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/parallel-factor decomposition of tensors of multiple modes and allows the simultaneous projections of an input tensor to more than one direction along each mode. Two empirical risk functions are studied, namely, the square loss and ϵ-insensitive loss functions. The former leads to higher rank tensor ridge regression (TRR), and the latter leads to higher rank support tensor regression (STR), 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 and STR. Experiments conducted for the problems of head-pose, human-age, and 3-D body-pose estimations 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*—Canonical decomposition (CANDECOMP)/parallel-factor (PARAFAC; CP) decomposition, Frobenius norm, group-sparsity norm, ridge regression (RR), support vector regression (SVR), tensors.**

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 have been recently proposed for a number of problems [1]–[3]. In most cases, it has been shown that they outperform their vector-based counterparts.

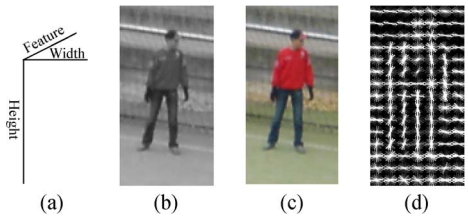


Fig. 1. Examples of visual data represented as tensors. (a) Three-mode tensor-based representations of visual objects. Examples include (b) a gray image (second-order tensor), (c) a color image (third-order tensor), and (d) a HoG descriptor (third-order tensor).

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-1 components, i.e., 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 third-order tensor consisting of slices of 2-D images and used a rank-1 tensor decomposition principle to derive new image bases that capture both spatial and temporal redundancies. The 2-D principle component analysis (PCA) represented an image in its natural matrix form (i.e., as a second-order tensor) and projected it to principal components along both horizontal and vertical directions. A 2-D subspace (manifold) learning that is based on the graph embedding of data represented as tensors were presented in [5]. In [3], the bilinear subspace analysis was generalized to a higher order multilinear PCA (MPCA) that maximizes data variance 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 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 nonnegative tensor factorization (NTF) [7], [8], in comparison

Manuscript received August 29, 2010; revised June 04, 2011 and July 27, 2011; accepted August 06, 2011. Date of publication August 18, 2011; date of current version January 18, 2012. This work was supported in part by the Engineering and Physical Sciences Research Council Grant "Recognition and Localization of Human Actions in Image Sequences" under Grant EP/G033935/1 and in part supported by China Scholarship Council. Part of this work has been done while W. Guo was in the National University of Defense Technology, China. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jenq-Neng Hwang.

W. Guo is with the College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China (e-mail: weiweiguo@nudt.edu.cn).

I. Kotsia and I. Patras are with the School of Electronic Engineering and Computer Science, Queen Mary, University of London, E1 4NS London, U.K. (e-mail: irene.kotsia@eecs.qmul.ac.uk; i.patras@eecs.qmul.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2165291

with its vector counterpart, i.e., nonnegative matrix factorization (NMF), not only preserves the local spatial relationship of images but is also 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 labeled training examples, the linear discriminant analysis was extended to multilinear discriminant analysis for face and gait recognition in [9] and [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 NMF was also extended to discriminant NTF 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-machine (SVM) methodology, again within the STL framework, was extended to handle more than one projection direction in [1], [2], and [13]. The work presented in [1] and [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-1 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 SVM 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 semidefinite 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 SVM, was proposed in [2] in the context of multitask 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 SVM subproblem for each tensor mode. Additionally, [1], [2], and [13] mainly deal with matrices (second-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 $\boldsymbol{\mathcal{X}}$ and tensor $\boldsymbol{\mathcal{W}}$ of the (unknown) parameters. That is, $f(\boldsymbol{\mathcal{X}}) = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}} \rangle + b$. Two empirical risk functions are studied, namely, the square loss and ϵ-insensitive loss functions. The former leads to what we refer to as higher rank tensor ridge regression (hrTRR), and the latter leads to higher rank support tensor regression (hrSTR). In both cases, the unknown tensor $\boldsymbol{\mathcal{W}}$ is learned in an iterative manner, where, at each iteration, using the canonical decomposition (CANDECOMP)/parallel-factor (PARAFAC; CP) decomposition [14], the data from the input tensors $\boldsymbol{\mathcal{X}}$ 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 with either low-order tensors (i.e., matrices [1], [2], [13]) or weight tensors of rank 1 [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 group-sparsity norms. The use of the Frobenius norm leads to the aforementioned hrTRR and hrSTR algorithms, requiring an *a priori* selection of the tensor rank R, typically obtained by cross validation. The use of the group-sparsity norm however allows us to enforce lower rank decomposition as a sparse constraint on rank-1 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 the tensor-based supervised learning framework.
2) We propose to use CP decomposition in order to learn simultaneously multiple projection vectors for each tensor mode. Our methodology handles high-rank tensors and directly learns 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 ϵ-insensitive loss functions, leading to hrTRR and hrSTR, respectively.
4) We investigate two regularization terms, namely, the Frobenius and group-sparsity norms. The former requires the *a priori* selection of the tensor rank and leads to hrTRR and hrSTR, whereas the latter enables us to automatically estimate the tensor rank and leads to orTRR and orSTR, respectively.

The remainder of this paper is organized as follows: In Section II, we present some useful notations regarding tensorial algebra that will be used throughout this paper. In Section III, we describe the proposed STL framework using two loss functions. More specifically, in Sections III-A and III-B, we derive the hrTRR and hrSTR algorithms using regularized square loss and 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 3-D body-pose estimations using publicly available data sets. Finally, in Section VI, we draw some conclusions.

