Journal of Mathematical Modeling
Vol. 11, No. 1, 2023, pp. 83–101. Research Article

JMM

Multilinear discriminant analysis using tensor-tensor products

Franck Dufrenois[†], **Alaa El Ichi**[‡], **Khalide Jbilou**[§]*

[†]*LISIC,50 rue F. Buisson, ULCO Calais, France*
[‡]*Laboratoire de Mathmatiques, Informatique et Applications, Sécurité de l'Information LABMIA-SI, University Mohamed V, Rabat Morocco; University Littoral Cote d'Oplae, France*
[§]*LMPA, 50 rue F. Buisson, ULCO Calais, France; Mohammed VI Polytechnic University, Green City, Morocco*
Email(s): franck.dufrenois@univ-littoral.fr, elichi.alaa@gmail.com, jbilou@univ-littoral.fr

Abstract. Multilinear Discriminant Analysis (MDA) is a powerful dimension reduction method specifically formulated to deal with tensor data. Precisely, the goal of MDA is to find mode-specific projections that optimally separate tensor data from different classes. However, to solve this task, standard MDA methods use alternating optimization heuristics involving the computation of a succession of tensor-matrix products. Such approaches are most of the time difficult to solve and not natural, highligthing the difficulty to formulate this problem in fully tensor form. In this paper, we propose to solve multilinear discriminant analysis (MDA) by using the concept of transform domain (TD) recently proposed in [15]. We show here that moving MDA to this specific transform domain make its resolution easier and more natural. More precisely, each frontal face of the transformed tensor is processed independently to build a separate optimization sub-problems easier to solve. Next, the obtained solutions are converted into projective tensors by inverse transform. By considering a large number of experiments, we show the effectiveness of our approach with respect to existing MDA methods.

Keywords: Krylov subspaces, linear tensor equations, tensor L-product.
AMS Subject Classification 2010: 65F10, 65F22.

1 Introduction

Linear Discriminant Analysis (LDA) is a supervised dimensionality reduction tool allowing the classification to multiple categories in datasets. There are used in numerous areas as diverse as speech and music

*Corresponding author.
Received: 26 August 2022 / Revised: 12 December 2022 / Accepted: 2 January 2023
DOI: 10.22124/JMM.2023.22841.2026

classification [1], video classification [25], outlier detection [28], supervised novelty detection [4,6], etc.. Considering that each data is labeled, the goal of LDA is to find a set of projections which maximizes the between-class scatter while minimizing the within-class scatter. In the litterature, this objective is commonly formulated in two different ways as solving: a **trace ratio problem** which is typically nonconvex and need an iterative optimization procedure or a **ratio trace problem** which is inexact but equivalent to a generalized eigenvalue problem (GEP).

When dealing with high order data such as grayscale images, RGB images, multispectral images, . . . a conventional practice is to vectorize the whole data set before applying LDA. This preprocessing involves systematically an increase of the dimensionality of the data sample and may result in singularity problems commonly referred as the small sample size problem (SSS).

A way to solve this question is to adopt the tensor representation which allows to preserve the natural multidimensional form of the data while reducing implicitly the dimensionality of the data. A n-order tensor can be seen as a hyper-parallepiped with n sides and where each side represents a “mode”. A tensor generalizes thus the notions of matrix (2-order tensor) and vector (1-order tensor).

In this framework, to perform LDA on tensor data, several methodologies have been introduced in the past. Among them, we can cite discriminant analysis with tensor representation (DATER) [33], Tensor subspace analysis (TSA) [11], multilinear discriminant analysis (MDA) [32, 33] and constrained multilinear discriminant analysis (CMDA) [20]. The principle of MDA is to find a lower dimensional tensor subspace represented by orthonormal matrices. However, the main drawback of the current approaches relies on its optimization step which is based on heuristic optimization approaches.

In this paper, we propose a new way to compute linear discriminant analysis from three-order tensors. This new method is based on recent developments on tensor-tensor products [3, 7, 8, 13, 15]. The first work on this issue is due to Kilmer et al. [15] where they introduced the notion of the t-product which allows to mutiply easily 3-order tensors. The multiplication uses a convolution-type operation which can be advantageously computed by Fast Fourier Transform (FFT). More recently, Kernfeld et al. extended this approach and defended the principle that any tensor-tensor product can be defined with arbitrary invertible linear transforms [13]. As an example, they introduced the tensor cosine transform product which is an alternative of the t-product and can be efficiently computed by the Discret Cosine Transform (DCT).

Motivated by these works, we propose to solve Tensor Linear Discriminant Analysis (TLDA) by using the concept of “transform-domain” and the cited new family of tensor-tensor products. We show here that moving TLDA to the transform domain (TD) makes its resolution easier and more natural. First, one of the fundamental step of the previous tensor LDA approaches is to compute some tensor-matrix products that involve a succession of tensor unfoldings. In this context, the corresponding optimization problem relies on the determination of a set of projection matrices i.e., two-dimensional projective subspaces. Such an approach shows the difficulty to formulate this problem in fully tensor form by directly searching for a projective tensor instead of a set of projective matrices. By moving in the transform domain, the main ingredients of TLDA can be formulated as tensor-tensor products which are efficiently computed by a sequence of matrix-matrix products. Secondly, a key point of this approach is that the optimization problem can be solved in the transform domain benefiting from its properties. More precisely, each frontal face of the transformed tensor are processed independently to build independent optimization sub-problems easier to address. The obtained solutions are returned in the form of a projective tensor by inverse transform. This paper is organized as follows: Section 2 introduces the notation and the main definitions. Section 3 recalls the main definitions of the tensor-tensor product under the concept of invertible linear transforms. In Section 4, after recalling the principle of the standard LDA (matricial case) and the tensor LDA (TLDA) formulated with the n-product, we present a new multilinear discriminant analysis based on third-order tensors and formulated with new tensor-tensor products. Section 5 analyses and compares the performance of the proposed approach with recent tensor based LDA approaches on several multidimensional data sets. Section 6 ends the paper with a conclusion.

2 Notation and preliminaries

Scalars, vectors, matrices and high-order tensors will be denoted by lowercase letters, e.g. a, boldface lowercase letters, e.g. $\boldsymbol{a}$, capital letters, e.g. A and Euler script letters, e.g. $\mathscr{A}$, respectively. In this work, we will limit our study to third-order tensors. Third-order tensors are compact and well adapted to represent multidimensional data from vision based applications such as face identification, video monitoring or classification of multispectral images. Let $\mathscr{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be a third-order tensor. By convention, the first dimension is devoted to the *pixels* of the images, the second dimension for the *number of images* and the third dimension is for the *number of modalities of the images*. As an illustrative example, if we consider a sequence of l color images of size $n \times m$, the corresponding third order tensor will be sized as follows: $n_1 = nm$, $n_2 = l$ and $n_3 = 3$.

The tensor $\mathscr{A}$ is sampled by a triplet of indexes (i, j, k) which allows to select different subparts of $\mathscr{A}$. By fixing the whole set of indexes we obtain a scalar entry of $\mathscr{A}$ denoted by a_{ijk} and fixing two indexes over three, we select a *fiber* of $\mathscr{A}$. We will denote a column (*1-mode*), row (*2-mode*) and tube (*3-mode*) fiber by $\boldsymbol{a}_{:jk}$, $\boldsymbol{a}_{i:k}$ and $\boldsymbol{a}_{ij:}$, respectively. Lastly, by fixing one index over three, we define a *slice* of $\mathscr{A}$. Therefore, slices are declined in three modes: horizontal (*1-mode*), lateral (*2-mode*) and frontal (*3-mode*) slides which are represented by $A_{i::}$, $A_{:j:}$ and $A_{::k}$, respectively. In the sequel, the k^{th} frontal slide of a third-order tensor will be denoted more compactly by$A^{(k)}$.

