Nuclear Norms for Tensors and Their Use for Convex Multilinear Estimation

Marco Signoretto[†] Lieven De Lathauwer[‡] Johan A. K. Suykens[†]

[†] *Katholieke Universiteit Leuven, ESAT-SCD/SISTA*
Kasteelpark Arenberg 10, B-3001 Leuven (BELGIUM)
{*marco.signoretto,johan.suykens*}*@esat.kuleuven.be*

[‡] *Group Science, Engineering and Technology*
Katholieke Universiteit Leuven, Campus Kortrijk
E. Sabbelaan 53, 8500 Kortrijk (BELGIUM)
{*lieven.delathauwer*}*@kuleuven-kortrijk.be*

Abstract

A promising convex relaxation for rank-constrained matrix problems relies on the use of the nuclear norm, namely the sum of the singular values of the optimization matrix. In this paper we introduce a proper extension of the concept of Shatten-q norms for matrices — including the nuclear norm — to higher order tensors. Successively we study a general class of non-smooth convex optimization problems where the tensorial nuclear norm is employed as a penalty function. We further explore the consequence for the optimization of the link between multilinear ranks and the new norm. From a methodological perspective, the considered general formulation can be specialized to accomplish different (and possibly supervised) data-driven modeling tasks. Existing preliminary works on the tensor completion problem can be regarded as a specific instance of the more general class of problems considered here. From an algorithmical perspective the main algorithm that we develop, termed Convex MultiLinear Estimation (CMLE), leads to exploiting results from distributed convex optimization. Variants of this computational scheme are additionally proposed to ensure fast convergence and to deal with certain equality constrained problems.

Keywords: nuclear norm, separable convex problem, accelerated first order methods, multilinear ranks, supervised and unsupervised learning

1. Introduction

$N-$order tensors are higher-order equivalents of vectors (1-order tensors or $1-$way arrays) and matrices (2-order tensors or $2-$way arrays). They find applications whenever the data of interest have intrinsically many dimensions [1]. This is the case for an increasing number of areas such as econometrics, chemometrics, psychometrics, (biomedical) signal processing and image processing. Regardless of the specific domain, a common task in the data analysis workflow amounts at finding some low dimensional representation of the process under study. Mathematically this task can be formalized as the problem of finding the rank-$(r_1, r_2, \ldots, r_N)$ tensor that best explains the data. For certain rank-constrained problems involving matrices, an exact solution can be found by the singular value decomposition. Sometimes these problems can be exactly reduced to the solution of linear systems [2]. In general, however, problems involving rank constraints are challenging non-convex optimization problems that suffer from local minima. Moreover, rank-related problems in multilinear algebra are even more complicated than their matrix counterparts. Best low rank approximation in least-squares sense for higher-order tensors were studied in [3],[4],[5],[6],[7],[8],[9],[10]. For matrix problems with rank constraints a variety of heuristics have been proposed, including alternating projections [11] and alternating LMIs [12]. A promising convex relaxation relies on the use of the nuclear norm, namely the sum of the singular values of the optimization matrix [13]. To a large extent the nuclear norm in convex heuristics for rank constrained matrix problems parallels the use of the l_1-norm in sparse approximation and cardinality minimization [14],[15],[16]. Besides yielding global optima it can be optimized efficiently following one of the multiple strategies already proposed in the literature [13],[17],[18].

This paper begins by extending the concept of Shatten-q norms from matrices to higher order tensors. Within this class we focus on a proper definition of nuclear-1 norm that relates to the concept of nuclear norm for tensors given recently in [19]. The latter reference deals with the so-called *tensor completion problem* while a first contribution of this manuscript is realized at the methodological level by considering a more general class of problems. The latter can be specialized to accomplish different tasks — including supervised learning with convex loss functions— by suitably defining a data driven contribution f in the objective function. We further study the relation between (tensorial) nuclear norm and multilinear ranks and, succes-

sively, explore the link between a general non-convex optimization problem, featuring a penalty on the multilinear ranks, and its convex relaxation based on the new norm. At the algorithmic level we show how the approach leads to exploiting results from distributed convex optimization [20]. We propose a scalable algorithm that can be adapted to solve a large class of data-driven tasks. In fact, the solution strategy is essentially a first order method and hence the only part that is problem-dependent is the gradient of the data driven function ∇f. We also elaborate on variants of the main procedure that are designed to deal with equality constrained problems and to improve the speed of convergence. We conclude by illustrating special cases that arise from the generalization of existing matrix problems and give numerical examples.

1.1. Outline of the Paper

In the remaining of this Section we introduce the notation that we use throughout the paper. In Section 2 we recall the basic mathematical operations on tensors. In Section 3 we review the concept of *multilinear singular value decomposition* (MLSVD) to which our definition of nuclear norm for tensors is related. In Section 4 we recall the definition of Shatten-q norms for matrices and introduce a proper generalization for tensors. In particular, within this class we define nuclear norms and focus on a particular instance. Section 5 is devoted to the use of the latter as penalty functions for a general class of optimization problems. Section 6 presents algorithmical approaches to cope with these problems. Successively in Section 7 we discuss special cases of interest and then present numerical experiments in Section 8. We finally draw our conclusions in Section 9.

1.2. Notation

We denote scalars by lower-case letters $(a, b, c, \ldots)$, vectors as capitals $(A, B, C, \ldots)$ and matrices as bold-face capitals $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \ldots)$. Tensors are written as calligraphic letters $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots)$. We write a_i to mean the $i-$th entry of a vector A. Similarly we write a_{ij} to mean the entry with row index i and column index j in a matrix $\boldsymbol{A}$, i.e., $a_{ij} = (\boldsymbol{A})_{ij}$ and $a_{i_1,\ldots,i_N}$ to denote $(\mathcal{A})_{i_1,\ldots,i_N}$. Furthermore, the $i-$th column vector of a matrix $\boldsymbol{A}$ is denoted as A_i , i.e., $\boldsymbol{A} = [A_1 A_2 \cdots]$. We frequently use i, j in the meaning of indices and with some abuse of notation we will use I, J to denote the index upper bounds. We further denote sets by Gothic letters $(\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \ldots)$. Notice that for ease of notation we do not distinguish between different type of sets:

depending on the context $\mathfrak{A}$ might equally well denote a subset of $\mathbb{R}^p$ or a subset of $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Finally we often write $\mathbb{N}_I$ to denote the set $\{1, \ldots, I\}$ and use $\mathcal{I}$ to denote the *indicator* function defined as

$$\mathcal{I}(a) := \begin{cases} 1, & a \geq 0 \\ 0, & a < 0 \end{cases} .$$

2. Basic Definitions

In this paper we deal with real-valued $N-$order tensors, namely elements of some finite dimensional vector space $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. This space is endowed with the inner product

$$\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} a_{i_1 i_2 \cdots i_N} b_{i_1 i_2 \cdots i_N} \tag{1}$$

and with the Hilbert-Frobenius norm [4][1]

$$\|\mathcal{A}\|_F := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} . \tag{2}$$

An $N-$order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ has rank-1 if it consists of the outer product of N vectors $U^{(1)} \in \mathbb{R}^{I_1}$, $U^{(2)} \in \mathbb{R}^{I_2}, \ldots,$ $U^{(N)} \in \mathbb{R}^{I_N}$ that is if

$$a_{i_1 i_2 \ldots i_N} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(N)}_{i_N}$$

for all values of the indices. A $n-$*mode vector* of $\mathcal{A}$ is an element of $\mathbb{R}^{I_n}$ obtained from $\mathcal{A}$ by varying the index i_n and keeping the other indices fixed. The $n-$*rank* of $\mathcal{A}$, indicated by $\text{rank}_n(\mathcal{A})$, is the straightforward generalization of the column (row) rank for matrices: it equals the dimension of the vector space spanned by the $n-$mode vectors. Whereas $\text{rank}_1(\mathcal{A}) = \text{rank}_2(\mathcal{A})$ for a $2-$order tensor $\mathcal{A}$ (in fact a matrix $\boldsymbol{A}$) the $n-$ranks differ from each other in the general $N-$order case. A tensor of which the $n-$ranks are equal to r_n for $n \in \mathbb{N}_N$ is called a rank-$(r_1, r_2, \ldots, r_N)$ tensor. Even when all the $n-$ranks are the same, they can still be different from the rank of the tensor, denoted as $\text{rank}(\mathcal{A})$. The latter is defined as the minimal number r of rank-1

[1] We stress that the given definitions of inner product and Hilbert-Frobenius norm cover the case where tensors are actually matrices.

tensors $\mathcal{A}^{(i)},\ i \in \mathbb{N}_r$ such that

$$\mathcal{A} = \sum_{i \in \mathbb{N}_r} \mathcal{A}^{(i)} \ .$$

Whereas for $2-$way tensors $\text{rank}_1(\mathcal{A}) = \text{rank}_2(\mathcal{A}) = \text{rank}(\mathcal{A})$ for the general case we can only establish that $\text{rank}_n(\mathcal{A}) \leq \text{rank}(\mathcal{A})$ for any $n \in \mathbb{N}_N$.

Definition 1 (n-mode product [21]). The $n-$mode product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by a matrix $\boldsymbol{U} \in \mathbb{R}^{J_n \times I_n}$, denoted by $\mathcal{A} \times_n \boldsymbol{U}$, is a $(I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N)-$tensor with entries

$$(\mathcal{A} \times_n \boldsymbol{U})_{i_1 i_2 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} := \sum_{i_n \in \mathbb{N}_{I_n}} a_{i_1 i_2 \cdots i_{n-1} i_n i_{n+1} \cdots i_N} u_{j_n i_n} \ .$$

Let $\boldsymbol{F} \in \mathbb{R}^{I_1 \times I_2}, \boldsymbol{U} \in \mathbb{R}^{J_1 \times I_1}$ and $\boldsymbol{V} \in \mathbb{R}^{J_2 \times I_2}$ and further let $\boldsymbol{V}^\top$ denote the transpose of the real valued matrix $\boldsymbol{V}$. Then according to the former definition the matrix product $\boldsymbol{U}\boldsymbol{F}\boldsymbol{V}^\top$ can be equivalently restated as $\boldsymbol{F} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}$. The following properties hold:

$$\begin{aligned} (\mathcal{A} \times_n \boldsymbol{U}) \times_m \boldsymbol{V} &= \quad (\mathcal{A} \times_m \boldsymbol{V}) \times_n \boldsymbol{U} = \quad \mathcal{A} \times_n \boldsymbol{U} \times_m \boldsymbol{V} \\ (\mathcal{A} \times_n \boldsymbol{U}) \times_n \boldsymbol{V} &= \quad \mathcal{A} \times_n (\boldsymbol{V}\boldsymbol{U}) \ . \end{aligned}$$

It is often convenient to rearrange the elements of a tensor so that they form a matrix. This operation is referred to as *matricization* or *unfolding* and can be defined in different ways. Unfolding is generally performed by simply stacking columns (resp. rows, ...) vectors one after another following a prescribed ordering. Different ordering rules are adopted in the literature on tensors. The convention followed in [22] is referred to in [23] as *backward cyclic* whereas the one used in [24] is called *forward cyclic*. In presenting the definition of $n-$mode matricization we comply with forward cyclic ordering.

Definition 2 ($n-$mode matricization [24]). Assume a $N-$th order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The $n-$th mode matrix unfolding is defined as the matrix

$$\mathbb{R}^{I_n \times J} \ni \boldsymbol{A}_{(n)} \ : \ a^{(n)}_{i_n j} := a_{i_1 i_2 \ldots i_N} \tag{3}$$

where $J := I_{n+1} I_{n+2} \cdots I_N I_1 I_2 I_3 \cdots I_{n-1}$ and for $R := [n+1\, n+2 \cdots$

$N\,1\,2\,3\,\cdots\,n-1]$ we have:

$$j = 1 + \sum_{l \in \mathbb{N}_{N-1}} \left[(i_{r_l} - 1) \prod_{\hat{l} \in \mathbb{N}_{l-1}} I_{r_{\hat{l}}} \right] . \tag{4}$$

The reader is referred to [23] for a visualization of both the backward and forward cyclic unfolding procedures for 3-order arrays. We stress that properties of tensors are invariant under different ordering rules, provided that the same ordering is preserved in going from tensors to matrices and vice-versa. Concerning the inverse operation of $n-$mode matricization we have the following definition.

Definition 3 ($n-$mode $(I_1, \ldots, I_N)$ tensorization). Assume a matrix $\boldsymbol{A} \in \mathbb{R}^{I_n \times J}$ with $J = I_{n+1}\, I_{n+2} \cdots I_N\, I_1\, I_2\, I_3 \cdots I_{n-1}$. The $n-$th mode $(I_1, \ldots, I_N)$ tensor folding $\mathcal{B}$ is defined as the tensor

$$\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N} \ni \mathcal{B} \ : \ b_{i_1 i_2 \ldots i_N} := a_{i_n j} \tag{5}$$

where j is defined as in (4).

Notice that $n-$mode matricization and tensorization both define linear operators that we shall introduce explicitly for convenience of notation. Specifically, based upon Definition 2 we can introduce the *matricization operator*, denoted by $\cdot_{<n>}$, as

$$\begin{array}{ccccc} \cdot_{<n>} & : & \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N} & \rightarrow & \mathbb{R}^{I_n \times (I_{n+1}\, I_{n+2} \cdots I_N\, I_1\, I_2\, I_3 \cdots I_{n-1})} \\ & & \mathcal{A} & \mapsto & \boldsymbol{A}_{(n)} \ . \end{array} \tag{6}$$

Similarly we can define the $n-$mode *tensorization operator*[2] $\cdot^{<n>}$ based upon Definition 3. It is not difficult to see that

$$\langle \boldsymbol{B}^{<n>}, \mathcal{A} \rangle = \langle \boldsymbol{B}, \mathcal{A}_{<n>} \rangle$$

and hence $\cdot^{<n>}$ can be equivalently defined as the adjoint operator of $\cdot_{<n>}$.

Via matricization, tensors can be manipulated via their matrix representations, reducing the $n-$mode multiplication, for example, to a matrix-matrix

[2]We omit the dimensions $(I_1, \ldots, I_N)$ in the symbol indicating the tensorization operator assuming they will be always clear from the context.

operation. A further advantage is that the $n-$rank of a given tensor can be analyzed by means of matrix techniques [22]:

Remark 1 ($n-$rank and $n-$mode matrix unfolding). Assume $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Then for any $n \in \mathbb{N}_N$, the $n-$mode vectors of $\mathcal{A}$ are the column vectors of its $n-$mode matrix unfolding $\mathcal{A}_{<n>}$ and

$$\text{rank}_n(\mathcal{A}) = \text{rank}(\mathcal{A}_{<n>}) \ .$$

The following definition will be used later on for the numerical computation of $n-$ranks.

