Tensor regression based on linked multiway parameter analysis

Tensor Regression Based on Linked Multiway Parameter Analysis

Yifan Fu and Junbin Gao
School of Computing and Mathematics
Charles Sturt University
Bathurst, NSW, Australia
Email: {yfu,jbgao}@csu.edu.au

Xia Hong
School of Systems Engineering
University of Reading
Reading, RG6 6AY, UK
Email: x.hong@reading.ac.uk

David Tien
School of Computing and Mathematics
Charles Sturt University
Bathurst, NSW, Australia
Email: dtien@csu.edu.au

***Abstract*—Classical regression methods take vectors as covariates and estimate the corresponding vectors of regression parameters. When addressing regression problems on covariates of more complex form such as multi-dimensional arrays (i.e. tensors), traditional computational models can be severely compromised by ultrahigh dimensionality as well as complex structure. By exploiting the special structure of tensor covariates, the tensor regression model provides a promising solution to reduce the model's dimensionality to a manageable level, thus leading to efficient estimation. Most of the existing tensor-based methods independently estimate each individual regression problem based on tensor decomposition which allows the simultaneous projections of an input tensor to more than one direction along each mode. As a matter of fact, multi-dimensional data are collected under the same or very similar conditions, so that data share some common latent components but can also have their own independent parameters for each regression task. Therefore, it is beneficial to analyse regression parameters among all the regressions in a linked way. In this paper, we propose a tensor regression model based on Tucker Decomposition, which identifies not only the common components of parameters across all the regression tasks, but also independent factors contributing to each particular regression task simultaneously. Under this paradigm, the number of independent parameters along each mode is constrained by a sparsity-preserving regulariser. Linked multiway parameter analysis and sparsity modeling further reduce the total number of parameters, with lower memory cost than their tensor-based counterparts. The effectiveness of the new method is demonstrated on real data sets.**

Keywords-tensor regression; linked multiway parameter analysis; sparse coding

I. Introduction

Advancing technologies are constantly producing large amounts of multi-dimensional data, such as electroencephalography ($2D$ matrix), video sequences ($3D$ array) and functional magnetic resonance images ($4D$ array). In multi-dimensional data analysis, a challenging problem is to predict the outcome of a continual criterion variable based on one or more predictor variables, which is as known as a regression. Traditional regression approaches in literature work on vector spaces that are derived by stacking the original multi-dimensional data into vectors in a random order. This vectorisation of data breaks the inherent spatial structure of high-dimensional data, and more seriously, leads to ultrahigh computational complexity and large memory requirements for multi-dimensional data. A typical solution is to firstly extract a vector of features from a given dataset, and then to feed the feature vector into a classical regression model [2], [11], [14], [12], [19]. Alternatively, as in [4], one first applies unsupervised dimension reduction (i.e. often using some variant of principal component analysis) to the data array, and then fits a regression model in the lower dimensional vector space. However, this feature selection or dimension reduction scheme could result in information loss in a regression setup. Genkin et al. [7] proposed a Bayesian approach to avoid overfitting, which uses a prior distribution that favours sparseness in the fitted model. Nonetheless, it hasn't solved the substantial problem and that structural information among the data is lost.

Recently, researchers have resorted to employing *tensor* in regression modelling, which naturally takes into account the spatial structure in the original data as a multi-way array. The advantages of tensor-based methods seem to stem from the way tensors are decomposed (e.g. CANDECOMP/PARAFAC (CP) decomposition or Tucker decomposition [10]). More specifically, the estimated output can be expressed by a predictor tensor along with a low-dimensional factor matrix at each mode. Factorizing the huge parameter space into low-dimensional factor components associated with each mode drastically reduces the number of unknown parameters to be estimated, as well as accounting structural information of the predictor spaces. Usually, the parameters associated with each mode are estimated in an iterative manner, where, at each iteration, only the parameters corresponding to a specific mode are updated.

Some works based on tensor decomposition have been proposed in recent years, such as [9], [25], [13], [21], [24]. Guo et al. [9] first addressed the regression problem using tensor representation with the CP decomposition. Specifically, the regression parameters are learnt in an iterative manner. In this scheme, the input data (tensor) is projected along a certain mode and the parameters associated with that mode are learned by solving a linear problem of reduced dimensionality. Zhou et al. [25] proposed a class of generalised tensor regression models based on the low rank CP decomposition. Similar work based on the Tucker Tensor regression model was proposed by Li et al. [13]. Gao and Wu [6] proposed an algorithm named Kernel Support Tensor Regression (KSTR) using tensors as input for function regression. Tan et al. [21] proposed a logistic tensor regression for classification. Zhao et al. [24] introduced a generalised partial least square (PLS) framework for high-order tensors and applied it to tensor subspace regression. In essence, tensor decomposition acts as a feature selection or

dimensionality reduction scheme to decrease the number of regression parameters to a manageable level. As a matter of fact, the parameter number may be further reduced by deeply exploiting components in each mode, since multi-dimensional data are usually collected under the same or very similar conditions. For example, a set of face images for different subjects recorded over many trials and under the same experimental setup. Such data share some common latent (hidden) spatial factors embedded across all regressions but can also have their own independent factors contributing to each particular regression task. Accordingly, it is quite necessary to separately analyse these two different types of factors in a linked way. Unfortunately, most existing tensor-based approaches build a regression model for each task separately, neglecting the common components across all the regression tasks; or they build models jointly without identifying individual factors.

