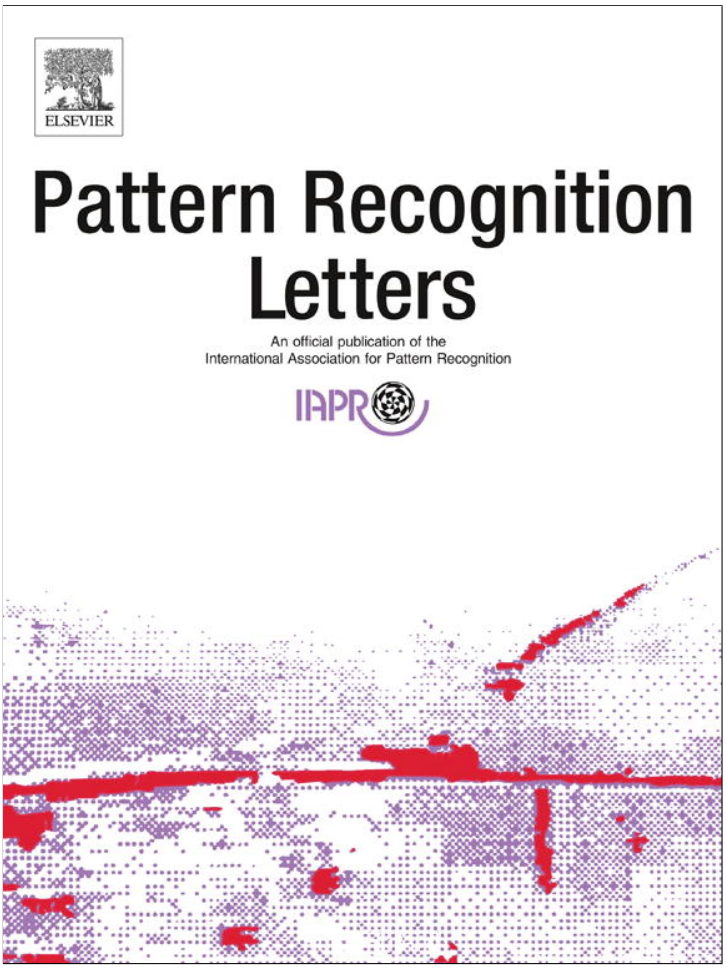

Pattern Recognition Letters 34 (2013) 1476–1484

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

# Supervised feature extraction for tensor objects based on maximization of mutual information

Ante Jukić *,[1], Marko Filipović

*Division of Laser and Atomic Research and Development, Ruđer Bošković Institute, Bijenička cesta 54, 10000 Zagreb, Croatia*

ARTICLE INFO

*Article history:*
Received 7 January 2013
Available online 29 May 2013

Communicated by G. Borgefors

*Keywords:*
Dimensionality reduction
Tensor decomposition
Feature extraction
Mutual information

ABSTRACT

Several supervised feature extraction methods for tensor objects have been proposed recently, with applications in recognition of objects, faces and handwritten digits. However, the existing methods usually use only second order statistics of the data, typically through calculation of the within- and between-class scatters. Here we propose a method for supervised feature extraction for tensor objects based on maximization of an approximation of mutual information. In this way we utilize information contained in the higher order statistics of the data. Several experiments show that the proposed method results in highly discriminative features.



## 1. Introduction

Methods for dimensionality reduction are very important in the field of machine learning and pattern recognition. Preprocessing high-dimensional data by a suitable dimensionality reduction procedure leads to a reduced computational cost in further processing, and often better generalization and easier interpretation of a smaller number of features. Dimensionality reduction can be performed by building a small set of new features through linear or nonlinear transformation of the original data, i.e., by performing a linear or nonlinear feature extraction (FE). Classical algorithm for unsupervised linear FE is principal component analysis (PCA), while typical supervised linear FE method is linear discriminant analysis (LDA). In both supervised and unsupervised scenario, the aim is to find a low-dimensional subspace that keeps all information about the original data, usually through optimization of some criterion. However, the difference is that the supervised methods use additional information provided in labels of the training samples, while unsupervised methods do not use such information. In classification problems we are interested in features that provide as much information as possible for class discrimination. Therefore, some kind of proxy for class discrimination should be used as an optimization criterion when seeking for optimal transformation of the features. Classical techniques based on discriminant analysis, such as LDA and its variants, are based on maximization of distance between classes and rely only on the first and second order statistics of the data. However, recent methods address the linear FE problem using information theoretic criteria and achieve superior results (Torkkola, 2003; Leiva-Murillo and Artès-Rodrìgues, 2007; Kamandar and Ghassemian, 2013).

In modern applications, such as neuroscience, chemometrics, text mining, image and video analysis, data is often represented by multi-way arrays, i.e., tensors (Cichocki et al., 2009). However, most of the methods for FE treat input samples as vectors, and thus ignore their natural multi-way structure. Additionally, vectorization of tensors results in vector samples with extremely high-dimensions, leading to computational problems and curse of dimensionality. Recently, several algorithms have been proposed for discriminative analysis of tensor objects (Yan et al., 2005; Tao et al., 2007; Zhang et al., 2009; Nie et al., 2009; Phan and Cichocki, 2010). They are mainly some kind of generalization of LDA to multi-way data. However, as noted in (Petridis and Perantonis, 2004), mutual information (MI) between features and labels is a more general criterion for evaluating the discriminative power of features. As opposed to LDA-based techniques, MI-based methods use information contained in the data beyond second order statistics. In this paper we propose a method for supervised FE from multi-way data based on maximization of an approximation of mutual information between the extracted features and class labels. By optimizing a suitable cost function we aim to obtain

* Corresponding author. Current address: Signal Processing Group, University of Oldenburg, 26111 Oldenburg, Germany. Tel.: +49 441 798 3377; fax: +49 441 798 3902.

*E-mail addresses:* ante.jukic@uni-oldenburg.de (A. Jukić), filipov@irb.hr (M. Filipović).

[1] This work was done while the author was with the Ruđer Bošković Institute, Zagreb, Croatia.

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.05.018

more discriminative features and achieve superior accuracy in classification problems.

The rest of the paper is organized as follows. In Section 2 we give basic preliminaries of tensor algebra, linear FE and some information about previous work in tensor discriminant analysis. In Section 3 we propose a method for supervised feature extraction for tensor objects. The experimental results are given in Section 4, and Section 5 contains concluding remarks.

## 2. Preliminaries and previous work

In this section we give definitions of basic operations with tensors and present information-theoretic linear FE method based on mutual information. Also, we briefly comment previous work on feature extraction for tensor objects. In the following scalars will be denoted by italic letters (e.g., $x$), vectors by bold lowercase letters (e.g., $\mathbf{x}$), matrices by bold capital letters (e.g., $\mathbf{X}$) and tensors by bold capital calligraphic letters (e.g., $\mathcal{X}$). Tensor notation in this paper mostly follows conventions presented in Kolda and Bader (2009) and Cichocki et al. (2009).