Manipulating tensors needs specific algebra. Here we just list some definitions which are directly relevant to this paper. For a detailed description, see for example [15, 17]. .

2.1 Tensor unfolding

Tensor unfolding or flattening consists in reordering the elements of a tensor into a matrix. Consider the general case of a N^{th}order tensor $\mathscr{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$, flattening $\mathscr{A}$ along the k^{th} mode or the ***k-mode*** matricization of $\mathscr{A}$ gives a matrix denoted $A_{(k)}$ which consists in arranging the *k-mode* fibers to be the columns of the resulting matrix.

2.2 Tensor products

Let us recall several tensor products.

Definition 1 (k-mode product). *Consider the general case of an N^{th} order tensor $\mathscr{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$. The **k-mode** product of $\mathscr{A}$ with a matrix $U \in \mathbb{R}^{m \times n_k}$ is a new tensor $\mathscr{B} \in \mathbb{R}^{n_1 \times \ldots \times n_{k-1} \times \boldsymbol{m} \times n_{k+1} \times \ldots \times n_N}$ defined by*

$$\mathscr{B} = \mathscr{A} \times_k U, \tag{1}$$

which is equivalent to the following matrix-matrix product

$$B_{(k)} = U A_{(k)}, \tag{2}$$

where $A_{(k)}$ and $B_{(k)}$ denotes the k-mode matricization (see section 2.1) of $\mathscr{A}$ and $\mathscr{B}$, respectively.

Definition 2 (generalization of the k-mode product). *Consider the general case of an N^{th}order tensor $\mathscr{A} \in \mathbb{R}^{n_1\times n_2\times \ldots \times n_N}$. The multiplication of $\mathscr{A}$ with a set of N matrices $\{U_k \in \mathbb{R}^{m_k\times n_k}, k=1,\ldots,N\}$ is defined by*

$$\mathscr{B} = \mathscr{A}\Pi_{i=1}^{n} \times_i U_i = \mathscr{A} \times_1 U_1 \times_2 U_2 \times \cdots \times_N U_N. \tag{3}$$

The k-mode matricization of $\mathscr{B}$ can be obtained by

$$B_{(k)} = U_k A_{(k)} \mathbb{U}_k, \tag{4}$$

where $\mathbb{U}_k = U_1 \otimes \cdots \otimes U_{k-1} \otimes U_{k+1} \otimes \cdots \otimes U_N$.

Definition 3 (face-wise product [13]). *Let $\mathscr{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathscr{B} \in \mathbb{R}^{n_2\times m\times n_3}$ be two third-order tensors, then the **face-wise** product between them consists in computing a matrix-matrix product between the 3-mode slides of $\mathscr{A}$ and $\mathscr{B}$ as follows*

$$(A\triangle B)^{(i)} = A^{(i)}B^{(i)}, \quad i=1,\ldots,n_3. \tag{5}$$

2.3 Specific block matrices

Tensor-tensor products require the definition of specific structured block matrices build from the frontal slices of the third-order tensor. We recall here some definitions

Definition 4. *The **Toeplitz-plus-Hankel** matrix of the tensor $\mathscr{A}$ is a $n_1n_3 \times n_2n_3$ block matrix composed of the frontal slices $A^{(i)}$, $i=1,\ldots,n_3$ of $\mathscr{A}$ and defined by*

$$mat(\mathscr{A}) = \begin{pmatrix} A^{(1)} & A^{(2)} & \cdots & A^{(n_3)} \\ A^{(2)} & A^{(1)} & \cdots & A^{(n_3-1)} \\ \vdots & \vdots & \ddots & \vdots \\ A^{(n_3)} & A^{(n_3-1)} & \cdots & A^{(1)} \end{pmatrix} + \begin{pmatrix} A^{(1)} & \cdots & A^{(n_3)} & \mathbf{0} \\ \vdots & \ddots & \ddots & A^{(n_3)} \\ A^{(n_3)} & \mathbf{0} & \ddots & \vdots \\ \mathbf{0} & A^{(n_3)} & \cdots & A^{(1)} \end{pmatrix}, \tag{6}$$

where $\mathbf{0}$ denotes the zero matrix of size $n_1 \times n_2$. We will denote ten the inverse operator such as:

$$ten(mat(\mathscr{A})) = \mathscr{A}.$$

Definition 5. *The **block circulant** matrix of a tensor $\mathscr{A}$ is the $n_1n_3 \times n_2n_3$ block matrix composed by the frontal slices of $\mathscr{A}$ and defined by*

$$bcirc(\mathscr{A}) = \begin{pmatrix} A^{(1)} & A^{(n_3)} & \cdots & A^{(2)} \\ A^{(2)} & A^{(1)} & \cdots & A^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ A^{(n_3)} & A^{(n_3-1)} & \cdots & A^{(1)} \end{pmatrix}. \tag{7}$$

Definition 6. *The **block diagonal** matrix of a tensor $\mathscr{A}$ is the $n_1n_3 \times n_2n_3$ block matrix composed by the frontal slices of $\mathscr{A}$ and defined by*

$$bdiag(\mathscr{A}) = \begin{pmatrix} A^{(1)} & 0 & \cdots & 0 \\ 0 & A^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A^{(n_3)} \end{pmatrix}. \tag{8}$$

3 Tensor-tensor products with invertible linear transform

Recently, a new type of tensor-tensor products, called t-product, has been proposed in [14]. The t-product generalizes matrix multiplication for third-order tensors. It is based on a convolution-like operation which is efficiently computed by the Fast Fourier Transform (FFT). This work opens the way towards the idea that there exits a *transform domain* where the tensor-tensor product can be defined. Motivated by this idea, Kernfeld et al. [13] extend this concept by introducing a new family of tensor-tensor products which can be efficiently computed in a *transform domain* for any invertible linear transform. To illustrate their principle, they defined the c-product which is an alternative to the t-product which can be efficiently computed in the transform domain via the discret cosinus transform (DCT). In the sequel, we will consider that the result of the transformation is at more of complex type. We first recall the main properties and definitions introduced in [13] .

Definition 7. *Let $L:\mathbb{R}^{1\times 1\times n_3}\to\mathbb{C}^{1\times 1\times n_3}$ be an invertible transform and $\mathscr{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$ be a third-order tensor. L transforms any tube fibers $\boldsymbol{a}\in\mathbb{R}^{1\times 1\times n_3}$ of $\mathscr{A}$ into $\widetilde{\boldsymbol{a}}\in\mathbb{C}^{1\times 1\times n_3}$ in the following way*

$$\widetilde{\boldsymbol{a}}^{(k)}=(L(\boldsymbol{a}))^{(k)}=(M.vec(\boldsymbol{a}))_k \quad k=1,\ldots,n_3,$$

where M is a $n_3\times n_3$ invertible matrix associated to L and $vec(\boldsymbol{a})$ is the vector in $\mathbb{R}^{n_3}$ whose elements are the elements of the tube $\boldsymbol{a}$. From a practical point of view, $\widetilde{\mathscr{A}}\in\mathbb{C}^{n_1\times n_2\times n_3}$, the transform domain version of $\mathscr{A}$, can be efficiently computed as follows

$$\widetilde{\mathscr{A}}=L(\mathscr{A})=\mathscr{A}\times_3 M. \tag{9}$$

Similarly, we have

$$\mathscr{A}=L^{-1}(\widetilde{\mathscr{A}})=\widetilde{\mathscr{A}}\times_3 M^{-1}, \tag{10}$$

*where $\times_3$ is the **3-mode** product as defined in (1).*

Notice that the matrix M is specific to the transform L and to the corresponding tensor-tensor product.

