Convex Envelopes for Low Rank Approximation

Viktor Larsson and Carl Olsson

Centre for Mathematical Sciences
Lund University, Sweden

Abstract. In this paper we consider the classical problem of finding a low rank approximation of a given matrix. In a least squares sense a closed form solution is available via factorization. However, with additional constraints, or in the presence of missing data, the problem becomes much more difficult. In this paper we show how to efficiently compute the convex envelopes of a class of rank minimization formulations. This opens up the possibility of adding additional convex constraints and functions to the minimization problem resulting in strong convex relaxations. We evaluate the framework on both real and synthetic data sets and demonstrate state-of-the-art performance. [1]

1 Introduction

The assumption that measurements consist of noisy observations from a low rank matrix has been proven useful in applications such as non-rigid and articulated structure from motion [1–3], photometric stereo [4] and optical flow [5]. The interpretation of the low rank assumption is that the observed data can be written as a linear combination of a few basis elements. The factorization approach, introduced to vision in [6], offers a simple way of determining both coefficients and basis elements. If the measurement matrix M is complete then the best approximation, in a least squares sense, can be computed in closed form [7] using the singular value decomposition (SVD). The main drawback is that the computation of a factorization requires a complete measurement matrix. In structure from motion this means that every point has to be visible in every image, something that rarely occurs in practice due to occlusions and tracking failures. In case there are missing entries and/or outliers the optimization problem is substantially more difficult.

The issue of outliers has received a lot of attention lately. In [8, 9] the more robust L_1-norm is considered. These methods build on the so called Wiberg algorithm [10] which jointly optimizes a product UV^T of two fixed size U and V matrices. As a consequence the quality of the result is dependent on initialization. Another approach [11, 3, 12] tackles the problem of missing data by replacing the rank constraint with the weaker but convex nuclear norm penalty and solves

$$\min_X \mu\|X\|_* + \|W \odot (X - M)\|_F^2, \tag{1}$$

[1] This work has been funded by the Swedish Research Council (grant no. 2012-4213) and the Crafoord Foundation.

where $W_{ij} = 0$ if the entry is missing and 1 otherwise. This approach is convex and therefore independent of initialization. In addition it can be shown that if the locations of the missing entries are random the approach gives the best low rank approximation [11]. The typical patterns of missing data in structure from motion still pose a problem for these approaches.

The motivation for using the nuclear norm in (1) is that it is the convex envelope of the rank function on the set $\{X; \sigma_{max}(X) \leq 1\}$. The constraint $\sigma_{max}(X) \leq 1$ is however artificial and not present in (1). In [13] it is shown that the so called localized rank function

$$f(X) = \mu \operatorname{rank}(X) + \|X - X_0\|_F^2, \tag{2}$$

has the convex envelope

$$f^{**}(X) = \sum_{i=1}^{n} \left(\mu - [\sqrt{\mu} - \sigma_i(X)]_+^2 \right) + \|X - X_0\|_F^2. \tag{3}$$

Note that the regularizer in (3) itself is not convex. The second term, enables a proportionally smaller penalty for large singular values, without losing convexity, giving a tighter convex envelope in the neighborhood of X_0. In fact, in contrast to the nuclear norm heuristic, minimizing (3) gives the same result as solving (2) with SVD. The advantage of using (3) is that it is convex and therefore can be combined with other convex constraints and functions. In [13] the missing data problem is solved by minimizing (3) on complete sub-blocks and enforcing agreement on the overlaps via linear constraints.

The formulation in [13] consists of a trade-of between matrix rank and data fit. In many cases it is of interest to search for a matrix of known fixed rank. For example for rigid structure from motion the measurement matrix is known to be of rank 4 (or 3 if the translation can be eliminated) [6]. In such cases the approach of solving (3) on sub-blocks requires determining an appropriate weight μ for each sub-block that gives the correct rank. In this paper we show that we can incorporate such knowledge by replacing (2) with

$$f_g(X) = g(\operatorname{rank}(X)) + \|X - X_0\|^2. \tag{4}$$

In particular we are interested in the case where

$$g(\operatorname{rank}(X)) = \mu \max(r_0, \operatorname{rank}(X)), \tag{5}$$

but our theory applies to a larger class of problems as well. The only requirement that we make is that g is a non-decreasing convex function.

The reason for considering (5) is that in case we know the rank of the sought matrix we can simply let μ be large thus avoiding iteration over the parameters which is done in [13]. Consequently our approach is essentially parameter free. The max term also effectively reduces bias towards low rank solutions like the zero solution that are often uninteresting, giving a tighter convex relaxation. Our main contribution is the computation of the convex envelope of (4) and its proximal operator. While the formulation does not admit closed form solutions we give simple and fast algorithms for evaluations. In addition we present a way of strengthening the convex envelopes using a trust-region formulation.