II. Notations and Preliminaries

Here, we will briefly describe some useful notations and concepts of tensorial algebra that will be used throughout this paper

and that are consistent with those presented in [14]. More specifically, matrices will be denoted by boldface capital letters, e.g., $\boldsymbol{A}$, vectors by boldface lowercase letters, e.g., $\boldsymbol{a}$, and scalars by lowercase letters, e.g., a. Tensors are regarded as multidimensional arrays and will be denoted by Euler script calligraphic letters, e.g., $\boldsymbol{\mathcal{X}} \in I\!R^{I_1 \times I_2 \cdots \times I_M}$. The number of dimensions (also known as modes) M of a tensor denotes the order of the tensor. The ith element of vector $\mathbf{x} \in I\!R^I$ is denoted by x_i, $i = 1, 2, \ldots, I$. In a similar way, the elements of an M-order tensor $\boldsymbol{\mathcal{X}}$ will be denoted by $x_{i_1 i_2 \ldots i_n}$, $i_\ell = 1, 2, \ldots, I_\ell$ and $\ell = 1, 2, \ldots, M$.

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

The d-mode product of tensor $\boldsymbol{\mathcal{X}} \in I\!R^{I_1 \times I_2 \cdots \times I_M}$ with matrix $\boldsymbol{U} \in I\!R^{J \times I_d}$, denoted as $\boldsymbol{\mathcal{X}} \times_d \boldsymbol{U}$, is a tensor of size $I_1 \times I_{d-1} \times J \times I_{d+1} \times I_M$, element-wise defined as

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

The equivalent unfolded expression is

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

The d-mode vector product of an M-order tensor $\boldsymbol{\mathcal{X}} \in I\!R^{I_1 \times I_2 \cdots \times I_M}$ with vector $\boldsymbol{u}$ is defined likewise but resulting in an $(M-1)$-order tensor of size $I_1 \times I_{d-1} \times I_{d+1} \cdots \times I_M$.

The multiplication in every mode is denoted by

$$\boldsymbol{\mathcal{X}} \times_1 U_1 \times_2 U_2 \cdots \times_M U_M \triangleq \boldsymbol{\mathcal{X}} \prod_{k=1}^{M} \times_k U_k \tag{3}$$

whereas the multiplication in every mode except d is defined as

$$\boldsymbol{\mathcal{X}} \times_1 U_1 \cdots \times_{d-1} U_{d-1} \times_{d+1} U_{d+1} \cdots \times_M U_M \triangleq \boldsymbol{\mathcal{X}} \prod_{k=1, k \neq d}^{M} \times_k U_k \triangleq \boldsymbol{\mathcal{X}} \bar{\times}_d U_d. \tag{4}$$

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

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

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

The CP decomposition factorizes an M-order tensor $\boldsymbol{\mathcal{X}} \in I\!R^{I_1 \times I_2 \cdots \times I_M}$ into a linear combination of number R of rank-1 tensors, written as

$$\boldsymbol{\mathcal{X}} \approx \sum_{r=1}^{R} u_r^{(1)} \circ u_r^{(2)} \cdots \circ u_r^{(M)} \triangleq \left[\!\!\left[\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(M)} \right]\!\!\right]. \tag{6}$$

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

$$\boldsymbol{X}_{(d)} = \boldsymbol{U}^{(d)} \left(\boldsymbol{U}^{(M)} \odot \cdots \odot \boldsymbol{U}^{(d+1)} \odot \boldsymbol{U}^{(d-1)} \odot \cdots \odot \boldsymbol{U}^{(1)} \right)^T \tag{7}$$

where $\odot$ denotes the *Khatri–Rao product*. The rank of tensor $\boldsymbol{\mathcal{X}}$, denoted as $R = \text{rank}(\boldsymbol{\mathcal{X}})$, is the smallest number of rank-1 tensors whose sum is equal to $\boldsymbol{\mathcal{X}}$.

III. Generalized Tensor Learning Model

A classic linear predictor in the vector space is given by

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

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

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

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

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

In terms of the unfolded tensor, (8) and (9) are equivalent. However, if the input space is of high dimensionality, the overfitting 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 $\boldsymbol{\mathcal{W}} \in I\!R^{I_1 \times I_2 \cdots \times I_M}$ to be a sum of R rank-1 tensors, following the principle of CP composition. That is

$$\boldsymbol{\mathcal{W}} = \sum_{r=1}^{R} \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(M)} \triangleq \left[\!\!\left[\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \cdots, \boldsymbol{U}^{(M)} \right]\!\!\right] \tag{10}$$

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

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

As shown in (11), the input features $\boldsymbol{\mathcal{X}}$ are projected along R directions for each mode k. These projections define the final subspace that is spanned by the learned R rank-1 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 be also 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 tensor $\boldsymbol{\mathcal{W}}$) to $R\sum_{k=1}^{M} I_k$.

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

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

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 overfitting. Two types of empirical loss functions are considered in this paper for the problem of regression, i.e., the square loss and ϵ-insensitive loss function. The former leads to what we refer to as hrTRR, and the latter leads to hrSTR. We also consider two types of regularization terms, i.e., the Frobenius norm, which requires the *a priori* selection of rank R of the tensor weight (something that is achieved with exhaustive search), and the group-sparsity norm, which allows the automatic selection of the decomposition rank as a part of the learning process. The former leads to the aforementioned hrTRR and hrSTR, whereas the latter leads to orTRR and orSTR. The main steps of the proposed tensor learning for regression algorithm are summarized in Algorithm 1.

Algorithm 1 TENSOR LEARNING FOR REGRESSION

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

Output: Weights $\{\boldsymbol{U}^1, \ldots, \boldsymbol{U}^M\}$ and the bias term $b \in I\!R$ that minimize the objective function.

1: Randomly initialize $\{\boldsymbol{U}^{(1)}, \ldots, \boldsymbol{U}^{(M)}\}^{(0)}$.

2: **repeat**

3: $t \leftarrow t + 1$

4: **for** $k = 1$ to M **do**

5: Solve with respect to $\boldsymbol{U}^{(k)}|_{(t)}$:
For hrTRR solve (14) or (18);
For hrSTR solve (21);
For orTRR, solve (28);
For orSTR, solve (29).

6: **end for**

7: **if** orTRR and orSTR **then**

8: Update parameter $\boldsymbol{\eta}$ given by Lemma 1.