Definition 8. *Let $op_L(\mathscr{A})$ be a structured block matrix build from $\mathscr{A}$ and specific to the L-transform. Let $\boldsymbol{A}\in\mathbb{C}^{n_1n_3\times n_2n_3}$ be a block diagonal matrix where each block represents a frontal slice of the transform tensor $\widetilde{\mathscr{A}}$, then it can be shown that $\widetilde{A}$ results from the block diagonalization of $op_L(\mathscr{A})$ as*

$$\boldsymbol{A}=bdiag\left(\widetilde{\mathscr{A}}\right)=(M\otimes I_{n_1})op_L(\mathscr{A})(M^{-1}\otimes I_{n_2}), \tag{11}$$

where M is the $n_3\times n_3$ transform matrix associated to L and I_n is the $n\times n$ identity matrix.

The L-product of two tensors is defined as follows

Definition 9. *Let $*_L:\mathbb{R}^{m\times l\times n_3}\times\mathbb{R}^{l\times p\times n_3}\to\mathbb{R}^{m\times p\times n_3}$ be the product operator in L defined such as*

$$L(\mathscr{A}*_L\mathscr{B})=L(\mathscr{A})\triangle L(\mathscr{B}), \tag{12}$$

*where $\triangle$ is the **face-wise** product as defined in (5). Let $\mathscr{C}\in\mathbb{R}^{m\times p\times n_3}$ be the result of the L-product between $\mathscr{A}$ and $\mathscr{B}$, then we have*

$$\mathscr{C}=\mathscr{A}*_L\mathscr{B}=L^{-1}\left(L(\mathscr{A})\triangle L(\mathscr{B})\right). \tag{13}$$

The main known and used L-products are the t-product and the c-product. For these two products, the matrix M is given as follows.
For the c-product which is based on DCT, M is given by

$$M = W^{-1}C\,(I+Z)\,, \tag{14}$$

where C is the $n_3 \times n_3$ DCT matrix where each entry is defined by

$$c_{ij} = \sqrt{\frac{2-\delta_{ij}}{n_3}}\cos\left(\frac{(i-1)(2j-1)}{2n_3}\right), \quad i,j = 1,\ldots,n_3, \tag{15}$$

where δ is the Kronecker indicator. $W = diag(\boldsymbol{c}_{.1})$ is the diagonal matrix build from the first column of C and Z is an $n_3 \times n_3$ circulant upshift matrix.

For the t-product which is based on FFT, M is given by

$$M = F, \tag{16}$$

where F is the $n_3 \times n_3$ FFT matrix where each entry is defined by

$$f_{ij} = \exp\left(-j2\pi\frac{(i-1)(j-1)}{n_3}\right). \tag{17}$$

We also notice that $\text{op}(\mathscr{A})$ is defined by

$$\text{op}_L(\mathscr{A}) = \begin{cases} \text{bcirc}(\mathscr{A}), & \text{for the t-product,} \\ \text{mat}(A), & \text{for the c-product,} \end{cases}$$

where the operators *bcirc* and *mat* are defined in Section 1.

From the relation (11), it can be shown that $\mathscr{C} = \mathscr{A} *_L \mathscr{B}$ is equivalent to compute $\boldsymbol{C} = \boldsymbol{AB}$ in the transform domain. Algorithm 1 allows us to compute in an efficient way the L-product of the tensors $\mathscr{A}$ and $\mathscr{B}$.

Algorithm 1 Tensortensor product via the operator L

Inputs: $\mathscr{A} \in \mathbb{R}^{m\times l\times n_3}$, $\mathscr{B} \in \mathbb{R}^{l\times p\times n_3}$
Output: $\mathscr{C} \in \mathbb{R}^{m\times p\times n_3}$
$\widetilde{\mathscr{A}} = L(\mathscr{A})$
$\widetilde{\mathscr{B}} = L(\mathscr{B})$
for $i = 1,\ldots,n_3$ **do**
  $\mathscr{C}^{(i)} = \mathscr{A}^{(i)}\mathscr{B}^{(i)}$.
end for
$\mathscr{C} = L^{-1}(\widetilde{\mathscr{C}})$

Some basic algebraic properties are associated to L-product such as associativity, distribution over addition and invertibility. $*_L$ has a identy element, the hermitian transpose, norms and inner products (for more details see [13]).

4 Tensor LDA using the L-product: $*_L$- TLDA

Linear discriminant analysis is a supervised dimensionality reduction method which aims to find a low-dimensional projective subspace which best separates n training data vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ into c classes or clusters. In the sequel, we will consider that each data vector $\boldsymbol{x}_k$ belongs to a class indexed as $l_k \in \{1, 2, \ldots, c\}$ and each class i is defined by a set of indices C_i of length n_i such as $n = \sum_{i=1}^{c} n_i$. In this section, considering the previous notation, we propose to develop the tensor linear discriminant analysis using the L-product which will be denoted by $*_L$-TLDA. Before all, let us recall the formulation of LDA in the matricial case and the tensor LDA (using the n-product).

4.1 LDA

Consider that the training samples are collected into a matrix $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{n_1 \times n}$ where each component $\boldsymbol{x}_i$ is an n_1 dimensional data vector. Let $\boldsymbol{m} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i$ be the global centroid of X and $\boldsymbol{m}_i = \frac{1}{n_i} \sum_{i \in C_i} \boldsymbol{x}_i$ be the centroid of the data vectors belonging to the cluster i, then the goal of LDA can be defined as follows.

Definition 10 (LDA). *Let $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m] \in \mathbb{R}^{n_1 \times m}$ be a matrix defining a low-dimensional projective subspace ($m \ll n_1$) and*

$$\begin{cases} \psi_B(V) = & \sum_{j=1}^{c} n_j \left\| V^T (\boldsymbol{m}_j - \boldsymbol{m}) \right\|_F^2, & (a) \\ \psi_W(V) = & \sum_{j=1}^{c} \sum_{i \in C_j} \left\| V^T (\boldsymbol{x}_i - \boldsymbol{m}_j) \right\|_F^2, & (b) \end{cases} \tag{18}$$

be the between and within scatters measured in the projective subspace V, respectively. Then, LDA consists in finding the projective subspace V^* which maximizes the between scatter measure (a) while minimizing the within scatter measure (b) which can be formulated as follows

$$V^* = \arg \max_{V^\top V = I} \frac{\psi_B(V)}{\psi_W(V)}. \tag{19}$$

Let us introduce

$$\begin{cases} S_B = & \sum_{j=1}^{c} n_j (\boldsymbol{m}_j - \boldsymbol{m}) (\boldsymbol{m}_j - \boldsymbol{m})^T, & (a) \\ S_W = & \sum_{j=1}^{c} \sum_{i \in C_j} (\boldsymbol{x}_i - \boldsymbol{m}_j) (\boldsymbol{x}_i - \boldsymbol{m}_j)^\top, & (b) \end{cases} \tag{20}$$

the between and within scatter matrices, respectively, then it can be shown easily that the problem (19) can be re-written as

$$V^* = \arg \max_{V^\top V = I} \frac{\text{Trace}\left(V^\top S_B V\right)}{\text{Trace}(V^\top S_W V)}. \tag{21}$$

Problem (21), also referred as the **trace ratio** problem, is non-convex and does not have a closed-form solution. Fortunately, it can be shown that it is equivalent to a trace difference problem

$$V^* = \arg \max_{V^\top V = I} \quad \text{Trace}(V^T (S_B - \rho S_W) V). \tag{22}$$

which can be solved iteratively by the Newton-Lanczos algorithm [23]. Algorithm 2 summarizes the main steps of the maximization of the trace ratio problem with the Newton-Lanczos algorithm (22).

Algorithm 2 Newton-Lanczos algorithm for Trace Ratio

Input : two matrix S_A and S_B.

