

Tensor Learning for Regression

Weiwei Guo, Irene Kotsia, *Member, IEEE*, and Ioannis Patras, *Senior Member, IEEE*,

Abstract

In this paper, we exploit the advantages of tensorial representations and propose several Tensor Learning models for regression. The model is based on the Canonical (CANDECOMP)/Parallel Factors (PARAFAC) decomposition of tensors of multiple modes and allows the simultaneous projections of an input tensor to more than one directions along each mode. Two empirical risk functions are studied, namely the square loss and the ϵ -insensitive loss functions. The former leads to higher rank Tensor Ridge Regression (hrTRR) and the latter to higher rank Support Tensor Regression (hrSTR), both formulated using the Frobenius norm for regularization. We also use the group sparsity norm for regularization, favoring in that way the low rank decomposition of the tensorial weight. In that way we achieve the automatic selection of the rank during the learning process and obtain the optimal rank TRR (orTRR) and optimal rank STR (orSTR). Experiments conducted for the problems of head pose, human age and 3D body pose estimation using real data from publicly available databases, verified not only the superiority of tensors over their vector-counterparts but also the efficiency of the proposed algorithms.

Index Terms

Manuscript received ; revised . This work was supported by the EPSRC grant 'Recognition and Localization of Human Actions in Image Sequences'(EP/G033935/1) and part of this work has been done while Weiwei Guo was in National University of Defense Technology, China

Copyright (c) 2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Weiwei Guo is with the School Computer Science and Electronic Engineering, Queen Mary, University of London, Mile End road, E1 4NS, London, UK. Email: weiwei.guo@eecs.qmul.ac.uk

Irene Kotsia is with the School Computer Science and Electronic Engineering, Queen Mary, University of London, Mile End road, E1 4NS, London, UK. Email: irene.kotsia@eecs.qmul.ac.uk

Ioannis Patras is with the School Computer Science and Electronic Engineering, Queen Mary, University of London, Mile End road, E1 4NS, London, UK. Email: I.Patras@eecs.qmul.ac.uk

Tensors, Canonical (CANDECOMP)/Parallel Factors (PARAFAC) decomposition, Ridge Regression, Support Vector Regression, Frobenius norm, Group Sparsity Norm

I. INTRODUCTION

Tensors can be regarded as natural representations of visual data (some examples are given in Fig. 1). However, most approaches in literature work on vector spaces that are derived by stacking the original tensor elements in a more or less arbitrary order. This vectorization of data creates many issues. First, the underlying structural information is disregarded. Second, the vectorization of a tensor results in the creation of a vector of potentially very high dimensionality. This may lead to overtraining, high computational complexity and large memory requirements. Therefore, several algorithms that used tensor representations were recently proposed for a number of problems [1]–[3]. In most cases it has been shown that they outperform their vector-based counterparts.

The advantages of tensor-based methods seem to stem from the way tensors are decomposed. More specifically, the unknown parameter (weight) tensor is usually constrained to be a linear combination of rank-one components, that is a linear combination of simple tensors that can be expressed as the outer product of low dimensional vectors. This leads to fewer parameters to be estimated and acts as a feature selection or dimensionality reduction scheme that takes the structure of the feature space into consideration. Factorizing the parameter space into a product of different factors, reduces the number of unknowns to be estimated. Usually, the parameters that are associated with each mode are estimated in an iterative manner, where at each iteration, only the parameters associated with a single mode are updated. Thus, at each iteration, a problem of reduced dimensionality needs to be solved.

Recently, several classic vector-based unsupervised and supervised learning approaches have been extended to deal with tensorial data. In the setting of unsupervised tensor dimension reduction, [4] represented a collection of images as a 3^{rd} order tensor consisting of slices of 2D images and used a rank-one tensor decomposition principle to derive new image bases that capture both spatial and temporal redundancies. The two dimensional (2D) Principle Component Analysis (PCA) represented an image in its natural matrix form (*i.e.*, as a 2^{nd} order tensor) and projected it to principal components along both horizontal and vertical directions. A 2D subspace (manifold) learning that is based on graph embedding of data represented as tensors was presented in [5]. In [3], the bilinear subspace analysis was generalized to a higher-order multilinear PCA that maximizes data variance

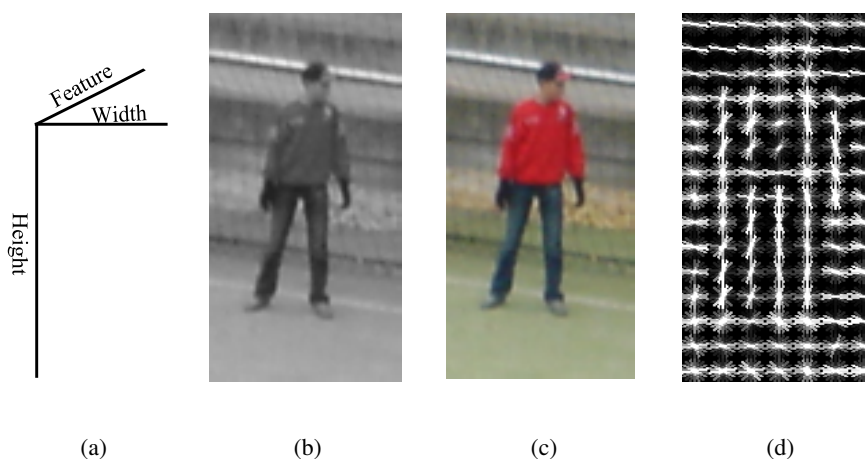


Fig. 1. Examples of visual data represented as tensors. (a) 3-mode tensor-based representations of visual objects. Examples include (b) a gray image (2^{nd} order tensor), (c) a color image (3^{rd} order tensor) and (d) a HoG descriptor (3^{rd} order tensor).

along each mode. Multilinear analysis was also successfully applied in multiple factor analysis by introducing the so-called "TensorFaces" in [6]. That work used the High-Order Singular Value Decomposition (HOSVD) in order to decompose an ensemble of images into basis images that capture the different underlying factors of variations, such as illumination, viewpoint and scene structures. Likewise, the Non-negative Tensor Factorization (NTF) [7], [8] in comparison with its vector counterpart–Non-negative Matrix Factorization (NMF)–not only preserves the local spatial relationship of images but is unique under certain mild conditions.

The unsupervised dimension reduction achieved when tensorial data are used, derives representations without taking into consideration the subsequent classification or recognition tasks. In order to achieve discriminant subspace learning from a set of labelled training examples, Linear Discriminant Analysis (LDA) was extended to Multilinear Discriminant Analysis (MDA) for face and gait recognition in [9], [10]. Such methods learn projection matrices along each mode of the tensor object in such a way that a discrimination criterion is maximized. The Discriminant Non-negative Matrix Factorization (DNMF) was also extended to Discriminant Non-negative Tensor Factorization (DNMF) in [11]. Similarly, in the general framework of Supervised Tensor Learning (STL) a method that learns one projection vector along each mode of a tensor was proposed in [12]. However, the use of only one projection vector for each mode may lead to loss of discriminative information. For this reason, the Support Vector Machines (SVMs) methodology, again within the STL framework, was extended to handle more than one projection directions in [1], [2], [13]. The work presented in [1], [13] regularizes the weights using matrix spectral norms (or matrix

trace, nuclear norm), favoring in that way low ranks for the weight matrix. The low-rank SVMs formulation proposed in [1] minimized the rank of the projection matrix instead of the classical maximum-margin criterion. The authors proposed an SVD-based iterative algorithm that, at each iteration, updated the parameters in the vector space using classic SVMs and reshaped them so as to reweigh the original tensor data. Although it is argued that rank one SVMs are better than their vector counterpart (SVMs), the proposed learning scheme does not reduce the computation complexity. This is as the parameters are updated via a classic SVMs optimizer using the original vector representation, without using a factorization of the weight tensor that would reduce the number of unknown parameters. In [13], the optimization problem is reformatted into a semi-definite one and solved using an interior point algorithm. However, these algorithms based on matrix rank regularization are not very direct when generalizing a matrix spectral norm to a higher order tensor. A biconvex formation, the so called bilinear SVMs, was proposed in [2], in the context of multi-task learning. The bilinear SVMs relax the orthogonality constraints on the columns of the weight matrix and a group coordinate descent learning scheme solves a classic SVMs subproblem for each tensor mode. Additionally, [1], [2], [13] mainly deal with matrices (2^{nd} order tensors).

In this paper, we study the regression problem using tensorial data and propose a Tensor Learning Model. To the best of our knowledge, this is the first work that addresses the regression problem using tensor representations. We adopt a linear regression model that is based on the inner product of the data tensor \mathcal{X} and a tensor \mathcal{W} of the (unknown) parameters. That is, $f(\mathcal{X}) = \langle \mathcal{X}, \mathcal{W} \rangle + b$. Two empirical risk functions are studied, namely the square loss and the ϵ -insensitive loss functions. The former leads to what we refer to as higher rank Tensor Ridge Regression (hrTRR) and the latter to higher rank Support Tensor Regression (hrSTR). In both cases, the unknown tensor \mathcal{W} is learned in an iterative manner, where at each iteration, using the Canonical (CANDECOMP)/Parallel Factors (PARAFAC) decomposition [14], the data from the input tensors \mathcal{X} are projected along a certain mode and the parameters that are associated to that mode are learned by solving a linear problem of reduced dimensionality. Contrary to previous tensor-based methods that deal either with low order tensors (*i.e.*, matrices [1], [2], [13]) or weight tensors of rank one [12], we derive the solutions for general tensors with multiple modes and in a way that allows simultaneous projections along multiple directions for each mode, thus dealing with higher-order tensors of higher rank.

Additionally, we consider two regularization terms, namely the Frobenius and the group sparsity norm. The use of Frobenius norm leads to the above mentioned hrTRR and hrSTR algorithms, requiring an a priori selection of the tensor rank R , typically obtained by cross-validation. The use of group sparsity norm however, allows us to enforce lower rank decomposition as a sparse constraint on rank one components and introduce in that way a novel algorithm that automatically determines the optimal tensor rank. This leads to the optimal rank TRR (orTRR) and optimal rank STR (orSTR) algorithms.

The contributions of this paper are the following:

- 1) We propose a framework for learning directly multilinear mappings from a tensorial input space to a continuous output space. To the best of our knowledge this is the first work that considers the regression problem within tensor-based supervised learning framework.
- 2) We propose to use Canonical (CANDECOMP)/Parallel Factors (PARAFAC) decomposition in order to learn simultaneously multiple projection vectors for each tensor mode. Our methodology handles high rank tensors and learns directly the parameters of the projections across each mode, contrary to other existing methods that first obtain the vector-based solution and then project it on the tensor subspace.
- 3) We investigate two regularized loss functions, namely the square loss and the ϵ -insensitive loss functions, leading to higher rank Tensor Ridge Regression (hrTRR) and higher rank Support Tensor Regression (hrSTR), respectively.
- 4) We investigate two regularization terms, namely the Frobenius and the group sparsity norms. The former requires the a priori selection of the tensor rank and leads to hrTRR and hrSTR, while the latter enables us to automatically estimate the tensor rank and leads to optimal rank TRR (orTRR) and optimal rank STR (orSTR), respectively.