Definition 4 ($\epsilon-$precision $n-$rank). Assume $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. For any $n \in \mathbb{N}_N$ and for $J = I_{n+1} I_{n+2} \cdots I_N I_1 I_2 I_3 \cdots I_{n-1}$ let

$$\sigma_1(\mathcal{A}_{<n>}) \geq \sigma_2(\mathcal{A}_{<n>}) \geq \cdots \geq \sigma_{\min(I_n, J)}(\mathcal{A}_{<n>}) \geq 0$$

be the singular values of $\mathcal{A}_{<n>}$. Then the $\epsilon-$precision $n-$rank of $\mathcal{A}$, denoted as $\text{rank}_{n,\epsilon}(\mathcal{A})$ or simply $n-\text{rank}_\epsilon$, is defined as

$$\text{rank}_{n,\epsilon}(\mathcal{A}) := \sum_{i \in \mathbb{N}_{\min(I_n, J)}} \mathcal{I}\left(\sigma_i(\mathcal{A}_{<n>}) - \epsilon\right) \ .$$

3. The Multilinear Singular Value Decomposition

In [25],[26],[21] the following generalization of the singular value decomposition (SVD) for matrices was developed. It was termed in [21] higher-order singular value decomposition in view of the many analogies with the matrix SVD. Nowadays the name multilinear singular value decomposition (MLSVD) seems to be more appropriate. It is not the only possible generalization though: see [21, Section 7] for alternative definitions.

Theorem 1 (MLSVD [21] [3]). *Any $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$ can be written as the product*

$$\mathcal{A} = \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \cdots \times_N \boldsymbol{U}^{(N)}$$

in which:

[3] In the present setting we state the theorem for the real valued case whereas the MLSVD theorem in [21] refers to the more general complex valued case.

- $\mathbb{R}^{I_n \times I_n} \ni \boldsymbol{U}^{(n)} = \left[U_1^{(n)} \, U_2^{(n)} \, \cdots \, U_{I_n}^{(n)} \right]$ *is a orthogonal matrix*
- *the* core tensor $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$ *is such that the* $(N-1)-$*order subtensor* $\mathcal{S}^{i_n=\alpha}$ *defined by*

$$s^{i_n=\alpha}_{i_1 i_2 \cdots i_{n-1} i_{n+1} \cdots i_N} := s_{i_1 i_2 \cdots i_{n-1} \, \alpha \, i_{n+1} \cdots i_N}$$

satisfies:

1. all-orthogonality*:*

$$\langle \mathcal{S}^{i_n=\alpha}, \mathcal{S}^{i_n=\beta} \rangle = 0, \ \forall \alpha \neq \beta, \alpha, \beta \in \mathbb{N}_n, \ n \in \mathbb{N}_N$$

2. ordering*:*

$$\|\mathcal{S}^{i_n=1}\|_F \geq \|\mathcal{S}^{i_n=2}\|_F \geq \cdots \geq \|\mathcal{S}^{i_n=I_N}\|_F \geq 0 \text{ for any } n \in \mathbb{N}_N.$$

The Hilbert-Frobenius norms $\|\mathcal{S}^{i_n=i}\|_F$*, symbolized by* $\sigma_i^{(n)}$*, are the* $n-$*mode singular values of* $\mathcal{A}$ *and the vector* $U_i^{(n)}$ *is the* $i-$*th* $n-$*mode singular vector.*

In practice the matrix $\boldsymbol{U}^{(n)}$ of $n-$mode singular vectors can directly be found from the left singular vectors of the $n-$mode unfolding $\mathcal{A}_{\langle n \rangle}$ with SVD decomposition

$$\mathcal{A}_{\langle n \rangle} = \boldsymbol{U}^{(n)} \boldsymbol{S}^{(n)} {\boldsymbol{V}^{(n)}}^{\top} . \tag{7}$$

The core tensor $\mathcal{S}$ then satisfies:

$$\mathcal{S} = \mathcal{A} \times_1 {\boldsymbol{U}^{(1)}}^{\top} \times_2 {\boldsymbol{U}^{(2)}}^{\top} \cdots \times_N {\boldsymbol{U}^{(N)}}^{\top} .$$

It is well known for matrices that the best rank-R approximation in least squares sense can be readily obtained from the truncated SVD decomposition. Although useful, the MLSVD is not as powerful: for general tensors it does not deliver best low rank approximation, no matter what concept of rank is considered.

4. A Class of Norms for Higher Order Tensors

We are now ready to formulate our generalization of nuclear norms for higher order tensors. Before, we recall some definitions and basic results about matrices.

4.1. Shatten-q Norms for Matrices

Definition 5 (Shatten-q Norms on $\mathbb{R}^{I_1 \times I_2}$). Assume a matrix $\boldsymbol{A} \in \mathbb{R}^{I_1 \times I_2}$ with singular values

$$\sigma_1(\boldsymbol{A}) \geq \sigma_2(\boldsymbol{A}) \geq \cdots \geq \sigma_r(\boldsymbol{A}) > \sigma_{r+1}(\boldsymbol{A}) = \ldots = \sigma_{\min(I_1,I_2)}(\boldsymbol{A}) = 0 \ .$$

The *Shatten-q* norms, $q \geq 1$ are defined by

$$\|\boldsymbol{A}\|_{\Sigma,q} = \left(\sum_{i \in \mathbb{N}_r} \sigma_i(\boldsymbol{A})^q \right)^{\frac{1}{q}} \ .$$

In particular, the norm $\|\boldsymbol{A}\|_{\Sigma,1}$ is called *nuclear* or *trace* norm.

Other special cases of Shatten-q norms are the Hilbert-Frobenius norm, $\|\boldsymbol{A}\|_F = \|\boldsymbol{A}\|_{\Sigma,2}$, and the *operator* (sometimes called *spectral*) norm $\|\boldsymbol{A}\|_2 := \sigma_1(\boldsymbol{A})$ that corresponds to $\|\boldsymbol{A}\|_{\Sigma,\infty}$.

The use of the nuclear norm to devise a convex heuristic of (non-convex) matrix optimization problems involving the rank function was firstly proposed in [27]. Recently, detailed conditions for exact recovery of low-rank matrices via nuclear norm optimization have been established, see [13],[28],[29] and reference therein. Ultimately the use of nuclear norm among all the possible convex heuristics can be motivated by a simple fact. Recall that the *convex envelope* [30],[31] of a function f is defined as the pointwise largest convex function g such that $g(x) \leq f(x)$ for all $x \in \mathrm{dom}(f)$. Geometrically, the definition implies that the epigraph of g is the convex hull of the epigraph of f.

Theorem 2 (Nuclear Norm as Convex Envelope [32])**.** *On the set* $\mathfrak{S} = \left\{ \boldsymbol{X} \in \mathbb{R}^{I_1 \times I_2} \ : \ \|\boldsymbol{X}\|_2 \leq 1 \right\}$*, the convex envelope of the function* $f(\boldsymbol{X}) = \mathrm{rank}(\boldsymbol{X})$ *is* $\|\boldsymbol{X}\|_{\Sigma,1}$*.*

Analogously on $\mathfrak{S}_M = \left\{ \boldsymbol{X} \in \mathbb{R}^{I_1 \times I_2} \ : \ \|\boldsymbol{X}\|_2 \leq M \right\}$ the convex envelope of the rank function is $\|\boldsymbol{X}\|_{\Sigma,1}/M$ [32]. In a nutshell the result means that the nuclear norm is the tightest convex lower approximation to the rank function. Hence it yields the tightest global lower bound on rank among all the possible convex approximations.

4.2. Shatten-$\{p, q\}$ Norms for Tensors

In (2) we already defined the Frobenius norm for tensors. In general it turns out that we can norm $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ solely based upon any $n-$mode matricization and some existing norm for matrices.

Lemma 1. *Assume a $N-$th order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Then for any norm $\|\cdot\|$ and any $n \in \mathbb{N}_N$, the application:*

$$\mathcal{A} \mapsto \|\mathcal{A}_{<n>}\|$$

defines a norm for the vector space $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.

Proof. The proof requires to check that $\|\mathcal{A}_{<n>}\|$ satisfies the axioms of a norm for the vector space $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ of which $\mathcal{A}$ is an element. It is immediate to see that $\|\mathcal{A}_{<n>}\| \geq 0$ for any $\mathcal{A}$, $\|\mathcal{A}_{<n>}\| = 0$ if and only if $\mathcal{A} = 0$ and also that $\|c\mathcal{A}_{<n>}\| = |c|\|\mathcal{A}_{<n>}\|$. Finally $\|(\mathcal{A}+\mathcal{B})_{<n>}\| \leq \|\mathcal{A}_{<n>}\| + \|\mathcal{B}_{<n>}\|$ since $\cdot_{<n>}$ is linear and $\|\cdot\|$ satisfies the triangular inequality. □

Before continuing we recall here the definition of monotonic norm [33]. Below, for $x \in \mathbb{R}^m$ we let $|x| := [|x_1|\,|x_2|\,\cdots\,|x_m|]^\top$ and write $|x| \leq |y|$ to mean $|x_i| \leq |y_i|$ for any $i \in \mathbb{N}_m$.

Definition 6. A norm of $\mathbb{R}^m$ is said to be *monotonic* if $|x| \leq |y|$ implies $\|x\| \leq \|y\|$.

It can be shown that all the norms of $\mathbb{R}^m$ for which $\|x\| = \||x|\|$ for any $x \in \mathbb{R}^m$ are monotonic [33]. In particular, this property is satisfied by all the l_p norms by definition. Monotonicity is a useful property to construct composite norms from some existing ones, see e.g. [33, Theorem 5.3.1]. The following result provides a general way to construct a norm for the vector space $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ starting from m arbitrary unfoldings together with a monotonic norm on $\mathbb{R}^m$.

Theorem 3 (Composite Norms on $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$)**.** *For any finite integer $m > 0$ let $\{c_1, \ldots, c_m\} \subset \mathbb{N}_N \times \mathbb{N}_N \times \cdots \times \mathbb{N}_N$ and for $j \in \mathbb{N}_m$ let $\|\cdot\|_{c_j} : \mathcal{A}_{<c_j>} \mapsto \mathbb{R}$ be a norm. Then for any monotonic norm $\|\cdot\|_m$ of $\mathbb{R}^m$, the function*

$$\mathcal{A} \mapsto \left\| \left[\|\mathcal{A}_{<c_1>}\|_{c_1} \; \|\mathcal{A}_{<c_2>}\|_{c_2} \; \cdots \; \|\mathcal{A}_{<c_m>}\|_{c_m} \right] \right\|_m \tag{8}$$

defines a norm on $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.

Proof. In order to show that (8) defines a norm for $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ we have to check again that all the axioms for norms are satisfied. Lemma 1 implies that $\|\cdot\|_{c_j}$ properly defines a norm for $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ for any $j \in \mathbb{N}_m$. It is now easy to check that (8) satisfies the first axioms. In order to show that further the triangular inequality is satisfied, let $a, b, c, d, e \in \mathbb{R}^m$ be defined entry-wise as

$$a_j := \|\mathcal{A}_{<c_j>}\|_{c_j},\ b_j := \|\mathcal{B}_{<c_j>}\|_{c_j},\ d_j := \|(\mathcal{A}+\mathcal{B})_{<c_j>}\|_{c_j} \text{ and } e_j := a_j + b_j$$

for any $j \in \mathbb{N}_m$. By the triangular inequality of $\|\cdot\|_{c_j}$ we have $d_j \leq e_j$. Now we have:

$$\|d\|_m \leq \|e\|_m \leq \|a\|_m + \|b\|_m$$

where the first inequality follows from the monotonicity of $\|\cdot\|_m$ and the second from the triangular inequality of $\|\cdot\|_m$. This shows that the application (8) satisfies the triangular inequality and hence terminates the proof. □

Whereas the former constructive result defines a general class of norms, the next definition introduces the specific instances we are interested in.

Definition 7 (Shatten-$\{p, q\}$ Norms on $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$). Assume a $N-$th order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. For $p, q \geq 1$ we define the Shatten-$\{p, q\}$ norms as

$$\|\cdot\|_{p,q} : \mathcal{A} \mapsto \left(\frac{1}{N} \sum_{n \in \mathbb{N}_N} \|\mathcal{A}_{<n>}\|_{\Sigma,q}^p \right)^{\frac{1}{p}} . \tag{9}$$

In particular, we call the norm $\|\mathcal{A}\|_{p,1}$ *nuclear-p* norm.

To our knowledge the class of norms for tensors introduced above has not been defined elsewhere. The only exception is what we call nuclear-1 norm here, which was proposed for the first time in [19] and called simply nuclear norm. Notice that the factor $\frac{1}{N}$ in (9) is merely instrumental to ensure that the definition for high order tensors is consistent with that for matrices. Indeed assume $\mathcal{A}$ is actually a 2-order tensor, namely a matrix $\boldsymbol{A} \in \mathbb{R}^{I_1 \times I_2}$. Since

$$\sigma_i(\mathcal{A}_{<1>}) = \sigma_i(\mathcal{A}_{<2>}) = \sigma_i(\boldsymbol{A}) \quad \forall\ i \in \mathbb{N}_{\min(I_1, I_2)}\ ,$$

we have by Definition 5:

$$\|\mathcal{A}\|_{p,q} = \left(\frac{1}{2}\|\mathcal{A}_{<1>}\|_{\Sigma,q}^p + \frac{1}{2}\|\mathcal{A}_{<2>}\|_{\Sigma,q}^p \right)^{\frac{1}{p}} = \|\boldsymbol{A}\|_{\Sigma,q}$$

which substantiates the attributed name of Shatten-$\{p, q\}$ norm. We finally remark that the definition allows for an ulterior degree of freedom w.r.t the Shatten-q norm for matrices: the parameter p specifies what l_p norm is used to combine the different modes. In the following we focus on the case $p = 1$ and, more precisely, on the nuclear-1 norm. The following result partially clarifies the relation between $\|\cdot\|_{1,1}$ and multilinear ranks.

Theorem 4 (Nuclear norm and multilinear ranks)**.** *Let* $c : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \to \mathbb{R}$ *be the convex envelope of the function* $\frac{1}{N}\sum_{n\in\mathbb{N}_N} \operatorname{rank}_n(\mathcal{X})$ *on* $\mathfrak{S}' = \left\{\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \ : \ \|\mathcal{X}_{<n>}\|_2 \leq 1 \ \forall\, n \in \mathbb{N}_N\right\}$*. Then we have:*

$$\|\mathcal{X}\|_{1,1} \leq c(\mathcal{X}) \ . \tag{10}$$

Before turning to a proof we shall point out that the result we have just stated is weaker than its counterpart Theorem 2. It merely shows that $\|\cdot\|_{1,1}$ underestimates the convex envelope of the sum of multilinear ranks over $\mathfrak{S}'$ (and of course the function $\frac{1}{N}\sum_{n\in\mathbb{N}_N} \operatorname{rank}_n(\mathcal{X})$ itself). As such, an approach based on $\|\cdot\|_{1,1}$ might not represent the best convex heuristic for minimizing the sum of the multilinear ranks. For 2-order tensors where $\|\cdot\|_{1,1}$ boils down to the matrix nuclear norm, however, Theorem 2 shows that the result is tight, i.e., the equality holds true in (10).