### 2.1. Basics of tensor notation, operations and decompositions

Tensor is a multi-way generalization of vector and matrix, and order of tensor is equal to the number of its indices. For example, tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is a tensor of order $N$ (i.e., an $N$-way tensor) with elements $x_{i_1 i_2 \ldots i_N}$. Vector $\mathbf{x}_{i_1,\ldots,i_{n-1},i_{n+1},\ldots,i_N} = \mathcal{X}(i_1,\ldots,i_{n-1},:,i_{n+1},\ldots,i_N)$ obtained by fixing all indices except $i_n$ is called mode-$n$ fiber. Often it is convenient to present tensor in matrix form, so we define mode-$n$ matricization of tensor $\mathcal{X}$ as a matrix $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times \prod_{k=1,k\neq n}^{N} I_k}$ that contains mode-$n$ fibers as columns. Ordering of columns in matricization is not important, as long as it is consistent in all computations. Inner product of tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with the same order and dimensions is defined as $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1,\ldots,i_N} x_{i_1,\ldots,i_N} y_{i_1,\ldots,i_N}$, and it induces a tensor norm $\|\mathcal{X}\| = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$. Mode-$n$ product of tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{n-1} \times R_n \times I_{n+1} \times \ldots \times I_N}$ and matrix $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ results in a new tensor $\mathcal{X} = \mathcal{Y} \times_n \mathbf{A}^{(n)} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{n-1} \times I_n \times I_{n+1} \times \ldots \times I_N}$. It is defined (in equivalent tensor, matricized and vectorized forms) as follows

$$\begin{aligned} &\mathcal{X} = \mathcal{Y} \times_n \mathbf{A}^{(n)} \\ &\mathbf{X}_{(n)} = \mathbf{A}^{(n)} \mathbf{Y}_{(n)} \\ &vec(\mathbf{X}_{(n)}) = \left(\mathbf{I} \otimes \mathbf{A}^{(n)}\right) vec(\mathbf{Y}_{(n)}), \end{aligned} \tag{1}$$

where $vec(\cdot)$ is column-wise vectorization of a matrix, $\mathbf{I}$ denotes an identity matrix of appropriate size, and $\otimes$ is the Kronecker product (Cichocki et al., 2009; Kolda and Bader, 2009). This multiplication is commutative when applied in distinct modes ($m \neq n$)

$$(\mathcal{X} \times_n \mathbf{A}) \times_m \mathbf{B} = (\mathcal{X} \times_m \mathbf{B}) \times_n \mathbf{A} = \mathcal{X} \times_n \mathbf{A} \times_m \mathbf{B}, \tag{2}$$

where matrices $\mathbf{A}$ and $\mathbf{B}$ have appropriate dimensions. Multiplication in all possible modes with a set of matrices $\mathbf{A}^{(n)}$, $n \in \{1,\ldots,N\}$ is denoted as

$$\mathcal{X} \times \{\mathbf{A}\} = \mathcal{X} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)}, \ldots, \times_N \mathbf{A}^{(N)}. \tag{3}$$

Multiplication in all modes except mode $n$ is denoted as

$$\mathcal{X} \times_{-n} \{\mathbf{A}\} = \mathcal{X} \times_1 \mathbf{A}^{(1)}, \ldots, \times_{n-1} \mathbf{A}^{(n-1)} \times_{n+1} \mathbf{A}^{(n+1)}, \ldots, \times_N \mathbf{A}^{(N)}. \tag{4}$$

Basic decomposition of tensor $\mathcal{X}$ is the Tucker decomposition that can be expressed as

$$\mathcal{X} \approx \mathcal{F} \times \{\mathbf{A}\}, \tag{5}$$

where $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$, $n \in \{1,\ldots,N\}$ are factor matrices, and $\mathcal{F} \in \mathbb{R}^{R_1 \times R_2 \times \ldots \times R_N}$ is the core tensor (Tucker, 1964; Tucker, 1966). In this paper we are interested in Tucker decomposition with $R_n \leqslant I_n$. If there exists such $m \in \{1,\ldots,N\}$ that $R_m < I_m$ then the core tensor $\mathcal{F}$ has smaller dimension in mode $m$ than the original tensor $\mathcal{X}$. This can be seen as a form of compression or dimensionality reduction along the mode $m$, resulting in smaller number of elements in the core tensor, than in the original tensor.

### 2.2. Maximization of mutual information for linear feature extraction

Let $\{\mathbf{x}_k, y_k\}$, $k \in \{1,\ldots,K\}$ be a set of $K$ available samples $\mathbf{x}_k \in \mathbb{R}^I$, paired with their class labels $y_k \in \{1,\ldots,C\}$ where $C$ is the number of classes. The task of linear FE is to find a matrix $\mathbf{W} \in \mathbb{R}^{I \times R}$ such that features

$$\mathbf{f}_k = \mathbf{W}^T \mathbf{x}_k \tag{6}$$

are as discriminative as possible, with typically $R \ll I$. The aim of LDA is to find a subspace in which the class means are well separated while at the same time within class scatters are small, and it is known to be optimal in the case of homoscedastic Gaussian model, when all classes have Gaussian distribution with the same covariance. However, this model is often too restrictive for modeling real-world data (Petridis and Perantonis, 2004). Another line of reasoning has led to the development of FE methods based on information theory (Torkkola, 2003; Leiva-Murillo and Artès-Rodrìgues, 2007; Kamandar and Ghassemian, 2013). It was shown that maximization of mutual information (MMI) is optimal criterion under the zero information loss (ZIL) model (Petridis and Perantonis, 2004). The ZIL model assumes that the observation space can be divided into $R$-dimensional signal subspace and $(I-R)$-dimensional noise subspace, with signal subspace containing all information about the original observations. This is a common assumption in source separation and it is more general then homoscedasticity or heteroscedasticity of class-conditional distributions. It was demonstrated that MMI-based methods achieve state-of-the-art results on several problems (Torkkola, 2003; Leiva-Murillo and Artès-Rodrìgues, 2007).

Let $\mathbf{x}$ denote a random vector with values in $\mathbb{R}^I$ that comes from the same statistical model as the available samples, paired with corresponding label $y$ that is a discrete random variable with values in $\{1,\ldots,C\}$. Then mutual information between $\mathbf{x}$ and $y$ is given as

$$I(\mathbf{x}, y) = h(\mathbf{x}) - \sum_{k=1}^{C} \mathbb{P}(y=k) h(\mathbf{x}|y=k), \tag{7}$$

where $h$ is Shannon's differential entropy, and $\mathbb{P}(y=k)$ *a priori* probability of class $k$. The definitions of entropies are given as follows

$$\begin{aligned} h(\mathbf{x}) &= -\int p_{\mathbf{x}}(\mathbf{t}) \log p_{\mathbf{x}}(\mathbf{t}) d\mathbf{t}, \text{ and } h(\mathbf{x}|y=k) \\ &= -\int p_{\mathbf{x}|y}(\mathbf{t}|k) \log p_{\mathbf{x}|y}(\mathbf{t}|k) d\mathbf{t}, \end{aligned} \tag{8}$$

with $p$ denoting appropriate probability density function (Cover and Thomas, 1991). It was noted in (Leiva-Murillo and Artès-Rodrìgues, 2007; Petridis and Perantonis, 2004) that if the ZIL model holds, then there is always an orthonormal matrix $\mathbf{W}^{I \times R}$ that satisfies $I(\mathbf{f}, y) = I(\mathbf{W}^T\mathbf{x}, y) = I(\mathbf{x}, y)$, i.e., there is no loss of information after dimensionality reduction. In real-world scenario ZIL model can only be approximately satisfied and the dimension of the signal subspace cannot be exactly determined. Since the MI cannot be increased by

any deterministic transformation (Cover and Thomas, 1991), a realistic objective is to find a suboptimal transformation that achieves maximal MI between obtained features and labels. The projection matrix for linear FE can be defined as a solution of the following optimization problem

$$\mathbf{W}^* = \arg\max_{\mathbf{W}\in\mathbb{R}^{I\times R}} I(\mathbf{f},y) = \arg\max_{\mathbf{W}\in\mathbb{R}^{I\times R}} I(\mathbf{W}^T\mathbf{x},y). \tag{9}$$

However, estimation of mutual information for $R$-dimensional random vector is not easy, since it involves estimation of entropy for a random vector. To alleviate this problem, an approximation $\tilde{I}$ that uses only MI between scalar random variables was proposed in Leiva-Murillo and Artès-Rodrìgues (2007) as

$$\tilde{I}(\mathbf{f},y) = \sum_{r=1}^{R} I(f_r,y) = \sum_{r=1}^{R} I(\mathbf{w}_r^T\mathbf{x},y), \tag{10}$$

where $\mathbf{w}_r$ denotes $r$-th column of matrix $\mathbf{W}$. Since there are several good estimators of entropy for scalar random variables this approximation can be computed efficiently, e.g., through approximation of negentropy as given in Appendix A (Hyvärinen et al., 2001). Mutual information $I$ and its approximation $\tilde{I}$ are connected through relation

$$\tilde{I}(\mathbf{f},y) = I(\mathbf{f},y) + [I(\mathbf{f}) - I(\mathbf{f}|y)]. \tag{11}$$

It was also demonstrated in Leiva-Murillo and Artès-Rodrìgues (2007) on several real-world datasets that the difference between the mutual information and its approximation is rather small, and use of approximation (10) is justified. Problem of linear FE can then be formulated as

$$\mathbf{W}^* = \arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \tilde{I}(\mathbf{f},y) = \arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \sum_{r=1}^{R} I(\mathbf{w}_r^T\mathbf{x},y). \tag{12}$$

In order to maximize (12), gradient of the cost function with respect to $\mathbf{W}$ has to be calculated as $\nabla_{\mathbf{W}}\tilde{I}(\mathbf{f},y) = \left[\nabla_{\mathbf{w}_1} I(\mathbf{w}_1^T\mathbf{x},y),\ldots,\nabla_{\mathbf{w}_R} I(\mathbf{w}_R^T\mathbf{x},y)\right] \in \mathbb{R}^{I\times R}$, with each column of $\nabla_{\mathbf{W}}\tilde{I}(\mathbf{f},y)$ calculated using (A.3) in Appendix A.