- Select a unitary matrix V with k columns and compute $\rho = \dfrac{\mathrm{Trace}(V^T S_B V)}{\mathrm{Trace}(V^T S_W V)}$.
- Until convergence do:
  1. Call the Lanczos algorithm to compute the largest k eigenvalues $\lambda_1(\rho), \ldots, \lambda_k(\rho)$, of $S_B - \rho S_W$ and the associated eigenvectors: $V = [v_1, \ldots, v_k]$.
  2. Set $\rho = \dfrac{\mathrm{Trace}(V_k^T S_B V_k)}{\mathrm{Trace}(V_k^T S_W V_k)}$ and go to Step 1.
- EndDo

It has been shown in [23] that this algorithm converges to a global optimum. However, the drawback of this procedure is the repeated calls to an eigensolver which can be time-consuming when the dimensionality of the data is very large. Another critical point concerns the choice of the reduced dimension. Indeed, the output dimension m is bounded by the rank of the matrix S_W since the $rank(S_B - \rho S_W) = rank(S_W)$; $(rank(S_B) < rank(S_W))$. Since $rank(S_W) < n - c$, then m is at most $n - c$. As a consequence, the optimal output dimension is related to the sample size n and its selection may be also time-consuming when the size of the training sample is very large. These observations often lead to replace the **trace ratio** problem by the simpler, but not equivalent **ratio trace** problem

$$V^* = \arg \max_{V^\top V = I} \quad \mathrm{Trace}\left((V^\top S_W V)^{-1} (V^\top S_B V) \right), \tag{23}$$

which has a closed-form solution. It is equivalent to solve the following generalized eigenvalue problem

$$S_B U = \Lambda S_W U, \tag{24}$$

where U denotes the matrix of eigenvectors and Λ the diagonal matrix of eigenvalues. Thus the projection matrix V is explicitly characterized through the eigen-decomposition of the matrix $S_W^{-1} S_B$ if S_W is nonsingular. Moreover, the dimension of the projective subspace is defined by the rank of S_B which implies that m is at most $c - 1$; ($m = c - 1$ when data are linearly independent). When S_W becomes singular, the problem is said “undersampled”, i.e., the sample size is smaller than the dimension of the data. A common strategy is to introduce regularization into the problem (24) which translates into

$$(S_W + \gamma I)^{-1} S_B U = \Lambda U, \tag{25}$$

where I is the identity matrix and $\gamma > 0$ the regularization parameter. The value of γ must be chosen with care and its selection can be obtained by cross validation.

4.2 The $\times_n$-TLDA

Consider the general case where each data sample is represented by an N^{th} order tensor $\mathscr{X}_i \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_N}$ and the sample set by a $(N+1)^{th}$ order tensor $\mathscr{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_N \times \boldsymbol{n}}$. Let $\mathscr{M} = \frac{1}{N} \sum_{i=1}^{n} \mathscr{X}_i$ be the global mean of $\mathscr{X}$ and $\mathscr{M}_i = \frac{1}{n_i}\sum_{i\in C_i}\mathscr{X}_i$ be the mean of the cluster i. Then the goal of the tensor LDA or multilinear DA can be defined as follows

Definition 11 (**$\times_n$-TLDA**). *Let $V_k\,|_{k=1}^K$ be a set low-dimensional projective matrices of size $n_k \times m_k$ with $m_k \ll n_k$ and let*

$$\begin{cases} \psi_B(V_k\,|_{k=1}^K) = & \sum_{j=1}^c n_j \left\|(\mathscr{M}_j - \mathscr{M})\,\Pi_{k=1}^K \times_k V_k\right\|_F^2, & (a) \\ \psi_W(V_k\,|_{k=1}^K) = & \sum_{j=1}^c \sum_{i\in C_j} \left\|(\mathscr{X}_i - \mathscr{M}_j)\,\Pi_{k=1}^K \times_k V_k\right\|_F^2, & (b) \end{cases} \tag{26}$$

be the between and within scatters measured in the set of projective subspace $U_k\,|_{k=1}^K$, respectively. Then, $\times_n$-TLDA consists in finding a set projective subspace V^* that maximizes the between scatter measure (a) while minimizing the within scatter measure (b), i.e.,

$$V_k^* = \arg\max_{V_k} \frac{\psi_B(V_k)}{\psi_W(V_k)},\ \ k = 1,\ldots,N. \tag{27}$$

However, the objective function (27) has no closed-form solution due to that the V_ks, $k = 1,\ldots,N$ depends on each other and the standard procedure is to solve it by an iterative optimization procedure. Considering that $\|\mathscr{X}\| = \left\|X_{(k)}\right\|_F$ and $\|X\|^2 = \text{Trace}\left(X^\top X\right) = \text{Trace}\left(XX^\top\right)$, if we assume that $K-1$ projective matrices $V_i\,|_{i=1,i\neq k}^K$ have been previously computed, then V_k is updated by maximizing

$$V_k^* = \arg\max_{V_k} \frac{\text{Trace}\left(V_k^\top S_{B(k)} V_k\right)}{\text{Trace}\left(V_k^\top S_{W(k)} V_k\right)}, \tag{28}$$

where $S_{B(k)}$ and $S_{W(k)}$ denote the between-class and within-class scatter matrices along the k^{th} mode, respectively and defined by

$$\begin{cases} S_{B(k)} = & \sum_{j=1}^{c} n_j \left((\mathscr{M}_j - \mathscr{M})_{(k)}\right) V_{\bar{k}}^\top V_{\bar{k}} \left((\mathscr{M}_j - \mathscr{M})_{(k)}\right)^\top, & (a) \\ S_{W(k)} = & \sum_{j=1}^c \sum_{i\in C_j} \left((\mathscr{X}_i - \mathscr{M}_j)_{(k)}\right) V_{\bar{k}}^\top V_{\bar{k}} \left((\mathscr{X}_i - \mathscr{M}_j)_{(k)}\right)^\top, & (b) \end{cases} \tag{29}$$

where $V_{\bar{k}} = V_K \otimes \cdots \otimes V_{k+1} \otimes V_k \otimes \cdots \otimes V_1$ (see Definition 2, (4)) and the terms $(\mathscr{A} - \mathscr{B})_{(k)}$ in (29) denote the k-mode matricization of the tensor $\mathscr{A} - \mathscr{B}$ (see Subsection 2.1). This iterative optimization procedure, also called k-mode optimization, have been originaly introduced in [32] and became the central part of several work to solve multilinear discriminant analysis (MDA), [20, 21, 32, 33]. However, all these methods solve the MDA problem from heuristic optimization procedures that do not rigorously optimize the MDA objective. In the sequel, we propose to solve MDA objective using the L-tensor-tensor products. The corresponding optimization problem can be moved into an invertible transform domain in which a closed-form solution exits. In the sequel, we propose to develop TLDA using the L-product,i.e., $*_L$-TLDA.