9: Prune columns $\boldsymbol{U}_{:r}^{(m)}$ of factor matrices,

$m \in \{1, 2, \ldots, M\}, r \in \{j | \eta_j \leq \epsilon, j = 1, 2, \ldots, R\}$

10: **end if**

11: **until** $||\boldsymbol{\mathcal{W}}^{(t)} - \boldsymbol{\mathcal{W}}^{(t-1)}||/||\boldsymbol{\mathcal{W}}^{(t-1)}|| \leq \varepsilon$ or $t \geq T_{\max}$.

A. hrTRR

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

$$
\begin{aligned}
&L\left(\left\{\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(M)}\right\}, b\right) \\
&\quad = \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \left\langle \boldsymbol{\mathcal{X}}_i, \left[\!\left[\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(M)}\right]\!\right] \right\rangle - b\right)^2 \\
&\qquad + \frac{\lambda}{2} \left|\left|\left[\!\left[\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(M)}\right]\!\right]\right|\right|_{\text{Fro}}^2. \qquad (13)
\end{aligned}
$$

We can see in the aforementioned equation that both the data and the regularization terms contain products of parameters $\boldsymbol{U}^1, \boldsymbol{U}^2, \ldots, \boldsymbol{U}^M$. Thus, a closed-form solution such as the one obtained for the vector-based ridge regression (RR) 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 $\{\boldsymbol{U}^1, \boldsymbol{U}^2, \ldots, \boldsymbol{U}^M\}$, whereas all the other parameters are kept fixed. The procedure is repeated until a convergence criterion is met.

At each iteration, we solve for parameters $\boldsymbol{U}^{(j)}$ that are associated with the projection along mode j while keeping parameters $\{\boldsymbol{U}^{(k)}\}|_{k=1,k\neq j}^{M}$ for the projections along all other modes fixed. Keeping $\{\boldsymbol{U}^{(k)}\}|_{k=1\,k\neq j}^{M}$ fixed, $\widehat{\boldsymbol{U}}^{(j)}$ that minimizes (13) is the one that minimizes

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

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

The reduced regularized least-square optimization problem previously defined can be solved in a closed form for the bias term b [i.e., $b = 1/N \sum_{i=1}^{N}(y_i - \mathrm{Tr}(\boldsymbol{U}^{(j)}\tilde{\boldsymbol{X}}_{i(j)}^T))$] but not for $\boldsymbol{U}^{(j)}$. To solve for $\boldsymbol{U}^{(j)}$, we need to resort to gradient-descent-style methods, e.g., a BFGS quasi-Newton optimizer. The gradient of objective L_j with respect to $\boldsymbol{U}^{(j)}$ is given by

$$\frac{\partial L_j}{\partial \boldsymbol{U}^{(j)}} = -\sum_{i=1}^{N}\left(y_i - \mathrm{Tr}\left(\boldsymbol{U}^{(j)}\tilde{\boldsymbol{X}}_{i(j)}^T\right) - b\right)\tilde{\boldsymbol{X}}_{i(j)} + \lambda \boldsymbol{U}^{(j)}\boldsymbol{U}^{(-j)T}\boldsymbol{U}^{(-j)} \quad (15)$$

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

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

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

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

$$\Omega(\boldsymbol{\mathcal{W}}) = \sum_{k=1}^{M}\left\|\boldsymbol{U}^{(k)}\right\|_{\mathrm{Fro}}^2 \quad (17)$$

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

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

where $\hat{\mathbf{u}}^{(j)} = [\mathrm{vec}(\hat{\boldsymbol{U}}^{(j)})^T\ b]^T$ is the vector of the unknowns, $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ are the targets, and the ith row of matrix $\boldsymbol{\Phi}$ is $[\mathrm{vec}(\tilde{\boldsymbol{X}}_{i(j)})^T\ 1]$.

B. hrSTR

Here, 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 $\psi(\Theta) = \|\boldsymbol{\mathcal{W}}\|_{\mathrm{Fro}}^2$. Following the same reasoning behind the support-vector-regression (SVR) methodology, we formulate the following optimization problem:

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

$$\mathrm{s.t.} - y_i + \langle \boldsymbol{\mathcal{X}}_i, \boldsymbol{\mathcal{W}}\rangle + b \geq \epsilon + \hat{\xi}_i,$$
$$y_i - \langle \boldsymbol{\mathcal{X}}_i, \boldsymbol{\mathcal{W}}\rangle - b \leq \epsilon + \xi_i,$$
$$\epsilon \geq 0,\ \xi_i \geq 0,\ \hat{\xi}_i \geq 0,\ \forall i = 1, 2, \ldots, N. \quad (19b)$$

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

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

$$\mathrm{s.t.} - y_i + \mathrm{Tr}\left(\boldsymbol{U}^{(j)}\boldsymbol{U}^{(-j)T}\boldsymbol{X}_{i(j)}^T\right) + b \geq \epsilon + \hat{\xi}i,$$
$$y_i - \mathrm{Tr}(\boldsymbol{U}^{(j)}\boldsymbol{U}^{(-j)T}\boldsymbol{X}_{i(j)}^T) - b \leq \epsilon + \xi_i,$$
$$\epsilon \geq 0,\ \xi_i \geq 0,\ \hat{\xi}_i \geq 0,\ \forall i = 1, 2, \ldots, N. \quad (20b)$$