### 2.3. Feature extraction for tensor objects

Let $\{\mathcal{X}_k, y_k\}$, $k\in\{1,\ldots,K\}$ be a set of $K$ available samples (i.e., a training set), represented by tensors $\mathcal{X}_k \in \mathbb{R}^{I_1\times\ldots\times I_N}$ paired with their class labels $y_k\in\{1,\ldots,C\}$ where $C$ is the number of classes. The goal of FE is to construct a relatively small set of $D$ discriminative features, with typically $D \ll \prod_{n=1}^{N} I_n$. Usually, FE for tensor objects is performed by representing each sample by its Tucker decomposition

$$\mathcal{X}_k \approx \mathcal{F}_k \times \{\mathbf{A}\}, k\in\{1,\ldots,K\}, \tag{13}$$

where factor matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n\times R_n}$ are common for all samples. This can be seen as a joint Tucker decomposition of $K$ tensors, i.e., a Tucker-$N$ decomposition of $(N+1)$-order tensor as in Phan and Cichocki (2010). The elements of the core tensor $\mathcal{F}_k$ are interpreted as features for the $k$-th sample. Decomposition (13) can be performed in supervised or unsupervised manner. For example, methods such as higher-order singular value decomposition (HOSVD) or higher order orthogonal iteration (HOOI) can be used to obtain decompositions in form (13) with orthogonal projection matrices, with objective being minimal norm between sample $\mathcal{X}_k$ and its Tucker decomposition (Phan and Cichocki, 2010). Also, constraints such as nonnegativity can be imposed on the core and factor matrices, depending on the nature of the data contained in the samples, to improve interpretability of the decomposition (Cong et al., 2012; Phan and Cichocki, 2011). These unsupervised approaches proved to be useful in various problems, such as image and EEG analysis. However, in classification scenario it is useful to use information contained in the class labels. This naturally leads to development of methods for tensor discriminant analysis, such as discriminant analysis with tensor representation (DATER) (Yan et al., 2005), general tensor discriminant analysis (Tao et al., 2007), tensor linear Laplacian discrimination (TLLD) (Zhang et al., 2009), local tensor discriminant analysis (LTDA) (Nie et al., 2009) and higher-order discriminant analysis (HODA) (Phan and Cichocki, 2010). Mentioned methods are generalizations of linear discriminant analysis to the tensor representation. Extracted features are obtained by projecting samples using orthogonal projection matrices $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n\times R_n}$ so that feature tensors $\mathcal{F}_k = \mathcal{X}_k \times \{\mathbf{U}\}^T$ have maximal between-class scatter and minimal within class scatter. Projection matrices are sought in alternating manner, by fixing all projection matrices except one, to obtain a locally optimal solution. In each step a single projection matrix is obtained by maximizing a trace ratio or trace difference problem, similar as in LDA. Additionally, local scatters are defined in LTDA to overcome the assumption of normally distributed samples in each class, inherited from LDA. This method is especially interesting since it includes an automatic method for selecting dimensions $R_n$ of the core tensor (Nie et al., 2007). However, the drawback of the above mentioned methods is that they use only information contained in the between- and within-scatter matrices, neglecting information beyond second order moments of the data.

## 3. Maximization of mutual information for tensor decomposition

In order to extract features from tensor objects that are more discriminative, we adopt MMI-based criteria for finding projection matrices.

### 3.1. Proposed method

Without loss of generality we can consider three-way samples. Let $\mathcal{X}$ denote a three-way random tensor with values in $\mathbb{R}^{I_1\times I_2\times I_3}$ and $y$ corresponding label that is a discrete random variable with values in $\{1,\ldots,C\}$. Then $\mathcal{X}$ can be represented through Tucker decomposition with orthogonal projection matrices and the core

$$\mathcal{F} = \mathcal{X}\times_1\mathbf{U}^{(1)T}\times_2\mathbf{U}^{(2)T}\times_3\mathbf{U}^{(3)T}. \tag{14}$$

The elements of the core tensor can be seen as features that can be used for classification. We aim to find projection matrices such that the features in the core tensor $\mathcal{F}$ are as discriminative as possible. Since most of the existing classifiers work with vectors[2], the core tensor is vectorized to obtain features $\mathbf{f} = vec(\mathbf{F}_{(1)})$, with values in $\mathbb{R}^{R_1R_2R_3}$. Then, the features can be expressed as

$$\mathbf{f} = \left(\mathbf{U}^{(3)T}\otimes\mathbf{U}^{(2)T}\otimes\mathbf{U}^{(1)T}\right) vec(\mathbf{X}_{(1)\cdot}). \tag{15}$$

This can be seen as a linear FE in form (6), with transformation matrix $\mathbf{W} = \mathbf{U}^{(3)}\otimes\mathbf{U}^{(2)}\otimes\mathbf{U}^{(1)} \in \mathbb{R}^{I_1I_2I_3\times R_1R_2R_3}$ that has structure induced by the Kronecker product. Finding all optimal projection matrices at once is not an easy task due to the size of the matrix $\mathbf{W}$ and involved cross-products of elements of the projection matrices. However, if we use alternating approach we can obtain a single projection matrix at a time, in similar fashion as with other tensor decomposition algorithms.

Let us assume that all projection matrices except $\mathbf{U}^{(1)}$ are fixed. Then the core can be expressed as

$$\mathcal{F} = \mathcal{Z}^{-1}\times_1\mathbf{U}^{(1)T}, \tag{16}$$

[2] Recent work on classifiers for tensor objects can be found in Signoretto et al. (2011).

with $\mathcal{Z}^{-n} = \mathcal{X}\times_{-n}\{\mathbf{U}\}^T$. This can be expressed in vectorized form as

$$vec(\mathbf{F}_{(1)}) = \left(\mathbf{I}\otimes\mathbf{U}^{(1)T}\right) vec\left(\mathbf{Z}_{(1)}^{-1}\right), \tag{17}$$

where $\mathbf{I}$ is the identity matrix with dimensions $R_2R_3 \times R_2R_3$. Let $\mathbf{f}_{r_2r_3}$ denote a mode-1 fiber of $\mathcal{F}$ defined as $\mathbf{f}_{r_2r_3} = \mathcal{F}(:, r_2, r_3)$. We aim to obtain projection matrix $\mathbf{U}^{(1)}$ that results in maximal mutual information between the extracted features $\mathbf{f} = vec(\mathbf{F}_{(1)})$ and the class label $y$. This can be performed through maximization of approximation (10) of MI, yielding

$$\tilde{I}_1(\mathbf{f}, y) = \sum_{r_1=1}^{R_1}\sum_{r_2=1}^{R_2}\sum_{r_3=1}^{R_3} I(f_{r_1r_2r_3}, y) = \sum_{r_2=1}^{R_2}\sum_{r_3=1}^{R_3}\tilde{I}(\mathbf{f}_{r_2r_3}, y), \tag{18}$$

where $\tilde{I}(\mathbf{f}_{r_2r_3}, y)$ is given as

$$\tilde{I}_1(\mathbf{f}_{r_2r_3}, y) = \tilde{I}\left(\mathbf{U}^{(1)T}\mathbf{z}_{r_2r_3}, y\right) = \sum_{r_1=1}^{R_1} I\left(\mathbf{u}_{r_1}^{(1)T}\mathbf{z}_{r_2r_3}, y\right). \tag{19}$$

