Low-Rank Regression with Tensor Responses

Guillaume Rabusseau and Hachem Kadri
Aix Marseille Univ, CNRS, LIF, Marseille, France
{firstname.lastname}@lif.univ-mrs.fr

Abstract

This paper proposes an efficient algorithm (HOLRR) to handle regression tasks where the outputs have a tensor structure. We formulate the regression problem as the minimization of a least square criterion under a multilinear rank constraint, a difficult non convex problem. HOLRR computes efficiently an approximate solution of this problem, with solid theoretical guarantees. A kernel extension is also presented. Experiments on synthetic and real data show that HOLRR computes accurate solutions while being computationally very competitive.

1 Introduction

Recently, there has been an increasing interest in adapting machine learning and statistical methods to tensors. Data with a natural tensor structure are encountered in many scientific areas including neuroimaging [30], signal processing [4], spatio-temporal analysis [2] and computer vision [16]. Extending multivariate regression methods to tensors is one of the challenging task in this area. Most existing works extend linear models to the multilinear setting and focus on the tensor structure of the input data (e.g. [24]). Little has been done however to investigate learning methods for *tensor-structured output data*.

We consider a multilinear regression task where outputs are tensors; such a setting can occur in the context of e.g. spatio-temporal forecasting or image reconstruction. In order to leverage the tensor structure of the output data, we formulate the problem as the minimization of a least squares criterion subject to a *multilinear rank* constraint on the regression tensor. The rank constraint enforces the model to capture low-rank structure in the outputs and to explain dependencies between inputs and outputs in a low-dimensional multilinear subspace.

Unlike previous work (e.g. [22, 24, 27]) we do not rely on a convex relaxation of this difficult non-convex optimization problem. Instead we show that it is equivalent to a multilinear subspace identification problem for which we design a fast and efficient approximation algorithm (HOLRR), along with a kernelized version which extends our approach to the nonlinear setting (Section 3). Our theoretical analysis shows that HOLRR provides good approximation guarantees. Furthermore, we derive a generalization bound for the class of tensor-valued regression functions with bounded multilinear rank (Section 3.3). Experiments on synthetic and real data are presented to validate our theoretical findings and show that HOLRR computes accurate solutions while being computationally very competitive (Section 4).

Proofs of all results stated in the paper can be found in supplementary material A.

Related work. The problem we consider is a generalization of the reduced-rank regression problem (Section 2.2) to tensor structured responses. Reduced-rank regression has its roots in statistics [10] but it has also been investigated by the neural network community [3]; non-parametric extensions of this method have been proposed in [18] and [6]. In the context of multi-task learning, a linear model using a tensor-rank penalization of a least squares criterion has been proposed in [22] to take into account the multi-modal interactions between

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

tasks. They propose an approach relying on a convex relaxation of the multlinear rank constraint using the trace norms of the matricizations, and a non-convex approach based on alternating minimization. Nonparametric low-rank estimation strategies in reproducing kernel Hilbert spaces (RKHS) based on a multilinear spectral regularization have been proposed in [23, 24]. Their method is based on estimating the regression function in the tensor product of RKHSs and is naturally adapted for tensor covariates. A greedy algorithm to solve a low-rank tensor learning problem has been proposed in [2] in the context of multivariate spatio-temporal data analysis. The linear model they assume is different from the one we propose and is specifically designed for spatio-temporal data. A higher-order extension of partial least squares (HOPLS) has been proposed in [28] along with a kernel extension in [29]. While HOPLS has the advantage of taking the tensor structure of the input into account, the questions of approximation and generalization guarantees were not addressed in [28]. The generalization bound we provide is inspired from works on matrix and tensor completion [25, 19].

2 Preliminaries

We begin by introducing some notations. For any integer k we use $[k]$ to denote the set of integers from 1 to k. We use lower case bold letters for vectors (e.g. $\mathbf{v} \in \mathbb{R}^{d_1}$), upper case bold letters for matrices (e.g. $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$) and bold calligraphic letters for higher order tensors (e.g. $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$). The identity matrix will be written as $\mathbf{I}$. The ith row (resp. column) of a matrix $\mathbf{M}$ will be denoted by $\mathbf{M}_{i,:}$ (resp. $\mathbf{M}_{:,i}$). This notation is extended to slices of a tensor in the straightforward way. If $\mathbf{v} \in \mathbb{R}^{d_1}$ and $\mathbf{v}' \in \mathbb{R}^{d_2}$, we use $\mathbf{v} \otimes \mathbf{v}' \in \mathbb{R}^{d_1 \cdot d_2}$ to denote the Kronecker product between vectors, and its straightforward extension to matrices and tensors. Given a matrix $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$, we use $\mathrm{vec}(\mathbf{M}) \in \mathbb{R}^{d_1 \cdot d_2}$ to denote the column vector obtained by concatenating the columns of $\mathbf{M}$.

2.1 Tensors and Tucker Decomposition

We first recall basic definitions of tensor algebra; more details can be found in [13]. A *tensor* $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times \cdots \times d_p}$ can simply be seen as a multidimensional array $(\boldsymbol{\mathcal{T}}_{i_1,\cdots,i_p} : i_n \in [d_n], n \in [p])$. The *mode-n* fibers of $\boldsymbol{\mathcal{T}}$ are the vectors obtained by fixing all indices except the nth one, e.g. $\boldsymbol{\mathcal{T}}_{:,i_2,\cdots,i_p} \in \mathbb{R}^{d_1}$. The *nth mode matricization* of $\boldsymbol{\mathcal{T}}$ is the matrix having the mode-n fibers of $\boldsymbol{\mathcal{T}}$ for columns and is denoted by $\mathbf{T}_{(n)} \in \mathbb{R}^{d_n \times d_1 \cdots d_{n-1} d_{n+1} \cdots d_p}$. The vectorization of a tensor is defined by $\mathrm{vec}(\boldsymbol{\mathcal{T}}) = \mathrm{vec}(\mathbf{T}_{(1)})$. The *inner product* between two tensors $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{T}}$ (of the same size) is defined by $\langle \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}} \rangle = \langle \mathrm{vec}(\boldsymbol{\mathcal{S}}), \mathrm{vec}(\boldsymbol{\mathcal{T}}) \rangle$ and the Frobenius norm is defined by $\|\boldsymbol{\mathcal{T}}\|_F^2 = \langle \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{T}} \rangle$. In the following $\boldsymbol{\mathcal{T}}$ always denotes a tensor of size $d_1 \times \cdots \times d_p$.

The *mode-n matrix product* of the tensor $\boldsymbol{\mathcal{T}}$ and a matrix $\mathbf{X} \in \mathbb{R}^{m \times d_n}$ is a tensor denoted by $\boldsymbol{\mathcal{T}} \times_n \mathbf{X}$. It is of size $d_1 \times \cdots \times d_{n-1} \times m \times d_{n+1} \times \cdots \times d_p$ and is defined by the relation $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{T}} \times_n \mathbf{X} \Leftrightarrow \mathbf{Y}_{(n)} = \mathbf{X}\mathbf{T}_{(n)}$. The *mode-n vector product* of the tensor $\boldsymbol{\mathcal{T}}$ and a vector $\mathbf{v} \in \mathbb{R}^{d_n}$ is a tensor defined by $\boldsymbol{\mathcal{T}} \bullet_n \mathbf{v} = \boldsymbol{\mathcal{T}} \times_n \mathbf{v}^\top \in \mathbb{R}^{d_1 \times \cdots \times d_{n-1} \times d_{n+1} \times \cdots \times d_p}$. The *mode-n rank* of $\boldsymbol{\mathcal{T}}$ is the dimension of the space spanned by its mode-n fibers, that is $\mathrm{rank}_n(\boldsymbol{\mathcal{T}}) = \mathrm{rank}(\mathbf{T}_{(n)})$. The *multilinear rank* of $\boldsymbol{\mathcal{T}}$, denoted by $\mathrm{rank}(\boldsymbol{\mathcal{T}})$, is the tuple of mode-$n$ ranks of $\boldsymbol{\mathcal{T}}$: *rank*$(\boldsymbol{\mathcal{T}}) = (R_1, \cdots, R_p)$ where $R_n = \mathrm{rank}_n(\boldsymbol{\mathcal{T}})$ for $n \in [p]$. We will write $\mathrm{rank}(\boldsymbol{\mathcal{T}}) \leq (S_1, \cdots, S_p)$ whenever $\mathrm{rank}_1(\boldsymbol{\mathcal{T}}) \leq S_1, \mathrm{rank}_2(\boldsymbol{\mathcal{T}}) \leq S_2, \cdots, \mathrm{rank}_p(\boldsymbol{\mathcal{T}}) \leq S_p$.

The *Tucker decomposition* decomposes a tensor $\boldsymbol{\mathcal{T}}$ into a core tensor $\boldsymbol{\mathcal{G}}$ transformed by an orthogonal matrix along each mode: (*i*) $\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_p \mathbf{U}_p$, where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_p}$, $\mathbf{U}_i \in \mathbb{R}^{d_i \times R_i}$ and $\mathbf{U}_i^\top \mathbf{U}_i = \mathbf{I}$ for all $i \in [p]$. The number of parameters involved in a Tucker decomposition can be considerably smaller than $d_1 d_2 \cdots d_p$. We have the following identities when matricizing and vectorizing a Tucker decomposition: $\mathbf{T}_{(n)} = \mathbf{U}_n \mathbf{G}_{(n)} (\mathbf{U}_p \otimes \cdots \otimes \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \cdots \otimes \mathbf{U}_1)^\top$ and $\mathrm{vec}(\boldsymbol{\mathcal{T}}) = (\mathbf{U}_p \otimes \mathbf{U}_{p-1} \otimes \cdots \otimes \mathbf{U}_1)\mathrm{vec}(\boldsymbol{\mathcal{G}})$.

It is well known that $\boldsymbol{\mathcal{T}}$ admits the Tucker decomposition (i) iff $\mathrm{rank}(\boldsymbol{\mathcal{T}}) \leq (R_1, \cdots, R_p)$ (see e.g. [13]). Finding an exact Tucker decomposition can be done using the higher-order SVD algorithm (HOSVD) introduced by [5]. Although finding the best approximation of

multilinear rank $(R_1, \cdots, R_p)$ of a tensor $\boldsymbol{\mathcal{T}}$ is a difficult problem, the truncated HOSVD algorithm provides good approximation guarantees and often performs well in practice.

