\|\nabla F_Y\|_F$ (right) for 20 missing elements in $A(0.1)$ (o) and $A(3)$ (+).

give the same behaviour for both $A(0.1)$ and $A(3)$. On the other hand with the update $X := \text{qf}(X + \Delta_X)$ we see some of the “difficulty” of $A(3)$ where the starting values from the SVD of $\hat{A}$ is on the verge of misconvergence and the one from the imputed values fails outright.

In our case we “cheated” and used the known rank 4 on X and Y . If we use rank 3-approximation we get nearly exactly the same figures as ?? and ?? and the same behaviour of all variants. On the other hand if rank 5 is used we do not get convergence except for a couple of cases with few missing elements where it was very irregular (more than 10 iterations to reach quadratic convergence mode).

Based on our limited testing we have seen similar behaviour in terms of success to converge of our Newton method as the algorithms in [?]. They advocate an alternating least squares approach which converges much more slowly, linearly instead of quadratically, but with cheaper iterations. We agree with the notion in [?] that in case of convergence problem we are probably modelling noise that give a large cluster of small singular values. We have also demonstrated that our algorithm can handle at least 20 % missing values. Also in [?] it is shown by an example that we have to recompute X and Y from scratch if the rank is increased, the result of a method that try to compute the columns of X and Y sequentially differ, There is no such result as (??) for the case when we have missing data.

To sum up the implementation issues we have observed that the geodesic update of X has better convergence properties and also that the starting values obtained from the partial SVD of $\hat{A}$ is a reliable choice that should be used unless any model for the missing values are known.

Figure 4: $\|\nabla F\|_F$ with 20 elements removed. Starting values from SVD of $\hat{A}$ for $A(0.1)$ (o), $A(3)$ (+) and from SVD of matrix with imputed mean values for $A(0.1)$ (square), $A(3)$ (*).

4.1 Comparison to Alternating Least Squares

In [?] several methods for (??) was tested. The slowest method was to impute the missing values with the partial SVD (??) and then recompute the SVD. Instead the authors favoured an alternating least squares procedure which we state in our notation as follows. We write the objective function (??) in the form

$$F(X, Y) = \|(\hat{A} - XY^T + \sum_{(i,j) \in \mathcal{M}} E_i^m XY^T E_j^n)\|_F \quad (48)$$

and assume that Y is known. [?] note that the problem of finding X decouples over the rows of the matrix in (??). We use the “Matlab notation” $X(I, :)$ for the I :th row of X . $E_i^m(I, :) = e_I^T$ if $i = I$ and 0 otherwise. This mean that the I :th row of the matrix in (??) is

$$\hat{A}(I, :) - X(I, :)Y^T(I - \sum_{j:(I,j) \in \mathcal{M}} E_j^m, \quad I = 1, 2, \dots, m \quad (49)$$

and those are minimized in the least squares sense independently. Analogously if X is known in (??) then we consider column J of the matrices. Now $E_j^n(:, J) = e_J$ if $j = J$ and 0 otherwise so we get

$$\hat{A}(:, J) - (I - \sum_{i:(i,J) \in \mathcal{M}} E_i^m)XY(J, :)^T, \quad J = 1, 2, \dots, n \quad (50)$$

that is minimized to obtain the J :th row of Y .

The overall algorithm now consist of using (??) and (??) interchangeably. We remark that without missing elements this is just a block version of simple power iteration with $A^T A$ and AA^T that converges linearly. We now repeat the experiment with our matrices with the X as the first columns of the SVD as for the Newton method.

Figure 5: Plot corresponding to figure ?? with 400 missing elements.

5 Concluding Remarks

We derived instances of the Newton algorithm to find a zero of a vector field on a pair of manifolds to solve the problem of low rank approximation to a given data matrix. Of particular interest is the case where a modest portion of the elements in the matrix is unavailable. This problem is treated in [?] for the case of rank one approximation. There one of the vectors is forced to be on the Grassmann manifold by normalisation of its length. This yield a class of variable projection type algorithms. We believe that this approach can be generalized to the case for rank $d > 1$ with the same formalism as in this paper. With $X \in \text{Gr}(d, m)$ the function $F(X, Y)$ with Y obtained from solving (??) will be a proper Grassmann function that can be minimized by Newtons algorithm or any optimization algorithm currently formulated on $\text{Gr}(d, m)$