Proof of Theorem 4. Define for any $n \in \mathbb{N}_N$ the set

$$\mathfrak{F}_{(n)} := \left\{\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times (I_{n+1}I_{n+2}\cdots I_N I_2 I_3 \cdots I_{n-1})} \ : \ \|\boldsymbol{X}_{(n)}\|_2 \leq 1\right\} \ .$$

It follows by Theorem 2 and by a known property of convex envelopes [34],[31], that the function $\frac{1}{N}\sum_{n\in\mathbb{N}_N} \|\boldsymbol{X}_{(n)}\|_{\Sigma,1}$ is the convex envelope of the function $\frac{1}{N}\sum_{n\in\mathbb{N}_N} \operatorname{rank}(\boldsymbol{X}_{(n)})$ over $\mathfrak{F} = \mathfrak{F}_{(1)} \times \mathfrak{F}_{(2)} \times \cdots \times \mathfrak{F}_{(N)}$. Now let $c' : \mathfrak{F} \to \mathbb{R}$ denote the convex envelope of the function $\frac{1}{N}\sum_{n\in\mathbb{N}_N} \operatorname{rank}(\boldsymbol{X}_{(n)})$ over the set

$$\mathfrak{F}' = \left\{(\boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(N)}) \in \mathfrak{F} : \exists\ \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \text{ s.t. } \boldsymbol{X}_{(n)} = \mathcal{X}_{<n>}\ \forall n \in \mathbb{N}_N\right\}.$$

Since $\mathfrak{F}' \subset \mathfrak{F}$, by a known result on convex envelopes (see e.g. [35][Proposition 3]) we have

$$\frac{1}{N}\sum_{n\in\mathbb{N}_N} \|\boldsymbol{X}_{(n)}\|_{\Sigma,1} \leq c'(\boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(N)})\ \forall\ (\boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(N)}) \in \mathfrak{F}'.$$

The theorem now follows from the fact that there exists an isomorphism b that maps $\mathfrak{S}'$ into $\mathfrak{F}'$ and we have $c(\mathcal{X}) = c'(b(\mathcal{X}))$ for any $\mathcal{X} \in \mathfrak{S}'$. $\square$

Table 1: A schematic illustration of the various norms. It remains an open problem to show whether the nuclear-1 norm is the tightest convex lower approximation to $\frac{1}{N}\sum_{n\in\mathbb{N}_N} \mathrm{rank}_n(\mathcal{X})$ or not.

Matrices		Higher Order Tensors
Shatten$-q$ norms		Shatten$-\{p,q\}$ norms
$\|\boldsymbol{A}\|_{\Sigma,q} := \left(\sum_{i\in\mathbb{N}_r} \sigma_i(\boldsymbol{A})^q\right)^{\frac{1}{q}}$	$\rightarrow$	$\|\mathcal{A}\|_{p,q} := \left(\frac{1}{N}\sum_{n\in\mathbb{N}_N} \|\mathcal{A}_{<n>}\|_{\Sigma,q}^p\right)^{\frac{1}{p}}$
$\downarrow$		$\downarrow$
nuclear norm		nuclear-1 norm
$\|\boldsymbol{A}\|_{\Sigma,1} := \sum_{i\in\mathbb{N}_r} \sigma_i(\boldsymbol{A})$	$\rightarrow$	$\|\mathcal{A}\|_{1,1} := \frac{1}{N}\sum_{n\in\mathbb{N}_N} \|\mathcal{A}_{<n>}\|_{\Sigma,1}$
$\boxed{\|\boldsymbol{A}\|_{\Sigma,1} \overset{\text{tight!}}{\leq} \mathrm{rank}(\boldsymbol{A})}$		$\boxed{\|\mathcal{A}\|_{1,1} \overset{\text{tight?}}{\leq} \frac{1}{N}\sum_{n\in\mathbb{N}_N} \mathrm{rank}_n(\mathcal{A})}$

5. A class of Optimization Problems Involving Nuclear Norms

Recent research [36],[37],[38] witnessed an increasing interest in defining composite norms for the purpose of learning and regularization. In this setting the object of interest is typically a parametric model represented by a vector of $\mathbb{R}^{I_1}$ or a matrix of $\mathbb{R}^{I_1\times I_2}$. Regularization via composite norms allows one to convey specific structural a-priori information about the model. In this Section we deal with a general class of non-smooth convex optimization problems aimed at finding a $N-$order tensor $\mathcal{X}$ that represents the model of interest. Specifically, we will mainly deal with the minimization problem:

$$\min_{\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}} F_\mu(\mathcal{X}) := f(\mathcal{X}) + \mu\|\mathcal{X}\|_{1,1} \tag{11}$$

where $\mu > 0$ is a a finite trade-off parameter. The function $f : \mathbb{R}^{I_1\times I_2\times\cdots\times I_N} \rightarrow \mathbb{R}$ is assumed to be convex smooth (i.e. continuously differentiable) with

Lipschitz continuous gradient, i.e.,

$$\|\nabla f(\mathcal{X}) - \nabla f(\mathcal{Y})\|_F \leq L_f \|\mathcal{X} - \mathcal{Y}\|_F, \quad \forall\, \mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}\ , \tag{12}$$

for some positive scalar L_f. Notice that f is application dependent: different specifications of f give rise to different problems, some of which are discussed in Section 7. These different problems all have in common the nuclear-1 norm for tensors

$$\|\mathcal{X}\|_{1,1} = \frac{1}{N} \sum_{n \in \mathbb{N}_N} \|\mathcal{X}_{<n>}\|_{\Sigma,1} \tag{13}$$

that plays the role of a structure inducing penalty. Being the sum of convex functions, the objective function F_μ is also convex and hence the optimization problem (11) is convex. Furthermore it is non-smooth as it follows from (13) and the well known fact that the nuclear norm is non differentiable[4]. In the next Sections we will present an algorithm, termed Convex MultiLinear Estimation (CMLE), and successively elucidate the role of the former problem in data-driven multilinear estimation.

Finally, even though our primary focus is on problem (11) we will also consider the equality constrained formulation:

$$\min_{\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}} F_\mu(\mathcal{X}) \tag{14}$$

$$\text{subject to} \quad A(\mathcal{X}) = \mathcal{Z} \tag{15}$$

where A is some linear map of the type $A : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \to \mathbb{R}^{D_1 \times D_2 \times \cdots D_M}$ and $\mathcal{Z} \in \mathbb{R}^{D_1 \times D_2 \times \cdots D_M}$. We will show how to exploit CMLE to solve this problem and elaborate on its practical importance.

5.1. Connection with Previous Works

Before continuing we point out the relation with the existing literature. In [40] the authors considered the specific task of approximating (in least squares sense) a given tensor by another tensor with pre-specified multilinear ranks. In comparison, our prototypical problem (11) can be specialized to deal with the penalized analogue of this task. Alternatively one can encode hard linear equality constraints via (14)-(15), see Section 7.2. Their method

[4]See e.g. [28] and reference therein for a characterization of the *subdifferential* [39] of nuclear norms.

is based on the $n-$mode tensor norms that can be seen as a specific case of our Lemma 1. In our approach we use the latter to define a composite norm that we use as a penalty function to induce the desired multilinear low rank properties. Instead, they employ separately the $n-$mode tensor norms to devise a multi-stage heuristic based on an alternating approach. This leads to a set of semidefinite programming subproblems for the solution of which the authors use a general SDP convex solver. In contrast, our approach is not based on a multi-stage heuristic but directly tackles the problem via a unique convex formulation. For the latter a specialized solver is developed in the next Section. More closely related is the approach developed in [19] for tensor completion. The problem formulation considered there reads

$$\begin{array}{rl} \min\limits_{\mathcal{X},\mathcal{Z}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}} & \frac{1}{2}\|\mathcal{X}-\mathcal{Z}\|_F^2 \\ \text{subject to} & \|\mathcal{X}\|_{1,1}\leq c \\ & \mathcal{X}_{\mathfrak{O}}=\mathcal{Z}_{\mathfrak{O}} \end{array} \tag{16}$$

where $\mathcal{X}_{\mathfrak{O}}$ denotes the restriction of $\mathcal{X}$ to the index set $\mathfrak{O}$. This problem represents a specific instance of our general framework. A more detailed discussion on this is deferred to Section 7.2. In order to solve the problem the authors first formulate an ad-hoc relaxed version of (16) and successively employ a coordinate descent strategy. This is closely related to our approach although we explicitly recognize the convex separable nature of the problem and point to the relevant literature. Another important difference is that their algorithm is limited to the specific formulation that they consider whereas our approach can be adapted to different problems, both supervised and unsupervised.

5.2. Nuclear Norm Heuristic and Multilinear Ranks

We begin our discussion of the link with multilinear ranks by pointing out that our structure inducing penalty in (11) treats the different modes in the same way. This aspect can be bypassed by simply considering a weighted version of our nuclear-1 norm. Namely we can replace (13) by

$$\frac{1}{N}\sum_{n\in\mathbb{N}_N}\frac{\|\mathcal{X}_{<n>}\|_{\Sigma,1}}{m_n} \tag{17}$$

where for any $n\in\mathbb{N}_N$, m_n is a user-defined weight. Notice that one could define these weights based upon a previous unweighted estimate. In the con-

text of sparse approximation this approach is used in the so-called adaptive-LASSO [41]. Whereas in some scenarios the plain LASSO is incapable of selecting the right model, it can be shown that its weighted counterpart performs as well as if the sparsity of the true underlying model were given in advance [41]. Now consider that in many different situations one might be interested in solving a non-convex penalized problem with multilinear ranks of the type:

$$\min_{\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}} F_{\text{rank}}(\mathcal{X}) := f(\mathcal{X}) + \frac{\mu}{N}\sum_{n\in\mathbb{N}_N}\text{rank}_n(\mathcal{X}) \tag{18}$$

where as before μ denotes a positive trade-off parameter. It is certainly of interest to explore the relation between the latter and a corresponding convex relaxation involving the (weighted) nuclear-1 norm. A simple result in this direction can be obtained by further relying on the relation between the nuclear norm and the rank function.

Let $\mathbb{R}^N_+$ be the positive orthant $\mathbb{R}^N_+ := \left\{X \in \mathbb{R}^N \ : \ x_n > 0 \ \forall\, n \in \mathbb{N}_N\right\}$. For $M \in \mathbb{R}^N_+$, define the set

$$\mathfrak{S}'_M = \left\{\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N} \ : \ \|\mathcal{X}_{<n>}\|_2 \leq m_n \ \forall\, n \in \mathbb{N}_N\right\}$$

and let the tensor $\mathcal{X}^c$ satisfy

$$\mathcal{X}^c \in \arg\min_{\mathcal{X}\in\mathfrak{S}'_M} F_{\mu,M}(\mathcal{X}) := f(\mathcal{X}) + \frac{\mu}{N}\sum_{n\in\mathbb{N}_N}\frac{\|\mathcal{X}_{<n>}\|_{\Sigma,1}}{m_n} \ . \tag{19}$$

Notice that the latter problem is a constrained version of the formulation obtained replacing the nuclear norm in (11) by its weighted version.

Theorem 5. *For all $n \in \mathbb{N}_N$ it holds that*

$$\frac{\|\mathcal{X}^c_{<n>}\|_{\Sigma,1}}{m_n} \leq \text{rank}_n(\mathcal{X}^c) \ . \tag{20}$$

If the upper bound in the latter is attained for all $n \in \mathbb{N}_N$ then $\mathcal{X}^c$ is a global solution of $\min_{\mathcal{X}\in\mathfrak{S}'_m} F_{\text{rank}}(\mathcal{X})$.

Proof. The first statement follows from the fact that the function $\|\boldsymbol{X}_{(n)}\|_{\Sigma,1}/m_n$ is the convex envelope of $\text{rank}(\boldsymbol{X}_{(n)})$ on

$$\mathfrak{F}^{m_n}_{(n)} := \left\{\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n\times(I_{n+1}I_{n+2}\cdots I_N I_2 I_3\cdots I_{n-1})} \ : \ \|\boldsymbol{X}_{(n)}\|_2 \leq m_n\right\} \ .$$

Now the function

$$\sum_{n\in\mathbb{N}_N} \frac{1}{m_n}\|\boldsymbol{X}_{(n)}\|_{\Sigma,1} \tag{21}$$

is the convex envelope of the function $\sum_{n\in\mathbb{N}_N} \operatorname{rank}(\boldsymbol{X}_{(n)})$ over

$$\mathfrak{F}_M := \mathfrak{F}_{(1)}^{m_1} \times \mathfrak{F}_{(2)}^{m_2} \times \cdots \times \mathfrak{F}_{(N)}^{m_N} \ .$$

In turn this implies that (21) underestimates $\sum_{n\in\mathbb{N}_N} \operatorname{rank}(\boldsymbol{X}_{(n)})$ over

$$\mathfrak{F}'_M = \left\{(\boldsymbol{X}_{(1)},\ldots,\boldsymbol{X}_{(N)}) \in \mathfrak{F}_M : \exists\ \mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N} \text{ s.t. } \boldsymbol{X}_{(n)} = \mathcal{X}_{<n>}\ \forall n \in \mathbb{N}_N\right\}.$$

We can conclude that, for any $\mathcal{X} \in \mathfrak{S}'_M$,

$$f(\mathcal{X}) + \frac{\mu}{N}\sum_{n\in\mathbb{N}} \frac{\|\mathcal{X}_{<n>}\|_{\Sigma,1}}{m_n} \leq f(\mathcal{X}) + \frac{\mu}{N}\sum_{n\in\mathbb{N}} \operatorname{rank}_n(\mathcal{X}) \ . \tag{22}$$

Finally if the equality is attained in (5) for any $n \in N$ then the equality is attained in (22) too. The optimality of $\mathcal{X}^c$ for problem (19) now implies its global optimality for problem (18). □

In light of the former result, a possible approach to deal with problem (18) would be to solve problem (19) for a large value of the entries of M. Successively to determine the global optimality of a solution $\mathcal{X}^c$ one could check a posteriori if for any n the equality (20) is attained. In general, however, this will not be true and one could only conclude that $F_\mu(\mathcal{X}^c) \leq F_{\text{rank}}(\tilde{\mathcal{X}})$ where $\tilde{\mathcal{X}}$ is any global solution of $\min_{\mathcal{X}\in\mathfrak{S}'_M} F_{\text{rank}}(\mathcal{X})$.

In the sequel we only deal with the unweighted version of the nuclear-1 norm both in the Algorithm and the experimental results. However we stress at this point that the material can be easily adapted to deal with the weighted case.