The remainder of the paper is organized as follows. In Section II we present some useful notations regarding tensorial algebra that will be used throughout the paper. In Section III we describe the proposed Supervised Tensor Learning framework using two loss functions. More specifically, in Sections III-A and Section III-B we derive the higher rank Tensor Ridge Regression (hrTRR) and the higher rank Support Tensor Regression (hrSTR) algorithms using a regularized square loss and the ϵ -insensitive loss functions, respectively, as well as the Frobenius

norm regularization. The group-sparsity norm regularization framework that automatically estimates the rank in the learning process is presented in Section IV. In Section V, we report experimental results for head pose, human age and 3D body pose estimation using publicly available datasets. Finally, in Section VI we draw some conclusions.

II. NOTATIONS AND PRELIMINARIES

In this Section we will briefly describe some useful notations and concepts of tensorial algebra that will be used throughout the paper and are consistent with those presented in [14]. More specifically, matrices will be denoted by boldface capital letters, e.g., \mathbf{A} , vectors by boldface lowercase letters, e.g., \mathbf{a} and scalars by lowercase letters, e.g., a . Tensors are regarded as multi-dimensional arrays and will be denoted by Euler script calligraphic letters, e.g., $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$. The number of dimensions (also known as modes) M of a tensor denotes the order of the tensor. The i -th element of a vector $\mathbf{x} \in \mathbb{R}^I$ is denoted by x_i , $i = 1, 2, \dots, I$. In a similar way, the elements of an M -order tensor \mathcal{X} will be denoted by $x_{i_1 i_2 \dots i_n}$, $i_\ell = 1, 2, \dots, I_\ell$, $\ell = 1, 2, \dots, M$.

The d -mode matricization, also known as unfolding or flattening, of an M -order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$, denoted by $\mathbf{X}_{(d)} \in \mathbb{R}^{I_d \times (I_1 \dots I_{d-1} I_{d+1} \dots I_M)}$ or $\text{mat}_d(\mathcal{X})$, is the reordering of the tensor elements into a matrix, in such a way that the d -mode fibres become the columns of the final matrix. In the same way we define the vectorization of a tensor \mathcal{X} , denoted as $\text{vec}(\mathcal{X})$, by stacking its elements into a vector.

The d -mode product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_d}$, denoted as $\mathcal{X} \times_d \mathbf{U}$, is a tensor of size $I_1 \times I_{d-1} \times J \times I_{d+1} \times I_M$, elementwise defined as

$$(\mathcal{X} \times_d \mathbf{U})_{i_1 \dots i_{d-1} i_{d+1} \dots i_M} = \sum_{i_d=1}^{I_d} x_{i_1 \dots i_M} u_{j i_n}. \quad (1)$$

The equivalent unfolded expression is

$$\mathcal{Y} = \mathcal{X} \times_d \mathbf{U} \Leftrightarrow \mathbf{Y}_{(d)} = \mathbf{U} \mathbf{X}_{(d)}. \quad (2)$$

The d -mode vector product of a M order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ with a vector \mathbf{u} is defined likewise, but resulting in a $(M - 1)$ order tensor of size $I_1 \times I_{d-1} \times I_{d+1} \dots \times I_M$.

The multiplication in every mode is denoted by:

$$\mathcal{X} \times_1 U_1 \times_2 U_2 \dots \times_M U_M \triangleq \mathcal{X} \prod_{k=1}^M \times_k U_k, \quad (3)$$

while the multiplication in every mode except d is defined as:

$$\begin{aligned} & \mathcal{X} \times_1 U_1 \cdots \times_{d-1} U_{d-1} \times_{d+1} U_{d+1} \cdots \times_M U_M \\ & \triangleq \mathcal{X} \prod_{k=1, k \neq d}^M \times_k U_k \triangleq \mathcal{X} \bar{\times}_d U_d. \end{aligned} \quad (4)$$

The inner product of two tensors of the same size $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_M}$ is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} x_{i_1 \cdots i_M} y_{i_1 \cdots i_M}. \quad (5)$$

The *Frobenius* norm of a tensor is thus defined as $\|\mathcal{X}\|_{\text{Fro}} = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$. It can be shown that $\|\mathcal{X}\|_{\text{FRO}} = \|\mathbf{X}_{(d)}\|_{\text{FRO}} = \sqrt{\mathbf{X}_{(d)} \mathbf{X}_{(d)}^T}$ for any mode d .

The Canonical (CANDECOMP) / Parallel Factors (PARAFAC), referred as CP, decomposition factorizes an M order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_M}$ into a linear combination of a number R of rank-one tensors, written as:

$$\mathcal{X} \approx \sum_{r=1}^R u_r^{(1)} \circ u_r^{(2)} \cdots \circ u_r^{(M)} \triangleq \llbracket \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(M)} \rrbracket. \quad (6)$$

The operator "o" is the outer product of vectors and the factor matrices $\mathbf{U}^{(k)} = [\mathbf{u}_1^{(k)}, \dots, \mathbf{u}_R^{(k)}]$, of the size $I_k \times R, k = 1, 2, \dots, M$. In terms of unfolded tensors, the CP decomposition can be expressed as

$$\mathbf{X}_{(d)} = \mathbf{U}^{(d)} \left(\mathbf{U}^{(M)} \odot \dots \odot \mathbf{U}^{(d+1)} \odot \mathbf{U}^{(d-1)} \odot \dots \odot \mathbf{U}^{(1)} \right)^T \quad (7)$$

where \odot denotes the *Khatri-Rao product*. The rank of a tensor \mathcal{X} , denoted as $R = \text{rank}(\mathcal{X})$, is the smallest number of rank-one tensors whose sum is equal to \mathcal{X} .

III. GENERALIZED TENSOR LEARNING MODEL

A classic linear predictor in the vector space is given by

$$y = f(\mathbf{x}; \mathbf{w}, b) = \langle \mathbf{x}, \mathbf{w} \rangle + b, \quad (8)$$

where \mathbf{x} is the input data in a vector format, \mathbf{w} is the parameter/weight vector, b is the bias and y the regression output. Scalar output regression is considered here.

We extend the above mentioned classic linear predictor from the vector space to the tensor space as

$$y = f(\mathcal{X}; \mathcal{W}, b) = \langle \mathcal{X}, \mathcal{W} \rangle + b, \quad (9)$$

where $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ contains the input features, represented now as tensors with M modes, and \mathcal{W} is the weight tensor of equal number of modes and dimensions to the data tensor \mathcal{X} . The scalar b is the bias.

In terms of the unfolded tensor, Eqn.(8) and Eqn.(9) are equivalent. But if the input space is of high dimensionality, the over-fitting and high computational complexity problems appear. Unsupervised dimensionality reduction is usually applied prior to learning the weights. In this paper, in order to perform feature selection or dimensionality reduction and capture the underlying structure of the data, we constrain the weight tensor $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ to be a sum of R rank-one tensors, following the principle of CP composition. That is,

$$\mathcal{W} = \sum_{r=1}^R \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \dots \circ \mathbf{u}_r^{(M)} \triangleq \llbracket \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(M)} \rrbracket, \quad (10)$$

where $\mathbf{U}^{(j)} = [\mathbf{u}_1^{(j)}, \dots, \mathbf{u}_R^{(j)}]$. By substituting Eqn.(10) in Eqn.(9), we get

$$\begin{aligned} y &= \langle \mathcal{X}, \mathcal{W} \rangle + b \\ &= \langle \mathcal{X}, \sum_{r=1}^R \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \dots \circ \mathbf{u}_r^{(M)} \rangle + b \\ &= \sum_{r=1}^R \langle \mathcal{X}, \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \dots \circ \mathbf{u}_r^{(M)} \rangle + b \\ &= \sum_{r=1}^R \mathcal{X} \prod_{k=1}^M \times_k \mathbf{u}_r^{(k)} + b. \end{aligned} \quad (11)$$

As can be seen in Eqn.(11), the input features \mathcal{X} are projected along R directions for each mode k . These projections define the final subspace that is spanned by the learned R rank-one tensors. The fact that multiple projections are used reduces the loss of information that occurs when the projection is performed along only one direction [3]. Such projections can also be interpreted as a supervised dimension reduction or a feature selection scheme. This decomposition reduces the number of parameters that need to be estimated from $\prod_{k=1}^M I_k$ (i.e. the number of elements of the tensor \mathcal{W}) to $R \sum_{k=1}^M I_k$.

Given a set of labelled training set $\{\mathcal{X}_i, y_i\}_{i=1}^N$, where $\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ is an M -mode tensor and y_i are the associated scalar targets, we aim at learning the parameters $\Theta = \{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, (\mathbf{U})^M, b\}$ by minimizing the following regularized empirical risk:

$$L(\Theta) = \frac{1}{2} \sum_{i=1}^N l(y_i, f(\mathcal{X}_i; \Theta)) + \frac{\lambda}{2} \psi(\Theta) \quad (12)$$

where $l(\cdot)$ is a loss function and $\psi(\cdot)$ is a regularization term, that is introduced to control the model complexity so as to avoid over-fitting. Two types of empirical loss functions are considered in this paper for the problem of regression: the square loss and the ϵ -insensitive loss function. The former leads to what we refer to as higher rank Tensor Ridge Regression (hrTRR) and the latter to higher rank Support Tensor Regression (hrSTR). We also consider two types of regularization terms: the Frobenius norm, that requires the a priori selection of the rank R of the tensor weight (something that is achieved with exhaustive search) and the group sparsity norm, that allows the automatic selection of the decomposition rank as a part of the learning process. The former leads to the above mentioned hrTRR and hrSTR while the latter leads to the optimal rank Tensor Ridge Regression (orTRR) and optimal rank Support Tensor Regression (orSTR). The main steps of the proposed Tensor Learning for regression algorithm are summarized in Algorithm 1.

A. Higher Rank Tensor Ridge Regression

In order to proceed with the formulation of the higher rank Tensor Ridge Regression algorithm we consider the square loss empirical loss function $l = (y - f)^2/2$ and the Frobenius regularization term $\psi(\Theta) = \|\mathcal{W}\|_{\text{Fro}}^2$. Then, Eqn.(12) is reformulated as:

$$\begin{aligned}
 L(\{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(M)}\}, b) \\
 = \frac{1}{2} \sum_{i=1}^N \left(y_i - \langle \mathcal{X}_i, \llbracket \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(M)} \rrbracket \rangle - b \right)^2 \\
 + \frac{\lambda}{2} \left\| \llbracket \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(M)} \rrbracket \right\|_{\text{Fro}}^2. \quad (13)
 \end{aligned}$$

We can see in the above equation that both the data and the regularization terms contain products of the parameters $\mathbf{U}^1, \mathbf{U}^2, \dots, \mathbf{U}^M$. Thus, a closed-form solution like the one obtained for the vector-based ridge regression cannot be obtained. In order to tackle this problem, we follow a coordinate-descent approach, also known as alternative projections [10], [12]. This is an iterative method, where at each iteration we solve a convex optimization problem with respect to one subset of the parameter set $\{\mathbf{U}^1, \mathbf{U}^2, \dots, \mathbf{U}^M\}$ while all the other parameters are kept fixed. The procedure is repeated until a convergence criterion is met.

