

Article

A Gradient System for Low Rank Matrix Completion

Carmela Scalone ^{1,*} and Nicola Guglielmi ²

¹ Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica (DISIM), Università dell'Aquila, via Vetoio 1, 67100 L'Aquila, Italy

² Section of Mathematics, Gran Sasso Science Institute, via Crispi 7, 67100 L'Aquila, Italy; nicola.guglielmi@gssi.it

* Correspondence: carmela.scalone@graduate.univaq.it; Tel.: +39-086-243-3136

Received: 7 May 2018; Accepted: 18 July 2018; Published: 24 July 2018



Abstract: In this article we present and discuss a two step methodology to find the closest low rank completion of a sparse large matrix. Given a large sparse matrix M , the method consists of fixing the rank to r and then looking for the closest rank- r matrix X to M , where the distance is measured in the Frobenius norm. A key element in the solution of this matrix nearness problem consists of the use of a constrained gradient system of matrix differential equations. The obtained results, compared to those obtained by different approaches show that the method has a correct behaviour and is competitive with the ones available in the literature.

Keywords: low rank completion; matrix ODEs; gradient system

1. Introduction

A large class of datasets are naturally stored in matrix form. In many important applications, the challenge of filling a matrix from a sampling of its entries can arise; this is known as the matrix completion problem. Clearly, such a problem needs some additional constraints to be well-posed. One of its most interesting variants is to find the lower rank matrices that best fit the given data. This constrained optimization problem is known as low-rank matrix completion.

Let $M \in \mathbb{R}^{m \times n}$ be a matrix that is only known on a subset, Ω , of its entries. In [1], the authors provided conditions on the sampling of observed entries, such that the problem which arises has a high probability of not being undetermined. The classical mathematical formulation for the low rank matrix completion problem is :

$$\begin{aligned} & \min \text{rank}(X) \\ & \text{s.t. } P_{\Omega}(X) = P_{\Omega}(M) \end{aligned}$$

where P_{Ω} is the projection onto Ω defined as a function

$$P_{\Omega} : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R}^{m \times n}$$

such that

$$X_{i,j} \longmapsto \begin{cases} X_{i,j} & \text{if } (i,j) \in \Omega \\ 0 & \text{if } (i,j) \notin \Omega \end{cases}$$

This approach may seem like the most natural to describe the problem, but it is not very useful in practice, since it is well known to be NP-hard [2]. In [3], the authors stated the problem as

$$\begin{aligned} & \min \|X\|_* \\ & \text{s.t. } P_{\Omega}(X) = P_{\Omega}(M) \end{aligned}$$

where $\| \cdot \|_*$ is the nuclear norm of the matrix, which is the sum of its singular values. This is a convex optimization problem and the authors proved that when Ω is sampled uniformly at random and is sufficiently large, the previous relaxation can recover any matrix of rank r with high probability. We will consider the following formulation as in [4,5],

$$\min \frac{1}{2} \|P_\Omega(X) - P_\Omega(M)\|_F^2$$

$$s.t. \text{rank}(X) = r$$

Notice that, the projection $P_\Omega(X)$ can be written as a Hadamard product. If we identify the subset Ω of the fixed entries with the matrix Ω such that

$$\Omega_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \Omega \\ 0 & \text{if } (i,j) \notin \Omega \end{cases}$$

it is clear that $P_\Omega(X) = \Omega \circ X$. By considering the manifold

$$\mathcal{M}_r = \{X \in \mathbb{R}^{n \times m} : \text{rank}(X) = r\}$$

we can write the problem as

$$\min_{X \in \mathcal{M}_r} \frac{1}{2} \|\Omega \circ (X - M)\|_F^2 \tag{1}$$

This approach is based on the assumption of knowing in advance the rank r of the target matrix. A key feature of the problem is that $r \ll \min\{m, n\}$, that translates, from a practical point of view in a small increase of the cost to eventually update r . In [6] is well explained the possibility of estimating the rank (unknown a priori) based on the gap between singular values of the “trimmed” partial target matrix M . Furthermore, the authors highlight that in collaborative filtering applications, r ranged between 10 and 30. In [4], the author employed optimization techniques widely exploiting the structure of smooth Riemannian manifold of \mathcal{M}_r . The same tools are used in [5], where the authors considered the matrix completion in the presence of outliers. In the recent work [7], the authors provide a non convex relaxation approach for matrix completion in presence of non Gaussian noise and or outliers, by employing the correntropy induced losses. In [8], the authors survey on the literature on matrix completions and deal with target matrices, whose entries are affected by a small amount of noise. Recently, the problem became popular thanks to collaborative filtering applications [9,10] and the Netflix problem [11]. It can also be employed in other fields of practical applications such as sensor network localization [12], signal processing [13] and reconstruction of damaged images [14]. A very suggestive use of modeling as low rank matrix completion problem has been done in biomathematics area, as shown in [15] for gene-disease associations. Applications to minimal representation of discrete systems can be considered as of more mathematical feature [16]. What makes the problem interesting are not just the multiple applications, but also its variants, such as, for example, structured [17] and Euclidean distance matrix cases [18]. In this paper a numerical technique to solve the low rank matrix completion problem is provided, which makes use of a gradient system of matrix ODEs.

