HIGHER ORDER SUPPORT TENSOR REGRESSION FOR HEAD POSE ESTIMATION

Weiwei Guo

Queen Mary, University of London
National University of Defence Technology
Changsha, China

Irene Kotsia, Ioannis Patras

Queen Mary, University of London
London, UK

ABSTRACT

In this paper, we exploit the advantages of tensor representations and propose a Supervised Multilinear Learning Model for regression. The model is based on the Canonical (CANDECOMP)/Parallel Factors (PARAFAC) decomposition of tensors of multiple modes and allows the simultaneous projection of an input tensor to more than one discriminative directions along each mode. These projection weights are obtained by optimizing a ϵ-insensitive loss functions which leads to generalized Support Tensor Regression (STR). The methods are validated on the problems of head pose estimation using real data from publicly available databases.

1. INTRODUCTION

Tensors are efficient representations of multidimensional objects whose elements can be accessed with two or more indices. They constitute the most natural representation of visual data such as images (2^{nd} order tensors), image sequences (3^{rd} order tensors), etc. However, most of the existing algorithms for visual analysis operate in a vector space that is derived by stacking the original image (sequence) elements.

Recently, general multilinear learning techniques that deal with data represented as matrices or higher order tensors were shown to outperform their equivalent vector methods, especially for small sample problems [1, 2, 3]. For dimensionality reduction, two dimensional Principal Component Analysis (PCA) was introduced in [4] representing an image as its natural matrix formation (2^{nd} order tensor) instead of its vectorized form. The image, was then projected to the principle components along both horizonal and vertical directions. A 2D manifold learning based on graph embedding was presented in [5]. In [6], the bilinear subspace analysis was generalized to a higher-order multilinear PCA. Linear Discriminative Analysis (LDA) was also generalized to deal with tensors for face and gait recognition in [7, 8]. Multilinear analysis can be also applied in multiple factor analysis such as in the case of "TensorFaces" introduced in [9]. There, the authors explicitly represent images from multiple imaging factors, such as illumination, viewpoint and scene structure, as tensors. As far as classification is concerned, several methods that deal with tensors have been recently proposed [1, 2, 10]. However, these methods either use 1st order tensors, thus allowing projections at each mode along a single direction, or do not deal with tensors with more than two modes.

The problem of regression is one of the main learning problems and has been extensively addressed in the literature [11, 12]. Linear regression using a square loss function is one of the simplest and earliest approaches. However, to the best of our knowledge, none of the regression methods utilize tensor representations. In this paper, we propose a supervised Multilinear Learning Model that deals with regression. This is the first work that addresses the regression problem using tensor representations. We adopt a linear regression model that is based on the inner product of the data tensor $\mathcal{X}$ and the tensor $\mathcal{W}$ of the (unknown) parameters. That is $f(\mathcal{X}) = \langle \mathcal{X}, \mathcal{W} \rangle + b$. The unknown tensor $\mathcal{W}$ is learned through optimizing a ϵ-insensitive loss function

The rest of the paper is organized as follows. In Section 2 we present some useful notations that will be used throughout the paper. In Section 3 we describe the proposed generalized Supervised Tensor Learning framework using the ϵ-insensitive loss function leading to generalized Support Tensor Regression (STR) algorithm . In Section 4, we report experimental results for head pose estimation using publicly available datasets. Finally in Section 5 we draw some conclusions and discuss future work.

2. NOTATIONS AND PRELIMINARIES

We will first briefly describe some useful notions and concepts of tensor algebra that will be used throughout the paper. 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., $\mathcal{X} \in I\!R^{I_1 \times I_2 \cdots \times I_M}$, M the number of modes. The i^{th}-element of a vector $\mathbf{x} \in I\!R^{I}_{+}$ is denoted by x_i, $i = 1, 2, \ldots, I$. The elements of a M order tensor $\mathcal{X}$ will be denoted by $x_{i_1 i_2 \ldots i_n}$,$i_\ell = 1, 2, \ldots, I_\ell$, $\ell = 1, 2, \ldots, M$.