If we denote $\boldsymbol{B} = \boldsymbol{U}^{(-j)T}\boldsymbol{U}^{(-j)}$, $\widetilde{\boldsymbol{U}}^{(j)} = \boldsymbol{U}^{(j)}\boldsymbol{B}^{1/2}$, and $\widetilde{\boldsymbol{X}}_{i(j)} = \boldsymbol{X}_{i(j)}\boldsymbol{U}^{(-j)}\boldsymbol{B}^{1/2}$, the optimization problem in (20) can be then rewritten as

$$\min_{\boldsymbol{U}^{(j)},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}} \quad \frac{1}{2}\mathrm{Tr}\left(\widetilde{\boldsymbol{U}}^{(j)}\widetilde{\boldsymbol{U}}^{(j)T}\right) + C\sum_{i=1}^{N}(\xi_i + \hat{\xi}_i) \quad (21a)$$

$$\mathrm{s.t.} - y_i + \mathrm{Tr}\left(\widetilde{\boldsymbol{U}}^{(j)}\widetilde{\boldsymbol{X}}_{i(j)}^T\right) + b \geq \epsilon + \hat{\xi}_i,$$
$$y_i - \mathrm{Tr}\left(\widetilde{\boldsymbol{U}}^{(j)}\widetilde{\boldsymbol{X}}_{i(j)}^T\right) - b \leq \epsilon + \xi_i,$$
$$\epsilon \geq 0,\ \xi_i \geq 0,\ \hat{\xi}_i \geq 0,\ \forall i = 1, 2, \ldots, N. \quad (21b)$$

If we vectorize $\widetilde{\boldsymbol{U}}^{(j)}$ and $\widetilde{\boldsymbol{X}}_{i(j)}$, since

$$\mathrm{Tr}\left(\widetilde{\boldsymbol{U}}^{(j)}\widetilde{\boldsymbol{U}}^{(j)T}\right) = \left\|\mathrm{vec}\left(\widetilde{\boldsymbol{U}}^{(j)}\right)\right\|^2$$
$$\mathrm{Tr}\left(\widetilde{\boldsymbol{U}}^{(j)}\widetilde{\boldsymbol{X}}_{i(j)}^T\right) = \left[\mathrm{vec}\left(\widetilde{\boldsymbol{U}}^{(j)}\right)\right]^T\left[\mathrm{vec}\left(\widetilde{\boldsymbol{X}}_{i(j)}\right)\right] \quad (22)$$

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

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

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, which is 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 $\boldsymbol{\mathcal{W}}$ 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 the $l_{1,2}$ norm [15], [16] favoring in that way the sparse-column characterization of the factor matrices of $\boldsymbol{\mathcal{W}}$, as depicted in Fig. 2.

This group-sparsity norm regularization can be written as

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

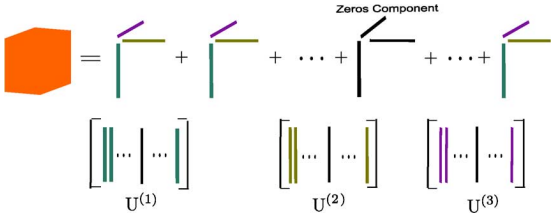


Fig. 2. Schematic description of sparse rank decomposition.

where $\boldsymbol{U}_{:,r}^{(m)}$ denotes the rth column of matrix $\boldsymbol{U}^{(m)}$. By applying this regularization, we force the same rth columns of the factor matrices $\boldsymbol{U}^{(m)}$, $m = 1, 2, \ldots, M$, 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:

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

The minimum is then obtained for $\eta_r = (\sum_{m=1}^{M}\|\boldsymbol{U}_{:,r}^{(m)}\|_2^2)^{1/2}$, $\forall r = 1, 2, \ldots, R$. *The proof is provided in* Appendix C.

By substituting (25) into (12), we formulate the following optimization problem that corresponds to what we will refer to as the optimal-rank tensor learning model:

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

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

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

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

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

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

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

Regarding the orSTR case, we consider the ϵ-insensitive loss function and rewrite subproblem (27) into the following equivalent form:

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

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

$$y_i - \mathrm{Tr}\left(\boldsymbol{U}^{(j)}\boldsymbol{U}^{(-j)T}\boldsymbol{X}_{i(j)}^{T}\right) - b \leq \epsilon + \xi_i,$$

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

that is similar to (21) if we replace $\boldsymbol{B}$ with Λ and can be hence 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 estimations.

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

1) The influence of 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 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 data sets, i.e., the IDIAP [17], Boston University (BU) [18], and Pointing'04 [19] data sets.

1) IDIAP Data Set: 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 data set, containing only the videos from meeting scenarios 1, 3, and 4, split 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, i.e., the normalized pixel intensity and the log-Gabor features. Each of the images formed a 40×30 second-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 (MAE) for each of $\{\alpha, \beta, \gamma\}$ [17].

TABLE I
BU 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

2) BU Data Set: The BU data set consists of 45 video sequences, depicting five subjects performing nine 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 (see Table I).

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

3) Pointing'04: The Pointing'04 head-pose database contains a variety of head poses ranging from $-90°$ to $90°$ 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 horizontally and seven vertically, as well as the two extreme cases of the vertical $90°$ and $-90°$, 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 four scales and eight orientations was applied at each image, and the local binary patterns were calculated. Each image was divided into nonoverlapping rectangular subregions 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 five training schemes, i.e., the closed-form solution updates in hrTRR [see (14)], the BFGS quasi-Newton solver for the hrTRR [B-hrTRR; see (18)], the Libsvm [20] optimizer for the hrSTR [see (21)], orTRR, and 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 shown, hrTRR, B-hrTRR, and hrSTR all converge quite fast, whereas 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.

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 rank R and the regularization parameters λ and C for hrTRR and hrSTR, we report the testing errors of the pointing vector against the rank and

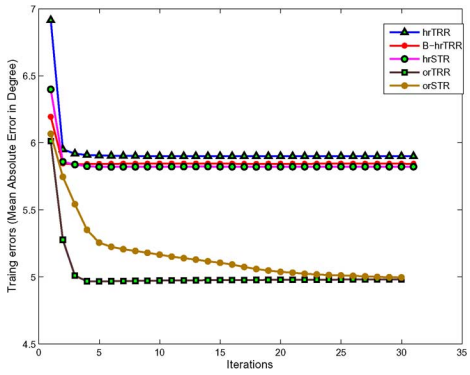


Fig. 3. Training error for the tilt angle on the IDIAP data set.

the regularization parameters in Fig. 4. It can be easily seen that the proposed tensor-based decomposition of parameters avoids overfitting while outperforming its vector-based counterparts, particularly 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 on, 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 data sets). We will therefore report the value of the optimal rank from now on. 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.