Notation. Throughout the paper we use $\sigma_i(X)$, $i = 1, ..., n$ to denote the ith singular value of a matrix X. Here n denotes the number of singular values and for notational convenience we will also define $\sigma_0(X) = \infty$ and $\sigma_{n+1}(X) = 0$. The vector of all singular values is denoted $\boldsymbol{\sigma}(X)$. With some abuse of notation we write the SVD of X as $U \operatorname{diag}(\boldsymbol{\sigma}(X))V^T$. For ease of notation we do not explicitly indicate the dependence of U and V on X. The scalar product is defined as $\langle X, Y\rangle = \operatorname{tr}(X^T Y)$, where tr is the trace function, and the Frobenius norm $\|X\|_F = \sqrt{\langle X, X\rangle} = \sqrt{\sum_{i=1}^n \sigma_i^2(X)}$. Truncation at zero is denoted $[a]_+$, that is, $[a]_+ = 0$ if $a < 0$ and a otherwise.

2 The Convex Envelope

In this section we compute the envelope of (4). We will assume that the function g can be written

$$g(k) = \begin{cases} g_0 & \text{if} \quad k = 0 \\ g_0 + \sum_{i=1}^k g_i & \text{otherwise} \end{cases}, \tag{6}$$

where the sequence g_i is non-negative and non-decreasing for $1 \leq i \leq n$. It is easy to see that this is possible if g is convex and non decreasing on $\mathbb{R}$. Furthermore, we will assume that $g_0 = 0$ since subtracting a constant from the objective function does not affect the minimizers (and only subtracts a constant from the convex envelope).

We will follow the approach of [13] which computes the bi-conjugate of (2) to find the convex envelope. In contrast to (2), we will not be able to find a closed form solution for the convex envelope of (4). Instead our approach will be to isolate a small set of singular value configurations that can possibly maximize the conjugate function. By numerically searching this solution set we are able to efficiently evaluate the convex envelope and compute its proximal operator.

2.1 The Conjugate Function

The convex envelope can be found by computing the second Fenchel conjugate $f_g^{**} = (f_g^*)^*$, where f_g^* is defined as

$$f_g^*(Y) = \sup_X \langle X, Y\rangle - f_g(X). \tag{7}$$

The calculations for the first conjugate roughly follows those of [13] and we only give the result here. We get that the first conjugate is given by

$$f_g^*(Y) = -\sum_{i=1}^n \min\left(g_i, \sigma_i^2\left(X_0 + \frac{Y}{2}\right)\right) - \frac{1}{2}\|X_0\|_F^2 + \left\|X_0 + \frac{Y}{2}\right\|_F^2. \tag{8}$$

2.2 Evaluation of the Bi-Conjugate

By completing squares and changing variables we get the bi-conjugate

$$f_g^{**}(X) = \mathcal{R}_g(X) + \|X - X_0\|_F^2, \tag{9}$$

where

$$\mathcal{R}_g(X) = \max_Z \left(\sum_{i=1}^{n} \min\left(g_i, \sigma_i^2(Z)\right) - \|Z - X\|_F^2 \right). \tag{10}$$

The next step in determining the convex envelope is to find the maximizing Z in (10). We first note that using von Neumann's trace theorem we can reduce the problem to a search over the singular values of Z. The norm term fulfills

$$-\|Z - X\|_F^2 \leq -\|Z\|_F^2 + 2\sum_{i=1}^{n} \sigma_i(Z)\sigma_i(X) - \|X\|_F^2, \tag{11}$$

with equality if Z and X have the same U and V in their singular value decompositions. Since the sum in (10) does not depend on U or V the optimal Z has to be of the form $Z = U\operatorname{diag}(\boldsymbol{\sigma}(Z))V^T$ if $X = U\operatorname{diag}(\boldsymbol{\sigma}(X))V^T$. This reduces the maximization in (10) to

$$\max_{\boldsymbol{\sigma}(Z)} \left(\sum_{i=1}^{n} \min\left(g_i, \sigma_i^2(Z)\right) - \sum_{i=1}^{n} (\sigma_i(Z) - \sigma_i(X))^2 \right). \tag{12}$$

Note that the elements of $\boldsymbol{\sigma}(Z)$ have to fulfill $\sigma_1(Z) \geq \sigma_2(Z) \geq ... \geq \sigma_n(Z)$ since these are singular values.

Properties of the Optimal $\boldsymbol{\sigma(Z)}$ To limit the search space for maximization over $\boldsymbol{\sigma}(Z)$ we will next derive some properties of the maximizer. Considering each singular value $\sigma_k(Z)$ separately they should solve a program of the type

$$\max_s \min(g_k, s^2) - (s - \sigma_k(X))^2 \tag{13}$$

$$\text{s.t.} \quad \sigma_{k+1}(Z) \leq s \leq \sigma_{k-1}(Z) \tag{14}$$

Note that for $k = 1$ there is no upper bound on s and for $k = n$ there is no positive lower bound since we use the convention that $\sigma_0(Z) = \infty$ and $\sigma_{n+1}(Z) = 0$. We first consider the unconstrained objective function. This function is the point wise minimum of the two concave functions $g_k - (s - \sigma_k(X))^2$ (for $s \geq \sqrt{g_k}$) and $s^2 - (s - \sigma_k(X))^2 = 2s\sigma_k(X) - \sigma_k^2(X)$. The function is concave and attains its optimum in $s = \sigma_k(X)$ if $\sigma_k(X) \geq \sqrt{g_k}$ and in $s = \sqrt{g_k}$ otherwise (see Figure 1). In case $\sigma_k(X) = 0$ the optimum is not unique. For simplicity we will assume that $\sigma_k(X) > 0$ in what follows. The solution we create will still be valid if $\sigma_k(X) = 0$ but might not be unique. Let s_k be the individual unconstrained optimizers of (13), i.e.