4.3 The $*_L$-TLDA

Assume the learning data set is composed of n samples and each sample is represented by a third-order tensor, i.e., $\left\{\mathscr{X}_i \in \mathbb{R}^{n_1\times 1\times n_3}, i = 1,\ldots,n\right\}$. The sample set can by represented by a unique third-order tensor $\mathscr{X} \in \mathbb{R}^{n_1\times \boldsymbol{n}\times n_3}$. Let $\mathscr{M} = \frac{1}{n}\sum_{i=1}^n \mathscr{X}_i$ be the global centroid of $\mathscr{X}$ and $\mathscr{M}_i = \frac{1}{n_i}\sum_{i\in C_i}\mathscr{X}_i$ be the centroid of the tensors belonging to the cluster i , then the goal of $*_L$-TLDA can be defined as follows

Definition 12 **($*_L$-TLDA).** *Let $\mathscr{V} \in \mathbb{R}^{n_1 \times K \times n_3} = [\mathscr{V}_1, \mathscr{V}_2, \ldots, \mathscr{V}_K]$, $\mathscr{V}_i \in \mathbb{R}^{n_1 \times 1 \times n_3}$ be a projective third-order tensor where K denotes the dimension of the projective subspace and*

$$\begin{cases} \psi_B(\mathscr{V}) = & \sum_{j=1}^{c} \left\| \mathscr{V}^T *_L (\mathscr{M}_j - \mathscr{M}) \right\|_F^2, \\ \psi_W(\mathscr{V}) = & \sum_{j=1}^{c} \sum_{i \in N_j} \left\| \mathscr{V}^T *_L (\mathscr{X}_i - \mathscr{M}_j) \right\|_F^2, \end{cases} \tag{30}$$

where $\psi_B(\mathscr{V})$ and $\psi_W(\mathscr{V})$ denote the between and within scatter measures, respectively. Then, the goal of $*_L$- LDA is to find a projective third-order tensor $\mathscr{V} \in \mathbb{R}^{n_1 \times k \times n_3}$ which maximizes the following objective function

$$\mathscr{V}^* \in \mathbb{R}^{n_1 \times K \times n_3} : \max_{\mathscr{V}} \frac{\psi_B(\mathscr{V})}{\psi_W(\mathscr{V})}. \tag{31}$$

Notice that the product $\mathscr{V}^T *_L (\mathscr{M}_i - \mathscr{M})$ represents the orthogonal projection of one lateral slice in $\mathbb{R}^{n_1 \times 1 \times n_3}$ or n_1 tube fibers onto $\mathscr{V}$ which generate K tube fibers. The same remark can be made for the product $\mathscr{V}^T *_L (\mathscr{X}_i - \mathscr{M}_i)$. Second, (31) can be solved more easily in the transform domain by using the fact that the L-product between tensors can be advantageously replaced by simple matrix-matrix products between the transform versions of the tensors (12). Then, the L-product is recovered by inverse transform. Let us develop the main steps:

From Definition 11, we denote $\boldsymbol{V} = bdiag(\widetilde{\mathscr{V}})$, $\boldsymbol{M} = bdiag(\widetilde{\mathscr{M}})$, $\boldsymbol{M}_i = bdiag(\widetilde{\mathscr{M}_i})$ and $\boldsymbol{X}_i = bdiag(\widetilde{\mathscr{X}_i})$ the block diagonal matrices build from the transform versions of the tensors $\mathscr{V}$, $\mathscr{M}$, $\mathscr{M}_i$ and $\mathscr{X}_i$, respectively. Since $\mathscr{C} = \mathscr{A} *_L \mathscr{B}$ is equivalent to compute $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}$ in the transform domain, the L-products $\mathscr{V}^T *_L (\mathscr{M}_i - \mathscr{M})$ and $\mathscr{V}^T *_L (\mathscr{X}_i - \mathscr{M}_i)$ can be computed via a simple product of block diagonal matrices, i.e., $\boldsymbol{V}^\top (\boldsymbol{M}_i - \boldsymbol{M})$ and $\boldsymbol{V}^\top (\boldsymbol{X}_i - \boldsymbol{M}_i)$. Then, the computation of $\psi_B(\mathscr{V})$ becomes in the transform domain

$$\begin{aligned} \widetilde{\psi}_B(\boldsymbol{V}) = & \sum_{j=1}^{c} \left\langle \boldsymbol{V}^\top (\boldsymbol{M}_j - \boldsymbol{M}), \boldsymbol{V}^\top (\boldsymbol{M}_j - \boldsymbol{M}) \right\rangle \\ = & \sum_{j=1}^{C} \text{Trace}\left(\boldsymbol{V}^\top (\boldsymbol{M}_j - \boldsymbol{M})(\boldsymbol{M}_j - \boldsymbol{M})^\top \boldsymbol{V} \right) \\ = & \text{Trace}\left(\boldsymbol{V}^\top \sum_{j=1}^{c} (\boldsymbol{M}_j - \boldsymbol{M})(\boldsymbol{M}_j - \boldsymbol{M})^\top \boldsymbol{V} \right) \\ = & \text{Trace}\left(\boldsymbol{V}^\top \boldsymbol{S}_B \boldsymbol{V} \right), \end{aligned} \tag{32}$$

with $\boldsymbol{S}_B = \sum_{j=1}^{c} (\boldsymbol{M}_j - \boldsymbol{M})(\boldsymbol{M}_j - \boldsymbol{M})^\top$ $(\in \mathbb{C}^{n_1 n_3 \times n_1 n_3})$. Using the same steps for $\psi_W(\mathscr{V})$, we obtain

$$\widetilde{\psi}_W(\boldsymbol{V}) = \text{Trace}\left(\boldsymbol{V}^\top \boldsymbol{S}_W \boldsymbol{V} \right), \tag{33}$$

with $\boldsymbol{S}_W = \sum_{j=1}^{c} \sum_{i \in C_j} (\boldsymbol{X}_i - \boldsymbol{M}_j)(\boldsymbol{X}_i - \boldsymbol{M}_j)^\top$ $(\in \mathbb{C}^{n_1 n_3 \times n_1 n_3})$.

It can be noticed that $\boldsymbol{S}_B$ (or $\boldsymbol{S}_W$) represents a block diagonal matrix where the i^{th} block is the the frontal slice $\widetilde{S}_B^{(i)}$ (or $\widetilde{S}_W^{(i)}$) of the third-order tensor $\widetilde{\mathscr{S}_B}$ (or $\widetilde{\mathscr{S}_W}$). Then a new objective function equivalent to (31) is defined in the transform domain by

$$\boldsymbol{V}^* \in \mathbb{R}^{n_1 n_3 \times K n_3} : \max_{\boldsymbol{V}} \frac{\widetilde{\psi}_B(\boldsymbol{V})}{\widetilde{\psi}_W(\boldsymbol{V})}. \tag{34}$$

As in the matrix case, (34) can be solved either by the Newton-Lanczos algorithm (21) or by eigendecomposition with regularization (25). The Newton-Lanczos algorithm involves iteratively the eigenvalue decomposition of the matrix $\boldsymbol{S}(\rho) = \boldsymbol{S}_B - \rho \boldsymbol{S}_W$. Since $\boldsymbol{S}(\rho)$ is a block diagonal matrix, i.e.,

$\boldsymbol{S}(\rho) = bdiag\left(\widetilde{S}^{(i)}(\rho)\right)$ where $\widetilde{S}(\rho) \in \mathbb{C}^{n_1 \times n_1}$, for $i = 1, \ldots, n_3$, eigen-value decomposition can be computed on each block separately. Concerning the regularized eigen-decomposition problem (21), the inversion of the block diagonal matrix $\boldsymbol{S}_W(\gamma) = (\boldsymbol{S}_W + \gamma \boldsymbol{I}) = bdiag(\widetilde{S}_W^{(i)}(\gamma))$ is also a block diagonal matrix where each block $\widetilde{S}_W^{(i)}$ is separately inverted. Since $\boldsymbol{S}_W(\gamma)$ and $\boldsymbol{S}_B$ are square matrices identically partitioned into block diagonal form, the product $\boldsymbol{S}_W^{-1}(\gamma)\boldsymbol{S}_B$ also forms a diagonal block matrix identically partitioned. Eigen-decomposition can then be computed on each block separately. Algorithms 3 and Algorithm 4 summarize the main steps for computing $*_L$-TLDA either formulated as the trace ratio problem or the ratio trace problem.