Then the cost function for estimating projection matrix $\mathbf{U}^{(1)}$ can be expressed in the following form

$$\tilde{I}_1(\mathbf{f}, y) = \sum_{r_1=1}^{R_1}\sum_{r_2=1}^{R_2}\sum_{r_3=1}^{R_3} I\left(\mathbf{u}_{r_1}^{(1)T}\mathbf{z}_{r_2r_3}, y\right). \tag{20}$$

Notice that the expression for $\tilde{I}_1$ involves calculation of mutual information between scalar random variables that can be computed efficiently. Mode-1 projection matrix $\mathbf{U}^{(1)}$ can then be found by solving the following optimization problem

$$\mathbf{U}^{(1)} = \underset{\mathbf{U}^{(1)T}\mathbf{U}^{(1)}=\mathbf{I}}{\arg\max}\,\tilde{I}_1(\mathbf{f}, y). \tag{21}$$

In order to employ the optimization procedure to find the mode-1 projection matrix we need to calculate the gradient with respect to $\mathbf{U}^{(1)}$, i.e., $\nabla_{\mathbf{U}^{(1)}}\tilde{I}_1(\mathbf{f}, y) = \left[\nabla_{\mathbf{u}_1^{(1)}}\tilde{I}_1(\mathbf{f}, y), \ldots, \nabla_{\mathbf{u}_{R_1}^{(1)}}\tilde{I}_1(\mathbf{f}, y)\right]$. Each column of the gradient matrix is given as

$$\nabla_{\mathbf{u}_{r_1}^{(1)}}\tilde{I}_1(\mathbf{f}, y) = \sum_{r_2=1}^{R_2}\sum_{r_3=1}^{R_3}\nabla_{\mathbf{u}_{r_1}^{(1)}} I\left(\mathbf{u}_{r_1}^{(1)T}\mathbf{z}_{r_2r_3}, y\right), \tag{22}$$

for $r_1 \in \{1, \ldots, R_1\}$. For calculation of (20) and (22) we use approximations of mutual information and its gradient for scalar variables given with (A.2) and (A.3) in Appendix A.

In case of a $N$-way tensor, cost function for estimation of the mode-$n$ projection matrix $\mathbf{U}^{(n)}$ is calculated as

$$\tilde{I}_n(\mathbf{f}, y) = \sum_{r_1,\ldots,r_N} I\left(\mathbf{u}_{r_n}^{(n)T}\quad \mathbf{z}_{r_1,\ldots,r_{n-1},r_{n+1},\ldots,r_N}, y\right), \tag{23}$$

and components of its gradient $\nabla_{\mathbf{U}^{(n)}}\tilde{I}_n(\mathbf{f}, y) = \left[\nabla_{\mathbf{u}_1^{(n)}}\tilde{I}_n(\mathbf{f}, y), \ldots, \nabla_{\mathbf{u}_{R_n}^{(n)}}\tilde{I}_n(\mathbf{f}, y)\right]$ as

$$\nabla_{\mathbf{u}_{r_n}^{(n)}}\tilde{I}_n(\mathbf{f}, y) = \sum_{r_1,\ldots,r_{n-1},r_{n+1},\ldots,r_N} \nabla_{\mathbf{u}_{r_n}^{(n)}} I\left(\mathbf{u}_{r_n}^{(n)T}\mathbf{z}_{r_1,\ldots,r_{n-1},r_{n+1},\ldots,r_N}, y\right). \tag{24}$$

The mode-$n$ projection matrix is found by solving the following optimization problem

$$\mathbf{U}^{(n)} = \underset{\mathbf{U}^{(n)T}\mathbf{U}^{(n)}=\mathbf{I}}{\arg\max}\,\tilde{I}_n(\mathbf{f}, y). \tag{25}$$

In this way an iterative alternating procedure can be used to estimate projection matrices $\mathbf{U}^{(n)}$. Pseudocode of the proposed approach for $N$-way tensor is given in Table 1.

### 3.2. Optimization and initialization

Some optimization method is needed to find a single projection matrix in each step of the alternating procedure by solving (25). In Leiva-Murillo and Artès-Rodrìgues (2007) a single column of the projection matrix was obtained at a time using the gradient ascent method. Orthogonality constraints were enforced after each step by performing Gramm–Schmidt (GS) orthogonalization of the current column with respect to previous columns. In this approach, search space for a column of projection matrix is orthogonal to the subspace spanned by previously obtained projections. We tested this approach but it resulted in inferior performance compared to the results reported in the experimental section of this paper.

Here we propose to obtain the whole projection matrix $\mathbf{U}^{(n)}$ at once by maximizing (25) over the Stiefel manifold $\mathcal{M}_{I_n}^{R_n} := \left\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n\times R_n} : \mathbf{U}^{(n)T}\mathbf{U}^{(n)} = \mathbf{I}\right\}$. A feasible method for optimization with orthogonality constraints was proposed recently in Wen and Yin (2012a). The proposed approach is a gradient-based method accelerated with a curvilinear search over the Stiefel manifold. Given a feasible initial point $\mathbf{U}^{(n)}$ and the gradient $\nabla_{\mathbf{U}^{(n)}}\tilde{I}_n(\mathbf{f}, y)$, a skew-symmetric matrix $\mathbf{A}$ is calculated as

$$\mathbf{A} := \mathbf{G}\mathbf{U}^{(n)T} - \mathbf{U}^{(n)}\mathbf{G}^T, \tag{26}$$

with $\mathbf{G} := -\nabla_{\mathbf{U}^{(n)}}\tilde{I}_n(\mathbf{f}, y)$. Then a simple closed form update rule can be used to find a new point as

$$\mathbf{U}^{(n)} \leftarrow \mathbf{Q}(\tau)\mathbf{U}^{(n)},\ \text{with}\ \mathbf{Q}(\tau) := \left(\mathbf{I} + \frac{\tau}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} - \frac{\tau}{2}\mathbf{A}\right), \tag{27}$$

with $\tau \in \mathbb{R}$ denoting the step size. This update has several favorable properties. It is easy to verify that the new point is also feasible w.r.t. orthogonality constraints, i.e., the condition $(\mathbf{U}^{(n)})^T(\mathbf{U}^{(n)}) = \mathbf{I}$ holds after the update, for all $\tau = \mathbb{R}$. For $\tau \geqslant 0$ update (27) defines an ascent path for $\tilde{I}_n(\mathbf{f}, y)$, so a curvilinear search can be utilized for selecting the appropriate step size $\tau$ and to ensure convergence to a stationary point (Wen and Yin, 2012a). Outline of this optimization procedure is given in Table 2. In our experiments we used nonmonotone curvilinear search with Barzilai–Borwein step size. For more details regarding the step size selection strategies see Wen and Yin (2012a,b) and references therein.

The alternating procedure is repeated until some convergence condition is satisfied. Our stopping criterion was the change of the cost function $\tilde{I}(\mathbf{f}, y)$ in subsequent iterations of the outer loop in Table 1. There is no guarantee that the alternating procedure will lead to the globally optimal solution, only to a point where the cost function ceases do decrease, similar as in other algorithms based on alternating optimization (Cichocki et al., 2009; Zhang et al., 2009; Nie et al., 2009; Phan and Cichocki, 2010). Therefore an appropriate initialization is important, since the iterative procedure can be interpreted as enhancement of the initial projection matrices by maximizing discriminative power of features. By wisely selecting initial point the described procedure can lead to a much better solution than initially given. A reasonable choice would be to use some of the existing tensor decomposition algorithms for initialization. In the experimental section we tested initialization with HOSVD and LTDA. An alternative would be to initialize projection matrices randomly, but this would require a multi-start strategy to identify the best initialization (Cichocki et al., 2009).

## 4. Experimental results

In order to assess the performance of our approach we performed several experiments on the standard datasets with images of objects, face images and handwritten digits. We compared the proposed method for feature extraction with HOSVD, HOOI (implemented in Tensor toolbox (Bader et al., 2012)), HODA and TLLD (both implemented in NFEA toolbox (Phan, 2012)) and LTDA[3].