Against this background, we propose a new tensor regression approach named *Linked Multiway Parameter Analysis based Tensor Regression* (LMPA-TR) in this paper. LMPA-TR imposes constraints on the estimated components for each mode, which is identically correlated across all the regressions with regard to their spatial distributions. Our proposed model employs Tucker Decomposition to identify not only the common components of parameters across all the regression tasks, but also independent factors contributing to each particular regression task simultaneously. Moreover, the number of independent parameters along each mode is constrained by a sparsity-preserving regulariser.

The contribution of this paper is two-fold. First, from an image analysis point of view, our proposal provides a systematic solution for the integrative analysis of multi-modality imaging data, such as human-pose estimation and neuroimaging analysis. Second, from a statistical methodology point of view, our proposal provides a novel and general framework for regression with multi-dimensional data. Although there have been some tensor regression methods utilising tensor decomposition to reduce computational complexity, our proposal, to the best of our knowledge, is the first work that analyses the variability and consistency of components for individual regression and across multiple regressions simultaneously.

The remainder of the paper is organized as follows. Some preliminaries on tensor and the problem formulation are presented in Section II. The proposed tensor regression method is detailed in Section III. Experimental results are reported in Section IV and we conclude the paper in Section V.

II. Notations and Problem Formulation

A. Definition and Notations

Here, we briefly introduce some tensor fundamentals and notations used throughout the paper. More specifically, tensors (or multi-way arrays) are denoted by calligraphic letters, e.g. $\mathcal{X}$, matrices by boldface capital letters, e.g. $\mathbf{X}$, and vectors by boldface lower-case letters, e.g, $\mathbf{x}$. The number of the dimensions (also known as modes) of a tensor is the order of the tensor. The ith entry of a vector $\mathbf{x}$ is denoted by x_i, the (i,j) element of a matrix $\mathbf{X}$ by $x_{i,j}$, and the $(i_1, i_2, \ldots, i_N)$ element of an N-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by $x_{i_1,i_2,\ldots,i_N}$.

Definition 1 (Kronecker Product): The Kronecker product of matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{P \times L}$, denoted by $\mathbf{A} \otimes \mathbf{B}$, is a matrix of size $(IP) \times (JL)$ defined by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix} \tag{1}$$

Definition 2 (Tensor Matricisation): Matricisation is the operation of rearranging the entries of a tensor so that it can be represented as a matrix. Let $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ be a tensor of order-N, the mode-n matricisation of $\mathcal{X}$ reorders the mode-n vectors to be columns of the resulting matrix, denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (I_{n+1} I_{n+2} \ldots I_N I_1 I_2 \ldots I_{n-1})}$.

Definition 3 (The n-mode Product): The n-mode product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ by a matrix $\mathbf{U} \in \mathbb{R}^{R \times I_n}$, denoted as $\mathcal{X} \times_n \mathbf{U}$, is a tensor with entries:

$$(\mathcal{X} \times_n \mathbf{U})_{i_1,\ldots,i_{n-1},r_n,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{r i_n} \tag{2}$$

The n-mode product is also denoted by each mode-n vector multiplied by the matrix $\mathbf{U}$. Thus, it can be expressed in terms of tensor matricisation as well:

$$\mathcal{Y} = \mathcal{X} \times_n \mathbf{U} \qquad \Leftrightarrow \qquad \mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)} \tag{3}$$

Definition 4 (Tucker Decomposition): An N-order tensor $\mathcal{X}$ admits a Tucker decomposition if it can be written as

$$\begin{aligned} \mathcal{X} &\equiv [\![\mathcal{G}; \mathbf{U}_1, ..., \mathbf{U}_N]\!] = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \ldots \times_N \mathbf{U}_N \\ &= \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \ldots r_N} \mathbf{u}_{r_1} \circ \mathbf{u}_{r_2} \ldots \circ \mathbf{u}_{r_N} \end{aligned} \tag{4}$$

where $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \ldots \ R_N}$ is called a core tensor and $\mathbf{U}^{(i)} \in \mathbb{R}^{I_i \times R_i} (1 \leq i \leq N)$ are the factor matrices at each mode.

For a Tucker tensor $\mathcal{X}$, its mode-n matricisation can be expressed as

$$\mathbf{X}_{(n)} = \mathbf{U}_n \mathbf{G}_{(n)} (\mathbf{U}_N \otimes \cdots \otimes \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \cdots \otimes \mathbf{U}_1)^T \tag{5}$$

Definition 5 (The Frobenius norm of a matrix): The Frobenius norm of a matrix $\mathbf{X} \in \mathbb{R}^{I \times R}$ is the square root of the sum of the squares of all its elements, i.e.,

$$||\mathbf{X}||_F = \sqrt{\sum_{i=1}^{I} \sum_{j=1}^{R} x_{i,j}^2} \tag{6}$$

Similarly we can define the Frobenius norm for any N-order tensors.