2. General Idea : Two-Level Method

Let us write the unknown matrix X of the problem (1) as $X = \varepsilon E$ with $\varepsilon > 0$ and $\|E\|_F = 1$. For a fixed norm $\varepsilon > 0$, we aim to minimize the functional

$$F_\varepsilon(E) := \|\Omega \circ (\varepsilon E - M)\|_F^2 \tag{2}$$

constrained by $E \in \mathcal{M}_r$ and $\|E\|_F = 1$ (see [19]). By computing the stationary point of a suitable differential equation, we will find a local minimum E_ε of the functional. Setting $f(\varepsilon) = F_\varepsilon(E_\varepsilon)$, we will look for the minimum value of ε , say ε_* , such that $f(\varepsilon_*) = 0$, by using a Newton-like method. The behaviour of $f(\varepsilon)$ in a left neighbourhood of ε_* is well understood. For $\varepsilon \geq \varepsilon_*$ it is more challenging.

We discuss two possible scenarios; ε^* can be a strict local minimum point or $f(\varepsilon)$ can become identically zero when ε exceeds ε^* . The two situations depend on the rank constraint and on the sparsity pattern. To motivate our assumption, we now present two simple illustrative examples. Suppose that we aim to recover the matrix

$$M = \begin{pmatrix} 1 & 1 & * \\ 1 & * & 1 \\ * & 1 & * \end{pmatrix}.$$

If we constrain the problem by imposing that the rank of the solution has to be equal to 1, we have a strict point of minimum for $\varepsilon^* = 3$ and the optimal rank-1 matrix that fits perfectly the given entries of M is

$$S = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

If we consider the problem of recovering the matrix

$$Y = \begin{pmatrix} 1 & 1 & 1 \\ 1 & * & 1 \\ * & * & 1 \end{pmatrix}$$

requiring that the solution has to be of rank 2, we have that the solutions of minimal norm $\varepsilon^* = 2.6458$ are

$$X_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad X_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

However, for all $\varepsilon > \varepsilon^*$, we have a point of minimum of $f(\varepsilon)$. To understand this behaviour, we can intuitively think that there are a lot of “possibilities” to realize a rank-2 matrix “filling” the unknown entries of Y . For example,

$$X_1(\alpha, \beta) = \left\{ \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ \alpha & \beta & 1 \end{pmatrix} \right\}_{\alpha, \beta \in \mathbb{R}, \alpha \neq 1 \vee \beta \neq 1}, \quad X_2(\alpha, \beta) = \left\{ \begin{pmatrix} 1 & 1 & 1 \\ 1 & \alpha & 1 \\ 1 & \beta & 1 \end{pmatrix} \right\}_{\alpha, \beta \in \mathbb{R}, \alpha \neq 1 \vee \beta \neq 1}$$

are families of solutions of the problem. In the Figure 1, we show the graphics of $f(\varepsilon)$ for the two problems considered.

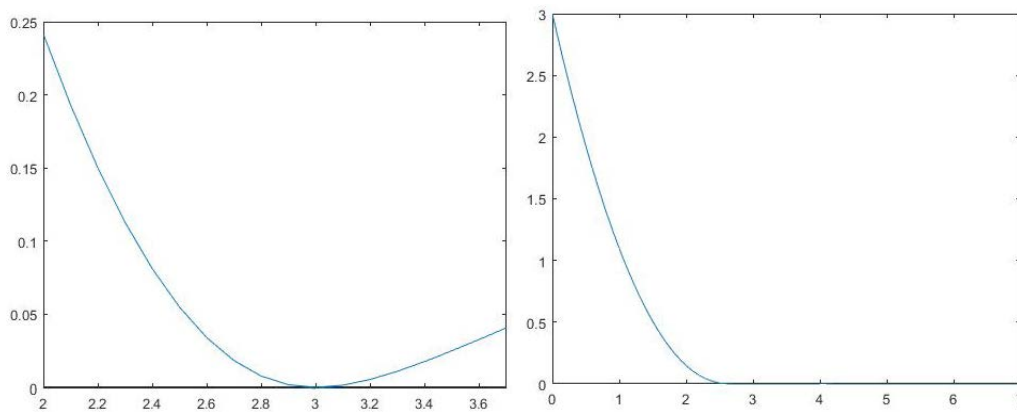


Figure 1. The figure on the left represents the graphic of $f(\varepsilon)$ when we consider the problem of recovering M by a rank-1 matrix. The figure in the right shows that $f(\varepsilon)$ is identically equals to zero, when $\varepsilon \geq \varepsilon^*$, if we require the rank of the solution to be equal to 2, when we compete Y .

The paper is structured as follows. In Sections 3 and 5 we discuss the two level method designed to solve the problem (1). A characterization of local extremizers for the functional (2) is given in Section 4. In Section 6 we present a suitable splitting method for rank- r matricial ODEs employed in the context of the inner iteration. Finally, numerical experiments are showed in Section 7.

3. Differential Equation for E

3.1. Minimizing $F_\varepsilon(E)$ for Fixed ε

Suppose that $E(t)$ is a smooth matrix valued function of t . (We omit the argument t of $E(t)$). Our goal is to find an optimal direction $\dot{E} = Z$ (see [19,20]) such that the functional (2) is characterized by the maximal local decrease, in a way that the matrix E remains in the manifold \mathcal{M}_r .

To deal with this goal, we differentiate (2) with respect to t

$$\frac{d}{dt} F_\varepsilon(E) = \frac{d}{dt} \frac{1}{2} \|\Omega \circ (\varepsilon E - M)\|_F^2 = \frac{1}{2} \frac{d}{dt} \langle \Omega \circ (\varepsilon E - M), \Omega \circ (\varepsilon E - M) \rangle = \varepsilon \langle \Omega \circ \dot{E}, \Omega \circ (\varepsilon E - M) \rangle$$

Setting

$$G := \Omega \circ (\varepsilon E - M) \tag{3}$$

and since by definition of Ω , $\Omega \circ \Omega = \text{Id}$, it is clear that $\Omega \circ G = G$.