At each iteration we solve for the parameters $\mathbf{U}^{(j)}$ that are associated with the projection along mode j , while keeping the parameters $\{\mathbf{U}^{(k)}\}_{k=1, k \neq j}^M$ for the projections along all other modes fixed. Keeping $\{\mathbf{U}^{(k)}\}_{k=1, k \neq j}^M$

Algorithm 1 TENSOR LEARNING FOR REGRESSION

Input: The set of training tensors and their corresponding targets, that is $\{\mathcal{X}_i, y_i\}_{i=1}^N$.

Output: The weights $\{\mathbf{U}^1, \dots, \mathbf{U}^M\}$ and the bias term $b \in \mathbb{R}$ that minimize the objective function.

- 1: Initialize randomly $\{\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(M)}\}^{(0)}$.
- 2: **repeat**
- 3: $t \leftarrow t + 1$
- 4: **for** $k = 1$ to M **do**
- 5: Solve with respect to $\mathbf{U}^{(k)}|_{(t)}$:
 For hrTRR solve Eqn.(14) or Eqn.(18);
 For hrSTR solve Eqn.(21);
 For orTRR, solve Eqn.(28);
 For orSTR, solve Eqn.(29).
- 6: **end for**
- 7: **if** orTRR and orSTR **then**
- 8: Update the the parameter $\boldsymbol{\eta}$ given by Lemma 1
- 9: Prune the columns $\mathbf{U}_{:r}^{(m)}$ of factor matrices,
 $m \in \{1, 2, \dots, M\}, r \in \{j | \eta_j \leq \epsilon, j = 1, 2, \dots, R\}$
- 10: **end if**
- 11: **until** $\|\mathcal{W}^{(t)} - \mathcal{W}^{(t-1)}\| / \|\mathcal{W}^{(t-1)}\| \leq \varepsilon$ or $t \geq T_{\max}$

fixed, the $\widehat{\mathbf{U}}^{(j)}$ that minimizes Eqn.(13) is the one that minimizes

$$L_j(\mathbf{U}^{(j)}, b) = \underbrace{\frac{1}{2} \sum_{i=1}^N \left(y_i - \text{Tr}(\mathbf{U}^{(j)} \mathbf{U}^{(-j)\text{T}} \mathbf{X}_{i(j)}^{\text{T}}) - b \right)^2}_{l_j(\mathbf{U}^{(j)}, b)} + \underbrace{\frac{\lambda}{2} \text{Tr}(\mathbf{U}^{(j)} \mathbf{U}^{(-j)\text{T}} \mathbf{U}^{(-j)} \mathbf{U}^{(j)\text{T}})}_{\Omega_j(\mathbf{U}^{(j)})} \quad (14)$$

where $\mathbf{U}^{(-j)} = \mathbf{U}^{(M)} \odot \dots \odot \mathbf{U}^{(j-1)} \odot \mathbf{U}^{(j+1)} \dots \odot \mathbf{U}^{(1)}$ and Tr refers to the trace operator.

The reduced regularized least square optimization problem defined above can be solved in a closed-form for the

bias term b (i.e., $b = \frac{1}{N} \sum_{i=1}^N (y_i - \text{Tr}(\mathbf{U}^{(j)} \tilde{\mathbf{X}}_{i(j)}^T))$), but not for $\mathbf{U}^{(j)}$. To solve for $\mathbf{U}^{(j)}$ we need to resort to gradient-descent-style methods, e.g., a BFGS quasi-Newton optimizer. The gradient of the objective L_j with respect to $\mathbf{U}^{(j)}$ is given by:

$$\begin{aligned} \frac{\partial L_j}{\partial \mathbf{U}^{(j)}} = & - \sum_{i=1}^N (y_i - \text{Tr}(\mathbf{U}^{(j)} \tilde{\mathbf{X}}_{i(j)}^T) - b) \tilde{\mathbf{X}}_{i(j)} \\ & + \lambda \mathbf{U}^{(j)} \mathbf{U}^{(-j)T} \mathbf{U}^{(-j)} \end{aligned} \quad (15)$$

where $\tilde{\mathbf{X}}_{i(j)} = \mathbf{X}_{i(j)} \mathbf{U}^{(-j)}$. The proof can be found in Appendix A.

The partial derivative with respect to the bias is given here without proof:

$$\frac{\partial L_j}{\partial b} = - \sum_{i=1}^N (y_i - \text{Tr}(\mathbf{U}^{(j)} \tilde{\mathbf{X}}_{i(j)}^T) - b). \quad (16)$$

Note that in Eqn.(15), the $R \times R$ matrix $\mathbf{U}^{(-j)T} \mathbf{U}^{(-j)}$ by which the unknowns $\mathbf{U}^{(j)}$ is multiplied introduces cross terms that do not allow the vectorization of Eqn.(15) with respect to the unknowns $\mathbf{U}^{(j)}$. Thus, we cannot obtain closed-form solutions for the subproblem. Alternatively, we can use a separable regularization term

$$\Omega(\mathcal{W}) = \sum_{k=1}^M \|\mathbf{U}^{(k)}\|_{\text{Fro}}^2, \quad (17)$$

which can provide a closed-form solution for $\mathbf{U}^{(j)}$ and b when the unknowns are written in a vectorized form. More specifically, since $\|\mathbf{U}^{(j)}\|_{\text{Fro}}^2 = \|\text{vect}(\mathbf{U}^{(j)})\|^2$, and $\text{Tr}(\mathbf{U}^{(j)} \tilde{\mathbf{X}}_{i(j)}^T) = [\text{vect}(\mathbf{U}^{(j)})]^T [\text{vect}(\tilde{\mathbf{X}}_{i(j)})]$ it can be easily shown that the closed form solution can be derived as

$$\hat{\mathbf{u}}^{(j)} = (\Phi^T \Phi + \lambda \mathbf{I})^{-1} \Phi^T \mathbf{y}, \quad (18)$$

where $\hat{\mathbf{u}}^{(j)} = [\text{vect}(\hat{\mathbf{U}}^{(j)})^T b]^T$ is the vector of the unknowns, $\mathbf{y} = [y_1, \dots, y_N]^T$ are the targets and the i^{th} row of the matrix Φ is $[\text{vect}(\tilde{\mathbf{X}}_{i(j)})^T 1]$.

B. Higher Rank Support Tensor Regression

In this section we will derive the maximum-margin solution to the tensor-based regression problem. To this end, we consider the ϵ -insensitive loss function $l = \max(0, |y - f| - \epsilon)$ and the Frobenius regularization term $\psi(\Theta) = \|\mathcal{W}\|_{\text{Fro}}^2$. Following the same reasoning behind the SVR methodology we formulate the following optimization

problem:

$$\min_{\mathcal{W}, b, \xi, \hat{\xi}} \frac{1}{2} \|\mathcal{W}\|_{\text{Fro}}^2 + C \sum_{i=1}^N (\xi_i + \hat{\xi}_i), \quad (19a)$$

$$s.t. -y_i + \langle \mathcal{X}_i, \mathcal{W} \rangle + b \geq \epsilon + \hat{\xi}_i,$$

$$y_i - \langle \mathcal{X}_i, \mathcal{W} \rangle - b \leq \epsilon + \xi_i,$$

$$\epsilon \geq 0, \xi_i \geq 0, \hat{\xi}_i \geq 0, \forall i = 1, 2, \dots, N. \quad (19b)$$

We should note here that the above defined optimization problem constitutes a reformulation of Eqn.(12). In order to solve Eqn.(19), we optimize the cost function with respect to $\mathbf{U}^{(j)}$ while keeping the other $\mathbf{U}^{(k)}, k \neq j$ fixed.

That is,

$$\min_{\mathbf{U}^{(j)}, b, \xi, \hat{\xi}} \frac{1}{2} \text{Tr} \left(\mathbf{U}^{(j)} \mathbf{U}^{(-j)\text{T}} \mathbf{U}^{(-j)} \mathbf{U}^{(j)\text{T}} \right) + C \sum_{i=1}^N (\xi_i + \hat{\xi}_i), \quad (20a)$$

$$s.t. -y_i + \text{Tr}(\mathbf{U}^{(j)} \mathbf{U}^{(-j)\text{T}} \mathbf{X}_{i(j)}^{\text{T}}) + b \geq \epsilon + \hat{\xi}_i,$$

$$y_i - \text{Tr}(\mathbf{U}^{(j)} \mathbf{U}^{(-j)\text{T}} \mathbf{X}_{i(j)}^{\text{T}}) - b \leq \epsilon + \xi_i,$$

$$\epsilon \geq 0, \xi_i \geq 0, \hat{\xi}_i \geq 0, \forall i = 1, 2, \dots, N. \quad (20b)$$

If we denote by $\mathbf{B} = \mathbf{U}^{(-j)\text{T}} \mathbf{U}^{(-j)}$, $\tilde{\mathbf{U}}^{(j)} = \mathbf{U}^{(j)} \mathbf{B}^{\frac{1}{2}}$ and $\tilde{\mathbf{X}}_{i(j)} = \mathbf{X}_{i(j)} \mathbf{U}^{(-j)} \mathbf{B}^{\frac{1}{2}}$, the optimization problem in Eqn.(20) can then be rewritten as:

$$\min_{\tilde{\mathbf{U}}^{(j)}, b, \xi, \hat{\xi}} \frac{1}{2} \text{Tr}(\tilde{\mathbf{U}}^{(j)} \tilde{\mathbf{U}}^{(j)\text{T}}) + C \sum_{i=1}^N (\xi_i + \hat{\xi}_i), \quad (21a)$$

$$s.t. -y_i + \text{Tr}(\tilde{\mathbf{U}}^{(j)} \tilde{\mathbf{X}}_{i(j)}^{\text{T}}) + b \geq \epsilon + \hat{\xi}_i,$$

$$y_i - \text{Tr}(\tilde{\mathbf{U}}^{(j)} \tilde{\mathbf{X}}_{i(j)}^{\text{T}}) - b \leq \epsilon + \xi_i,$$

$$\epsilon \geq 0, \xi_i \geq 0, \hat{\xi}_i \geq 0, \forall i = 1, 2, \dots, N. \quad (21b)$$

If we vectorize $\tilde{\mathbf{U}}^{(j)}, \tilde{\mathbf{X}}_{i(j)}$, since

$$\text{Tr}(\tilde{\mathbf{U}}^{(j)} \tilde{\mathbf{U}}^{(j)\text{T}}) = \|\text{vec}(\tilde{\mathbf{U}}^{(j)})\|^2,$$

$$\text{Tr}(\tilde{\mathbf{U}}^{(j)} \tilde{\mathbf{X}}_{i(j)}^{\text{T}}) = [\text{vec}(\tilde{\mathbf{U}}^{(j)})]^{\text{T}} [\text{vec}(\tilde{\mathbf{X}}_{i(j)})], \quad (22)$$

then the problem in Eqn.(21) can be easily solved using a typical SVMs/SVR optimizer. Once $\tilde{\mathbf{U}}^{(j)}$ is obtained, we can solve for $\mathbf{U}^{(j)}$ as

$$\mathbf{U}^{(j)} = \tilde{\mathbf{U}}^{(j)} \mathbf{B}^{-\frac{1}{2}}. \quad (23)$$

IV. OPTIMAL RANK TENSOR LEARNING MODEL

In hrTRR and hrSTR, the Frobenius norm regularization was used, requiring the a priori selection of the tensor rank. This task is usually performed using cross-validation, something that is achieved with an exhaustive search for the optimal rank R . A low model complexity is in general ensured by obtaining a lower generalization error bound, implying that the weight tensor \mathcal{W} should have a lower rank. However, the optimization of the tensor decomposition rank is NP-hard [14]. To this end, we regularize the decomposed components of the weight tensor with $l_{1,2}$ norm [15], [16] favoring in that way the sparse-column characterization of the factor matrices of \mathcal{W} , as depicted in Fig 2.