2.2 Low-Rank Regression

Multivariate regression is the task of recovering a function $f : \mathbb{R}^d \to \mathbb{R}^p$ from a set of input-output pairs $\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^N$ sampled from the model with an additive noise $\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is the error term. To solve this problem, the *ordinary least squares* (OLS) approach assumes a linear dependence between input and output data and boils down to finding a matrix $\mathbf{W} \in \mathbb{R}^{d\times p}$ that minimizes the squared error $\|\mathbf{XW} - \mathbf{Y}\|_F^2$, where $\mathbf{X} \in \mathbb{R}^{N\times d}$ and $\mathbf{Y} \in \mathbb{R}^{N\times p}$ denote the input and the output matrices. To prevent overfitting and to avoid numerical instabilities a ridge regularization term (i.e. $\gamma\|\mathbf{W}\|_F^2$) is often added to the objective function, leading to the *regularized least squares* (RLS) method. It is easy to see that the OLS/RLS approach in the multivariate setting is equivalent to performing p linear regressions for each scalar output $\{\mathbf{y}_j\}_{j=1}^p$ independently. Thus it performs poorly when the outputs are correlated and the true dimension of the response is less than p. *Low-rank regression* (or reduced-rank regression) addresses this issue by solving the rank penalized problem $\min_{\mathbf{W}\in\mathbb{R}^{d\times p}} \|\mathbf{XW} - \mathbf{Y}\|_F^2 + \gamma\|\mathbf{W}\|_F^2$ s.t. $\operatorname{rank}(\mathbf{W}) \leq R$ for a given integer R. The rank constraint was first proposed in [1], whereas the term *reduced-rank regression* was introduced in [10]. Adding a ridge regularization was proposed in [18]. In the rest of the paper we will refer to this approach as low-rank regression (LRR). For more description and discussion of reduced-rank regression, we refer the reader to the books [21] and [11].

3 Low-Rank Regression for Tensor-Valued Functions

3.1 Problem Formulation

We consider a multivariate regression task where the input is a vector and the response has a tensor structure. Let $f : \mathbb{R}^{d_0} \to \mathbb{R}^{d_1\times d_2\times\cdots\times d_p}$ be the function we want to learn from a sample of input-output data $\{(\mathbf{x}^{(n)}, \boldsymbol{\mathcal{Y}}^{(n)})\}_{n=1}^N$ drawn from the model $\boldsymbol{\mathcal{Y}} = f(\mathbf{x}) + \boldsymbol{\mathcal{E}}$, where $\boldsymbol{\mathcal{E}}$ is an error term. We assume that f is linear, that is $f(\mathbf{x}) = \boldsymbol{\mathcal{W}} \bullet_1 \mathbf{x}$ for some regression tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{d_0\times d_1\times\cdots\times d_p}$. The vectorization of this relation leads to $\operatorname{vec}(f(\mathbf{x})) = \mathbf{W}_{(1)}^\top \mathbf{x}$ showing that this model is equivalent to the standard multivariate linear model. One way to tackle this regression task would be to vectorize each output sample and to perform a standard low-rank regression on the data $\{(\mathbf{x}^{(n)}, \operatorname{vec}(\boldsymbol{\mathcal{Y}}^{(n)}))\}_{n=1}^N \subset \mathbb{R}^{d_0} \times \mathbb{R}^{d_1\cdots d_p}$. A major drawback of this approach is that the tensor structure of the output is lost in the vectorization step. The low-rank model tries to capture linear dependencies between components of the output but it ignores *higher level dependencies* that could be present in a tensor-structured output. For illustration, suppose the output is a matrix encoding the samples of d_1 continuous variables at d_2 different time steps, one could expect structural relations between the d_1 time series, e.g. linear dependencies between the rows of the output matrix.

Low-rank regression for tensor responses. To overcome the limitation described above we propose an extension of the low-rank regression method for tensor-structured responses by enforcing low multilinear rank of the regression tensor $\boldsymbol{\mathcal{W}}$. Let $\{(\mathbf{x}^{(n)}, \boldsymbol{\mathcal{Y}}^{(n)})\}_{n=1}^N \subset \mathbb{R}^{d_0} \times \mathbb{R}^{d_1\times d_2\times\cdots\times d_p}$ be a training sample of input/output data drawn from the model $f(\mathbf{x}) = \boldsymbol{\mathcal{W}} \bullet_1 \mathbf{x} + \boldsymbol{\mathcal{E}}$ where $\boldsymbol{\mathcal{W}}$ is assumed of low multilinear rank. Considering the framework of empirical risk minimization, we want to find a low-rank regression tensor $\boldsymbol{\mathcal{W}}$ minimizing the loss on the training data. To avoid numerical instabilities and to prevent overfitting we add a ridge regularization to the objective function, leading to the minimization of $\sum_{n=1}^N \ell(\boldsymbol{\mathcal{W}} \bullet_1 \mathbf{x}^{(n)}, \boldsymbol{\mathcal{Y}}^{(n)}) + \gamma\|\boldsymbol{\mathcal{W}}\|_F^2$ w.r.t. the regression tensor $\boldsymbol{\mathcal{W}}$ subject to the constraint $\operatorname{rank}(\boldsymbol{\mathcal{W}}) \leq (R_0, R_1, \cdots, R_p)$ for some given integers $R_0, R_1, \cdots, R_p$ and where ℓ is a loss function. In this paper, we consider the squared error loss between tensors defined by $\mathcal{L}(\boldsymbol{\mathcal{T}}, \hat{\boldsymbol{\mathcal{T}}}) = \|\boldsymbol{\mathcal{T}} - \hat{\boldsymbol{\mathcal{T}}}\|_F^2$. Using this loss we can rewrite the minimization problem as

$$\min_{\boldsymbol{\mathcal{W}}\in\mathbb{R}^{d_0\times d_1\times\cdots\times d_p}} \|\boldsymbol{\mathcal{W}} \times_1 \mathbf{X} - \boldsymbol{\mathcal{Y}}\|_F^2 + \gamma\|\boldsymbol{\mathcal{W}}\|_F^2 \quad \text{s.t. } \operatorname{rank}(\boldsymbol{\mathcal{W}}) \leq (R_0, R_1, \cdots, R_p), \tag{1}$$

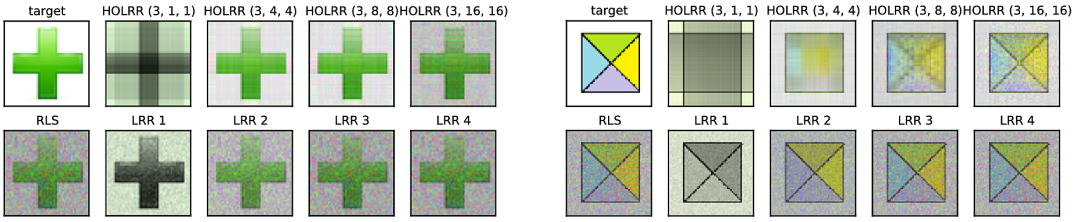

Figure 1: Image reconstruction from noisy measurements: $\mathcal{Y} = \mathcal{W} \bullet_1 \mathbf{x} + \mathcal{E}$ where $\mathcal{W}$ is a color image (RGB). Each image is labeled with the algorithm and the rank parameter.

where the input matrix $\mathbf{X} \in \mathbb{R}^{N \times d_0}$ and the output tensor $\mathcal{Y} \in \mathbb{R}^{N \times d_1 \times \cdots \times d_p}$ are defined by $\mathbf{X}_{n,:} = (\mathbf{x}^{(n)})^\top$, $\mathcal{Y}_{n,:,\cdots,:} = \mathcal{Y}^{(n)}$ for $n = 1, \cdots, N$ ($\mathcal{Y}$ is the tensor obtained by stacking the output tensors along the first mode).

Low-rank regression function. Let $\mathcal{W}^*$ be a solution of problem (1), it follows from the multilinear rank constraint that $\mathcal{W}^* = \mathcal{G} \times_1 \mathbf{U}_0 \times_2 \cdots \times_{p+1} \mathbf{U}_p$ for some core tensor $\mathcal{G} \in \mathbb{R}^{R_0 \times \cdots \times R_p}$ and orthogonal matrices $\mathbf{U}_i \in \mathbb{R}^{d_i \times R_i}$ for $0 \leq i \leq p$. The regression function $f^* : \mathbf{x} \mapsto \mathcal{W}^* \bullet_1 \mathbf{x}$ can thus be written as $f^* : \mathbf{x} \mapsto \mathcal{G} \times_1 \mathbf{x}^\top \mathbf{U}_0 \times_2 \cdots \times_{p+1} \mathbf{U}_p$.

This implies several interesting properties. First, for any $\mathbf{x} \in \mathbb{R}^{d_0}$ we have $f^*(\mathbf{x}) = \mathcal{T}_\mathbf{x} \times_1 \mathbf{U}_1 \times_2 \cdots \times_p \mathbf{U}_p$ with $\mathcal{T}_\mathbf{x} = \mathcal{G} \bullet_1 \mathbf{U}_0^\top \mathbf{x}$, which implies $\text{rank}(f^*(\mathbf{x})) \leq (R_1, \cdots, R_p)$, that is the image of f^* is a set of tensors with low multilinear rank. Second, the relation between $\mathbf{x}$ and $\mathcal{Y} = f^*(\mathbf{x})$ is explained in a low dimensional subspace of size $R_0 \times R_1 \times \cdots \times R_p$. Indeed one can decompose the mapping f^* into the following steps: (i) project $\mathbf{x}$ in $\mathbb{R}^{R_0}$ as $\bar{\mathbf{x}} = \mathbf{U}_0^\top \mathbf{x}$, (ii) perform a low-dimensional mapping $\bar{\mathcal{Y}} = \mathcal{G} \bullet_1 \bar{\mathbf{x}}$, (iii) project back into the output space to get $\mathcal{Y} = \bar{\mathcal{Y}} \times_1 \mathbf{U}_1 \times_2 \cdots \times_p \mathbf{U}_p$.

To give an illustrative intuition on the differences between matrix and multilinear rank regularization we present a simple experiment[1] in Figure 1. We generate data from the model $\mathcal{Y} = \mathcal{W} \bullet_1 \mathbf{x} + \mathcal{E}$ where the tensor $\mathcal{W} \in \mathbb{R}^{3 \times m \times n}$ is a color image of size $m \times n$ encoded with three color channels RGB. The components of both $\mathbf{x}$ and $\mathcal{E}$ are drawn from $\mathcal{N}(0, 1)$. This experiment allows us to visualize the tensors returned by RLS, LRR and our method HOLRR that enforces low multilinear rank of the regression function. First, this shows that the function learned by vectorizing the outputs and performing LRR does not enforce any low-rank structure. This is well illustrated in (Figure 1) where the regression tensors returned by HOLRR-(3,1,1) are clearly of low-rank while the ones returned by LRR-1 are not. This also shows that taking into account the low-rank structure of the model allows one to better eliminate the noise when the true regression tensor is of low rank (Figure 1, left). However if the ground truth model does not have a low-rank structure, enforcing low mutlilinear rank leads to underfitting for low values of the rank parameter (Figure 1, right).