Thus, we have

$$\langle \Omega \circ \dot{E}, \Omega \circ G \rangle = \langle \Omega \circ \dot{E}, G \rangle = \sum_{i,j} \Omega_{ij} \dot{E}_{ij} G_{ij} = \sum_{i,j} \dot{E}_{ij} \Omega_{ij} G_{ij} = \langle \dot{E}, \Omega \circ G \rangle = \langle \dot{E}, G \rangle$$

Hence, we have

$$\frac{d}{dt} F_\varepsilon(E(t)) = \varepsilon \langle \dot{E}, G \rangle \tag{4}$$

which identifies G as the free gradient of the functional. We have now to include the constraint $\|E\|_F^2 = 1$. By differentiation

$$\frac{d}{dt} \|E\|_F^2 = 0 \Rightarrow \langle E, \dot{E} \rangle = 0$$

we gain a linear constraint for \dot{E} . By virtue of the rank constraint in (5), we must guarantee that the motion of E remains in the manifold \mathcal{M}_r for all t . In order to get it, we require the derivative \dot{E} to lie in the tangent space in E to \mathcal{M}_r , for all t . These considerations led us to the following optimization problem.

$$Z_* = \arg \min_{\|Z\|_F=1, \langle E, Z \rangle=0, Z \in \mathcal{T}_E \mathcal{M}_r} \langle Z, G \rangle \tag{5}$$

where $\mathcal{T}_E \mathcal{M}_r$ denotes the tangent space to \mathcal{M}_r at E . The constraint $\|Z\|_F = 1$ is simply introduced to get a unique direction Z . In the following, we will denote by $P_E(\cdot)$ the orthogonal projection on $\mathcal{T}_E \mathcal{M}_r$.

3.2. Rank- r Matrices and Their Tangent Matrices

See [21]. Every real rank- r matrix E of dimension $n \times m$ can be written in the form

$$E = USV^T \tag{6}$$

where $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{m \times r}$ have orthonormal columns, i.e.,

$$U^T U = I_r, \quad V^T V = I_r$$

and nonsingular $S \in \mathbb{R}^{r \times r}$. In particular, when S is diagonal, we find the SVD. The decomposition (6) is not unique; simply replacing U by $\bar{U} = UP$ and V by $\hat{V} = VQ$ with orthogonal matrices $P, Q \in \mathbb{R}^{r \times r}$

and S by $\hat{S} = P^T S Q$, we get the same matrix $E = USV^T = \hat{U}\hat{S}\hat{V}^T$. However, we can make the decomposition unique in the tangent space. For all E in the manifold \mathcal{M}_r , let us consider the tangent space $\mathcal{T}_E \mathcal{M}_r$. It is a linear space and every tangent matrix is of the form

$$\dot{E} = \dot{U}SV^T + U\dot{S}V^T + US\dot{V}^T$$

where $\dot{S} \in \mathbb{R}^{r \times r}$, $U^T \dot{U}$ and $V^T \dot{V}$ are skew-symmetric $r \times r$ matrices. $\dot{S}, \dot{U}, \dot{V}$ are uniquely determined by \dot{E} and U, V, S by imposing the gauge conditions

$$U^T \dot{U} = 0, \quad V^T \dot{V} = 0$$

We consider the following important result from [21], thanks to which it is possible to obtain a formula for the projection of a matrix onto the tangent space to a rank- r matrices.

Lemma 1. *The orthogonal projection onto the tangent space $\mathcal{T}_E \mathcal{M}_r$ at $E = USV^T \in \mathcal{M}_r$ is given by*

$$P_E(Z) = Z - (I - UU^T)Z(I - VV^T)$$

for $Z \in \mathbb{R}^{n \times m}$.

3.3. Steepest Descent Dynamics

Lemma 2. *Let $E \in \mathbb{R}^{n \times m}$ be a real matrix of unit Frobenius norm, such that it is not proportional to $P_E(G)$. Then, the solution of (5) is given by*

$$\mu Z_* = -P_E(G) + \langle E, P_E(G) \rangle E \tag{7}$$

where μ is the reciprocal of the Frobenius norm on the right-hand side.

Proof. Let be $E_\perp = \{Z \in \mathbb{R}^{n \times m} : \langle E, Z \rangle = 0\}$. The function $\langle Z, G \rangle$ is an inner product and the feasible region $\mathcal{R} = E_\perp \cap \mathcal{T}_E \mathcal{M}_r$ is a linear subspace, since it is intersection of subspaces. By observing that the inner product with a given vector is minimized over a subspace by orthogonally projecting the vector onto the subspace, we can say that the solution of (5) is a matrix proportional to the normalized orthogonal projection of the free gradient G onto \mathcal{R} . Therefore,

$$P_{\mathcal{R}}(G) = P_{E_\perp}(P_E(G)) = P_E(G) - \frac{\langle E, P_E(G) \rangle}{\langle E, E \rangle} P_E(G)$$

Note that $P_{\mathcal{R}}(G) = P_{E_\perp}(P_E(G))$, since P_{E_\perp} and P_E commute. Since $\|E\|_F = 1$, we have that the solution is given by (7). \square

The Expression (4), jointly with the Lemma 2 suggest to consider the following gradient system for $F_\varepsilon(E)$

$$\dot{E} = -P_E(G) + \langle E, P_E(G) \rangle E \tag{8}$$

To get the differential equation in a form involving the factors in $E = USV^T$, we use the following result.