5.3. Continuous Shrinkage and Regularization Path

Optimizing a function f as above subject to hard rank constraints is a non convex problem. Correspondingly, practical algorithms deliver possibly suboptimal solutions which in turn are guaranteed to be rank-$(r_1, r_2, \ldots, r_N)$ where $(r_1, r_2, \ldots, r_N)$ is a prescribed $N-$tuple. On the contrary, by continuous shrinkage, problem (11) produces optimal solutions which only approximately satisfy the prescribed ranks requirements. For the related

problem of sparse reconstruction of vector represented models, continuous shrinkage is known to be of crucial importance. Estimators like the LASSO (*Least Absolute Shrinkage and Selection Operator*, [14]) continuously shrink the model coefficients and eventually set some of them exactly to zero. This results in a more stable approach than by inherently discrete procedures such as *Subset Selection* [42]. By smoothly varying the regularization parameter one obtains an entire set of solutions, the so-called *regularization path* [43]. In the present setting, the latter is constituted by a set of models $\{\hat{\mathcal{X}}_{\mu_t}\}_t$ with approximately decreasing $n-$ranks as the parameter μ_t increases with t. In this way one trades the rank feasibility of a single solution with the global optimality of each element of $\{\hat{\mathcal{X}}_{\mu_t}\}_t$. Next, a specific solution can be selected according to a suitable criterion.

6. Algorithms

It is well known for matrices that a large class of convex optimization problems involving nuclear norms can be recast as semidefinite programs [32]. However, this approach does not scale well with the number of entries of the optimization variable. Software such as SDPT3 [44] and SeDuMi [45] cannot go far beyond 10^4 entries (e.g. a matrix of size 100×100) [13],[46]. Thus the first and main purpose of this Section is to develop a simple and scalable optimization strategy to find a solution of problem (11). In the next Subsections we go through the necessary steps to explain the proposed procedure, reported in Algorithm 1. In Subsection 6.6 we present an accelerated variant that falls within the class of optimal first order method. Successively in Subsection 6.7, we show how to modify these algorithms to cope with the equality constrained problem (14)-(15).

6.1. Fast prototyping of convex programs

Before turning to a customized algorithm, we notice that problem (11) can also be implemented in a straightforward fashion by using one of the on-the-shelf softwares for fast prototyping of convex programs. As an example, in Table 2 we report the code necessary to implement (11) via `CVX` [47], a Matlab-based modeling system for convex optimization. This toolbox allows constraints and objectives to be specified using standard Matlab expression syntax. On the other hand, before being fed to the actual solver, the problem is automatically converted into a standard form. This generates a consider-

Table 2: The code to implement problem (11) in `CVX`, version 1.2. It is assumed the optimization variable is here a $N-$order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The convex function f is user-defined at line 2 whereas the value of the trade-off parameter is specified at line 3. Notice that at line 9 the Matlab function `matricize` (non reported) implements the $n-$mode unfolding of Definition 2. On the same line `norm_nuc` is a built in function of the `CVX` library that computes the nuclear norm for matrices.

```
 1. dims=[I_1 ... I_N];
 2. f=@(X) ...;
 3. mu=...;
 4. cvx_begin
 5.    variables X(dims) nuc_n(N)
 6.    minimize(f(X)+mu*1/N*sum(nuc_n))
 7.    subject to
 8.    for n=1:N
 9.        norm_nuc(matricize(X,n))<=nuc_n(n);
10.    end
11. cvx_end
```

able number of slack variables, a fact that penalizes running times even for small size problems. We now turn to our customized solution strategy.

6.2. Equivalent Separable Convex Formulation

For $n \in \mathbb{N}_N$ let $f_n : \mathbb{R}^{I_n \times (I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N)} \to \mathbb{R}$ be defined as

$$f_n(\mathcal{X}_{<n>}) := f((\mathcal{X}_{<n>})^{<n>})$$

that is, as the composition of f with the tensorization operator $\cdot^{<n>}$. We begin by recasting problem (11) into the following equivalent form

$$\min_{\mathcal{X}, \boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(N)}} \frac{1}{N} \sum_{n \in \mathbb{N}_N} \left(f_n(\boldsymbol{X}_{(n)}) + \mu \|\boldsymbol{X}_{(n)}\|_{\Sigma,1} \right) \tag{23}$$

$$\text{subject to } \boldsymbol{X}_{(n)} = \mathcal{X}_{<n>}, \quad n \in \mathbb{N}_N \tag{24}$$

where $\{\boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(N)}\}$ are additional slack matrix variables. This problem consists of minimizing a separable objective function subject to the coupling

linear equality constraints (24). The latter ensure that each matrix variable $\boldsymbol{X}_{(n)}$ is in fact the $n-$mode unfolding of the same tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Separable convex problems [20], [48], [49] i.e. problems with a separable objective function that are coupled via constraints, arise in many fields of engineering such as networks [50], [51], distributed model predictive control (MPC) [52] and stochastic programming [53].

6.3. Solution via Sequential Convex Programming

For any $n \in \mathbb{N}_N$ define for a scalar $c > 0$ the function of matrices:

$$\bar{F}_{n,c}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) := f_n(\boldsymbol{Y}) + \langle \nabla f_n(\boldsymbol{Y}), \boldsymbol{X} - \boldsymbol{Y} \rangle + \mu \|\boldsymbol{X}\|_{\Sigma,1} + \frac{c}{2} \|\boldsymbol{Z} - \boldsymbol{Y}\|_F^2 \,. \quad (25)$$

Our approach is to model (23) at the current estimate $\bar{\mathcal{X}}^{(i)}$ by a proximal subproblem based on (25) and then use the minimizer of this subproblem to define the new estimate $\bar{\mathcal{X}}^{(i+1)}$. More specifically, at the i-th iterate we solve the *proximal subproblem*:

$$\min_{\mathcal{X}, \boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(N)}} \frac{1}{N} \sum_{n \in \mathbb{N}_N} \bar{F}_{n,c_1} \left(\boldsymbol{X}_{(n)}, \bar{\mathcal{X}}^{(i)}_{<n>}, \mathcal{X}_{<n>} \right) \quad (26)$$

$$\text{subject to } \boldsymbol{X}_{(n)} = \mathcal{X}_{<n>}, \quad n \in \mathbb{N}_N \quad (27)$$

where $c_1 > 0$ is proximal parameter. A comparison between the objective functions in (23) and (26) highlights the specific structure of the latter. In the proximal subproblem (26)-(27), the objective function is formed by linearizing f_n around the current estimate $\bar{\mathcal{X}}^{(i)}_{<n>}$ and adding a proximal quadratic term. For matrix problems related proximal expansions were considered in [54] and [17]. For both matrices and higher order tensors this type of approximation allows one to conveniently solve each subproblem by relying on a result about nuclear norm optimization recalled below.

6.4. Augmented Lagrangian Approach

To solve problem (26)-(27) we apply the Alternating Direction Method of Multipliers. A full derivation of the general method, originally developed in the 70's, can be found in [20, Section 3.4.4] together with convergence proofs. Interestingly there seems to be a renewed interest in this approach [55]. The technique is based on the Augmented Lagrangian function [56] that

for problem (26)-(27) reads:

$$L(\mathcal{X}, \boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(N)}, \boldsymbol{P}_{(1)}, \ldots, \boldsymbol{P}_{(N)}) := \frac{1}{N} \sum_{n \in \mathbb{N}_N} \Big(\bar{F}_{n,c_1} \left(\boldsymbol{X}_{(n)}, \bar{\mathcal{X}}^{(i)}_{<n>}, \mathcal{X}_{<n>} \right) - \left\langle \boldsymbol{P}^{(k)}_{(n)}, \boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>} \right\rangle + \frac{c_2}{2} \| \boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>} \|^2_F \Big) \quad (28)$$

where $c_2 > 0$ is a suitable parameter. The approach consists now of iteratively updating $\mathcal{X}$, $\{\boldsymbol{X}_{(1)}, \ldots, \boldsymbol{X}_{(N)}\}$ and finally carrying out the update of the dual variables $\{\boldsymbol{P}_{(1)}, \ldots, \boldsymbol{P}_{(N)}\}$. Each update involves a single variable and is conditioned to the fixed value of the others. A considerable advantage is that to a large extent each iterate can be performed in a distributed way since it consists of independent problems. At the $k-$th iterate[5] the update of $\boldsymbol{X}_{(n)}$ for $n \in \mathbb{N}_N$ is based on solving the unconstrained optimization problem:

$$\min_{\boldsymbol{X}_{(n)}} \quad \bar{F}_{n,c_1} \left(\boldsymbol{X}_{(n)}, \bar{\mathcal{X}}^{(i)}_{<n>}, \mathcal{X}_{<n>} \right) - \left\langle \boldsymbol{P}^{(k)}_{(n)}, \boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>} \right\rangle + \tfrac{c_2}{2} \| \boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>} \|^2_F \ . \quad (29)$$

Concerning this we have the following result.

Proposition 1. *A matrix $\hat{\boldsymbol{X}}_{(n)}$ is optimal for problem* (29) *if and only if it is optimal for*

$$\min_{\boldsymbol{X}_{(n)}} \quad \tfrac{1}{2} \left\| \boldsymbol{X}_{(n)} - \left(\mathcal{X}^{(k)}_{<n>} - \tfrac{1}{c_2} \nabla f_n \left(\bar{\mathcal{X}}^{(i)}_{<n>} \right) + \tfrac{1}{c_2} \boldsymbol{P}^{(k)}_{(n)} \right) \right\|^2_F + \tfrac{\mu}{c_2} \| \boldsymbol{X}_{(n)} \|_{\Sigma,1} \ . \quad (30)$$

Proof. The result is obtained by simple calculus. Replacing (25) into the objective of (29) we obtain

$$f_n(\bar{\mathcal{X}}^{(i)}_{<n>}) + \langle \nabla f_n(\bar{\mathcal{X}}^{(i)}_{<n>}), \boldsymbol{X}_{(n)} - \bar{\mathcal{X}}^{(i)}_{<n>} \rangle + \mu \| \boldsymbol{X}_{(n)} \|_{\Sigma,1} + \frac{c_1}{2} \| \mathcal{X}_{<n>} - \bar{\mathcal{X}}^{(i)}_{<n>} \|^2_F + \frac{c_2}{2} \| \boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>} \|^2_F - \left\langle \boldsymbol{P}^{(k)}_{(n)}, \boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>} \right\rangle \ . \quad (31)$$

We now substitute $\boldsymbol{X}_{(n)} - \bar{\mathcal{X}}^{(i)}_{<n>}$ by $\boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>}$ in the second term, drop the terms that do not depend on $\boldsymbol{X}_{(n)}$ and add $\frac{1}{2c_2} \| \nabla f_n(\bar{\mathcal{X}}^{(i)}_{<n>}) - \boldsymbol{P}^{(k)}_{(n)} \|^2_F$ to

[5] Notice that by index k we refer to the *inner* iterations necessary to solve the subproblem (26)-(27). On the other hand we indexed by i the *outer* iterations that refers to the sequential convex approximations around successive estimate $\bar{\mathcal{X}}^{(i)}$.

(31):

$$\langle \nabla f_n(\bar{\mathcal{X}}^{(i)}_{<n>}), \boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>} \rangle + \mu \|\boldsymbol{X}_{(n)}\|_{\Sigma,1} + \frac{1}{2c_2}\|\nabla f_n(\bar{\mathcal{X}}^{(i)}_{<n>}) - \boldsymbol{P}^{(k)}_{(n)}\|_F^2 + \frac{c_2}{2}\|\boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>}\|_F^2 - \left\langle \boldsymbol{P}^{(k)}_{(n)}, \boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>} \right\rangle . \quad (32)$$

Dividing by c_2 gives:

$$\frac{1}{c_2}\langle \nabla f_n(\bar{\mathcal{X}}^{(i)}_{<n>}) - \boldsymbol{P}^{(k)}_{(n)}, \boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>} \rangle + \frac{1}{2c_2^2}\|\nabla f_n(\bar{\mathcal{X}}^{(i)}_{<n>}) - \boldsymbol{P}^{(k)}_{(n)}\|_F^2 + \frac{\mu}{c_2}\|\boldsymbol{X}_{(n)}\|_{\Sigma,1} + \frac{1}{2}\|\boldsymbol{X}_{(n)} - \mathcal{X}^{(k)}_{<n>}\|_F^2 . \quad (33)$$

Simple manipulation leads now to (30). The proposition now follows from the fact that none of the performed operations change the optimal point in the minimization with respect to $\boldsymbol{X}_{(n)}$. □

This reformulation immediately allows us to exploit the following result.

Theorem 6 (Singular Values Soft-thresholding Operator [57])**.** *For $\boldsymbol{U} \in \mathbb{R}^{I_1 \times r}$, $\boldsymbol{\Sigma} = \operatorname{diag}\left(\{\sigma_i\}_{i\in\mathbb{N}_r}\right) \in \mathbb{R}^{r\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{I_2\times r}$ let $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be the (thin) SVD of $\boldsymbol{Y} \in \mathbb{R}^{I_1\times I_2}$ and define the Singular Values Soft-thresholding Operator D_τ by*

$$D_\tau(\boldsymbol{Y}) := \boldsymbol{U} \operatorname{diag}\left(\{\mathcal{I}(\sigma_i - \tau)\}_{i\in\mathbb{N}_r}\right)\boldsymbol{V}^\top .$$

Then we have

$$D_\tau(\boldsymbol{Y}) = \arg\min_{\boldsymbol{X}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Y}\|_F^2 + \tau\|\boldsymbol{X}\|_{\Sigma,1} .$$

In light of the former result, the unique solution of (29) is simply

$$D_{\frac{\mu}{c_2}}\left(\mathcal{X}^{(k)}_{<n>} - \frac{1}{c_2}\nabla f_n\left(\bar{\mathcal{X}}^{(i)}_{<n>}\right) + \frac{1}{c_2}\boldsymbol{P}^{(k)}_{(n)}\right) .$$

The update rule for $\mathcal{X}$ is obtained from (25) starting by first order optimality conditions. For any $n \in \mathbb{N}_N$, $\boldsymbol{P}_{(n)}$ is adjusted according to a standard procedure, see [56, Section 13.4] for details.

Overall, each inner iterate necessary to solve problem (26)-(27) is dominated by the cost of computing N SVDs, the same as for the MLSVD in

Section 3. In the standard practice of Augmented Lagrangian approaches the parameter c_2 is either kept fixed or increased along the iterations [20],[56]. Here for $\beta > 1$ we adopt the updating rule $c_2^{(k)} = \beta\, c_2^{(k-1)}$ and let $c_2^{(1)} = \alpha\, c_1$ for some $0 < \alpha \leq 1$. In our implementation we take $\alpha = 0.1$ and $\beta = 1.1$. Next we address the choice of a suitable parameter c_1.