$$s_k = \max(\sqrt{g_k}, \sigma_k(X)). \tag{15}$$

Note that this sequence is decreasing when $\sigma_k(X)$ is larger than $\sqrt{g_k}$. We choose k_0 such that s_{k_0} is the smallest value in the sequence s_k.

We now consider the constrained problem (13)-(14). Since $\sigma_{k+1}(Z) \leq \sigma_{k-1}(Z)$ we see that the optimization over $\sigma_k(Z)$ can be limited to three choices

$$\sigma_k(Z) = \begin{cases} s_k & \text{if } \sigma_{k+1}(Z) \leq s_k \leq \sigma_{k-1}(Z) \\ \sigma_{k-1}(Z) & \text{if } \quad \sigma_{k-1}(Z) < s_k \\ \sigma_{k+1}(Z) & \text{if } \quad s_k < \sigma_{k+1}(Z) \end{cases}. \tag{16}$$

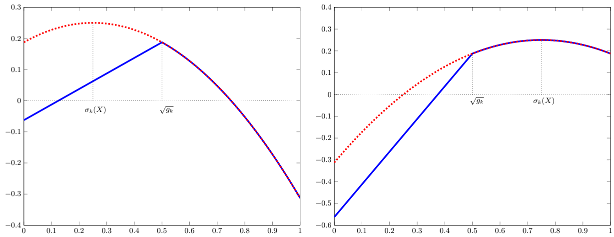

Fig. 1. The objective function in (13) for $\sigma_k(X) \leq \sqrt{g_k}$ and $\sigma_k(X) \geq \sqrt{g_k}$.

Lemma 1. *If Z is an optimal solution to* (12) *then there is a $k \leq k_0$ such that*

$$\sigma_i(Z) = \quad s_i, \qquad \text{if } i < k, \tag{17}$$

$$\sigma_i(Z) = \sigma_k(Z), \quad \text{if } k \leq i \leq k_0. \tag{18}$$

Proof. Using induction we first prove the recursion

$$\sigma_i(Z) = \max(s_i, \sigma_{i+1}(Z)) \quad \text{for } i \leq k_0. \tag{19}$$

For $i = 1$ we see from (16) that s_1 is the optimal choice if $s_1 > \sigma_2(Z)$ otherwise $\sigma_2(Z)$ is optimal. Therefore $\sigma_1(Z) = \max(s_1, \sigma_2(Z))$. Next assume that $\sigma_{i-1}(Z) = \max(s_{i-1}, \sigma_i(Z))$ for some $i \leq k_0$. Then

$$\sigma_{i-1}(Z) \geq s_{i-1} \geq s_i, \tag{20}$$

therefore we can ignore the second case in (16), which proves the recursion (19).

To prove the lemma assume $\sigma_k(Z) \neq s_k$ for some $k \leq k_0$. From (19) it follows that

$$\sigma_k(Z) = \sigma_{k+1}(Z) > s_k. \tag{21}$$

But s_k is decreasing for $k \leq k_0$ which implies that $\sigma_{k+1}(Z) > s_{k+1}$. By repeating the argument it follows that

$$\sigma_k(Z) = \sigma_{k+1}(Z) = \sigma_{k+2}(Z) = ... = \sigma_{k_0}(Z). \tag{22}$$

□

Lemma 2. *If Z is an optimal solution to* (12) *then*

$$\sigma_i(Z) = \sigma_{i+1}(Z), \quad \text{if } i \geq k_0. \tag{23}$$

Proof. Consider $\sigma_i(Z)$ for some $i \geq k_0$. If $\sigma_i(Z) > s_i$ it must have been bounded from below in (16), i.e. $\sigma_i(Z) = \sigma_{i+1}(Z)$. If instead $\sigma_i(Z) \leq s_i$ we have $\sigma_{i+1}(Z) \leq \sigma_i(Z) \leq s_i \leq s_{i+1}$. Then similarly σ_{i+1}(Z) is bounded from above in (16) which implies $\sigma_{i+1}(Z) = \sigma_i(Z)$.

□

Algorithm We now summarize the properties derived in the previous section into an algorithm. Since we do not know which value the k of Lemma 1 will take the algorithm essentially consists of looping over k and testing the obtained solutions for feasibility. Furthermore the operations in each iteration are fast so that in practice the search for k is dominated by other steps such as computation of the singular value decomposition of X.