Lemma 3 (See [21]). *For $E = USV^T \in \mathcal{M}_r$, with nonsingular $S \in \mathbb{R}^{r \times r}$ and with $U \in \mathbb{R}^{n \times r}$ and $\mathbb{R}^{n \times r}$ having orthonormal columns, the equation $\dot{E} = P_E(Z)$ is equivalent to $\dot{E} = \dot{U}SV^T + U\dot{S}V^T + US\dot{V}^T$, where*

$$\begin{cases} \dot{S} = U^T Z V, \\ \dot{U} = (I - UU^T) Z V S^{-1}, \\ \dot{V} = (I - VV^T) Z^T V S^{-T} \end{cases}$$

In our case $Z = -P_E(G) + \langle E, P_E(G) \rangle E$, this yields that the differential Equation (8) for $E = USV^T$ is equivalent to the following system of differential equations for S, V, U ,

$$\begin{cases} \dot{S} = U^T(-P_E(G) + \langle E, P_E(G) \rangle E)V, \\ \dot{U} = (I - UU^T)(-P_E(G) + \langle E, P_E(G) \rangle E)VS^{-1}, \\ \dot{V} = (I - VV^T)(-P_E(G) + \langle E, P_E(G) \rangle E)^T VS^{-T} \end{cases} \tag{9}$$

The following monotonicity result is an immediate consequence of the fact that the differential equation is the gradient flow for F_ϵ on the manifold of matrices, of fixed rank r , and unit norm.

Theorem 1. *Let be $E(t) \in \mathcal{M}_r$ a solution of unit Frobenius norm of the matrix differential Equation (8). Then*

$$\frac{d}{dt} F_\epsilon(E(t)) \leq 0$$

Proof. By Cauchy-Schwarz inequality,

$$|\langle E, P_E(G) \rangle| \leq \|E\|_F \|P_E(G)\|_F = \|P_E(G)\|_F$$

Therefore, using (8),

$$\begin{aligned} \frac{d}{dt} F_\epsilon(E(t)) &= \epsilon \langle \dot{E}, G \rangle = \epsilon \langle -P_E(G) + \langle E, P_E(G) \rangle E, G \rangle = \epsilon (-\langle P_E(G), G \rangle + \langle E, P_E(G) \rangle \langle E, G \rangle) = \\ &= \epsilon (-\langle P_E(G), P_E(G) \rangle + \langle E, P_E(G) \rangle^2) = \epsilon (-\|P_E(G)\|_F^2 + \langle E, P_E(G) \rangle^2) \leq 0 \end{aligned}$$

□

4. Stationary points

Since we are interested to minimize $F_\epsilon(E)$, we focus on the equilibria of (8), which represents local minima of (2).

Lemma 4. *The following statements are equivalent along the solutions of (8):*

- (a) $\frac{d}{dt} F_\epsilon(E) = 0$.
- (b) $\dot{E} = 0$.
- (c) E is a real multiple of $P_E(G)$.

Proof. From the expression (4), clearly (b) implies (a).

Supposing (c), we can write $P_E(G) = \alpha E$ with $\alpha \in \mathbb{R}$ and by substitution in (8) we get

$$\dot{E} = -\alpha E + \langle E, \alpha E \rangle E = \alpha (-E + \|E\|_F E) = \alpha (-E + E) = 0$$

that is (b). So, it remains to show that (a) implies (c).

Note that:

$$\begin{aligned} \frac{d}{dt} F_\epsilon(E(t)) &= \epsilon \langle \dot{E}, G \rangle = \epsilon \langle -P_E(G) + \langle E, P_E(G) \rangle E, G \rangle \\ &= \epsilon (-\langle P_E(G), G \rangle + \langle E, P_E(G) \rangle \langle E, G \rangle) = \epsilon (-\|P_E(G)\|_F^2 + \langle E, P_E(G) \rangle^2) \end{aligned}$$

So, since $\epsilon > 0$, we have

$$\frac{d}{dt} F_\epsilon(E(t)) = 0 \iff -\|P_E(G)\|_F^2 + \langle E, P_E(G) \rangle^2 = 0$$

the last equality holds only if $E = \alpha P_E(G)$ for $\alpha \in \mathbb{R}$, that is (c). \square

The following result characterizes the local extremizers.

Theorem 2. *Let $E_* \in \mathcal{M}_r$ be a real matrix of unit Frobenius norm. Then, the following two statements are equivalent:*

(a) *Every differentiable path $E(t) \in \mathcal{M}_r$ (for small $t \geq 0$) with $\|E\|_F = 1$ and $E(0) = E_*$ satisfies*

$$\frac{d}{dt} F_\varepsilon(E(t)) \geq 0$$

(b) *There exists a $\gamma > 0$ such that*

$$E_* = -\gamma P_E(G)$$

Proof. The strategy of the proof is similar to [22]. Assume that (a) does not hold. Then, there exists a path $E(t) \in \mathcal{M}_r$ through E_* such that $\frac{d}{dt} F_\varepsilon(E(t))|_{t=0} < 0$. Thus, Lemma 2 shows that also the solution path of (8) passing through E_* is such a path. So E_* is not a stationary point of (8), and according to the Lemma 4, it is not a real multiple of $P_E(G)$.