3.2 Higher-Order Low-Rank Regression and its Kernel Extension

We now propose an efficient algorithm to tackle problem (1). We first show that the ridge regularization term in (1) can be incorporated in the data fitting term. Let $\tilde{\mathbf{X}} \in \mathbb{R}^{(N+d_0) \times d_0}$ and $\tilde{\mathcal{Y}} \in \mathbb{R}^{(N+d_0) \times d_1 \times \cdots \times d_p}$ be defined by $\tilde{\mathbf{X}}^\top = (\mathbf{X} \mid \gamma \mathbf{I})^\top$ and $\tilde{\mathbf{Y}}_{(1)}^\top = \left(\mathbf{Y}_{(1)} \mid \mathbf{0}\right)^\top$. It is easy to check that the objective function in (1) is equal to $\|\mathcal{W} \times_1 \tilde{\mathbf{X}} - \tilde{\mathcal{Y}}\|_F^2$. Minimization problem (1) is then equivalent to

$$\min_{\substack{\mathcal{G} \in \mathbb{R}^{R_0 \times R_1 \times \cdots \times R_p}, \\ \mathbf{U}_i \in \mathbb{R}^{d_i \times R_i} \text{ for } 0 \leq i \leq p}} \|\mathcal{W} \times_1 \tilde{\mathbf{X}} - \tilde{\mathcal{Y}}\|_F^2 \quad \text{s.t. } \mathcal{W} = \mathcal{G} \times_1 \mathbf{U}_0 \cdots \times_{p+1} \mathbf{U}_p, \mathbf{U}_i^\top \mathbf{U}_i = \mathbf{I} \text{ for all } i. \quad (2)$$

We now show that this minimization problem can be reduced to finding $p + 1$ projection matrices onto subspaces of dimension $R_0, R_1, \cdots, R_p$. We start by showing that the core tensor $\mathcal{G}$ solution of (2) is determined by the factor matrices $\mathbf{U}_0, \cdots, \mathbf{U}_p$.

[1] An extended version of this experiment is presented in supplementary material B.

Theorem 1. *For given orthogonal matrices* $\mathbf{U}_0, \cdots, \mathbf{U}_p$ *the tensor* $\mathcal{G}$ *that minimizes (2) is given by* $\mathcal{G} = \tilde{\mathcal{Y}} \times_1 (\mathbf{U}_0^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \mathbf{U}_0)^{-1} \mathbf{U}_0^\top \tilde{\mathbf{X}}^\top \times_2 \mathbf{U}_1^\top \times_3 \cdots \times_{p+1} \mathbf{U}_p^\top$.

It follows from Theorem 1 that problem (1) can be written as

$$\min_{\mathbf{U}_i \in \mathbb{R}^{d_i \times R_i}, 0 \leq i \leq p} \|\tilde{\mathcal{Y}} \times_1 \mathbf{\Pi}_0 \times_2 \cdots \times_{p+1} \mathbf{\Pi}_p - \tilde{\mathcal{Y}}\|_F^2 \tag{3}$$

subject to $\mathbf{U}_i^\top \mathbf{U}_i = \mathbf{I}$ for all i, $\mathbf{\Pi}_0 = \tilde{\mathbf{X}}\mathbf{U}_0 \left(\mathbf{U}_0^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \mathbf{U}_0\right)^{-1} \mathbf{U}_0^\top \tilde{\mathbf{X}}^T$, $\mathbf{\Pi}_i = \mathbf{U}_i \mathbf{U}_i^\top$ for $i \geq 1$. Note that $\mathbf{\Pi}_0$ is the orthogonal projection onto the space spanned by the columns of $\tilde{\mathbf{X}}\mathbf{U}_0$ and $\mathbf{\Pi}_i$ is the orthogonal projection onto the column space of $\mathbf{U}_i$ for $i \geq 1$. Hence solving problem (1) is equivalent to finding $p+1$ low-dimensional subspaces $U_0, \cdots, U_p$ such that projecting $\tilde{\mathcal{Y}}$ onto the spaces $\tilde{\mathbf{X}}U_0, U_1, \cdots, U_p$ along the corresponding modes is close to $\tilde{\mathcal{Y}}$.

HOLRR algorithm. Since solving problem (3) for the $p+1$ projections simultaneously is a difficult non-convex optimization problem we propose to solve it independently for each projection. This approach has the benefits of both being computationally efficient and providing good theoretical approximation guarantees (see Theorem 2). The following proposition gives the analytic solutions of (3) when each projection is considered independently.

Proposition 1. *For* $0 \leq i \leq p$*, using the definition of* $\mathbf{\Pi}_i$ *in (3), the optimal solution of* $\min_{\mathbf{U}_i \in \mathbb{R}^{d_i \times R_i}} \|\tilde{\mathcal{Y}} \times_{i+1} \mathbf{\Pi}_i - \tilde{\mathcal{Y}}\|_F^2$ *s.t.* $\mathbf{U}_i^\top \mathbf{U}_i = \mathbf{I}$ *is given by the top* R_i *eigenvectors of* $(\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^\top \tilde{\mathbf{Y}}_{(1)} \tilde{\mathbf{Y}}_{(1)}^\top \tilde{\mathbf{X}}$ *if* $i = 0$ *and* $\tilde{\mathbf{Y}}_{(i+1)} \tilde{\mathbf{Y}}_{(i+1)}^\top$ *otherwise.*

The results from Theorem 1 and Proposition 1 can be rewritten in terms of the original input matrix $\mathbf{X}$ and output tensor $\mathcal{Y}$ using the identities $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} = \mathbf{X}^\top \mathbf{X} + \gamma \mathbf{I}$, $\tilde{\mathcal{Y}} \times_1 \tilde{\mathbf{X}}^\top = \mathcal{Y} \times_1 \mathbf{X}^\top$ and $\tilde{\mathbf{Y}}_{(i)} \tilde{\mathbf{Y}}_{(i)}^\top = \mathbf{Y}_{(i)} \mathbf{Y}_{(i)}^\top$ for any $i \geq 1$. The overall Higher-Order Low-Rank Regression procedure (HOLRR) is summarized in Algorithm 1. Note that the Tucker decomposition of the solution returned by HOLRR could be a good initialization point for an Alternative Least Square method. However, studying the theoretical and experimental properties of this approach is beyond the scope of this paper and is left for future work.

HOLRR Kernel Extension We now design a kernelized version of the HOLRR algorithm by analyzing how it would be instantiated in a feature space. We show that all the steps involved can be performed using the Gram matrix of the input data without having to explicitly compute the feature map. Let $\phi : \mathbb{R}^{d_0} \to \mathbb{R}^L$ be a feature map and let $\mathbf{\Phi} \in \mathbb{R}^{N \times L}$ be the matrix with rows $\phi(\mathbf{x}^{(n)})^\top$ for $n \in [N]$. The higher-order low-rank regression problem in the feature space boils down to the minimization problem

$$\min_{\mathcal{W} \in \mathbb{R}^{L \times d_1 \times \cdots \times d_p}} \|\mathcal{W} \times_1 \mathbf{\Phi} - \mathcal{Y}\|_F^2 + \gamma \|\mathcal{W}\|_F^2 \qquad \text{s.t. } \operatorname{rank}(\mathcal{W}) \leq (R_0, R_1, \cdots, R_p) \ . \tag{4}$$

Following the HOLRR algorithm, one needs to compute the top R_0 eigenvectors of the $L \times L$ matrix $(\mathbf{\Phi}^\top \mathbf{\Phi} + \gamma \mathbf{I})^{-1} \mathbf{\Phi}^\top \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \mathbf{\Phi}$. The following proposition shows that this can be done using the Gram matrix $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top$ without explicitly knowing the feature map ϕ.

Proposition 2. *If* $\boldsymbol{\alpha} \in \mathbb{R}^N$ *is an eigenvector with eigenvalue* λ *of the matrix* $(\mathbf{K} + \gamma \mathbf{I})^{-1} \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \mathbf{K}$*, then* $\mathbf{v} = \mathbf{\Phi}^\top \boldsymbol{\alpha} \in \mathbb{R}^L$ *is an eigenvector with eigenvalue* λ *of the matrix* $(\mathbf{\Phi}^\top \mathbf{\Phi} + \gamma \mathbf{I})^{-1} \mathbf{\Phi}^\top \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \mathbf{\Phi}$.

Let $\mathbf{A}$ be the top R_0 eigenvectors of the matrix $(\mathbf{K} + \gamma \mathbf{I})^{-1} \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \mathbf{K}$. When working with the feature map ϕ, it follows from the previous proposition that line 1 in Algorithm 1 is equivalent to choosing $\mathbf{U}_0 = \mathbf{\Phi}^\top \mathbf{A} \in \mathbb{R}^{L \times R_0}$, while the updates in line 3 stay the same. The regression tensor $\mathcal{W} \in \mathbb{R}^{L \times d_1 \times \cdots \times d_p}$ returned by this algorithm is then equal to $\mathcal{W} = \mathcal{Y} \times_1 \mathbf{P} \times_2 \mathbf{U}_1 \mathbf{U}_1^\top \times_2 \cdots \times_{p+1} \mathbf{U}_p \mathbf{U}_p^\top$, where $\mathbf{P} = \mathbf{\Phi}^\top \mathbf{A} \left(\mathbf{A}^\top \mathbf{\Phi} (\mathbf{\Phi}^\top \mathbf{\Phi} + \gamma \mathbf{I}) \mathbf{\Phi}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{\Phi} \mathbf{\Phi}^\top$. It is easy to check that $\mathbf{P}$ can be rewritten as $\mathbf{P} = \mathbf{\Phi}^\top \mathbf{A} \left(\mathbf{A}^\top \mathbf{K} (\mathbf{K} + \gamma \mathbf{I}) \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{K}$.