This group sparsity norm regularization can be written as

$$\psi(\mathcal{W}) = \sum_{r=1}^R \left(\sum_{m=1}^M \|\mathbf{U}_{:,r}^{(m)}\|_2^2 \right)^{\frac{1}{2}} \quad (24)$$

where $\mathbf{U}_{:,r}^{(m)}$ denotes the r^{th} column of the matrix $\mathbf{U}^{(m)}$. By applying this regularization we force the same r^{th} columns of the factor matrices $\mathbf{U}^{(m)}$, $m = 1, 2, \dots, M$ to simultaneously become zero. In order to achieve that, we further minimize the upper bound based on the following variational inequality over the regularization term [15]:

Lemma 1.

$$\begin{aligned} \psi(\mathcal{W}) &= \sum_{r=1}^R \left(\sum_{m=1}^M \|\mathbf{U}_{:,r}^{(m)}\|_2^2 \right)^{\frac{1}{2}} \\ &= \min_{\boldsymbol{\eta} \in \mathbb{R}^R} \frac{1}{2} \sum_{r=1}^R \frac{\sum_{m=1}^M \|\mathbf{U}_{:,r}^{(m)}\|_2^2}{\eta_r} + \frac{1}{2} \|\boldsymbol{\eta}\|_1. \end{aligned} \quad (25)$$

The minimum is then obtained for $\eta_r = \left(\sum_{m=1}^M \|\mathbf{U}_{:,r}^{(m)}\|_2^2 \right)^{\frac{1}{2}}$, $\forall r = 1, 2, \dots, R$. The proof is provided in Appendix C.

By substituting Eqn.(25) into Eqn.(12) we formulate the following optimization problem that corresponds to what

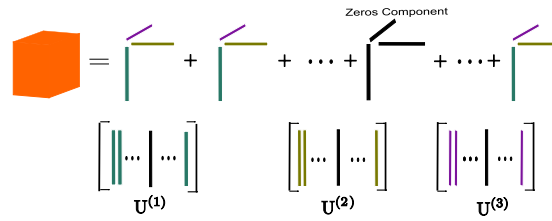


Fig. 2. Schematic description of sparse rank decomposition

we will refer to as the Optimal Rank Tensor Learning model:

$$\begin{aligned} \min_{\mathbf{U}^{(m)}|_{m=1}^M, b, \boldsymbol{\eta}} L(\mathbf{U}^{(m)}|_{m=1}^M, b, \boldsymbol{\eta}) \\ = \frac{1}{2} \sum_{i=1}^N l(y_i, \langle \mathcal{X}_i, \llbracket \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(M)} \rrbracket \rangle + b) \\ + \frac{\lambda}{4} \left(\sum_{r=1}^R \frac{\sum_{m=1}^M \|\mathbf{U}_{:,r}^{(m)}\|_2^2}{\eta_r} + \|\boldsymbol{\eta}\|_1 \right). \end{aligned} \quad (26)$$

If we adopt the block coordinate descent algorithm (its analytic form is provided in Appendix D) and keep $\{\mathbf{U}^{(m)}\}_{m=1}^M|_{m \neq j}$ fixed, the subproblem for $\mathbf{U}^{(j)}$ is given by

$$\begin{aligned} L_j(\mathbf{U}^{(j)}, b) = \frac{1}{2} \sum_{i=1}^N l \left(y_i, \text{Tr}(\mathbf{U}^{(j)} \mathbf{U}^{(-j)\text{T}} \mathbf{X}_{i(j)}^{\text{T}}) + b \right) \\ + \frac{\lambda}{4} \text{Tr}(\mathbf{U}^{(j)} \boldsymbol{\Lambda} \mathbf{U}^{(j)\text{T}}) \end{aligned} \quad (27)$$

where $\mathbf{U}^{(-j)} = \mathbf{U}^{(M)} \odot \dots \odot \mathbf{U}^{(j-1)} \odot \mathbf{U}^{(j+1)} \dots \odot \mathbf{U}^{(1)}$, and $\boldsymbol{\Lambda} = \text{diag}(\frac{1}{\eta_1}, \dots, \frac{1}{\eta_R})$. This optimization scheme can be seen as a variant of an iterative reweighted least square algorithm.

Regarding the Optimal Rank TRR (orTRR) case we consider the loss square empirical loss function. Then the subproblem (27) is a weighted least square problem which can be solved by a closed-form solution as

$$\hat{\mathbf{u}}^{(j)} = \left(\Phi^{\text{T}} \Phi + \frac{\lambda}{2} \tilde{\boldsymbol{\Lambda}} \right)^{-1} \Phi^{\text{T}} \mathbf{y}, \quad (28)$$

where $\tilde{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda} \otimes \mathbf{I}_{I_j \times I_j}$.

Regarding the Optimal Rank STR (orSTR) case we consider the ϵ -insensitive loss function and rewrite the

subproblem (27) into the following equivalent form

$$\min_{\mathbf{U}^{(j)}, b, \xi} \frac{1}{2} \text{Tr} \left(\mathbf{U}^{(j)} \Lambda \mathbf{U}^{(j)\text{T}} \right) + C \sum_{i=1}^N (\xi_i + \hat{\xi}_i), \quad (29a)$$

$$s.t. -y_i + \text{Tr}(\mathbf{U}^{(j)} \mathbf{U}^{(-j)\text{T}} \mathbf{X}_{i(j)}^{\text{T}}) + b \geq \epsilon + \hat{\xi}_i,$$

$$y_i - \text{Tr}(\mathbf{U}^{(j)} \mathbf{U}^{(-j)\text{T}} \mathbf{X}_{i(j)}^{\text{T}}) - b \leq \epsilon + \xi_i,$$

$$\epsilon \geq 0, \xi_i \geq 0, \hat{\xi}_i \geq 0, \forall i = 1, 2, \dots, N, \quad (29b)$$

that is similar to the Eqn.(21) if we replace \mathbf{B} with Λ , and hence can be solved in a similar way.

V. EXPERIMENTAL RESULTS

In order to investigate the performance of the proposed tensor-based regression schemes we conducted experiments using real, publicly available data for the problems of head pose, human age and body pose estimation.

More specifically, we investigated and compared the performance of the vector-based and tensor-based algorithms with respect to the following:

- 1) The influence of the rank R of the weight tensor that controls the number of simultaneous projections along each mode.
- 2) The influence of the values of the regularization parameters and more specifically of the parameter λ for hrTRR and RR and of C for hrSTR and SVR.
- 3) The effectiveness of the automatic procedure for finding the optimal rank for orTRR and orSTR algorithms.

A. Head Pose Estimation

Regarding head pose estimation, we carried experiments using three publicly available datasets, the IDIAP [17], the Boston University [18] and the Pointing'04 [19] datasets.

1) *IDIAP dataset*: The IDIAP database comprises of 23 video sequences involving people engaged in natural activities. In total, 16 different subjects participate in the video database. We used a subset of the dataset, containing only the videos from meeting scenarios 1, 3 and 4, splitted in training and testing sets following the protocol described in [17]. The ground truth provided is in the form of pan, tilt and roll angles (i.e. Euler angles with respect to the camera coordinate system) for each frame of the video sequences. A face detector was used to

extract the bounding box of each face in every video frame. All the acquired image regions were resized to 40×30 pixels. Two types of features were extracted, the normalized pixel intensity and the log-Gabor features. Each of the images formed a 40×30 2nd order tensor that was used as input to the proposed tensor-based algorithms. The head pose Euler angles $\{\alpha, \beta, \gamma\}$, corresponding to pan, tilt and roll were calculated from the rotation matrix of the head configuration with respect to the camera position. We report the mean absolute angular error of the pointing vector defined by $\{\alpha, \beta, \gamma\}$ and the mean absolute error for each of $\{\alpha, \beta, \gamma\}$ [17].

2) *Boston University dataset*: The Boston University (BU) dataset consists of 45 video sequences, depicting 5 subjects performing 9 different motions under uniform illumination in a standard office setting. The training and testing subsets are chosen in such a way so as to ensure that the subjects in the testing set do not appear in the training set (Table I).

TABLE I

BOSTON UNIVERSITY HEAD POSE DATA

Subject	Training Sequences	Testing Sequences
jam	1, 2, 3	4, 5, 6, 7, 8, 9
jim	7, 8, 9	1, 2, 3, 4, 5
llm		1,2,3,4,5,6,7,8,9
ssm	2, 8	1, 3, 4, 5, 6, 7, 9
vam	4, 5, 6	1, 2, 3, 7, 8, 9

The features extracted were the same as with the IDIAP dataset. The head pose Euler angles $\{\alpha, \beta, \gamma\}$, corresponding to roll, yaw and pitch in the BU dataset, were also calculated and we report the mean absolute angular and mean absolute errors of the pointing vector.

3) *Pointing4*: The Pointing'04 head pose database contains a variety of head poses ranging from -90 to 90 degrees in both horizontal and vertical directions. The data set comprises of 15 subjects of various skin colors, with or without glasses, each one performing 13 pose variations in horizontal and 7 in vertical, as well as the two extreme cases of the vertical 90 and -90 degrees, to a total of 2790 images.

The bounding box for each image was provided. All images were resized to be of size 32×32 . Subsequently,

a log-Gabor filter with 4 scales and 8 orientations was applied at each image and the Local Binary Patterns were calculated. Each image was divided into non-overlapping rectangular sub-regions of size 8×8 and a set of histograms were computed for each subregion (considering 59 bins). Finally each histogram was organized in a 3-D tensor of dimension $944 \times 4 \times 8$.