The d-mode matricization (or unfolding, flattening) of an M-order tensor $\mathcal{X} \in I\!R^{I_1 \times I_2 \cdots \times I_M}$, denoted by $\boldsymbol{X}_{(d)} \in$

$I\!R^{I_d\times(I_1\times\cdots\times I_{d-1}\times I_{d+1}\times I_M)}$ or $\mathrm{mat}_d(\mathcal{X})$, is the reordering of the tensor elements into a matrix, such that the d-mode fibres become the columns of the final matrix. The vectorization of a tensor $\mathcal{X}$, $\mathrm{vect}(\mathcal{X})$, is defined by stacking its elements into a vector following a certain order.

The d-mode product of a tensor $\mathcal{X}\in I\!R^{I_1\times I_2\cdots\times I_M}$ with a matrix $\boldsymbol{U}\in I\!R^{J\times I_d}$ is a tensor of size $I_1\times I_{d-1}\times J\times I_{d+1}\times I_M$, defined as in [13].

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

The inner product of two tensors of the same size $\mathcal{X},\mathcal{Y}\in I\!R^{I_1\times I_2\cdots\times I_M}$ is defined as $\langle\mathcal{X},\mathcal{Y}\rangle=\sum_{i_1=1}^{I_1}\cdots\sum_{i_M=1}^{I_M} x_{i_1\cdots i_M}y_{i_1\cdots i_M}$. The *Frobenius* norm of a tensor is thus defined as $\|\mathcal{X}\|_{\mathrm{Fro}}=\sqrt{\langle\mathcal{X},\mathcal{X}\rangle}$. It can be shown that $\|\mathcal{X}\|=\|\boldsymbol{X}_{(d)}\|=\sqrt{\boldsymbol{X}_{(d)}\boldsymbol{X}_{(d)}^{\mathrm{T}}}$.

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

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

The operator "$\circ$" is the outer product of vectors and $\boldsymbol{U}^{(k)}=[\boldsymbol{u}_1^{(k)},\cdots,\boldsymbol{u}_R^{(k)}]$. The size of the CP decomposition is equal to $I_k\times R, k=1,2,\cdots,M$. The rank of a tensor $\mathcal{X}$, denoted as $R=\mathrm{rank}(\mathcal{X})$, is the smallest number of rank-one tensors whose sum can accurately generate $\mathcal{X}$.

3. GENERALIZED TENSOR REGRESSION

Let us define a linear predictor in the vector space as $y=f(\boldsymbol{x};\boldsymbol{w},b)=\langle\boldsymbol{x},\boldsymbol{w}\rangle+b$, 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. We consider here scalar output regression, and extend the above mentioned classic linear predictor from the vector space into the tensor space as

$$y=f(\mathcal{X};\mathcal{W},b)=\langle\mathcal{X},\mathcal{W}\rangle+b, \tag{2}$$

where $\mathcal{X}\in I\!R^{I_1\times I_2\cdots\times I_M}$ contains the input features M mode tensor. $\mathcal{W}$ is the weight tensor, with number of modes and dimensions equal to the ones of the data tensor $\mathcal{X}$, and b is a scalar bias.

In order to perform feature selection or dimensionality reduction, we constrain the weight tensor $\mathcal{W}\in I\!R^{I_1\times I_2\cdots\times I_M}$ to be a sum of R rank-one tensors, following the CP composition:

$$\mathcal{W}=\sum_{r=1}^{R}\boldsymbol{u}_r^{(1)}\circ\boldsymbol{u}_r^{(2)}\circ\cdots\circ\boldsymbol{u}_r^{(M)}\triangleq[\![\,\boldsymbol{U}^1,\boldsymbol{U}^2,\cdots,\boldsymbol{U}^M]\!], \tag{3}$$

where $\boldsymbol{U}^j=[\boldsymbol{u}_1^{(j)},\cdots,\boldsymbol{u}_R^{(j)}]$. By substituting Eq. 3 in Eq. 2 we get

$$\begin{aligned}y&=\langle\mathcal{X},\mathcal{W}\rangle+b=\langle\mathcal{X},\sum_{r=1}^{R}\boldsymbol{u}_r^{(1)}\circ\boldsymbol{u}_r^{(2)}\circ\cdots\circ\boldsymbol{u}_r^{(M)}\rangle+b\\&=\sum_{r=1}^{R}\mathcal{X}\prod_{k=1}^{M}\times_k\boldsymbol{u}_r^{(k)}+b.\end{aligned} \tag{4}$$