Algorithm 3 $*_L$TLDA - Trace ratio optimization

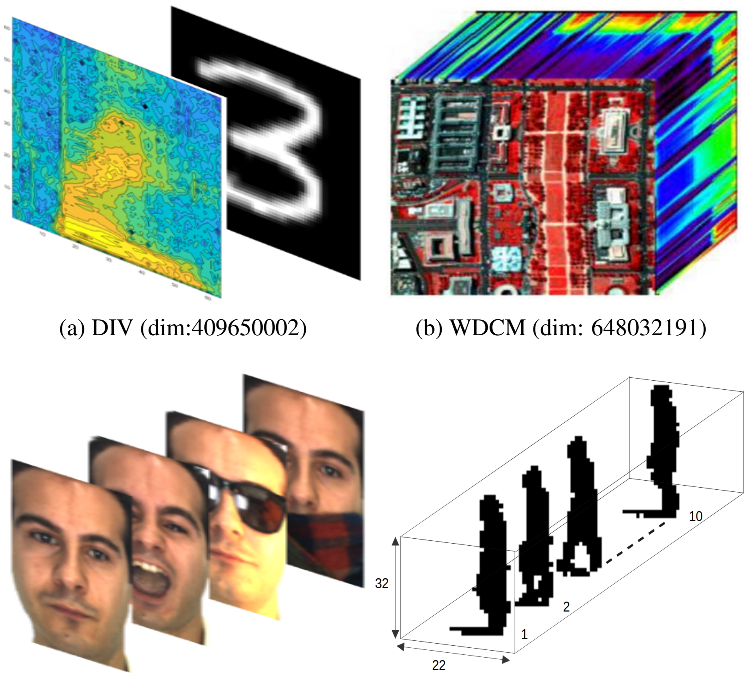
Inputs: $\mathscr{X} \in \mathbb{R}^{n_1 \times K \times n_3}$ (input data:third-order tensor),
Y (labels: c classes)
K (reduced dimension)
Output: $\mathscr{V}^* \in \mathbb{R}^{n_1 \times K \times n_3}$ (projective tensor)

$\widetilde{\mathscr{X}} = L(\mathscr{X})$
for $i = 1, \ldots, n_3$ **do**
$\widetilde{S}_W^{(i)}, \widetilde{S}_B^{(i)} \leftarrow$BuildScatters($\widetilde{X}^{(i)}$,$Y$)
$(\widetilde{V}^*)^{(i)} \leftarrow$NewtonLanczos($\widetilde{S}_W^{(i)}$,$\widetilde{S}_B^{(i)}$,$K$) (see Algorithm 1)
end for
$\mathscr{V}^* = L^{-1}(\widetilde{\mathscr{V}}^*)$

Algorithm 4 $*_L$TLDA - Ratio trace optimization

Inputs: $\mathscr{X} \in \mathbb{R}^{n_1 \times K \times n_3}$ (input data:third-order tensor),
Y (labels: c classes)
$\gamma > 0$ (regularization parameter)
Output: $\mathscr{V}^* \in \mathbb{R}^{n_1 \times K \times n_3}$ (projective tensor)

$\widetilde{\mathscr{X}} = L(\mathscr{X})$
for $i = 1, \ldots, n_3$ **do**
$\widetilde{S}_W^{(i)}, \widetilde{S}_B^{(i)} \leftarrow$ BuildScatters($\widetilde{X}^{(i)}$,Y)
$K \leftarrow rank(S_B)$
$(\widetilde{V}^*)^{(i)} \leftarrow$ eigs($(\widetilde{S}_W^{(i)} + \gamma I)^{-1}\widetilde{S}_B^{(i)}$,K,'lm')
end for
$\mathscr{V}^* = L^{-1}(\widetilde{\mathscr{V}}^*)$



(a) DIV (dim:409650002) (b) WDCM (dim: 648032191)

(c) AR (dim:62526003) (d) GAIT (dim:102473110)

Figure 1: Data sets.

Table 1: Gait data set : characteristics.

Data sets	Gallery (GAR)	A (GAL)	B (GBR)	C (GBL)
nb of seq.	731	727	422	419
nb of subj.	71	71	41	41
Diff. from the gallery set	-	View	Shoe	Shoe-view

5 Experimental results

5.1 Data sets

The experimental evaluation is based on five multidimensional data sets: The Digit Image Voice (DIV) data set, the Facial Recognition Technology (FERET) database , the AR Face (AR) database, the Washington DC Mall (WDCM) dataset and the HumanID gait (GAIT) data set. Figure 1 illustrates four examples of the studied data sets.

The WDCM dataset is a 191 band hyperspectral image of Washington DC Mall collected by the Hyperspectral Digital Imagery Collection Experiment (HYDICE) [10][1]. The whole image contains 1208307 pixels. From this image we defined 4 classes: ‘grass land’, ‘tree’, ‘roof’ and ‘road’ which

[1] http://lesun.weebly.com/hyperspectral-data-set.html

are hand-built and defined by 77 image blocs of pixels. We collected a total of 8032 blocs: 1894 for the 'tree', 1919 for the 'grass land', 2616 for the 'roof' and 1603 for the 'road'. A third-order tensor of size 648032191 is then built where the first dimension corresponds to the vectorization of the 77 image bloc.

The DIV dataset build a tensor representation of the digits 0-9 obtained from two modalities: visual and audio and based on the MNIST[2] and FSDD datasets[3], respectively. The MNIST dataset contains 60000 training and 10000 test grayscale images of handwritten digits, with dimensions of 28x28 pixels. The FSDD dataset consists of 500, 8 kHz recordings of English pronunciations of the digits 0-9. These recordings are of varying durations, with a mean of approximately 0.5s. We preprocessed the recordings by converting them into 6464 grayscale spectrograms. In order to harmonize the image sizes between modalities, the MNIST images are resized to 6464 pixels. A tensor representation is then generated from 5000 samples combining the two modalities where each sample is randomly selected both in the resized MNIST dataset and the FSDD data sets. We obtain a 3th-order tensor of size 409650002 where the first dimension defines the vectorization of the 6464 image.

The FERET database is a standard facial image collection including 14126 images of size 3232 from 1199 individuals with different view points [27][4]. In our experiment, we select a subset composed of 80 subjects where each of them having at least 10 images, resulting in 1145 images. We generate a third-order tensor of size 102411451.

The AR dataset contains over 4,000 color images corresponding to 126 people's faces (70 men and 56 women) [22][5] . Images feature frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). The pictures were taken under strictly controlled conditions. No restrictions on wear (clothes, glasses, etc.), make-up, hair style, etc. were imposed to participants. A subset of 100 subjects have been considered corresponding to a total of 2600 images of size 2525. A third-order tensor of size 62526003 was generated.

The GAIT dataset is build from the USF HumanID Gait Challenge data sets version 1.7 [29][6]. This data set is composed of 452 sequences from 74 subjects walking in elliptical paths in front of the camera. For each subject, there are three covariates: viewpoint (left/right), shoe type (two different types) and surface type (grass/concrete). In our experiments, we consider only the sequences corresponding to the *grass* type surface defining thus the gallery set. This dataset contains 731 sequences from 71 subjects (persons) and each subject has an average of roughly 10 samples available under the form of binary silhouette images of size 3232 (see Figure 1). Thus we define a 3th-order training tensor of size 102473110. The test set is based on three probe sets named A, B and C as detailed in Table 1. More precisely, the image acquisition conditions for the gallery set and each probe set are summarized in brackets after the data name in Table 1, where G, A, B, L, and R stand for grass surface, shoe type A, shoe type B, left view, and right view, respectively. There is no redundancy between the gallery set and each probe set, i.e., there are no common subjects and sequences between them.

[2] http://yann.lecun.com/exdb/mnist
[3] https://github.com/Jakobovski/free-spoken-digit-dataset
[4] https://old.datahub.io/dataset/feret-database
[5] http://cbcsl.ece.ohio-state.edu/ARdatabaseNew.html
[6] http://www.eng.usf.edu/cvprg/Gait-Data.html

Table 2: Performances on DIV dataset (9 categories).