The rank-1 components $\{u_r^{(1)} \circ u_r^{(2)} \circ \cdots \circ u_r^{(M)}\}|_{r=1}^{R}$ of the weight tensor $\mathcal{W}$ that are obtained at the 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 data set. 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 clearer spatial patterns that can be interpreted as filters applied at the input image, when compared with the equivalent weights acquired for RR and SVR.

As mentioned before, the projections along the tensor modes that are performed by 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 tensor (MPCA [3]) representation before a classic vector-based regression algorithm (RR or SVR) is trained. The number of PCA (or MPCA) components is 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 with the ones obtained by the proposed direct tensor-based regression.

The angular error for pan and tilt, as well as their average value obtained for the Pointing'04 data set for RR, SVR, hrTRR, hrSTR, orTRR, and orSTR, are given in Table V. The optimal

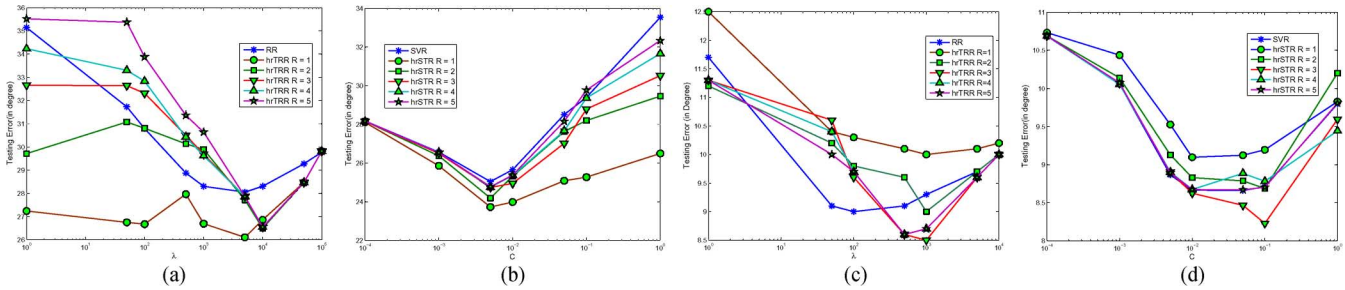


Fig. 4. Testing error for different rank R and regularization parameters λ and C for (a) TRR, (b) STR on IDIAP, and (c) TRR and (d) STR on BU.

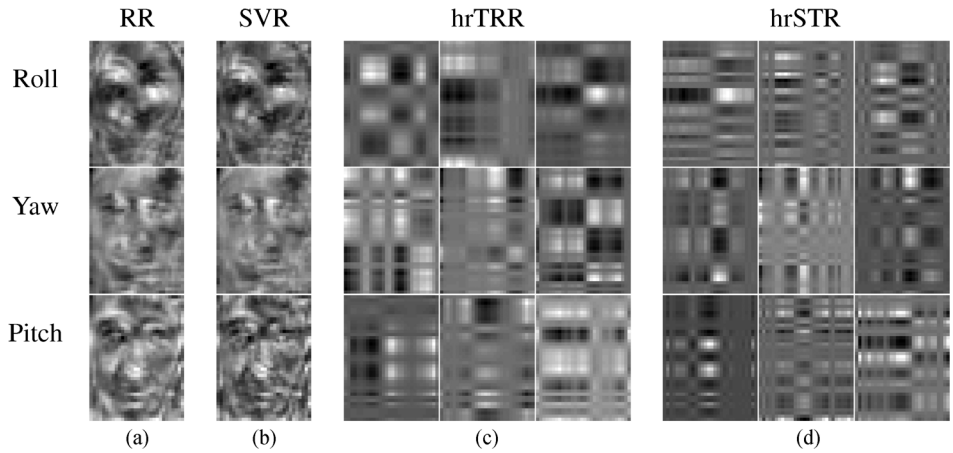


Fig. 5. Images of weights obtained by (a) RR and (b) SVR. Three rank-1 weight tensors for (c) hrTRR and (d) hrSTR for the BU data set.

TABLE II
ANGULAR ERROR FOR THE IDIAP DATA SET

	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 DATA SET

	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

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

Features	Dataset	RR	SVR	PCA +RR	PCA +SVR	MPCA +RR	MPCA +SVR	hrTRR	hrSTR
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**

rank selected was equal to 4. As shown, the results acquired with hrSTR outperform the state-of-the-art results presented in [21].

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

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 data set, i.e., the FG-NET data set [22]. The FG-NET data set comprises of 1002 facial images of 82 people from 0 to 69 years old. A set of 68 labeled facial landmarks characterizing shape features are also provided for each facial image. The protocol followed for experiments was the leave-one-person-out protocol. 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 where normalized to a template using piecewise affine transform. The features extracted were the same as in the case of the Pointing'04 data set. The performance of our algorithms was measured using the 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, i.e., $\text{MAE} = \sum_{k=1}^{N} |\overline{\text{Age}}_k - \text{Age}_k|/N$, where, with $\overline{\text{Age}}_k$, we denote the estimated age for a test image k, with Age_k being its ground-truth age and N being the total number of test images. The estimation accuracy can be estimated by the CS, defined as

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

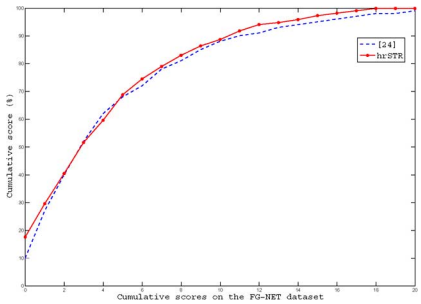


Fig. 6. Cumulative scores on the FG-NET data set.