Suppose now that the feature map ϕ is induced by a kernel $k : \mathbb{R}^{d_0} \times \mathbb{R}^{d_0} \to \mathbb{R}$. The prediction for an input vector $\mathbf{x}$ is then given by $\mathcal{W} \bullet_1 \mathbf{x} = \mathcal{C} \bullet_1 \mathbf{k}_\mathbf{x}$ where the nth component

Algorithm 1 `HOLRR`

Input: $\mathbf{X} \in \mathbb{R}^{N \times d_0}$, $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N \times d_1 \times \cdots \times d_p}$, rank $(R_0, R_1, \cdots, R_p)$ and regularization parameter γ.

$\mathbf{U}_0 \leftarrow$ top R_0 eigenvectors of $(\mathbf{X}^\top \mathbf{X} + \gamma \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \mathbf{X}$
for $i = 1$ **to** p **do**
  $\mathbf{U}_i \leftarrow$ top R_i eigenvec. of $\mathbf{Y}_{(i+1)} \mathbf{Y}_{(i+1)}^\top$
end for
$\mathbf{M} = \left(\mathbf{U}_0^\top (\mathbf{X}^\top \mathbf{X} + \gamma \mathbf{I}) \mathbf{U}_0\right)^{-1} \mathbf{U}_0^\top \mathbf{X}^\top$
$\boldsymbol{\mathcal{G}} \leftarrow \boldsymbol{\mathcal{Y}} \times_1 \mathbf{M} \times_2 \mathbf{U}_1^\top \times_3 \cdots \times_{p+1} \mathbf{U}_p^\top$
return $\boldsymbol{\mathcal{G}} \times_1 \mathbf{U}_0 \times_2 \cdots \times_{p+1} \mathbf{U}_p$

Algorithm 2 `Kernelized HOLRR`

Input: Gram matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$, $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N \times d_1 \times \cdots \times d_p}$, rank $(R_0, R_1, \cdots, R_p)$ and regularization parameter γ.

$\mathbf{A} \leftarrow$ top R_0 eigenvectors of $(\mathbf{K} + \gamma \mathbf{I})^{-1} \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \mathbf{K}$
for $i = 1$ **to** p **do**
  $\mathbf{U}_i \leftarrow$ top R_i eigenvec. of $\mathbf{Y}_{(i+1)} \mathbf{Y}_{(i+1)}^\top$
end for
$\mathbf{M} \leftarrow \left(\mathbf{A}^\top \mathbf{K} (\mathbf{K} + \gamma \mathbf{I}) \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{K}$
$\boldsymbol{\mathcal{G}} \leftarrow \boldsymbol{\mathcal{Y}} \times_1 \mathbf{M} \times_2 \mathbf{U}_1^\top \times_3 \cdots \times_{p+1} \mathbf{U}_p^\top$
return $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{G}} \times_1 \mathbf{A} \times_2 \mathbf{U}_1 \times_3 \cdots \times_{p+1} \mathbf{U}_p$

of $\mathbf{k}_\mathbf{x} \in \mathbb{R}^N$ is $\langle \phi(\mathbf{x}^{(n)}), \phi(\mathbf{x}) \rangle = k(\mathbf{x}^{(n)}, \mathbf{x})$ and the tensor $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{N \times d_1 \times \cdots \times d_p}$ is defined by $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{G}} \times_1 \mathbf{A} \times_2 \mathbf{U}_1 \times_2 \cdots \times_{p+1} \mathbf{U}_p$, with $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{Y}} \times_1 \left(\mathbf{A}^\top \mathbf{K} (\mathbf{K} + \gamma \mathbf{I}) \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{K} \times_2 \mathbf{U}_2^\top \times_3 \cdots \times_{p+1} \mathbf{U}_p$. Note that $\boldsymbol{\mathcal{C}}$ has multilinear rank $(R_0, \cdots, R_p)$, hence the low mutlilinear rank constraint on $\boldsymbol{\mathcal{W}}$ in the feature space translates into the low rank structure of the coefficient tensor $\boldsymbol{\mathcal{C}}$.

Let $\mathcal{H}$ be the reproducing kernel Hilbert space associated with the kernel k. The overall procedure for kernelized HOLRR is summarized in Algorithm 2. This algorithm returns the tensor $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{N \times d_1 \times \cdots \times d_p}$ defining the regression function $f : \mathbf{x} \mapsto \boldsymbol{\mathcal{C}} \bullet_1 \mathbf{k}_\mathbf{x} = \sum_{n=1}^N k(\mathbf{x}, \mathbf{x}^{(n)}) \boldsymbol{\mathcal{C}}^{(n)}$, where $\boldsymbol{\mathcal{C}}^{(n)} = \boldsymbol{\mathcal{C}}_{n:\cdots:} \in \mathbb{R}^{d_1 \times \cdots \times d_p}$.

3.3 Theoretical Analysis

Complexity analysis. HOLRR is a polynomial time algorithm, more precisely it has a time complexity in $\mathcal{O}((d_0)^3 + N((d_0)^2 + d_0 d_1 \cdots d_p) + \max_{i \geq 0} R_i (d_i)^2 + N d_1 \cdots d_p \max_{i \geq 1} d_i)$. In comparison, LRR has a time complexity in $\mathcal{O}((d_0)^3 + N((d_0)^2 + d_0 d_1 \cdots d_p) + (N + R)(d_1 \cdots d_p)^2)$. Since the complexity of HOLRR only have a linear dependence on the product of the output dimensions instead of a quadratic one for LRR, we can conclude that HOLRR will be more efficient than LRR when the output dimensions $d_1, \cdots, d_p$ are large. It is worth mentioning that the method proposed in [22] to solve a convex relaxation of problem 2 is an iterative algorithm that needs to compute SVDs of matrices of size $d_i \times d_1 \cdots d_{i-1} d_{i+1} \cdots d_p$ for each $0 \leq i \leq p$ *at each iteration*, it is thus computationally more expensive than HOLRR. Moreover, since HOLRR only relies on simple linear algebra tools, readily available methods could be used to further improve the speed of the algorithm, e.g. randomized-SVD [8] and random feature approximation of the kernel function [12, 20].

Approximation guarantees. It is easy to check that problem (1) is NP-hard since it generalizes the problem of fitting a Tucker decomposition [9]. The following theorem shows that HOLRR is a $(p+1)$-approximation algorithm for this problem. This result generalizes the approximation guarantees provided by the truncated HOSVD algorithm for the problem of finding the best low multilinear rank approximation of an arbitrary tensor.

Theorem 2. *Let $\boldsymbol{\mathcal{W}}^*$ be a solution of problem (1) and let $\boldsymbol{\mathcal{W}}$ be the regression tensor returned by Algorithm 1. If $\mathcal{L} : \mathbb{R}^{d_0 \times \cdots \times d_p} \to \mathbb{R}$ denotes the objective function of (1) w.r.t. $\boldsymbol{\mathcal{W}}$ then $\mathcal{L}(\boldsymbol{\mathcal{W}}) \leq (p+1) \mathcal{L}(\boldsymbol{\mathcal{W}}^*)$.*

Generalization Bound. The following theorem gives an upper bound on the excess-risk for the function class $\mathcal{F} = \{\mathbf{x} \mapsto \boldsymbol{\mathcal{W}} \bullet_1 \mathbf{x} \ : \ \text{rank}(\boldsymbol{\mathcal{W}}) \leq (R_0, \cdots, R_p)\}$ of tensor-valued regression functions with bounded multilinear rank. Recall that the expected loss of an hypothesis $h \in \mathcal{F}$ w.r.t. the target function f^* is defined by $R(h) = \mathbb{E}_\mathbf{x}[\mathcal{L}(h(\mathbf{x}), f^*(\mathbf{x}))]$ and its empirical loss by $\hat{R}(h) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(h(\mathbf{x}^{(n)}), f^*(\mathbf{x}^{(n)}))$.

Theorem 3. *Let $\mathcal{L} : \mathbb{R}^{d_1 \times \cdots \times d_p} \to \mathbb{R}$ be a loss function satisfying $\mathcal{L}(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}) = \frac{1}{d_1 \cdots d_p} \sum_{i_1, \cdots, i_p} \ell(\boldsymbol{\mathcal{A}}_{i_1, \cdots, i_p}, \boldsymbol{\mathcal{B}}_{i_1, \cdots, i_p})$ for some loss-function $\ell : \mathbb{R} \to \mathbb{R}^+$ bounded by M. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample of size N, the follow-*

ing inequality holds for all $h \in \mathcal{F}$*:* $R(h) \leq \hat{R}(h) + M\sqrt{2D\log\left(\frac{4e(p+2)d_0d_1\cdots d_p}{\max_{i\geq 0} d_i}\right)\log(N)/N} + M\sqrt{\log\left(\frac{1}{\delta}\right)/(2N)}$*, where* $D = R_0R_1\cdots R_p + \sum_{i=0}^{p} R_id_i$*.*

Proof. (Sketch) The complete proof is given in the supplementary material. It relies on bounding the pseudo-dimension of the class of real-valued functions $\tilde{\mathcal{F}} = \left\{(\mathbf{x}, i_1, \cdots, i_p) \mapsto (\boldsymbol{\mathcal{W}} \bullet_1 \mathbf{x})_{i_1,\cdots,i_p} \ : \ \text{rank}(\boldsymbol{\mathcal{W}}) = (R_0, \cdots, R_p)\right\}$. We show that the pseudo-dimension of $\tilde{\mathcal{F}}$ is upper bounded by $(R_0R_1\cdots R_p + \sum_{i=0}^{p} R_id_i)\log\left(\frac{4e(p+2)d_0d_1\cdots d_p}{\max_{i\geq 0} d_i}\right)$. This is done by leveraging the following result originally due to [26]: the number of sign patterns of r polynomials, each of degree at most d, over q variables is at most $(4edr/q)^q$ for all $r > q > 2$ [25, Theorem 2]. The rest of the proof consists in showing that the risk (resp. empirical risk) of hypothesis in $\mathcal{F}$ and $\tilde{\mathcal{F}}$ are closely related and invoking standard error generalization bounds in terms of the pseudo-dimension [17, Theorem 10.6]. □

Note that generalization bounds based on the pseudo-dimension for multivariate regression without low-rank constraint would involve a term in $\mathcal{O}(\sqrt{d_0d_1\cdots d_p})$. In contrast, the bound from the previous theorem only depends on the product of the output dimensions in a term bounded by $\mathcal{O}(\sqrt{\log(d_1\cdots d_p)})$. In some sense, taking into account the low mutlilinear rank of the hypothesis allows us to significantly reduce the dependence on the output dimensions from $\mathcal{O}(\sqrt{d_0\cdots d_p})$ to $\mathcal{O}(\sqrt{(R_0\cdots R_p + \sum_i R_id_i)(\sum_i \log(d_i))})$.