Objective	Methods	ACC (%)	Times (sec)	DIM
trace ratio	Fisherfaces	83.08(±0.026)	0.33 (±0.013)	9
	*c-TDA	90.58 (±0.02)	0.45 (±0.018)	18
	*t-TDA	89.75 (±0.02)	0.46 (±0.026)	18
	DATER	87.98 (±0.02)	0.33 (±0.016)	928
	CMDA	89.43(±0.023)	0.42 (±0.058)	493
	UMDA	74.95 (±0.04)	11.44 (±0.08)	28
ratio trace	*c-TDA	89.33(±0.017)	1.79 (±1.9)	823
	*t-TDA	88.91 (±0.02)	1.50(±0.39)	815
	DGTDA	89.13(±0.017)	0.34 (±0.097)	968
	HODA	87.46(±0.022)	0.64 (±0.07)	956

Table 3: Performances on WDCM dataset (4 categories).

Objective	Methods	ACC (%)	Times (sec)	DIM
trace ratio	Fisherfaces	91.6(±0.019)	0.478 (±0.05)	3
	*c-TDA	95.05 (±0.01)	0.45 (±0.010)	384
	*t-TDA	95.51 (±0.01)	0.61 (±0.01)	564
	DATER	94.21(±0.016)	2.28 (±0.017)	150
	CMDA	96.65(±0.01)	2.11 (±0.054)	305
	UMDA	91.95 (±0.025)	118.08 (±3.52)	28
ratio trace	*c-TDA	93.78(±0.04)	0.75 (±0.08)	964
	*t-TDA	94.41 (±0.02)	1.01(±0.15)	1000
	DGTDA	94.35(±0.017)	2.60 (±0.09)	350
	HODA	94.45(±0.044)	4.42 (±0.77)	783

5.2 Competitors

We compare our approach with five supervised learning algorithms: PCA+LDA (Fisherfaces) [2], Discriminant analysis with tensor representation (DATER) [31, 32], Constrained Multilinear Discriminant Analysis (CMDA) [20], Direct General Tensor Discriminant Analysis (DGTDA) [20], Higher Order Discriminant Analysis (HODA) [26], Uncorrelated Multilinear Discriminant Analysis with regularization (UMLDA) [21]. Our approach will be tested with two tensor-tensor products: t-product and the c-product. When using the t-product, our approach will be referred as $_{*t}$-TDA and $_{*c}$-TDA for the c-product.

Fisherfaces' implementation is based on the ratio trace criteria and uses a vector to vector projection.

Table 4: Performances on FERET dataset (80 categories).

Objective	Methods	ACC (%)	Times (sec)	DIM
trace ratio	Fisherfaces	88.71($\pm$0.02)	0.61 ($\pm$0.02)	79
	*c-TDA	87.93 ($\pm$0.02)	1.51 ($\pm$0.02)	68
	*t-TDA	87.78 ($\pm$0.02)	1.54 ($\pm$0.016)	68
	DATER	81.23 ($\pm$0.03)	1.11 ($\pm$0.19)	45
	CMDA	75.46($\pm$0.02)	1.32 ($\pm$0.02)	559
	UMDA	78.2 ($\pm$0.03)	5.85 ($\pm$0.27)	28
ratio trace	*c-TDA	63.75($\pm$0.03)	6.08 ($\pm$1.24)	675
	*t-TDA	64.01 ($\pm$0.03)	5.50($\pm$1.44)	527
	DGTDA	64.46($\pm$0.04)	0.487 ($\pm$0.02)	607
	HODA	67.96($\pm$0.02)	1.38($\pm$0.61)	29

Table 5: Performances on AR dataset (100 categories).

Objective	Methods	ACC (%)	Times (sec)	DIM
trace ratio	Fisherfaces	92.25($\pm$0.014)	0.51 ($\pm$0.021)	99
	*c-TDA	94.36 ($\pm$0.02)	1.49 ($\pm$0.04)	294
	*t-TDA	95.58 ($\pm$0.01)	2.92 ($\pm$0.08)	240
	DATER	68.9 ($\pm$0.03)	4.44 ($\pm$0.16)	101
	CMDA	72.23($\pm$0.03)	5.13 ($\pm$0.17)	164
	UMDA	68.2 ($\pm$0.035)	13.46 ($\pm$0.73)	23
ratio trace	*c-TDA	33.86($\pm$0.03)	2.31 ($\pm$0.05)	956
	*t-TDA	34.48 ($\pm$0.03)	4.21($\pm$0.36)	438
	DGTDA	33.83($\pm$0.02)	1.47 ($\pm$0.077)	892
	HODA	34.88($\pm$0.04)	1.54 ($\pm$0.076)	984

In order to avoid the singularity problem of the within-class scatter matrix, PCA is beforehand computed reducing thus the dimension of the feature space. To set the output dimension, a classical heuristic is to retain the k eigenvectors that capture a certain percentage of the total variance. In all the experiments, we will consider at least 95% of the total variance. DATER, CMDA, DGTDA, HODA and UMLDA are multidimensional variants of LDA. The first four methods use a tensor to tensor projection while the last one uses a tensor to vector projection. DATER, CMDA, UMLDA and Algorithm 4 are formulated as a trace ratio problem while DGTDA, HODA and Algorithm 3 solve the ratio trace problem.

When the optimization problem is formulated as the trace ratio problem, a regularization parameter is used to avoid the SSS problem (singularity problem of scatter matrices). The regularization parameter is defined in the range $10^{-[0:7]}$ and the best value is estimated by k-fold cross validation while the dimension

Table 6: Performances on the Gait sequence (72 categories).

Objective	Methods	rank 1			rank 5		
		A	B	C	A	B	C
trace ratio	Fisherfaces	43.6	43.9	31.7	71.8	63.4	51.20
	*c-TDA	90.1	84.9	65.8	100	92.7	85.4
	*t-TDA	94.4	80.5	65.8	100	92.7	87.8
	DATER	69.0	70.7	48.8	93.0	82.9	68.3
	CMDA	80.2	73.7	53.6	95.8	80.5	70.7
	UMDA	77.5	68.3	41.5	94.4	80.5	73.2
ratio trace	*c-TDA	65.2	69.3	46.1	92.4	80.5	67.2
	*t-TDA	67.8	72.3	47.2	94.2	81.2	69.5
	DGTDA	63.4	73.2	44.0	90.1	82.9	68.3
	HODA	70.4	70.7	48.8	92.9	80.5	68.3

of the projective subspace is given by the rank of the between-scatter matrix S_B. When the optimization problem is formulated as the ratio trace problem, the dimension of the projective subspace is unknown and it is also determined by k-fold cross validation. In our study, k is set to 10 and the average accuracy (30 repetitions) is used as classification performance.

Tables 2–5 summarize the performances of the different methods when applied to DIV, WDCM, FERET and AR data sets, respectively. These tables show the values of the average accuracy (ACC) recorded by the methods (third column), the average training times (fourth column) and the maximal output dimensions (fifth column). First, from a general view, we observe that the proposed $*_L$-TLDA provide similar results whether it is based on the c-product or the t-product. Secondly, when formulated as the trace ratio problem, the proposed method clearly records better classification results with the other tensor decomposition methods based on the same objective such as DATER, CMDA and UMDA. It also outperforms the version based on the ratio trace objective and very clearly when the number of categories increases as in the AR data base (Table 5) and the FERET data base (Table 4). This result is also valid for the other competitors based on the ratio trace objective such as DGTDA and HODA.

Moreover, we observe that $*_L$-TLDA (trace ratio criterion) records competitive training times with the other studied methods. By using a tensor to vector projection strategy, UMLDA shows the highest complexity making this method very time consuming.