B. Problem Formulation

A classical linear predictor on a vector space is given by

$$y = f(\mathbf{x}; \mathbf{u}, e) = \langle \mathbf{x}, \mathbf{u} \rangle + e \tag{7}$$

where $\mathbf{x}$ is the input data in a vector format, $\mathbf{u}$ is the parameter weight vector, $\langle \cdot, \cdot \rangle$ is the inner product of vectors. e and y are the error and the regression output, respectively. Note that scalar output regression is considered here and we have

assumed the bias terms to be zero in the model as it is easy to centralise output data to achieve zero bias.

When the classic linear predictor is extended from vector space cases to tensor space cases, the regression model is formulated as follows

$$\begin{aligned} y &= f(\mathcal{X}; \mathbf{u}, \mathbf{e}) = \mathcal{X} \times_1 \mathbf{u}_1 \times_2 \ldots \times_N \mathbf{u}_N + e \\ &= [\![\mathcal{X}; \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N]\!] + e \end{aligned} \tag{8}$$

where $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ represents the input tensorial features with N modes and $\mathbf{u} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}$ is a set of parameter weight components $\mathbf{u}_i \in \mathbb{R}^{1 \times I_i}$ along each mode. Scalars e and y represent error and output values respectively.

Remark 1: In terms of unfolded tensor, Eq. (7) and Eq. (8) are equivalent. However, if the input space is of high dimensionality, the overfitting and high computational complexity problem occur in the classical regression model with a large number of parameters.

Given a dataset containing m N-order data $\mathcal{X}_q (1 \leq q \leq m)$, each of them has an output vector $\mathbf{y}_q = \{y_q^1, y_q^2, \ldots, y_q^K\}$ corresponding to values of K target continuous variables. Our objective is to learn K tensor regression models as represented by Eq.(8) for the target output, that is, to find an optimal parameter weight set $\mathbf{u}^j = \{\mathbf{u}_1^j, \mathbf{u}_2^j, \ldots, \mathbf{u}_N^j\}$ for task-j where $j = 1, 2, ..., K$, such that

$$\min_{\mathbf{u}^j} \sum_{q=1}^{m} \| y_q^j - \mathcal{X}_q \times_1 \mathbf{u}_1^j \times_2 \mathbf{u}_2^j \cdots \mathbf{u}_N^j \|_F^2 \tag{9}$$

We create an $(N+1)$-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times m}$ by stacking all the $\mathcal{X}_q$ along the $(N+1)$-mode, and an output vector of a specific output variable in the entire data set represented by $\mathbf{y}^j = \{y_1^j, y_2^j, \ldots, y_m^j\}$, regarded as a $1 \times 1 \times \cdots \times 1 \times m$ tensor, where $1 \leq j \leq K$. Then the optimisation problem (9) can be equivalent to

$$\min_{\mathbf{u}^j} \| \mathbf{y}^j - \mathcal{X} \times_1 \mathbf{u}_1^j \times_2 \mathbf{u}_2^j \cdots \mathbf{u}_N^j \times \mathbf{U}_{N+1} \|_F^2 \tag{10}$$

where $\mathbf{U}_{N+1}$ is the identity matrix of size m (i.e. $\mathbf{I}_m$).

III. Tensor Regression Based on Linked Multiway Parameter Analysis

Various tensor regression methods have been proposed by employing either CP or Tucker decomposition to find parameter components for each individual regression estimation independently, usually without imposing any constraints on components for each mode. However, in many scenarios some *links* need to be considered to analyse variability and consistency of the parameter components across regression tasks. In other words, components in each mode do not need to be necessarily independent, they can partially share common bases for all regressions, which identify the same correlation with regard to their spatial distribution across all tasks. This leads to a new type of model called linked multiway parameter analysis (LMPA).

The LMPA equips us with enhanced flexibility to decide the suitable number of regression parameters. However our intention is to further explore the correlation information of spatial distribution existent across different regression tasks. To this end, we add additional constraints on parameter components to formulate a new tensor-based regression model represented by Eq. (11). To be specific, for an individual regression, we decompose each factor component $\mathbf{u}_i^j$ in Eq. (10) into $\mathbf{u}_{iI}^j \mathbf{U}_{iC}$, where $\mathbf{u}_{iI}^j \in \mathbf{R}^{1 \times R_i}$, corresponding to task-dependent individual components and $\mathbf{U}_{iC} \in \mathbb{R}^{R_i \times I_i}$ (with $0 \leq R_i \leq I_i$), corresponding to the common bases for all regressions. This results in the following multiple tasks regression problem,

$$\min_{\mathbf{u}_I^j, \mathbf{U}_C} \sum_{j=1}^{K} \| \mathbf{y}^j - [\![\mathcal{X}; \mathbf{u}_{1I}^j \mathbf{U}_{1C}, \cdots, \mathbf{u}_{NI}^j \mathbf{U}_{NC}, \mathbf{U}_{N+1}]\!] \|_F^2 \tag{11}$$

Denote $\mathbf{u}_I^j = [\mathbf{u}_{1I}^j, \mathbf{u}_{2I}^j, \ldots, \mathbf{u}_{NI}^j]$ and $\mathbf{U}_C = [\mathbf{U}_{1C}, \mathbf{U}_{2C}, \ldots, \mathbf{U}_{NC}]$ for simplicity.