4 Experiments

In this section, we evaluate HOLRR on both synthetic and real-world datasets. Our experimental results are for tensor-structured output regression problems on which we report root mean-squared errors (RMSE) averaged across all the outputs. We compare HOLLR with the following methods: regularized least squares **RLS**, low-rank regression **LRR** described in Section 2.2, a multilinear approach based on tensor trace norm regularization **ADMM** [7, 22], a nonconvex multilinear multitask learning approach **MLMT-NC** [22], an higher order extension of partial least squares **HOPLS** [28] and the greedy tensor approach for multivariate spatio-temporal analysis **Greedy** [2].

For experiments with kernel algorithms we use the readily available kernelized RLS and the LRR kernel extension proposed in [18]. Note that ADMM, MLMT-NC and Greedy only consider a linear dependency between inputs and outputs. The greedy tensor algorithm proposed in [2] is developed specially for spatio-temporal data and the implementation provided by the authors is restricted to third-order tensors. Although MLMLT-NC is perhaps the closest algorithm to ours, we applied it only to simulated data. This is because MLMLT-NC is computationally very expensive and becomes intractable for large data sets. Average running times are reported in supplementary material B.

4.1 Synthetic Data

We generate both linear and nonlinear data. Linear data is drawn from the model $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{W}} \bullet_1 \mathbf{x} + \boldsymbol{\mathcal{E}}$ where $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{10\times10\times10\times10}$ is a tensor of multilinear rank $(6, 4, 4, 8)$ drawn at random, $\mathbf{x} \in \mathbb{R}^{10}$ is drawn from $\mathcal{N}(0, \mathbf{I})$, and each component of the error tensor $\boldsymbol{\mathcal{E}}$ is drawn from $\mathcal{N}(0, 0.1)$. Nonlinear data is drawn from $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{W}} \bullet_1 (\mathbf{x}\otimes\mathbf{x}) + \boldsymbol{\mathcal{E}}$ where $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{25\times10\times10\times10}$ is of rank $(5, 6, 4, 2)$ and $\mathbf{x} \in \mathbb{R}^5$ and $\boldsymbol{\mathcal{E}}$ are generated as above. Hyper-parameters for all algorithms are selected using 3-fold cross-validation on the training data.

These experiments have been carried out for different sizes of the training data set, 20 trials have been executed for each size. The average RMSEs on a test set of size 100 for the 20 trials are reported in Figure 2. We see that HOLRR algorithm clearly outperforms the other methods on the linear data. MLMT-NC method achieved the second best performance, it is however much more computationally expensive (see Table 1 in supplementary material B). On the nonlinear data LRR achieves good performances but HOLRR is still significantly more accurate, especially with small training datasets.

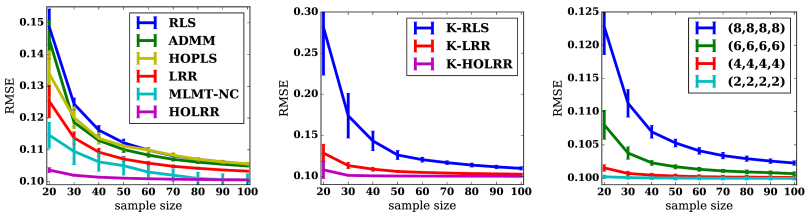

Figure 2: Average RMSE as a function of the training set size: (left) linear data, (middle) nonlinear data, (right) for different values of the rank parameter.

Table 1: RMSE on forecasting task.

Data set	ADMM	Greedy	HOPLS	HOLRR	K-HOLRR (poly)	K-HOLRR (rbf)
CCDS	0.8448	0.8325	0.8147	0.8096	0.8275	**0.7913**
Foursquare	0.1407	**0.1223**	**0.1224**	0.1227	**0.1223**	0.1226
Meteo-UK	0.6140	–	0.625	0.5971	0.6107	**0.5886**

To see how sensitive HOLLR is w.r.t. the choice of the multilinear rank, we carried out a similar experiment comparing HOLLR performances for different values of the rank parameter, see Fig. 2 (right). In this experiment, the rank of the tensor $\mathcal{W}$ used to generate the data is $(2, 2, 2, 2)$ while the input and output dimensions and the noise level are the same as above.

4.2 Real Data

We evaluate our algorithm on a forecasting task on the following real-world data sets:
CCDS: the comprehensive climate data set is a collection of climate records of North America from [15]. The data set contains monthly observations of 17 variables such as Carbon dioxide and temperature spanning from 1990 to 2001 across 125 observation locations.
Foursquare: the Foursquare data set [14] contains users' check-in records in Pittsburgh area categorized by different venue types such as Art & University. It records the number of check-ins by 121 users in each of the 15 category of venues over 1200 time intervals.
Meteo-UK: The data set is collected from the meteorological office of the UK[2]. It contains monthly measurements of 5 variables in 16 stations across the UK from 1960 to 2000.

The forecasting task consists in predicting all variables at times $t+1,\ldots,t+k$ from their values at times $t-2$, $t-1$ and t. The first two real data sets were used in [2] with $k=1$ (i.e. outputs are matrices). We consider here the same setting for these two data sets. For the third dataset we consider higher-order output tensors by setting $k=5$. The output tensors are thus of size respectively 17×125, 15×121 and $16\times 5\times 5$ for the three datasets.

For all the experiments, we use 90% of the available data for training and 10% for testing. All hyper-parameters are chosen by cross-validation. The average test RMSE over 10 runs are reported in Table 1 (running times are reported in Table 1 in supplementary material B). We see that HOLRR and K-HOLRR outperforms the other methods on the CCDS dataset while being orders of magnitude faster for the kernelized version (0.61s vs. 75.47s for Greedy and 235.73s for ADMM in average). On the Foursquare dataset HOLRR performs as well as Greedy and on the Meteo-UK dataset K-HOLRR gets the best results with the RBF kernel while being much faster than ADMM (1.66s vs. 40.23s in average).

5 Conclusion

We proposed a low-rank multilinear regression model for tensor-structured output data. We developed a fast and efficient algorithm to tackle the multilinear rank penalized minimization problem and provided theoretical guarantees. Experimental results showed that capturing low-rank structure in the output data can help to improve tensor regression performance.

[2] http://www.metoffice.gov.uk/public/weather/climate-historic/

Acknowledgments
We thank François Denis and the reviewers for their helpful comments and suggestions. This work was partially supported by ANR JCJC program MAD (ANR- 14-CE27-0002).

References

[1] T. W. Anderson. Estimating linear restrictions on regression coefficients for multivariate normal distributions. *Annals of Mathematical Statistics*, 22:327–351, 1951.
[2] M. T. Bahadori, Q. R. Yu, and Y. Liu. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *NIPS*. 2014.
[3] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.
[4] A. Cichocki, R. Zdunek, A.H. Phan, and S.I. Amari. *Nonnegative Matrix and Tensor Factorizations*. Wiley, 2009.
[5] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
[6] R. Foygel, M. Horrell, M. Drton, and J. D. Lafferty. Nonparametric reduced rank regression. In *NIPS*, 2012.
[7] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.
[8] N. Halko, P. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM*, 53(2):217–288, 2011.
[9] C. J. Hillar and L. Lim. Most tensor problems are np-hard. *JACM*, 60(6):45, 2013.
[10] A. J. Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975.
[11] A. J. Izenman. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer-Verlag, New York, 2008.
[12] P. Kar and H. Karnick. Random feature maps for dot product kernels. In *AISTATS*, 2012.
[13] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
[14] X. Long, L. Jin, and J. Joshi. Exploring trajectory-driven local geographic topics in foursquare. In *UbiComp*, 2012.
[15] A. C. Lozano, H. Li, A. Niculescu-Mizil, Y. Liu, C. Perlich, J. Hosking, and N. Abe. Spatial-temporal causal modeling for climate change attribution. In *KDD*, 2009.
[16] H. Lu, K.N. Plataniotis, and A. Venetsanopoulos. *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. CRC Press, 2013.
[17] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT, 2012.
[18] A. Mukherjee and J. Zhu. Reduced rank ridge regression and its kernel extensions. *Statistical analysis and data mining*, 4(6):612–622, 2011.
[19] M. Nickel and V. Tresp. An analysis of tensor models for learning on structured data. In *Machine Learning and Knowledge Discovery in Databases*, pages 272–287. Springer, 2013.
[20] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.
[21] G.C. Reinsel and R.P. Velu. *Multivariate reduced-rank regression: theory and applications*. Lecture Notes in Statistics. Springer, 1998.
[22] B. Romera-Paredes, M. H. Aung, N. Bianchi-Berthouze, and M. Pontil. Multilinear multitask learning. In *ICML*, 2013.
[23] M. Signoretto, L. De Lathauwer, and J. K. Suykens. Learning tensors in reproducing kernel hilbert spaces with multilinear spectral penalties. *arXiv preprint arXiv:1310.4977*, 2013.
[24] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. K. Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Mach. Learn., 1–49*, 2013.
[25] N. Srebro, N. Alon, and T. S. Jaakkola. Generalization error bounds for collaborative prediction with low-rank matrices. In *NIPS*, 2004.
[26] Hugh E Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133(1):167–178, 1968.
[27] K. Wimalawarne, M. Sugiyama, and R. Tomioka. Multitask learning meets tensor factorization: task imputation via convex optimization. In *NIPS*. 2014.
[28] Q. Zhao, C. F. Caiafa, D. P. Mandic, Z. C. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki. Higher-order partial least squares (hopls). *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(7):1660–1673, 2012.
[29] Q. Zhao, Guoxu Z., T. Adalı, L. Zhang, and A. Cichocki. Kernel-based tensor partial least squares for reconstruction of limb movements. In *ICASSP*, 2013.
[30] H. Zhou, L. Li, and H. Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

Low-Rank Regression with Tensor Responses (Supplementary Material)

Guillaume Rabusseau and Hachem Kadri
Aix Marseille Univ, CNRS, LIF, Marseille, France
{firstname.lastname}@lif.univ-mrs.fr

A Proofs

A.1 Proof of Theorem 1

Theorem. *For given orthogonal matrices* $\mathbf{U}_0, \cdots, \mathbf{U}_p$ *the tensor* $\boldsymbol{\mathcal{G}}$ *that minimizes (2) is given by*

$$\boldsymbol{\mathcal{G}} = \tilde{\boldsymbol{\mathcal{Y}}} \times_1 (\mathbf{U}_0^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \mathbf{U}_0)^{-1} \mathbf{U}_0^\top \tilde{\mathbf{X}}^\top \times_2 \mathbf{U}_1^\top \times_3 \cdots \times_{p+1} \mathbf{U}_p^\top \ .$$