4) *Head pose estimation results:* We first studied the convergence in terms of the training errors with respect to the number of outer iterations required for the below mentioned five training schemes: the closed-form solution updates in hrTRR (Eq. 14), the BFGS quasi-Newton solver for the hrTRR (B-hrTRR) (Eq. 18), the Libsvm [20] optimizer for the hrSTR (Eq. 21), the orTRR and the orSTR.

The unknown parameters are randomly initialized. The training error plotted against the number of outer iterations is depicted in Fig. 3, where the convergence of all five training processes is shown. As we can see, the hrTRR, B-hrTRR and hrSTR all converge quite fast, while the orTRR and orSTR require more iterations to reach convergence. The difference in the reported values is due to different values for the λ parameter used, chosen by cross-validation.

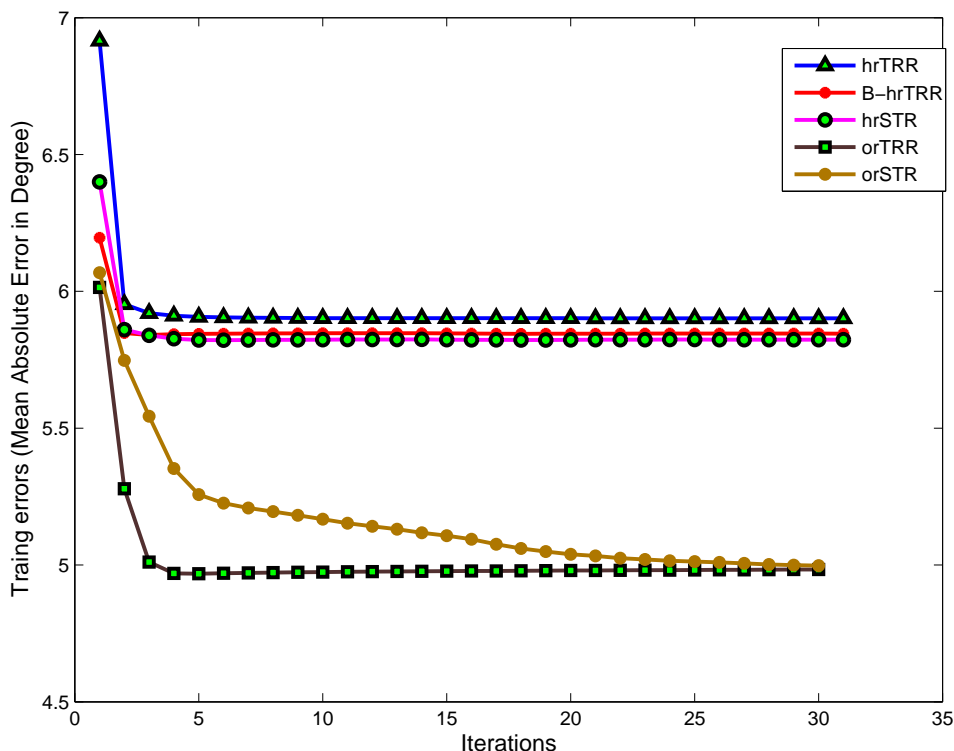


Fig. 3. The training error for the tilt angle on the IDIAP dataset.

Subsequently, we investigate the sensitivity with respect to the initialization of \mathcal{W} in the hrTRR and hrSTR

schemes. The results acquired for different initializations (either random or using the values obtained from the corresponding vector-based problems) were almost identical.

In order to investigate on the significance of the rank R and the regularization parameters λ, C for hrTRR and hrSTR, we report the testing errors of the pointing vector against the rank and the regularization parameters in Fig. 4. It can be easily seen that the proposed tensor-based decomposition of parameters avoids over-fitting while outperforming its vector-based counterparts, especially at low regularization levels (*i.e.*, small values for λ for hrTRR and high values for C for hrSTR). Tables II and III report the lowest testing errors that the different regressors achieved (RR, SVR, hrTRR, hrSTR, orTRR and orSTR) in Fig. 4. From now onwards, the best results will be highlighted in bold. The automatic rank selection procedure provided us with the same rank as the one selected by cross validation (equal to 1 for the IDIAP and 3 for the BU datasets). We will therefore report the value of the optimal rank from now onwards. Additionally, one can observe here that the hrTRR and hrSTR algorithms provide different results than those of orTRR and orSTR. This is due to the different regularization term used for the formulation of the orTRR and orSTR problems.

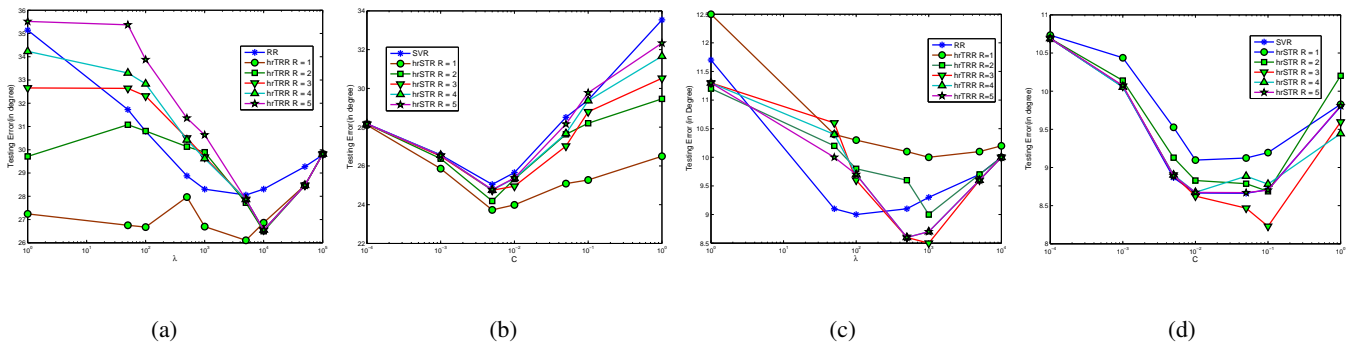


Fig. 4. Testing error for different rank R and regularization parameters λ, C for (a) TRR, (b) STR on IDIAP, and (c) TRR, (d) STR on BU

The rank one components $\{u_r^{(1)} \circ u_r^{(2)} \circ \dots \circ u_r^{(M)}\}_{r=1}^R$ of the weight tensor \mathcal{W} that are obtained at convergence of hrTRR and hrSTR are visualized as gray images in Fig. 5. We visualize the solutions that we obtain for each of the three output angles $\{\alpha, \beta, \gamma\}$ for the BU dataset. For the tensor-based methods we set $R = 3$. It is clear that the tensor weights for the cases of hrTRR and hrSTR form better and more clear spatial patterns that can be interpreted as filters applied at the input image, when compared to the equivalent weights acquired for RR and SVR.

TABLE II

ANGULAR ERROR FOR THE IDIAP DATASET.

	RR	SVR	hrTRR	hrSTR	orTRR	orSTR
pan	25.2	21.9	22.7	20.7	21.51	20.9
tilt	8.5	9.0	9.4	9.0	8.7	8.7
roll	11.1	11.3	10.7	10.0	10.5	10.1
Pointing	28.1	25.0	26.0	23.6	24.5	23.7

TABLE III

ANGULAR ERROR FOR THE BU DATASET.

	RR	SVR	hrTRR	hrSTR	orTRR	orSTR
Roll	6.1	5.8	5.8	5.6	5.7	5.3
Yaw	5.4	5.3	5.0	4.9	5.0	5.3
Pitch	5.1	4.8	5.4	4.8	5.4	4.9
Pointing	9.0	8.7	8.5	8.2	8.5	8.5

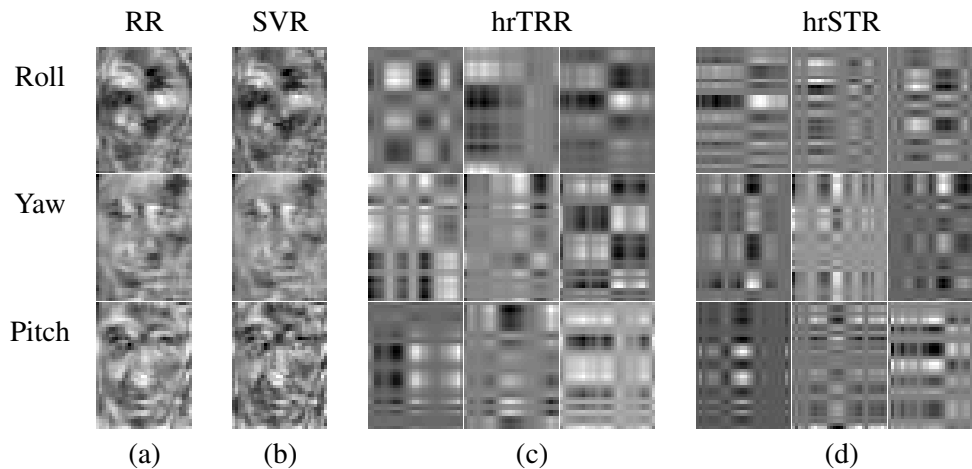


Fig. 5. The images of weights obtained by (a) RR, (b) SVR. Three rank one weight tensors for (c) hrTRR and (d) hrSTR for the BU dataset.

As mentioned before, the projections along the tensor modes that are performed by the matrices $U^{(d)}$ can be interpreted as a form of dimension reduction, or feature extraction. In the proposed method, this is performed in a supervised manner for regression. Below we compare with the results obtained if we perform unsupervised dimension reduction either in the vector (PCA) or in the tensor (MPCA [3]) representation before a classic vector-

based regression algorithm (RR or SVR) is trained. The number of PCA (or MPCA) components are chosen so that around 97% of the energy is preserved. The results are summarized in Table IV, where it is clear that the application of the unsupervised dimension reduction methods does not lead to results comparable to the ones obtained by the proposed direct tensor-based regression.

TABLE IV

COMPARISON OF TESTING ERRORS OF POINTING VECTOR OF OUR SUPERVISED MODEL AND UNSUPERVISED MODELS

Features	Dataset	RR	SVR	PCA		MPCA		hrTRR	hrSTR
				+RR	+SVR	+RR	+SVR		
Intensity	IDIAP	28.1	25.0	28.0	25.6	28.1	25.6	26.0	23.7
	BU	9.0	8.7	9.0	8.6	9.0	8.7	8.5	8.2
Log-Gabor	IDIAP	33.0	32.3	31.68	30.8	35.2	30.8	27.06	25.6
	BU	10.3	10.0	10.0	10.1	9.8	9.8	9.2	8.4

The angular error for pan and tilt, as well as their average value obtained for the Pointing'04 dataset for RR, SVR, hrTRR, hrSTR, orTRR and orSTR are given in Table V. The optimal rank selected was equal to 4. As we can see the results acquired with hrSTR outperform the state of art results presented in [21] .