If E_* is not a multiple of $P_E(G)$, then E_* is not a stationary point of (8) and Theorem 1 and Lemma 4 ensure that $\frac{d}{dt} F_\varepsilon(E(t)) \leq 0$ along the solution path of (8). If $E_* = \gamma P_E(G)$ with $\gamma \geq 0$, we can consider the path $E(t) = (1 - t)E_*$ for small $t \geq 0$. This path is such that

$$\|E\|_F = \|(1 - t)E_*\|_F = |1 - t|\|E_*\|_F \leq 1$$

and,

$$\frac{d}{dt} E(t) = -E_*$$

So, we have

$$\begin{aligned} \frac{d}{dt} F_\varepsilon(E(t)) &= \varepsilon \langle \dot{E}, G \rangle = -\varepsilon \langle E_*, G \rangle = -\varepsilon \gamma \langle P_E(G), G \rangle = \\ &= -\varepsilon \gamma \langle P_E(G), P_E(G) \rangle = -\varepsilon \gamma \|P_E(G)\|_F^2 < 0 \end{aligned}$$

in contradiction with (a). \square

5. Numerical Solution of Rank-r Matrix Differential Equation

We have seen that the matrix ODE (8) is equivalent to the system (9), involving the factors of the decomposition (6). In (9), the inverse of S appear. Therefore, when S is nearly singular, problems of stability can arise, working with a standard numerical methods for ODEs. To avoid this difficulties, we employ the first order projector-splitting integrator of [23]. The algorithm directly approximate the solution of the Equation (8). It starts from the normalized rank r matrix $E_0 = U_0 S_0 V_0^*$ at the time t_0 , obtained by the SVD of the matrix to recover. At the time $t_1 = t_0 + h$, one step of the method works as follows

Projector-splitting integrator	
Data: $E_0 = U_0 S_0 V_0^*, G_0 = G(E_0)$	% G is the free gradient (3)
Result: E_1	
begin ;	
Set $K_1 = U_0 S_0 - G_0 V_0 ;$	
Compute $U_1 \widehat{S}_1 = K_1 ;$	% QR factorization % U_1 orthonormal columns % \widehat{S}_1 $r \times r$ matrix
Set $\overline{S}_0 = \widehat{S}_1 + h U_1^T G_0 V_0 ;$	
Set $L_1 = V_0 \overline{S}_0^T - h G_0^T U_1 ;$	
Compute $V_1 S_1^T = L_1 ;$	% QR factorization % V_1 orthonormal columns % S_1 $r \times r$ matrix
Set $\widehat{E}_1 = U_1 S_1 V_1^T ;$	
Normalize $E_1 = \frac{\widehat{E}_1}{\ E_1\ _F} ;$	

E_1 is taken as approximation to $E(t_1)$. All the nice features of the integrator are presented in [23], but it is already clear that, there is no matrix inversion in the steps of the algorithm.

6. Iteration on ϵ

In this section we show the outer iteration to manage ϵ (see [22]).

6.1. Qualitative Tools

For every fixed $\epsilon > 0$, the gradient system (8) returns a stationary point $E(\epsilon)$ of unit Frobenius norm that is a local minimum of F_ϵ .

Setting $f(\epsilon) = F_\epsilon(E(\epsilon))$, our purpose is to solve the problem

$$\min\{\epsilon > 0 : f(\epsilon) = 0\}$$

employing a Newton-like method. We assume that $E(\epsilon)$ is a smooth function of ϵ , so that, also the function $f(\epsilon) = F_\epsilon(E(\epsilon))$ is differentiable with respect to ϵ . Let us focus on its derivative,

$$\begin{aligned} f'(\epsilon) &= \frac{d}{d\epsilon} F_\epsilon(E(\epsilon)) = \frac{d}{d\epsilon} \|\Omega \circ (\epsilon E - M)\|_F^2 = \frac{d}{d\epsilon} \langle \Omega \circ (\epsilon E - M), \Omega \circ (\epsilon E - M) \rangle = \\ &= \left\langle \frac{d}{d\epsilon} (\Omega \circ (\epsilon E - M)), \Omega \circ (\epsilon E - M) \right\rangle = \left\langle \Omega \circ \left(\frac{d}{d\epsilon} (\epsilon E(\epsilon)) \right), \Omega \circ (\epsilon E - M) \right\rangle = \\ &= \left\langle \frac{d}{d\epsilon} (\epsilon E(\epsilon)), \Omega \circ (\epsilon E - M) \right\rangle = \langle E(\epsilon) + \epsilon E'(\epsilon), G \rangle \end{aligned}$$

If we denote $\epsilon^* = \min\{\epsilon > 0 : f(\epsilon) = 0\}$.

By the expression of the free gradient (3), it is clear that

$$0 = f(\epsilon^*) = \frac{1}{2} \|\Omega \circ (\epsilon^* E^* - M)\|^2 = \frac{1}{2} \|G(\epsilon^*)\|^2 \Leftrightarrow G(\epsilon^*) = 0$$

Therefore,

$$f'(\epsilon^*) = \langle E(\epsilon^*) + \epsilon^* E'(\epsilon^*), G(\epsilon^*) \rangle = \langle E(\epsilon^*) + \epsilon^* E'(\epsilon^*), 0 \rangle = 0$$

This means that ϵ^* is a double root for $f(\epsilon)$.

6.2. Numerical Approximation of ε^*

The presence of the double root ensures that $f(\varepsilon)$ is convex for $\varepsilon \leq \varepsilon^*$, therefore, the classical Newton method will approach ε^* from the left. In this case, we are not able to find an analytical formulation for the derivative $f'(\varepsilon)$, so we approximate it with backward finite differences.