6.5. Choice of the proximal parameter

To do so, let us define

$$\bar{F}_{c_1}(\mathcal{X}, \mathcal{Y}) := f(\mathcal{Y}) + \langle \nabla f(\mathcal{Y}), \mathcal{X} - \mathcal{Y} \rangle + \mu \|\mathcal{X}\|_{1,1} + \frac{c_1}{2} \|\mathcal{Y} - \mathcal{X}\|_F^2 \ .$$

Observe that the next estimate $\bar{\mathcal{X}}^{(i+1)}$ obtained via problem (26)-(27) could be equivalently found solving the unconstrained optimization problem:

$$\bar{\mathcal{X}}^{(i+1)} = \arg \min_{\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}} \bar{F}_{c_1}\left(\mathcal{X}, \bar{\mathcal{X}}^{(i)}\right) \ . \tag{34}$$

This shows that the solution strategy presented so far can be seen as a first order method in the tensor unknown. The fact that ∇f is Lipschitz continuous with parameter L_f ensures now that [39]

$$f(\mathcal{X}) \leq f(\mathcal{Y}) + \langle \nabla f(\mathcal{Y}), \mathcal{X} - \mathcal{Y} \rangle + \frac{L_f}{2} \|\mathcal{X} - \mathcal{Y}\|_F^2, \quad \forall\, \mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \ . \tag{35}$$

With reference to (11) this implies that, for any choice of $c_1 \geq L_f$, the sequence of function values generated by the algorithm is non-increasing, namely we have

$$F_\mu\left(\bar{\mathcal{X}}^{(i+1)}\right) \leq \bar{F}_{c_1}\left(\bar{\mathcal{X}}^{(i+1)}, \bar{\mathcal{X}}^{(i)}\right) \leq \bar{F}_{c_1}\left(\bar{\mathcal{X}}^{(i)}, \bar{\mathcal{X}}^{(i)}\right) = F_\mu\left(\bar{\mathcal{X}}^{(i)}\right) \ ,$$

as expected. This suggests that the choice of c_1 should be based upon the Lipschitz constant. In turn, a number of results help the user in determining (an upper-bound on) the latter, see e.g. [39]. Still in practical applications, especially when the dimensionality of the problem is considerable, it might be inconvenient to compute it. Here we follow the line search strategy suggested in [58] on the base of the defining property (35). Starting from an optimistic initial guess $c_1^{(0)} < L_f$, the Lipschitz parameter is adjusted according to two update parameters $\gamma_u > 1$ and $\gamma_d \geq 1$ ($\gamma_u = 2$ and $\gamma_d = 1.1$ in our practical implementation). The entire procedure is reported in Algorithm

1. To improve readability the Alternating Direction strategy used within CMLE to solve (26)-(27) is reported in a separate pseudocode environment (Algorithm 2).

Two additional remarks are in order. First of all we stress that CMLE can be easily adapted to any specific choice of f. In fact the only parts that are problem-dependent are the evaluation of f and that of ∇f. Secondly we remark that in many applications we are interested in computing solutions associated to an entire grid of adjacent parameter values $\{\mu_t\}_t$. In these cases we can exploit warm starting: often only a small number of iterations is required to compute a solution for μ_t, provided that we initialize CMLE according to the solution obtained for μ_{t-1}.

Algorithm 1: $\text{CMLE}_f(\mathcal{X}^{(0)}, \boldsymbol{P}_{(1)}^{(0)}, \ldots, \boldsymbol{P}_{(N)}^{(0)}, \mu, c_1^{(0)}, \gamma_u, \gamma_d)$

---

comment: Solve (11) by generating a sequence of proximal problems

$c_1 \leftarrow c_1^{(0)}$

$i \leftarrow 1$

$\bar{\mathcal{X}}^{(i)} \leftarrow \mathcal{X}^{(0)}$

$\boldsymbol{P}_{(n)}^{(i)} \leftarrow \boldsymbol{P}_{(n)}^{(0)}$

repeat

$c_1 \leftarrow \max\left(c_1^{(0)}, \frac{c_1}{\gamma_d}\right)$

$(\mathcal{V}, \boldsymbol{Q}_{(1)}, \ldots, \boldsymbol{Q}_{(N)}) \leftarrow \textsc{AlternatingDirection}(\bar{\mathcal{X}}^{(i)}, \boldsymbol{P}_{(1)}^{(i)}, \ldots, \boldsymbol{P}_{(N)}^{(i)}, \mu, c_1)$

while $f(\mathcal{V}) > f(\bar{\mathcal{X}}^{(i)}) + \left\langle \nabla f\left(\bar{\mathcal{X}}^{(i)}\right), \mathcal{V} - \bar{\mathcal{X}}^{(i)} \right\rangle + \frac{c_1}{2}\|\mathcal{V} - \bar{\mathcal{X}}^{(i)}\|_F^2$

do $c_1 \leftarrow \gamma_u c_1$

$\bar{\mathcal{X}}^{(i+1)} \leftarrow \mathcal{V}$

$\boldsymbol{P}_{(n)}^{(i+1)} \leftarrow \boldsymbol{Q}_{(n)}$

$i \leftarrow i + 1$

until converge criterion met

return $(\bar{\mathcal{X}}^{(i-1)}, \boldsymbol{P}_{(1)}^{(i-1)}, \ldots, \boldsymbol{P}_{(N)}^{(i-1)}, c_1)$

Algorithm 2: AlternatingDirection($\mathcal{X}^{(0)}, \boldsymbol{P}_{(1)}^{(0)}, \ldots, \boldsymbol{P}_{(N)}^{(0)}, \mu, c_1$)

comment: Solve problem (26)-(27).

$\mathcal{S} \leftarrow \mathcal{X}^{(0)}$

$\mathcal{S}_{\nabla} \leftarrow \nabla f\left(\mathcal{X}^{(0)}\right)$

$k \leftarrow 1$

$c_2 \leftarrow \alpha\, c_1$

$\boldsymbol{P}_{(n)}^{(k)} \leftarrow \boldsymbol{P}_{(n)}^{(0)}$

$\boldsymbol{X}_{(n)}^{(k)} \leftarrow \mathcal{X}_{\langle n \rangle}^{(0)}$

repeat

for each $n \in \mathbb{N}_N$

do $\begin{cases} \boldsymbol{X}_{(n)}^{(k+1)} \leftarrow D_{\frac{\mu}{c_2}}\left(\left(\mathcal{X}^{(k)} - \frac{1}{c_2}\mathcal{S}_{\nabla}\right)_{<n>} + \frac{1}{c_2}\boldsymbol{P}_{(n)}^{(k)}\right) \\ \boldsymbol{P}_{(n)}^{(k+1)} \leftarrow \boldsymbol{P}_{(n)}^{(k)} + c_2\left(\mathcal{X}_{<n>}^{(k)} - \boldsymbol{X}_{(n)}^{(k+1)}\right) \end{cases}$

$\mathcal{X}^{(k+1)} \leftarrow \frac{c_1}{c_1+c_2}\mathcal{S} + \frac{1}{N(c_1+c_2)}\sum_{n\in\mathbb{N}_N}\left(c_2\boldsymbol{X}_{(n)}^{(k+1)} - \boldsymbol{P}_{(n)}^{(k+1)}\right)^{<n>}$

$c_2 \leftarrow \beta\, c_2$

$k \leftarrow k + 1$

until convergence criterion met

return $(\mathcal{X}^{(k-1)}, \boldsymbol{P}_{(1)}^{(k-1)}, \ldots, \boldsymbol{P}_{(N)}^{(k-1)})$

6.6. Accelerated CMLE

For smooth optimization problems it is known that the convergence rate of the standard gradient method is suboptimal within the class of first order methods. In turn, there exist certain accelerated first order schemes that achieve the optimal rate [59],[39]. Recently it was shown that the optimal rate can be also achieved for problems like (11) where the objective is formed by a smooth part and a simple non-smooth one [58],[60],[61]. This is the case, in particular, for l_1 penalized problems [58],[60] as well as matrix problems involving a nuclear norm penalty [17], [54]. In all the cases the accelerated convergence is achieved by successively constructing search points relying on

one or multiple scalars forming the so-called *estimated sequences* [39]. These sequences can be obtained via different updating schemes, see [60],[58],[61] for further details. In accelerating CMLE we form the estimated sequence $\left\{\alpha^{(i)}\right\}_i$ by the same updating scheme that appeared in [60] and was later used in [54]. The resulting procedure is reported in Algorithm 3. Notice that the acceleration pertains the sequence of proximal problems and hence refers to the iterations indexed by i in Algorithm 1. The solution of the proximal problem (26)-(27) is found via Algorithm 2, which iterations are indexed by k, following the same approach used in CMLE. As a consequence the acceleration is effective whenever a significant number of outer iteration is required to achieve the optimum. On the contrary, if in the total number of iteration $\sum_{i\in\mathbb{N}_I} k(i)$ is mainly due to inner iterations (I is a small number) then the acceleration is ineffective. This phenomenon is illustrated by the experiments in Subsections 8.1.2 and 8.4.

6.7. Dealing with Equality Constraints

Either the vanilla CMLE or its accelerated version deal with the unconstrained problem (11). However, as we shall see in the next Section, there are multilinear estimation problems for which a constrained formulation of the type (14)-(15) is best suited. Given the solution strategy developed for problem (11), a simple yet convenient way to deal with these cases is by means of a *penalty function method* [56],[62]. Let $\left\{d^{(t)}\right\}_t$ be a sequence of scalars such that, for any t, $d^{(t+1)} > d^{(t)}$ and, further, $\lim_{t\to\infty} d^{(t)} = +\infty$. In our setting this approach requires solving, for successive values of t, the unconstrained optimization problem[6]

$$\mathcal{X}^{(t)} = \underset{\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}}{\arg\min} f(\mathcal{X}) + \mu\|\mathcal{X}\|_{1,1} + \frac{d^{(t)}}{2}\|A(\mathcal{X}) - \mathcal{T}\|_F^2 \qquad (36)$$

starting from the previous solution $\mathcal{X}^{(t-1)}$. It is clear that for large $d^{(t)}$ we will have $A(\mathcal{X}^{(t)}) \approx \mathcal{T}$. Hence the requirement over $\left\{d^{(t)}\right\}_t$ ensures that for $t \to \infty$ $\mathcal{X}^{(t)}$ will be a feasible point. In turn the approach of slowly increasing the penalty parameter $d^{(t)}$ circumvents possible numerical problems [62]. We follow the latter reference and update the penalty parameters according to $d^{(t)} = \beta d^{(t-1)}$. In particular, in our experiments (see Subsection 8.2) we take

[6] We used a quadratic penalty, by far the most common one [56].

Algorithm 3: A-CMLE$_f(\mathcal{X}^{(0)}, \boldsymbol{P}^{(0)}_{(1)}, \ldots, \boldsymbol{P}^{(0)}_{(N)}, \mu, c^{(0)}, \gamma_u, \gamma_d)$

comment: Accelerated variant of Algorithm 1

$c \leftarrow c^{(0)}$
$i \leftarrow 1$
$\bar{\mathcal{X}}^{(i)} \leftarrow \mathcal{X}^{(0)}$
$\bar{\mathcal{W}}^{(i)} \leftarrow \mathcal{X}^{(0)}$
$\alpha^{(i)} \leftarrow 1$
$\boldsymbol{P}^{(i)}_{(n)} \leftarrow \boldsymbol{P}^{(0)}_{(n)}$
repeat
  $c \leftarrow \max\left(c^{(0)}, \frac{c}{\gamma_d}\right)$
  $(\mathcal{V}, \boldsymbol{Q}_{(1)}, \ldots, \boldsymbol{Q}_{(N)}) \leftarrow$ AlternatingDirection$(\bar{\mathcal{W}}^{(i)}, \boldsymbol{P}^{(i)}_{(1)}, \ldots, \boldsymbol{P}^{(i)}_{(N)}, \mu, c)$
  while $f(\mathcal{V}) > f(\bar{\mathcal{W}}^{(i)}) + \left\langle \nabla f\left(\bar{\mathcal{W}}^{(i)}\right), \mathcal{V} - \bar{\mathcal{W}}^{(i)} \right\rangle + \frac{c}{2}\|\mathcal{V} - \bar{\mathcal{W}}^{(i)}\|_F^2$
    do $c \leftarrow \gamma_u c$
  $\bar{\mathcal{X}}^{(i+1)} \leftarrow \mathcal{V}$
  $\boldsymbol{P}^{(i+1)}_{(n)} \leftarrow \boldsymbol{Q}_{(n)}$
  $a^{(i+1)} \leftarrow \frac{1+\sqrt{1+4\left(\alpha^{(i)}\right)^2}}{2}$
  $\bar{\mathcal{W}}^{(i+1)} \leftarrow \bar{\mathcal{X}}^{(i+1)} + \frac{\alpha^{(i)}-1}{\alpha^{(i+1)}}\left(\bar{\mathcal{X}}^{(i+1)} - \bar{\mathcal{X}}^{(i)}\right)$
  $i \leftarrow i + 1$
until converge criterion met
return $(\bar{\mathcal{X}}^{(i-1)}, \boldsymbol{P}^{(i-1)}_{(1)}, \ldots, \boldsymbol{P}^{(i-1)}_{(N)})$

$\beta = 1.1$. For each t a solution of (36) can be readily computed by means of Algorithm 1 or 3 as soon as we replace ∇f in the pseudocode of the latter by ∇f_t where f_t is defined as:

$$f_t(\mathcal{X}) := f(\mathcal{X}) + \frac{d^{(t)}}{2}\|A(\mathcal{X}) - \mathcal{T}\|_F^2 \ . \tag{37}$$

This fact suggests the procedure reported in Algorithm 2 and termed Equality-constrained Convex MultiLinear Estimation (E-CMLE). Within the main iteration, indexed by t, we tackle problem (36) by performing a fixed small number of iterations (say 10) of (A-)CMLE and successively increase t and

update f_t and ∇f_t accordingly.

Algorithm 4: E-CMLE($\mathcal{X}^{(0)}, \boldsymbol{P}_{(1)}^{(0)}, \ldots, \boldsymbol{P}_{(N)}^{(0)}, \mu, c^{(0)}, \gamma_u, \gamma_d$)

comment: Solve problem (14)-(15) via Algorithm 1

$t \leftarrow 1$

repeat

$\left(\mathcal{X}^{(t)}, \boldsymbol{P}_{(1)}^{(t)}, \ldots, \boldsymbol{P}_{(N)}^{(t)}, c^{(t)}\right)$ (A-)CMLE$_{f_t}(\mathcal{X}^{(t-1)}, \boldsymbol{P}_{(1)}^{(t-1)}, \ldots, \boldsymbol{P}_{(N)}^{(t-1)}, \mu, c^{(t-1)}, \gamma_u, \gamma_d)$

$t \leftarrow t + 1$

until convergence criterion met

7. Multilinear Estimation with Nuclear Norm Penalties

In this section we turn to the methodological aspects pertaining data-driven modeling and discuss both supervised and unsupervised applications. More specifically we present two special cases of (11) that arise from the generalization of vector and matrix problems studied in the literature. In previous works [63],[64] tensor representations of structured data were shown to be crucial for improving generalization of learning machines trained with a limited number of observations. This comes as no surprise: it has been argued that an appropriate input representation is a crucial aspect for learning machines when the number of training examples is limited [65]. On the other hand the ability to learn from very few examples has been recognized as an important aspect of intelligence. For the sake of clarity, we begin by discussing the difference between what we consider here as supervised and unsupervised multilinear estimation (or learning) and point out the distinction with respect to more classical views based on vector valued input data.