B. Human age estimation

We also conducted experiments for the estimation of age from facial images, namely the age estimation problem. To this end, we used one publicly available dataset, the FG-NET dataset [22]. The FG-NET dataset comprises of 1002 facial images of 82 people being from 0 to 69 years old. A set of 68 labeled facial landmarks characterizing

TABLE V

POINTING DATABASE

Range	[21]	RR	SVR	hrTRR	hrSTR	orTRR	orSTR
horizontal	9.37	11.63	15.01	7.74	8.45	9.05	8.08
vertical	7.84	12.13	13.97	8.51	7.17	8.78	8.02
Average	8.61	11.88	14.49	8.12	7.81	8.91	8.05

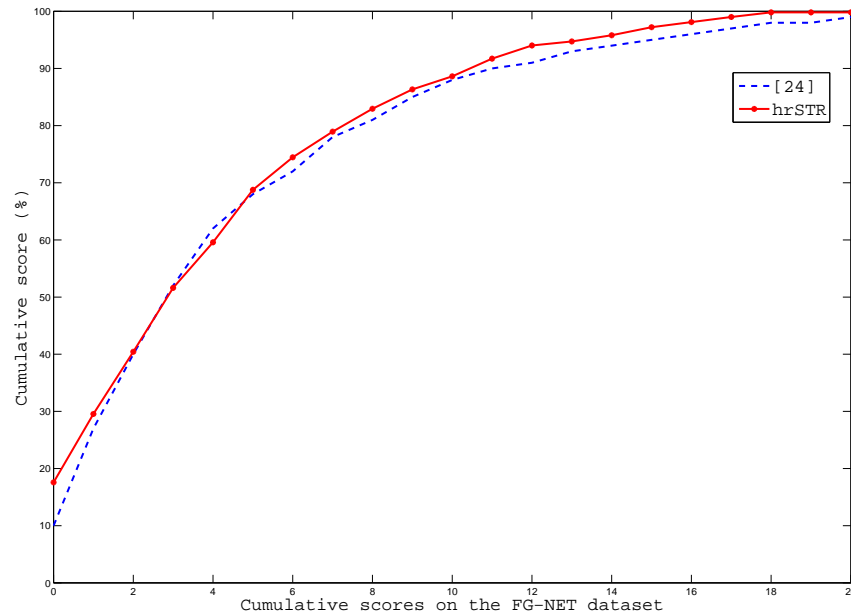
TABLE VI

MAES ON THE FG-NET DATABASE FOR RR, SVR, hrTRR, hrSTR, orTRR AND orSTR.

Range	images	[23]	[22]	RR	SVR	hrTRR	hrSTR	orTRR	orSTR
0-9	371	2.3	3.19	3.58	8.77	4.73	2.82	4.27	4.83
10-19	339	4.86	3.90	6.48	1.28	2.62	1.75	1.84	2.39
20-29	144	4.02	4.29	16.01	10.80	1.96	1.00	1.59	0.37
30-39	70	7.32	9.17	25.90	20.64	7.47	8.61	6.89	8.48
40-49	46	15.24	13.76	35.72	30.52	13.84	17.87	16.60	14.47
50-59	15	22.20	20.06	45.00	39.77	25.60	25.71	25.52	24.98
60-69	8	55.28	32.25	33.15	50.10	29.19	36.14	29.56	31.90
Average	1002	4.95	4.96	10.38	9.07	4.69	3.88	4.25	4.53

shape features are also provided for each facial image. The protocol followed for experiments was the leave-one-person-out. In our experiments the images were aligned using the set of 68 points. More precisely, we triangulated the landmarks using Delauney triangulations and then all images were normalized to a template using piece-wise affine transform. The features extracted were the same as in the case of the Pointing'04 dataset. The performance of our algorithms was measured using the mean absolute error (MAE) and the cumulative score (CS). The MAE is defined as the average of the absolute errors between the estimated ages and the ground truth $MAE = \sum_{k=1}^N |\overline{Age}_k - Age_k|/N$, where with \overline{Age}_k we denote the estimated age for a test image k , with Age_k its ground truth age and with N the total number of test images. The estimation accuracy can be estimated by the cumulative score (CS), defined as $CS(j) = N_{e \leq j}/N \times 100\%$, where $N_{e \leq j}$ is the number of test images on which the estimator makes an absolute error non-higher than j years. The optimal rank selected was equal to 7. The MAE results obtained using RR, SVR, hrTRR, hrSTR, orTRR and orSTR are summarized in Table VI and the cumulative score achieved is depicted in Fig. 6, where we also plot the state of the art results [23].

As can be seen from the results in Table VI, tensors significantly outperform vectors, providing at the same time better than the state of the art results. Furthermore, one may notice here that when many examples are available (and



(a)

Fig. 6. Cumulative scores on the FG-NET dataset.

higher rank tensor representations are used), such as in the age categories 0-9, 10-19 and 20-29, hrSTR significantly outperforms not only the equivalent vector cases but also the hrTRR algorithm.

C. Human Pose Estimation

For the 3D human pose estimation experiments, we used the HumanEva-I training and validation sets [24]. More specifically, we conducted experiments for 3 subjects and for the following four actions: walk, jog, gestures and box [25], [26], following the protocol presented in [24].

Direct mappings between the image features and each of the target outputs (3D positions of 19 body joints) are learned. The features were based on Histograms of Oriented Gradients (HoGs) that were extracted from the silhouette obtained after background subtraction [25]. In order to extract HoGs, the bounding box containing the silhouette was first divided into 6×5 blocks. The gradient orientations in each block were quantized in 9 orientation bins. Thus, each image was represented as a $6 \times 5 \times 9$ tensor of order 3. The 3D human body pose was encoded in a 57-dimensional vector \mathbf{y} , corresponding to the 3D positions of 19 joint centers. Each of the joint positions was defined relatively to the torso “torsoDistal”. The estimation error was defined in millimeters and measured as

the average Euclidean distance between the estimated joints positions $\mathbf{y}_i^{(n)} \in \mathbb{R}^3$ and the ground truth over all M joints and N frames [24]. That is, $Errr = \frac{1}{N} \sum_{n=1}^N \frac{1}{M} \sum_{i=1}^M \|\mathbf{y}_i^{(n)} - \bar{\mathbf{y}}_i^{(n)}\|$. The experiments were performed by training an individual predictor for each output.

1) *Globally trained model*: Due to the non-linear relationship between an observation and its corresponding pose, we first performed training and testing on the same subject for every action, by using the publicly available splitting into two sets of roughly equal size.

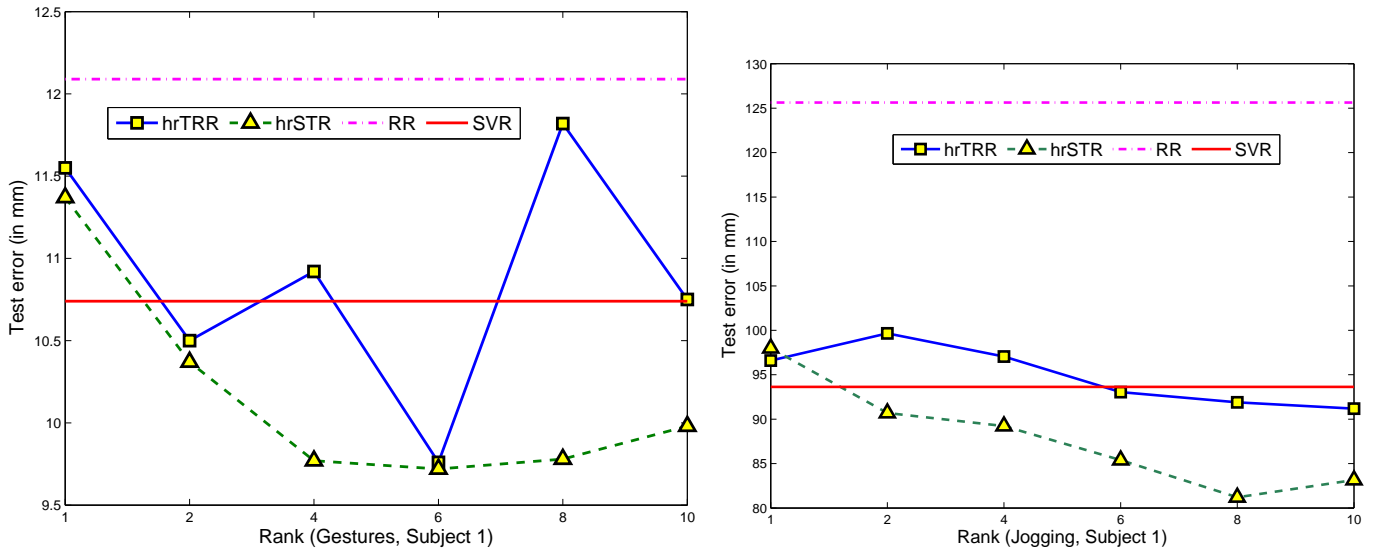


Fig. 7. Mean Angular Error vs rank for gestures and jogging of Subject 1.

We experimented with different values for rank R , namely the ones in the set $\{1, 2, 4, 6, 8, 10\}$, and report results for the actions Gestures and Jogging in Fig. 7. For comparison, in each plot we show the baseline vector-based regressors (i.e. SVR and RR). The detailed results of average testing errors over 3 subjects are reported in Table VII.

In Fig. 8 we depict the estimated body model superimposed on two frames of the Gestures action sequence and a diagram depicting the average error for four points on the right arm. For this sequence, the body remains roughly still and the error is dominated by the estimation error of the joints of the moving arm. The average errors of the RR, hrTRR, SVR and hrSTR for those joints are 29.4, 25.5, 27.2 and 24.3, respectively, a result that clearly shows the superior performance of the tensor-based methods. Clearer improvements are observed for the body joints that can be better predicted by our simple linear model. Similar conclusions can be drawn for the other actions. In

particular we are able to recover with satisfactory accuracy the positions of the torso joints and the upper parts of the limbs (e.g. elbows and knees). For the “jog” sequence, the errors for RR, hrTRR, SVR and hrSTR for those joints are 107.0, 76.3, 78.3 and 68.6, respectively. However, our linear model shows its limitations in accurately estimating the position of the lower parts of the limbs (i.e. angles and wrists) for which the errors are greater than 130mm.