Algorithm for computing ε^*

Data: the matrix M to recover is given, $tol, \varepsilon_0, \varepsilon_1$ such that $f(\varepsilon_0) > 0, f(\varepsilon_1) > 0$
Result: $\varepsilon_*, E(\varepsilon_*)$
begin ;
 $k = 1;$
while $|\varepsilon_k - \varepsilon_{k-1}| > tol$ **do**
 Compute $f'_k \approx \frac{f_k - f_{k-1}}{\varepsilon_k - \varepsilon_{k-1}};$
 Update $\varepsilon_{k+1} = \varepsilon_k - \frac{f_k}{f'_k};$
 Compute $f_{k+1} = f(\varepsilon_{k+1});$
 $k = k + 1;$
end
 $\varepsilon_k = \varepsilon^* ;$

7. Numerical Experiments

In the following experiments we randomly generate some matrices of low rank. As in [3,4], r is the fixed rank, we generate two random matrices $A_L, A_R \in \mathbb{R}^{n \times r}$ with i.i.d. standard Gaussian entries. We build the matrix $A = A_L A_R^T$ and generate a uniformly distributed sparsity pattern Ω . We work on the matrix M , that is the matrix resulting from the projection of A onto the pattern Ω . In this way we are able to compare the accuracy of the matrix solution of our code with the true solution A . As stopping criteria for the integrator of the ODE in the inner level, we use

$$\|\Omega \circ (X - M)\|_F / \|M\|_F < tol$$

where tol is an input tolerance parameter together with a maximum number of iterations and a minimum value for the integrator stepsize. We provide a stepsize control that reduces the step h with a factor γ (the default value is 1.25), when the functional is not decreasing, but increases the step as $h\gamma$ when the value of the objective decreases with respect to the previous iteration. Some computational results are shown in the Tables 1 and 2. In particular, they show the values of the cost function evaluated in ε^* , computed by the outer method, thanks to which, the accuracy of the method when we recover matrices of different rank and different dimension is highlight.

Table 1. Computational results from recovering three matrices of different dimensions and 30% of known entries. The rank is fixed to be 10.

Dim	$f(\varepsilon_*)$	Err	Iter
2000×300	1.0944×10^{-24}	1.8960×10^{-12}	6
2000×650	2.0948×10^{-24}	2.5971×10^{-12}	5
2000×1000	1.3071×10^{-23}	6.4837×10^{-12}	7

Table 2. Computational results from recovering three matrices of different ranks and 30% of known entries. The dimension is always 1000×1000 .

Rank	$f(\epsilon_*)$	Err	Iter
10	3.6533×10^{-24}	1.0079×10^{-12}	10
20	7.6438×10^{-24}	4.9793×10^{-12}	9
30	6.4411×10^{-24}	6.6683×10^{-12}	9

7.1. Computational Variants of the Outer Level

Observe that the presence of the double root would allow us to use a modified Newton iteration (from the left)

$$\epsilon_{k+1} = \epsilon_k - 2 \frac{f(\epsilon_k)}{f'(\epsilon_k)}$$

getting quadratic convergence. Since our purpose is to find an upper bound for ϵ_* , if it should happen that $\epsilon_k > \epsilon_*$, we need a bisection iteration to preserve the approximation from the left. Furthermore, we can observe that, if we indicate by $g(\epsilon) = \|G\|_F$, where G is defined in (3), it is clear that $f(\epsilon) = \frac{1}{2}g^2(\epsilon)$, therefore they have common zeros. This allows us to employ the function $g(\epsilon)$ instead of $f(\epsilon)$ in the outer level, joining classical Newton and bisection. In practice, this results to be the most efficient approach. Tables 3 and 4 show the behaviours of the two alternative approaches on a test matrix M of dimension 150×150 , $\approx 50\%$ of known entries and rank 15.

Table 3. Computational results obtained by coupling modified Newton method and bisection.

Iter	ϵ	$f(\epsilon)$
1	$4.666972133737625 \times 10^2$	145.0744
2	$4.684932414610240 \times 10^2$	122.9347
3	$4.884388614255482 \times 10^2$	0.2481
4	$4.885195311133194 \times 10^2$	0.2074
5	$4.893418324967433 \times 10^2$	4.1842×10^{-4}
\vdots	\vdots	\vdots
18	$4.893805094395407 \times 10^2$	9.4499×10^{-21}
19	$4.893805094397171 \times 10^2$	5.8554×10^{-24}
20	$4.893805094397174 \times 10^2$	5.3080×10^{-24}

Table 4. Computational results got by employing the function $g(\epsilon)$.

Iter	ϵ	$g(\epsilon)$
1	$4.664656983250553 \times 10^2$	17.2083
2	$4.880638859063273 \times 10^2$	0.9850
3	$4.893752126994476 \times 10^2$	0.0040
4	$4.893805081705695 \times 10^2$	9.4925×10^{-7}
5	$4.893805094397189 \times 10^2$	2.0713×10^{-12}
6	$4.893805094397218 \times 10^2$	2.2306×10^{-13}
7	$4.893805094397220 \times 10^2$	3.2750×10^{-13}

We tested also the classic Newton method on M . The comparison is summarized in Tables 5 and 6.

Table 5. Comparison between different approach to the outer iteration. The table show the number of iterations done by each method and the optimal values of the cost function.

Method	Iter	$f(\epsilon_*)$
N2	20	4.9734×10^{-24}
g	7	5.3630×10^{-26}
N	70	9.3812×10^{-26}

Table 6. Comparison between different approach to the outer iteration. The table show the real error and the time.

Method	Err	Time
N2	5.2117×10^{-13}	27.684 s
g	4.4562×10^{-13}	3.6980 s
N	3.7471×10^{-13}	57.648 s

The accuracy is the same for all the choices, but in the case of selecting g instead of f , the computational cost is sharply reduced, both in terms of number of iterations and in terms of timing.

7.2. Experiments with Quasi Low Rank Matrices

The following simulations are devoted to check the “robustness” of the method with respect to small perturbations of the singular values. More precisely, we consider a rank r matrix A , built as introduced in this section, and we perturb it in order to get a matrix A_p of almost rank r . In other words, we aim to get a matrix A_p that has r significant singular values, whereas the remaining ones become very small. Let $A = U\Sigma V^T$ be the SVD decomposition of A , therefore Σ is diagonal with only the first r diagonal values different from zero. If $\hat{\Sigma}$ is the diagonal matrix such that the first r diagonal entries are zero and the remaining ones (all or a part of them) are put equal to random small values, we build A_p as

$$A_p = U(\Sigma + \hat{\Sigma})V^T$$

where,

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & \\ & \cdot & & & \\ & & \sigma_r & & \\ & & & 0 & \\ & 0 & & & 0 \end{pmatrix}, \hat{\Sigma} = \begin{pmatrix} 0 & & & & \\ & \cdot & & & \\ & & 0 & & \\ & & & \hat{\sigma}_{r+1} & \\ 0 & & & & \hat{\sigma}_n \end{pmatrix}$$

Table 7 shows the numerical results, obtained by considering a rank 9 matrix A , of size 200 and perturbations of different amplitude.