[3] Comparison of tensor-based FE and linear FE methods can be found in Nie et al. (2009), Zhang et al. (2009) and Hou et al. (2013).

**Table 1**
Outline of the proposed method.

**Input**: Set of $K$ training samples with labels, $\{\mathcal{X}_k \in \mathbb{R}^{I_1 \times \ldots \times I_N}, y_k\}$, $k \in \{1,\ldots,K\}$.
**Parameters**: Number of features for each mode $(R_1,\ldots,R_N)$.
Initialize $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$, $n \in \{1,\ldots,N\}$.
**Repeat**
  % alternating procedure
  **For** $n = 1$ to $N$
    % assume all matrices except in mode-$n$ are fixed
    $\mathcal{Z}_k^n = \mathcal{X}_k \times_{-n} \{\mathbf{U}\}^T$
    % find the mode-$n$ matrix using the optimization procedure in Table 2
    $\mathbf{U}^{(n)} = \underset{\mathbf{U}^{(n)T}\mathbf{U}^{(n)}=\mathbf{I}}{\arg\max}\, \tilde{I}_n(\mathbf{f}, y)$
  **End**
**Until** (convergence)
**Output**: Projection matrices $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$, $n \in \{1,\ldots,N\}$.

**Table 2**
Optimization procedure for (25), (Wen and Yin, 2012a).

**Input**: Feasible initial projection matrix $\mathbf{U}^n$.
$k \leftarrow 0$
**Repeat**
  Calculate gradient $\nabla_{\mathbf{U}^{(n)}} \tilde{I}_n \quad (\mathbf{f}, y)$, according to (24).
  Calculate $\mathbf{A}$, according to (26).
  Select the step size $\tau_k$ using curvilinear search.
  Update $\mathbf{U}^{(n)} \leftarrow \mathbf{Q}(\tau_k)\mathbf{U}^n$, according to (27).
  $k \leftarrow k + 1$
**Until** $\left(\left\|\nabla_{\mathbf{U}^{(n)}} \tilde{I}_n(\mathbf{f}, y)\right\|_F \leq tolerance\right)$
**Output**: New projection matrix $\mathbf{U}^{(n)}$.

The proposed method is labeled as mutual information-based tensor decomposition (MITD). In all experiments we used $k$-nearest neighbor classifier with $k = 3$ neighbors and Euclidean distance (kNN3), and linear support vector machine (linSVM) implemented in LIBSVM library (C-SVM with parameter $C = 1$) (Chang and Lin, 2011). Parameters of the classifiers were fixed, and accuracy was estimated through 50 random partitions of the data set. Note that our goal was not to achieve maximal accuracy, but to show that the proposed feature extraction method can be used to improve classification performance. Therefore we used simple and well-known classifiers. Some samples from the databases used in the experiments are shown in Fig. 1.

**Fig. 1.** Images used in the experiments. From top to bottom: COIL-20, COIL-100, SFD, and MNIST.

Implementation details of the proposed algorithm were as follows. The maximum number of iterations was set to 50, and the tolerance on change of the cost function was set to $10^{-5}$. However, in almost all experiments a small number of iterations of the outer loop were performed (typically two). Projection matrices were initialized using HOSVD or LTDA (in tables initialization is denoted in brackets). Optimization problem (25) was solved using gradient ascent over Stiefel manifold and nonmonotone curvilinear search, with code available at Wen and Yin (2012b). The maximum number of iterations for the solver was set to 100, parameter $\tau$ for line search was set to $10^{-3}$ and all tolerances to $10^{-5}$. For NFEA toolbox we set the maximum number of iterations to 50, tolerance to $10^{-8}$, initialization to *eig*, and we used the *traceratio* method without regularization to extract fully discriminative projection matrices (Phan, 2012). For LTDA we set $k_w = 3$ and $k_b = 20$ with maximum of 20 iterations and tolerance $10^{-5}$, while we used HOOI with tolerance $10^{-8}$ and maximally 1000 iterations. All experiments were performed in MATLAB 2010b environment.

### 4.1. Object recognition

The Columbia University Image Library (COIL-20) dataset consists of grayscale images of 20 objects. Each object is represented by 72 grayscale images obtained by rotating the object with step of 5°. We downsampled each image to $32 \times 32$ pixels, and four, six or eight samples per class were randomly selected for training set with remaining samples forming the test set. No preprocessing was done, i.e., raw images were used as input to the feature extraction procedure. The number of components in each mode was set to $(R_1,R_2) \in \{(5,5),(10,10)\}$ and no feature selection was performed on the extracted features. Hence, we use all of the 25 or 100 features per sample. Classification accuracy was estimated over 50 random partitions and results are presented in Table 3. We also performed a series of experiments with the automatically selected number of components using the procedure described in Nie et al. (2009). The LTDA with automatically selected dimensions was compared with the proposed method initialized using LTDA. In almost all experiments MITD initialized with LTDA resulted in best classification accuracy. While other methods achieve lower accuracy when the number of features is increased, the proposed approach results in even better class discrimination.

We also performed comparative analysis on object recognition from color images, using the COIL-100 dataset that contains color images of 100 objects. Each object is represented by 72 color (RGB) images, obtained through the same rotation procedure as previously described. Each image was downsampled to $32 \times 32$ pixels and we performed the same experiments as with COIL-20 data: with fixed number of components $(R_1,R_2,R_3) \in \{(5,5,3),(10,10,3)\}$, and with dimensions automatically selected using LTDA. Results are shown in Table 4. It can be seen that the proposed approach significantly outperforms competing methods. While the MITD initialized with HOSVD is better than other methods, even better results are obtained when MITD is initialized with LTDA. Again, only the proposed approach benefits from the increased number of features, while the accuracy for other methods is significantly lower. Note that in experiments on COIL-100 there is no dimensionality reduction along the mode-3, since for RGB images $I_3 = 3$ and in experiments dimension in mode-3 was set to $R_3 = 3$ (both automatically and manually). However, as can be seen from the Eq. (15), it is important to perform decomposition along the mode-3 since the projection matrix $\mathbf{U}^{(3)}$ affects the subspace that contains the extracted features.

**Table 3**
Object recognition (COIL-20). Accuracy was estimated using 50 random partitions. HO denotes hold-out ratio. Results reported in % as accuracy (standard deviation).

| Method | 8 train (HO90%) | | 6 train (HO93%) | | 4 train (HO95%) | |
|---|---|---|---|---|---|---|
| | kNN3 | linSVM | kNN3 | linSVM | kNN3 | linSVM |
| Projection matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ with 5 components, $(R_1,R_2)=(5,5)$ | | | | | | |
| HOSVD | 81.84 (1.42) | 89.49 (1.49) | 77.92 (1.85) | 84.89 (1.97) | 72.41 (2.01) | 79.49 (1.78) |
| HOOI | 81.92 (1.35) | 89.48 (1.47) | 78.02 (1.91) | 85.03 (1.97) | 72.31 (1.97) | 79.48 (1.83) |
| HODA | 75.67 (2.46) | 84.44 (2.62) | 70.70 (2.62) | 79.84 (2.77) | 63.91 (3.11) | 73.27 (3.02) |
| TLLD | 75.88 (2.36) | 83.43 (2.15) | 70.45 (3.11) | 78.66 (2.98) | 62.63 (3.27) | 72.44 (2.85) |
| LTDA | 86.38 (2.37) | 91.88 (1.54) | 83.01 (1.70) | 88.98 (2.23) | **76.52 (2.90)** | **82.44 (2.77)** |
| MITD (HOSVD) | 81.52 (1.71) | 89.53 (1.61) | 77.33 (2.37) | 84.67 (2.37) | 71.82 (2.10) | 79.13 (2.12) |
| MITD (LTDA) | **86.66 (2.35)** | **92.01 (1.60)** | **83.04 (1.74)** | **89.06 (2.32)** | 76.30 (4.11) | 82.13 (3.81) |
| Projection matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ with 10 components, $(R_1,R_2)=(10,10)$ | | | | | | |
| HOSVD | 69.89 (1.94) | 82.78 (1.58) | 64.51 (2.33) | 77.52 (2.01) | 58.41 (2.90) | 71.24 (2.20) |
| HOOI | 69.83 (2.05) | 82.75 (1.57) | 64.42 (2.32) | 77.45 (2.03) | 58.22 (2.92) | 71.05 (2.25) |
| HODA | 61.21 (2.76) | 78.79 (1.97) | 55.90 (2.54) | 73.77 (2.09) | 49.79 (2.99) | 67.72 (2.42) |
| TLLD | 60.99 (2.84) | 77.98 (2.00) | 55.17 (2.95) | 72.75 (2.00) | 48.62 (3.75) | 66.34 (2.27) |
| LTDA | 75.18 (2.07) | 85.69 (1.89) | 71.13 (2.69) | 81.08 (2.41) | 65.43 (2.75) | 73.66 (2.48) |
| MITD (HOSVD) | 70.45 (1.73) | 82.93 (1.66) | 65.27 (2.15) | 77.78 (2.18) | 58.96 (2.91) | 71.53 (2.10) |
| MITD (LTDA) | **87.87 (1.86)** | **94.74 (1.26)** | **84.95 (2.88)** | **91.62 (2.01)** | **78.48 (6.45)** | **84.89 (5.63)** |