Proof. Since the Frobenius norm of a tensor is equal to the one of its vectorization the objective function in (2) can be written as

$$\|(\mathbf{U}_p \otimes \mathbf{U}_{p-1} \otimes \cdots \otimes \mathbf{U}_1 \otimes \tilde{\mathbf{X}}\mathbf{U}_0)\mathrm{vec}(\boldsymbol{\mathcal{G}}) - \mathrm{vec}(\tilde{\boldsymbol{\mathcal{Y}}})\|_F^2 \ .$$

Let $\mathbf{M} = \mathbf{U}_p \otimes \mathbf{U}_{p-1} \otimes \cdots \otimes \mathbf{U}_1 \otimes \tilde{\mathbf{X}}\mathbf{U}_0$. The solution w.r.t. $\mathrm{vec}(\boldsymbol{\mathcal{G}})$ of this classical linear least-squares problem is given by $(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top$. Using the mixed-product and inverse properties of the Kronecker product and the column-wise orthogonality of $\mathbf{U}_1, \cdots, \mathbf{U}_p$ we obtain $\mathrm{vec}(\boldsymbol{\mathcal{G}}) = \left(\mathbf{U}_p \otimes \cdots \otimes \mathbf{U}_1 \otimes (\mathbf{U}_0^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \mathbf{U}_0)^{-1} \mathbf{U}_0^\top \tilde{\mathbf{X}}^\top\right) \mathrm{vec}(\tilde{\boldsymbol{\mathcal{Y}}})$. □

A.2 Proof of Proposition 1

Proposition. *For* $0 \leq i \leq p$*, using the definition of* $\boldsymbol{\Pi}_i$ *in (3), the optimal solution of*

$$\min_{\mathbf{U}_i \in \mathbb{R}^{d_i \times R_i}} \|\tilde{\boldsymbol{\mathcal{Y}}} \times_{i+1} \boldsymbol{\Pi}_i - \tilde{\boldsymbol{\mathcal{Y}}}\|_F^2 \ \textit{s.t.} \ \mathbf{U}_i^\top \mathbf{U}_i = \mathbf{I}$$

is given by the eigenvectors of

$$\begin{cases} (\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^\top \tilde{\mathbf{Y}}_{(1)} \tilde{\mathbf{Y}}_{(1)}^\top \tilde{\mathbf{X}} & \text{if } i = 0 \\ \tilde{\mathbf{Y}}_{(i)} \tilde{\mathbf{Y}}_{(i)}^\top & \text{otherwise} \end{cases}$$

that corresponds to the R_i *largest eigenvalues.*

Proof. For any $0 \leq i \leq p$, since $\boldsymbol{\Pi}_i$ is a projection we have $\langle \tilde{\boldsymbol{\mathcal{Y}}} \times_1 \boldsymbol{\Pi}_i, \tilde{\boldsymbol{\mathcal{Y}}} \rangle = \langle \boldsymbol{\Pi}_i \tilde{\mathbf{Y}}_{(i)}, \tilde{\mathbf{Y}}_{(i)} \rangle = \|\boldsymbol{\Pi}_i \tilde{\mathbf{Y}}_{(i)}\|_F^2$, thus minimizing $\|\tilde{\boldsymbol{\mathcal{Y}}} \times_i \boldsymbol{\Pi}_i - \tilde{\boldsymbol{\mathcal{Y}}}\|_F^2$ is equivalent to minimizing $\|\boldsymbol{\Pi}_i \tilde{\mathbf{Y}}_{(i)}\|_F^2 - 2\langle \boldsymbol{\Pi}_i \tilde{\mathbf{Y}}_{(i)}, \tilde{\mathbf{Y}}_{(i)} \rangle = -\|\boldsymbol{\Pi}_i \tilde{\mathbf{Y}}_{(i)}\|_F^2$. For $i \geq 1$, we have $\|\boldsymbol{\Pi}_i \tilde{\mathbf{Y}}_{(i)}\|_F^2 = \mathrm{Tr}(\mathbf{U}_i^\top \tilde{\mathbf{Y}}_{(i)} \tilde{\mathbf{Y}}_{(i)}^\top \mathbf{U}_i)$ which is maximized by letting the columns of $\mathbf{U}_i$ be the top R_i eigenvectors of the matrix $\tilde{\mathbf{Y}}_{(i)} \tilde{\mathbf{Y}}_{(i)}^\top$. For $i = 0$ we have $\|\boldsymbol{\Pi}_0 \tilde{\mathbf{Y}}_{(i)}\|_F^2 = \mathrm{Tr}(\boldsymbol{\Pi}_0 \tilde{\mathbf{Y}}_{(1)} \tilde{\mathbf{Y}}_{(1)}^\top \boldsymbol{\Pi}_0^\top) = \mathrm{Tr}\left((\mathbf{U}_0^\top \mathbf{A} \mathbf{U}_0)^{-1} \mathbf{U}_0^\top \mathbf{B} \mathbf{U}_0\right)$ with $\mathbf{A} = \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$ and $\mathbf{B} = \tilde{\mathbf{X}}^\top \tilde{\mathbf{Y}}_{(1)} \tilde{\mathbf{Y}}_{(1)}^\top \tilde{\mathbf{X}}$, which is maximized by the top R_0 eigenvectors of $\mathbf{A}^{-1}\mathbf{B}$. □

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

A.3 Proof of Proposition 2

Proposition. *If $\boldsymbol{\alpha} \in \mathbb{R}^N$ is an eigenvector with eigenvalue λ of the matrix*

$$(\mathbf{K} + \gamma \mathbf{I})^{-1} \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \mathbf{K} \ ,$$

then $\mathbf{v} = \boldsymbol{\Phi}^\top \boldsymbol{\alpha} \in \mathbb{R}^L$ is an eigenvector with eigenvalue λ of the matrix $(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \gamma \mathbf{I})^{-1} \boldsymbol{\Phi}^\top \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \boldsymbol{\Phi}$.

Proof. Let $\boldsymbol{\alpha} \in \mathbb{R}^N$ be the eigenvector from the hypothesis. We have

$$\begin{aligned}
\lambda \mathbf{v} = \boldsymbol{\Phi}^\top (\lambda \boldsymbol{\alpha}) &= \boldsymbol{\Phi}^\top \left((\mathbf{K} + \gamma \mathbf{I})^{-1} \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \mathbf{K} \right) \boldsymbol{\alpha} \\
&= \boldsymbol{\Phi}^\top (\boldsymbol{\Phi} \boldsymbol{\Phi}^\top + \gamma \mathbf{I})^{-1} \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \boldsymbol{\Phi} \boldsymbol{\Phi}^\top \boldsymbol{\alpha} \\
&= \left((\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \gamma \mathbf{I})^{-1} \boldsymbol{\Phi}^\top \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^\top \boldsymbol{\Phi} \right) \mathbf{v} \ .
\end{aligned}$$

□

A.4 Proof of Theorem 2

Theorem. *Let $\boldsymbol{\mathcal{W}}^*$ be a solution of problem (1) and let $\boldsymbol{\mathcal{W}}$ be the regression tensor returned by Algorithm 1. If $\mathcal{L} : \mathbb{R}^{d_0 \times \cdots \times d_p} \to \mathbb{R}$ denotes the objective function of (1) with respect to $\boldsymbol{\mathcal{W}}$ then*

$$\mathcal{L}(\boldsymbol{\mathcal{W}}) \leq (p+1) \mathcal{L}(\boldsymbol{\mathcal{W}}^*).$$

The proof of this theorem relies on the following lemma which was proved in [1] to obtain a nice and elegant proof for the approximation guarantees of the HOSVD algorithm for the problem of low multilinear rank approximation of a given tensor.

Lemma 1. *Let $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times \cdots \times d_p}$ be a pth order tensor, let $m, n \in [p]$, and let $\mathbf{P} \in \mathbb{R}^{d_m \times d_m}$ and $\mathbf{Q} \in \mathbb{R}^{d_n \times d_n}$ be two orthogonal projection matrices. Then*

$$\| \boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{T}} \times_m \mathbf{P} \times_n \mathbf{Q} \|_F^2 \leq \| \boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{T}} \times_m \mathbf{P} \|_F^2 + \| \boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{T}} \times_n \mathbf{Q} \|_F^2.$$

Proof. First observe that for any orthogonal projection matrix $\boldsymbol{\Pi}$ and any tensors $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}$ we have

$$\| \boldsymbol{\mathcal{A}} \times_n \boldsymbol{\Pi} \|_F^2 \leq \| \boldsymbol{\mathcal{A}} \|_F^2 \quad \text{and} \quad \| \boldsymbol{\mathcal{A}} \times_n (\mathbf{I} - \boldsymbol{\Pi}) + \boldsymbol{\mathcal{B}} \times_n \boldsymbol{\Pi} \|_F^2 = \| \boldsymbol{\mathcal{A}} \times_n (\mathbf{I} - \boldsymbol{\Pi}) \|_F^2 + \| \boldsymbol{\mathcal{B}} \times_n \boldsymbol{\Pi} \|_F^2.$$

Both equations follow from the fact that the Frobenius norm of a tensor is equal to the one of any of its matricization. Indeed

$$\| \boldsymbol{\mathcal{A}} \times_n \boldsymbol{\Pi} \|_F^2 = \| \boldsymbol{\Pi} \mathbf{A}_{(n)} \|_F^2 \leq \| \mathbf{A}_{(n)} \|_F^2 = \| \boldsymbol{\mathcal{A}} \|_F^2$$

since $\boldsymbol{\Pi}$ is a projection. The second equality is proved similarly using the orthogonality of $\boldsymbol{\Pi}$ and $\mathbf{I} - \boldsymbol{\Pi}$.