From the previous section it follows that the optimal solutions $\boldsymbol{\sigma}(Z)$ must have the form

$$\sigma_i(Z) = \begin{cases} \sigma_i(X) & i \leq k \\ s & i > k \end{cases}, \tag{24}$$

for some $k \leq k_0$ and $s \leq \sigma_k(X)$. We can find the optimal k and s by considering the following optimization problem

$$\max_{k \leq k_0} \max_{s} \sum_{i=1}^{k} g_i + \sum_{i=k+1}^{n} \min(s^2,\ g_i) - \sum_{i=k+1}^{n} (s - \sigma_i(X))^2. \tag{25}$$

For a fixed $k < k_0$ it follows from Lemma 1 that $s^* = \sigma_{k+1}(Z)$ must satisfy

$$\sigma_{k+1}(X) \leq \sigma_{k+1}(Z) \leq \sigma_k(Z) = \sigma_k(X). \tag{26}$$

Thus for each $k < k_0$ we only need to consider s in the interval $[\sigma_{k+1}(X),\ \sigma_k(X)]$. Since g_i are increasing we can further divide this interval into subintervals. We let $I_l = [\sqrt{g_{k_l}},\ \sqrt{g_{k_l+1}}]$, where $\sqrt{g_{k_l}}$, $l = 1, ..., m-1$ is the subsequence with terms in the (open) interval $(\sigma_{k+1}(X),\ \sigma_k(X))$. Furthermore, we let $I_0 = [\sigma_{k+1}(X),\ \sqrt{g_{k_1}}]$ and $I_m = [\sqrt{g_{k_m}},\ \sigma_k(X)]$. Note that on each of these subintervals the objective function can be written as a concave quadratic function

$$f_l^k(s) = \sum_{g_i \leq g_{k_l}} g_i + \sum_{g_i > g_{k_l}} s^2 - \sum_{i=k+1}^{n} (s - \sigma_i(X))^2, \quad s \in I_l \tag{27}$$

We can therefore rewrite the inner optimization in (25) as the piecewise smooth problem

$$\max_{0 \leq l \leq m} \max_{s \in I_l} f_l^k(s). \tag{28}$$

The optimum must lie at either a feasible stationary point of f_l^k or at one of the boundaries of I_l for some l. To find the optimal s we can simply enumerate all the possibilities and choose the maximizing one. Since each $\sqrt{g_i}$ only lies in one of the intervals $[\sigma_{k+1}(X), \sigma_k(X)]$ we only need to consider each g_i once. This makes the number of possible solutions depend linearly on the number of singular values.

The steps of the method are summarized in Algorithm 1.

Data: X, g
Result: $\boldsymbol{\sigma}(Z^*)$
for $k = 0 : k_0$ **do**
  Compute s^* and l^* from (28);
  if $f_{l^*}^k(s^*) > f_{opt}$ **then**
    $\sigma_i(Z^*) := \sigma_i(X), \quad \forall i < k;$
    $\sigma_i(Z^*) := s^*, \quad \forall i \geq k;$
    $f_{\text{opt}} := f_{l^*}^k(s^*);$
  end
end

Algorithm 1: Finding maximizing Z for (10)

2.3 The Proximal Operator of f_g^{**}

In order to optimize the convex envelope $f_g^{**}(X)$ efficiently we need to be able to compute its proximal operator

$$\text{prox}_{f_g^{**}}(M) = \underset{X}{\text{argmin}}\, f_g^{**}(X) + \rho \|X - M\|_F^2. \tag{29}$$

The approach we will take is similar to how we evaluate $f_g^{**}(X)$ itself but will require looping over two variables instead of one. The key observation is that switching the order of the minimization over X with maximization over Z enables us to characterize optimal solutions similarly to Section 2.2. [2] We therefore solve

$$\max_Z \min_X \sum_{i=1}^n \min(g_i, \sigma_i^2(Z)) - \|X - Z\|_F^2 + \|X - X_0\|_F^2 + \rho \|X - M\|_F^2. \tag{30}$$

The inner minimization in X is a simple least squares problem. By completing squares one sees that the optimal X is given by

$$X = M + \frac{X_0 - Z}{\rho}. \tag{31}$$

Inserting into (30) we get after some manipulations

$$\max_Z \sum_{i=1}^n \min(g_i, \sigma_i^2(Z)) - \frac{\rho + 1}{\rho} \|Z - Y\|_F^2 + C, \tag{32}$$

where C is a constant that does not depend on Z and

$$Y = \frac{X_0 + \rho M}{1 + \rho}. \tag{33}$$

[2] If $\rho > 0$ the objective function is closed, proper convex-concave, continuous and the optimization can be restricted to a compact set. Switching optimization order is therefore justified by the existence of a saddle point, see [14].