Automatically selected number of components

| Number of components | 8 train (HO90%) | | 6 train (HO93%) | | 4 train (HO95%) | |
|---|---|---|---|---|---|---|
| | $R_1$ | $R_2$ | $R_1$ | $R_2$ | $R_1$ | $R_2$ |
| Min | 12 | 4 | 12 | 5 | 13 | 6 |
| Max | 16 | 6 | 16 | 8 | 17 | 9 |
| Average | 13.5 | 5.46 | 14.04 | 6.28 | 14.78 | 7.64 |
| Method | kNN | linSVM | kNN3 | linSVM | kNN3 | linSVM |
| LTDA | 85.20 (2.13) | 91.90 (1.55) | 79.53 (3.34) | 87.20 (2.73) | 66.68 (3.28) | 77.04 (2.85) |
| MITD (LTDA) | **87.16 (2.02)** | **93.21 (1.50)** | **82.12 (3.54)** | **88.66 (3.03)** | **73.35 (6.16)** | **80.63 (5.13)** |

**Table 4**
Object recognition (COIL-100). Accuracy was estimated using 50 random partitions. HO denotes hold-out ratio. Results reported in % as accuracy (standard deviation).

| Method | 8 train (HO90%) | | 6 train (HO93%) | | 4 train (HO95%) | |
|---|---|---|---|---|---|---|
| | kNN3 | linSVM | kNN3 | linSVM | kNN3 | linSVM |
| Projection matrices $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ with $(R_1,R_2,R_3)=(5,5,3)$ | | | | | | |
| HOSVD | 79.57 (0.82) | 88.12 (0.86) | 74.62 (0.91) | 83.49 (0.93) | 67.45 (0.94) | 76.69 (1.20) |
| HOOI | 79.71 (0.78) | 88.34 (0.84) | 74.81 (0.94) | 83.79 (0.95) | 67.74 (0.93) | 77.01 (1.20) |
| HODA | 73.36 (1.16) | 81.47 (1.28) | 68.07 (1.13) | 76.32 (1.09) | 61.63 (2.05) | 70.19 (2.46) |
| TLLD | 72.60 (0.96) | 80.88 (0.91) | 67.41 (1.11) | 75.53 (1.08) | 59.67 (1.17) | 68.00 (1.14) |
| LTDA | 81.08 (0.96) | 88.35 (0.79) | 76.64 (1.50) | 84.19 (1.14) | 70.28 (1.40) | 76.94 (1.22) |
| MITD (HOSVD) | 83.44 (1.94) | 91.38 (1.77) | 78.33 (3.13) | 86.89 (2.97) | 71.12 (4.14) | 79.99 (4.13) |
| MITD (LTDA) | **87.97 (2.11)** | **93.63 (1.51)** | **85.82 (2.15)** | **91.77 (1.77)** | **80.22 (2.71)** | **87.01 (2.46)** |
| Projection matrices $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ with $(R_1,R_2,R_3)=(10,10,3)$ | | | | | | |
| HOSVD | 64.61 (0.99) | 79.40 (0.92) | 59.47 (1.21) | 74.41 (0.79) | 52.23 (1.52) | 66.88 (0.87) |
| HOOI | 64.55 (0.97) | 79.43 (0.96) | 59.30 (1.20) | 74.41 (0.78) | 52.10 (1.49) | 66.84 (0.86) |
| HODA | 62.47 (1.12) | 77.60 (0.77) | 57.59 (1.54) | 72.49 (0.76) | 50.43 (2.02) | 65.05 (0.99) |
| TLLD | 61.26 (1.07) | 77.26 (0.83) | 56.00 (1.51) | 72.05 (0.80) | 48.44 (1.97) | 64.71 (0.91) |
| LTDA | 71.69 (1.47) | 81.19 (0.96) | 67.68 (1.81) | 76.73 (1.34) | 61.09 (1.79) | 69.62 (1.43) |
| MITD (HOSVD) | 83.71 (1.46) | 92.52 (1.15) | 79.55 (1.47) | 89.47 (1.23) | 74.07 (1.76) | 84.63 (1.67) |
| MITD (LTDA) | **88.68 (1.87)** | **94.44 (1.18)** | **85.06 (1.98)** | **91.53 (1.50)** | **80.46 (1.95)** | **87.80 (1.89)** |

Automatically selected number of components

| Number of components | 8 train (HO90%) | | | 6 train (HO93%) | | | 4 train (HO95%) | | |
|---|---|---|---|---|---|---|---|---|---|
| | $R_1$ | $R_2$ | $R_3$ | $R_1$ | $R_2$ | $R_3$ | $R_1$ | $R_2$ | $R_3$ |
| Min | 6 | 6 | 3 | 6 | 7 | 3 | 7 | 6 | 3 |
| Max | 8 | 9 | 3 | 9 | 10 | 3 | 11 | 11 | 3 |
| Average | 6.64 | 7.68 | 3.00 | 7.90 | 7.96 | 3.00 | 9.52 | 8.76 | 3.00 |
| Method | kNN | linSVM | kNN3 | linSVM | kNN3 | linSVM | | | |
| LTDA | 76.80 (1.60) | 84.88 (1.21) | 71.90 (1.95) | 79.82 (1.58) | 63.30 (2.42) | 71.35 (2.06) | | | |
| MITD (LTDA) | **88.71 (1.65)** | **94.29 (1.24)** | **85.50 (2.37)** | **91.69 (1.78)** | **81.32 (1.85)** | **88.46 (1.74)** | | | |

### 4.2. Face recognition

The Sheffield Face database (SFD) consists of 575 images of 20 individuals with mixed race gender and appearance. Each individual is shown in a range of poses from profile to frontal views (Graham and Allison, 1998), with each image cropped to $112 \times 92$ pixels with 8 bit gray levels per pixels. Prior to feature extraction all images were downsampled to $28 \times 23$ pixels, and raw images were used as input for feature extraction. Training set was formed by randomly selecting four, six or eight samples

**Table 5**
Face recognition (SFD). Accuracy was estimated using 50 random partitions. Results reported in % as accuracy (standard deviation).