Without any prior knowledge or regularisation, all the entries of $\mathbf{u}_{iI}^j$ in problem (11) tend to be nonzero, so it is hard to pinpoint which predictor features are most relevant to the response in a task. To identify relevant features for any tasks, we propose to employ sparse regularization methods. The state-of-the-art l_1 norm regularization [23] is a commonly used approach in producing sparse solutions with many zeros, thus it helps in eliminating predictors that are not essential to the task. To this end, we add a l_1 norm regularization on each independent components $\mathbf{u}_{iI}^j$ to Eq. (11), then the problem can be re-expressed by

$$\begin{aligned} \min_{\mathbf{u}_I^j, \mathbf{U}_C} & \sum_{j=1}^{K} \| \mathbf{y}^j - [\![\mathcal{X}; \mathbf{u}_{1I}^j \mathbf{U}_{1C}, \cdots, \mathbf{u}_{NI}^j \mathbf{U}_{NC}, \mathbf{U}_{N+1}]\!] \|_F^2 \\ & + \lambda \sum_{j=1}^{K} \| \mathbf{u}_I^j \|_1 \end{aligned} \tag{12}$$

We employ an iterative algorithm called the Block Coordinate Descent (BCD) [3] to solve the optimisation problem (12) by fixing all the other variables to solve for one variable at a time alternatively.

First, the common components $\mathbf{U}_{iC}$ for $1 \leq i \leq N$ are fixed, the task-dependent coefficients $\mathbf{u}_I^j$ can be obtained by solving K independent tensor regression subproblems. That is, for each $j = 1, 2, \ldots, K$, $\mathbf{u}_I^j$ is obtained by minimizing

$$\min_{\mathbf{u}_I^j} \| \mathbf{y}^j - [\![\mathcal{X}; \mathbf{u}_{1I}^j \mathbf{U}_{1C}, \cdots, \mathbf{u}_{NI}^j \mathbf{U}_{NC}, \mathbf{U}_{N+1}]\!] \|_F^2 + \lambda \| \mathbf{u}_I^j \|_1 \tag{13}$$

with the BCD algorithm as well. For instance, LMPA-TR fixes $\mathbf{u}_{1I}^j, \ldots, \mathbf{u}_{n-1I}^j, \mathbf{u}_{n+1I}^j, \ldots, \mathbf{u}_{NI}^j$ to minimize the variable $\mathbf{u}_{nI}^j$, which is equivalent to solve the following problem

$$\min_{\mathbf{u}_{nI}^j} \| \mathbf{y}^j - [\![\mathcal{X}; \mathbf{u}_{1I}^j \mathbf{U}_{1C}, \cdots, \mathbf{u}_{NI}^j \mathbf{U}_{NC}, \mathbf{U}_{N+1}]\!] \|_F^2 + \lambda \| \mathbf{u}_{nI}^j \|_1 \tag{14}$$

Remark 2: Problem (14) can be considered as parameter estimation on the low-dimensional representation of $\mathcal{X}$. The common components $\mathbf{U}_{nC} (1 \leq n \leq N)$ can be extracted from each mode, and then perform the n-mode product with $\mathcal{X}$ to generate the low dimensional representation of $\mathcal{X}$. Next a Tucker decomposition on the new representation can be carried out, with parameters in each mode denoted by $\mathbf{u}_{nI}$.

This equivalent formulation of problem (14) is written as

$$\min_{\mathbf{u}_{nI}^j} \|\mathbf{y}^j - [\![\mathcal{G}; \mathbf{u}_{1I}^j, \cdots, \mathbf{u}_{NI}^j]\!]\|_F^2 + \lambda ||\mathbf{u}_{nI}^j||_1 \quad (15)$$

where $\mathcal{G} = \mathcal{X} \times_1 \mathbf{U}_{1C} \times_2 \cdots \times_N \mathbf{U}_{NC} \times_{N+1} \mathbf{U}_{N+1}$.

Using tensorial matricisation, problem (14) can be rewritten in terms of matrices as follows:

$$\min_{\mathbf{u}_{nI}^j} ||\mathbf{y}^j - \mathbf{u}_{nI}^j \mathbf{Q}||_F^2 + \lambda ||\mathbf{u}_{nI}^j||_1 \quad (16)$$

where $\mathbf{Q} = \mathbf{U}_{nC}\mathbf{X}_{(j)}(\mathbf{U}_{N+1} \otimes \mathbf{u}_{NI}^j\mathbf{U}_{NC} \cdots \otimes \mathbf{u}_{(n+1)I}^j\mathbf{U}_{(n+1)C} \otimes \mathbf{u}_{(n-1)I}^j\mathbf{U}_{(n-1)C} \cdots \otimes \mathbf{u}_{1I}^j\mathbf{U}_{1C})^T$. Problem (16) can be solved by the Orthogonal Matching Pursuit (OMP) algorithm [15], [16] or any basic pursuit algorithm [5], [22].