$\mathrm{CS}(j) = N_{e\leq j/N} \times 100\%$, where $N_{e\leq j}$ is the number of test images on which the estimator makes an absolute error not 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 CS achieved is depicted in Fig. 6, where we also plot the state-of-the-art results [23].

As shown in the results in Table VI, tensors significantly outperform vectors, at the same time providing better than the state-of-the-art results. Furthermore, one may notice here that, when many examples are available (and 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 3-D human-pose estimation experiments, we used the HumanEva-I training and validation sets [24]. More specifically, we conducted experiments for three subjects and for four actions, i.e., 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 (3-D 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 $\times$ 5 blocks. The gradient orientations in each block were quantized in nine orientation bins. Thus, each image was represented as a 6 $\times$ 5 $\times$ 9 tensor of order 3. The 3-D human-body pose was encoded in a 57-dimensional vector $\mathbf{y}$, corresponding to

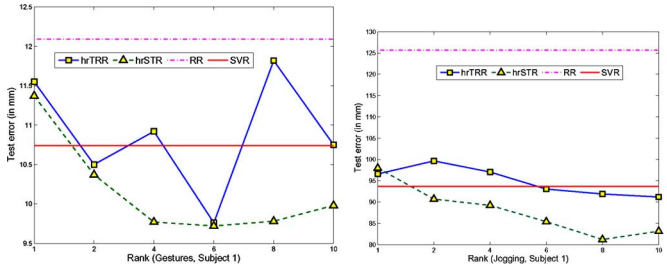


Fig. 7. MAE versus rank for gestures and jogging of subject 1.

TABLE VII
MEAN TESTING ERRORS OF GLOBALLY TRAINED MODELS

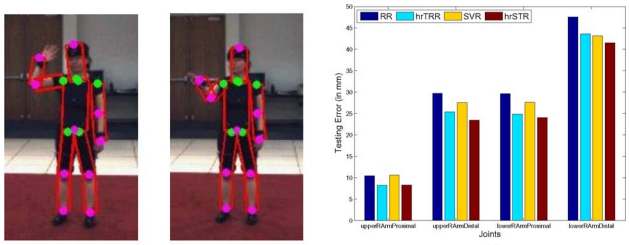
Action	RR	SVR	hrTRR	hrSTR	orTRR	orSTR
Walking	77.4	**74.5**	78.4	77.6	78.4	76.1
Jog	91.0	76.0	75.9	73.7	80.5	**68.1**
Gesture	98.2	63.6	79.4	**63.0**	70.0	66.8
Box	101.0	78.3	86.3	74.4	70.5	**62.8**
Average	91.9	73.1	80.0	72.2	74.9	**68.5**

the 3-D positions of 19 joint centers. Each of the joint positions was relatively defined 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, $Err = 1/N \sum_{n=1}^{N} 1/M \sum_{i=1}^{M} \|\mathbf{y}_i^{(n)} - \bar{\mathbf{y}}_i^{(n)}\|$. The experiments were performed by training an individual predictor for each output.

1) Globally Trained Model: Due to the nonlinear 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.

We experimented with different values for rank R, namely, the ones in 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 three 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 RR, hrTRR, SVR, and hrSTR for those joints are 29.4, 25.5, 27.2, and 24.3, respectively, i.e., 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



(a) (b) (c)

Fig. 8. "Gestures" action: [(a) and (b)] Estimated body model. (c) Error for right arm joints.

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**

(i.e., angles and wrists) for which the errors are greater than 130 mm.

2) Locally Trained Model: The globally 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 multiclass classifier. In this paper, 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.

D. Discussion

To sum up, from the experiments previously presented, someone can observe the following:

1) 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 data set and for the human-age estimation problem, tensors improve the state-of-the-art results by 9.29% and 27.58%, respectively.
2) 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.
3) 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. CONCLUSION

In this paper, we have proposed 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 CP decomposition. Two empirical risk functions are studied, namely, the square loss and ϵ-insensitive loss functions. These lead to the generalization of two well-known regression schemes, namely, RR and SVR, to their corresponding tensor-based regression methods, namely, hrTRR and hrSTR. The aforementioned algorithms have been formulated using as a regularization term, i.e., the Frobenius norm. We have also studied the group-sparsity norm in order to achieve automatic tensor rank selection, thus formulating the equivalent orTRR and orSTR. Experiments performed using publicly available real data for the problems of head-pose, human-age, and body-pose estimation have verified not only the superiority of the tensors-based algorithms when compared with the vector-based ones but also the efficiency of the proposed algorithms.

APPENDIX A
PROOF OF (15)

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

$$\frac{\partial \text{Tr}\left(\boldsymbol{U}^{(j)}\boldsymbol{U}^{(-j)T}\boldsymbol{U}^{(-j)}\boldsymbol{U}^{(j)T}\right)}{\partial \boldsymbol{U}^{(j)}} = 2\boldsymbol{U}^{(j)}\boldsymbol{U}^{(-j)T}\boldsymbol{U}^{(-j)} \tag{30}$$

$$\frac{\partial \text{Tr}\left(\boldsymbol{U}^{(j)}\tilde{\boldsymbol{X}}_{(j)}^T\right)}{\partial \boldsymbol{U}^{(j)}} = \tilde{\boldsymbol{X}}_{(j)} \tag{31}$$

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

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

■

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], and [30].

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

$$L:\cup_1\times\cdots\times\cup_M\times\mathbb{R}\to\mathbb{R} \tag{33}$$

where $\boldsymbol{U}^{(j)}\in\cup_j\subset\mathbb{R}^{I_j\times K}$ and bias $b\in\mathbb{R}$, and the functions, i.e.,

$$L:\cup_j\times\mathbb{R}\to\mathbb{R} \tag{34}$$

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

$$g\left(\boldsymbol{U}_*^{(j)},b_*^j\right)\triangleq\underset{\boldsymbol{U}^{(j)},b}{\arg\min}\,L\left(\boldsymbol{U}^{(l)}|_{l=1}^{M},b\right)=\underset{\boldsymbol{U}^{(j)},b}{\arg\min}\,L_j\left(\boldsymbol{U}^{(j)},b\right) \tag{35}$$

and $*$ denotes optimality.