Table 7. Computational results from recovering three matrices of different ranks and 30% of known entries. The dimension is always 1000×1000 .

$\hat{\sigma}_{r+1,r+1}$	ϵ_*	$f(\epsilon_*)$	Err
$\approx 1 \times 10^{-4}$	$6.167926656792961 \times 10^2$	2.4002×10^{-5}	0.0037
$\approx 1 \times 10^{-6}$	$6.167937831641854 \times 10^2$	1.9209×10^{-9}	7.2224×10^{-5}
$\approx 1 \times 10^{-8}$	$6.167937831639130 \times 10^2$	1.9211×10^{-13}	2.8729×10^{-7}
0	$6.167937832319908 \times 10^2$	2.1332×10^{-23}	7.7963×10^{-12}

The columns $\hat{\sigma}_{r+1,r+1}$ and err contain the orders of magnitude of the greater perturbed singular value, and the values of the real error, computed as the Frobenius distance between A and the optimal

matrix, that comes out the code, respectively. As it is natural to expect, the optimal matrix remains close to A , but the error is affected by the perturbations as they grow.

Another interesting example in terms of robustness, when we work with quasi low rank matrices, is given by considering matrices with exponentially decaying singular values. In particular we build a $n \times n$ matrix A , which singular values are given by the sequence $\{exp(-x_i)\}_{i=1,\dots,n}$ where $x_1 \leq x_2 \leq \dots \leq x_n$ are random increasing numbers in an interval $[a, b]$. We build the matrix M to recover, by projecting A onto a sparsity pattern Ω . In the following experiment we fix $n = 100$, $a = 20$ and $b = 1$. The singular values of A range in the interval $[0.3511, 2.0870 \times 10^{-9}]$, and the mask M has about 30% of known elements. We choose the values of rank in the experiments for the completion by considering the changes of order of magnitude of singular values. Table 8 shows the results, in particular, the value of the cost function $f(\epsilon_*)$ which we compare to f , the one given by the code of [4].

Table 8. The table show the behaviours of the codes when we recover the matrix M fixing different values of the rank, accordingly with the order of magnitude of the singular value of the exact full rank solution A .

r	$\hat{\sigma}_{r+1,r+1}$	$f(\epsilon_*)$	f
4	$\approx 1 \times 10^{-2}$	0.0061	0033
20	$\approx 1 \times 10^{-3}$	4.98621×10^{-11}	4.2904×10^{-8}
24	$\approx 1 \times 10^{-4}$	8.98261×10^{-32}	6.0490×10^{-16}

7.3. Experiment with Theoretical Limit Number of Samples

In the seminal paper [1], the authors focus on determine a lower bound for the cardinality Ω of the known set of entries, such that it is possible recovering the matrix with high probability. In particular they proved that, most of $n \times n$ matrix of rank r (assumed not too large) can be perfectly recovered solving a convex optimization problem, if $|\Omega| \geq Cn^{1.2}r \log n$, for some positive constant C . Table 9 shows the results when we compare our code with the method in [4]. In particular, we present the best value of the objective functions, the real errors and the computational times. We consider $n = 50$, $r = 3$, therefore, according with the previous bound, we have to set $|\Omega| \geq C \cdot 1.2832 \times 10^3$. This means that, for $C = 1$, the corresponding target matrix M will have $\approx 51.33\%$ of given entries.

Table 9. The table show the behaviours of the codes when we recover the 50×50 matrix M with $\approx 49.88\%$ of given elements. The rank is 3. Our results are marked by an asterisk.

f	Err	Time
1.951391×10^{-26}	3.37141×10^{-13}	0.066 s
f^*	Err*	Time*
4.08441×10^{-27}	1.0071×10^{-13}	2.28 s

7.4. Behaviour with Respect to Different Ranks

Given a test matrix M , our purpose, in this section, is to understand the behaviour of the cost function, when we set the rank different from the exact one. In particular, we consider a matrix M , with $\approx 44.71\%$ of known elements, of dimension 70×70 , and such that the exact rank of the solution is 8. We compute the values of the cost function evaluated in the best fixed rank k approximation (say b^k) and in the solution given by our code (say f^k). For every fixed rank k , the error is given by

$$|b^k - f^k| / \|M\|_F^2$$

The results are shown in the Table 10(a).

Table 10. The table shows the values of the cost function for different value of the rank and the relative errors. The true rank is 8. The values of the objective for the different ranks are represented on the figure (b).