Secondly, we compute common bases $\mathbf{U}_C$ by fixing all the task-dependent components $\mathbf{u}_I^j$ for $(1 \leq j \leq K)$. $\mathbf{U}_C$ can be obtained by combining all the K regression models together, which is formulated as

$$\min_{\mathbf{U}_C} \sum_{j=1}^{K} \|\mathbf{y}^j - [\![\mathcal{X}; \mathbf{u}_{1I}^j\mathbf{U}_{1C}, \cdots, \mathbf{u}_{NI}^j\mathbf{U}_{NC}, \mathbf{U}_{N+1}]\!]\|_F^2 \quad (17)$$

Similarly, the mode-n common component $\mathbf{U}_{nC}$ $(1 \leq n \leq N)$ is computed by fixing $\mathbf{U}_{1C}, \ldots, \mathbf{U}_{(n-1)C}, \mathbf{U}_{(n+1)C}, \ldots, \mathbf{U}_{NC}$ by solving

$$\min_{\mathbf{U}_{nC}} \sum_{j=1}^{K} ||\mathbf{y}^j - [\![\mathcal{X}; \mathbf{u}_{1I}^j\mathbf{U}_{1C}, \cdots, \mathbf{u}_{NI}^j\mathbf{U}_{NC}, \mathbf{U}_{N+1}]\!]||_F^2 \quad (18)$$

Using tensorial matricisation, problem (18) can be rewritten in terms of matrices as follows:

$$\min_{\mathbf{U}_{nC}} \sum_{j=1}^{K} \|\mathbf{y}^j - \mathbf{u}_{nI}^j\mathbf{U}_{nC}\mathbf{X}_{(n)}\mathbf{V}_{nj}^T\|_F^2 \quad (19)$$

where $\mathbf{V}_{nj} = \mathbf{U}_{N+1} \otimes \mathbf{u}_{Nj}\mathbf{U}_{NC} \otimes \cdots \otimes \mathbf{u}_{(n+1)j}\mathbf{U}_{(n+1)C} \otimes \mathbf{u}_{(n-1)j}\mathbf{U}_{(n-1)C} \otimes \cdots \otimes \mathbf{u}_{1j}\mathbf{U}_{1C}$. This problem is equivalent to a least square problem. Let

$$\begin{aligned} f &= \sum_{j=1}^{K} \|\mathbf{y}^j - \mathbf{u}_{nI}^j\mathbf{U}_{nC}\mathbf{X}_{(n)}\mathbf{V}_{nj}^T\|_F^2 \\ &= \sum_{j=1}^{K} \mathrm{tr}(\mathbf{y}^j\mathbf{y}^{jT} - 2\mathbf{u}_{nI}^j\mathbf{U}_{nC}\mathbf{X}_{(n)}\mathbf{V}_{nj}^T\mathbf{y}^{jT} \\ &\quad + \mathbf{u}_{nI}^j\mathbf{U}_{nC}\mathbf{X}_{(n)}\mathbf{V}_{nj}^T\mathbf{V}_{nj}\mathbf{X}_{(n)}^T\mathbf{U}_{nC}^T{\mathbf{u}_{nI}^j}^T) \end{aligned} \quad (20)$$

$\mathbf{U}_{nC}$ is obtained by making the partial derivative of f with respect to $\mathbf{U}_{nC}$ equal to zero, which is written as

$$\frac{\partial f}{\partial \mathbf{U}_{nC}} = 2\sum_{j=1}^{K} \mathbf{A}_j\mathbf{U}_{nC}\mathbf{B}_j - 2\mathbf{C} = 0 \quad (21)$$

where $\mathbf{A}_j = {\mathbf{u}_{nI}^j}^T\mathbf{u}_{nI}^j$, $\mathbf{B}_j = \mathbf{X}_{(n)}\mathbf{V}_{nj}^T\mathbf{V}_{nj}\mathbf{X}_{(n)}^T$ and $\mathbf{C} = \sum_{j=1}^{K} {\mathbf{u}_{nI}^j}^T\mathbf{y}_j\mathbf{V}_{nj}\mathbf{X}_{(n)}^T$. The vectorization of $\mathbf{U}_{nC}$ can be obtained by solving Problem (21), that is,

$$\mathrm{vec}(\mathbf{U}_{nC}) = \left[\sum_{j=1}^{K} (\mathbf{B}_j^T \otimes \mathbf{A}_j)\right]^{-1} \mathrm{vec}(\mathbf{C}) \quad (22)$$

The final $\mathbf{U}_{nC}$ can be obtained by converting the result in Eq.(22) into a matrix format.

After iteratively solving subproblem (13) and (17) until the maximum iterations are achieved or the iteration converges, we finally obtain the common components across all the tasks and the independent parameter components along each mode for each of K regression tasks. Algorithm 1 outlines the whole process of our proposed method LMPA-TR.

Algorithm 1 Tensor Regression Based on Linked Multiway Parameter Analysis (LMPA-TR)

Require: K output vectors $\mathbf{y}^j$ for $1 \leq j \leq K$, input features $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times M}$, sparsity S in the OMP algorithm and tolerance ϵ.