Appendix

We will apply the vec-operator to the modifications (??)-(??) to the Newton equation to obtain a matrix times $\text{vec}(\Delta_X)$ or $\text{vec}(\Delta_Y)$ with the usual rule for the terms of a Sylvester type equation i.e. $\text{vec}(AX) = (I \otimes A)\text{vec}X$ and $\text{vec}(XH) = (H^T \otimes I)\text{vec}X$. Also the more general

$$\text{vec}(AXB) = (B^T \otimes A)\text{vec}X$$

from e.g. [?] will be needed. For the two terms in the sum for (??) we obtain with the sum omitted

$$\text{vec}(G_{XX}(\Delta_X)) = (Y^T E_j^n Y \otimes \Pi_X E_i^m) \text{vec} \Delta_X - (X^T E_i^m X Y^T E_j^n Y \otimes I_m) \text{vec} \Delta_X.$$

With the sum reinserted we then get the matrix representation of $\text{vec}(G_{XX}(\Delta_X))$ to apply to $\text{vec} \Delta_X$

$$G_{XX} = \sum_{(i,j) \in \mathcal{M}} (Y^T E_j^n Y \otimes \Pi_X E_i^m) - (X^T E_i^m X Y^T E_j^n Y \otimes I_m) \quad (51)$$

For the two terms in (??) we get

$$\text{vec}(G_{XX}(\Delta_Y)) = (I_d \otimes \Pi_X E_i^m X Y^T E_j^n) \text{vec} \Delta_Y + (Y^T E_j^n \otimes \Pi_X E_i^m X) \text{vec}(\Delta_Y^T)$$

With $\text{vec}(Y^T) = P(d, n) \text{vec}(Y)$ where $P(d, n)$ (notation from [?]) is a permutation matrix we get the matrix representation of $\text{vec}(G_{XY}(\Delta_Y))$

$$G_{XY} = \sum_{(i,j) \in \mathcal{M}} (I_d \otimes \Pi_X E_i^m X Y^T E_j^n) + (Y^T E_j^n \otimes \Pi_X E_i^m X) P(d, n) \quad (52)$$

In exactly the same way we have matrices representing $G_{YX}(\Delta_X)$ and $G_{YY}(\Delta_Y)$ in (??),(??)

$$G_{YX} = \sum_{(i,j) \in \mathcal{M}} (I_d \otimes E_j^n Y X^T E_i^m) + (X^T E_i^m \otimes E_j^n Y) P(d, m), \quad (53)$$

$$G_{YY} = \sum_{(i,j) \in \mathcal{M}} (X^T E_i^m X \otimes E_j^n). \quad (54)$$

Finally we subtract (??)-(??) from the Newton equations for the full matrix problem in (??) and (??)

$$\begin{pmatrix} (\hat{H}^T \otimes I_m) - G_{XX} & (I_d \otimes \Pi_X \hat{A}) - G_{XY} \\ (I_d \otimes \hat{A}^T) - G_{YX} & I_{nd} - G_{YY} \end{pmatrix} \begin{pmatrix} \text{vec} \Delta_X \\ \text{vec} \Delta_Y \end{pmatrix} = \begin{pmatrix} \text{vec}(\hat{A}Y - X\hat{H} - \nabla G_X) \\ \text{vec}(\hat{A}^T X - Y - \nabla G_Y) \end{pmatrix} \quad (55)$$

References

- [1] Karim M. Abadir and Jan R. Magnus, *Matrix Algebra*, Cambridge University Press, 2005.
- [2] P.-A. Absil, R. Mahony and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2008.
- [3] Z .Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.
- [4] M. P. do Carmo, *Riemannian Geometry*, Birkhäuser, Boston, 1992.
- [5] A. Edelman, T. Arias and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 303-353.
- [6] G. H. Golub, C. F. Van Loan, *Matrix Computations* 3rd ed., The Johns Hopkins University Press, 1996.

- [7] B. Grung, R. Manne, *Missing values in principal component analysis*, Chemometrics and Intelligent Laboratory Systems 42 (1998), pp. 125-139.
- [8] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer Verlag, 2001.
- [9] M. Hochstenbach, *A Jacobi-Davidson Type SVD Method*, SIAM J. on Sci. Comp., 23 (2001), pp. 606-628.
- [10] Roger A. Horn and Charles R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1991.
- [11] E. Lundström and L. Eldén, *Adaptive eigenvalue computations using Newton's method on the Grassmann manifold*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 819-839.
- [12] J. H. Manton, *Optimization Algorithms Exploiting Unitary Constraints*, IEEE Trans. Signal Processing, vol. 50 March 2002, pp. 635-650.
- [13] J.H. Manton, R. Mahony and Y. Hua, *The Geometry of Weighted Low-Rank Approximations*, IEEE Trans. Signal Processing, vol. 51 February 2003, pp. 500-514.
- [14] A. Ruhe *Numerical computation of principal components when several observations are missing*, Tech. report, UMINF-48, Umeå, 1974.
- [15] L. Simonsson, *Subspace Computations via Matrix Decompositions and Geometric Optimization*, PhD Thesis, Linköping University, December 2006.