(a)			(b)
Rank	f	Err	
1	6.73461×10^3	0.0008	
2	5.07831×10^3	0.0086	
3	3.57211×10^3	0.0136	
4	2.35091×10^3	0.0177	
5	1.53641×10^3	0.0029	
6	0.9613	0.0070	
7	0.4461	0.0019	
8	9.1260×10^{-31}	0.22641×10^{-7}	
9	1.6394×10^{-30}	2.76671×10^{-7}	
10	3.9834×10^{-15}	2.83831×10^{-7}	

8. Conclusions

The matrix completion problem consists of recovering a matrix from a few samples of its entries. We formulate the problem as a minimization of the Frobenius distance on the set of the fixed entries, over the manifold of the matrices of fixed rank. In this paper we introduce a numerical technique to deal with this problem. The method works on two levels; the inner iteration computes the fixed norm matrix that best fits the data entries, solving low rank matrix differential equations, the outer iteration optimizes the norm by employing a Newton-like method. A key feature of the method is to avoid the problem of the lack of vector space structure for \mathcal{M}_r , moving the dynamics in the tangent space. Numerical experiments show the high accuracy of the method and its robustness with respect to small perturbations of the singular values. However, in presence of very challenging problems it could be suitable to relax tolerance parameters. The method is particularly suited for problems for which guessing the rank is simple. In the field of research on low rank matrix completion, it would be useful to study real databases types of matrices in order to try to establish gaps for the values of the rank. Moreover, since this is a typical context, in which we work with very large matrices, future work could be devoted to develop methods working in parallel. Structured variants, such as nonnegative low rank completions, are suggested from applications. These may be subject of a future work.

Author Contributions: Both authors developed the theoretical part in order to provide the treatment of the problem and obtain a solution. In particular, N.G. suggested the problem and the adopted formulation, and supervised the organisation and the coherence of the work. C.S. focused on the numerical technical part of the work.

Funding: The authors thank INdAM GNCS (Gruppo Nazionale di Calcolo Scientifico) for financial support.

Acknowledgments: The authors thank four anonymous referees for their valuable remarks.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Candés, E.J.; Recht, B. Exact Matrix Completion via Convex Optimization. *Found. Comput. Math.* **2009**, *9*, 717–772. [[CrossRef](#)]
2. Gillis, N.; Glineur, F. Low-rank matrix approximation with weights or missing data is NP-hard. *SIAM J. Matrix Anal. Appl.* **2011**, *4* 1149–1165. [[CrossRef](#)]
3. Cai, J.F.; Candés, E.J.; Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* **2010**, *20*, 1956–1982. [[CrossRef](#)]

4. Vandereycken, B. Low-Rank matrix completion by Riemaniann optimization. *SIAM J. Optim.* **2013**, *23*, 367–384. [[CrossRef](#)]
5. Cambier, L.; Absil, P.-A. Robust Low-Rank Matrix Completion by Riemannian Optimization. *SIAM J. Sci. Comput.* **2016**, *38*, 440–460. [[CrossRef](#)]
6. Keshavan, R.H.; Montanari, A.; Oh, S. Matrix Completion from a Few Entries. *IEEE Trans. Inf. Theory* **2010**, *38*, 440–460. [[CrossRef](#)]
7. Yang, Y.; Feng, Y.; Suykens, J.A.K. Correntropy Based Matrix Completion. *Entropy* **2018**, *20*, 171. [[CrossRef](#)]
8. Candés, E.J.; Plan, Y. Matrix completion with noise. *Proc. IEEE* **2010**, *98*, 925–936. [[CrossRef](#)]
9. Goldberg, D.; Nichols, D.; Oki, B.M.; Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM* **1992**, *35*, 61–70. [[CrossRef](#)]
10. Rodger, A. Toward reducing failure risk in an integrated vehicle health maintenance system: A fuzzy multi-sensor data fusion Kalman filter approach for IVHMS. *Expert Syst. Appl.* **2012**, *39*, 9821–9836. [[CrossRef](#)]
11. Bennet, J.; Lanning, S. The netflix prize. In Proceedings of the KKD Cup and Workshop, San Jose, CA, USA, 12 August 2007; p. 35.
12. Nguyen, L.T.; Kim, S.; Shim, B. Localization in the Internet of Things Network: A Low-Rank Matrix Completion Approach. *Sensors* **2016**, *16*, 722. [[CrossRef](#)] [[PubMed](#)]
13. Wang, X.; Weng, Z. Low-rank matrix completion for array signal processing. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012.
14. Cai, M.; Cao, F.; Tan, Y. Image Interpolation via Low-Rank Matrix Completion and Recovery. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1261–1270.
15. Natarajan, N.; Dhillon, I.S. Inductive Matrix Completion for Predicting Gene-Disease Associations. *Bioinformatics* **2014**, *30*, i60–i68. [[CrossRef](#)] [[PubMed](#)]
16. Bakonyi, M.; Woederman, H.J. *Matrix Completion, Moments, and Sums of Hermitian Squares*; Princeton University Press: Princeton, NJ, USA, 2011.
17. Markovsky, I.; Usevich, K. Structured Low-Rank Approximation with Missing Data. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 814–830. [[CrossRef](#)]
18. Mishra, B.; Meyer, G.; Sepulchre, R. Low-rank optimization for distance matrix completion. In Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 12–15 December 2011.
19. Guglielmi, N.; Lubich, C. Low-rank dynamics for computing extremal points of real pseudospectra. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 40–66. [[CrossRef](#)]
20. Guglielmi, N.; Lubich, C. Differential equations for roaming pseudospectra: paths to extremal points and boundary tracking. *SIAM J. Numer. Anal.* **2011**, *49*, 1194–1209. [[CrossRef](#)]
21. Koch, O.; Lubich, C. Dynamical low-rank approximation. *SIAM J. Matrix Anal. Appl.* **2007**, *29*, 434–454. [[CrossRef](#)]
22. Guglielmi, N.; Lubich, C.; Mehrmann, V. On the nearest singular matrix pencil. *SIAM J. Matrix Anal. Appl.* **2017**, *38*, 776–806. [[CrossRef](#)]
23. Lubich, C.; Oseledets, I.V. A projector-splitting integrator for dynamical low-rank approximation. *Numer. Math.* **2014**, *54*, 171–188 [[CrossRef](#)]