Ensure: Common components across all tasks $\mathbf{U}_C = [\mathbf{U}_{1C}, \mathbf{U}_{2C}, \ldots, \mathbf{U}_{NC}]$ and regression parameters for task-j $\mathbf{u}_I^j = [\mathbf{u}_{1I}^j, \mathbf{u}_{2I}^j, \ldots, \mathbf{u}_{NI}^j]$ $(1 \leq j \leq K)$

1: Initialize common components along each mode $\mathbf{U}_{nC}(1 \leq n \leq N)$
2: **while** reach maximum iteration times or converge to stop **do**
3: Get the independent component for mode-n of task-j: $\mathbf{u}_{nI}^j$ using the OMP algorithm for ($j = 1$ to K and $n = 1$ to N);
4: Update the common components for mode-n: $\mathbf{U}_{nC}$ by solving the problem (18) for ($n = 1$ to N);
5: **end while**
6: **return** $\mathbf{U}_C$ and $\mathbf{u}_I^j(1 \leq j \leq K)$

IV. Experimental Results

Our baseline methods include two classic vector-based regression algorithms (i.e. ridge regression (RR) [18] and support vector regression (SVR) [20]) and a generalized tensor regression model based on CP decomposition [9] (CPTR). As our regression model is based on Tucker decomposition, we also consider a Tucker tensor regression model (TTR), which allows directly simultaneous projections of an original input tensor to each direction along each mode, with a sparsity regularisation on each mode. To facilitate a fair comparison and to illustrate the advantage of the joint dimensionality reduction and regression framework (as explained in *Remark 2*), we also include a method called Tucker Dimensionality Reduction + TTR (TDR+TTR) that performs a Tucker decomposition as dimensionality reduction and TTR on the new low-dimensional representation as regression, but in a disjoint manner.

In order to investigate the performance of the proposed method, we conducted experiments using two publicly available real world data sets for the problem of head-pose, i.e., IDIAP [1] and Pointing'04 [8] data sets[1].

I. IDIAP Data set: The IDIAP Head Pose dataset comes from 8 meeting sequences of 360×288 frame resolution, where two individuals were captured while discussing about various topics in a 4-person dialogue scenario. The total number of different subjects captured is 15. They had their head orientations continuously annotated using a magnetic field location

[1]The code and data can be downloaded from https://sites.google.com/site/yifanfu01/code

TABLE I. ALGORITHM MEMORY AND TIME COMPARISONS

(a) Parameter Number for Each Regression Estimation

	IDIAP	Pointing'04
RR	5625	110592
SVR	5625	110592
CPTR	450	2021
TTR	150	672
TDR+TTR	50	180
LMPA-TR	50	180

(b) Algorithm Running Time Comparison

	IDIAP	Pointing'04
RR	3069.26s	11838.26s
SVR	2895.98s	10936.85s
CPTR	269.45s	1804.26s
TTR	275.02s	1854.62s
TDR+TTR	597.26s	3981.23s
LMPA-TR	589.23s	3269.39s

and orientation sensor tracker. 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 75×75 pixels. The ground truth provided is in the form of pan, tilt and roll angles (i.e. Euler angles with respect to the camera coordination system). The video repository has been employed for the CLEAR2007 head orientation estimation challenge, following the protocol described in [1]: 21152 samples were selected as training data and 23991 as testing data. Since the training samples are particularly unbiased on certain orientations, we flip them and then we randomly extract a subset of 5288 images from above training data as our experimental database.

II. Pointing'04 Data set: The Pointing'04 head-pose database contains a variety of head poses ranging from -90 degrees to $+90$ degrees in both horizontal and vertical directions. The data set is formed by 15 subjects of various skin colours, with or without glasses, each one preforming 13 pose variations horizontally and seven vertically as well as two extreme cases of the vertical $+90$ degrees and -90 degrees, to a total of 2790 images. All the images are of size 384×288.

1) Performance with respect to different training sample sizes: It is noted that the majority of linear projection techniques follow an implicit assumption of nearest neighbour, that is, the local structure is preserved in the low dimensional subspace. Linear regressors are simple, yet effective for the training sets with clear nearest neighbour characteristic. However, it might not be the optimal choice when training samples do not have such characteristic. Therefore, we investigate the performance of LMPA-TR with *small* size of training samples without explicit nearest neighbour property.

To generate such training sets, we generated 20 random splits on the IDIAP and Pointing'04 data sets. In each split, the images are randomly selected from each subject for training and the rest are used for testing. Then we report the average performance on all these splits with respect to different size of training samples in Table II and III. Among various methods, we note that LMPA-TR performs best with different training sample sizes ranging from 3 to 7.

As we expected, LMPA-TR based on optimisation on the subspaces (i.e. common bases) outperforms all the linear projection based methods. Although tensor based linear regression methods like TTR and CPTR take spatial structure into consideration, they still follow the nearest neighbour characteristic that makes them ineffective for small training sets selected in a random manner. When we compare our proposed method LMPA-TR to TDR+TTR, better performance gain is observed. This is probably because separating dimension reduction and regression updates common bases and task independent bases independently, component information obtained by one task is not used for the other optimisation task.

2) Comparison regarding Memory cost and Running time: We compare the memory cost and the running time on both data sets among all the baseline methods in Table I. Compared with vector-based methods, the number of parameters used in tensor-based counterparts is significantly reduced as shown in Table I(a). This observation suggests that taking spatial structure into consideration can effectively reduce memory cost with respect to the regression parameters, especially for high-dimensional data set like Pointing'04.

Moreover, integrating hidden subspace exploitation (i.e. dimension reduction) into tensor regression can further reduce the memory usage for regression parameters. In terms of algorithms' running time, it is apparent that vector-based techniques like RR and SVR have the highest computational cost, while tensor-based regression methods CPTR and TTR are much faster than RR and SVR as the number of estimated parameters is much less in a format of tensor. We also note that TDR+TTR and LMPA-TR take longer time to run than CPTR and TTR, because they need to solve the optimisation problems on dimension reduction and regression parameter estimation.