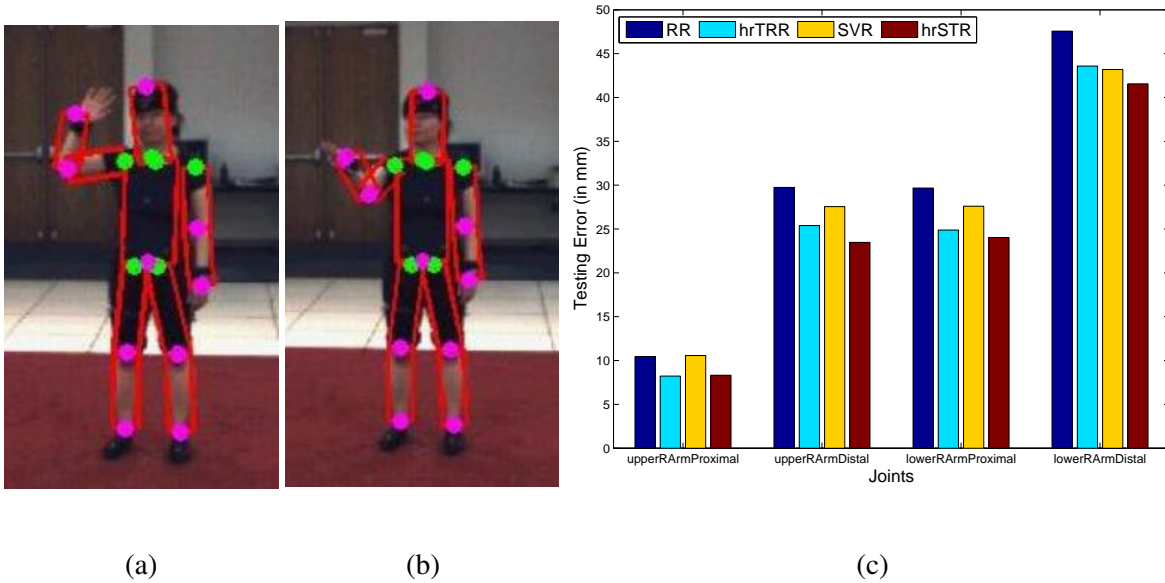


Fig. 8. “Gestures” action: (a)-(b) Estimated body model (c) Error for right arm joints.

2) *Locally trained model*: The global trained linear models are not very plausible to model the actual nonlinear relationship between the observations and the poses. Hence, we adopt a cluster-classification scheme to train a set of local linear models that piecewisely approximates the global nonlinear model. We cluster training samples in the pose space so that the samples with similar poses are assigned to one cluster. The clusters define several corresponding pose classes which can be discriminated by any multi-class classifier. In this work we use random forests [27], [28] for this purpose. A linear tensor-based regression model is then trained for each cluster. At the test stage, the clusters are chosen from the trained random forest and the estimation can be obtained by applying the corresponding trained tensor model. In this experiment, the cluster-classification-regression models are trained on the entire training subset. Table VIII shows the average testing errors over all three subjects of this local tensor-based regression models.

TABLE VII

MEAN TESTING ERRORS OF GLOBAL TRAINED MODELS

Action	RR	SVR	hrTRR	hrSTR	orTRR	orSTR
Walking	77.4	74.5	78.4	77.6	78.4	76.1
Jog	91.0	76.0	75.9	73.7	80.5	68.1
Gesture	98.2	63.6	79.4	63.0	70.0	66.8
Box	101.0	78.3	86.3	74.4	70.5	62.8
Average	91.9	73.1	80.0	72.2	74.9	68.5

TABLE VIII

TESTING ERRORS OF LOCAL MODELS.

Action	RR	SVR	hrTRR	hrSTR	orTRR	orSTR
Walking	82.9	64.6	68.5	64.7	66.0	61.9
Jog	68.8	62.2	66.1	60.8	68.5	59.2
Gesture	90.6	71.4	86.4	71.3	80.6	78.5
Box	104.	82.9	89.3	82.0	89.0	78.6
Average	86.6	70.3	77.6	69.7	76.1	69.6

D. Discussion

To sum up, from the experiments presented above someone can observe that:

- tensor-based methods consistently outperform their corresponding vector-based methods. When more complex features were used, results better than the state of the art were acquired. More specifically, for the head pose estimation problem in the Pointing'04 dataset and for the human age estimation problem, tensors improve the state of the art results by 9.29% and 27.58%, respectively.
- tensor-based methods are more robust both with respect to different values of the regularization parameters and with respect to different initializations of the weight parameters.
- the algorithms proposed for the automatic choice of the rank are successful in finding the optimal rank and in some occasions outperform the results acquired from their equivalent higher rank versions.

VI. CONCLUSIONS

In this paper we propose a novel generalized Supervised Multilinear Learning Model that deals with regression. The proposed method allows the simultaneous projections of an input tensor to more than one directions along each mode, exploiting the properties of the Canonical (CANDECOMP)/Parallel Factors (PARAFAC) decomposition. Two empirical risk functions are studied, namely the square loss and the ϵ -insensitive loss functions. These lead to the generalization of two well known regression schemes, namely Ridge Regression and Support Vector Regression, to their corresponding tensor-based regression methods, namely, higher rank Tensor Ridge Regression (hrTRR) and higher rank Support Tensor Regression (hrSTR). The above algorithms are formulated using as a regularization term the Frobenius norm. We also study the group sparsity norm, in order to achieve automatic tensor rank selection, thus formulating the equivalent optimal rank TRR (orTRR) and optimal rank STR (orSTR). Experiments performed using publicly available real data for the problems of head pose, human age and body pose estimation verified the superiority of the tensors-based algorithms when compared to the vector-based ones but also the efficiency of the proposed algorithms.

APPENDIX A

PROOF OF (15)

Proof: From the unfolded tensor equivalents, $\|\mathcal{W}\|_{\text{Fro}}^2 = \|\mathbf{W}_{(j)}\|_{\text{Fro}}^2 = \text{Tr}(\mathbf{W}_{(j)}\mathbf{W}_{(j)}^T)$, and $\langle \mathcal{X}, \mathcal{W} \rangle = \langle \mathbf{X}_{(j)}, \mathbf{W}_{(j)} \rangle = \text{Tr}(\mathbf{X}_{(j)}\mathbf{W}_{(j)}^T)$. Then, by substituting the rank-one tensor decomposition (Eqn.(7)) $\mathbf{W}_{(j)} = \mathbf{U}^{(j)} (\mathbf{U}^{(M)} \odot \dots \odot \mathbf{U}^{(j+1)} \odot \mathbf{U}^{(j-1)} \odot \dots \odot \mathbf{U}^{(1)})^T$ into these equivalents and by denoting $\mathbf{U}^{(-j)} = \mathbf{U}^{(M)} \odot \dots \odot \mathbf{U}^{(j-1)} \odot \mathbf{U}^{(j+1)} \dots \mathbf{U}^{(1)}$ we obtain that $\|\mathcal{W}\|_{\text{Fro}}^2 = \text{Tr}(\mathbf{U}^{(j)}\mathbf{U}^{(-j)T}\mathbf{U}^{(-j)}\mathbf{U}^{(j)T})$, and $\langle \mathcal{X}, \mathcal{W} \rangle = \text{Tr}(\mathbf{X}_{(j)}\mathbf{U}^{(-j)}\mathbf{U}^{(j)T}) = \text{Tr}(\mathbf{U}^{(j)}\tilde{\mathbf{X}}_{(j)}^T)$, where $\tilde{\mathbf{X}}_{(j)} = \mathbf{X}_{(j)}\mathbf{U}^{(-j)}$. Since

$$\frac{\partial \text{Tr}(\mathbf{U}^{(j)}\mathbf{U}^{(-j)T}\mathbf{U}^{(-j)}\mathbf{U}^{(j)T})}{\partial \mathbf{U}^{(j)}} = 2\mathbf{U}^{(j)}\mathbf{U}^{(-j)T}\mathbf{U}^{(-j)}, \quad (30)$$

and,

$$\frac{\partial \text{Tr}(\mathbf{U}^{(j)}\tilde{\mathbf{X}}_{(j)}^T)}{\partial \mathbf{U}^{(j)}} = \tilde{\mathbf{X}}_{(j)}, \quad (31)$$

the partial derivatives of L_j with respect to $\mathbf{U}^{(j)}$ is given by

$$\begin{aligned} \frac{\partial L_j}{\partial \mathbf{U}^{(j)}} = & - \sum_{i=1}^N \left(y_i - \text{Tr}(\mathbf{U}^{(j)} \tilde{\mathbf{X}}_{i(j)}^T) - b \right) \tilde{\mathbf{X}}_{i(j)} \\ & + \lambda \mathbf{U}^{(j)} \mathbf{U}^{(-j)T} \mathbf{U}^{(-j)}. \end{aligned} \quad (32)$$

■

APPENDIX B

PROOF OF THE CONVERGENCE OF ALGORITHM 1

Proof: Here, we provide a proof of convergence for the proposed algorithm based on the one presented in [3], [10], [29], [30].

More precisely, the alternating projection method used never increases the value of the function $L(\{\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(M)}\}, b)$ between two successive iterations, as it can be regarded to be a monotonic function. We define a continuous function of the form:

$$L : \mathbb{U}_1 \times \dots \times \mathbb{U}_M \times \mathbb{R} \rightarrow \mathbb{R} \quad (34)$$

where $\mathbf{U}^{(j)} \in \mathbb{U}_j \subset \mathbb{R}^{I_j \times K}$ and the bias $b \in \mathbb{R}$ and the functions

$$L : \mathbb{U}_j \times \mathbb{R} \rightarrow \mathbb{R} \quad (35)$$

defined as $L(\mathbf{U}_j, b) = L(\mathbf{U}_j, b; \mathbf{U}^{(l)}(t)|_{l=1}^{j-1}, \mathbf{U}^{(l)}|_{l=j+1}^M)$ (i.e, fixing all but $\mathbf{U}^{(j)}$). By definition the function L has M mappings:

$$g(\mathbf{U}_*^{(j)}, b_*^j) \triangleq \underset{\mathbf{U}^{(j)}, b}{\text{argmin}} L(\mathbf{U}^{(l)}|_{l=1}^M, b) = \underset{\mathbf{U}^{(j)}, b}{\text{argmin}} L_j(\mathbf{U}^{(j)}, b) \quad (36)$$

and $*$ denotes optimality.

For the case of Generalized Tensor Ridge Regression (GTRR) we have:

$$g(\mathbf{U}_*^{(j)}, b_*^j) = (\mathbf{\Phi}_{(j)}^T \mathbf{\Phi}_{(j)} + \lambda \mathbf{I})^{-1} \mathbf{\Phi}_{(j)} \mathbf{y} \quad (37)$$

where $\mathbf{\Phi}_{(j)}$ is defined as in (18) and

$$L(\mathbf{U}_*^{(j)}, b_*^j) \geq L(\mathbf{U}^{(1)}, b^{(j)}). \quad (38)$$

Given an initial estimate $\mathbf{U}^{(l)}(0)|_{l=1}^M$, Algorithm 1 generates a sequence of solutions $\{\mathbf{U}_*^{(l)}(t)|_{l=1}^M, b_*^{(j)}(t)\}$ via

$$g(\mathbf{U}_*^{(j)}(t), b_*^{(j)}(t)) \triangleq \underset{\mathbf{U}^{(j)}, b}{\operatorname{argmin}} L(\mathbf{U}^{(j)}, b) \quad (39)$$

with $j \in \{1, 2 \dots M\}$. The sequence of produced solutions are characterized by the following relationships:

$$\begin{aligned} a_1 &= L(\mathbf{U}_*^{(1)}(1), b_*^{(1)}(1)) \\ &\geq L(\mathbf{U}_*^{(2)}(1), b_*^{(2)}(1)) \geq \dots \geq L(\mathbf{U}_*^{(M)}(1), b_*^{(M)}(1)) \\ &\geq L(\mathbf{U}_*^{(1)}(2), b_*^{(1)}(2)) \geq \dots \geq L(\mathbf{U}_*^{(1)}(t), b_*^{(1)}(t)) \\ &\geq L(\mathbf{U}_*^{(2)}(t), b_*^{(2)}(t)) \\ &\geq L(\mathbf{U}_*^{(1)}(T), b_*^{(1)}(T)) \geq \dots \geq L(\mathbf{U}_*^{(M)}(T), b_*^{(M)}(T)) \geq a_2 \end{aligned} \quad (40)$$

where $T \rightarrow \infty$ and a_1, a_2 are limit values in \mathbb{R} . Therefore we can regard the alternating optimization procedure to be a composition of M subalgorithms defined as:

$$\Omega^j : (\mathbf{U}^{(l)}|_{l=1}^M, b) \rightarrow \mathbb{R}^{I_1 \times K} \times \dots \times \mathbb{R}^{I_j \times K} \times \mathbb{R} \quad (41)$$

producing $\mathbf{U}^{(j)}$ and b . Then $\Omega = \Omega_1 \circ \Omega_2 \circ \dots \circ \Omega_M = \circ_{d=1}^M \Omega_d$ is closed when all \mathbb{U} are compact. We should emphasize here that since all subalgorithms decrease the value of L , Ω is monotonic with respect to L . Consequently, we can say that the alternating projection method converges.