As can be seen in Eq. 4, the input features $\mathcal{X}$ are projected along R directions for each mode k. This reduces the loss of information that occurs when the projection is performed along only one direction [6]. Such projections can also be interpreted as a dimensionality reduction or feature selection scheme. This decomposition reduces the number of parameters that need to be estimated from $\prod_{k=1}^{M}I_k$ (i.e. the number of elements of the tensor $\mathcal{W}$) to $R\sum_{k=1}^{M}I_k$.

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

$$L(\Theta)=\frac{1}{2}\sum_{i=1}^{N}l(y_i,f(\mathcal{X}_i;\Theta))+\frac{\lambda}{2}\|\mathcal{W}\|_{\mathrm{Fro}}^2. \tag{5}$$

where the empirical data cost function is the ϵ-insensitive loss function $l=\max(0,|y-f|-\epsilon)$, then following the support vector regression (SVR) methodology we arrive at the following optimization problem:

$$\min_{\mathcal{W},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}}\quad\frac{1}{2}\|\mathcal{W}\|_{\mathrm{Fro}}^2+C\sum_{i=1}^{N}(\xi_i+\hat{\xi}_i), \tag{6a}$$

$$s.t.-\epsilon-\xi_i\le y_i-\langle\mathcal{X}_i,\mathcal{W}\rangle-b\le\epsilon+\xi_i, \tag{6b}$$

$$\epsilon\ge0,\ \xi_i\ge0,\ \hat{\xi}_i\ge0,\ \forall i=1,2,\cdots,N. \tag{6c}$$

Notice that this is a reformulation of Eq. 5. In order to solve Eq. 6, we optimize the cost function with respect $\boldsymbol{U}^{(j)}$ while keeping the other $\boldsymbol{U}^{(k)},k\neq j$ fixed. That is,

$$\min_{\boldsymbol{U}^{(j)},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}}\quad\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{U}^{(j)}\boldsymbol{U}^{(-j)\mathrm{T}}\boldsymbol{U}^{(-j)}\boldsymbol{U}^{(j)\mathrm{T}}\right)+C\sum_{i=1}^{N}(\xi_i+\hat{\xi}_i), \tag{7a}$$

$$s.t.\epsilon-\xi_i\le y_i-\mathrm{Tr}(\boldsymbol{U}^{(j)}\boldsymbol{U}^{(-j)\mathrm{T}}\boldsymbol{X}_{i(j)}^{\mathrm{T}})-b\le\epsilon+\xi_i, \tag{7b}$$

$$\epsilon\ge0,\ \xi_i\ge0,\ \hat{\xi}_i\ge0,\ \forall i=1,2,\cdots,N. \tag{7c}$$

Following the approach presented in [2], if we denote by $\boldsymbol{B}=\boldsymbol{U}^{(-j)\mathrm{T}}\boldsymbol{U}^{(-j)}$, $\widetilde{\boldsymbol{U}}^{(j)}=\boldsymbol{U}^{(j)}\boldsymbol{B}^{\frac{1}{2}}$ and $\widetilde{\boldsymbol{X}}_{i(j)}=\boldsymbol{X}_{i(j)}\boldsymbol{U}^{(-j)}\boldsymbol{B}^{\frac{1}{2}}$, the optimization problem Eq. 7 can be rewritten as:

$$\min_{\boldsymbol{U}^{(j)}, b, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}} \quad \frac{1}{2}\mathrm{Tr}(\widetilde{\boldsymbol{U}}^{(j)}\widetilde{\boldsymbol{U}}^{(j)\mathrm{T}}) + C\sum_{i=1}^{N}(\xi_i+\hat{\xi}_i), \tag{8a}$$

$$s.t. - \epsilon - \xi_i \leq y_i - \mathrm{Tr}(\widetilde{\boldsymbol{U}}^{(j)}\widetilde{\boldsymbol{X}}^{\mathrm{T}}_{i(j)}) - b \leq \epsilon + \xi_i, \tag{8b}$$

$$\epsilon \geq 0, \, \xi_i \geq 0, \, \hat{\xi}_i \geq 0, \, \forall i = 1, 2, \cdots, N. \tag{8c}$$