7.1. A Taxonomy of Learning Multilinear Models

In the classical setting of statistical machine learning[7] the modeler is faced with a set of input observations (a.k.a. input instances, input points, input

[7]See e.g. the first Chapter of [66].

data, input patterns or input examples) that represent specific objects. Each of such instances is often represented as a $D-$dimensional vector $Z \in \mathbb{R}^D$. By contrast, here we consider the more general case where each instance is represented by a tensor $\mathcal{Z} \in \mathbb{R}^{D_1 \times D_2 \times \cdots \times D_M}$ which might capture more naturally certain aspects of the represented object.

In *unsupervised learning* we are given a training set $\left\{\mathcal{Z}^{(i)}\right\}_{i \in \mathbb{N}_K}$ where $K \geq 1$. Following a classical assumption for vector-represented data, we might assume that the objects are produced by a *generator* inducing a corresponding probability model $P(\mathcal{Z})$ on the associated tensor representations. Loosely speaking the common goal of unsupervised learning techniques is then to find interesting structure in the training set. This framework includes tasks such as *clustering*, *novelty detection* and *dimensionality reduction* that can be defined similarly as for the case of vector-valued inputs [66]. The problem of *factorization* as well as that of *denoising* and *completion* considered in the next Subsection can be regarded as a special case where $K = 1$.

In *supervised learning*, on the other hand, we have additional information in the form of *output* instances (a.k.a. labels). Depending on the specific task, the output space $\mathfrak{Y}$ is either $\mathbb{R}$ or some discrete set, such as $\mathfrak{Y} = \{+1, -1\}$. By contrast, notice that the distinctive characteristic of unsupervised learning is that there is no *supervision*: the only information available corresponds to the — possibly singleton — set of input points. In supervised learning the training set is instead composed by given input-output pairs $\left\{\left(y_i, \mathcal{Z}^{(i)}\right)\right\}_{i \in \mathbb{N}_K}$ that are assumed to be drawn from a fixed but unknown joint distribution $P(y, \mathcal{Z})$. The general objective of supervised methods amounts to approximate the so-called *supervisor response* $P(y|\mathcal{Z})$ by estimating a mapping from $\mathcal{X}$ to the associated y in a given function class. In the following we consider the set of linear functions of $\mathcal{Z}$ namely functions of the form $\langle \mathcal{Z}, \mathcal{X} \rangle$ for some $\mathcal{X} \in \mathbb{R}^{D_1 \times D_2 \times \cdots \times D_M}$.

7.2. Denoising and Minimum-rank Tensor Completion Problem

As a special case of unsupervised learning we consider here the following task. Suppose we are given some linear map $A : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \to \mathbb{R}^{D_1 \times D_2 \times \cdots \times D_M}$ and want to recover a low-rank tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ such that $A(\mathcal{X})$ is close in some metric to $\mathcal{Z} \in \mathbb{R}^{D_1 \times D_2 \times \cdots \times D_M}$. Recall the unsupervised setting that we discussed above and notice that in such a framework $\mathcal{Z}$ can be seen to play the role of the unique training point. Furthermore let the norm $\|\cdot\|_\star$ denote the metric of interest, assumed to be smooth. Then the described

task can be accomplished by solving the following problem:

$$\min_{\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}} \|A(\mathcal{X})-\mathcal{Z}\|_\star^2+\mu\|\mathcal{X}\|_{1,1}\ . \tag{38}$$

If we let $f(\mathcal{X})=\|A(\mathcal{X})-\mathcal{Z}\|_\star^2$ the latter is readily seen to be a special case of (11). The differentiability of $\|\cdot\|_\star$ ensures that the smoothness requirement over f is satisfied and ultimately guarantees the applicability of our approach. With respect to related constrained formulations, such as

$$\begin{aligned}\min_{\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}} \quad & \|\mathcal{X}\|_{1,1}\\ \text{subject to} \quad & A(\mathcal{X})=\mathcal{Z}\end{aligned} \tag{39}$$

problem (38) is especially suited for the case where measurements are contaminated with noise. In the simplest application, in fact, $\mathcal{Z}$ is a given noisy tensor observation and we are interested in recovering its latent version under the assumption that the latter should have low $n-$ranks. This amounts to the case where A corresponds to the identity mapping, namely

$$(A(\mathcal{X}))_{i_1^j,\ldots,i_N^j}=x_{i_1^j,\ldots,i_N^j}\ . \tag{40}$$

The problem is much more general though, and one can easily envision applications in the most diverse fields. In some cases of interest, for example, it makes sense to represent the input information as a vector $Z\in\mathbb{R}^D$ and define A entry-wise as $(A(\mathcal{X}))_j=z_j$. In the *tensor completion problem*, for instance, we let

$$\mathfrak{O}=\left\{(i_1^j,\ldots,i_N^j)\in\mathbb{N}_{I_1}\times\cdots\times\mathbb{N}_{I_N}:j\in\mathbb{N}_K\right\}$$

and define A by

$$(A(\mathcal{X}))_j=x_{i_1^j i_2^j\cdots i_N^j}\ . \tag{41}$$

With reference to (16), notice that we have $A(\mathcal{X})=\mathcal{X}_{\mathfrak{O}}$. Further, it is not difficult to see that this set-up turns (38) into a penalized version of (16) as soon as we let $\|\cdot\|_\star$ be the Frobenius norm. Alternatively, our framework allows one to tackle the equality constrained problem (46) by relying on a specialized version of (14)-(15). In fact, with respect to the latter problem it is enough to let f be the zero function and set $\mu=1$. The use of Algorithm 4 to solve this constrained optimization problem is considered in Subsection 8.2

in a number of experiments. For the case of matrices, the completion problem has been extensively studied in the recent literature, see e.g. [28],[17],[32] and references therein. In the area of recommendation systems, for example, users submit ratings on a subset of entries of a users/items grid. In order to provide recommendations based on the users' preferences vendors want to infer the values of the missing entries. The task is ill-posed since there are infinitely many ways to fill the missing values and usually a very limited number of rated items is available. To correct for ill-posedness one then exploits the common belief that users behavior is dictated by a limited number of common factors. This is translated into the low-rank assumption and makes the search feasible by suitably constraining the hypothesis space via the nuclear norm penalty.

7.3. Penalized Empirical Risk Minimization

To illustrate another special case of (11), for an output set $\mathfrak{Y}$ let $l : \mathfrak{Y} \times \mathbb{R} \to \mathbb{R}^+$ be a *loss function* assumed to be convex and differentiable with respect to its second argument. Given input-output pairs $\left\{\left(y_i, \mathcal{Z}^{(i)}\right)\right\}_{i \in \mathbb{N}_K}$, a number of *supervised learning* tasks involving tensor valued input data can be tackled via the optimization problem:

$$\min_{\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}} \sum_{i \in \mathbb{N}_K} l\left(y_i, \left\langle \mathcal{Z}^{(i)}, \mathcal{X} \right\rangle\right) + \mu \|\mathcal{X}\|_{1,1} \ , \tag{42}$$

again a special case of problem (11). Notice that one can easily modify the problem (and the corresponding solution strategy) to consider a bias term, as we do in the experiments about classification in Section 8. In this case the value of the estimated output is $\hat{y} = \langle \mathcal{Z}, \mathcal{X} \rangle + b$ instead of simply $\hat{y} = \langle \mathcal{Z}, \mathcal{X} \rangle$. Concerning the loss function common choices are the quadratic loss for regression/binary classification:

$$l(y, \hat{y}) = \frac{1}{2}\left(y - \hat{y}\right)^2 \tag{43}$$

and the logistic loss for binary classification:

$$l(y, \hat{y}) = \log\left(1 + e^{-y\hat{y}}\right) \ . \tag{44}$$

This falls within the general *penalized empirical risk minimization* paradigm that has been extensively studied for the case where the generic input point

$\mathcal{Z}$ is a vector [67],[68]. The generalization in (42) might be interesting in a number of cases. As an example in the next Section we consider the classification of human actions from videos, a task that is motivated by many surveillance applications [69]. Notice that replacing $\|\cdot\|_{1,1}$ by $\|\cdot\|_F^2$ is equivalent to vectorizing the input tensors and then use the squared l_2 norm penalty as in *ridge regression* [70]. Similarly, using the l_1 norm penalty after vectorization, corresponds to impose a sparse prior over the entries of the model, as in the LASSO. In either case the tensor structure of the problem is ignored. On the contrary, the nuclear norm penalty $\|\cdot\|_{1,1}$ corresponds to the assumption that the different modes of the supervisor response jointly lie in low dimensional subspaces. In related problems, this type of assumption has been proven to be useful, especially when the number of input-output pairs is limited [63],[64].

8. Numerical Examples

In this Section we illustrate the attributes of the proposed approach with reference to the tasks of Section 7.2.

8.1. Tensor Denoising

As a first example we consider the application of (38) for the recovering of a tensor with small multilinear rank from a noisy version. Our purpose is mainly to explore the rank revealing property of the proposed nuclear norm. To do so we computed the entire regularization path on a tight grid of parameters. As a baseline, we compared with the regularization path obtained using $\|\cdot\|_F^2$ as the penalty in place of $\|\cdot\|_{1,1}$.

8.1.1. Data Generation

As training pattern we took

$$\mathcal{Z} = \mathcal{Z}^t + \eta\, \mathcal{N} \tag{45}$$

where $\mathcal{Z}^t \in \mathbb{R}^{5\times 6\times 7}$ was a latent tensor with $\text{rank}_1(\mathcal{Z}^t) = \text{rank}_2(\mathcal{Z}^t) = \text{rank}_3(\mathcal{Z}^t) = 2$, $\mathcal{N}$ was a tensor with normally distributed entries and finally η was set so that the signal-to-noise ratio (SNR [8]) was approximately 4. With reference to problem (38) we let $\|\cdot\|_\star$ be the Frobenius norm and let A be as

[8] We use $\text{SNR} = ((1/342)\sum_i(\text{vec}(\mathcal{Z}^t)_i - m)^2)/(\eta^2)$ where $m = (1/343)\sum_i \text{vec}(\mathcal{Z}^t)_i$.

in (40). In Figure 1 we reported the evolution of the $n-$mode singular values of solutions obtained for decreasing values of the parameter μ. In Figure 2(a) we reported the mean squared errors defined as MSE $:= \frac{1}{343}\|\mathcal{Z}^t - \hat{\mathcal{X}}\|_F^2$ where $\hat{\mathcal{X}}$ is a generic solution along the path and 343 is the number of entries of the tensor. As a reference we also reported the MSE of 500 rank-$(2,2,2)$ tensors computed with the Higher-Order Orthogonal Iteration (HOOI, [4]) algorithm initialized with randomly generated factors.

8.1.2. Comparison of Running Times: Denoising

Table 3: Comparison of running times for denoising in randomly generated problems. The tensor unknown is $\mathcal{X} \in \mathbb{R}^{m\times m\times m}$ and m goes up to 61, corresponding to about $2.27 \cdot 10^5$ entries. Both CMLE and A-CMLE converge usually within $10-15$ outer iterations (indexed by i in Algorithm 1 and 3). The accelerated variant performs worse in this case, see Subsection 6.6. Missing values, denoted by $-$, indicates impossibility to complete the optimization.

CPU times (in sec.)

m=4			m=12		
CMLE	A-CMLE	`CVX`	CMLE	A-CMLE	`CVX`
0.07(0.03)	**0.07(0.03)**	0.56(0.12)	**0.16(0.11)**	0.21(0.16)	97.75(6.78)
m=25			m=36		
CMLE	A-CMLE	`CVX`	CMLE	A-CMLE	`CVX`
1.96(0.10)	2.16(0.06)	–	**6.05(0.16)**	6.56(0.13)	–
m=48			m=61		
CMLE	A-CMLE	`CVX`	CMLE	A-CMLE	`CVX`
21.09(0.63)	22.97(1.05)	–	**48.40(1.16)**	50.82(1.79)	–

In order to assess the time complexity of the proposed solution strategies we solved multiple random problems of the type above on a standard computer[9]. More specifically we generated full rank tensors via (45) where $\mathcal{Z}^t \in \mathbb{R}^{m\times m\times m}$ and the level of noise was set so that the SNR was 10. For each m in Table 3 we reported mean and standard deviation (in parenthesis) of CPU times required to solve a population of 20 such problems with fixed regularization parameter $\mu = 1$. We tested our specialized solution strategy (Algorithm 1 and 3 implemented in Matlab) against the `CVX` implementation

[9]Intel Core 2 Duo with a 2,4 GHz processor and 4 GB of RAM.

reported in Table 2. Termination criteria were set to give a comparable level of accuracy.

8.2. Image Completion

In this Subsection we consider the case where we are provided with an incomplete color image $\mathcal{Z}$, represented as a $3-$order tensor, and want to reconstruct its latent true version $\mathcal{Z}^t$ [19]. For the reconstruction we considered the constrained version of the completion problem discussed in Subsection 7.2, namely:

$$\begin{array}{ll} \min\limits_{\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}} & \|\mathcal{X}\|_{1,1} \\ \text{subject to} & \mathcal{X}_\Omega = Z \end{array} \tag{46}$$

where elements of Ω index here the given pixels and entries of Z report the value of those pixels in the RGB space. As an important advantage this formulation does not require the setting of regularization parameters. To compute a solution we use Algorithm 4 where we have, at iteration t,

$$f_t(\mathcal{X}) = \frac{d_t}{2}\|\mathcal{X}_\Omega - \mathcal{T}\|_F^2 \ .$$

Figure 3 reports the original pictures, the pixel provided for the reconstruction and the outcome of our algorithm for a number of examples.