Then, under the hypothesis of the lemma, we have

$$\begin{aligned}
\| \boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{T}} \times_m \mathbf{P} \times_n \mathbf{Q} \|_F^2 &= \| \boldsymbol{\mathcal{T}} \times_m (\mathbf{I} - \mathbf{P}) + (\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{T}} \times_n \mathbf{Q}) \times_m \mathbf{P} \|_F^2 \\
&= \| \boldsymbol{\mathcal{T}} \times_m (\mathbf{I} - \mathbf{P}) \|_F^2 + \| (\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{T}} \times_n \mathbf{Q}) \times_m \mathbf{P} \|_F^2 \\
&\leq \| \boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{T}} \times_m \mathbf{P} \|_F^2 + \| \boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{T}} \times_n \mathbf{Q} \|_F^2.
\end{aligned}$$

□

Let $\mathbf{U}_0, \cdots, \mathbf{U}_p$ be the matrices defined in Algorithm 1 and let $\boldsymbol{\Pi}_0, \cdots, \boldsymbol{\Pi}_p$ be the orthogonal projection matrices defined in problem (3). The regression tensor $\boldsymbol{\mathcal{W}}$ returned by HOLRR satisfies

$$\boldsymbol{\mathcal{W}} \times_1 \tilde{\mathbf{X}} = \tilde{\boldsymbol{\mathcal{Y}}} \times_1 \boldsymbol{\Pi}_0 \times_2 \cdots \times_{p+1} \boldsymbol{\Pi}_p.$$

Similarly, it follows from Theorem 1 that a solution $\boldsymbol{\mathcal{W}}^*$ of problem (1) satisfies

$$\boldsymbol{\mathcal{W}}^* \times_1 \tilde{\mathbf{X}} = \tilde{\boldsymbol{\mathcal{Y}}} \times_1 \boldsymbol{\Pi}_0^* \times_2 \cdots \times_{p+1} \boldsymbol{\Pi}_p^*$$

for some orthogonal projection matrices $\boldsymbol{\Pi}_i^*$ for $0 \leq i \leq p$.

Using successive applications of the previous Lemma we obtain

$$\mathcal{L}(\mathcal{W}) = \|\mathcal{W} \times_1 \tilde{\mathbf{X}} - \tilde{\mathcal{Y}}\|_F^2 = \|\tilde{\mathcal{Y}} \times_1 \mathbf{\Pi}_0 \times_2 \cdots \times_{p+1} \mathbf{\Pi}_p - \tilde{\mathcal{Y}}\|_F^2 \leq \sum_{i=0}^{p} \|\tilde{\mathcal{Y}} \times_{i+1} \mathbf{\Pi}_i - \tilde{\mathcal{Y}}\|_F^2.$$

By Proposition 1, each summand in this upper bound is minimal with respect to $\mathbf{\Pi}_i$, hence $\|\tilde{\mathcal{Y}} \times_{i+1} \mathbf{\Pi}_i - \tilde{\mathcal{Y}}\|_F^2 \leq \|\tilde{\mathcal{Y}} \times_{i+1} \mathbf{\Pi}_i^* - \tilde{\mathcal{Y}}\|_F^2$ for any $i \in [p]$. It remains to show that

$$\|\tilde{\mathcal{Y}} \times_{i+1} \mathbf{\Pi}_i^* - \tilde{\mathcal{Y}}\|_F^2 \leq \|\tilde{\mathcal{Y}} \times_1 \mathbf{\Pi}_0^* \times_2 \cdots \times_{p+1} \mathbf{\Pi}_p^* - \tilde{\mathcal{Y}}\|_F^2 = \mathcal{L}(\mathcal{W}^*)$$

for all $i \in [p]$. Indeed, using the fact that the Frobenius norm of a tensor is equal to the one of its matricization, we obtain for the case $i = 0$

$$\begin{aligned}
\|\tilde{\mathcal{Y}} \times_1 \mathbf{\Pi}_0^* \times_2 \cdots \times_{p+1} \mathbf{\Pi}_p^* - \tilde{\mathcal{Y}}\|^2 &= \|\mathbf{\Pi}_0^* \tilde{\mathbf{Y}}_{(1)} (\mathbf{\Pi}_p^* \otimes \cdots \otimes \mathbf{\Pi}_1^*)^\top - \tilde{\mathbf{Y}}_{(1)}\|_F^2 \\
&= \|(\mathbf{\Pi}_0^* - \mathbf{I}_{d_0}) \tilde{\mathbf{Y}}_{(1)} + \mathbf{\Pi}_0^* \tilde{\mathbf{Y}}_{(1)} (\mathbf{\Pi}_p^* \otimes \cdots \otimes \mathbf{\Pi}_1^* - \mathbf{I}_{d_1 d_2 \cdots d_p})^\top\|_F^2 \\
&= \|(\mathbf{\Pi}_0^* - \mathbf{I}_{d_0}) \tilde{\mathbf{Y}}_{(1)}\|_F^2 + \|\mathbf{\Pi}_0^* \tilde{\mathbf{Y}}_{(1)} (\mathbf{\Pi}_p^* \otimes \cdots \otimes \mathbf{\Pi}_1^* - \mathbf{I}_{d_1 d_2 \cdots d_p})^\top\|_F^2 \\
&\geq \|(\mathbf{\Pi}_0^* - \mathbf{I}_{d_0}) \tilde{\mathbf{Y}}_{(1)}\|_F^2 \\
&= \|\tilde{\mathcal{Y}} \times_1 \mathbf{\Pi}_0^* - \tilde{\mathcal{Y}}\|_F^2
\end{aligned}$$

where we used the orthogonality of $\mathbf{\Pi}_0^*$ and $\mathbf{\Pi}_0^* - \mathbf{I}_{d_0}$. The proofs for other values of i are similar.

A.5 Proof of Theorem 3

We start by bounding the pseudo-dimension of the class of real-valued functions with domain $\mathbb{R}^{d_0} \times [d_1] \times \cdots \times [d_p]$

$$\tilde{\mathcal{F}} = \left\{ (\mathbf{x}, i_1, \cdots, i_p) \mapsto (\mathcal{W} \bullet_1 \mathbf{x})_{i_1, \cdots, i_p} \ : \ \text{rank}(\mathcal{W}) = (R_0, \cdots, R_p) \right\}.$$

We first recall the definition of the pseudo-dimension of a class of real-valued functions.

Definition 1. *A class $\mathcal{F}$ of real-valued functions pseudo-shatters the points $x_1, \cdots, x_m$ with thresholds $t_1, \cdots, t_m$ if for every binary labeling of the points $(s_1, \cdots, s_m) \in \{-, +\}^m$ there exists $f \in \mathcal{F}$ s.t. $f(x_i) < t_i$ iff $s_i = -$. The pseudo-dimension of a class F is the supremum over m for which there exist m points that are pseudo-shattered by $\mathcal{F}$ (with some thresholds).*

We say that a set of polynomials $p_1, p_2, \cdots, p_k$ has at least m sign patterns if there exist $x_1, \cdots, x_m$ such that such that the sign vectors $\mathbf{v}_i = [sign(p_1(x_i)), \cdots, sign(p_k(x_i))]^\top$ are pairwise distinct. Following [4], the following theorem bounds the number of sign patterns for a set of polynomials.

Theorem. *[3, Theorem 34, 35] The number of sign patterns of r polynomials, each of degree at most d, over q variables is at most $\left(\frac{4edr}{q}\right)^q$ for all $r > q > 2$.*

The following lemma gives an upper bound on the pseudo-dimension of $\tilde{\mathcal{F}}$ using the previous theorem.

Lemma 2. *The pseudo-dimension of the real-valued function class $\tilde{\mathcal{F}}$ is upper bounded by $(R_0 R_1 \cdots R_p + \sum_{i=0}^{p} R_i d_i) \log\left(\frac{4e(p+2) d_0 d_1 \cdots d_p}{d_0 + d_1 + \cdots + d_p}\right)$.*

Proof. It is well known that the pseudo-dimension of a vector space of real-valued functions is equal to its dimension [2, Theorem 10.5]. Since $\tilde{\mathcal{F}}$ is a (non-linear) subspace of the $d_0 d_1 \cdots d_p$-dimensional vector space

$$\left\{ (\mathbf{x}, i_1, \cdots, i_p) \mapsto (\mathcal{W} \bullet_1 \mathbf{x})_{i_1, \cdots, i_p} \ : \ \mathcal{W} \in \mathbb{R}^{d_0 \times \cdots \times d_p} \right\}$$

of real-valued functions with domain $\mathbb{R}^{d_0} \times [d_1] \times \cdots \times [d_p]$, the pseudo-dimension of $\tilde{\mathcal{F}}$ is bounded by $d_0 d_1 \cdots d_p$.

Now, let $m \leq d_0 \cdots d_p$ and let $\{(\mathbf{x}^k, i_1^k, \cdots, i_p^k)\}_{k=1}^m$ be a set of points that are pseudo-shattered by $\tilde{\mathcal{F}}$ with thresholds $t_1, \cdots, t_m \in \mathbb{R}$. Then for each sign pattern $(s_1, \cdots, s_m) \in$

$\{-,+\}^m$, there exists $\tilde{f} \in \tilde{\mathcal{F}}$ such that $sign(\tilde{f}(\mathbf{x}^k, i_1^k, \cdots, i_p^k) - t_k) = s_k$. Any function $\tilde{f} \in \tilde{\mathcal{F}}$ can be written as

$$(\mathbf{x}, j_1, \cdots, j_p) \mapsto \left(\boldsymbol{\mathcal{G}} \times_1 \mathbf{x}^\top \mathbf{U}_0 \times_2 \mathbf{U}_1 \cdots \times_{p+1} \mathbf{U}_p\right)_{j_1, \cdots, j_p}$$

for some $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_0 \times \cdots \times R_p}$, $\mathbf{U}_i \in \mathbb{R}^{d_i \times R_i}$ for $0 \leq i \leq p$. Thus, considering the entries of $\boldsymbol{\mathcal{G}}, \mathbf{U}_0, \cdots, \mathbf{U}_p$ as variables, the set $\{\tilde{f}(\mathbf{x}^k, i_1^k, \cdots, i_p^k) - t_k\}_{k=1}^m$ can be seen as a set of m polynomials of degree at most $p+2$ over these $D = R_0 \cdots R_p + \sum_{i=0}^p d_i R_i$ variables. It then follows from the previous theorem that $2^m \leq \left(\frac{4e(p+2)m}{D}\right)^D$. The result follows using $m \leq d_0 \cdots d_p$ and $D \geq \sum_{i=0}^p d_i$.