To sum up, our proposed method LMPA-TR has small memory requirements and comparable computational cost.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a new tensor regression method based on linked multiway parameter analysis. The proposed method identifies the common components of parameters across all the regression estimations and the independent ones for each specific regression task simultaneously. Meanwhile, sparse coding is employed for the estimation of task-dependent parameters along each mode, which further reduces memory requirements regarding these parameters. This sparsity scheme generates discriminative and general regression models by identifying the most relevant factors for each particular regression task, effectively avoiding the overfitting problem. Experimental results show that our new method has advantages over state-of-art regression methods in memory and computational cost and convergence speed, especially when the size of training samples is small.

In this work, we have chosen the quadratic loss over the learning function, which is particularly satisfactory for regression problem. When it comes to classification tasks, it is reported that the quadratic loss does not work well for categorial outputs [17]. One of our future work is to use our framework with other types of loss functions, rather than the quadratic loss. For example, one can incorporate the idea of learning the projection matrix in the context of SVM formulation for classification of multi-dimensional data.

ACKNOWLEDGMENT

This work is supported by the Australian Research Council (ARC) through Discovery Project Grant DP130100364.

TABLE II. ANGULAR ERROR FOR THE IDIAP DATA SET

		RR	SVR	CPTR	TTR	TDR+TTR	LMPA-TR
Sample size=3	pan	$23.8_{\pm 1.81}$	$20.6_{\pm 1.37}$	$20.7_{\pm 1.27}$	$19.8_{\pm 1.34}$	$17.2_{\pm 1.17}$	$\mathbf{16.1}_{\pm 0.97}$
	tilt	$8.5_{\pm 1.42}$	$9.2_{\pm 1.28}$	$8.7_{\pm 1.93}$	$8.9_{\pm 1.66}$	$7.5_{\pm 1.35}$	$\mathbf{6.1}_{\pm 1.14}$
	roll	$12.3_{\pm 2.01}$	$11.6_{\pm 1.88}$	$10_{\pm 1.12}$	$9.7_{\pm 1.39}$	$9.19_{\pm 1.06}$	$\mathbf{8.3}_{\pm 1.01}$
Sample size=5	pan	$21.2_{\pm 1.54}$	$18.4_{\pm 1.27}$	$16.9_{\pm 1.98}$	$15.4_{\pm 1.13}$	$12.7_{\pm 2.04}$	$\mathbf{10.5}_{\pm 1.45}$
	tilt	$7.4_{\pm 1.27}$	$7.9_{\pm 2.56}$	$5.3_{\pm 1.67}$	$5.5_{\pm 1.09}$	$3.2_{\pm 1.33}$	$\mathbf{2.9}_{\pm 0.87}$
	roll	$11.6_{\pm 1.69}$	$9.8_{\pm 1.73}$	$8.6_{\pm 1.51}$	$7.8_{\pm 2.09}$	$7.1_{\pm 1.86}$	$\mathbf{6.9}_{\pm 1.12}$
Sample size=7	pan	$19.6_{\pm 1.74}$	$16.7_{\pm 1.98}$	$12.7_{\pm 1.52}$	$13.01_{\pm 2.63}$	$9.7_{\pm 1.54}$	$\mathbf{8.2}_{\pm 1.09}$
	tilt	$5.6_{\pm 2.47}$	$5.8_{\pm 1.23}$	$3.6_{\pm 1.48}$	$4.1_{\pm 1.60}$	$1.8_{\pm 0.65}$	$\mathbf{1.5}_{\pm 0.38}$
	roll	$9.8_{\pm 3.05}$	$8.5_{\pm 1.64}$	$6.2_{\pm 1.23}$	$5.9_{\pm 0.98}$	$5.4_{\pm 0.96}$	$\mathbf{4.9}_{\pm 0.93}$

TABLE III. ANGULAR ERROR FOR THE POINTING'04 DATA SET

		RR	SVR	CPTR	TTR	TDR+TTR	LMPA-TR
Sample size=3	horizontal	$5.9_{\pm 0.81}$	$5.7_{\pm 1.07}$	$5.3_{\pm 1.72}$	$5.4_{\pm 0.98}$	$4.9_{\pm 1.23}$	$\mathbf{4.6}_{\pm 0.27}$
	vertical	$5.4_{\pm 1.09}$	$5.3_{\pm 1.36}$	$4.9_{\pm 1.56}$	$4.8_{\pm 1.08}$	$4.6_{\pm 1.77}$	$\mathbf{4.2}_{\pm 1.23}$
Sample size=5	horizontal	$5.4_{\pm 1.36}$	$5.2_{\pm 1.08}$	$4.8_{\pm 1.58}$	$4.6_{\pm 1.38}$	$4.2_{\pm 0.98}$	$\mathbf{3.9}_{\pm 1.31}$
	vertical	$4.8_{\pm 1.67}$	$5.0_{\pm 1.25}$	$4.6_{\pm 1.02}$	$4.2_{\pm 1.52}$	$3.9_{\pm 0.95}$	$\mathbf{3.7}_{\pm 0.66}$
Sample size=7	horizontal	$4.6_{\pm 1.32}$	$4.2_{\pm 1.51}$	$3.9_{\pm 1.09}$	$3.7_{\pm 1.52}$	$3.5_{\pm 0.88}$	$\mathbf{3.1}_{\pm 0.95}$
	vertical	$4.2_{\pm 1.23}$	$4.1_{\pm 2.04}$	$3.8_{\pm 1.18}$	$3.6_{\pm 1.26}$	$3.2_{\pm 0.87}$	$\mathbf{2.9}_{\pm 0.96}$

REFERENCES

[1] S. Ba and J. Odobez, "Evaluation of multiple cue head pose estimation algorithms in natural environements," in *Multimedia and Expo IEEE International Conference on*, July 2005, pp. 1330–1333.