The convergence proof of the Generalizer Support Tensor Regression (GSTR) can be similarly formulated using instead of L the function f defined in (20a)

$$f : \mathbb{U}_1 \times \dots \times \mathbb{U}_M \times \mathbb{R}^N \times \mathbb{R} \rightarrow \mathbb{R} \quad (42)$$

which has an extra set of parameters $\xi \in \mathfrak{R}^N$ and the mappings are given by:

$$g(\mathbf{U}_*^{(j)}, \xi, b_*^j) \triangleq \underset{\mathbf{U}^{(j)}, b}{\operatorname{argmin}} f(\mathbf{U}^{(l)}|_{l=1}^M, \xi, b) = \underset{\mathbf{U}^{(j)}, b}{\operatorname{argmin}} f(\mathbf{U}^{(j)}, b). \quad (43)$$

■

APPENDIX C

PROOF OF THE LEMMA 1

Proof: Since $(z - y)^2 \geq 0, \forall z, y \geq 0$, then

$$z \leq \frac{z^2}{2y} + \frac{1}{2}y \quad (44)$$

where we assume that $\frac{z^2}{y} = 0$ when $z = 0, y = 0$ otherwise $+\infty$ for $z \neq 0, y = 0$. The equality holds for $y = z$.

Thus,

$$\begin{aligned} \|z\|_{l_1} &= \sum_r |z_r| \leq \sum_r \left(\frac{z_r^2}{2y_r} + \frac{1}{2}y_r \right) \\ &= \sum_r \frac{z_r^2}{2y_r} + \frac{1}{2}\|y\|_{l_1}. \end{aligned} \quad (45)$$

Since $z_r = \sqrt{\sum_{m=1}^M \|\mathbf{U}_{:,r}^{(m)}\|_2^2}$, $\eta_r = y_r$, we obtain Eqn.(25). ■

APPENDIX D

ANALYTIC FORM OF THE BLOCK COORDINATE DESCENT ALGORITHM

The block coordinate descent algorithm we adopt is given by the following iterative scheme:

$$\left\{ \begin{array}{l} (\mathbf{U}^{(j)(k+1)}, b^{k+1}) \leftarrow \arg \min_{\mathbf{U}^{(j)}, b} \\ L(\mathbf{U}^{(j)}, \mathbf{U}^{(m)(k)}|_{m=1, m \neq j}, \boldsymbol{\eta}^{(k)}), \\ \boldsymbol{\eta}^{(k+1)} \leftarrow \arg \min_{\boldsymbol{\eta}} L(\boldsymbol{\eta}, \mathbf{U}^{(m)(k)}|_{m=1}, b^{(k)}). \end{array} \right. \quad (46)$$

Given that $\{\mathbf{U}^{(m)}\}_{m=1, m \neq j}^M, b\}$, the update of $\boldsymbol{\eta}$ is obtained in a straightforward way by the closed form solution provided by Lemma 1, that is

$$\eta_r = \left(\sum_{m=1}^M \|\mathbf{U}_{:,r}^{(m)}\|_2^2 \right)^{\frac{1}{2}} + \varepsilon, r = 1, 2, \dots, R \quad (47)$$

where pulsing $0 < \varepsilon \ll 1$ in order to avoid numeric singular.

ACKNOWLEDGMENT

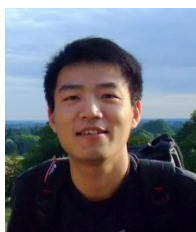
This work was supported by the EPSRC grant 'Recognition and Localization of Human Actions in Image Sequences'(EP/G033935/1), and is in part supported from the China Scholarship Council(CSC).

REFERENCES

- [1] L. Wolf, H. Jhuang, and T. Hazan, "Modeling appearances with low-rank svm," in *IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, Jun. 2007.
- [2] H. Pirsiavash, D. Ramanan, and C. Fowlkes, "Bilinear classifiers for visual recognition," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 2009.

- [3] H. Lu, K. Plataniotis, and A. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.
- [4] A. Shashua and A. Levin, "A linear image coding for regression and classification using the tensor-rank principle," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2001, Kauai, HI, USA.
- [5] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Advances in Neural Information Processing Systems*, Canada, Dec. 2006.
- [6] M. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *European Conference on Computer Vision*, Copenhagen, Denmark, May. 2002.
- [7] T. Hazan, S. Polak, and A. Shashua, "Sparse image coding using a 3d non-negative tensor factorization," in *Proc. IEEE Int. Conf. Computer Vision*, October 2005, Beijing, China.
- [8] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *International Conference on Machine Learning*. ACM, August 2005, Bonn, Germany.
- [9] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Trans. Image Processing*, vol. 16, no. 1, p. 212, 2007.
- [10] D. Tao, X. Li, X. Wu, and S. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, p. 1700, 2007.
- [11] S. Zafeiriou, "Discriminant nonnegative tensor factorization algorithms," *IEEE Trans. Neural Networks*, vol. 20, no. 2, pp. 217–235, 2009.
- [12] D. Tao, X. Li, X. Wu, W. Hu, and S. Maybank, "Supervised tensor learning," *Knowledge and Information Systems*, 2007.
- [13] R. Tomioka and K. Aihara, "Classifying matrices with a spectral regularization," in *International Conference on Machine Learning*, Corvallis, USA, Jun. 2007.
- [14] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [15] R. Jenatton, G. Obozinski, and F. Bach, "Structured sparse principal component analysis," *Arxiv preprint arXiv:0904.3523*, 2009.
- [16] A. Rakotomamonjy, "Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms," *Signal Processing*, 2011.
- [17] S. Ba and J. Odobez, "Evaluation of multiple cues head pose tracking algorithms in indoor environments," in *International Conference on Multimedia and Exposition*, Amsterdam, July 2005.
- [18] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3D models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, 2000.
- [19] N. Gouvier, D. Hall, and J. Crowley, "Estimating face orientation from robust detection of salient facial structures," in *Pointing 2004, ICPR, International Workshop on Visual Observation of Deictic Gestures*, Cambridge, UK, 2004.
- [20] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," Tech. Rep., 2001, <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [21] G. Guo, Y. Fu, C. R. Dyer, and T. Huang, "Head pose estimation: Classification or regression?" in *19th International Conference on Pattern Recognition (ICPR)*, 2008.

- [22] A. Agarwal, B. Triggs, I. Rhone-Alpes, and F. Montbonnot, "The fg-net aging database," <http://www.fgnet.rsunit.com/>. accessed: April, 2010.
- [23] S. Yan, X. Zhou, M. Liu, M. Hasegawa-Johnson, and T. Huang, "Regression from patch-kernel," *CVPR*, 2008.
- [24] L. Sigal, A. Balan, and M. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International Journal of Computer Vision*, August 2009.
- [25] R. Poppe, "Evaluating example-based pose estimation: Experiments on the humaneva sets," in *CVPR 2nd Workshop on EHuM*, June 2007, minnesota, USA.
- [26] L. Bo and C. Sminchisescu, "Structured output-associative regression," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, miami, USA.
- [27] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [28] V. Lepetit, P. Lagger, and P. Fua, "Randomized trees for real-time keypoint recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2005, san Diego, USA.
- [29] Y.Panagakis, C.Kotropoulos, and G.R.Arce, "Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 576–588, 2010.
- [30] D. Luenberger, "Linear and nonlinear programming," 3rd ed. New York: Springer, 2008.



Weiwei Guo received the B.Sc. and M.Sc. degrees in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2005 and in 2007, respectively. He is currently pursuing his Ph.D. Degree in Queen Mary, University of London, UK. His main research interests lie in pattern recognition and machine learning and their applications in the fields of computer vision.



Irene Kotsia received the diploma and PhD in Informatics from the Aristotle University of Thessaloniki, Greece in 2002 and 2008, respectively. From 2008 to 2009 she was a Research Associate in Artificial Intelligence and Information Analysis (AIIA) laboratory in the department of Informatics at Aristotle University of Thessaloniki. Since September 2009 she has been a Research Associate with the Multimedia and Vision Research group (MMV) in School of Electronic Engineering and Computer Science, Queen Mary University of London. She has coauthored many journal publications in a number of scientific journals, including IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and IEEE Transactions on Forensics and Security. Her current research interests lie in the areas of image and signal processing, statistical pattern recognition especially for human actions localization and recognition, facial expression recognition from static images and image sequences as well as in the areas of graphics and animation.



Ioannis (Yiannis) Patras (S' 1997, M'2002, SM' 2011) received the B.Sc. and M.Sc. degrees in computer science from the Computer Science Department, University of Crete, Heraklion, Greece, in 1994 and in 1997, respectively, and the Ph.D. degree from the Department of Electrical Engineering, Delft University of Technology, The Netherlands, in 2001. He has been a Postdoctorate Researcher in the area of multimedia analysis at the University of Amsterdam, and a Postdoctorate Researcher in the area of vision-based human machine interaction at TU Delft. Between 2005 and 2007 was a Lecturer in Computer Vision at the Department of Computer Science, University of York, York, UK. He is a Senior Lecturer in Computer Vision in the School of Electronic Engineering and Computer Science in the Queen Mary, University of London. He is/has been in the organizing committee of IEEE SMC 2004, Face and Gesture Recognition 2008, ICMR 2011, ICMI 2011 and ACM Multimedia 2013 and has been the general chair of WIAMIS 2009. He is associate editor in the Image and Vision Computing Journal. His research interests lie in the areas of computer vision and pattern recognition, with emphasis on motion analysis, and their applications in multimedia data management, multimodal human computer interaction, and visual communications. Currently, he is interested in the analysis of Human Motion, including the detection, tracking and understanding of facial and body gestures.