For the case of generalized TRR, we have

$$g\left(\boldsymbol{U}_*^{(j)},b_*^j\right)=\left(\boldsymbol{\Phi}_{(j)}^T\boldsymbol{\Phi}_{(j)}+\lambda\mathbf{I}\right)^{-1}\boldsymbol{\Phi}_{(j)}\mathbf{y} \tag{36}$$

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

$$L\left(\boldsymbol{U}_*^{(j)},b_*^{(j)}\right)\geq L\left(\boldsymbol{U}^{(1)},b^{(j)}\right). \tag{37}$$

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

$$g\left(\boldsymbol{U}^{(j)}{}_*(t),b_*^{(j)}(t)\right)\triangleq\underset{\boldsymbol{U}^{(j)},b}{\arg\min}\,L\left(\boldsymbol{U}^{(j)},b\right) \tag{38}$$

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

$$\begin{aligned}
a_1&=L\left(\boldsymbol{U}_*^{(1)}(1),b_*^{(1)}1\right)\\
&\geq L\left(\boldsymbol{U}_*^{(2)}(1),b_*^{(2)}(1)\right)\geq\cdots\geq L\left(\boldsymbol{U}_*^{(M)}(1),b_*^{M}(1)\right)\\
&\geq L\left(\boldsymbol{U}_*^{(1)}(2),b_*^{(1)}(2)\right)\geq\cdots\geq L\left(\boldsymbol{U}_*^{(1)}(t),b_*^{(1)}(t)\right)\\
&\qquad\geq L\left(\boldsymbol{U}_*^{(2)}(t),b_*^{(2)}(t)\right)\\
&\geq L\left(\boldsymbol{U}_*^{(1)}(T),b_*^{1}(T)\right)\geq\cdots\geq L\left(\boldsymbol{U}_*^{(M)}(T),b_*^{M}(T)\right)\geq a_2
\end{aligned} \tag{39}$$

where $T\to\infty$ 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

$$\Omega^j:\left(\boldsymbol{U}^{(l)}|_{l=1}^{M},b\right)\to\mathbb{R}^{I_1\times K}\times\cdots\times\mathbb{R}^{I_j\times K}\times R \tag{40}$$

producing $\mathbf{U}^{(j)}$ and b. Then, $\Omega=\Omega_1\circ\Omega_2\circ\ldots\Omega_M=\circ_{d=1}^{M}\Omega_d$ is closed when all $\cup$ 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 generalized STR can be similarly formulated using, instead of L, function f defined in (20a), i.e.,

$$f:\cup_1\times\cdots\times\cup_M\times\mathbb{R}^N\times\mathbb{R}\to\mathbb{R} \tag{41}$$

which has an extra set of parameters $\xi\in\Re^N$, and the mappings are given by

$$\begin{aligned}
g\left(\boldsymbol{U}_*^{(j)},\xi,b_*^j\right)&\triangleq\underset{\boldsymbol{U}^{(j)},b}{\arg\min}\,f\left(\boldsymbol{U}^{(l)}|_{l=1}^{M},\xi,b\right)\\
&=\underset{\boldsymbol{U}^{(j)},b}{\arg\min}\,f\left(\boldsymbol{U}^{(j)},b\right).
\end{aligned} \tag{42}$$

■

Appendix C
Proof of the Lemma 1

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

$$z\leq\frac{z^2}{2y}+\frac{1}{2}y \tag{43}$$

where we assume that $z^2/y=0$ when $z=0$ and $y=0$, and otherwise $+\infty$ for $z\neq 0$ and $y=0$. The equality holds for $y=z$. Thus

$$\begin{aligned}
||\boldsymbol{z}||_{l_1}&=\sum_r|z_r|\leq\sum_r\left(\frac{z_r^2}{2y_r}+\frac{1}{2}y_r\right)\\
&=\sum_r\frac{z_r^2}{2y_r}+\frac{1}{2}||y||_{l_1}.
\end{aligned} \tag{44}$$

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

Appendix D
Analytic Form of the Block Coordinate-Descent Algorithm

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

$$\begin{cases}
\left(\boldsymbol{U}^{(j)(k+1)},b^{k+1}\right) & \leftarrow\underset{\boldsymbol{U}^{(j)},b}{\arg\min}\\
& \qquad L\left(\boldsymbol{U}^{(j)},\boldsymbol{U}^{(m)(k)}|_{m=1,m\neq j}^{M},\boldsymbol{\eta}^{(k)}\right)\\
\boldsymbol{\eta}^{(k+1)} & \leftarrow\underset{\boldsymbol{\eta}}{\arg\min}\,L\left(\boldsymbol{\eta},\boldsymbol{U}^{(m)(k)}|_{m=1}^{M},b^{(k)}\right).
\end{cases} \tag{45}$$

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

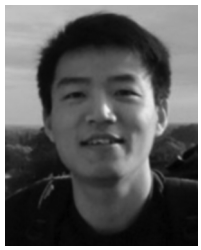
$$\eta_r=\left(\sum_{m=1}^{M}\left\|\boldsymbol{U}_{:,r}^{(m)}\right\|_2^2\right)^{\frac{1}{2}}+\varepsilon,\ r=1,2,\ldots,R \tag{46}$$

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

References

[1] L. Wolf, H. Jhuang, and T. Hazan, "Modeling appearances with low-rank SVM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, Jun. 2007, pp. 1–6.