Therefore we see that the singular value $\sigma_k(Z)$ must solve the problem

$$\max_s \min(g_i, s^2) - \frac{\rho+1}{\rho}(s - \sigma_k(Y))^2 \tag{34}$$

$$\text{s.t.} \quad \sigma_{k+1}(Z) \leq s \leq \sigma_{k-1}(Z). \tag{35}$$

The objective function (34) is the pointwise minimum of the two quadratic strictly (assuming $\rho > 0$) concave functions,

$$q_1(s) = g_i - \frac{\rho+1}{\rho}(s - \sigma_k(Y))^2, \quad q_2(s) = s^2 - \frac{\rho+1}{\rho}(s - \sigma_k(Y))^2. \tag{36}$$

The objective function is equal to $q_1(s)$ for $s \geq \sqrt{g_k}$ and $q_2(s)$ otherwise. The functions q_1 and q_2 attain their maximum values at $s = \sigma_k(Y)$ and $s = (\rho + 1)\sigma_k(Y)$ respectively. Note that since $(\rho+1)\sigma_k(Y) > \sigma_k(Y)$ at most one of these can be feasible. It can also happen that neither is feasible, i.e. $\sigma_k(Y) \leq \sqrt{g_k} \leq (\rho+1)\sigma_k(Y)$. In this case the optimal $s = \sqrt{g_k}$. Figure 2 illustrates the shape of the objective function in the three possible cases.

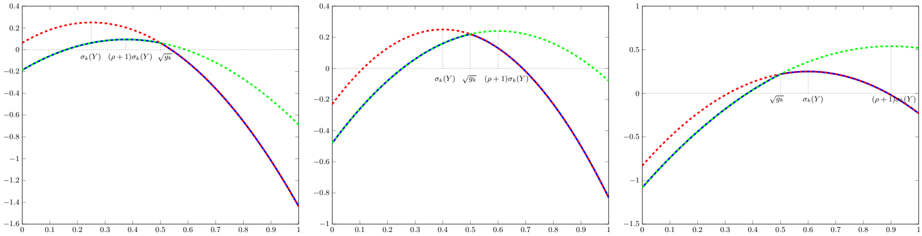

Fig. 2. The objective function in (34) for left: $(\rho+1)\sigma_k(Y) \leq \sqrt{g_k}$, middle: $\sigma_k(Y) \leq \sqrt{g_k}$ and $(\rho+1)\sigma_k(Y) \geq \sqrt{g_k}$ and right: $\sigma_k(Y) \geq \sqrt{g_k}$

Let s_k be the individual unconstrained maximizers of (34), i.e.

$$s_k = \begin{cases} \sigma_k(Y) & \text{if} \quad \sigma_k(Y) \geq \sqrt{g_k} \\ \sqrt{g_k} & \text{if } \sigma_k(Y) \leq \sqrt{g_k} \leq (\rho+1)\sigma_k(Y) \\ (\rho+1)\sigma_k(Y) & \text{if} \quad (1+\rho)\sigma_k(Y) \leq \sqrt{g_k} \end{cases}. \tag{37}$$

Lemma 3. *If Z is optimal in (32) then there is k_1 and k_2 such that*

$$\sigma_i(Z) = s_i, \quad \text{if } i < k_1 \tag{38}$$

$$\sigma_i(Z) = s^*, \quad \text{if } k_1 \leq i \leq k_2 \tag{39}$$

$$\sigma_i(Z) = s_i, \quad \text{if } i > k_2, \tag{40}$$

where s^* solves

$$\max_s \sum_{i=k_1}^{k_2} \min(g_i, s^2) - \frac{\rho+1}{\rho}(s - \sigma_i(Y))^2 \tag{41}$$

$$s.t. \quad \sigma_{k_2+1}(Z) \leq s \leq \sigma_{k_1-1}(Z). \tag{42}$$

Proof. By construction there will exist $p, q \in \mathbb{N}$ with $p \leq q$ such that s_i is; decreasing for $1 \leq i \leq p$, increasing for $p \leq i \leq q$ and decreasing for $q \leq i \leq n$. For $1 \leq i \leq q$ we are in the same situation as in Lemma 1 and Lemma 2 with $k_0 = p$.

Consider now instead $i \geq q$. We will show that

$$\sigma_i(Z) = \min(s_i,\ \sigma_{i-1}(Z)) \quad \text{for} \quad i \geq q. \tag{43}$$

It is clear from (16) that this holds for $i = n$. We continue using induction by assuming $\sigma_{i+1}(Z) = \min(s_{i+1}, \sigma_i(Z))$ holds. Then

$$\sigma_{i+1}(Z) \leq s_{i+1} \leq s_i, \tag{44}$$

since s_i are decreasing for $i \geq q$. This means that for $\sigma_i(Z)$ we can ignore the third case in (16). Thus it follows that $\sigma_i(Z) = \min(s_i, \sigma_{i-1}(Z))$. So (43) holds for all $i \geq q$.