□

Once the pseudo-dimension of the function class $\tilde{\mathcal{F}}$ is bounded, one can invoke standard error generalization bounds in terms of the pseudo-dimension [2, Theorem 10.6] to obtain the following theorem that gives an upper bound on the excess risk for the class of function

$$\mathcal{F} = \{\mathbf{x} \mapsto \boldsymbol{\mathcal{W}} \bullet_1 \mathbf{x} \ : \ \text{rank}(\boldsymbol{\mathcal{W}}) = (R_0, \cdots, R_p)\}.$$

Theorem. *Let $\mathcal{L} : \mathbb{R}^{d_1 \times \cdots \times d_p} \to \mathbb{R}$ be a loss function satisfying*

$$\mathcal{L}(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}) = \frac{1}{d_1 \cdots d_p} \sum_{i_1, \cdots, i_p} \ell(\boldsymbol{\mathcal{A}}_{i_1, \cdots, i_p}, \boldsymbol{\mathcal{B}}_{i_1, \cdots, i_p})$$

for some loss function $\ell : \mathbb{R} \to \mathbb{R}^+$ bounded by M. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample of size N, the following inequality holds for all $h \in \mathcal{F}$:

$$R(h) \leq \hat{R}(h) + M\sqrt{\frac{2D \log\left(\frac{4e(p+2)d_0 d_1 \cdots d_p}{d_0 + d_1 + \cdots + d_p}\right) \log N}{N}} + M\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2N}}$$

where $D = R_0 R_1 \cdots R_p + \sum_{i=0}^p R_i d_i$.

Proof. For any $h : \mathbb{R}^{d_0} \to \mathbb{R}^{d_1 \times \cdots \times d_p}$ we define $\tilde{h} : \mathbb{R}^{d_0} \times [d_1] \times \cdots \times [d_p] \to \mathbb{R}$ by $\tilde{h}(\mathbf{x}, i_1, \cdots, i_p) = h(\mathbf{x})_{i_1 \cdots i_p}$. Let $\mathcal{D}$ denote the distribution of the input data. We have

$$\begin{aligned} R(h) &= \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}}[\mathcal{L}(f(\mathbf{x}), h(\mathbf{x}))] = \frac{1}{d_1 \cdots d_p} \sum_{i_1, \cdots, i_p} \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}}[\ell(f(\mathbf{x})_{i_1 \cdots i_p}, h(\mathbf{x})_{i_1 \cdots i_p})] \\ &= \mathop{\mathbb{E}}_{\substack{\mathbf{x} \sim \mathcal{D} \\ i_k \sim \mathcal{U}(d_k), k \in [p]}} [\ell(\tilde{f}(\mathbf{x}, i_1, \cdots, i_p), \tilde{h}(\mathbf{x}, i_1, \cdots, i_p))] \end{aligned}$$

where $\mathcal{U}(k)$ denotes the discrete uniform distribution on $[k]$ for any integer $k \geq 1$. It follows that $R(h) = R(\tilde{h})$. Similarly, one can show that $\hat{R}(h) = \hat{R}(\tilde{h})$. The result then directly follows using Theorem 10.6 in [2] (see below) to bound $R(\tilde{h}) - \hat{R}(\tilde{h})$. □

Theorem (Theorem 10.6 in [2])**.** *Let H be a family of real-valued functions and let $G = \{x \mapsto L(h(x), f(x)) : h \in H\}$ be the family of loss functions associated to H. Assume that the pseudo-dimension of G is bounded by d and that the loss function L is bounded by M. Then, for any $\delta > 0$, with probability at least δ over the choice of a sample of size m, the following inequality holds for all $h \in H$:*

$$R(h) \leq \hat{R}(h) + M\sqrt{\frac{2d \log\left(\frac{em}{d}\right)}{m}} + M\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}.$$

B Experiments

B.1 Running Times

The running times of different tensor response regression algorithms on synthetic and real data sets are given in Table 1.

Table 1: Average running times in seconds for some of the experiments. We did not run MLMT-NC on the real world data sets because it is computationally very expensive. The implementation of the Greedy algorithm is limited to 2nd order output tensors, this is why we did not run it on the synthetic and Meteo UK data sets. Finally, the synthetic non linear data was generated using a polynomial relation which is why the RBF kernel was not used on this data set.

Data set	MLMTL-NC	ADMM	Greedy	HOPLS	HOLRR	K-HOLRR (poly)	K-HOLRR (rbf)
Synthetic	945.79	12.92	–	0.12	0.04	0.53	–
CCDS	–	235.73	75.47	121.28	100.94	0.46	0.61
Foursquare	–	33.83	37.70	22.3	14.41	19.20	19.67
Meteo UK	–	40.23	–	2.12	1.67	1.57	1.66

B.2 Image Reconstruction from Noisy Measurements

To give an illustrative intuition on the differences between matrix and multilinear rank regularization we generate data from the model $\mathcal{Y} = \mathcal{W} \bullet_1 \mathbf{x} + \mathcal{E}$ where the tensor $\mathcal{W}$ is a color image of size $m \times n$ encoded with three color channels RGB. We consider two different tasks depending on the input dimension: (i) $\mathcal{W} \in \mathbb{R}^{3\times m\times n}$, $\mathbf{x} \in \mathbb{R}^3$ and (ii) $\mathcal{W} \in \mathbb{R}^{n\times m\times 3}$, $\mathbf{x} \in \mathbb{R}^n$. In both tasks the components of both $\mathbf{x}$ and $\mathcal{E}$ are drawn from $\mathcal{N}(0, 1)$ and the regression tensor $\mathcal{W}$ is learned from a training set of size 200.

This experiment allows us to visualize the tensors returned by the RLS, LRR and HOLRR algorithms. The results are shown in Figure 1 for three images: a green cross (of size 50×50), a thumbnail of a Rothko painting (44×70) and a square made of triangles (70×70), note that the first two images have a low rank structure which is not the case for the third one.

We first see that HOLRR clearly outperforms LRR on the task where the input dimension is small (task (i)). This is to be expected since the rank of the matrix $\mathbf{W}_{(1)}$ is at most 3 and LRR is unable to enforce a low-rank structure on the output modes of $\mathcal{W}$. When the rank constraint is set to 1 for LRR and $(3, 1, 1)$ for HOLRR, we clearly see that (unlike HOLRR) the LRR approach does not enforce any low-rank structure on the regression tensor along the output modes. On task (ii) the difference is more subtle, but we can see that setting a rank constraint of 2 for the LRR algorithm prevents the model from capturing the white border around the green cross and creates the vertical lines artifact in the Rothko painting. For higher values of the rank the model starts to learn the noise. The tensor returned by HOLRR with rank $(2, 2, 3)$ for the cross image and $(4, 4, 3)$ for the Rothko painting do not exhibit these behaviors and give better results on these two images. On the square image which does not have a low-rank structure both algorithms exhibit underfitting for low values of the rank parameter. Overall, we see that capturing the multilinear low-rank structure of the output data allows HOLRR to separate the noise from the true signal better than RLS and LRR.

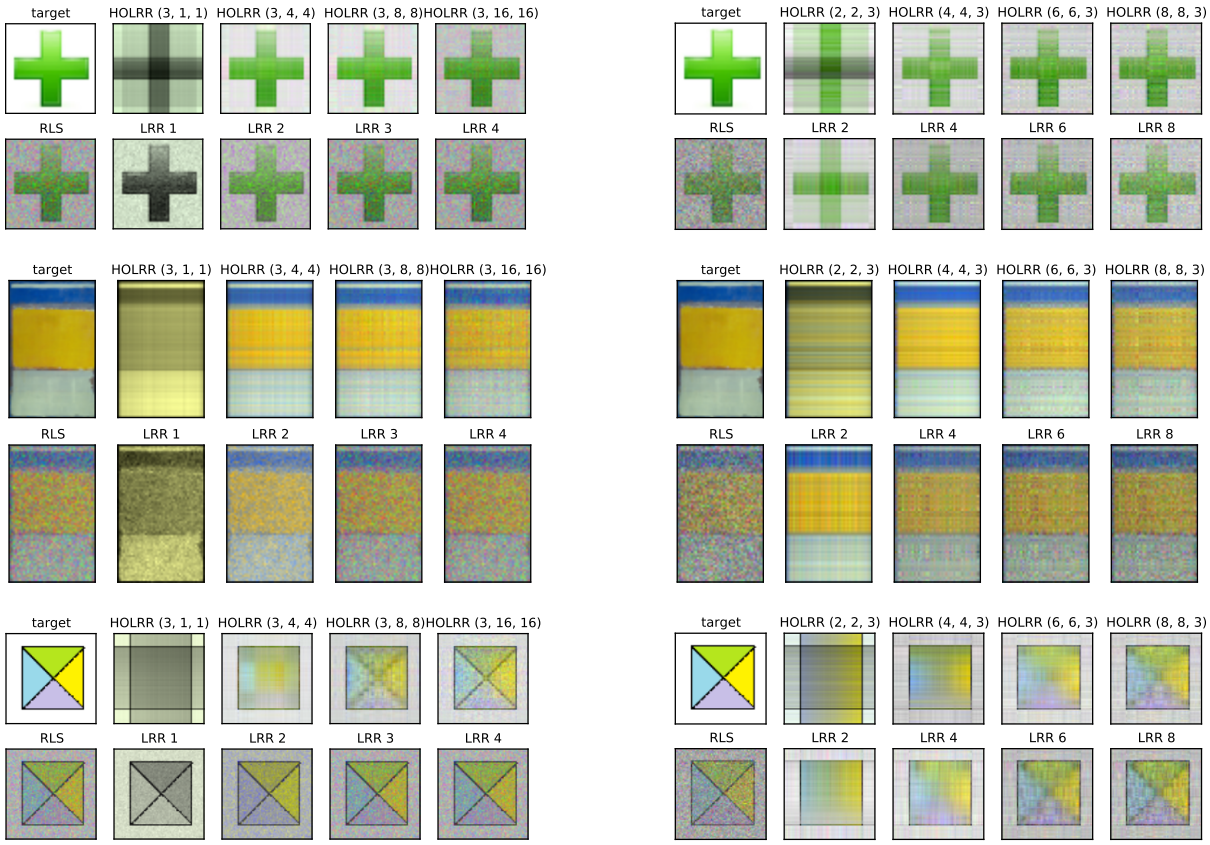

Figure 1: Image reconstruction from noisy measurements: $\mathcal{Y} = \mathcal{W} \bullet_1 \mathbf{x} + \mathcal{E}$ where $\mathcal{W}$ is a color image (RGB). (left) Task (i): input dimension is the number of channels. (right) Task (ii): input dimension is the height of the image. Each image is labeled with the name of the algorithm followed by the value used for the rank constraint.

References

[1] Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

[2] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT, 2012.

[3] N. Srebro. *Learning with Matrix Factorizations*. PhD thesis, MIT, 2004.

[4] Hugh E Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133(1):167–178, 1968.