If we vectorize $\widetilde{\boldsymbol{U}}^{(j)}, \widetilde{\boldsymbol{X}}_{i(j)}$, the problem Eq. 8 can be easily solved using a typical SVM/SVR optimizer. Once we obtain $\widetilde{\boldsymbol{U}}^{(j)}$, we can then solve for $\boldsymbol{U}^{(j)}$ as

$$\boldsymbol{U}^{(j)} = \widetilde{\boldsymbol{U}}^{(j)}\boldsymbol{B}^{-\frac{1}{2}}. \tag{9}$$

The algorithm is summarized in Algorithm 1

Algorithm 1 GENERALIZED SUPERVISED TENSOR LEARNING ALGORITHM

```
Input: The set of training tensors and their corresponding
    targets, that is $\{\mathcal{X}_i, y_i\}|_{i=1}^{N}$.
Output: The weights $\{\boldsymbol{U}^1, \cdots, \boldsymbol{U}^M\}$ and the bias term $b \in \mathbb{R}$ that minimize the objective function.
    Initialize randomly $\{\boldsymbol{U}^{(1)}, \cdots, \boldsymbol{U}^{(M)}\}^{(0)}$.
    repeat
        $t \leftarrow t + 1$
        for $k = 1$ to $M$ do
            Solve with respect to $\boldsymbol{U}^{(k)}|_{(t)}$:
            For Support Tensor Regression (STR) solve Eq. 8.
        end for
    until $\|\mathcal{W}^{(t)} - \mathcal{W}^{(t-1)}\|/\|\mathcal{W}^{(t-1)}\| \leq \varepsilon$ or $t \geq T_{\max}$
```

4. EXPERIMENTAL RESULTS

We conducted two experiments on the two publicly available IDIAP head pose [14] and Boston University head pose datasets [15, 16]. Regarding the IDIAP database, we chose 3000 frames for training and 6000 frames for testing. The Boston University (BU) dataset consists of 45 video sequences, depicting 5 subjects performing 9 different motions under uniform illumination in a standard office setting. The training and testing subsets are chosen manually. A face detector was used to extract the bounding box of each face in every video frame. All the acquired image regions were resized to 40×30 pixels. Two types of features were extracted, the normalized pixel intensity and the log-Gabor features. Each of the normalized images formed a 40×30 2nd order tensor that was used as input to the proposed tensor-based algorithms. The head pose Euler angles $\{\alpha, \beta, \gamma\}$, corresponding to pan, tilt and roll in the IDIAP dataset and roll, yaw and pitch in the BU dataset, respectively, were calculated from the rotation matrix of the head configuration with respect to the camera position.

Table 1. Estimation angular errors for both datasets.

	IDIAP		BU	
	SVR	STR	SVR	STR
Pan/Roll	21.9	20.7	5.8	5.6
Tilt /Yaw	9.0	9.2	5.3	4.9
Roll/Pitch	11.3	11.7	4.8	4.8
Pointing	25.0	23.7	8.7	8.2

We report the mean angular error of the pointing vector defined by $\{\alpha, \beta, \gamma\}$ and the mean absolute error for each of $\{\alpha, \beta, \gamma\}$ [14]. The rank R and the regularization parameter C are 1, 0.005, and 3, 0.1 for IDIAP and BU dataset, respectively,at which the lowest testing errors arrived. The errors are reported in Table 1 with comparison with the estimation errors through support vector regression.

The rank one 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 are visualized as gray images in Fig. 1, taking the weights for the output roll in BU dataset for example. It is clear that the tensor weights for the cases STR form better and more clear spatial patterns compared with the vector case.