Now assume that for some $i \geq q$ we have $\sigma_i(Z) \neq s_i$. By (43) we must have that

$$\sigma_i(Z) = \sigma_{i-1}(Z) < s_i \leq s_{i-1}. \tag{45}$$

By repeating the argument we get

$$\sigma_i(Z) = \sigma_{i-1}(Z) = \sigma_{i-2}(Z) = ... = \sigma_q(Z), \tag{46}$$

and the result follows. □

Algorithm The properties listed in Lemma 3 allows us to find the optimal Z by searching over the two parameters k_1 and k_2. The goal is to find all sequences $\sigma_i(Z)$ of the type given in the lemma and determine which one gives the best objective value. For fixed k_1 and k_2 the problem in (41) is a piecewise smooth problem similar to (13) which we can solve in the same way by considering the feasible stationary points as well as the boundaries. Note that for feasible solutions we must have $1 \leq k_1 \leq p$ and $q \leq k_2 \leq n$. We outline the steps in Algorithm 2.

3 Block Decomposition with ADMM

Next we consider the problem of missing data. The approach we take here follows [13] and we only give a very brief account of it here for completeness. The idea is to try to enforce low rank of sub-blocks of the matrix where no measurements are missing using our convex relaxation. We seek to minimize the non-convex function

$$f(X) = \sum_{i=1}^{K} g(\text{rank}(\mathcal{P}_i(X)) + \|\mathcal{P}_i(X) - \mathcal{P}_i(M)\|_F^2, \tag{47}$$

```
Data: X_0, ρ, μ, M
Result: Set of possible solutions S
S := ∅ ;
Define p, q as in proof of Lemma 3;
if s_i is decreasing with i then
    S := {s_i};
    return;
else
    for k_1 = 1 : p do
        for k_2 = q : n do
            Compute s* from (41) and form σ(Z) as in Lemma 3;
            if σ_i(Z) is decreasing with i then
                S := S ∪ {σ(Z)};
            end
        end
    end
end
```

Algorithm 2: Finding maximizing Z for the proximal operator (32)

by replacing it with the convex relaxation

$$f_{\mathcal{R}}(X) = \sum_{i=1}^{K} \mathcal{R}_g(\mathcal{P}_i(X)) + \|\mathcal{P}_i(X) - \mathcal{P}_i(M)\|_F^2. \tag{48}$$

Here the operator $\mathcal{P}_i$ extracts elements corresponding to sub-block i. We do not explicitly penalize the rank of X, but instead accomplish this via the rank penalization of the sub-matrices.

To optimize (48) we use ADMM [15]. For each block $\mathcal{P}_i(X)$ we introduce a separate set of variables X_i and enforce consistency via the linear constraints $X_i - \mathcal{P}_i(X) = 0$. We formulate an augmented Lagrangian of (48) as

$$\sum_{i=1}^{K} \left(\mathcal{R}_g(X_i) + \|X_i - \mathcal{P}_i(M)\|_F^2 + \rho\|X_i - \mathcal{P}_i(X) + \Lambda_i\|_F^2 - \rho\|\Lambda_i\|_F^2\right). \tag{49}$$

At each iteration t of ADMM we solve the subproblems

$$X_i^{t+1} = \underset{X_i}{\operatorname{argmin}}\, \mathcal{R}_g(X_i) + \|X_i - \mathcal{P}_i(M)\|_F^2 + \rho\|X_i - \mathcal{P}_i(X^t) + \Lambda_i^t\|_F^2, \tag{50}$$

for $i = 1, ..., K$ and

$$X^{t+1} = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{K} \rho\|X_i^{t+1} - \mathcal{P}_i(X) + \Lambda_i^t\|_F^2. \tag{51}$$

Here Λ_i^t, $i = 1, ..., K$ are the scaled dual variables whose updates at iteration t are given by $\Lambda_i^{t+1} = \Lambda_i^t + X_i^{t+1} - \mathcal{P}_i(X^{t+1})$. The first problem (50) can be solved using the proximal operator derived in the previous section. The second subproblem (51) is a separable least squares problem with closed form solution.

3.1 Extending the Solution

To extend the solution beyond the blocks we employ a nullspace matching scheme which has previously been used in [16] and [17]. The goal is find a rank r factorization of the full solution $X = UV^T$ given the solution on the blocks. Each block $\mathcal{P}_k(X)$ can be factorized as $\mathcal{P}_k(X) = U_k V_k^T$. Then $\mathcal{P}_k(U)$ [3] must lie in the column space of U_k or equivalently it must be orthogonal to the complement, i.e. $(U_k^\perp)^T \mathcal{P}_k(U) = 0$. We can also write this as

$$A_k U = \begin{bmatrix} 0 & (U_k^\perp)^T & 0 \end{bmatrix} U = 0. \tag{52}$$

Collecting these into matrix, $AU = 0$, we can find U by minimizing $||AU||$. Since the scale of U is arbitrary we can consider this as a homogeneous least squares problem which can be solved using SVD. For known U we can simply find V by minimizing $||W \odot (M - UV^T)||$.