[2] B. Blankertz, G. Curio, and K. Mller, "Classifying single trial EEG: Towards brain computer interfacing," *NIPS*, pp. 157–164, 2002.

[3] M. Blondel, K. Seki, and K. Uehara, "Block coordinate descent algorithms for large-scale sparse multiclass classification," *Machine Learning*, vol. 93, no. 1, pp. 31–52, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10994-013-5367-2

[4] B. S. Caffo, C. M. Crainiceanu, G. Verduzco, S. Joel, S. H. Mostofsky, S. S. Bassett, and J. J. Pekar, "Two-stage decompositions for the analysis of functional connectivity for fMRI with application to alzheimer's disease risk," *NeuroImage*, vol. 51, no. 3, pp. 1140 – 1149, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811910002685

[5] M. Donoho, D. L.and Elad, "Optimally sparse representation in general (non-orthogonal) dictionaries via l1 minimization," in *Proc. Natl Acad. Sci.*, 2003, pp. 2197–2202.

[6] C. Gao and X. Wu, "Kernel support tensor regression," *Procedia Engineering*, vol. 29, pp. 3986 – 3990, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877705812006169

[7] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale Bayesian logistic regression for text categorization," *Technometrics*, vol. 49, pp. 291–304(14), 2007.

[8] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial structures," in *In Proceedings of ICPR, International Workshop on Visual Observation of Deictic Gestures*, 2004.

[9] W. Guo, I. Kotsia, and I. Patras, "Tensor learning for regression," *Image Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 816–827, Feb 2012.

[10] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/07070111X

[11] D. Kontos, V. Megalooikonomou, N. Ghubade, and C. Faloutsos, "Detecting discriminative functional MRI activation patterns using space filling curves," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Sept 2003, pp. 963–966.

[12] S. LaConte, S. Strother, V. Cherkassky, J. Anderson, and X. Hu, "Support vector machines for temporal classification of block design fMRI data," *NeuroImage*, vol. 26, no. 2, pp. 317 – 329, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811905000893

[13] X. Li, H. Zhou, and L. Li, "Tucker tensor regression and neuroimaging analysis," *http://arxiv.org/abs/1304.5637*, 2013.

[14] T. Mitchell, R. Hutchinson, R. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman, "Learning to decode cognitive states from brain images," *Machine Learning*, vol. 57, no. 1-2, pp. 145–175, 2004. [Online]. Available: http://dx.doi.org/10.1023/B%3AMACH.0000035475.85309.1b

[15] D. Needell, J. Tropp, and R. Vershynin, "Greedy signal recovery review," in *Proceedings of the 42nd Asilomar Conf. Signals, Systems and Computers, Pacific Grove*. IEEE, 2008, pp. 1048–1050.

[16] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Commun. ACM*, vol. 53, no. 12, pp. 93–100, Dec. 2010. [Online]. Available: http://doi.acm.org/10.1145/1859204.1859229

[17] D.-S. Pham and S. Venkatesh, "Robust learning of discriminative projection for multicategory classification on the stiefel manifold," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, June 2008, pp. 1–7.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. New York, USA: Cambridge University Press, 2007.

[19] S. V. Shinkareva, H. C. Ombao, B. P. Sutton, A. Mohanty, and G. A. Miller, "Classification of functional brain images with a spatio-temporal dissimilarity map," *NeuroImage*, vol. 33, no. 1, pp. 63 – 71, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811906007105

[20] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004. [Online]. Available: http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88

[21] X. Tan, Y. Zhang, S. Tang, J. Shao, F. Wu, and Y. Zhuang, "Logistic tensor regression for classification," in *Intelligent Science and Intelligent Data Engineering*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 7751, pp. 573–581. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36669-7_70

[22] J. Tropp, "Greed is good: algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, 2004.

[23] J. Wang, K.-H. Lee, and K.-S. Leung, "L1-norm regularization based nonlinear integrals," ser. Lecture Notes in Computer Science, vol. 5551. Springer, 2009, pp. 201–208. [Online]. Available: http://dblp.uni-trier.de/db/conf/isnn/isnn2009-1.html#WangLL09

[24] Q. Zhao, C. F. Caiafa, D. P. Mandic, L. Zhang, T. Ball, A. Schulze-bonhage, and A. S. Cichocki, "Multilinear subspace regression: An orthogonal tensor decomposition approach," in *Advances in Neural Information Processing Systems 24*, J. Shawe-taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 1269–1277. [Online]. Available: http://books.nips.cc/papers/files/nips24/NIPS2011_0748.pdf

[25] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *Journal of American Statistical Association*, vol. 108, pp. 540–552, 2013.