| Method | 8 train (HO90%) | | 6 train (HO93%) | | 4 train (HO95%) | |
|---|---|---|---|---|---|---|
| | kNN3 | linSVM | kNN3 | linSVM | kNN3 | linSVM |
| Projection matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ with 5 components, $(R_1,R_2)=(5,5)$ | | | | | | |
| HOSVD | 86.14 (2.73) | 94.06 (1.78) | 79.37 (2.66) | 90.26 (2.38) | 69.08 (3.80) | 81.56 (2.96) |
| HOOI | 85.81 (2.92) | 94.22 (1.86) | 79.16 (2.80) | 90.46 (2.69) | 68.85 (4.07) | 81.71 (3.14) |
| HODA | 77.35 (3.19) | 92.12 (2.91) | 70.76 (3.87) | 86.51 (4.29) | 60.30 (4.25) | 74.63 (4.63) |
| TLLD | 76.80 (2.89) | 92.72 (2.44) | 70.53 (3.54) | 87.21 (3.15) | 60.27 (4.00) | 75.14 (4.17) |
| LTDA | **90.24 (2.74)** | 94.79 (1.95) | **84.16 (2.44)** | **90.62 (2.28)** | **73.71 (3.65)** | 81.38 (2.69) |
| MITD (HOSVD) | 86.22 (2.63) | 94.07 (1.88) | 80.00 (2.93) | 90.57 (2.40) | 69.68 (4.01) | **82.06 (3.05)** |
| MITD (LTDA) | 90.20 (2.82) | **94.87 (2.01)** | 83.96 (2.48) | 90.54 (2.27) | 73.60 (3.41) | 81.67 (2.94) |
| Projection matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ with 10 components, $(R_1,R_2)=(10,10)$ | | | | | | |
| HOSVD | 81.35 (3.13) | 92.56 (2.32) | 74.90 (3.52) | 87.28 (2.98) | 64.29 (3.63) | 76.36 (2.94) |
| HOOI | 81.27 (3.13) | 92.55 (2.28) | 74.72 (3.52) | 87.17 (2.99) | 64.13 (3.64) | 76.25 (2.96) |
| HODA | 77.73 (3.15) | 90.35 (2.53) | 69.99 (3.77) | 83.60 (3.61) | 59.79 (3.69) | 72.17 (3.48) |
| TLLD | 76.54 (3.41) | 89.93 (2.64) | 68.94 (3.44) | 83.11 (3.24) | 58.90 (3.97) | 70.98 (3.94) |
| LTDA | 79.45 (4.14) | 88.31 (2.76) | 71.69 (3.80) | 82.37 (2.81) | 60.79 (3.84) | 69.99 (3.45) |
| MITD (HOSVD) | 82.01 (4.11) | 93.50 (2.57) | 73.89 (3.68) | 87.32 (3.06) | 63.39 (3.40) | 76.16 (3.19) |
| MITD (LTDA) | **92.11 (2.44)** | **96.43 (1.52)** | **87.07 (2.65)** | **93.82 (2.01)** | **77.54 (3.55)** | **86.89 (3.29)** |
| Automatically selected number of components | | | | | | |
| Number of components | 8 train (HO90%) | | 6 train (HO93%) | | 4 train (HO95%) | |
| | $R_1$ | $R_2$ | $R_1$ | $R_2$ | $R_1$ | $R_2$ |
| Min | 11 | 3 | 11 | 4 | 11 | 5 |
| Max | 13 | 6 | 14 | 6 | 15 | 11 |
| Average | 11.8 | 4.6 | 12.3 | 5.02 | 13.3 | 6.68 |
| Method | kNN | linSVM | kNN3 | linSVM | kNN3 | linSVM |
| LTDA | 89.49 (2.71) | 94.63 (1.86) | 83.47 (2.71) | 90.10 (2.25) | 67.56 (4.09) | 76.19 (3.66) |
| MITD (LTDA) | **90.42 (2.42)** | **95.28 (1.75)** | **84.59 (2.56)** | **91.01 (2.26)** | **73.94 (6.15)** | **82.89 (5.11)** |

**Table 6**
Recognition of handwritten digits with automatically selected number of components. Accuracy was estimated using 50 random partitions. HO denotes hold-out ratio. Results reported in % as accuracy (standard deviation).

| Number of components | 10 train (HO80%) | | 5 train (HO90%) | |
|---|---|---|---|---|
| | $R_1$ | $R_2$ | $R_1$ | $R_2$ |
| Min | 5 | 5 | 7 | 5 |
| Max | 9 | 9 | 10 | 9 |
| Average | 7.38 | 6.22 | 8.42 | 6.70 |
| Method | kNN | linSVM | kNN3 | linSVM |
| HOSVD | 70.31 (2.30) | 73.39 (2.80) | 59.14 (3.45) | 63.44 (3.61) |
| HOOI | 70.22 (2.13) | 73.64 (2.85) | 59.43 (3.42) | 63.85 (3.64) |
| HODA | 66.63 (3.74) | 71.19 (3.31) | 55.92 (3.32) | 62.73 (3.66) |
| TLLD | 66.72 (3.44) | 70.94 (2.93) | 56.69 (3.91) | 62.70 (3.75) |
| LTDA | 71.44 (2.59) | 72.81 (2.24) | 57.17 (3.45) | 59.38 (3.09) |
| MITD (HOSVD) | 71.17 (2.37) | 74.00 (2.64) | 61.33 (4.79) | **65.26 (4.30)** |
| MITD (LTDA) | **73.39 (2.27)** | **74.56 (2.40)** | **61.38 (3.89)** | 62.41 (4.46) |

for each class with remaining images forming the test set. In our experiments we set the number of components $(R_1,R_2)\in\{(5,5),(10,10)\}$ and no feature selection was performed on the extracted features. Classification accuracy was estimated on 50 random partitions and results are presented in Table 5. We also performed a series of experiments with automatically selected number of components, and compared LTDA with automatically selected number of dimensions against MITD initialized using LTDA. With the number of components in each projection matrix fixed to five, in most cases the best results are obtained using LTDA features. However, when the number of features is increased, MITD initialized with LTDA extracts more discriminative features and achieves superior accuracy. In direct comparison of LTDA and MITD initialized using automatically selected number of projections the proposed method is much better. Also, this experiment demonstrates that the performance of LTDA using automatically selected dimensions is suboptimal, while the proposed procedure can provide significant increase in overall performance.

### 4.3. Classification of handwritten digits

In this experiment we used a subset of MNIST database of images of handwritten digits. MNIST originally consists of 60,000 training samples and 10,000 test samples, with each sample containing size-normalized and centered image of a digit. All images are in grayscale and have fixed size of $28\times28$ pixels. We selected first 50 images of each digit to create data set for our experiments. Training set was formed by randomly selecting five or ten images per class, with remaining images used as a test set. Classification accuracy was estimated using 50 random partitions and results are presented in Table 6. Here we performed comparison of all methods using the automatically selected dimensions. It can be seen that HOSVD features result in better accuracy than the ones obtained with LTDA. However, in almost all experiments MITD initialized with LTDA achieved the best accuracy.

### 4.4. Discussion

The results clearly demonstrate the ability of the proposed approach to improve discriminative power of the extracted features. In almost all experiments the proposed approach achieved superior results. As validated through experiments, features extracted using MITD result in considerably better classification performance. However, selecting the number of projections for each of the modes remains an important practical issue. This is a well-known problem in feature extraction for vector objects, and it is even more emphasized in tensor case since the number of possible combinations grows exponentially with the order of a tensor. Experiments showed that the proposed approach enables extraction of more discriminative features even when their number is significantly increased. As demonstrated, all methods except MITD achieve lower accuracies when the number of features is increased, meaning that their discriminative power is reduced, while the MITD-based features lead to even better accuracy. Still, selecting the optimal number of projections remains an open problem for future research. The drawback of the proposed method is the computationally demanding optimization procedure that needs to be solved in each alternating step. Previously proposed methods based on discriminant analysis (such as HODA, LTDA) solve a (generalized) eigenvalue problem in each alternation and thus are significantly faster. However, the speed issue should not be critical, since in the actual application projection matrices are learned only once on a large training set, and then used repeatedly to extract features from the test samples.

## 5. Conclusion

In this paper we proposed a novel approach for supervised feature extraction for tensor objects. The projection matrices are obtained by maximizing an approximation of mutual information between the extracted features and class labels. As opposed to methods that exploit only second order statistics of the data, more discriminative features can be obtained by using higher order statistics. It was shown in several experiments that the proposed approach can be used to significantly improve discriminative ability of the features extracted from tensor objects. Note that even better results can be expected with utilization of more powerful classifiers. Since the extracted features are also in tensor form, the overall performance could further benefit from using classifiers designed for tensor objects. We hope that the results presented here will provide motivation for further research in information-theoretic approaches for discriminative analysis of tensor objects.