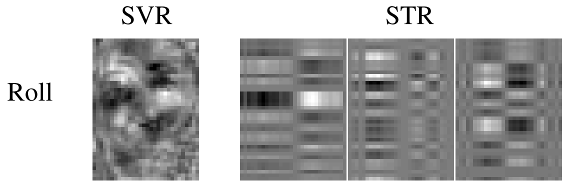

Fig. 1. The images of weights obtained for the BU dataset

As we mentioned before, the projections along the tensor modes that is performed by the matrices $U^{(d)}$ can be interpreted as a form of dimension reduction, or feature extraction. In the proposed method, this is performed in a supervised manner for regression. Below we compare with the results obtained if we perform unsupervised dimension reduction either in the vector (PCA) or in the tensor (MPCA [6]) representation before a classic vector-based regression algorithm (ridge regression (RR) or SVR) is trained. The number of PCA (or MPCA) components are chosen so that around 97% of the energy is preserved. The results are summarized in Table 2, where it is clear that application of the unsupervised dimension reduction methods has not lead to results comparable to the ones obtained by the proposed direct tensor-based regression.

5. CONCLUSION AND FUTURE WORK

A novel unified Supervised Multilinear Learning Model that deals with regression is proposed in this paper. The presented method enables the simultaneous projections of an input tensor to more than one discriminative directions along each mode,

Table 2. 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	STR
Intensity	IDIAP	28.1	25.0	28.0	25.6	28.1	25.6	23.7
	BU	9.0	8.7	9.0	8.6	9.0	8.7	8.2
Log-Garbor	IDIAP	33.0	32.3	31.68	30.8	35.2	30.8	25.6
	BU	10.3	10.0	10.0	10.1	9.8	9.8	8.4

exploiting the properties of the Canonical (CANDECOMP)/ Parallel Factors (PARAFAC) decomposition of a tensor. Experiments performed using publicly available real data for the problems of head pose and body pose estimation showed that tensors-based algorithms consistently outperform vector-based ones.

For the tensors-based regressors proposed in this paper, the optimal rank can be obtained using cross-validation. In the future we will attempt to achieve automatic determination of the rank number, based on matrix track norm regularization. Also, we intend to address the limitations introduced by the fact that the proposed models are global and linear. We will do so either by adopting local regression schemes or by adopting non-linear models.

6. ACKNOWLEDGEMENT

This research is supported by the EPSRC grant 'Recognition and Localization of Human Actions in Image Sequences' (EP/G033935/1).

7. REFERENCES

[1] L. Wolf, H. Jhuang, and T. Hazan, "Modeling appearances with low-rank svm," in *CVPR*, Washington, DC, USA, Jun. 2007.

[2] H. Pirsiavash, D. Ramanan, and C. Fowlkes, "Bilinear classifiers for visual recognition," in *Neural Information Processing Systems*, Vancouver, Canada, Dec. 2009.

[3] D. Tao, X. Li, X. Wu, W. Hu, and S.J. Maybank, "Supervised tensor learning," *Knowledge and Information Systems*, 2007.

[4] J. Ye, R. Janardan, and Q. Li, "Gpca: an efficient dimension reduction scheme for image compression and retrieval," in *ACM SIGKDD Intertional conference on knowledge discovery and data mining*, Seattle, WA, USA, Aug. 2004.

[5] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Neural Information Processing Systems*, Canada, Dec. 2006.

[6] H. Lu, KN Plataniotis, and A.N. Venetsanopoulos, "Mpca: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[7] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Trans. Image Processing*, vol. 16, no. 1, pp. 212, 2007.

[8] D. Tao, X. Li, X. Wu, and S.J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1700, 2007.

[9] M.A.O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *ECCV*, Copenhagen, Denmark, May. 2002.

[10] R. Tomioka and K. Aihara, "Classifying matrices with a spectral regularization," in *International Conferenceon Machine Learning*, Corvallis, USA, Jun. 2007.

[11] C.M. Bishop et al., *Pattern recognition and machine learning*, 2006.

[12] Michael E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[13] T.G. Kolda and B.W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[14] S. Ba and J. Odobez, "Evaluation of multiple cues head pose tracking algorithms in indoor environments," in *International Conference on Multimedia and Exposition*, Amsterdam, July 2005.

[15] 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 Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, 2000.

[16] R. Valenti, Z. Yucel, and T. Gevers, "Robustifying eye center localization by head pose cues," in *CVPR*, June 2009, Miami, USA.