8.3. Classification: Toy Example

Here we apply problem (42) to the case of classification with quadratic loss. We considered a simple toy problem where we generated input data in $\mathbb{R}^{5\times5\times8}$ with entries distributed according to a standard gaussian model. The output set was $\mathfrak{Y} = \{+1, -1\}$ and labels were generated according to $y = \text{sign}\,(\bar{y})$ where $\bar{y}$ followed a normal distribution with mean $\langle\mathcal{Z}, \mathcal{X}\rangle + b$ and variance η, $\mathcal{Z}$ was a tensor with multilinear ranks $(2, 2, 2)$ and the value of b was set to have a pre-specified probability $P(y = +1)$. We solved problem (42) with K input-output pairs $\left\{\left(y_i, \mathcal{Z}^{(i)}\right)\right\}_{i\in\mathbb{N}_K}$ and test the performance of the obtained classification model against the classifiers obtained — after vectorization of the input patterns — via a standard implementation of *least-squares support vector machines* [71] (LS-SVMlab toolbox [72]) with both linear and gaussian-RBF kernel for different values of $P(y = +1)$, η and K. Notice that this specific choice was made to have a comparison between methodologies sharing the same loss function (43) but differing in the type of

regularization mechanism used. In all the cases the parameters were chosen according to the leave-one-out misclassification error (see the tutorial [72] for details). Figure 4 depicts the receiver operating characteristic (ROC) computed on 4000 test points for the representative case of $K = 200$, $P(y = +1) = 0.8$ and $\eta = 1$.

8.4. Comparison of Running Times: Classification

To assess the time complexity of binary classification tasks we considered the same setup as in the previous example. We take balanced randomly generated classification problems with different numbers K of training pairs with input patterns lying in $\mathbb{R}^{m\times m\times m}$. For all the cases we let $\mathcal{Z}$ (see previous example) be a tensor with multilinear ranks $(2, 2, 2)$ and let $\mu = 1$. We compare CMLE and A-CMLE and report the average CPU times over 20 such problems (standard deviation in parenthesis) in Table 4. For all the cases the convergence requires many tens of iterations and a considerable gain is achieved by the accelerated algorithm. As an example the average number of iterations for $m = 30$ and $K = 400$ is 250 for CMLE and 90 for A-CMLE.

Table 4: Running times of CMLE and A-CMLE for randomly generated classification problems. The results refer to problems where K input-output pairs were used for training and input patterns where taken in $\mathbb{R}^{m\times m\times m}$.

CPU times (in hundreds of sec.)

	CMLE	A-CMLE	CMLE	A-CMLE	CMLE	A-CMLE
K	m=5		m=10		m=20	
20	0.11(0.09)	**0.01(0.01)**	0.12(0.08)	**0.03(0.01)**	0.39(0.04)	**0.12(0.05)**
400	0.97(0.96)	**0.18(0.13)**	2.29(0.55)	**0.52(0.08)**	11.08(3.02)	**2.22(0.64)**
1600	1.11(1.02)	**0.29(0.17)**	12.40(10.91)	**2.93(1.38)**	38.31(83.43)	**4.08(8.37)**

8.5. Classification: Aerial View

To test the proposed approach in a real life setting we considered the dataset Aerial View Activity Classification [69]. The goal is to recognize different human actions from the given low-resolution grayscale videos, 12 per action. We considered the task of discriminating between different pairs of actions, namely waving (type I) versus waving (type II), carrying versus running, pointing versus standing and finally digging versus jumping. Each video is a 3−order tensor where the first two dimensions represent number

Table 5: Accuracy for the real-life examples of Section 8.5.

AUC performance

waving (I) vs waving (II)		carrying vs running	
nuc norm	lin LS-SVM	nuc norm	lin LS-SVM
0.83(**0.14**)	0.81(0.15)	**0.82**(**0.11**)	0.60 (0.16)
pointing vs standing		digging vs jumping	
nuc norm	lin LS-SVM	nuc norm	lin LS-SVM
0.98(**0.02**)	0.86(0.07)	**1**(**0**)	0.98(0.04)

of pixels of each frame and the third dimension is the number of frames, see Figure 5. As a preprocessing step we normalize the videos in the datasets. Each frame of each video is resampled to match the common size of 10×13 pixels. To cope with the different number of frames per video, we perform dimensionality reduction along the time mode and extract 4 eigen-images separately for all the videos. As a result of this normalization procedure for each binary classification task we are left with 24 $10 \times 13 \times 4$ input tensors and corresponding target labels. For each task we considered 8 tensors for training and the remaining 16 for testing. The performance of the tensor model obtained with nuclear norm regularization (nuc norm) was compared against LS-SVM classifiers with linear kernel (lin LS-SVM). The parameters were chosen according to the leave-one-out misclassification error for all the cases. Table 5 reports the average (standard deviation between parenthesis) Area Under the ROC Curve (AUC) computed according to the test videos for 20 random splits of the original set into training and test points.

9. Concluding Remarks

Inspired by recent advances on matrix problems, we have introduced in this work the concept of Shatten-$\{p, q\}$ norms. This class comprises the nuclear-p norms that in turn generalizes the notion of matrix nuclear norm. The use of the latter to define convex heuristics for matrix problems involving the rank function is largely motivated by Theorem 2. The results in Theorem 4 and later in Theorem 5 concerning the nuclear-1 norm and the multilinear ranks are weaker. In general, it remains a matter of future research to fully characterize the introduced class of norms, especially in relation to the

different notions of tensorial ranks. Here we focused on the nuclear-1 norm and developed a general regularization method that promotes solutions with small $n-$ranks. This idea mimics the concept of continuous shrinkage that underlies many convex statistical methods such as the LASSO. By contrast, the majority of the existing methodologies for tensors are non-convex and rely on hard multilinear rank constraints. From an algorithmical perspective the proposed methodology take advantage of a fundamental characteristic of the problem: its equivalence to a convex separable problem. Fully exploiting this feature leads to algorithms that are amenable to parallelization. Later on the general optimization problems (11) and (14)-(15) were specialized to cope with different tasks. Extending the existing taxonomy of learning to the case where input patterns are represented as tensors, we have called them unsupervised and supervised. We conclude by noticing that this generalization is instrumental to deal with important statistical learning aspects often overlooked in the tensor literature; these include but are not limited to issues like model selection, the choice of loss functions and the problem of out-of-sample extension.

Acknowledgements

Research supported by Research Council KUL: GOA Ambiorics, GOA MaNet, CoE EF/05/006 Optimization in Engineering(OPTEC), CIF1 and STRT1/08/023 IOF-SCORES4CHEM. Flemish Government: FWO: PhD/ postdoc grants, projects: G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0427.10N, G.0302.07 (SVM/Kernel), G.0588.09 (Brain-machine) research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); EU: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940).

References

[1] R. Coppi, S. Bolasco, Multiway data analysis, North-Holland Publishing Co. Amsterdam, The Netherlands, 1989.

[2] P. Parrilo, S. Khatri, On cone-invariant linear matrix inequalities, IEEE Transactions on Automatic Control 45 (2000) 1558–1563.

[3] P. M. Kroonenberg, J. De Leeuw, Principal component analysis of three-mode data by means of alternating least squares algorithms, Psychometrika 45 (1980) 69–97.

[4] L. De Lathauwer, B. De Moor, J. Vandewalle, On the best rank-1 and rank-(r_1, r_2,$\cdots$, r_n) approximation of higher order tensors, SIAM Journal on Matrix Analysis and Applications 21 (2000) 1324–1342.

[5] P. Kroonenberg, Applied multiway data analysis, Wiley-Interscience, 2008.

[6] M. Ishteva, L. De Lathauwer, P. A. Absil, S. Van Huffel, Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust region scheme, Submitted (17p.). Internal Report 09-142, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), Lirias number: 243104. (2009).

[7] M. Ishteva, P. Absil, S. Van Huffel, L. De Lathauwer, Tucker compression and local optima, Chemometr. Intell. Lab. Sys. Published online doi:10.1016/j.chemolab.2010.06.006. (2010).

[8] B. Savas, L. H. Lim, Best multilinear rank approximation of tensors with quasi-Newton methods on Grassmannians, Tech. Rep. LITH-MAT-R-2008–01-SE, Dept. Mathematics, Linköpings Universitet. (2008).

[9] L. Eldén, B. Savas, A Newton–Grassmann method for computing the best multilinear rank-(r_1, r_2, r_3) approximation of a tensor, SIAM J. Matrix Anal. Appl 31 (2009) 248–271.

[10] M. Ishteva, L. De Lathauwer, P. Absil, S. Van Huffel, Differential-geometric Newton algorithm for the best rank-(r_1, r_2, r_3) approximation of tensors, Numerical Algorithms 51 (2009) 179–194.

[11] K. Grigoriadis, E. Beran, Alternating projection algorithms for linear matrix inequalities problems with rank constraints, volume Advances in linear matrix inequality methods in control: advances in design and control, SIAM, pp. 251–267, 2000.

[12] R. Skelton, T. Iwasaki, K. Grigoriadis, A unified algebraic approach to linear control design, CRC, 1997.

[13] B. Recht, M. Fazel, P. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Rev. 52 (2007) 471–501.

[14] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) 58 (1996) 267–288.

[15] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM Review 43 (2001) 129–159.

[16] D. Donoho, Compressed Sensing, IEEE Transactions on Information Theory 52 (2006) 1289–1306.

[17] K. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems, Accepted for publication in Pacific Journal of Optimization (2009).

[18] Z. Liu, L. Vandenberghe, Interior-point method for nuclear norm approximation with application to system identification, SIAM Journal on Matrix Analysis and Applications 31 (2009) 1235–1256.

[19] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, in: IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan, pp. 8, 2009.

[20] D. Bertsekas, J. Tsitsiklis, Parallel and distributed computation, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[21] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM Journal on Matrix Analysis and Applications 21 (2000) 1253–1278.

[22] L. De Lathauwer, J. Vandewalle, Dimensionality reduction in higher-order signal processing and rank-(R_1, R_2,$\cdots$, R_N) reduction in multilinear algebra, Linear Algebra and its Applications 391 (2004) 31–55.

[23] B. Bader, T. Kolda, Algorithm 862: Matlab tensor classes for fast algorithm prototyping, ACM Trans. Math. Software 32 (2006).

[24] H. Kiers, Towards a standardized notation and terminology in multiway analysis, Journal of Chemometrics 14 (2000) 105–122.

[25] L. R. Tucker, The extension of factor analysis to three-dimensional matrices, volume Contributions to Mathematical Psychology, Holt, Rinehart Winston, NY, pp. 109–127, 1964.

[26] L. R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (1966) 279–311.

[27] M. Fazel, H. Hindi, S. Boyd, A rank minimization heuristic with application to minimum ordersystem approximation, in: Proceedings of the American Control Conference, 2001, volume 6, pp. 4734–4739.

[28] E. Candès, B. Recht, Exact matrix completion via convex optimization, Foundations of Computational Mathematics 9 (2009) 717–772.

[29] B. Recht, W. Xu, B. Hassibi, Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization, in: 47th IEEE Conference on Decision and Control (CDC), pp. 3065–3070, 2008.

[30] J. Hiriart-Urruty, C. Lemaréchal, Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods, volume 306 of *Grundlehren der Mathematischen Wissenschaften*, Springer-Verlag, 1993.

[31] C. Floudas, P. Pardalos, Encyclopedia of optimization, Kluwer academic publishers, 2001.

[32] M. Fazel, Matrix rank minimization with applications, Ph.D. thesis, Elec. Eng. Dept, Stanford University, 2002.

[33] R. Horn, C. Johnson, Matrix analysis, Cambridge University Press, Cambridge, 1985.

[34] J. Falk, Lagrange multipliers and nonconvex programs, SIAM Journal on Control 7 (1969) 534.

[35] F. Tardella, Existence and sum decomposition of vertex polyhedral convex envelopes, Optimization Letters 2 (2008) 363–375.

[36] S. Negahban, P. Ravikumar, M. Wainwright, B. Yu, A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers, in: Advances in Neural Information Processing Systems (NIPS), 2009, pp. 1348–1356.

[37] P. Zhao, G. Rocha, B. Yu, The composite absolute penalties family for grouped and hierarchical variable selection, Annals of Statistics 37 (2009) 3468–3497.

[38] R. Jenatton, J. Audibert, F. Bach, Structured variable selection with sparsity-inducing norms, Technical Report - http://arxiv.org/abs/0904.3523v1 (2009).

[39] Y. Nesterov, Introductory lectures on convex optimization: A basic course, Kluwer Academic Pub, 2003.

[40] C. Navasca, L. De Lathauwer, Low multilinear rank tensor approximation via semidefinite programming, in: Proceedings 17th European Signal Processing Conference (EUSIPCO), pp. 520–524, 2009.

[41] H. Zou, The adaptive Lasso and its oracle properties, Journal of the American Statistical Association 101 (2006) 1418–1429.

[42] L. Breiman, Heuristics of instability and stabilization in model selection, Annals of Statistics 24 (1996) 2350–2383.

[43] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Annals of Statistics 32 (2004) 407–499.

[44] K. Toh, M. Todd, R. Tutuncu, SDPT3—a Matlab software package for semidefinite programming, Optimization Methods and Software 11 (1999) 545–581.

[45] J. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optimization Methods and Software 11 (1999) 625–653.

[46] S. Ma, D. Goldfarb, L. Chen, Fixed point and Bregman iterative methods for matrix rank minimization, to appear in Mathematical Programming (2009).

[47] M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 1.21, http://cvxr.com/cvx, 2010.

[48] I. Necoara, J. A. K. Suykens, Application of a smoothing technique to decomposition in convex optimization, IEEE Transactions on Automatic Control 53 (2008) 2674–2679.

[49] I. Necoara, J. A. K. Suykens, Interior-Point Lagrangian Decomposition Method for Separable Convex Optimization, Journal of Optimization Theory and Applications 143 (2009) 567–588.

[50] L. Xiao, M. Johansson, S. Boyd, Simultaneous routing and resource allocation via dual decomposition, IEEE Transactions on Communications 52 (2004) 1136–1144.

[51] D. Palomar, M. Chiang, Alternative distributed algorithms for network utility maximization: Framework and applications, IEEE Transactions on Automatic Control 52 (2007) 2254–2269.

[52] A. Venkat, I. Hiskens, J. Rawlings, S. Wright, Distributed MPC strategies with application to power system automatic generation control, IEEE Transactions on Control Systems Technology 16 (2008) 1192–1206.

[53] R. Rockafellar, J. Roger, Scenarios and policy aggregation in optimization under uncertainty, Mathematics of Operations Research 16 (1991) 119–147.

[54] S. Ji, J. Ye, An accelerated gradient method for trace norm minimization, in: Proceedings of the 26th Annual International Conference on Machine Learning (ICML), pp. 457–464, 2009.

[55] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Unfinished working draft, Information Systems Laboratory, Department of Electrical Engineering, Stanford University (version of September 27, 2010).

[56] D. Luenberger, Linear and nonlinear programming, Addison-Wesley (1984).

[57] J. Cai, E. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization 20 (2008) 1956–1982.

[58] Y. Nesterov, Gradient methods for minimizing composite objective function, Center for Operations Research and Econometrics (CORE), Université catholique de Louvain, Tech. Rep. (2007).