## Acknowledgements

This work has been supported in part through grant 098-0982903-2558 funded by the Ministry of Science, Education and Sports, Republic of Croatia. The authors would like to thank the three anonymous reviewers for comments that helped to improve the manuscript.

## Appendix A. Estimation of MI and its gradient for scalar variables

Here we provide expressions for estimation of mutual information for scalar variable and its gradient using approximation as in Leiva-Murillo and Artès-Rodrìgues (2007). Consider a continuous (scalar) random variable $f$, with $f = \mathbf{w}^T\mathbf{x}$, and a discrete (scalar) random variable $y$ with values in $\{1,\dots,C\}$. Negentropy of a random variable $f$ is defined as

$$\mathcal{J}(f) = h(f_{Gauss}) - h(f), \tag{A.1}$$

where $f_{Gauss}$ is a random variable with the same mean and variance as $f$. The mutual information $I(f,y)$ between scalar random variables can be expressed as

$$I(f,y) = \log\left((2\pi e)^{\frac{1}{2}}\sigma_f\right) - \mathcal{J}(f) - \sum_{k=1}^{C}\mathbb{P}(y=k)\left[\log((2\pi e)^{\frac{1}{2}}\sigma_{f|y=k}) - \mathcal{J}(f|y=k)\right], \tag{A.2}$$

and its gradient with respect to the transformation $\mathbf{w}$ as

$$\nabla_{\mathbf{w}} I(f,y) = \nabla_{\mathbf{w}} I(\mathbf{w}^T\mathbf{x},y) = \sum_{k=1}^{C}\mathbb{P}(y=k)\nabla_{\mathbf{w}}\mathcal{J}(f|y=k) - \nabla_{\mathbf{w}}\mathcal{J}(f) - \sum_{k=1}^{C}\mathbb{P}(y=k)\frac{\mathbf{C}_{\mathbf{x}|y=k}\mathbf{w}}{\mathbf{w}^T\mathbf{C}_{\mathbf{x}|y=k}\mathbf{w}}, \tag{A.3}$$

with $\mathbb{P}(y=k)$ being probability of class $k$, $\sigma$ standard deviation, and $\mathbf{C}$ covariance matrix, all estimated using the training set. In our calculations, negentropy $\mathcal{J}$ and its gradient are approximated using nonpolynomial functions (see Eq. (5.48) in Hyvärinen et al. (2001)) as follows

$$\mathcal{J}(f) = a_1\left(\mathbb{E}\left[f\exp\left(\frac{-f^2}{2}\right)\right]\right)^2 + a_2\left(\mathbb{E}\left[\exp\left(\frac{-f^2}{2}\right)\right] - \sqrt{\frac{1}{2}}\right)^2, \tag{A.4}$$

$$\nabla_{\mathbf{w}}\mathcal{J}(f) = \nabla_{\mathbf{w}}\mathcal{J}(\mathbf{w}^T\mathbf{x}) = 2a_1\mathbb{E}\left[f\exp\left(\tfrac{-f^2}{2}\right)\right]\mathbb{E}\left[\mathbf{w}(1-f^2)\exp\left(\tfrac{-f^2}{2}\right)\right] - 2a_2\left(\mathbb{E}\left[\exp\left(\tfrac{-f^2}{2}\right)\right] - \sqrt{\tfrac{1}{2}}\right)\mathbb{E}\left[\mathbf{w}f\exp\left(\tfrac{-f^2}{2}\right)\right], \tag{A.5}$$

with constants $a_1 = 36/(8\sqrt{3}-9)$ and $a_2 = 24/(16\sqrt{3}-27)$. Note that definitions of negentropy and its gradient involve mathematical expectation of a random variable that is estimated by averaging over the training samples.

## References

Bader, B., Kolda, T., and others, 2012. MATLAB Tensor Toolbox Version 2.5. <http://www.sandia.gov/~tgkolda/TensorToolbox/index-2.5.html>.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3), 27:1–27:27.

Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.I., 2009. Nonnegative Matrix and Tensor Factorizations. John Wiley & Sons, Chichester, UK.

Cong, F., Phan, A. H., Zhao, Q., Wu, Q., Ristaniemi, T., Cichocki, A., 2012. Feature Extraction by Nonnegative Tucker Decomposition from EEG Data Including Testing and Training Observations. In: Proceedings of The International Conference on Neural Information Processing (ICONIP), Doha, Qatar, pp. 166–173.

Cover, T.M., Thomas, J.A., 1991. Elements of Information Theory. Wiley, New York, USA.

Graham, D.B., Allison, N.M., 1998. Characterizing virtual eigensignatures for general purpose face recognition. In: Wechsler, H., Phillips, P.J., Bruce, V., Fogelman-Soulie, S., Huang, T.S. (Eds.), Face Recognition: From Theory to Applications. NATO ASI Series F, 163. Computer and Systems Sciences, pp. 446–456.

Hou, C., Feiping, N., Yi, D., Wu, Y., 2013. Efficient image classification via multiple rank regression. IEEE Trans. Image Process. 22 (1), 340–352.

Hyvärinen, A., Karhunen, J., Oja, E., 2001. Independent Component Analysis. Wiley, New York, USA.

Kamandar, M., Ghassemian, H., 2013. Linear feature extraction for hyperspectral images based on information theoretic learning. IEEE Geosci. Remote Sens. Lett. 10 (4), 702–706.

Kolda, T., Bader, B.W., 2009. Tensor decompositions and applications. SIAM Rev. 51 (3), 455–500.

Leiva-Murillo, J.M., Artès-Rodrìgues, A., 2007. Maximization of mutual information for supervised linear feature extraction. IEEE Trans. Neural Networks 18 (5), 1433–1441.

Nie, F., Xiang, S., Song, Y., Zhang, C., 2007. Optimal dimensionality discriminant analysis and its application to image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07), Minneapolis, Minnesota, USA, pp. 1–8.

Nie, F., Xiang, S., Song, Y., Zhang, C., 2009. Extracting the optimal dimensionality for local tensor discriminant analysis. Pattern Recogn. 42, 105–114.

Petridis, S., Perantonis, S.J., 2004. On the relation between discriminant analysis and mutual information for supervised linear feature extraction. Pattern Recogn. 37, 857–874.

Phan, A. H., 2012. NFEA: Tensor Toolbox for Feature Extraction and Applications. <http://www.bsp.brain.riken.jp/~phan/nfea/nfea.html>.

Phan, A.H., Cichocki, A., 2010. Tensor decompositions for feature extraction and classification of high dimensional datasets. IEICE Nonlinear Theory Appl. 1, 37–68.

Phan, A.H., Cichocki, A., 2011. Extended HALS algorithm for nonnegative Tucker decomposition and its applications for multiway analysis and classfication. Neurocomputing 74, 1956–1969.

Signoretto, M., De Lathauwer, L., Suykens, J.A., 2011. A kernel-based framework for tensorial data analysis. Neural Networks 24 (8), 861–874.

Tao, D., Li, X., Wu, X., Maybank, S.J., 2007. General tensor discriminant analysis and Gabor features for gait recognition. IEEE Trans. Pattern Anal. Mach. Intell. 29 (10), 1700–1715.

Torkkola, K., 2003. Feature extraction by non-parametric mutual information maximization. J. Mach. Learn. Res. 3, 1415–1438.

Tucker, L.R., 1964. The extension of factor analysis to three-dimensional matrices. In: Gulliksen, H., Frederiksen, N. (Eds.), Contributions to Mathematical Psychology. Holt, Reinhart and Winston, New York, pp. 110–127.

Tucker, L.R., 1966. Some mathematical notes on three-mode factor analysis. Psychometrika 31, 279–311.

Wen, Z., Yin, W., 2012a. A feasible method for optimization with orthogonality constraints. Math. Program..

Wen, Z., Yin, W., 2012b. Optimization with Orthogonality Constraints. <http://optman.blogs.rice.edu/>.

Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X. Z.-J., 2005. Discriminant Analysis with Tensor Representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, USA, pp. 526–532.

Zhang, W., Lin, Z., Xiaoou, T., 2009. Tensor linear Laplacian discrimination (TLLD) for feature extraction. Pattern Recogn. 42, 1941–1948.