In a second experiment, we study the performance of the proposed algorithm on the Gait sequence. The identification performance is measured by the Cumulative Match Characteristic (CMC) as defined in [5] which plots identification rates within a given rank k. More precisely, rank k results report the percentage of probe subjects whose the true match in the gallery set was in the top k matches. The rank 1 and the rank 5 gait recognition results using the modified angle distance (MAD) [19] are presented in Table 6. As previously, $*_L$-TLDA formulated with the trace ratio criterion shows the best recognition rates on all the probe sets. When the ratio trace criterion is optimized, the results are markedly lower. This result confirms those of Tables 2–5. We can notice that Fisherfaces records the lowest recognition rates making it clear that a matricial treatment of this kind of data set is not well suited.

6 Conclusion

In this work, we proposed a new Tensor Linear Discriminant Analysis based on the concept of transform domain as defined in [13]. Considering any fixed, invertible linear transformation L, our $*_L$TDA procedure is based on the computation of a new $*_L$-family product such as the t-product or the c-product. In this context, we showed that the solution of our MDA can be obtained in a fully tensor form instead of a sequence of projective matrices, solutions of existing MDA methods. Another key aspect is that the obtained solution is the result of independent optimization problems easier to solve and more robust compared to existing MDA methods that are based on alternating optimization heuristics. The experimental evaluation based on these two products show similar classification performances with a slight advantage of the c-product in terms of training time. The experimental evaluations show that the choice of the optimization criterion, i.e., the trace ratio or the ratio trace, influences significantly the classification performances of our method. The conclusions of our experimental evaluation $*_L$TDA, based on the trace ratio criterion performs very well and outperforms most of the existing MDA methods.

Several issues remain to be investigated. First, the proposed MDA is based on the building of three-order tensors and its extension to higher-order tensors could be the subject of future work. Second, the concept of rank being clearly defined in traditional Linear Discriminant Analysis, it will be interesting to address this issue shortly in the framework of $*_L$TDA in order to bound the dimensionality of the solution.

Acknowledgements

The authors would like to thank the two referees for carefully checking the paper and providing a number of helpful remarks and suggestions.

References

[1] E. Alexandre-Cortizo, M. Rosa-Zurera, F. Lopez-Ferreras, *Application of Fisher linear discriminant analysisto speech/music classification*, EUROCON 2005 - The International Conference on "Computer as a Tool", **2** (2005), 1666–1669.

[2] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, *Eigenfaces vs. Fisherfaces: recognition using class specific linear projection*, IEEE Trans. Pattern Anal. Mach. Intell. **19** (1997) 711–720.

[3] A.H. Bentbib, M. El Guide, K. Jbilou, L. Reichel, *Global Golub–Kahan bidiagonalization applied to large discrete ill-posed problems*, J. Comput. Appl. Math. **322** (2017) 46–56.

[4] P. Bodesheim,A. Freytag, E. Rodner, M. Kemmler, J. Denzler, *Kernel null space methods for novelty detection*, Comput. Vis. Pattern Recognit. (2013), 3374–3381.

[5] N.V. Boulgouris and K.N. Plataniotis, D. Hatzinakos, *Gait recognition using linear time normalization*, Pattern Recognit. **39** (2006) 969–979.

[6] F. Dufrenois, J.C Noyer, *One class proximal support vector machines*, Pattern Recognit. **52** (2016) 96–112.

[7] M. El Guide, A. El Ichi, K. Jbilou, F.P.A Beik, *On tensor GMRES and Golub-Kahan methods via the T-product for color image processing*, Electron. J. Linear Algebra **37** (2021) 524–543.

[8] M. El Guide, A. El Ichi, K. Jbilou, *Discrete cosine transform LSQR and GMRES methods for multidimensional ill-posed problems*, J. Math. Model. **10** (2022) 21–37.

[9] N. Hao, M.E. Kilmer, K. Braman, R.C. Hoover, *Facial recognition using tensor-tensor decompositions*, SIAM J. Imaging Sci. **6** (2013) 437–463.

[10] W. He, H. Zhang, L. Zhang, H. Shen, *Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation*, in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **8** (2015) 3050–3061.

[11] X He, D Cai, P Niyogi, *Tensor subspace analysis*, Proceedings of the 18th International Conference on Neural Information Processing Systems (2005) 499–506.

[12] P. Howland, H. Park, *Generalizing discriminant analysis using the generalized singular value decomposition*, IEEE Trans. Patt. Anal. Pat. Anal. Mach. Intel. **26** (2004) 995–1006.

[13] E. Kernfeld, M. Kilmer, S. Aeron, *Tensor-tensor products with invertible linear transforms*, Linear Algebra Appl. **485** (2015) 545–570.

[14] M.E. Kilmer, K. Braman, N. Hao, R.C. Hoover, *Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging*, SIAM J. Matrix Anal. Appl. **34** (2013) 148–172.

[15] M.E. Kilmer, C.D. Martin, *Factorization strategies for third-order tensors*, Linear Algebra Appl. **435** (2011) 641–658.

[16] E. Kokiopoulou, J. Chen, Y. Saad, *Trace optimisation and eigenproblems in dimension reduction methods*, Numer. Linear Algebra Appl. **18** (2011) 565–602.

[17] T. G. Kolda, B. W. Bader, *Tensor decompositions and applications*, SIAM Rev. **3** (2009) 455–500.

[18] X.T. Li, M.K. Ng, *Solving sparse non-negative tensor equations: algorithms and applications*, Front. Math. China **10** (2015) 649–680.

[19] H. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, *MPCA: Multilinear principal component analysis of tensor objects*, IEEE Trans. Neural Netw. **19** (2008) 18–39.

[20] Q. Li and D. Schonfeld, *Multilinear discriminant analysis for higher-order tensor data classification*, in IEEE Transactions on Pattern Analysis and Machine Intelligence (2014) 2524–2537.

[21] H. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, *Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition*, IEEE Trans. Neural Netw. **20** (2009) 103–123.

[22] A. Martinez, R. Benavente, *The AR face database*, Tech. Rep. 24 CVC Technical Report, 1998.

[23] T.T. Ngo, M. Bellalij, Y. Saad, *The trace ratio optimization problem*, SIAM Rev. **54** (2012) 545–569

[24] M.K. Ng, R.H. Chan, W. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput. **21** (1999) 851–866.

[25] S. Pang, S. Ozawa, N. Kasabov, *Incremental linear discriminant analysis for classification of data streams*, IEEE Trans. Syst. Man Cybern. **35** (2005) 905–914.

[26] A. Phan, A. Cichocki, *Tensor decompositions for feature extraction and classification of high dimensional datasets*, Nonlinear Theory Appl. IEICE. **1** (2010) 37–68.

[27] P. J. Phillips, H. Moon, S.A. Rizvi, P. Rauss, *The FERET evaluation method for face recognition algorithms*, IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000) 1090–1104.

[28] V. Roth, *Kernel Fisher discriminant for outlier detection*, Neural Comput. **18** (2006) 942–960.

[29] S. Sarkar, P. J. Phillips, Z. Liu, I. Robledo, P. Grother, K. W. Bowyer, *The human ID gait challenge problem: Data sets, performance, and analysis*, IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005) 162–177.

[30] M.A.O. Vasilescu, D. Terzopoulos, *Multilinear image analysis for facial recognition*, in ICPR 2002: Proceedings of the 16th International Conference on Pattern Recognition (2002) 511–514.

[31] M. Visani, C. Garcia, J. Jolion, *Normalized radial basis function networks and bilinear discriminant analysis for face recognition*, IEEE Conference on Advanced Video and Signal Based Surveillance (2005) 342–347.

[32] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, H.J. Zhang, *Multilinear discriminant analysis for face recognition*, IEEE Trans. Ima. Proc. **16** (2012) 212–220.

[33] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang , H.J. Zhang, *Discriminant analysis with tensor representation*, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (2005) 526–532.