4 Stronger Relaxations using a Trust Region Formulation

In case of very large noise levels the regularizer $\mathcal{R}_g$ may not be strong enough to enforce low rank of the solution. In this section we present an approach to strengthen it by restricting the algorithm to a local search close to a current solution estimate X_k. We consider minimization of

$$g(\text{rank}(X)) + \|X - X_0\|_F^2 + \lambda\|X - X_k\|_F^2 \tag{53}$$

The third term can be thought of as a restriction of the step-length of X to a region where our convex relaxation is accurate. By completing squares the expression above can be written

$$(1+\lambda)\left(\frac{1}{1+\lambda} g(\text{rank}(X)) + \left\| X - \frac{X_0 + \lambda X_k}{1+\lambda} \right\|_F^2 + C\right), \tag{54}$$

where C is a constant that depends on λ, X_0 and X_k. Therefore we find that the convex envelope of (53) is

$$(1+\lambda)\mathcal{R}_{\frac{g}{1+\lambda}}(X) + \|X - X_0\|_F^2 + \lambda\|X - X_k\|_F^2. \tag{55}$$

It can be shown that the term $(1+\lambda)\mathcal{R}_{\frac{g}{1+\lambda}}(X) \to g(\text{rank}(X))$ when $\lambda \to \infty$, that is, we have point wise convergence. Figure 3 shows a one-dimensional version of $(1+\lambda)\mathcal{R}_{\frac{g}{1+\lambda}}$ with $g(k) = k$ for varying λ.

Our trust region approach consists of two steps. First we minimize (55) with respect to X. Then we update X_k and repeat the process. Note that at any fix point $X = X_k$ we have a (possibly local) solution to

$$\min_X \; (1+\lambda)\mathcal{R}_{\frac{g}{1+\lambda}}(X) + \|X - X_0\|_F^2. \tag{56}$$

In practice we make the X_k update at each step in the ADMM algorithm instead of running the ADMM until convergence before updating X_k. This greatly increases speed of convergence.

[3] Here $\mathcal{P}_k(U)$ denotes the rows corresponding to block k.

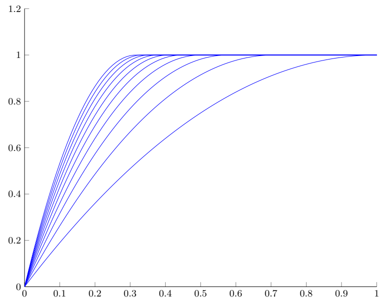

Fig. 3. The regularizer $r(\sigma) = 1 - [1 - \sqrt{1+\lambda}\sigma]_+^2$ for different λ.

5 Implementation and Experiments

In the experiments we focus our attention to the function $g(k) = \mu \max(r_0,\ k)$. This choice allows us to penalizes a rank higher than r_0 while not being biased towards lower rank solutions.

5.1 Comparison to [13]

We first compare the performance of the envelope of [13] and our convex relaxation (CR) in the block decomposition approach (48). We consider the same three image sequences (*book*, *hand* and *banner*) which was used in [13]. Since we are looking for fixed rank solutions we simply choose our weight μ to be sufficiently large. This makes the approach essentially parameter free. In contrast [13] iterates over weights to find a correct rank solution. The difficulty of finding the optimal parameters is heavily depending on the amount of noise in the data. For problems with noisy data and many large blocks (such as the *banner* sequence) this may be computationally infeasible. We also compare to the trust region based iterative method (TR) described in section 4.

Figure 4 displays the singular values of a single block in the solutions for the three image sequences. Note the logarithmic scale. The methods perform very similarly for the *book* and *hand* sequence. This is due to these sequences having low levels of noise and the problem instance being small enough for it to be feasible to iteratively find a good μ. The reconstruction error for the three sequences can be seen in Table 1.

	[13]	CR	TR
book	1.2731	1.2733	1.2678
hand	0.91386	0.9141	0.91508
banner	3950.2	3373.2	3373.2

Table 1. The errors $\|W \odot (X - M)\|_F$ after extending the solution beyond the blocks as described in Section 3.1 (which ensures the correct rank).

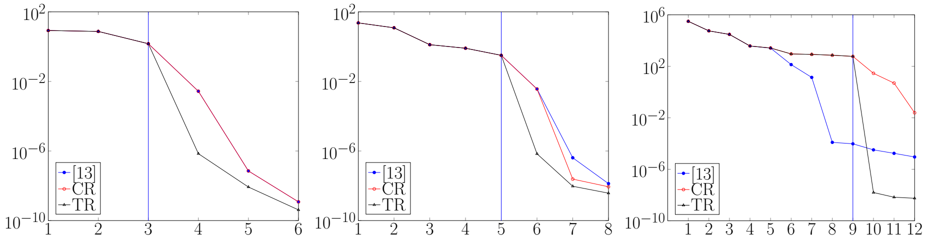

Fig. 4. Singular values for a single block in the *book*, *hand* and *banner* sequence. The vertical blue line indicates the desired rank.