[2] H. Pirsiavash, D. Ramanan, and C. Fowlkes, "Bilinear classifiers for visual recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2009.
[3] H. Lu, K. Plataniotis, and A. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.
[4] A. Shashua and A. Levin, "A linear image coding for regression and classification using the tensor-rank principle," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Kauai, HI, Jun. 2001, pp. I-42–I-49.
[5] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, Dec. 2006.
[6] M. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proc. Eur. Conf. Comput. Vis.*, Copenhagen, Denmark, May 2002, pp. 447–460.
[7] T. Hazan, S. Polak, and A. Shashua, "Sparse image coding using a 3-D non-negative tensor factorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Beijing, China, Oct. 2005, pp. 50–57.
[8] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *Proc. Int. Conf. Mach. Learn.*, Bonn, Germany, Aug. 2005, pp. 792–799.
[9] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 212–220, Jan. 2007.
[10] D. Tao, X. Li, X. Wu, and S. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.
[11] S. Zafeiriou, "Discriminant nonnegative tensor factorization algorithms," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 217–235, Feb. 2009.
[12] D. Tao, X. Li, X. Wu, W. Hu, and S. Maybank, "Supervised tensor learning," *Knowl. Inf. Syst.*, vol. 13, no. 1, pp. 1–42, Sep. 2007.
[13] R. Tomioka and K. Aihara, "Classifying matrices with a spectral regularization," in *Proc. Int. Conf. Mach. Learn.*, Corvallis, Jun. 2007, pp. 895–902.
[14] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[15] R. Jenatton, G. Obozinski, and F. Bach, Structured Sparse Principal Component Analysis 2009, Arxiv preprint arXiv:0904.3523.
[16] A. Rakotomamonjy, "Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms," *Signal Process.*, vol. 91, no. 7, pp. 1505–1526, Jul. 2011.
[17] S. Ba and J. Odobez, "Evaluation of multiple cues head pose tracking algorithms in indoor environments," in *Proc. Int. Conf. Multimedia Expo.*, Amsterdam, The Netherlands, Jul. 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 3-D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, Apr. 2000.
[19] N. Gourier, D. Hall, and J. Crowley, "Estimating face orientation from robust detection of salient facial structures," in *Proc. Pointing ICPR, Int. Workshop Vis. Observation Deictic Gestures*, Cambridge, U.K., 2004.
[20] C.-C. Chang and C.-J. Lin, LIBSVM: A library for support vector machines Dept. Comput. Sci. Inf. Eng., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2001 [Online]. Available: 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 *Proc. 19th ICPR*, 2008, pp. 1–4.
[22] A. Agarwal, B. Triggs, I. Rhone-Alpes, and F. Montbonnot, The FG-NET Aging Database Apr. 2010 [Online]. Available: http://www.fgnet.rsunit.com/
[23] S. Yan, X. Zhou, M. Liu, M. Hasegawa-Johnson, and T. Huang, "Regression from patch-kernel," in *Proc. CVPR*, 2008, pp. 1–8.
[24] L. Sigal, A. Balan, and M. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *Int. J. Comput. Vis.*, vol. 87, no. 1/2, pp. 4–27, Mar. 2010.
[25] R. Poppe, "Evaluating example-based pose estimation: Experiments on the humaneva sets," in *Proc. CVPR 2nd Workshop EHuM*, Minnesota, MN, Jun. 2007, pp. 1–8.
[26] L. Bo and C. Sminchisescu, "Structured output-associative regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, Jun. 2009, pp. 2403–2410.
[27] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[28] V. Lepetit, P. Lagger, and P. Fua, "Randomized trees for real-time keypoint recognition," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, San Diego, CA, Jun. 2005, pp. 775–781.
[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. Acoust., Speech, Signal Process.*, vol. 18, no. 3, pp. 576–588, Mar. 2010.
[30] D. Luenberger, *Linear and Nonlinear Programming*, 3rd ed. New York: Springer-Verlag, 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, where he is currently working toward the Ph.D. degree.

He studied at Queen Mary, University of London, London, U.K., from 2008 to 2010. His main research interests are in pattern recognition and machine learning and their applications in the fields of computer vision.

Irene Kotsia (M'11) received the Diploma and the Ph.D. degree in informatics from Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2002 and 2008, respectively.

From 2008 to 2009, she was a Research Associate with the Artificial Intelligence and Information Analysis Laboratory, Department of Informatics, Aristotle University of Thessaloniki. Since September 2009, she has been a Research Associate with the Multimedia and Vision Research group, School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K. She has coauthored many journal publications in a number of scientific journals, including the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS, and the IEEE TRANSACTIONS ON FORENSICS AND SECURITY. Her current research interests are in the areas of image and signal processing, statistical pattern recognition, particularly for human actions localization and recognition, and facial expression recognition from static images and image sequences, as well as in the areas of graphics and animation.

Ioannis (Yiannis) Patras (S'97–M'02–SM'11) received the B.Sc. and M.Sc. degrees in computer science from the University of Crete, Heraklion, Greece, in 1994 and in 1997, respectively, and the Ph.D. degree from Delft University of Technology (TU Delft), Delft, The Netherlands, in 2001.

He has been a Postdoctorate Researcher in the area of multimedia analysis with the University of Amsterdam, Amsterdam, The Netherlands, and a Postdoctorate Researcher in the area of vision-based human–machine interaction with TU Delft. Between 2005 and 2007, he was a Lecturer in computer vision with the Department of Computer Science, University of York, York, U.K. He is currently a Senior Lecturer in computer vision with the School of Electronic Engineering and Computer Science, Queen Mary, University of London, London, U.K. His research interests are 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 detection, tracking and understanding of facial and body gestures.

Dr. Patras has been in the organizing committee of the IEEE International Conference on Systems, Man, and Cybernetics 2004, the Face and Gesture Recognition 2008, the International Conference on Multimedia Retrieval 2011, the International Conference on Multimodal Interaction 2011, and the Association for Computing Machinery Multimedia 2013. He has been the general chair of the Workshop on Image Analysis for Multimedia Interactive Services 2009. He is associate editor in the Image and Vision Computing Journal.