[59] Y. Nesterov, A method of solving a convex programming problem with convergence rate $O(\frac{1}{k^2})$, in: Soviet Mathematics Doklady, volume 27, pp. 372–376, 1983.

[60] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences 2 (2009) 183–202.

[61] P. Tseng, On accelerated proximal gradient methods for convex-concave optimization, submitted to SIAM Journal on Optimization (2008).

[62] M. Bazaraa, H. Sherali, C. Shetty, Nonlinear programming: theory and algorithms, John Wiley and Sons, 2006.

[63] X. He, D. Cai, P. Niyogi, Tensor subspace analysis, in: Advances in Neural Information Processing Systems (NIPS), 2006, pp. 499–506.

[64] M. Vasilescu, D. Terzopoulos, Multilinear analysis of image ensembles: Tensor faces, in: 7th European Conference on Computer Vision (ECCV), in Computer Vision – ECCV 2002, Lecture Notes in Computer Science, volume 2350, pp. 447–460, 2002.

[65] T. Poggio, S. Smale, The mathematics of learning: Dealing with data, Notices of the American Mathematical Society (AMS) 50 (2003) 537–544.

[66] X. Zhu, A. Goldberg, Introduction to semi-supervised learning, Morgan Claypool Publishers, 2009.

[67] V. Vapnik, Statistical learning theory, Wiley, New York, 1998.

[68] L. Györfi, M. Kohler, A. Krzyżak, H. Walk, A distribution-free theory of nonparametric regression, Springer Series in Statistics, 2002.

[69] C. Chen, M. Ryoo, J. Aggarwal, UT-Tower Dataset: Aerial View Activity Classification Challenge, http://cvrc.ece.utexas.edu/SDHA2010/Aerial_View_Activity.html, 2010.

[70] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning: data mining, inference, and prediction, Springer, 2001.

[71] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least squares support vector machines, World Scientific, 2002.

[72] K. De Brabanter, P. Karsmakers, F. Ojeda, C. Alzate, J. De Brabanter, K. Pelckmans, B. De Moor, J. Vandewalle, J. A. K. Suykens, LS-SVMlab toolbox user's guide version 1.7, Internal Report 10-146, ESAT-SISTA, K.U.Leuven (Leuven, Belgium) (2010).

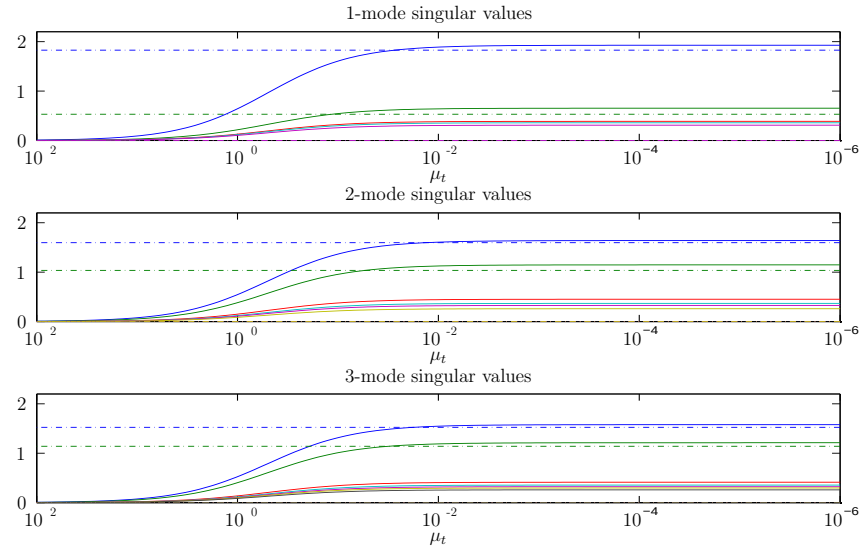
[73] A. Cichocki, R. Zdunek, S. Amari, Nonnegative matrix and tensor factorization, IEEE Signal Processing Magazine 25 (2008) 142–145.

[74] T. Kolda, B. Bader, Tensor decompositions and applications, SIAM Review 51 (2009) 455–500.

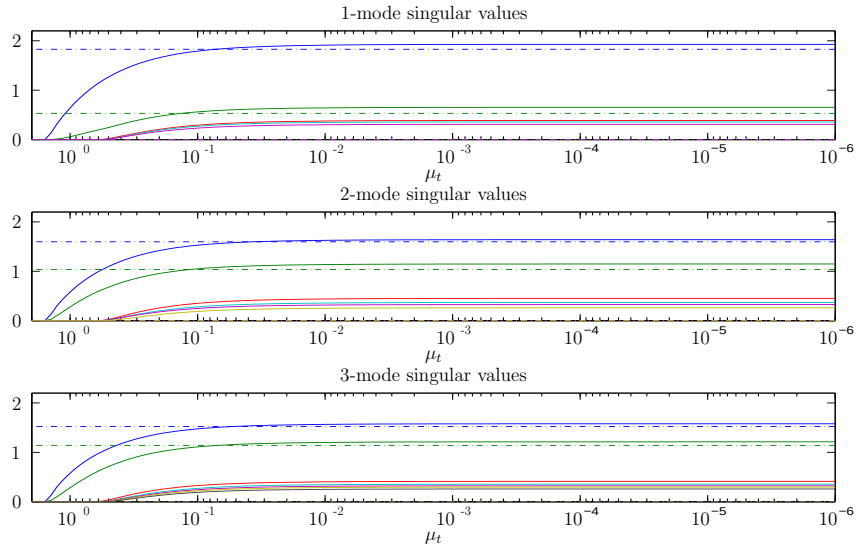
[75] A. Smilde, R. Bro, P. Geladi, Multi-way analysis with applications in the chemical sciences, Wiley, 2004.

[76] P. Comon, C. Jutten, Handbook of Blind Source Separation: Independent component analysis and applications, Academic Press, 2010.

[77] A. Argyriou, T. Evgeniou, M. Pontil, Convex multi-task feature learning, Machine Learning 73 (2008) 243–272.

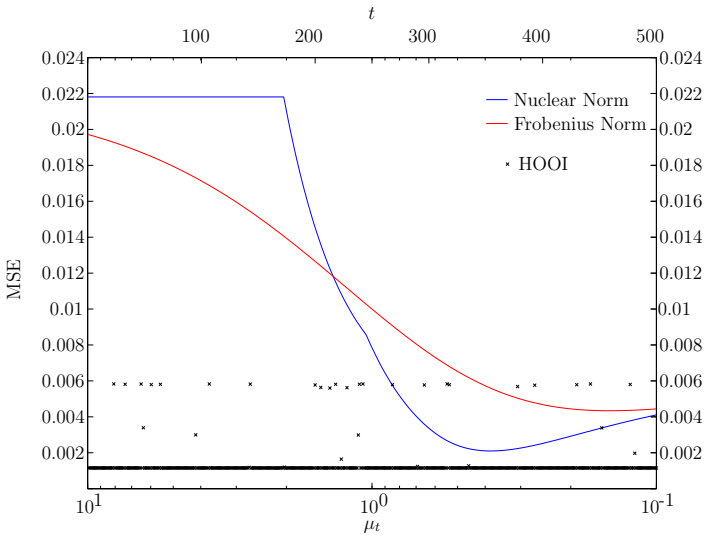


(a)

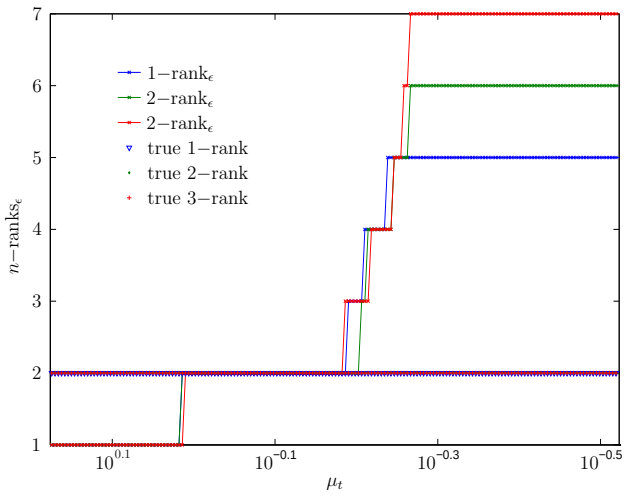


(b)

Figure 1: Tensor Denoising. The path for the $n-$mode singular values with squared Frobenius penalty (a) and with nuclear norm penalty (b). The dotted lines refer to the value of the $n-$mode singular values of $\mathcal{Z}^t$.



(a)



(b)

Figure 2: Tensor Denoising. (a) The mean square error (MSE) achieved with the two penalties along the solution paths together with the 500 solutions obtained via the HOOI algorithm with random initializations. (b) The $n-\text{ranks}_\epsilon$ of the solutions corresponding to the nuclear norm penalty for the case $\epsilon(\boldsymbol{A}) = \sigma_1(\boldsymbol{A})\ 10^{-2}$. The dotted lines refer to the true $n-$ranks. The $n-\text{ranks}_\epsilon$ of the solutions obtained with Frobenius norm are always maximal and are not reported.

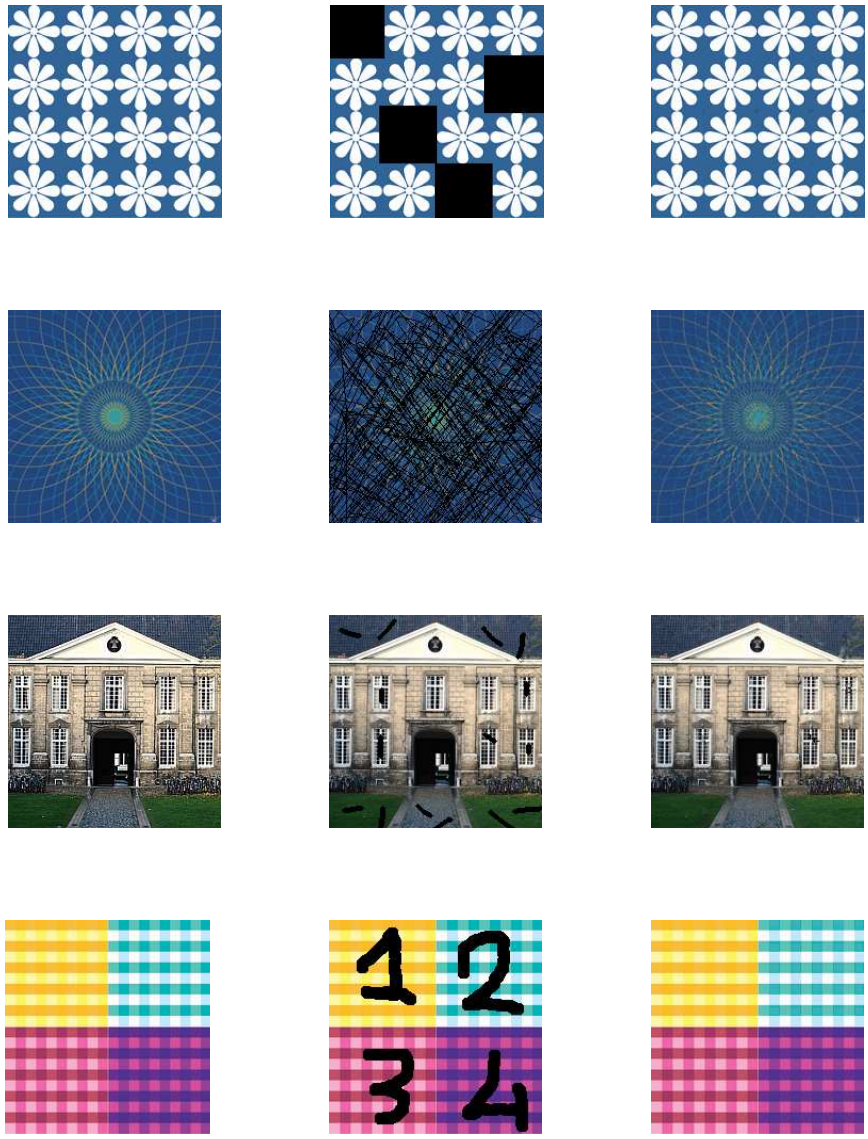


Figure 3: Original images (left column), the given pixels (middle column — black filled areas denote portions of the images that were removed) and finally the reconstruction by Algorithm 4 (right column).

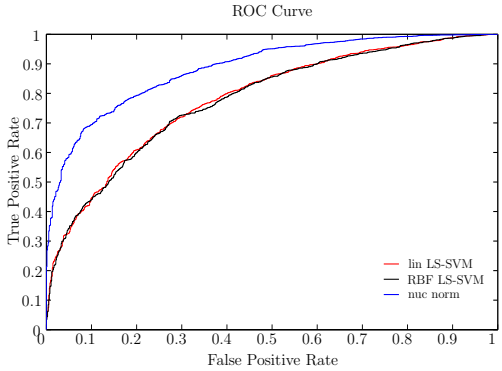


Figure 4: ROC curve for the example in Subsection 8.3. It was computed based upon 4000 test points for the representative case of $K = 200$, $P(y = +1) = 0.8$ and $\eta = 1$. Expectedly, since the generating mechanism is linear, a non-linear LS-SVM classifier with gaussian-RBF kernel does not improve over the linear LS-SVM classifier in this case. In turn, since the generating model is low-rank, the solution of a version of problem (42) that keeps into account a bias term largely outperforms the other models.

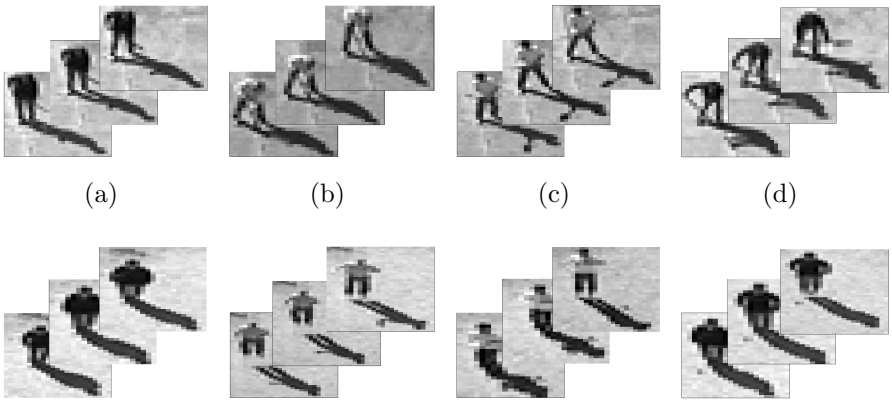


Figure 5: Before the preprocessing step each input point is a low-resolution grayscale video (a $3-$order tensor) of which we depict here the clipped version of the first three frames. In particular, the first row ((a)-(d)) refers to 4 points in the class *digging* whereas the second row ((e)-(h)) refers to 4 points in the class *jumping*.