5.2 Comparison to Non-Convex Methods

Next we compare the performance of the proposed method to three state-of-the-art non-convex methods; OptSpace [18], Truncated Nuclear Norm Regularization [19] and Damped Wiberg-L2 [20].

The measurement matrix was chosen as $M = UV^T + N$ where $U, V \in \mathbb{R}^{100\times 5}, N \in \mathbb{R}^{100\times 100}$ and $U_{ij}, V_{ij} \sim \mathcal{N}(0, 1)$ and $N_{ij} \sim \mathcal{N}(0, \sigma)$. If σ is small then M will be approximately rank 5. The observation matrix W consisted of overlapping blocks along the diagonal and had 72% missing data. To the left in Figure 5 we can see the average of $||W \odot (X - M)||_F$ over 100 instances. The performance of the proposed method and Damped Wiberg-L2 is very similar on this data. To illustrate the benefit of the proposed method we also performed an experiment on another family of instances generated by replacing the fifth column of V by $10^3\mathbb{1}$. This essentially makes M have one very dominant singular value which is common in applications. The averaged result for these instances can be seen to the right in Figure 5.

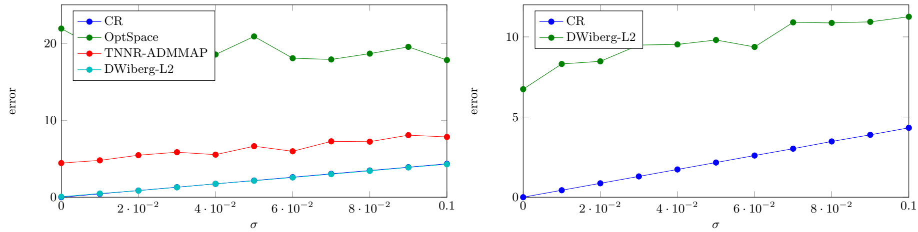

Fig. 5. Comparison with non-convex methods. *Left:* Initial experiment. (Note that the errors for our approach and DWiberg-L2 are very similar). *Right:* Experiment with adjusted row-mean.

References

1. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: IEEE Conference on Computer Vision and Pattern Recognition. (2000)

2. Yan, J., Pollefeys, M.: A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. IEEE Trans. Pattern Anal. Mach. Intell. **30**(5) (2008) 865–877
3. Garg, R., Roussos, A., de Agapito, L.: Dense variational reconstruction of non-rigid surfaces from monocular video. In: IEEE Conference on Computer Vision and Pattern Recognition. (2013)
4. Basri, R., Jacobs, D., Kemelmacher, I.: Photometric stereo with general, unknown lighting. Int. J. Comput. Vision **72**(3) (May 2007) 239–257
5. Garg, R., Roussos, A., Agapito, L.: A variational approach to video registration with subspace constraints. Int. J. Comput. Vision **104**(3) (2013) 286–314
6. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. Journal on Computer Vision **9**(2) (1992) 137–154
7. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika **1**(3) (1936) 211–218
8. Eriksson, A., Hengel, A.: Efficient computation of robust weighted low-rank matrix approximations using the L_1 norm. IEEE Trans. Pattern Anal. Mach. Intell. **34**(9) (2012) 1681–1690
9. Strelow, D.: General and nested Wiberg minimization. In: IEEE Conference on Computer Vision and Pattern Recognition. (2012)
10. Wiberg, T.: Computation of principal components when data are missing. Proc. Symposium of Computational Statistics (1976) 229–326
11. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. on Optimization **20**(4) (2010) 1956–1982
12. Angst, R., Zach, C., Pollefeys, M.: The generalized trace-norm and its application to structure-from-motion problems. In: International Conference on Computer Vision. (2011)
13. Larsson, V., Olsson, C., Bylow, E., Kahl, F.: Rank minimization with structured data patterns. In: European Conference on Computer Vision ECCV. (2014)
14. Rockafellar, R.T.: Convex analysis. Princeton Mathematical Series. Princeton University Press, Princeton, N. J. (1970)
15. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1) (January 2011) 1–122
16. Olsen, S., Bartoli, A.: Implicit non-rigid structure-from-motion with priors. Journal of Mathematical Imaging and Vision **31**(2-3) (2008) 233–244
17. Jacobs, D.: Linear fitting with missing data: applications to structure-from-motion and to characterizing intensity images. In: IEEE Conference on Computer Vision and Pattern Recognition,. (1997)
18. Keshavan, R.H., Montanari, A., Oh, S.: Matrix completion from a few entries. IEEE Trans. Inf. Theory **56**(6) (2010) 2980–2998
19. Hu, Y., Zhang, D., Ye, J., Li, X., He, X.: Fast and accurate matrix completion via truncated nuclear norm regularization. IEEE Trans. Pattern Anal. Mach. Intell. **35**(9) (2013) 2117–2130
20. Okatani, T., Yoshida, T., Deguchi, K.: Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. In: Proceedings of the International Conference on Computer Vision. (2011)